MODELING, IDENTIFICATION AND REAL TIME POSITION CONTROL OF A TWO-AXIS GIMBALLED MIRROR SYSTEM

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

BY

KARTAL ÇAĞATAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

FEBRUARY 2010

Approval of the thesis:

MODELING, IDENTIFICATION AND REAL TIME POSITION CONTROL OF A TWO-AXIS GIMBALLED MIRROR SYSTEM

submitted by **KARTAL ÇAĞATAY** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Süha Oral _____
Head of Department, **Mechanical Engineering**

Prof. Dr. Bülent E. Platin _____
Supervisor, **Mechanical Engineering Dept., METU**

Prof. Dr. Tuna Balkan _____
Co-Supervisor, **Mechanical Engineering Dept., METU**

Examining Committee Members:

Asst. Prof. Dr. İlhan Konukseven _____
Mechanical Engineering Dept., METU

Prof. Dr. Bülent E. Platin _____
Mechanical Engineering Dept., METU

Prof. Dr. Tuna Balkan _____
Mechanical Engineering Dept., METU

Asst. Prof. Dr. Melik Dölen _____
Mechanical Engineering Dept., METU

Asst. Prof. Dr. Afşar Saranlı _____
Electrical and Electronics Engineering Dept., METU

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: Kartal Çağatay

Signature :

ABSTRACT

MODELING, IDENTIFICATION AND REAL TIME POSITION CONTROL OF A TWO-AXIS GIMBALLED MIRROR SYSTEM

Çağatay, Kartal

M.Sc., Department of Mechanical Engineering

Supervisor: Prof. Dr. Bülent E. Platin

Co-Supervisor: Prof. Dr. Tuna Balkan

February 2010, 157 pages

This work focuses on modeling, parameter estimation, and real-time position control of a two axis Gimbaled Mirror System (GMS) which is designed and manufactured to move an IR spot generated by an Infra Red Scene Generator System (IRSGS) in two orthogonal axes (elevation and azimuth) within the IR scene which is also generated by the IRSGS.

Mathematical models of the GMS, the control system, and the disturbance torque originated from the movements of Flight Motion Simulator (FMS), on which the IRSGS will be mounted, are constructed using MATLAB®/Simulink® and MATLAB/Simulink/SimMechanics®. Parameter estimations of the GMS and control system elements are achieved using MATLAB/Simulink Parameter Estimation Tool®.

The controller tuning is performed using the developed mathematical models in MATLAB/Simulink environment. Optimized digital PID controllers are implemented in the real-time control system. Performances of the controllers for

both GMS axes are evaluated by both real system tests and simulation runs; and the results of these runs are compared. Controller performances under the effect of disturbances are analyzed by using the mathematical models developed in the MATLAB/ Simulink environment.

Keywords: Two Axis Gimbal System, System Identification, Modeling, Controller Tuning, Real-Time Control, MATLAB, LabView®

ÖZ

İKİ EKSENLİ BİR GİMBALLİ AYNA SİSTEMİNİN MODELLENMESİ, TANILANMASI VE GERÇEK ZAMANLI KONTROLÜ

Çağatay, Kartal
Yüksek Lisans, Makine Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Bülent E. Platin
Ortak Tez Yöneticisi: Prof. Dr. Tuna Balkan

Şubat 2010, 157 sayfa

Bu çalışmada, bir Kızıl Ötesi Görüntü Oluşturma Sistemi (KÖGOS) tarafından oluşturulan kızıl ötesi spotun, yine KÖGOS tarafından oluşturulan kızıl ötesi sahne içerisinde birbirine dik iki eksende (istikamet ve yükseliş) konumlandırılabilmesi için tasarlanan ve üretilen iki eksenli bir Gimballi Ayna Sistemi (GAS)'nin ve onun eksenlerinin gerçek zamanlı konum kontrolünün gerçekleştirilebilmesi için kurulan gerçek zamanlı kontrol sisteminin modellenmesi, oluşturulan modelde yer alan bilinmeyen parametrelerin belirlenmesi ve GAS eksenlerinin gerçek zamanlı konum kontrolü gerçekleştirilmiştir.

GAS'ın, kontrol sistemi elemanlarının ve KÖGOS'in üzerine oturtulacağı hareketli bir platform olan Uçuş Hareket Simülatörü (UHS)'nün hareketlerinden kaynaklanan bozanetkenlerin matematiksel modelleri ise MATLAB/Simulink ve MATLAB/Simulink/SimMechanics yazılımları kullanılarak oluşturulmuştur. GAS ve kontrol sisteminin parametreleri MATLAB/Simulink Parameter Estimation Tool kullanılarak elde edilmiştir.

Kontrol sistemi parametreleri MATLAB/Simulink ortamında geliştirilen matematiksel modeller kullanılarak belirlenmiş ve bu parametreler gerçek sisteme uygulanmıştır. Denetleyici başarımları her iki GAS ekseni için de hem gerçek sistem ile yapılan testler hem de MATLAB/Simulink ortamında geliştirilen benzetim modeli kullanılarak değerlendirilmiş ve elde edilen sonuçlar karşılaştırılmıştır. Denetleyicilerin bozanetkenlerin etkisi altında iken gösterdiği başarım ise MATLAB/Simulink ortamında geliştirilen matematiksel modeller kullanılarak analiz edilmiştir.

Anahtar Kelimeler: İki Eksenli Gimbal Sistemi, Sistem tanılanması, Modelleme, Kontrol Sistemi parametrelerinin optimizasyonu, Gerçek Zamanlı Kontrol, MATLAB, LabView

ACKNOWLEDGMENTS

First, I would like to express my gratitude to my thesis supervisor Prof. Dr. Bülent E. Platin and co-supervisor Prof. Dr. Tuna Balkan for their supervision throughout the completion of this work.

I would like to thank my colleagues in Roketsan Missiles Inc. for their valuable comments and assistance.

I also would like to thank TUBITAK BIDEB for their support all through this hard work.

I would like to thank my father for his belief in me, my mother for her compassion and my dear little sister for the warmest hugs.

Finally, I would like to thank Seda, for every single cup of coffee and for every single cup of tea that helped me survive sleepless nights. But most of all I would like to thank her for every single smile on her face which certainly makes the world turn happier.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

a_k	Effective length of link (k)
A_k	Intersection point of the axes along $\vec{u}_1^{(k)}$ and $\vec{u}_3^{(k-1)}$
b	Viscous damping coefficient
b_{azm}	Viscous damping coefficient of GMS azimuth axis
b_{elv}	Viscous damping coefficient of GMS elevation axis
$\hat{C}^{(k-1,k)}$	The rotation matrix between link k and link (k-1)
$G_c(z)$	Discrete time domain controller transfer function
$G_{OL}(s)$	Continuous time domain open loop transfer function
$G_{OL}(z)$	Discrete time domain open loop transfer function
$G_{\theta M}(s)$	Continuous time domain transfer function between the controlled output and the angular position
$\hat{I}$	Identity matrix
J	Mass moment of Inertia
J_{azm}	Mass moment of inertia of GMS azimuth axis
J_{elv}	Mass moment of inertia of GMS elevation axis
J_{11}, J_{22}, J_{33}	Diagonal elements of an inertia matrix
$\hat{J}_c^{(b)}$	Inertia matrix w.r.t. center of mass in body fixed reference frame
k	Spring constant
k_{azm}	Spring constant of GMS azimuth axis
k_{elv}	Spring constant of GMS elevation axis
K_a	Amplifier gain
K_d	Derivative gain

K_{dac}	Digital to Analog Converter (DAC) gain
K_e	Voltage constant of the motor
$K_{e_{azm}}$	Voltage constant of the motor used for GMS azimuth axis
$K_{e_{elv}}$	Voltage constant of the motor used for GMS elevation axis
K_i	Integral gain
$K_{i_{azm}}$	Integral gain of the PI current regulator of the servo amplifier used for GMS azimuth axis
K_p	Proportional gain
$K_{p_{azm}}$	Proportional gain of the PI current regulator of the servo amplifier used for GMS azimuth axis
$K_{p_{elv}}$	Proportional gain of the PI current regulator of the servo amplifier used for GMS elevation axis
$K_{rad\,2count}$	Conversion factor from radians to motor encoder counts
$K_{t_{azm}}$	Torque constant of the motor used for GMS azimuth axis
$K_{t_{elv}}$	Torque constant of the motor used for GMS elevation axis
L_{azm}	Inductance of each coil of the three phase brushless motor of GMS azimuth axis
L_{elv}	Inductance of the coil of the single phase brushless motor of GMS elevation axis
$M(s)$	Controlled output in s-domain
O_k	Intersection point of the axes along $\vec{u}_1^{(k)}$ and $\vec{u}_3^{(k)}$
R_{azm}	Resistance of each coil of the three phase brushless motor of GMS azimuth axis
R_{elv}	Resistance of the coil of the single phase brushless motor of GMS elevation axis

s_k	Linear displacement of link (k) w.r.t. link (k-1)
T	Controller update period
T_f	Friction torque
T_c	Coulomb friction torque
$T_{c_{azm}}$	Coulomb friction torque acting on GMS azimuth axis
$T_{c_{elv}}$	Coulomb friction torque acting on GMS elevation axis
T_{cable}	Torque acting on the axis because of the elasticity of the cable
$T_{d_{azm}}$	Disturbance torque acting on GMS azimuth axis
$T_{d_{elv}}$	Disturbance torque acting on GMS elevation axis
$T_{preload}$	Torque acting on the axis because of the potential energy accumulated in the cables at the axis zero position
$T_{preload_{azm}}$	Torque acting on the GMS azimuth axis because of preloading of the cable
$T_{preload_{elv}}$	Torque acting on the GMS elevation axis because of preloading of the cable
$\vec{u}_1^{(k)}, \vec{u}_2^{(k)}, \vec{u}_3^{(k)}$	Unit vectors along the x, y and z axes of the coordinate frame fixed to link k
$\tilde{u}_1, \tilde{u}_3$	The skew symmetric matrices
$\bar{u}_1, \bar{u}_3$	Column matrix representations of $\vec{u}_1$ and $\vec{u}_3$
V_{bemf}	Back electromotive force (emf) voltage
x_1	Distance between the first peak and the steady state value
x_2	Distance between the second peak and the steady state value
$\bar{\alpha}_b^{(b)}$	Angular acceleration about the body fixed reference frame, expressed in the body fixed frame
α_k	Twist angle of link (k)

$\overline{\alpha}_k^{(0)}$	Angular acceleration of link k w.r.t. inertial frame
$\overline{\alpha}_k^{(k)}$	Angular acceleration of link k w.r.t. its own body frame
δ	Second logarithmic decrement
$\hat{\Phi}_k$	Orientation Matrix of link k
$\overline{\omega}_b^{(b)}$	Angular velocity about the body fixed reference frame and written in the body fixed frame
$\tilde{\omega}_b^{(b)}$	Skew symmetric angular velocity matrix
ω_d	Damped natural frequency
$\overline{\omega}_k^{(0)}$	Angular velocity of link k w.r.t. inertial frame
$\overline{\omega}_k^{(k)}$	Angular velocity of link k w.r.t. its own body frame
ω_n	Natural frequency
θ_k	Angular displacement of link (k) w.r.t. link (k-1)
$\dot{\theta}$	Angular velocity
ζ	Damping ratio
ADC	Analog to Digital Converter
CPU	Central Processing Unit
DAC	Digital to Analog Converter
DC	Direct Current
D-H	Denavit-Hartenberg
DIP	Dual In-line Package
DSP	Digital Signal Processing
EMF	Electromotive Force
FMS	Flight Motion Simulator
FOR	Field of Regard
FOV	Field of View
FPGA	Field Programmable Gate Array
GMS	Gimbaled Mirror System

HIL	Hardware in the Loop
IR	Infra Red
IRSGS	Infra Red Scene Generator System
LQ	Linear Quadratic
LSB	Least Significant Bit
PD	Proportional-Derivative
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
PIV	Proportional-Integral-Velocity
PWM	Pulse Width Modulation
RF	Radio Frequency
TMS	Target Motion Simulator
UUT	Unit Under Test
ZOH	Zero Order Hold

CHAPTER 1

INTRODUCTION

1.1 Background and motivation

Today the missile technology is so advanced that it is not possible for a jet or a helicopter to escape from a guided missile just by making some maneuvers. Therefore, recently designed flying platforms are equipped with different kind of equipments, which are designed to avoid the fatal consequences of being hit by a missile, either by destroying the missile or deceiving its sensors. These equipments are called as countermeasures.

The classification of guided missiles according to how their seeker section works, leads to two main categories; IR (infrared) guided missiles and RF (radio frequency) guided missiles. Countermeasures designed to counter the guided missiles could also be divided into two main categories as IR countermeasures and RF countermeasures.

Infrared countermeasures are the devices designed to protect platforms from IR guided missiles by confusing the missiles' infrared guidance system. Flares are the most common aerial infrared countermeasures. They are used to counter IR guided surface-to-air or air-to-air missiles. They are mostly made of a spontaneous combustion metal which burns hotter than jet exhaust. Once the presence of an IR missile is sensed, flares are released by the aircraft. The aircraft would then pull away at a sharp angle from the flare and then reduce the engine power in attempt to

cool the thermal signature. This change in temperature and presence of new IR signatures, theoretically confuse the missile's seeker and hopefully cause the missile to follow flare(s) rather than the aircraft.

Flares could be dispensed from an aircraft in short intervals, one at a time, long intervals, or in clusters. To be able to counter different missiles effectively, flare dispensing programs should be optimized according to the type of missile. In order to be able to optimize the flare dispensing programs, some tests should be conducted with different dispensing programs to observe the behavior of the missile, whether it hits the target or not. However, conducting these tests with real systems (i.e. launching an IR guided missile to a target and observing if the aircraft would be able to avoid the missile hit by using predetermined flare dispensing programs) will inevitably result in personnel and expensive system loss, which is certainly not an option. Hardware in the Loop (HIL) Simulation is a method to conduct these tests in a safe and cost effective manner.

HIL simulation is a technique that is used increasingly in the development and test of complex real-time embedded systems, in which some components of the complicated system are simulated in software, while some other components are used as actual entities, between which appropriate interfaces are constructed to transform signals. Figure 1.1 shows a general schematic of a HIL simulation system.

Roketsan Inc., is developing a HIL simulation system which will be used as a tool for optimizing flare dispensing programs to effectively counter various kinds of IR guided missiles. Real missile hardware will be used in this HIL simulation as the unit under test (UUT). The HIL simulation architecture that is being developed by Roketsan Inc., is shown in Figure 1.2.

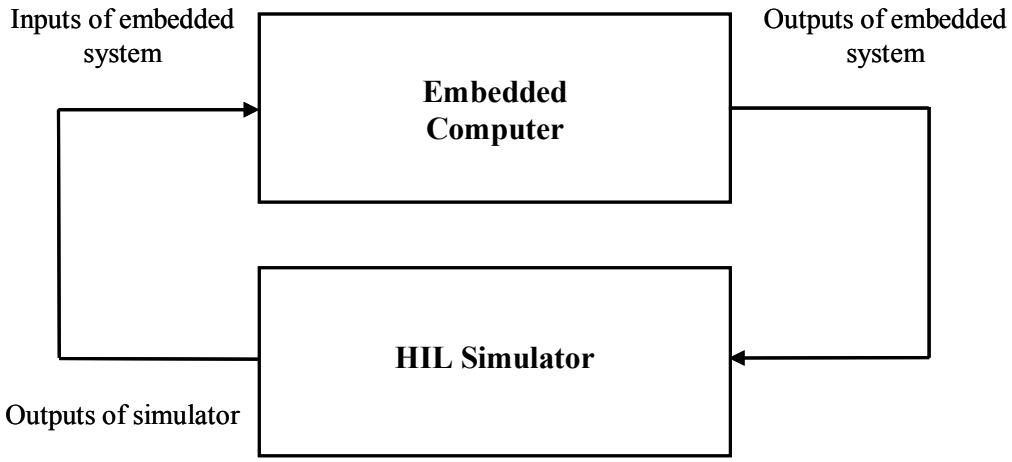

Figure 1.1. General schematic description of a HIL system

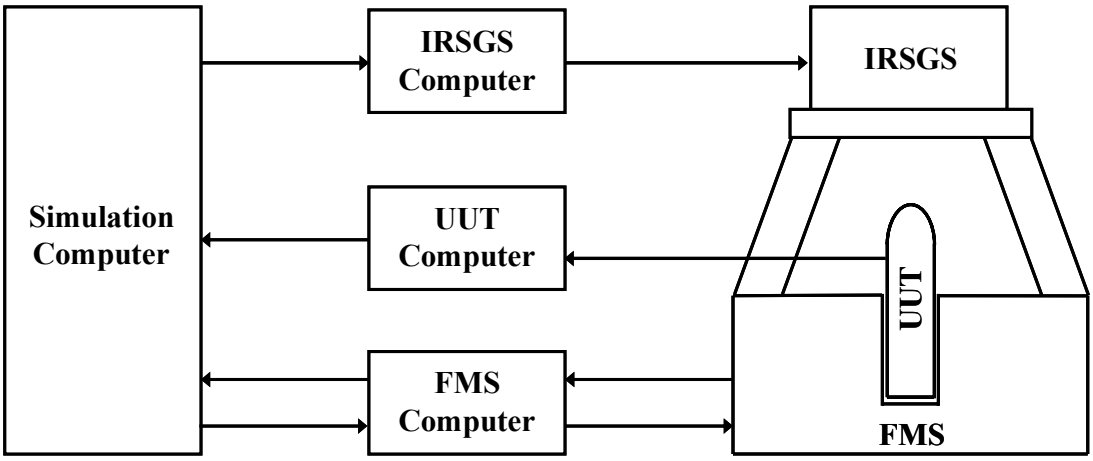

Figure 1.2. HIL architecture

A Flight Motion Simulator (FMS) will be used in the developed HIL simulation system in order to be able to simulate the motion of the missile in 3 axes; pitch, yaw

and roll. According to the IR Scene that is generated considering the scenario to be simulated, the seeker of the missile creates and sends appropriate signals to the CPU of the missile and the CPU decides the necessary actions and sends relevant signals to the actuators. However, in the HIL simulation, these signals are sent to the actuators of the FMS instead of the control actuation system of the missile so the FMS assures the missile to make the necessary motion. A typical 3 axes FMS is shown in Figure 1.3.

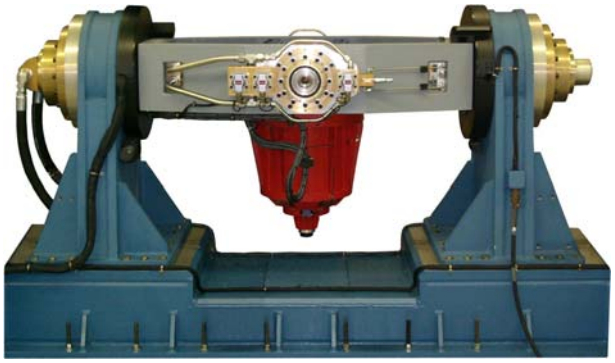

Figure 1.3. A typical three axes FMS [1]

For the HIL simulation system developed by Roketsan Inc. however, a 5 axes FMS will be used instead of a classic 3 axes type. This is because the head of the missile could also move independently from the body of the missile and so there is a necessity to move the IR scene according to the movements of the missile head. The seeker of a missile can only see the IR images which are within the range of its Field of View (FOV), so the angular magnitude of the background of the IR scene that will be generated should be equal to the FOV of the missile. On the other hand there is another concept called Field of Regard (FOR) for the missile, which indicates the

angular magnitude that the head of the missile could scan. There is a need to move the IR scene inside the FOR and this is why another two axes for the FMS are needed which are called together as Target Motion Simulator (TMS) on which the Infrared Scene Generation System (IRSGS) will be placed. With these 2 axes the FMS needed for the HIL simulation will be a 5 axes one. A typical 5 axes FMS is shown in Figure 1.4.

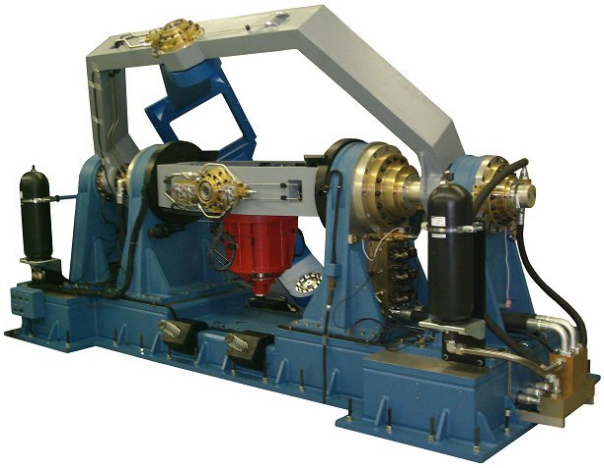

Figure 1.4. A five axes FMS [2]

The IRSGS, which is also being developed by Roketsan Inc. as a subsystem of HIL simulation system, is the system in charge of the formation of the IR spots, which are the elements of the IR scene that was calculated by the Central Control System, on the seeker of the missile in real-time. The dynamic and radiometric states of the platforms and the countermeasures, which are inside the FOV, at the angle that the head of the missile looks, are transferred from the Central Control System to the IRSGS. According to these transferred information, an IR Scene is formed and

reflected on the missile's seeker. The IRSGS is composed of some optical and electromechanical components which are used for forming the IR spots, changing the angular magnitudes and power of the IR spots, and moving the generated IR spots in two orthogonal axes.

A Gimbaled Mirror System (GMS) is procured by Roketsan Inc., for the purpose of moving the generated IR spot in azimuth and elevation axes to simulate the motion of the target relative to the seeker of the missile. Furthermore, a real-time control system, which composed of a real-time platform, a motion control card, an amplifier for each axis, and the GMS, is constituted in order to achieve real-time closed loop position control of GMS axes. Using the real-time control structure developed, accurate and yet rapid tracking of the position commands sent from the IRSGS computer for both GMS axes is crucial for fidelity and efficiency of the HIL simulations.

1.2 Literature survey

Literature survey conducted covers mainly two areas. Components used in infrared scene generation systems to simulate target motion in two orthogonal axes and position control algorithms for two axis gimbal systems.

1.2.1 Literature survey on components used to simulate target motion in two orthogonal axis for infrared scene generation purposes

The results of the literature survey made, reveals that the components which are extensively used to simulate target motion in two orthogonal axes for infrared scene generation purposes are; galvanometric scanners, polygon scanners, and gimbaled mirror systems.

1.2.1.1 Galvanometric scanners

Galvanometric scanners (galvos) are rotational devices which direct a light beam in one axis by rotating a mirror. The magnitude of the applied command signal determines the turn angle of the mirror. The combination of two galvanometric scanners can be used to position a light beam in two axes (Figure 1.5). Actually this combination is being used in two axes laser scanners.

Figure 1.5. Combination of two galvanometric scanners to form a X-Y scanner [3]

As an example of galvanometric scanners' usage in the scene generation systems to simulate the target motion, a supplemental projector system for the simulation of high intensity point source targets created in 1998 by AMCOM's MRDEC can be assessed [4]. This projector is capable of simulating very high intensity point sources which may be dynamically positioned anywhere within the seeker FOV. A single

point source is created using one laser diode, and it is steered independently in two axes within the FOV using a galvanometer scanner [4].

1.2.1.2 Polygon scanners

In Figure 1.6 several different polygon mirrors are shown. A polygon mirror is a multi-faced mirror. When coupled with a rotary actuator, it can be used (also named) as a polygon mirror scanner. Rotating polygon mirrors are being used for a wide variety of different tasks. For example; in LIDAR systems combinations of polygon mirrors are used for scanning the azimuth and elevation axes. In laser printers, they are used to direct the laser beam aimed at them through a system of lenses and mirrors onto the photoreceptor. In omni-directional barcode scanners they are used to produce a pattern of beams in varying orientations allowing them to read barcodes presented to it at different angles. On the other hand, there are some polygon mirrors with faces of different inclination. Using this type of polygon mirror, directing of a light beam in two orthogonal axes could be achieved.

Figure 1.6. Polygon mirrors[5]

An infrared scene projector, which uses a polygon scanner to create a two-dimensional scene across the UUT's FOV is explained in [6]. The Laser Diode Array Projector (LDAP) is a laser scanning system which was operational at US Army Missile Command's (MICOM) Research, Development, and Engineering Center (RDEC). It consisted of a linear array of diode lasers, an optical scanning system and drive electronics and was specifically designed for testing sensors which utilize a focal plane array (FPA). The output intensity of each diode laser is modulated with the polygon mirror to effectively generate a two-dimensional scene across the UUT's FOV. Three out of each group of four polygon facets are machined with different inclinations in order to obtain a scan pattern with a 4:1 field interlace.

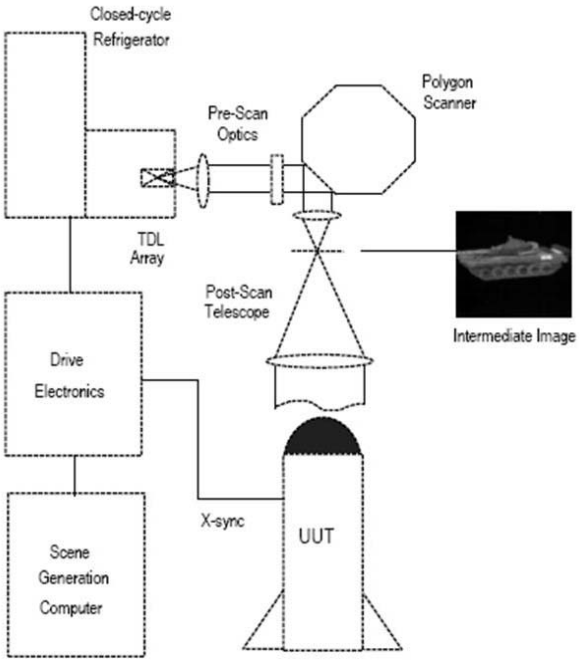

Figure 1.7. Schematic of LDAP [6]

Actually the polygon mirror scanner used in [6] is not used for simulating the target motion. The purpose of its usage is creating a scene on the UUT's seeker by scanning the scene with smaller parts. But since the x-y positioning of a light beam can be achieved by the same logic as in its usage in [6], the use of them is considered as a solution.

1.2.1.3 Gimbaled Mirror System

An example for the use of a gimbaled mirror system to simulate the target motion is given in [7]. That study explains the structure and working principles of an infrared scene generator system being developed by RAFAEL which is called TSG (Target Scene Generator).

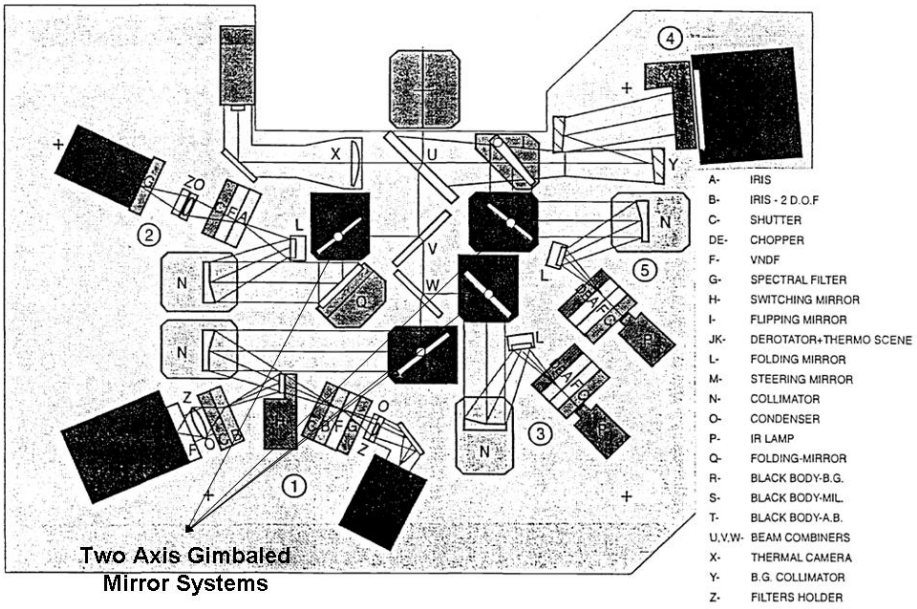

Figure 1.8. Configuration of the multiple elements on the IR scene generation table of TSG being developed by RAFAEL[7]

TSG is planned to be used for open loop testing of classical electro-optical missile seekers. A part of TSG is shown in Figure 1.8. From Figure 1.8 it is seen that at the end of each optical channel (before the beams enter beam combiners) there is a gimbaled mirror system to allow target pointing to any position within the 10° FOV. The gimbaled mirror system configuration used in TSG is shown in Figure 1.9.

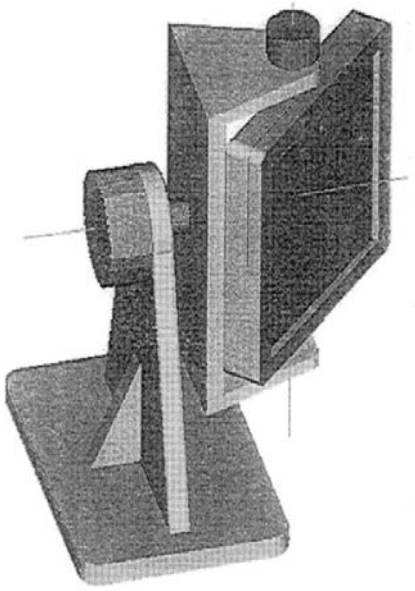

Figure 1.9. Gimbaled mirror system of TSG being developed by RAFAEL[7]

1.2.2 Literature survey on two axis gimbal position control algorithms

Two axis gimbals are used for a wide range of different applications so the position control of these systems is an important and widely studied topic involving various disciplines. There are several studies in the literature, which specifically sought to

develop position control algorithms in order to yield a better dynamic performance for the gimbal axes. In this section, some important studies on the topic are presented.

The thesis study conducted by Swarup [8] specifically sought to design and control a two-axis gimbal system which will be used for positioning a miniature video camera to perform visual tracking experiments for the object-catching architecture built around the MIT Whole Arm Manipulator at the Artificial Intelligence Laboratory of Massachusetts Institute of Technology in the USA. The scope of the work mainly included the design phase of the gimbal system, modeling kinematics, and dynamics of the system, design and implementation of two different controllers which are, proportional-derivative (PD) controller and computed torque controller.

Optics of the camera used allowed only a limited field of view to be sensed, so, to be able to accurately track the objects while maintaining maximum visual information flow, it was crucial to design a controller which rapidly positions the FOV in response to fast moving objects. On the other hand, the steady-state positioning error was less of a concern since the FOV allows for some error.

Due to the asymmetry in the camera inertial tensor, centripetal and Coriolis forces were expected to affect the dynamics of the system significantly. Because of this cross-coupling, a multivariable methodology was deemed necessary when designing an appropriate control system.

Since a very small steady-state error was not vital, a PD controller was selected. The position and derivative gains for the PD controller were tuned by applying a step function and modifying the gains. As the result of the optimization, overshoot less than 5% and 50 ms rise time achieved for a step input with $\pi/6$ amplitude for both axes. In that study, it is noted that even faster rise times were achievable, in order to prevent the system from saturating the amplifiers at high inputs, a compromise was

made by reducing the gains to allow the gimbal to achieve larger inputs although at shorter rise times.

The second controller designed was a computed torque controller. The purpose of the computed torque controller was improving tracking performance by explicitly accounting for the nonlinear dynamics of the system. Proportional and derivative feedback terms were also added to the control law to account for the inaccuracies in modeling the system inertia tensor and nonlinear terms. Thus the controller is decomposed into n decoupled double integrators of the form:

$$u_j = \ddot{q}_{dj} - k_{jD}\dot{\tilde{q}}_j - k_{jP}\tilde{q}_j \qquad j = 1,...,n \tag{1.1}$$

where

$\ddot{q}_{dj}$ Desired acceleration

u_j Control law

k_{jD} Derivative gain

k_{jP} Proportional gain

$\tilde{q}_j$ Joint displacement error

Since the acceleration is not physically measurable, numerical methods must be used to differentiate position and velocity data. These computations, however, introduce phase lag into the calculated acceleration, introducing error.

An elliptical trajectory was chosen in order to quantify each controller's performance with respect to the coupled nonlinear dynamics. As a result, it was observed that the bandwidth of the computed torque controller was obviously much lower than that of the PD controller, but the tracking performance was slightly

better. It was concluded that a high servo rate PD controller satisfactorily tracked moving objects with minimal overshoot. Although the computed torque controller employs the inverse plant dynamics in an attempt to linearize the system, the high frequency content of the controller resulted in a lower overall system bandwidth due to delays in computing. A PD controller was selected at the end since it is more robust and has much higher bandwidth. The gyroscopic coupling effects of the gimbal system were insignificant within the motion space necessary for visual sensing.

The results of the reviewed study show that, depending on the severity of the disturbances that affects the system; a simple control algorithm like PD control can provide better results than a more complex control algorithm which takes the disturbances and uncertainties affecting the system into consideration. An in depth analysis of the disturbances should be made before deciding the necessity of implementation of a complex control algorithm.

Another important study [9] mainly considers the performance of sliding mode control on a two axis gimbal system which is typical to tactical missiles. The objective of the gimbal control system is to follow a desired trajectory developed in real-time as rapidly as possible with a minimum steady state error. The scope of the study includes modeling of the rotational motion, design of sliding mode control, and evaluation of the performance of the controller via simulation.

Euler's equation of motion was used for modeling the rotational motion of the system. Since the antenna mounted on the gimbal post was modeled as a circular plate, the motion for the pitch and yaw axes were not coupled. Because of this, only one axis, which is the pitch axis, was considered in the study. Viscous damping, friction and the torques generated by the missile's motion were also taken into account. The friction torque and the torque produced on the pitch axes due to missile

motion were considered as unknown disturbances. The combined motor and pitch axis dynamics were obtained in state variable form as

$$\dot{x}_1 = x_2 \tag{1.2.a}$$

$$\dot{x}_2 = \varphi_0(x_1, x_2) + \Delta\varphi(x_1, x_2) + \{b_0(x_1, x_2) + \Delta b(x_1, x_2)\}u \tag{1.2.b}$$

$$y = x_1 \tag{1.2.c}$$

where

x_1, x_2 …….State variables

u …….Control law

$\varphi_0(x_1, x_2)$ Nominal value of $\varphi(x_1, x_2)$ assuming all motor parameters, moments of inertia and other values are known. The friction torque and disturbance torques are not included.

$\Delta\varphi(x_1, x_2)$ Disturbances and deviations in the nominal values

$b_0(x_1, x_2)$ Nominal value of $b(x_1, x_2)$

$\Delta b(x_1, x_2)$ Unknown variations in the nominal values of $b(x_1, x_2)$

In the reviewed study, the sliding mode controller design was performed following the steps below:

1) A sliding surface of the form $\sigma = \tilde{x}_2 + c_1\tilde{x}_1$ was selected such that the closed loop system motion on this surface exhibits a desired behavior regardless of plant uncertainties and disturbances. $\tilde{x}_1$ and $\tilde{x}_2$ are given by the relations:

$$\tilde{x}_1 = x_1^* - x_1 \tag{1.3.a}$$

$$\tilde{x}_2 = x_2^* - x_2 \tag{1.3.b}$$

where * denotes desired system behavior and c_1 was selected considering the required settling time.

2) A positive definite candidate to the Lyapunov function was selected as

$$V = \frac{1}{2} b^{-1}(x_1, x_2)\sigma^2 \tag{1.4}$$

3) For the sliding surface to be reached in a finite amount of time, the derivative of the Lyapunov function candidate should satisfy the following inequality:

$$\dot{V} < -\rho_1 |\sigma|, \qquad \rho_1 > 0 \tag{1.5}$$

A suitable control law, which guarantees the above inequality holds true, is decided.

The pitch axis with sliding mode control was simulated by using a software model. A realistic desired trajectory was chosen. Simulations ran for both nominal case where the disturbances and deviations from the nominal values are zero, and for the case with disturbances. When the simulation results are examined, it can be seen that the developed sliding mode controller provided a very accurate tracking even in the presence of disturbances and uncertainties.

The results of this particular study show that the sliding mode control is a powerful method when there are uncertainties in the system parameters or unmodeled disturbances. Without knowing anything about the disturbances it rejects them in an effective manner. Because of the switching behavior of the control law, it is not sensitive to the disturbances and provides a rapid but yet robust response.

Another study [10] presents two different advanced controller design methods, namely, robust inverse dynamics control and adaptive control, for motion control of a two degree of freedom gimbal which will be attached to an aviation vehicle and will be used for air surveys. Due to dynamic modeling errors, friction and disturbances from the outside environment, which may degrade the tracking accuracy of an airborne gimbal, controllers that are more advanced than a proportional-integral-derivative (PID) controller was thought to be necessary. A stabilizer or rate sensor was mounted on the base of the system to measure the disturbance. Tracking and disturbance rejection performances of the designed controllers were observed by both conducting experiments on the system and simulation runs with various reference commands and with disturbances due to all kinds of disturbances possible. The results showed that both robust inverse dynamics control and adaptive control performed tracking and disturbance rejection satisfactorily. Another study [11] presents the implementation of sliding mode control for the same system used in [10]. The results of this study showed the effectiveness of the proposed sliding mode controller in rejecting the disturbances.

The thesis study conducted by Skoglar [12] specifically sought to find a solution to the trajectory planning and motion control problem of a two axis camera gimbal which hosts an experimental sensor system consisting of an IR sensor, a video camera and an integrated navigation system. Two different control strategies, PID with anti-windup and linear quadratic (LQ), were implemented and tested for the motion control of the gimbal system. The challenge was to perform control that responds quickly, but do not excite the damping flexibility too much. The LQ-controller used a linearization of the dynamic model to fulfill these requirements.

From the simulation results it was observed that when PID control algorithm was used, the reference tracking performance was not so good since the flexibility causes large overshoots. Thus, it was concluded that the PID-regulator is inadequate for control of mechanical structure with flexibility. Instead, an LQ regulator was

developed to handle the flexibility. Since the control design requires a linear model of the system, a linearization of the nonlinear dynamic model was performed. The simulation results showed that the LQ-regulator takes the flexibility into consideration and the results with the LQ-regulator are better than the results with the PID-regulator.

Other than these studies, Rzasa [13] designed proportional-integral-velocity (PIV) control with velocity and acceleration feedforward for rapid and accurate position control of a two axis gimbal, which was used for moving a transceiver in directional wireless communications systems.

The gimbal systems considered in the reviewed studies are generally relatively large systems which house large payloads while the two axis gimbal system used in this thesis study is a relatively small one. Furthermore, the production of GMS used in this study is a high quality one with negligible unbalance for the axes and very low friction values in both axes. Also, since no transmission element (such as gear, belt, etc.) is used to drive the GMS axes, backlash is not a problem. So, no control action is needed to compensate the bearing friction, backlash or unbalance of the axes.

The systems in [9], [10], [11] are expected to be affected by disturbance torques induced by the motion of the base on which they are mounted. Since the GMS will be mounted on a moving FMS, some disturbance torques will be induced also on the GMS axes. However, since the FMS motions are not expected to be severe and no serious vibrations will be present, advanced controller algorithms like sliding mode control, adaptive control or robust inverse dynamics control, which are used in studies [9], [10], [11], are not thought to be necessary. Instead, a digital PID controller is decided to be used. Swarup showed in [8] that a PD controller can be adequate for position control of gimbal axes.

1.3 Research objectives

Three main objectives of this study can be given as:

The first objective is to develop a detailed model of the real-time control system using MATLAB/Simulink environment. This task includes developing models for each component of the control system. Some tests and analyses should be performed to decide the phenomenon to be modeled. Also an estimation of some parameters should be carried out since values for some parameters are unknown.

The second objective is to mathematically model the disturbance torques that are imposed on the GMS axes because of the motion of FMS axes. The rotational movements of FMS axes cause torques acting on GMS axes dictated by the angular momentum equation. These torques are uncontrolled torques hence they behave as disturbances on the control system. Their effects on the performance of the controllers could be observed by conducting tests. These tests should include mounting GMS on a motion simulator and observing the GMS axes motions while moving the motion simulator by different motion profiles. However, since the angular velocity and angular acceleration values attained by the motion simulator would be high for the tests to be performed it is not safe to perform these tests without having a clue about how the GMS axes will behave under these conditions. In addition, using a simulation model is certainly less time-consuming than performing tests with real equipment. If a high fidelity model on the kinematics and dynamics of the system is developed, several different operating conditions could be analyzed within relatively short time intervals.

The third and last objective is to optimize the controller parameters of the motion control card that will be used for real-time position control of GMS axes, considering the time domain requirements defined for motion of GMS axes. The

performances of the controllers for static environment are intended to be evaluated by using the developed MATLAB/Simulink model of the system and also by performing tests with the real system. The performances of the controllers under dynamic conditions are intended to be evaluated by using the devised MATLAB/Simulink model.

1.4 Thesis outline

Thesis study is composed of seven chapters;

Chapter 1 is the introduction chapter, which explains the background, motivation and objectives of this study as well as giving some basic definitions related to the study. The published studies in literature on methods for position controlling of the axes of a two axis gimbal (pitch-yaw gimbal) are reviewed and discussed. Methods for simulating target motion in two orthogonal axes for infrared scene generation purposes are also reviewed and explained in this chapter.

In Chapter 2, real-time control system elements (Gimbaled Mirror System (GMS), the motion control card, amplifiers, and the real-time platform) are explained and technical data for these elements are given.

Modeling of the whole system dynamics is explained in Chapter 3. The model of each component of the real-time control system developed by using MATLAB/Simulink is explained in detail. Mathematical expressions for disturbance torques acting on GMS axes because of the motion of FMS axes are developed analytically by using Euler equations for rigid body motion and these expressions are verified by using a model developed by using MATLAB/Simulink/ SimMechanics modules. Denavit-Hartenberg convention is used for kinematics analysis of the FMS-GMS system.

Chapter 4 covers the parameter estimation of the unknown parameters of the system in which MATLAB/Simulink Parameter Estimation Tool is used. Outputs are compared for the real system and the MATLAB/Simulink model, in which the estimated parameters are used.

Chapter 5 explains the parameter optimization of the digital PID controller of the motion control card for both GMS axes. First, the requirements for the position control of both GMS axes are defined. Then the discrete time transfer functions between the voltage outputs of the motion control card and angular positions are obtained for both axes since the controller design is performed in discrete time domain. Finally, the PID controller parameters' optimizations for both axes are achieved by using pole placement method via MATLAB/SISO Design Tool®

Chapter 6 explains the assessment of performance of the digital PID controllers, whose parameters are optimized, for both GMS axes by using the simulation model developed using MATLAB/Simulink and also by conducting some tests with the real system.

Chapter 7 summarizes the whole study, presents the conclusions arrived, and gives recommendations for the future work.

CHAPTER 2

REAL TIME CONTROL SYSTEM ARCHITECTURE AND COMPONENTS

To achieve a real-time closed-loop position control of the two GMS axes independent from each other, a real time control system is constituted. The established real-time control system consists of a GMS (axis gimbals, actuators and encoders), a motion control card, and an amplifier for each GMS axis. The control system architecture is shown in Figure 2.1.

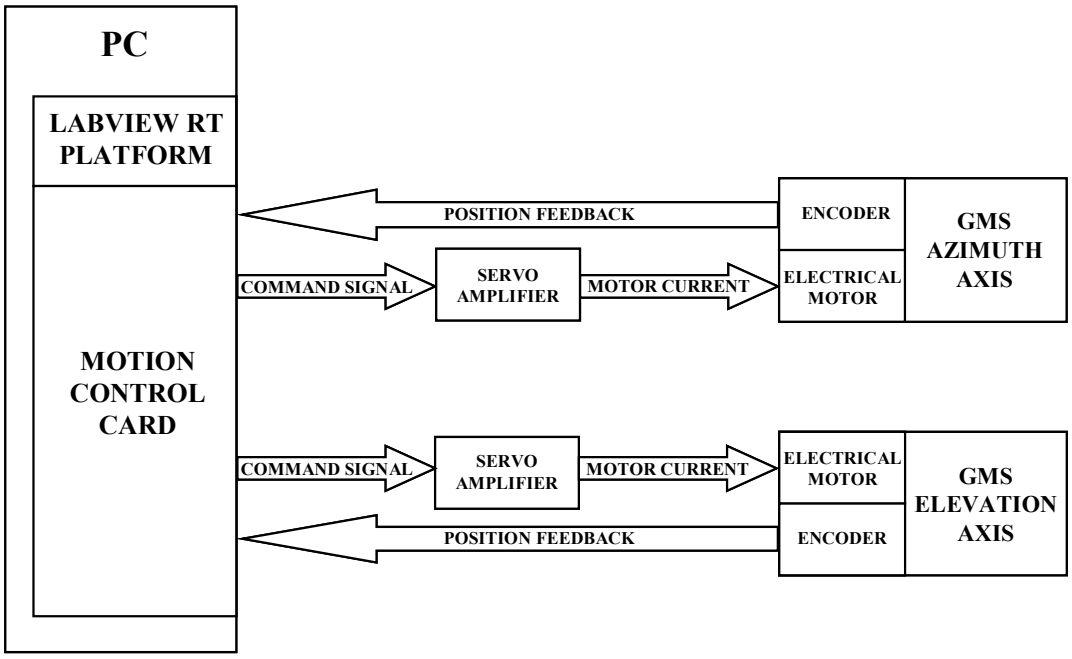

Figure 2.1. Real-time control system architecture

2.1 Gimbaled Mirror System (GMS)

The GMS consists of a stationary base and two gimbals for azimuth and elevation movements producing a motion in two orthogonal axes. Figure 2.2 depicts the main components of GMS. The maximum dimensions for GMS along three orthogonal axes (azimuth axis, elevation axis and third axis perpendicular to the first two) are 100, 138, and 108 mm, respectively. The total mass of the system is approximately 1 kg. The mass moment of inertia values of the GMS axis gimbals about center of mass of the axis gimbals w.r.t. the body fixed coordinate frames are supplied by the manufacturer, which were obtained from the solid model of the GMS.

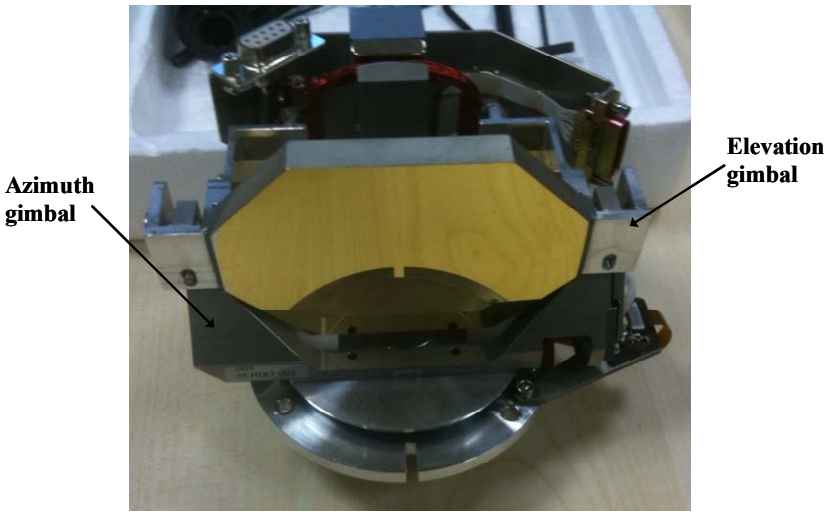

Figure 2.2. Gimbaled mirror system

The unbalances in both axes are negligible so it is possible to activate these axes separately. For balancing the parts, counter-masses were added to the elevation

frames. These counter-masses are made of brass and can be shifted in a slotted hole to balance the assembly.

2.1.1 Actuators and encoders

The elevation axis of GMS is actuated by a single phase brushless voice coil actuator while, a 3-phase brushless pancake motor is used to drive the azimuth axis. Since the actuator used for the elevation axis is a single phase system, there is no need for commutation. However, the 3-phase brushless motor used for the azimuth axis should be driven by commutation. Sinusoidal commutation is applied to provide smooth and precise control of the motor. In order to get satisfactory results from the sinusoidal commutation, an accurate measurement of rotor position is required. A high resolution encoder is used as position sensor to satisfy this requirement.

Optical, rotary incremental encoders are used for both the elevation and azimuth axes. The encoders have three incremental output signals; 2 channels with complements (sin/cos signal), in quadrature and an index gated signal (ref signal for 0-position). The encoders have 2,000,000 counts per revolution. Unfortunately, no detailed technical data for the encoders or the actuators is available since they were not provided by the supplier.

2.2 Motion control card and real-time platform

An NI PCI-7358 manufactured by National Instruments Corporation, which is shown in Figure 2.3, is used as the motion control card for position control of the GMS axes. It can perform closed loop motion control of servo axes in a real time environment. It has a 32 bit CPU which runs a real-time operating system, a DSP

unit for the closed loop control and a custom FPGA that performs the encoder interfacing.

Servo axes can be used to control either brushed or brushless DC motors. An axis which is used for the motion control of a servo motor, consists of at least a trajectory generator, a PID control block, an encoder or ADC channel feedback and one or two DAC outputs depends on the type of the motor which is being driven.

The trajectory generator generates the path from initial position to desired position, considering the set acceleration and velocity values for an axis and it feeds the instantaneous position to the PID loop each sampling period. The sampling period can be set as some discrete values between 62.5 microseconds and 5 milliseconds range. However, when all eight axes for the motion control card are enabled, the minimum attainable controller update period becomes 250 microseconds. Since all eight axes of the card will be used for HIL applications, the minimum controller update period will be 250 microseconds. So the controller update period is set as 250 microseconds and this value is used all through the study.

The discrete time transfer function of the digital PID controller of the motion control card is not given among its specifications. It should be obtained by performing tests with the motion control card. The DAC of the card is bipolar 10 Volts and the resolution of the DAC is 16 bits (0.000305 Volts/LSB). The motion control card communicates with the real-time platform through PCI bus. Detailed technical specifications of the motion control card are given in Appendix A.

The real-time platform is the computer which hosts the motion control card. It communicates with the motion control card through PCI bus. The LabView 8.6 RT Module runs on the real-time platform.

Figure 2.3. Motion control card used for motion control of GMS [14]

2.3 Amplifiers

An S16A8 three phase sinusoidal brushless servo amplifier manufactured by Advanced Motion Controls and shown in Figure 2.4 is used to drive three phase brushless motor for the GMS azimuth axis. Since the sinusoidal commutation is performed by the motion control card for the three phase brushless motor using the position feedback data from the encoder for azimuth axis, an amplifier which does not perform any commutation is selected for this axis. For the elevation axis, the same amplifier is used.

The additional characteristics of the amplifier used for both GMS axes are as follows: It is a current (torque) mode amplifier; it closes an internal current loop while the motion controller closes the outer position loop; there are total of two proportional-integral (PI) current regulators in each amplifier for each of two motor

phases. Their purpose is to generate a current value to pass through the motor windings, which is proportional to the voltage value of the command signals from the motion control card. The current in the third motor winding is the negative sum of the currents in the other two windings. The amplifier gain for the amplifier used is 1.6 Amperes/Volt, which means the amplifier tries to output a current value which is 1.6 times the voltage value which is applied to the input pin of the amplifier. PI regulators compare the sinusoidal current command signals from the motion controller with the actual currents flowing through the motor windings and form their outputs by manipulating these current errors with the proportional and integral gains. There are two DIP (Dual In-line Package) switches on the amplifier to increase/decrease the proportional gain and activate/deactivate the integral gain of the PI current regulators. The outputs of the PI current regulators are fed to pulse width modulation (PWM) blocks and PWM block outputs are fed to the motor terminals. PWM switching frequency of the amplifier is 33 kHz. Detailed technical specifications of the amplifier are given in Appendix B.

Figure 2.4. Amplifiers used for motion control of GMS [15]

CHAPTER 3

MODELING OF SYSTEM DYNAMICS

The mathematical models for each of the real time control system components including GMS are developed in MATLAB/Simulink environment. These models are explained throughout this chapter. Furthermore, the mathematical modeling of torques acting on GMS axes due to the motion of FMS axes is represented. Tests, which are conducted to decide which effects should be included in the constructed models, are also explained.

3.1 GMS model

The GMS model developed in MATLAB/Simulink environment is composed of actuator models, Coulomb and viscous friction models, and cable models. Each submodel is explained in detail in the following sections.

3.1.1 Actuator model

The voltage reduction at actuator terminals that is proportional to the speed of the rotor due to back electromotive force (emf) effect is included in the developed actuator models. The proportionality between the voltage reduction and the angular speed of the rotor is defined by the voltage constant of actuator denoted as K_e. The voltage constants for both actuators are unknown.

The relationship between the voltage across actuator terminals and the current through the actuator coils is defined by the inductance and resistance of the stator windings. Hence, electrical dynamics of the actuators are modeled using inductance and resistance values of the stator windings. In electrical actuators the conversion from the electrical system to the rotational mechanical system is defined by the torque constant of the actuator thus electrical system to mechanical system conversion in the actuators is modeled by torque constants of the actuators. The torque constants for azimuth and elevation axes actuators are also unknown. The dynamics of the actuators are expressed with the following equations and MATLAB/Simulink model developed for actuators is shown in Figure 3.1.

$$V_{bemf}(t) = K_e \dot{\theta}(t) \tag{3.1.a}$$

$$V(t) - V_{bemf}(t) = Ri(t) + L\int i(t)dt \tag{3.1.b}$$

$$T(t) = K_t i(t) \tag{3.1.c}$$

where

V_{bemf}	Back emf voltage [V]
K_e	Voltage constant of actuator [V.s/rad]
$\dot{\theta}$	Angular velocity [rad/s]
V	Voltage applied across actuator terminals by amplifier [V]
R	Resistance of actuator coil [Ω]
L	Inductance of actuator coil [H]
i	Current through actuator coil [A]
T	Torque applied by actuator [N.m]
K_t	Torque constant of actuator [N.m/A]

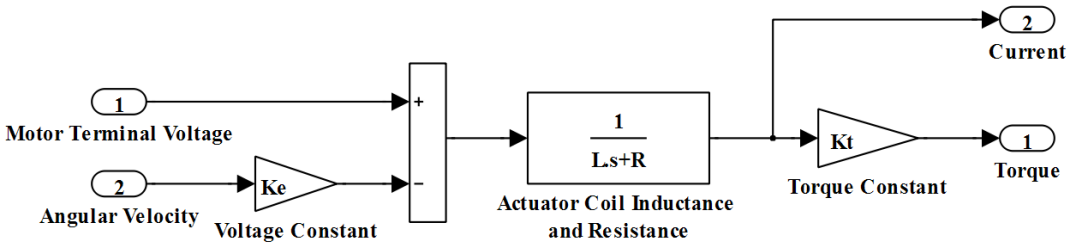

Figure 3.1. Actuator model developed in MATLAB/Simulink environment

3.1.2 Friction model

Friction model developed in MATLAB/Simulink environment is shown in Figure 3.2. The bearing of the azimuth gimbal is a duplex bearing in O-arrangement. The elevation gimbal is mounted on the azimuth axis gimbal with two small diameter duplex bearings, one on the left and one on the right side. The relative motion between the inner and outer rings of the bearings results in Coulomb friction torque and viscous friction torque to act on the axis gimbals. Static friction (stiction) is neglected in the developed model. The relation between the friction torque acting on the axes gimbals and the relative motion at the bearings is modeled with the equation:

$$T_f = T_c sign(\dot{\theta}) + b\dot{\theta} \tag{3.2}$$

where

T_f Friction torque [N.m]

T_c Coulomb friction torque [N.m]

b Viscous friction coefficient [N.m.s/rad]

$\dot{\theta}$ Angular velocity [rad/s]

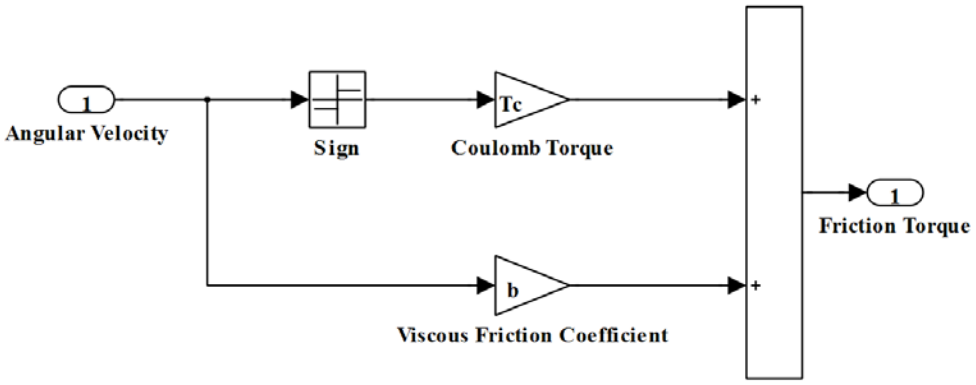

Figure 3.2. Friction model developed in MATLAB/Simulink environment

Actually, in order to obtain a smoother friction model the first term in (3.2), which is $T_c sign(\dot{\theta})$, could be replaced by a term like $(2T_c/\pi)\tan^{-1}(a\dot{\theta})$ where, a is a relatively large constant. However, since it is known that the friction values for both axes are very low, using this expression is not expected to make a significant difference between the results

3.1.3 Cable model

For both axes of GMS, it is observed that, when the axes are rotated to an arbitrary position and released from that position without any initial angular velocity, they move. To find the reason of this behavior, some tests are performed. Each axis is

commanded to and then released from -1.75 degrees for five times one right after the other. The same procedure is repeated for 1.75 and 0 degrees with both axes. The plots for GMS azimuth axis are shown in Figure 3.3, through Figure 3.5 while the plots for GMS elevation axis are shown in Figure 3.6 through Figure 3.8.

As can be seen from the graphs, a position depending torque is acting on both axis. When the axis gimbals are released from the same position, they stop at a different position in each run. Actually from the graphs it can be said that the axes gimbals do not even stop. Another conclusion that can be extracted from the graphs is that, in each run, the axes go nearer to the point that they are released than the runs before. The reason for these behaviors of the GMS axes is due to the dynamic interaction of GMS with the cables coming from the motor and encoder connectors on the GMS structure and attached to the axes.

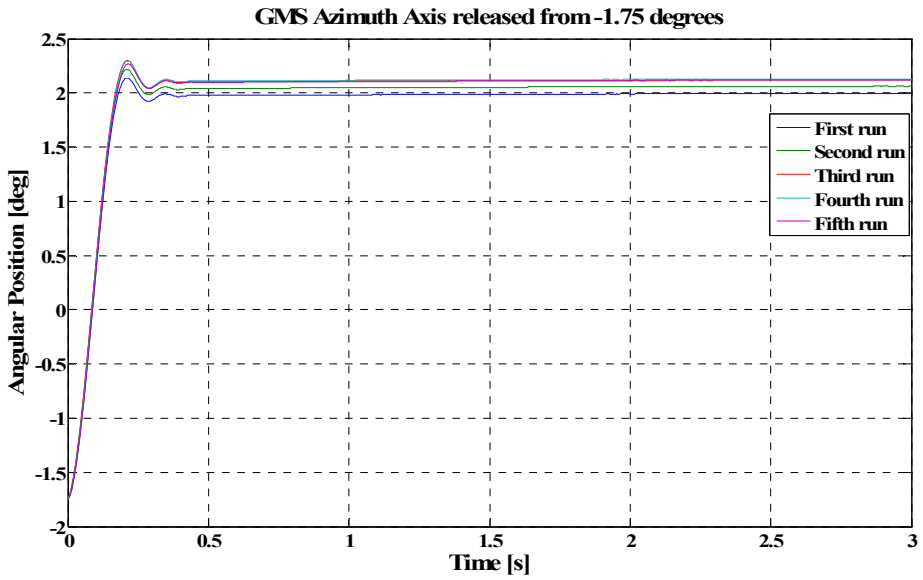

Figure 3.3. GMS azimuth axis released from -1.75 degrees

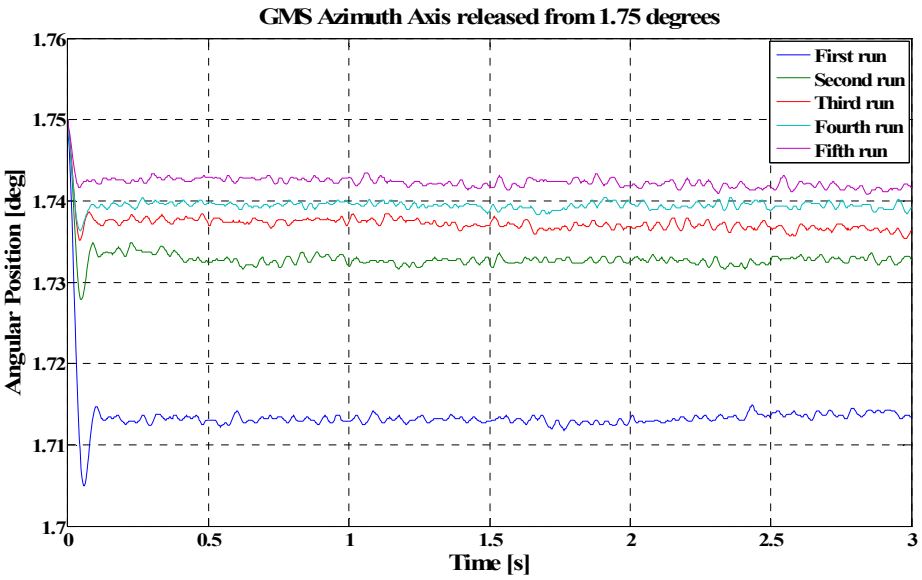

Figure 3.4. GMS azimuth axis released from 1.75 degrees

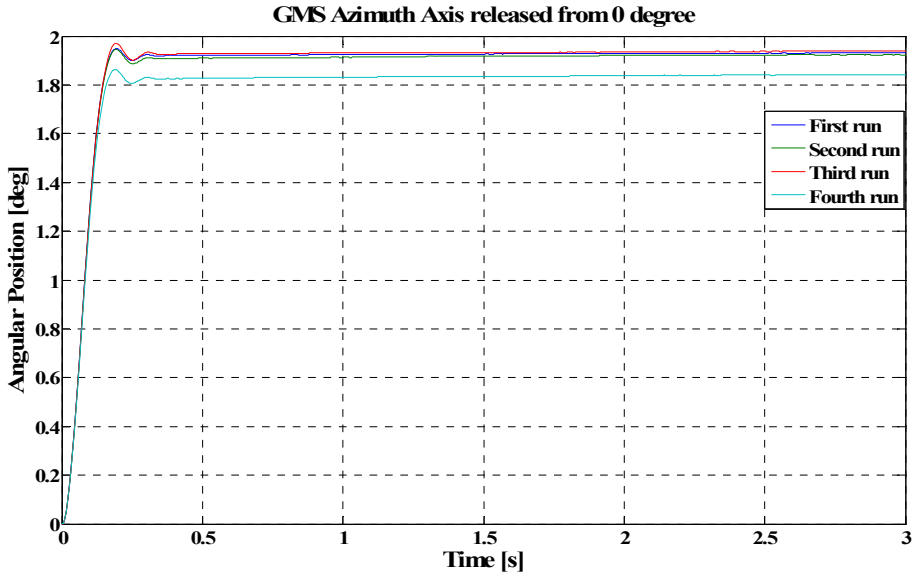

Figure 3.5. GMS azimuth axis released from 0 degree

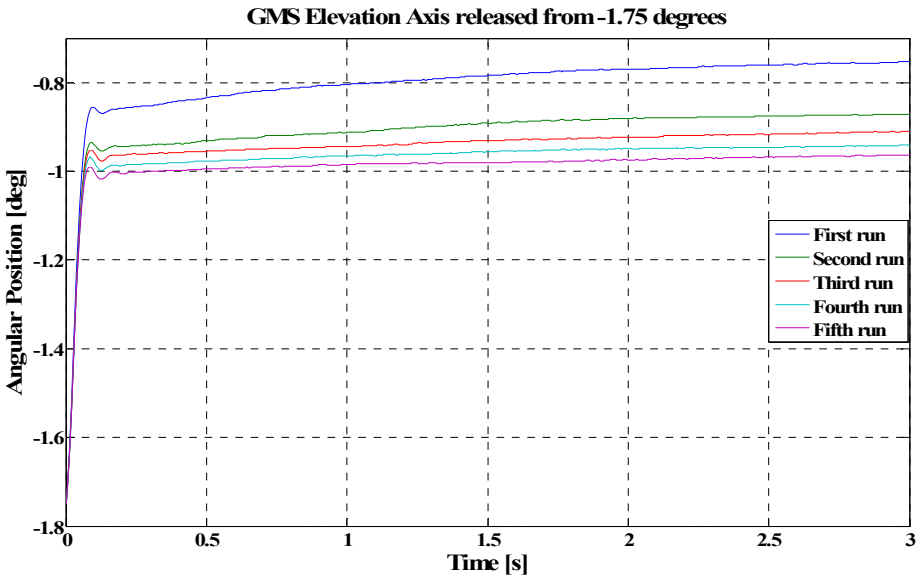

Figure 3.6. GMS elevation axis released from -1.75 degrees

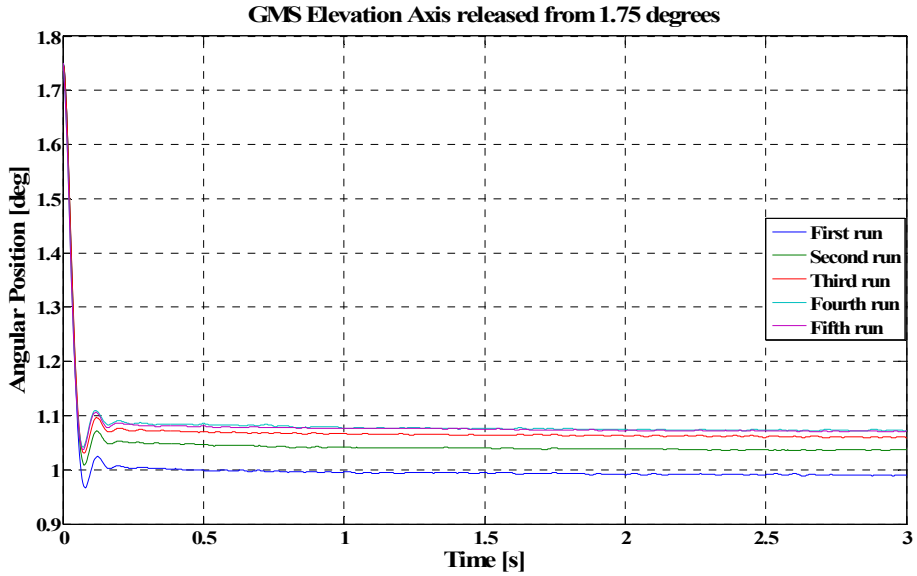

Figure 3.7. GMS elevation axis released from 1.75 degrees

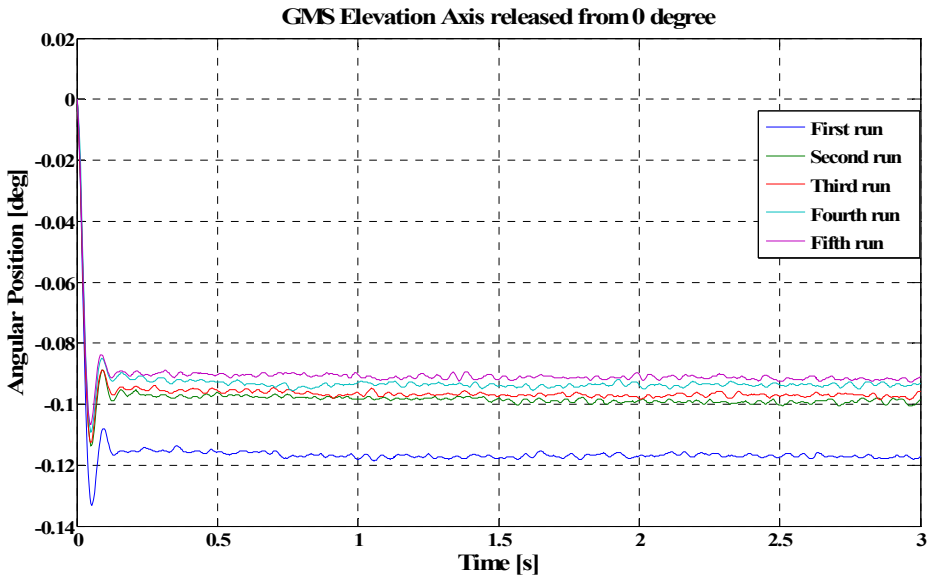

Figure 3.8. GMS elevation axis released from 0 degree

One of these cables carries the encoder data from the encoders to the encoder connector, which is fixed to the structure of GMS (Figure 3.9). This cable is connected to only azimuth axis therefore, applying torque only on the azimuth axis. On the other hand the other cable carries the power signals from the motor connector to the motor windings (Figure 3.10). The motor connector is also fixed to the structure of GMS. This cable is connected to both of the axes. But the stator of the three phase brushless motor used for the azimuth axis is not moving. So this cable interacts with the elevation axis. But since the elevation axis is moving together with the azimuth axis, the cable is also interacting with the azimuth axis in an indirect manner.

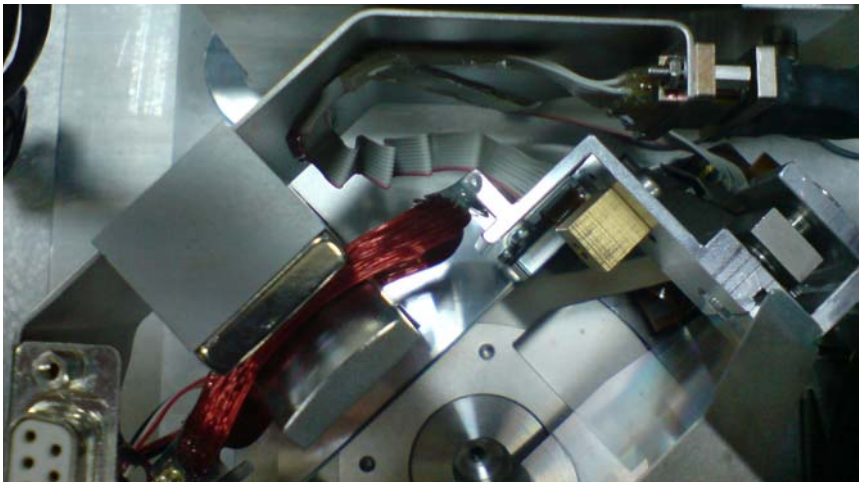

Figure 3.9. Encoder cable

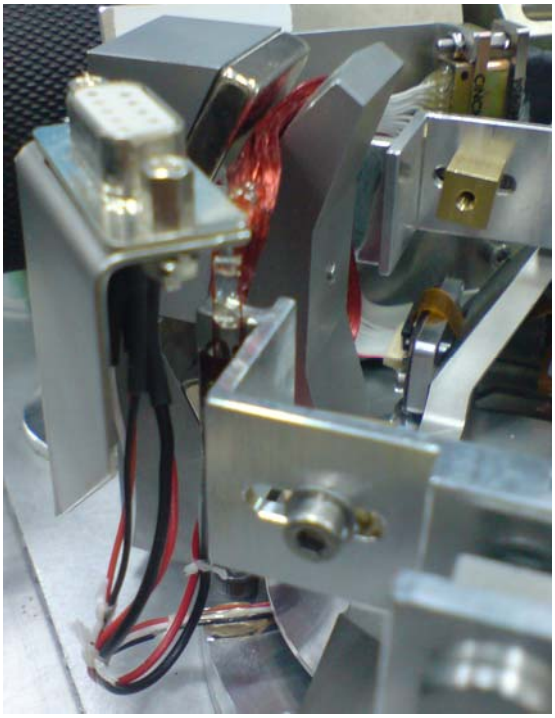

Figure 3.10. Cable from the motor connector to elevation axis gimbal

The cables do not act like ideal torsional springs exactly. As the axes move they apply force on the cables as well as the cables apply force on the axes. As a result the forces applied to the cables cause them to deform elastically and plastically. This effect is more obvious for the encoder cable attached to the azimuth axis gimbal. The reasons behind this behavior are its type and shape as well as its connection conditions at both ends.

Each time a force is applied to the cables; they deform. The degree of this deformation depends on the magnitude of the force and the time period it is applied. The longer the axes stay at an angular position, the more the cables are deformed plastically to stay at that angular position. This is why, at each run of the tests, the axes go nearer to the point that they are released than the runs before.

Although the cables are non-ideal elements which also dissipate energy as well as storing it, they are modeled as ideal springs since developing a model which reflects the true dynamics of the cable would be cumbersome. Furthermore, since the non ideal behaviors of the cables are not expected to affect the whole system dynamics significantly, it would not worth the effort. Hence, the cables are modeled as ideal springs with preload. The relationship between the torques applied on axes by the cables and the angular positions of the axis gimbals is defined with the following equation:

$$T_{cable} = k\theta + T_{preload} \tag{3.3}$$

where

T_{cable}	Torque acting on the axis because of the presence of the cable [N.m]
k	Spring constant [N.m/rad]
θ	Angular position [rad]

$T_{preload}$ Torque acting on the axis because of the potential energy stored in the cables at the axis zero position [N.m]

Figure 3.3 through Figure 3.8 show free vibration traces of the GMS axes. By the help of these traces, the viscous damping coefficients for the axes could be obtained by using logarithmic decrement concept. Furthermore, by knowing the inertia values for the axes and obtaining the viscous friction coefficients, the spring constants could also be obtained. The equations needed for this analysis are given below:

$$\delta \cong \ln \frac{x_1}{x_2} \tag{3.4}$$

$$\zeta = \frac{\delta}{\sqrt{(2\pi)^2 + \delta^2}} \tag{3.5}$$

$$\omega_n = \frac{\omega_d}{\sqrt{1-\zeta^2}} \tag{3.6}$$

$$k = \omega_n{}^2 J \tag{3.7}$$

$$b = 2\zeta\sqrt{kJ} \tag{3.8}$$

where

x_1 Difference between the first peak and the steady state value [deg]

x_2 Difference between the second peak and the steady state value [deg]

δ Second logarithmic decrement

ζ Damping ratio

ω_n Natural frequency [rad/s]

ω_d Damped natural frequency [rad/s]

J Inertia of the axes [kg.m^2]

b Viscous damping coefficient [N.m.s/rad]

k Spring constant [N.m/rad]

First, x_1, x_2 and ω_d values are extracted from the free vibration plots. Then, damping coefficients are calculated by using (3.4) and (3.5). Using the damping coefficient values obtained natural frequency values are calculated by using (3.6). Then, spring constants and viscous damping coefficients are obtained by using (3.7) and (3.8). The spring constant and viscous damping coefficients obtained are given in Table 3.1 through Table 3.6.

Table 3.1. Viscous damping coefficient and spring constant values obtained by releasing GMS azimuth axis from -1.75 ^{o}

	Viscous Damping Coefficient [N.m.s/rad]	**Spring Constant [N.m/rad]**
Test Run 1	0.0200	1.37
Test Run 2	0.0200	1.37
Test Run 3	0.0198	1.27
Test Run 4	0.0198	1.32
Test Run 5	0.0206	1.38

Table 3.2. Viscous damping coefficient and spring constant values obtained by releasing GMS azimuth axis from 1.75^{o}

	Viscous Damping Coefficient [N.m.s/rad]	Spring Constant [N.m/rad]
Test Run 1	0.00620	4.21
Test Run 2	0.0434	6.89
Test Run 3	0.0224	4.09
Test Run 4	0.0296	4.26
Test Run 5	0.0509	5.37

Table 3.3. Viscous damping coefficient and spring constant values obtained by releasing GMS azimuth axis from 0^{o}

	Viscous Damping Coefficient [N.m.s/rad]	Spring Constant [N.m/rad]
Test Run 1	0.0190	2.03
Test Run 2	0.0236	1.92
Test Run 3	0.0166	1.80
Test Run 4	0.0202	1.86

Table 3.4. Viscous damping coefficient and spring constant values obtained by releasing GMS elevation axis from -1.75^{o}

	Viscous Damping Coefficient [N.m.s/rad]	Spring Constant [N.m/rad]
Test Run 1	0.00250	0.438
Test Run 2	0.00250	0.438
Test Run 3	0.00450	0.517
Test Run 4	0.00460	0.487
Test Run 5	0.00210	0.432

Table 3.5. Viscous damping coefficient and spring constant values obtained by releasing GMS elevation axis from 1.75^{o}

	Viscous Damping Coefficient [N.m.s/rad]	Spring Constant [N.m/rad]
Test Run 1	0.00370	0.462
Test Run 2	0.00360	0.432
Test Run 3	0.00350	0.405
Test Run 4	0.00330	0.400
Test Run 5	0.00340	0.403

Table 3.6. Viscous damping coefficient and spring constant values obtained by releasing GMS elevation axis from 0^o

	Viscous Damping Coefficient [N.m.s/rad]	Spring Constant [N.m/rad]
Test Run 1	0.00460	0.555
Test Run 2	0.00400	0.539
Test Run 3	0.00450	0.553
Test Run 4	0.00470	0.557
Test Run 5	0.00510	0.656

The steady-state values used in the calculations are not reliable since the continuous deformation of the cables causes the steady-state position to change. Hence, the values determined for spring constant and viscous damping coefficient values are certainly not precise but still the orders of the values are consistent. Therefore this information can be used for parameter estimation to define the limits and initial guesses for the spring constant and viscous damping coefficients.

3.2 Motion control card model

The MATLAB/Simulink model constructed for the motion control card consists of trajectory generator model, PID control block model, DAC converter model, and the limiters for the integral term and output of the PID controller (Figure 3.11). Each submodel will be explained in detail in the following sections.

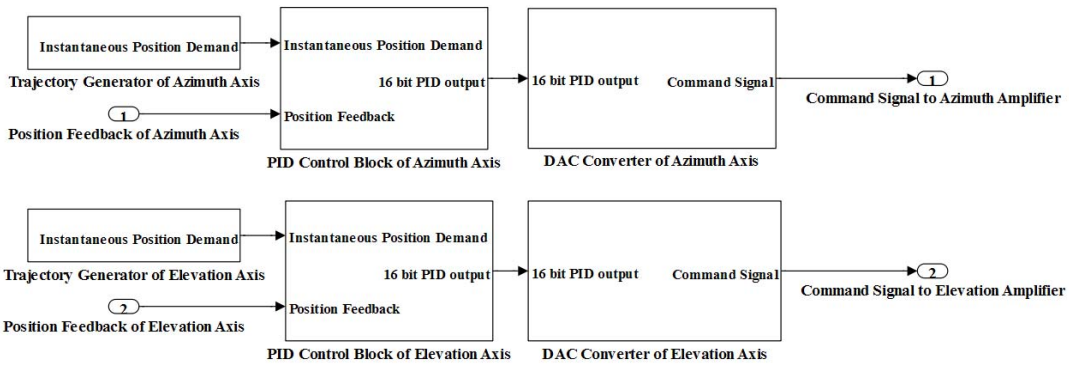

Figure 3.11. Motion control card model developed in MATLAB/Simulink environment

3.2.1 Trajectory generator model

Trajectory generator model developed in MATLAB/Simulink environment is shown in Figure 3.12. As mentioned in Chapter 2, the trajectory generator of the motion control card takes in the move constraints set for the axis (maximum velocity and acceleration/deceleration), initial position and the position command from the supervisory control, and generates a trapezoidal velocity profile, in real-time.

Trapezoidal profile means that the axis accelerates at the value set until the velocity of the axis reaches the maximum velocity defined. Then the axis cruises at maximum velocity and decelerates to a stop at the desired position (Figure 3.13). Sometimes, when the desired movement is small or the acceleration/deceleration value defined is high, the axis could not reach the maximum velocity defined. For this case, the velocity profile is as shown in Figure 3.14.

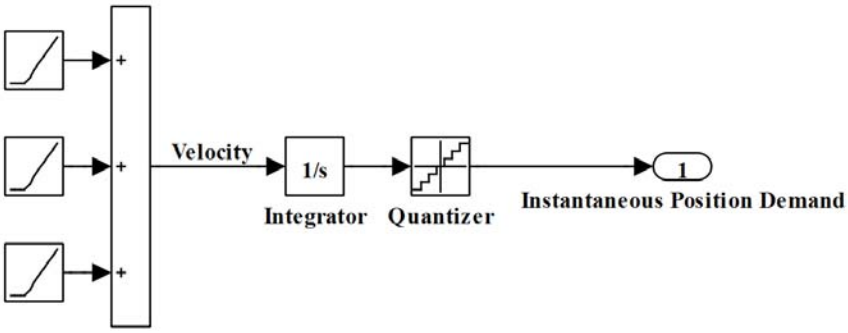

Figure 3.12. Trajectory generator model developed in MATLAB/Simulink environment

Since the angular position ranges of the GMS axes are only a few degrees, it is observed that the maximum velocities that are set for the axes are not reached during the movements of the axes. Therefore the model of the trajectory generator is constructed based on the velocity profile shown in Figure 3.14.

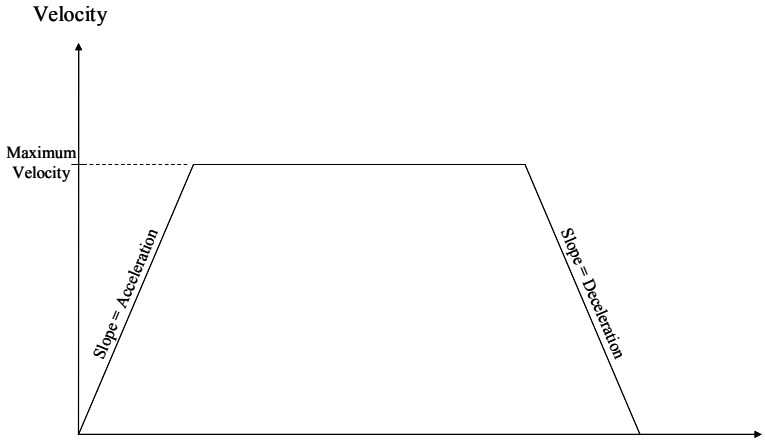

Figure 3.13. Trapezoidal velocity profile (maximum velocity reached)

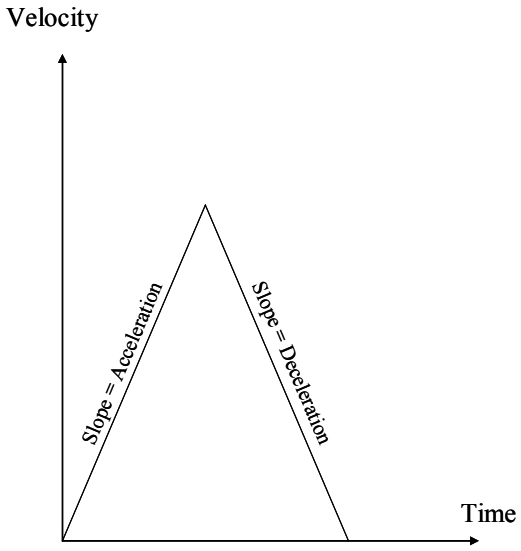

Figure 3.14. Trapezoidal velocity profile (maximum velocity is not reached)

3.2.2 PID control block model

The digital PID control block of the motion control card takes in instantaneous position commands from the trajectory generator and position feedback from the encoder each PID controller update period, and outputs a 16-bit value which is converted to the corresponding voltage value by the DAC (Digital to Analog Converter) of the motion control card. PID control block model developed in MATLAB/Simulink environment is shown in Figure 3.15.

By using PID control, three control laws, which are; P (proportional control), I (integral control), and D (derivative control) are utilized. The discrete-time domain transfer functions for integral (I) and derivative (D) controls depends on the numerical techniques used for integration and differentiation. Backward difference,

forward difference, and Tustin (bilinear) transformation are the commonly used methods.

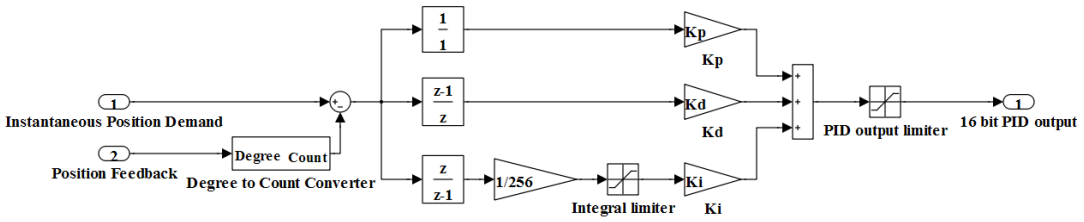

Figure 3.15. PID control block model developed in MATLAB/Simulink environment

As mentioned in Chapter 2, the discrete time transfer function of the digital PID controller of the motion control card is unknown; but, it is known that a widely used technique for the utilization of integral and derivative control laws in discrete-time is the backward difference technique. For a motion control card which uses backward difference technique for differentiation and integration, the discrete-time domain transfer function of the PID controller is written as;

$$G_c(z) = K_p + K_i \frac{Tz}{z-1} + K_d \frac{z-1}{Tz} \tag{3.9}$$

where

T Controller update period [s]

K_p Proportional gain [LSB/encoder count]

K_i Integral gain [LSB/encoder count.s]

K_d Derivative gain [LSB.s/encoder count]

However in the manual of the motion control card it is written that the integral of error is scaled by dividing by 256 prior to being multiplied by K_i [16]. Using this information, the discrete-time domain transfer function of the PID controller of the motion control card is given by:

$$G_c(z) = K_p + \frac{K_i}{256}\left(\frac{Tz}{z-1}\right) + K_d\left(\frac{z-1}{Tz}\right) \tag{3.10}$$

To be certain about the transfer function of the PID controller, some tests are conducted. By leaving the motors unpowered, and setting the position feedback to zero position, commands are sent to the motion control card and the DAC output of the motion controller is collected for different PID parameter sets by using a simple LabView program. For each PID parameter set a position command of 100 encoder counts is sent to the motion control card, while the axis velocity and acceleration values are set to 60,000 counts/s and 60,000 counts/s^2 respectively.

When the motion control card and the model outputs are plotted on the same graph, it was seen that there were significant differences between the outputs. Further analysis revealed that the motion control card accepts the K_i value entered as K_iT and K_d value entered as $\frac{K_d}{T}$.

After the controller transfer function had been revised considering this knowledge and the tests were repeated, it was seen that the control card and the model outputs was more or less the same. The motion control card outputs and the model outputs plotted on the same graphs are shown in Figure 3.16, Figure 3.17 and Figure 3.18.

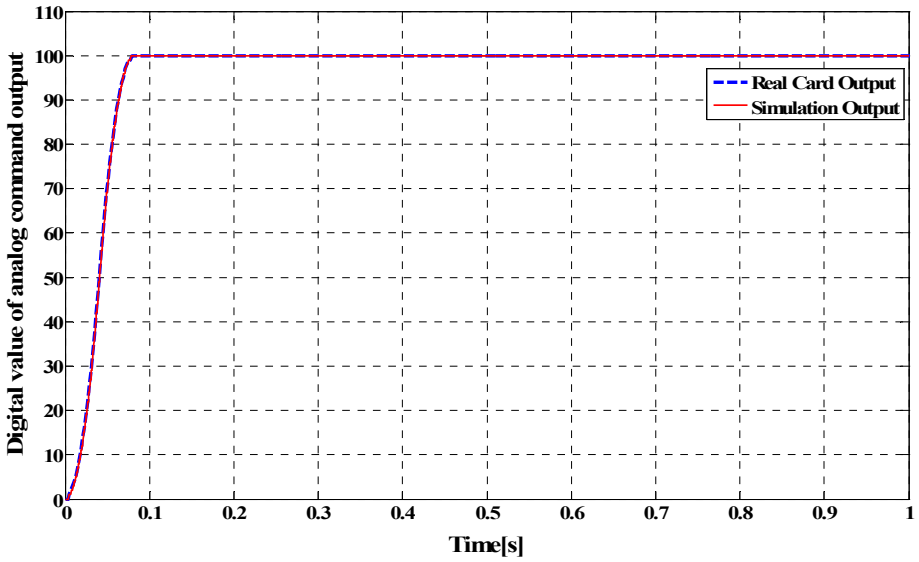

Figure 3.16. Motion control card output for K_p=1, K_i=0, K_d=0

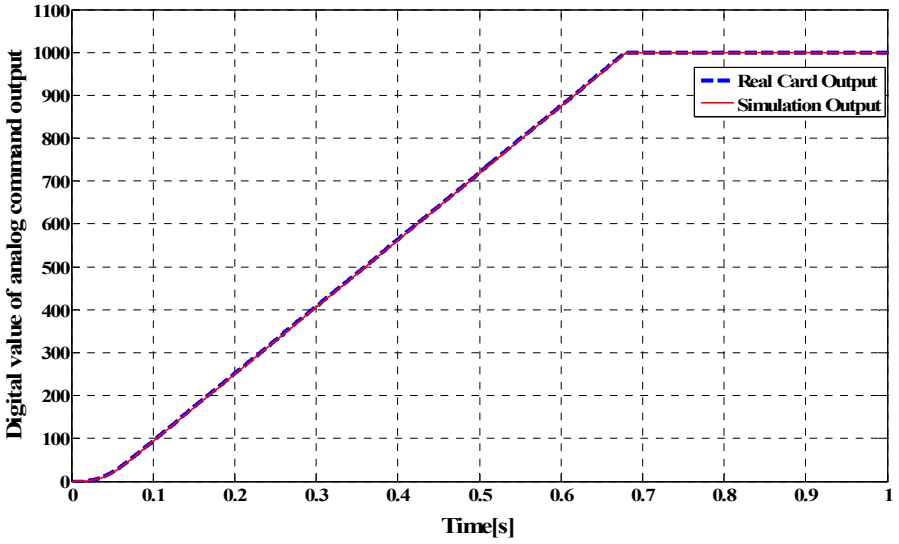

Figure 3.17. Motion control card output for K_p=0, K_i=1, K_d=0 and integral limit of 1000

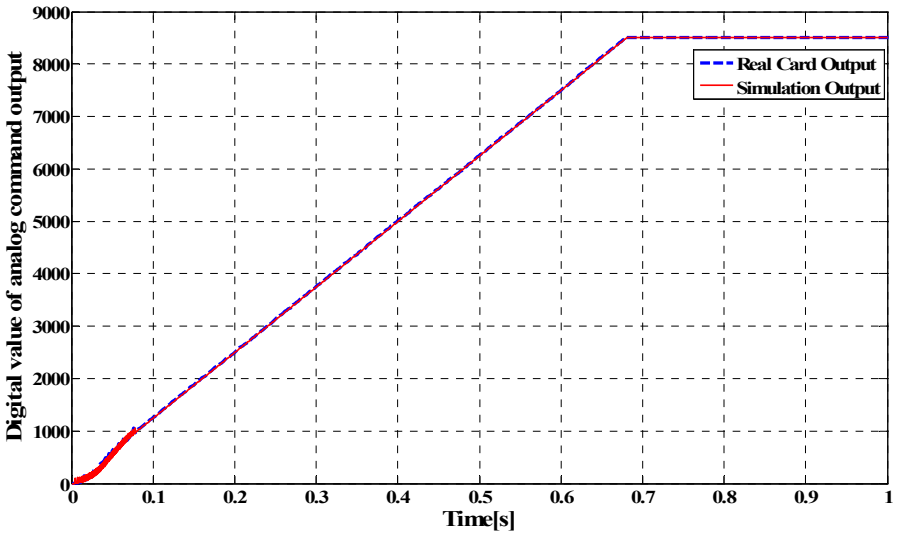

Figure 3.18. Motion control card output for K_p=5, K_i=8, K_d=100 and integral limit of 1000

The maximum errors for the first two plots are 3, which corresponds to 0.915 mV. For the third case, where derivative control is also added, a detailed view is given in Figure 3.19. Because of the derivative control the response dynamics is fast but the data sampling rate is not as fast as the response dynamics. This is because the execution of the LabView function *'Read DAC'*, which is used to collect the digital values of the analog command output of the motion control card, takes 2-3 ms. Hence, the fastest data sampling rate that can be achieved by using this function is less than 500 Hz. A data acquisition card (DAQ) card could be used to sample data at appropriate rates; however, a suitable DAQ card is not available in this study. The only way to collect the needed data was using the *'Read DAC'* function of LabView.

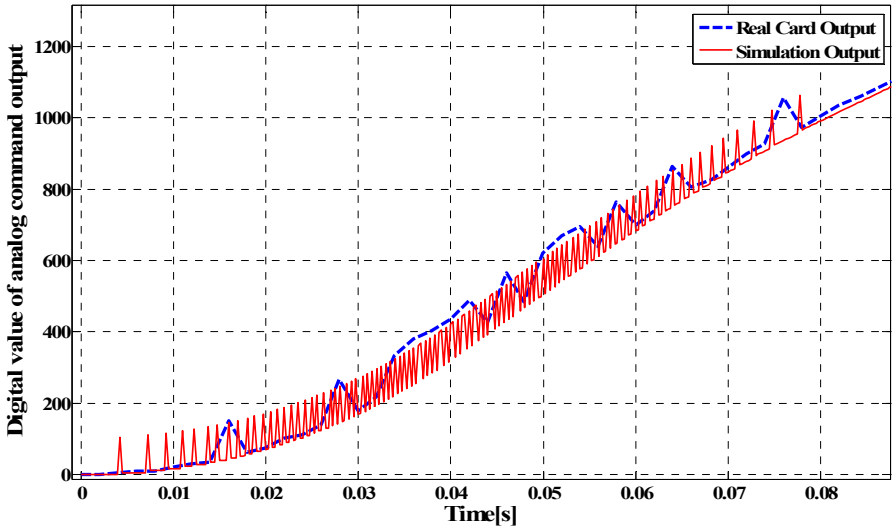

Figure 3.19. Detailed view of Figure 3.18

From Figure 3.19 it can be seen that at points where real system data is collected, the simulation and real system outputs are in agreement. To be sure about the transfer function, PID block model, which is developed by assuming bilinear transformation method is used by the control card, was also constructed in MATLAB/Simulink environment and simulation was run with this model. From this simulation results, it was observed that the simulation outputs are clearly different from these of the real system (Figure 3.20). A model for forward difference method could not be developed since the degree of the numerator is greater than the denominator for the derivative term when the mapping is performed by forward difference method. As a result, it is concluded that the backward difference technique is used by the control card for differentiation and integration and the discrete-time domain transfer function for the PID controller is;

$$G_c(z) = K_p + \frac{K_i}{256}\left(\frac{z}{z-1}\right) + K_d\left(\frac{z-1}{z}\right) \tag{3.11}$$

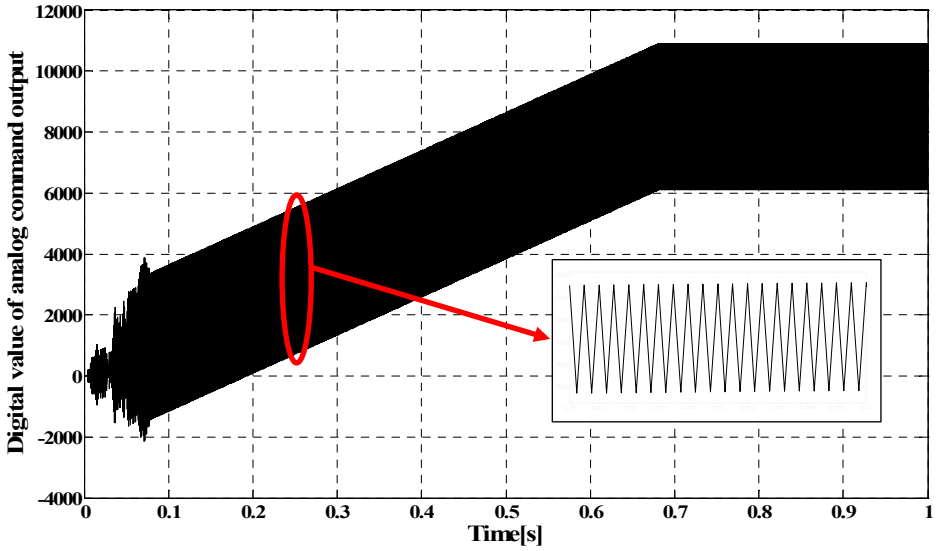

Figure 3.20. Output of simulation with Tustin method for K_p=5, K_i=8, K_d=100 and integral limit of 1000

There are two limiters in the PID control block of the motion control card. One of them limits the output of the PID control loop within $\pm 2^{15}$ bit interval while the other limits the integral term of the PID output to a value set by the "Integral limit" parameter of the motion control card. During this study, the integral limit is set to $\pm 2^{15}$ (its maximum value).

3.2.3 DAC model

DAC model developed in MATLAB/Simulink environment is shown in Figure 3.21. 16 bit DAC converts the digital PID loop output, which is in $\pm 2^{15}$ range to an analog voltage value in ± 10 Volts range. DAC of the motion control card is modeled with a ZOH (Zero Order Hold) block and a Digital to Analog Converter gain (K_{dac}) whose value is $20V/2^{16}$.

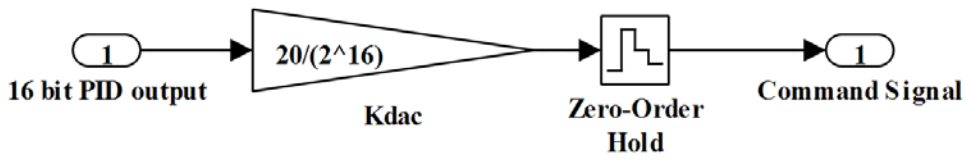

Figure 3.21. DAC model developed in MATLAB/Simulink environment

3.3 Amplifier model

Amplifier model developed in MATLAB/Simulink environment is shown in Figure 3.22. Amplifiers are supplied with ± 10 Volts command signals from the motion control card and try to output a proportional current value. As mentioned in Chapter 2, the proportionality constant which relates the input command voltage to the output current value is called the amplifier gain and it is 1.6 Amperes/Volt for the amplifiers used.

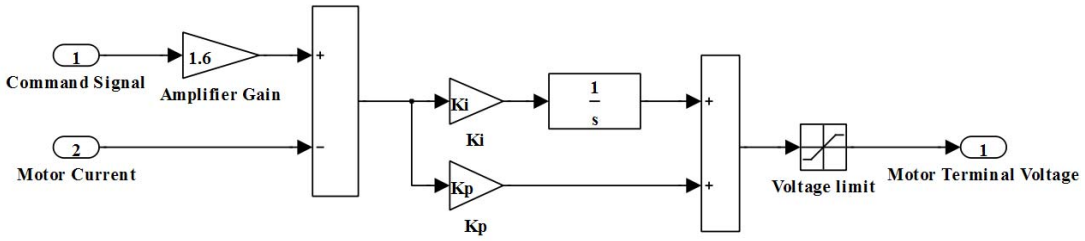

Figure 3.22. Amplifier model developed in MATLAB/Simulink environment

However, the currents in the motor windings can not be controlled directly. To be able to generate a current value to pass through the motor windings proportional to the command signal, analog PI current regulators are used in the amplifiers. The difference between the current demand and the actual current sensed is fed to the PI current regulator.

As mentioned in Chapter 2, the amplifiers have DIP switches to increase/decrease the proportional gain and activate/deactivate the integral gain of the PI current regulators. For the control system configuration, using these switches, the proportional gain is increased and the integral gain is activated for the amplifier used for the GMS azimuth axis while the proportional gain is decreased and the integral gain is deactivated for the amplifier used for the GMS elevation axis. This configuration of the DIP switches is not decided by considering a specific purpose. At the beginning of the study, the DIP switches are set in this configuration and stayed as they are throughout the study.

The outputs of the PI current regulators are fed to the PWM blocks of the amplifiers. Since the switching frequency of PWMs is 33 kHz, the effect of their dynamics to

the whole system dynamics would probably be insignificant so PWM model is not included in the developed model.

PWM blocks modulate the supply voltage of the amplifiers which is 24 Volts. So for 100% duty cycle value the PWM blocks output 24 Volts and the output voltage value can not exceed this value. This phenomenon is called as amplifier saturation and included in the model using a saturation block.

3.4 Disturbance torque model

Since the IRSGS is mounted on the outer two axes of the FMS, it also makes the same motions as the FMS, so does the GMS. The angular motions of the FMS axes impose torques on the GMS axes obeying the angular momentum equation. Two mathematical models for disturbance torques are obtained, one by using Euler equations and another in MATLAB/Simulink/SimMechanics environment. These two models are explained and also their results are compared in the following sections.

3.4.1 Disturbance torque model by using Euler equation for rigid body motion

The torques, which are imposed on the GMS axes, originating from the motion of the FMS axes, are obtained using Euler equation in matrix form. The Euler equation is used in body fixed reference frame rather than fixed reference frame to simplify the calculations since the moment of inertia tensor is constant in body fixed reference frame. The matrix form of the Euler equation written in the body fixed reference frame is

$$\sum \overline{M}_c^{(b)} = \hat{J}_c^{(b)} \overline{\alpha}_b^{(b)} + \tilde{\omega}_b^{(b)} \hat{J}_c^{(b)} \overline{\omega}_b^{(b)} \tag{3.12}$$

where

$\hat{J}_c^{(b)}$	Inertia tensor w.r.t. center of mass in body fixed reference frame [kg.m^2]
$\overline{\alpha}_b^{(b)}$	Angular acceleration about the body fixed reference frame, written in the body fixed frame [rad/s^2]
$\overline{\omega}_b^{(b)}$	Angular velocity about the body fixed reference frame and written in the body fixed frame [rad/s]
$\widetilde{\omega}_b^{(b)}$	Skew symmetric angular velocity matrix [rad/s]

To obtain the angular velocity and acceleration terms in the above equation, a kinematics analysis of the mechanism, which consists of the two outer axes of the FMS and the two axes of the GMS, should be performed. Denavit and Hartenberg (D-H) convention is used for the kinematic analysis of the FMS-GMS system.

Any robotic system can be described kinematically by giving the values of four quantities for each link, two describe the link itself, and two describe the link's connection to a neighboring link [17]. These four quantities are; s_k, linear displacement of link (k) w.r.t. link (k-1), a_k, effective length of link (k), θ_k, angular displacement of link (k) w.r.t. link (k-1) and α_k, twist angle of link (k). These four parameters are called D-H parameters. In the case of a revolute joint, θ_k is called the joint variable, and the other three quantities would be fixed link parameters. The definition of mechanisms by means of these quantities is a convention usually called the D-H notation [17].

The mechanism consists of two axes of FMS and GMS axes, has a total of five links where link 0 is the fixed link, link 1 is the outer FMS axis gimbal, link 2 is the inner

FMS axis gimbal, link 3 is the GMS azimuth axis gimbal and link 4 is the GMS elevation axis gimbal. Each link is connected to the prior link with a revolute joint.

First step for the kinematics analysis is assigning the coordinate frames to the links and obtaining D-H parameters. Let $\vec{u}_1^{(k)}$, $\vec{u}_2^{(k)}$ and $\vec{u}_3^{(k)}$ be the unit vectors along the x, y and z axes of the coordinate frame fixed to link k. Then $\vec{u}_1^{(k)}$, $\vec{u}_2^{(k)}$ and $\vec{u}_3^{(k)}$ could be assigned as described below.

The unit vector $\vec{u}_3^{(k)}$ is assigned to be the axis of actuation for joint (k+1). Its sense is chosen arbitrarily. $\vec{u}_1^{(k)}$ is assigned to be the unit vector along the common normal between joint (k) and joint (k+1). It is oriented from joint (k) to joint (k+1) if their axes are not concurrent. If these axes are concurrent, then the sense of $\vec{u}_1^{(k)}$ is chosen arbitrarily. Finally, $\vec{u}_2^{(k)}$ is selected as to complete a right handed frame [18]. The coordinate frames assigned for the FMS-GMS mechanism using D-H convention are shown in Figure 3.23 and Figure 3.24.

Once the coordinate frames are assigned, D-H parameters are obtained using the relations below.

$$s_k = \left| O_{k-1} A_k \right| \tag{3.13.a}$$

$$a_k = \left| A_k O_k \right| \tag{3.13.b}$$

$$\theta_k = \angle \left[\vec{u}_1^{(k-1)} \rightarrow \vec{u}_1^{(k)} @ \vec{u}_3^{(k-1)} \right] \tag{3.13.c}$$

$$\alpha_k = \angle \left[\vec{u}_3^{(k-1)} \rightarrow \vec{u}_3^{(k)} @ \vec{u}_1^{(k)} \right] \tag{3.13.d}$$

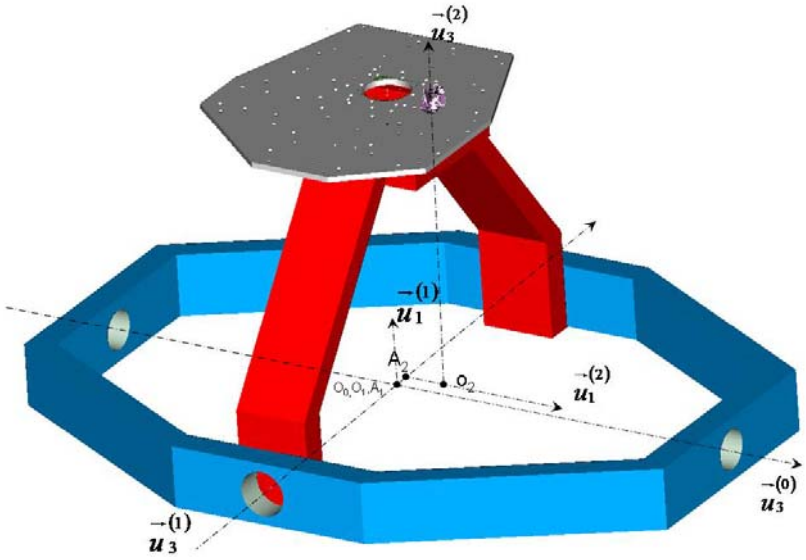

Figure 3.23.Coordinate frames assigned for the FMS-GMS mechanism

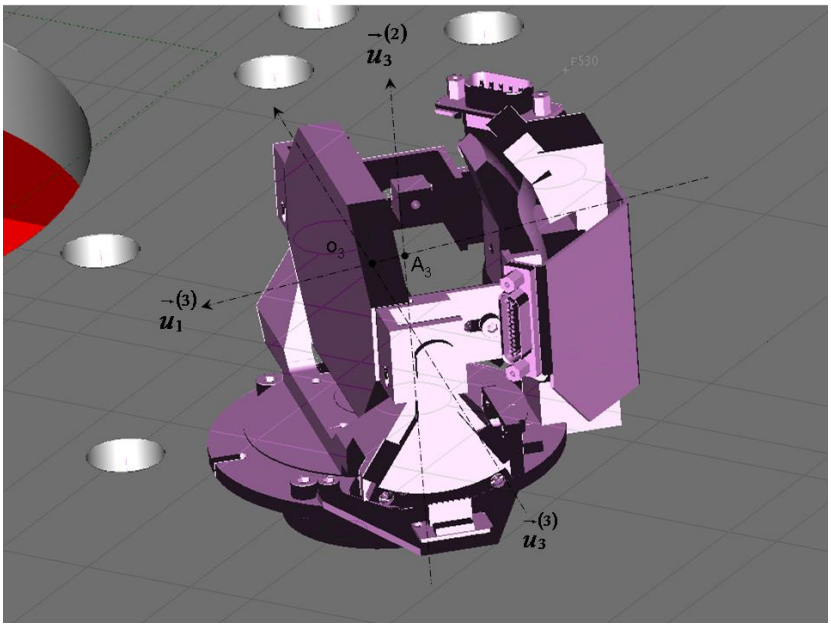

Figure 3.24.Coordinate frames assigned for the FMS-GMS mechanism

where

O_k Intersection point of the axes along $\vec{u}_1^{(k)}$ and $\vec{u}_3^{(k)}$

A_k Intersection point of the axes along $\vec{u}_1^{(k)}$ and $\vec{u}_3^{(k-1)}$ [18].

To analyze the kinematics of the system in rotational frames, the D-H parameters, s_k and a_k are not needed. $\theta_1, \theta_2, \theta_3$, and θ_4 are the joint variables for joint 1, joint 2, joint 3, and joint 4 respectively. Twist angles of links obtained by using (3.13.d) are

$$\alpha_1 = \angle\left[\vec{u}_3^{(0)} \rightarrow \vec{u}_3^{(1)} @ \vec{u}_1^{(1)}\right] = 90^o \tag{3.14.a}$$

$$\alpha_2 = \angle\left[\vec{u}_3^{(1)} \rightarrow \vec{u}_3^{(2)} @ \vec{u}_1^{(2)}\right] = 90^o \tag{3.14.b}$$

$$\alpha_3 = \angle\left[\vec{u}_3^{(2)} \rightarrow \vec{u}_3^{(3)} @ \vec{u}_1^{(3)}\right] = 90^o \tag{3.14.c}$$

Based on D-H convention, the rotation matrix between link k and link (k-1) could be written as; [18]

$$\hat{C}^{(k-1,k)} = e^{\tilde{u}_3\theta_k} e^{\tilde{u}_1\alpha_k} \tag{3.15}$$

where

$$e^{\tilde{u}_3\theta_k} = \hat{I}\cos\theta_k + \tilde{u}_3\sin\theta_k + \bar{u}_3\bar{u}_3^{\,t}\left(1-\cos\theta_k\right) \tag{3.16}$$

$$e^{\tilde{u}_1\alpha_k} = \hat{I}\cos\alpha_k + \tilde{u}_1\sin\alpha_k + \bar{u}_1\bar{u}_1^{\,t}\left(1-\cos\alpha_k\right) \tag{3.17}$$

$$\bar{u}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix} \qquad \Rightarrow \qquad \tilde{u}_1 = \begin{bmatrix}0 & 0 & 0\\0 & 0 & -1\\0 & 1 & 0\end{bmatrix} \tag{3.18}$$

$$\bar{u}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \Rightarrow \qquad \tilde{u}_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.19}$$

By substituting (3.19) into (3.16);

$$e^{\tilde{u}_3 \theta_k} = \begin{bmatrix} \cos\ \theta_k & -\sin\ \theta_k & 0 \\ \sin\ \theta_k & \cos\ \theta_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.20}$$

And by substituting (3.18) into (3.17);

$$e^{\tilde{u}_1 \alpha_k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_k & -\sin\alpha_k \\ 0 & \sin\alpha_k & \cos\alpha_k \end{bmatrix} \tag{3.21}$$

Finally, by substituting (3.20) and (3.21) into (3.15), the rotation matrix between link k and link (k-1) could be written as;

$$\hat{C}^{(k-1,k)} = \begin{bmatrix} \cos\theta_k & -\sin\theta_k \cos\alpha_k & \sin\theta_k \sin\alpha_k \\ \sin\theta_k & \cos\theta_k \cos\alpha_k & -\cos\theta_k \sin\alpha_k \\ 0 & \sin\alpha_k & \cos\alpha_k \end{bmatrix} \tag{3.22}$$

Angular velocities and angular accelerations of links w.r.t. inertial frame are given with (3.23) and (3.24) [17], while angular velocities and angular accelerations of links w.r.t. their body frames are given with (3.25) and (3.26) [17].

$$\bar{\omega}_k^{(0)} = \bar{\omega}_{k-1}^{(0)} + \dot{\theta}_k \hat{\Phi}_{k-1} \bar{u}_3 \tag{3.23}$$

$$\bar{\alpha}_k^{(0)} = \bar{\alpha}_{k-1}^{(0)} + \left[\ddot{\theta}_k \hat{I} + \dot{\theta}_k \tilde{\omega}_{k-1}^{(0)} \right] \hat{\Phi}_{k-1} \bar{u}_3 \tag{3.24}$$

$$\overline{\omega}_k^{(k)} = \hat{C}^{(0,k)T}\overline{\omega}_k^{(0)} \qquad \Rightarrow \qquad \overline{\omega}_k^{(k)} = \Phi_k{}^T\overline{\omega}_k^{(0)} \tag{3.25}$$

$$\overline{\alpha}_k^{(k)} = \hat{C}^{(0,k)T}\overline{\alpha}_k^{(0)} \qquad \Rightarrow \qquad \overline{\alpha}_k^{(k)} = \Phi_k{}^T\overline{\alpha}_k^{(0)} \tag{3.26}$$

where orientation matrix of link k, $\hat{\Phi}_k$, could be expressed as [18];

$$\hat{\Phi}_k = \hat{C}^{(0,k)} \tag{3.27}$$

Since non-diagonal terms of the inertia matrices for GMS azimuth and elevation axes are negligible when compared to the diagonal terms, they are taken as zero. Thus, the inertia matrices take the form:

$$\hat{J}_3^{(3)} = \hat{J}_4^{(4)} = \begin{bmatrix} J_{11} & 0 & 0 \\ 0 & J_{22} & 0 \\ 0 & 0 & J_{33} \end{bmatrix} \tag{3.28}$$

GMS azimuth axis:

$$\sum \overline{M}_{3c}^{(3)} = \hat{J}_3^{(3)}\overline{\alpha}_3^{(3)} + \tilde{\omega}_3^{(3)}\hat{J}_3^{(3)}\overline{\omega}_3^{(3)} \tag{3.29}$$

The second row of the above matrix operation gives the expression of the disturbance torque acting on GMS azimuth axis, since the GMS azimuth axis rotates around $\vec{u}_2^{(3)}$. Using (3.29), the disturbance torque for GMS azimuth axis is obtained:

$$T_{d_{azm}} = (J_{33} - J_{11})\left[\left(\dot{\theta}_2{}^2 - \dot{\theta}_1{}^2(\sin\theta_2)^2\right)\cos\theta_3\sin\theta_3 + \left((\cos\theta_3)^2 - (\sin\theta_3)^2\right)\dot{\theta}_1\dot{\theta}_2\sin\theta_2\right]$$
$$+ J_{22}\left(\ddot{\theta}_3 - \ddot{\theta}_1\cos\theta_2 + \dot{\theta}_2\dot{\theta}_1\sin\theta_2\right) \tag{3.30}$$

Disturbance torque model for GMS azimuth axis developed in MATLAB/Simulink environment using (3.30) is shown in Figure 3.25.

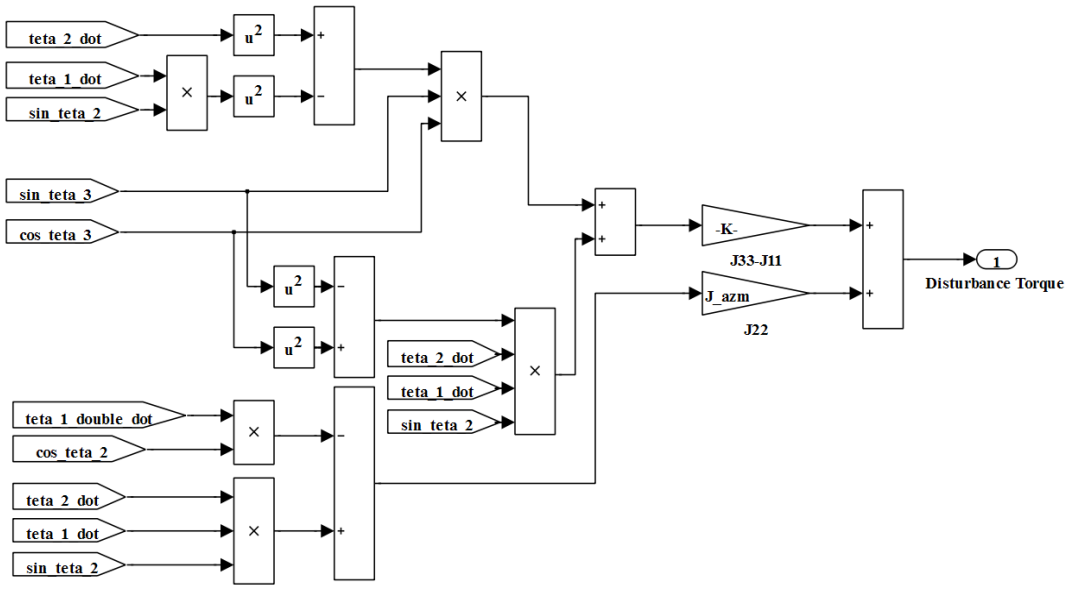

Figure 3.25. Disturbance torque model for GMS azimuth axis developed in MATLAB/Simulink environment

GMS elevation axis:

$$\sum \overline{M}_{4c}^{(4)} = \hat{J}_4^{(4)} \overline{\alpha}_4^{(4)} + \tilde{\omega}_4^{(4)} \hat{J}_4^{(4)} \overline{\omega}_4^{(4)} \tag{3.31}$$

The third row of the above matrix operation gives the expression of the disturbance torque acting on GMS elevation axis, since the GMS elevation axis rotates around $\vec{u}_3^{(4)}$. Using (3.31), the disturbance torque for GMS elevation axis is obtained:

$$T_{d_{elv}} = J_{33}\left(\dot{\theta}_3\dot{\theta}_2 \sin\theta_3 + \dot{\theta}_1\dot{\theta}_2 \cos\theta_2 \sin\theta_3 + \dot{\theta}_1\dot{\theta}_3 \sin\theta_2 \cos\theta_3 - \ddot{\theta}_2 \cos\theta_3 + \ddot{\theta}_4 + \ddot{\theta}_1 \sin\theta_2 \sin\theta_3\right)$$
$$+\left(J_{11} - J_{22}\right)\left(-\dot{\theta}_3 \cos\theta_4 + \dot{\theta}_1 \cos\theta_2 \cos\theta_4 + \dot{\theta}_2 \sin\theta_3 \sin\theta_4 + \dot{\theta}_1 \cos\theta_3 \sin\theta_2 \sin\theta_4\right)$$
$$\left(\dot{\theta}_3 \sin\theta_4 - \dot{\theta}_1 \cos\theta_2 \sin\theta_4 + \dot{\theta}_2 \sin\theta_3 \cos\theta_4 + \dot{\theta}_1 \cos\theta_3 \sin\theta_2 \cos\theta_4\right) \quad (3.32)$$

Disturbance torque model for GMS elevation axis developed in MATLAB/Simulink environment using (3.32) is shown in Figure 3.26.

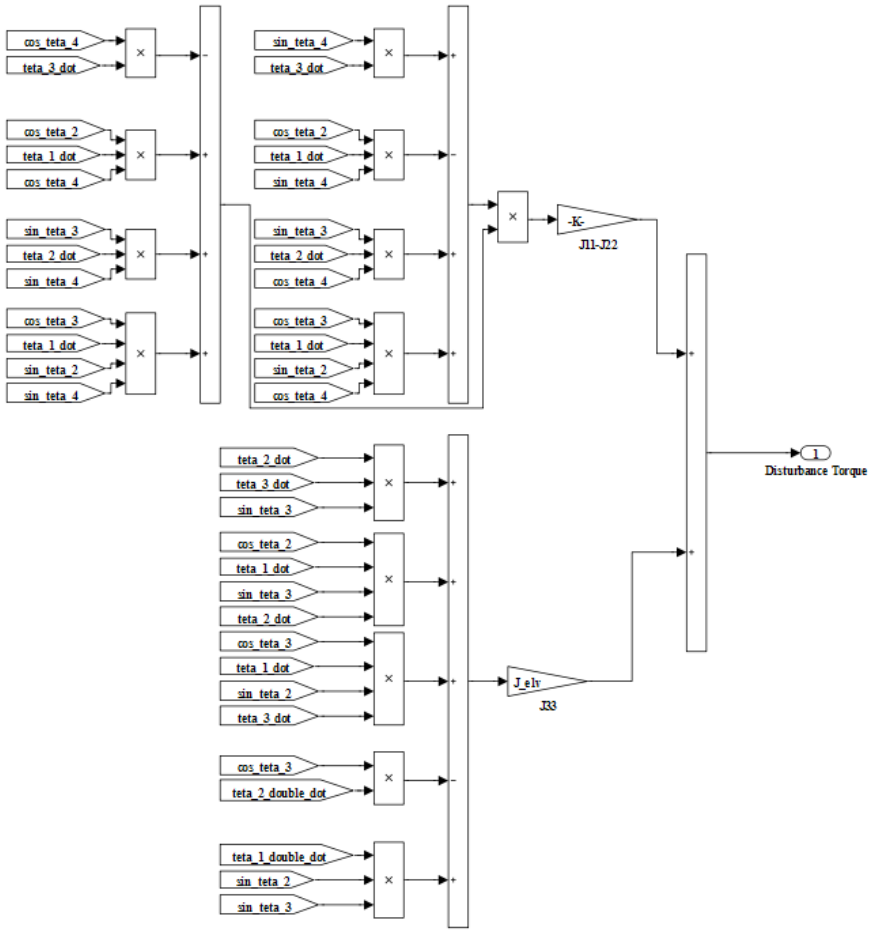

Figure 3.26. Disturbance torque model for GMS elevation axis developed in MATLAB/Simulink environment

3.4.2 Disturbance torque model developed in MATLAB/Simulink/ SimMechanics environment

Disturbance torques acting on the GMS axes (due to the motion of FMS axes) are also modeled in MATLAB/Simulink/SimMechanics environment to verify the model constructed in MATLAB/Simulink environment. Constructed MATLAB/ Simulink/SimMechanics model is shown in Figure 3.27. In MATLAB/Simulink /SimMechanics, the links of the FMS-GMS mechanism and the revolute joints between them are represented by the SimMechanics blocks “Body” and “Revolute”. The parameters needed for the “Body” block are the mass and inertia tensor of the link, and the position and orientation of the CG (Center of Gravity) and the points where the revolute joints are connected to the link while for the “Revolute” block, the axis of action should be entered.

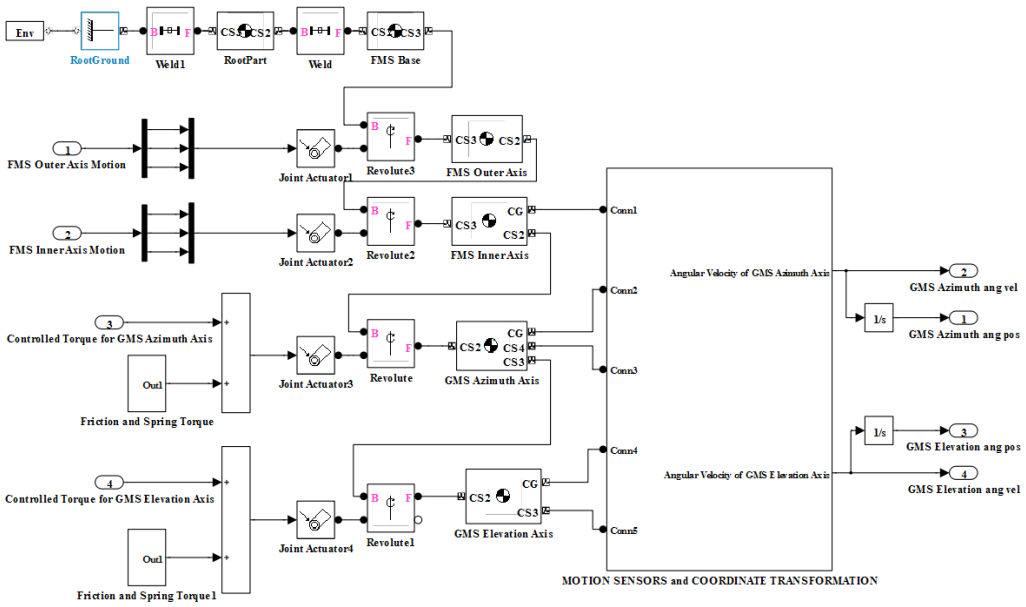

Figure 3.27. Disturbance torque model developed in MATLAB/Simulink/ SimMechanics environment

After the needed parameters are entered, “Body” and “Revolute” blocks are connected to each other. To simulate the actuation of the joints in the model, “Joint Actuator” blocks are used. “Joint Actuator” block has a parameter called “Actuate with” using which the joint could be actuated either with motion or generalized forces. For the model, the FMS axes are actuated with motion. The GMS axes, on the other hand, are actuated with torques.

3.4.3 Verification of the two disturbance torque models

Various FMS motion profiles are applied to both MATLAB/Simulink model and MATLAB/Simulink/SimMechanics model and the outputs are compared via plotting the outputs of both models on the same graphs. All the unknown parameters except the torque constants of the motors are set to zero for the simulations.

Actually, the inertia matrix of GMS azimuth gimbal depends on the angular position of the elevation gimbal. Since the elevation gimbal is mounted on the azimuth gimbal, the geometry of azimuth gimbal differs for different angular positions of the elevation gimbal. However, because of the limited angular freedom of the GMS elevation axis, which is constrained between ± 1.75 degrees by software and ± 3 degrees mechanically, this effect is not taken into consideration and so is not included in MATLAB/Simulink model. On the other hand, MATLAB/Simulink/ SimMechanics model includes this effect inherently. As a result, difference between the simulation results of the models for azimuth axis are expected to become larger as the elevation axis position departs from zero position because the axes angular positions are not limited in the simulations.

First case:

A constant angular acceleration of 5 deg/s^2 is applied to both FMS axes while the controlled torques for both axes are set to zero. The initial positions of the FMS axes at t=0 is shown in Figure 3.28. A relatively small angular acceleration is selected for this scenario to make it more meaningful. Since there is no controlled torque on the GMS axes, the application of a larger angular acceleration to FMS axes would cause GMS axes to move large angles (which actually is impossible since the angular freedom of both GMS axes are limited to few degrees in the real system). The results of the simulation for both models and the differences between them are shown in Figure 3.29 through Figure 3.32.

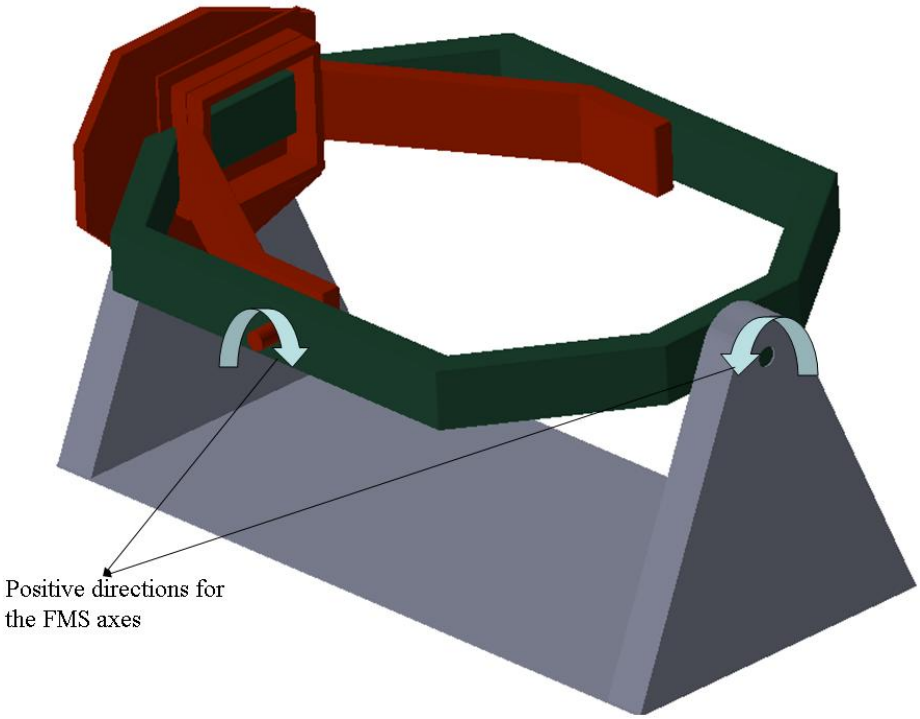

Figure 3.28. Initial position of FMS axes for the first case

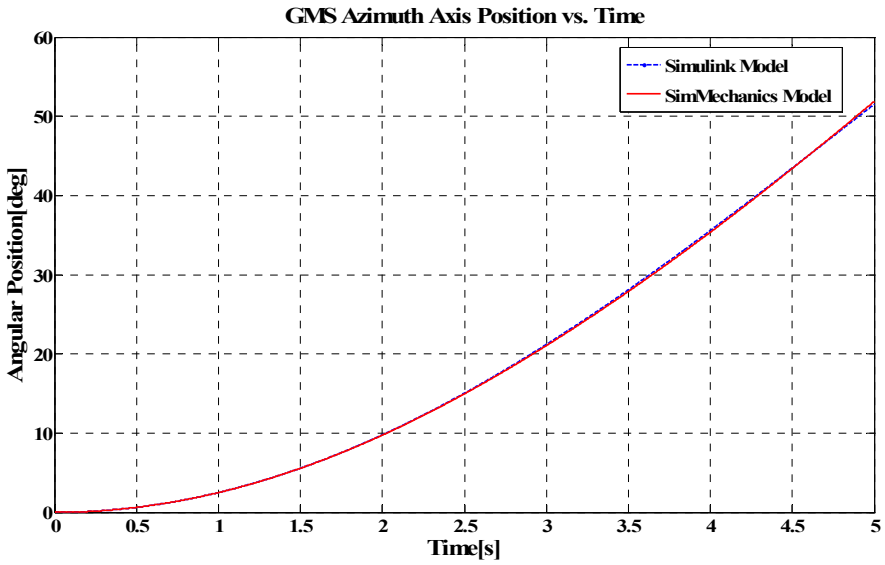

Figure 3.29. GMS azimuth axis position for the first case

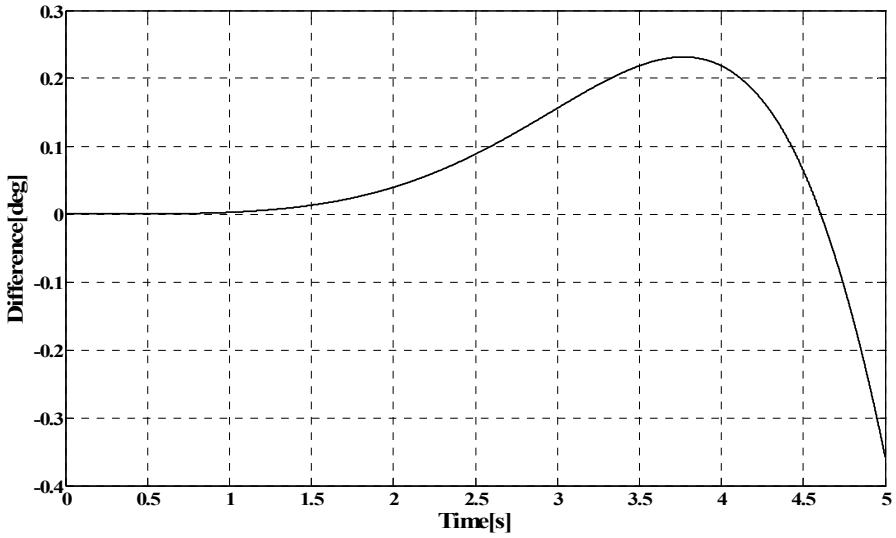

Figure 3.30. Angular position difference between model outputs for azimuth axis

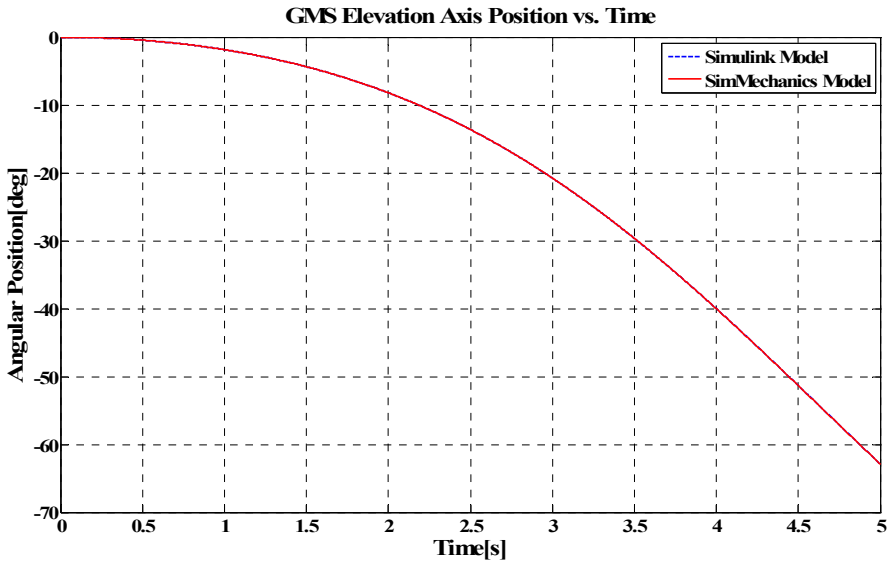

Figure 3.31. GMS elevation axis position for the first case

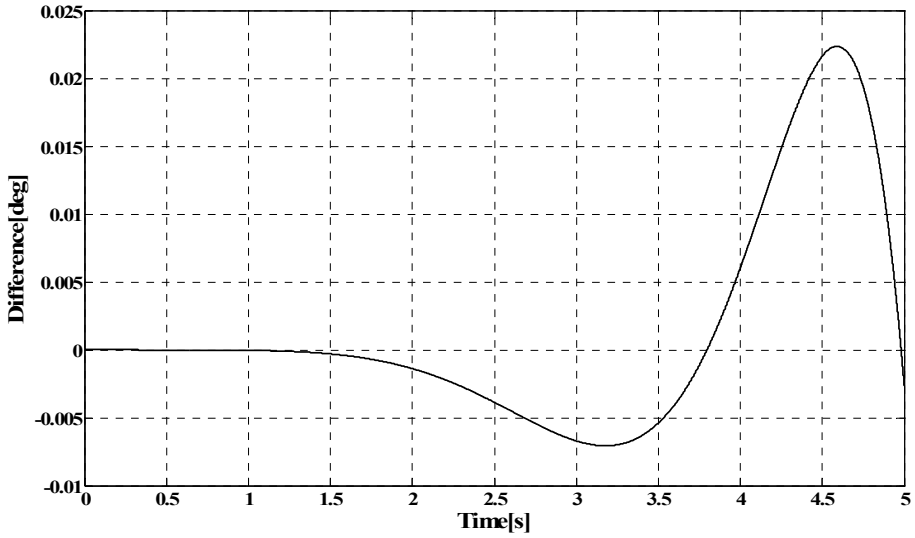

Figure 3.32. Angular position difference between model outputs for elevation axis

Second Case:

A sinusoidal position profile of 22 degrees amplitude and 0.81 Hz frequency is applied to both FMS axes while both GMS axes are commanded to zero angular position. The sinusoidal position profile is selected specifically for the FMS axes to reach their maximum velocity and acceleration limits during the motion. The PID parameters for the digital controller of motion control card are arbitrarily selected as K_p=1, K_d=100 and K_i=1 for both GMS axes. The initial positions of the FMS axes at t=0 is shown in Figure 3.33.The results of the simulation for both models and the differences between them are shown in Figure 3.34 through Figure 3.37.

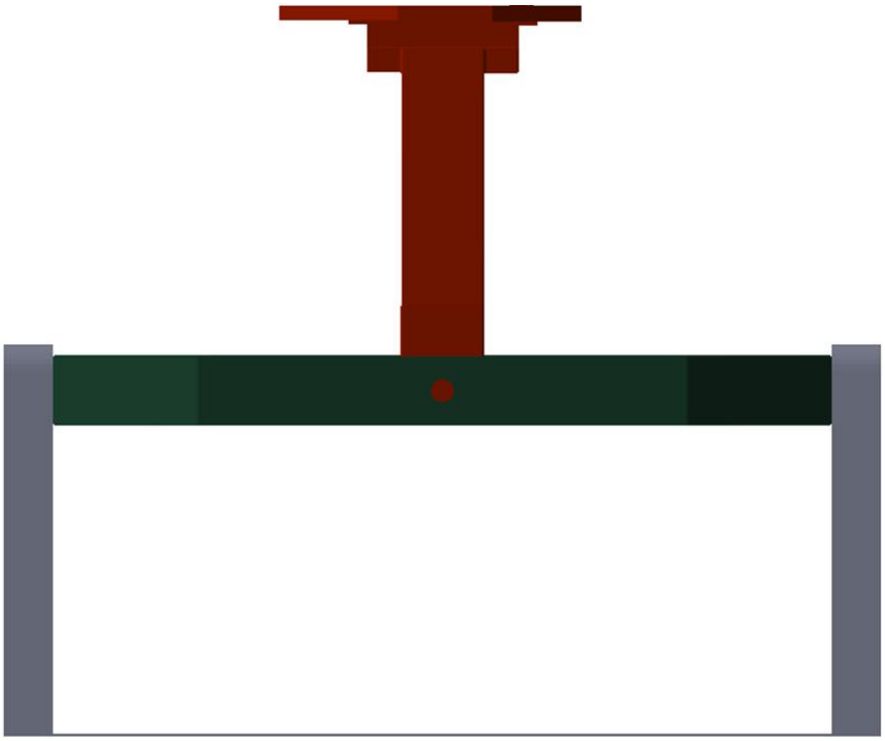

Figure 3.33. Initial position of FMS axes for the second case

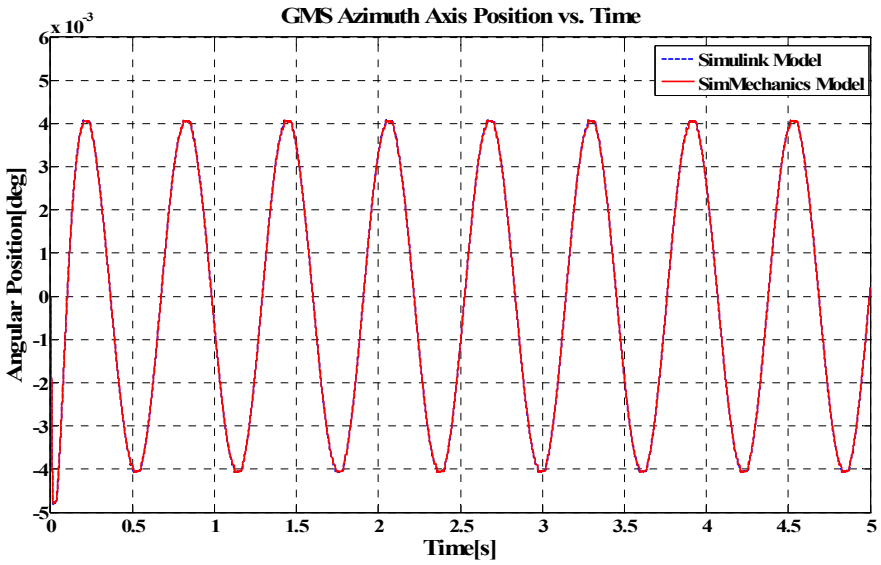

Figure 3.34. GMS azimuth axis position for the second case

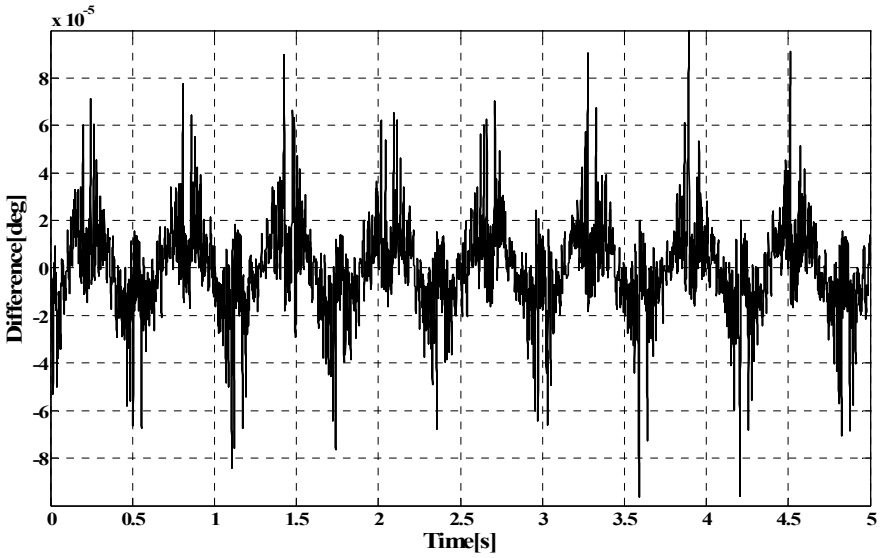

Figure 3.35. Angular position difference between model outputs for azimuth axis

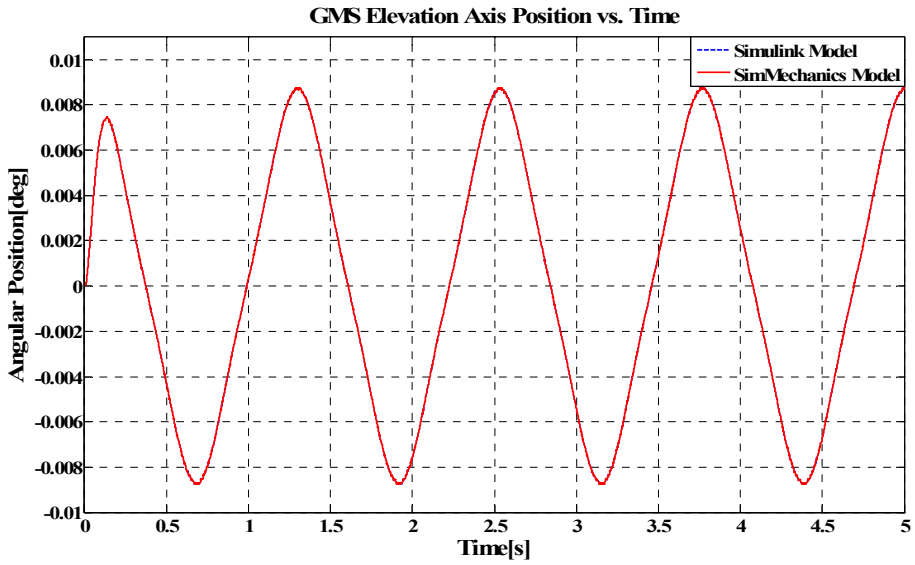

Figure 3.36. GMS elevation axis position for the second case

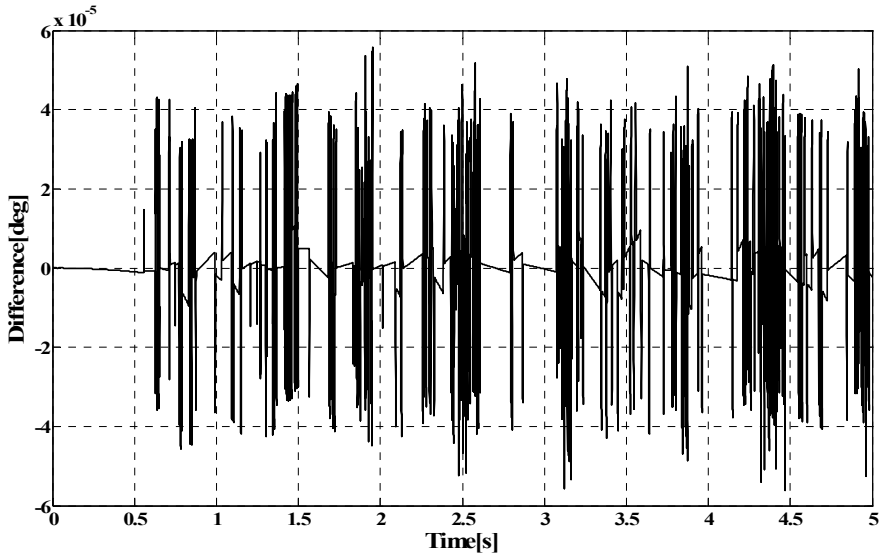

Figure 3.37. Angular position difference between model outputs for elevation axis

Results for the first case show that the difference between the azimuth angular position outputs increases as expected, as the elevation axis position departs from zero position. The maximum difference between the model outputs is 0.2 degrees for azimuth axis while 0.02 degrees for elevation axis. However, these difference values are attained at 35 degrees for azimuth axis and 60 degrees for elevation axis, which are not meaningful for the actual system since the angular motions of GMS axes are constrained between ± 1.75 degrees by software and ± 3 degrees mechanically in the real system. The error plots show that the differences between the model values for GMS axes angular positions, which are within the real system's range, are negligible. In addition to the first case conclusions, the second case results show that when controller torque is applied to the GMS axes, the differences between the model outputs decreases greatly compared to the first case which is expected. To summarize, both model outputs are in perfect agreement and hence the models developed for disturbance torques are verified.

CHAPTER 4

PARAMETER ESTIMATION

There are some parameters in the developed MATLAB/Simulink model, whose values are unknown. These unknown parameters are either the parameters of the GMS model or the parameters of the amplifier model. In order to obtain the complete model, the values of these unknown parameters should be determined. In this chapter, how the parameter estimation is performed by using MATLAB/Simulink Parameter Estimation Tool is explained and also the results of the estimation are discussed.

4.1. Parameters to be estimated

The parameters that are used in the developed MATLAB/Simulink model of the real-time control system are:

K_{dac}	Digital to Analog Converter (DAC) gain
K_a	Amplifier gain
$K_{t_{azm}}$	Torque constant of the motor used for GMS azimuth axis
$K_{t_{elv}}$	Torque constant of the motor used for GMS elevation axis
$K_{e_{azm}}$	Voltage constant of the motor used for GMS azimuth axis
$K_{e_{elv}}$	Voltage constant of the motor used for GMS elevation axis
$K_{rad2count}$	Conversion factor from radians to motor encoder counts

J_{azm}	Mass moment of inertia of the GMS azimuth axis
J_{elv}	Mass moment of inertia of the GMS elevation axis
b_{azm}	Viscous damping coefficient of GMS azimuth axis
b_{elv}	Viscous damping coefficient of GMS elevation axis
$T_{c_{azm}}$	Coulomb friction torque acting on GMS azimuth axis
$T_{c_{elv}}$	Coulomb friction torque acting on GMS elevation axis
k_{azm}	Spring constant of GMS azimuth axis
k_{elv}	Spring constant of GMS elevation axis
$T_{preload_{azm}}$	Torque acting on the GMS azimuth axis because of preloading of the cable
$T_{preload_{elv}}$	Torque acting on the GMS elevation axis because of preloading of the cable
$K_{p_{azm}}$	Proportional gain of the PI current regulator of the servo amplifier used for GMS azimuth axis
$K_{p_{elv}}$	Proportional gain of the PI current regulator of the servo amplifier used for GMS elevation axis
$K_{i_{azm}}$	Integral gain of the PI current regulator of the servo amplifier used for GMS azimuth axis
L_{azm}	Inductance of each coil of the three phase brushless motor of GMS azimuth axis
L_{elv}	Inductance of the coil of the single phase brushless motor of GMS elevation axis
R_{azm}	Resistance of each coil of the three phase brushless motor of GMS azimuth axis
R_{elv}	Resistance of the coil of the single phase brushless motor of GMS elevation axis

The numerical values for K_{dac}, K_a and $K_{rad2count}$ are known and given in the previous chapters. L_{azm}, L_{elv}, R_{azm} and R_{elv} are also known since they are given by the manufacturer. The numerical values for the other parameters should be estimated. The moment of inertia values of the GMS axes gimbals about their rotation axes are also estimated although their values are given. This is because the inertia values supplied are obtained from the solid model of the system hence the real values may not be exactly the same.

4.2. Estimation of the system parameters by using MATLAB/Simulink Parameter Estimation Tool

At first, the parameter estimation was thought to be performed for GMS axes and the amplifiers, separately. For the estimation of the amplifier parameters, the manipulation signal from the motion control card, which is the input to the amplifiers, and the current output of the amplifiers, should be collected. On the other hand, for the estimation of the GMS axes parameters, the current in the motor windings, which is the input to the GMS axes, and the angular position of the axes, should be collected.

The PID controller sampling frequency of the motion control card is 4 kHz while the PWM switching frequency of the amplifier is 33 kHz. For the collected command signal and current data to be meaningful, the sampling rate should at least be twice these values, which means the sampling rate should be at least 8 kHz for the command signal from the motion control card and 66 Hz for the current data. Collecting data at these rates is possible via DAQ cards; however, a suitable DAQ card could not be supplied for this study. The only way to collect the needed data was using the ADC ports of the motion control card. However, the execution of the LabView function *'Read ADC.vi'*, which is used to read the digital value of analog

voltage on ADC channels of the control card, takes about 2 ms. Hence, the fastest data sampling rate that can be achieved by using this function is less than 500 Hz. Furthermore, the time intervals between the collected data points are not equal if this method is used for collecting data. It is concluded that, by using the motion control card and LabView, it is not possible to collect the data as fast as needed to perform the parameter estimation for the components separately. Of course, instead of using LabView a driver for the motion control card can be written in any programming language and this way the data on ADC channels of the card can be read. This solution is not preferred since another method for system parameter estimation, which does not require handling of each system component separately, is possible.

When the whole control system is considered, the input to the system is the position commands sent to the motion control card and the output is the angular position of the GMS axis gimbals. It is possible to collect the angular position of GMS axes, at 333 Hz using the *'Acquire Trajectory Data.vi'* of LabView. Even though this sampling rate is not adequate to resemble the dynamics of the motion control card or amplifier, it is thought to be adequate to reflect the dynamics of the GMS axes since the response of the GMS axes are not expected to have spectral components beyond 100 Hertz.

MATLAB/Simulink Parameter Estimation Tool is used for parameter estimation. By using this tool and the constructed MATLAB/Simulink model, it is possible to estimate the model parameters by defining input and output ports on the model and introducing position commands and defining GMS axis gimbal positions as input-output pairs. For the parameter estimation of each GMS axis, a total of sixteen input-output data sets are used. Four different PID gain sets, which make the system stable, are used for each GMS axis as the PID parameters for the controller of the motion control card. For each PID gain set, four different inputs are applied to each GMS axis, which are; step position inputs with magnitudes of 0.009 and 0.018 degrees and position inputs of 0.009 and 0.018 degrees which are achieved by

setting the acceleration/deceleration of the GMS axes to 10.8 degrees/s^2 (60,000 counts/s^2).

The algorithm used for the parameter estimation task is the "pattern search" method which uses an advanced pattern search algorithm while the cost function that the estimation algorithm attempts to minimize is selected as the sum of squared errors. Pattern search methods belong to a class of optimization methods known as direct search methods. The algorithm starts with a finite initial step length. As the iteration approaches the solution, the algorithm reduces the length of the steps. Eventually, when the step length becomes smaller than a certain tolerance, the algorithm is said to be converged and the search stops. To perform the parameter estimation, some simulations of the model is run. A fixed step size of 250 microseconds is used for the simulations while the solver is selected as Dormand-Prince. Before performing the parameter estimation, feasible values for minimum and maximum limits and initial guesses are defined for each parameter to be estimated.

The real system outputs and the outputs of the Simulink model with the estimated parameters plotted on the same graphs and also the errors between the real system outputs and simulation outputs are shown in Figure 4.1 through Figure 4.16 for azimuth axis and in Figure 4.17 through Figure 4.32 for elevation axis.

As can be seen from the position graphs, the general trends of real system and simulation outputs are same. From error graphs it is observed that, the difference between real system and simulation outputs are generally in the order of 10^{-4} degrees. Parameter estimation results are slightly better for the elevation axis of GMS. For the azimuth axis, the error reaches to a maximum of 0.005 degrees while for the elevation axis, the maximum error attained is 0.002 degrees. This is due to the fact that; modeling the cable which is attached to elevation axis as a torsional spring is more accurate, since this cable is less prone to deformation. The parameter values obtained as the result of estimation are shown in Table 4.1 and Table 4.2.

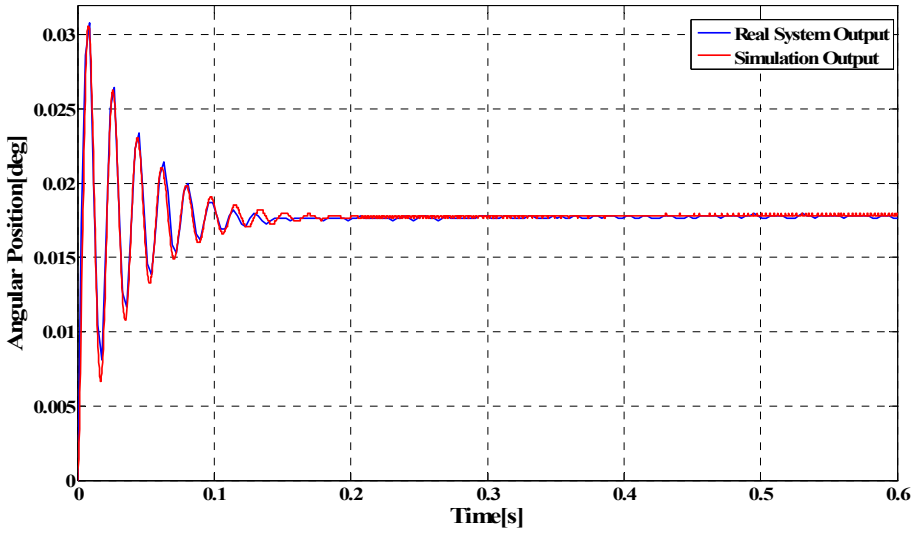

Figure 4.1. GMS azimuth axis response to 0.018 degrees step input (K_p=8, K_i=1, K_d=60)

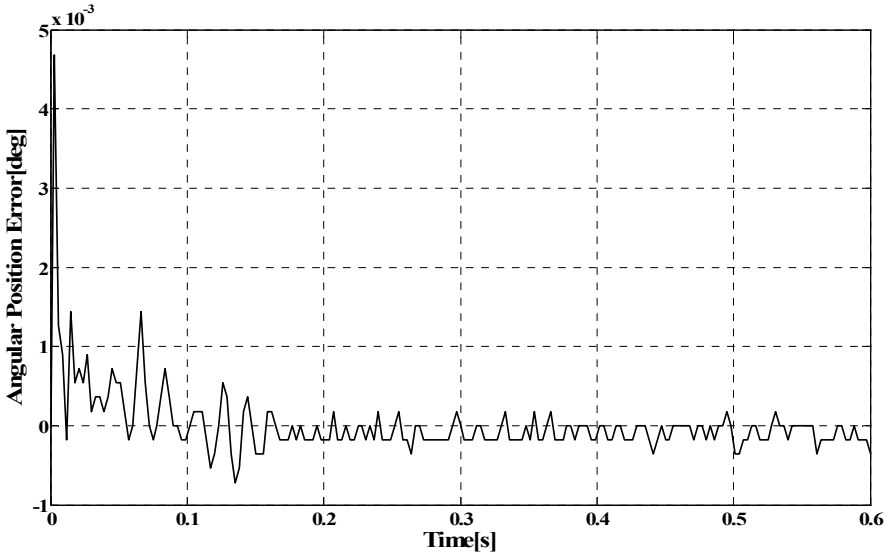

Figure 4.2. Error between real system and simulation outputs

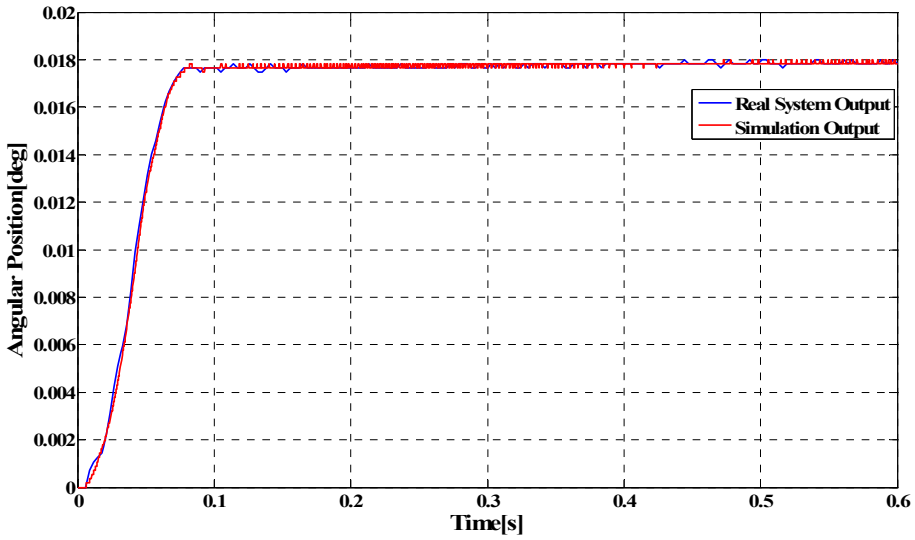

Figure 4.3. GMS azimuth axis response to 0.018 degrees input (K_p=8, K_i=1, K_d=60)

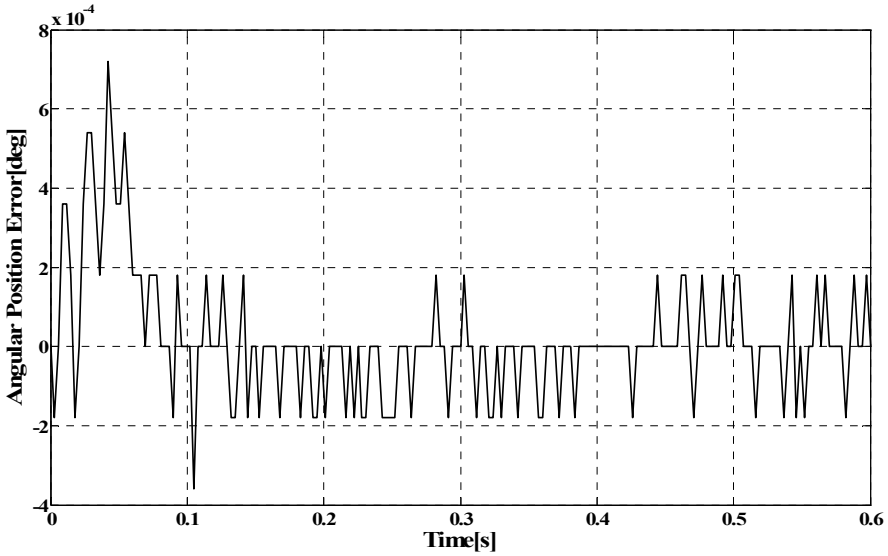

Figure 4.4. Error between real system and simulation outputs

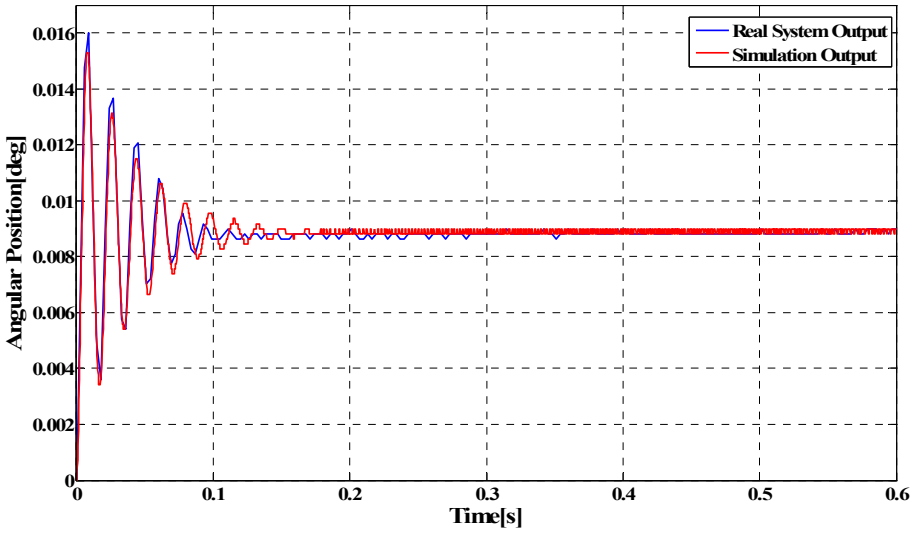

Figure 4.5. GMS azimuth axis response to 0.009 degrees step input (K_p=8, K_i=1, K_d=60)

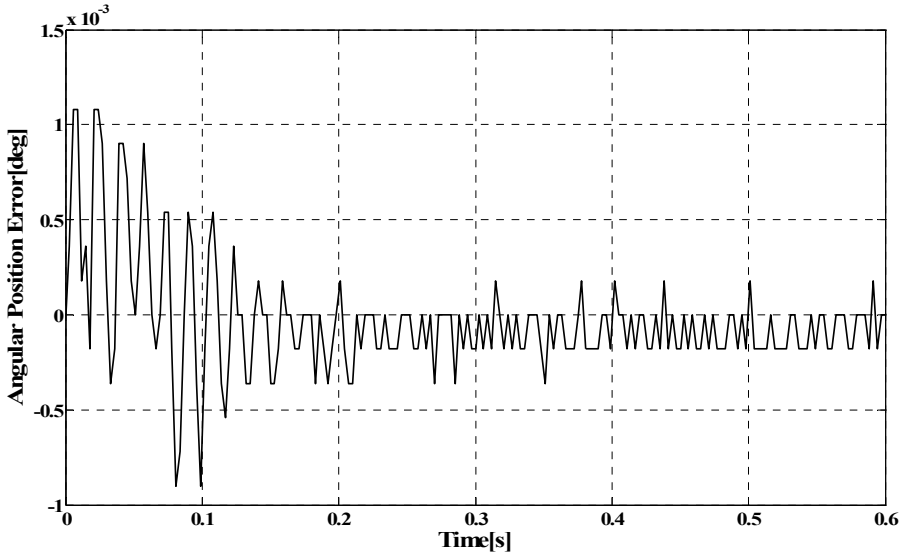

Figure 4.6. Error between real system and simulation outputs

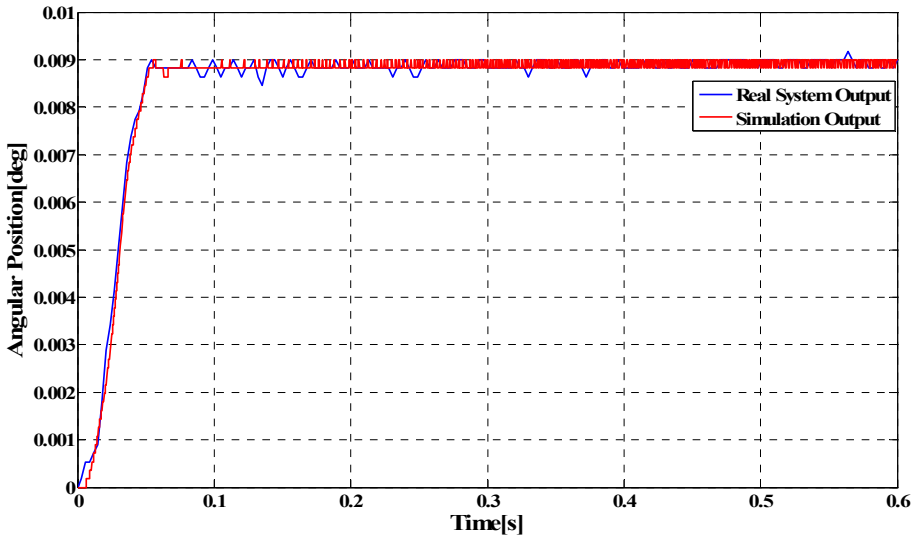

Figure 4.7. GMS azimuth axis response to 0.009 degrees input (K_p=8, K_i=1, K_d=60)

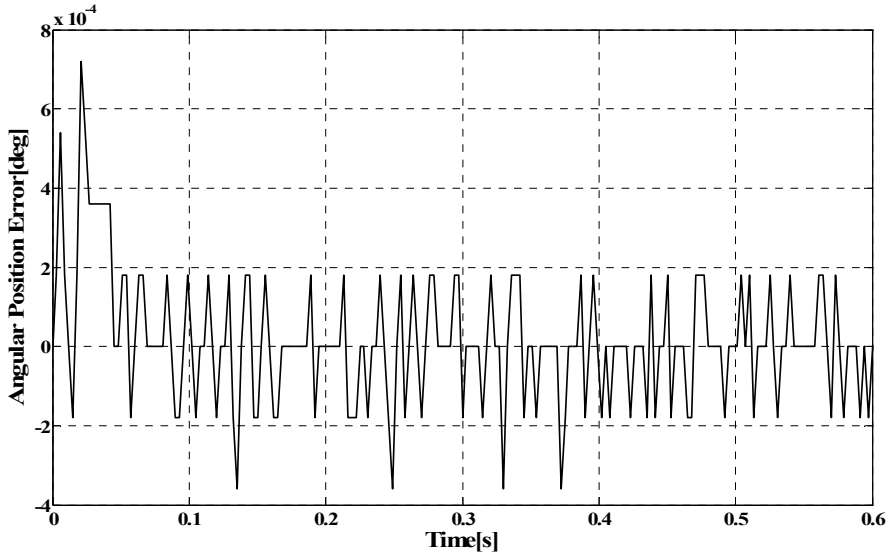

Figure 4.8. Error between real system and simulation outputs

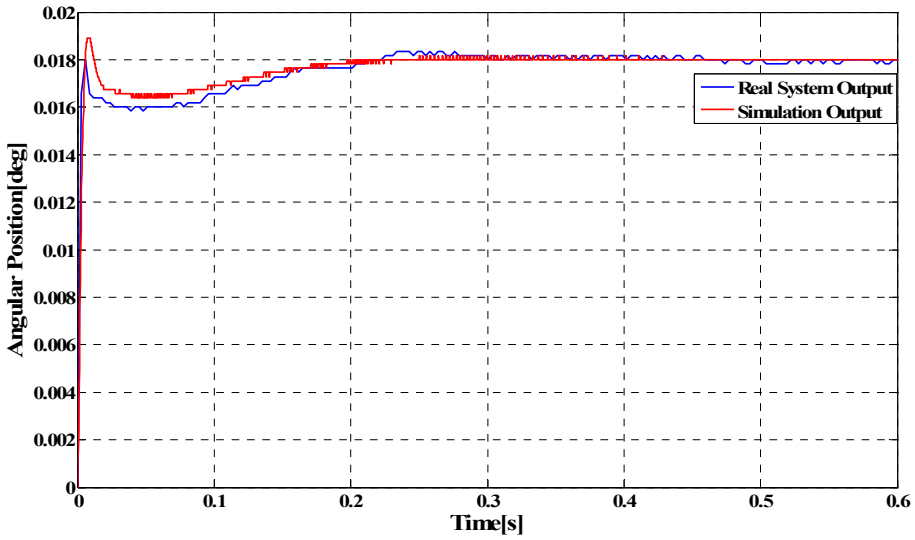

Figure 4.9. GMS azimuth axis response to 0.018 degrees step input (K_p=1, K_i=1, K_d=150)

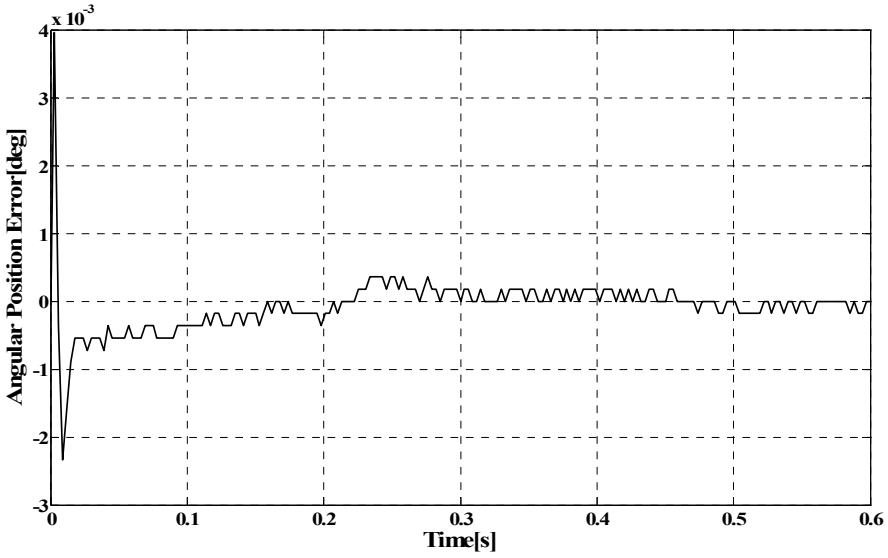

Figure 4.10. Error between real system and simulation outputs

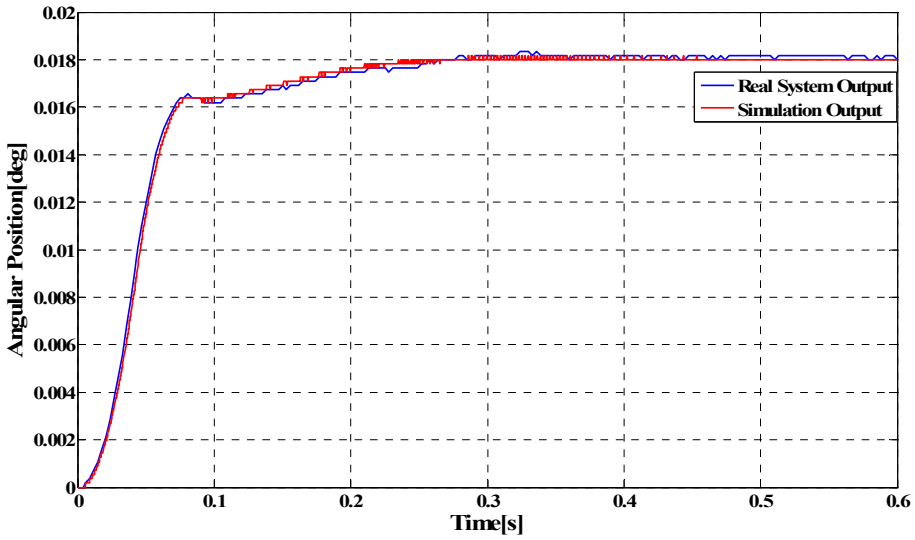

Figure 4.11. GMS azimuth axis response to 0.018 degrees step input (K_p=1, K_i=1, K_d=150)

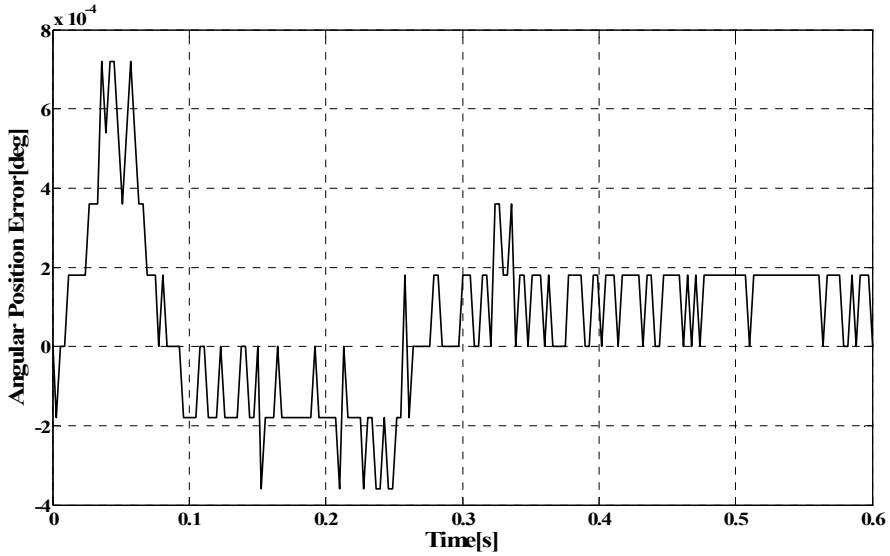

Figure 4.12. Error between real system and simulation outputs

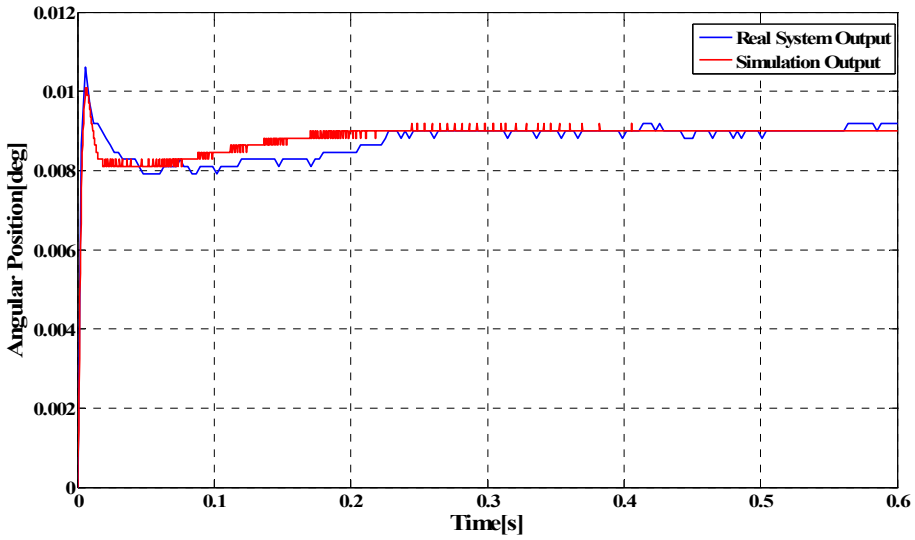

Figure 4.13. GMS azimuth axis response to 0.009 degrees step input (K_p=1, K_i=1, K_d=150)

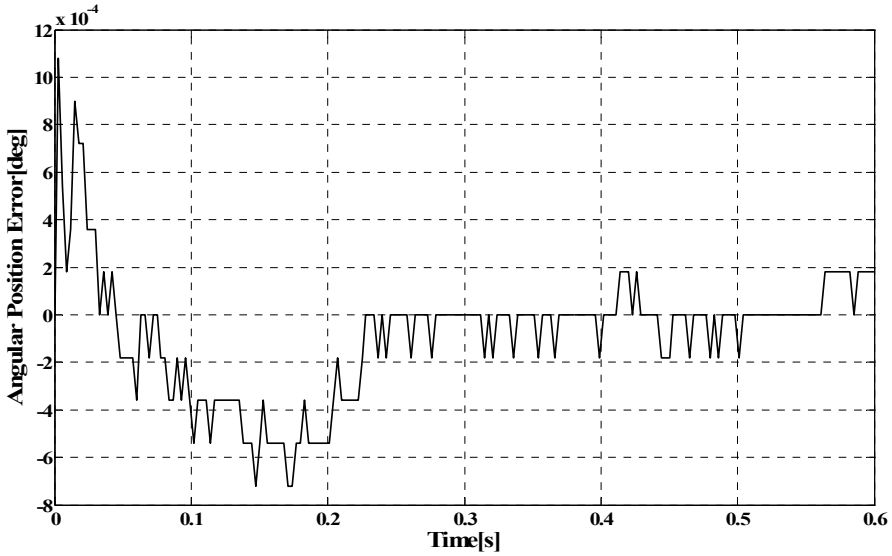

Figure 4.14. Error between real system and simulation outputs

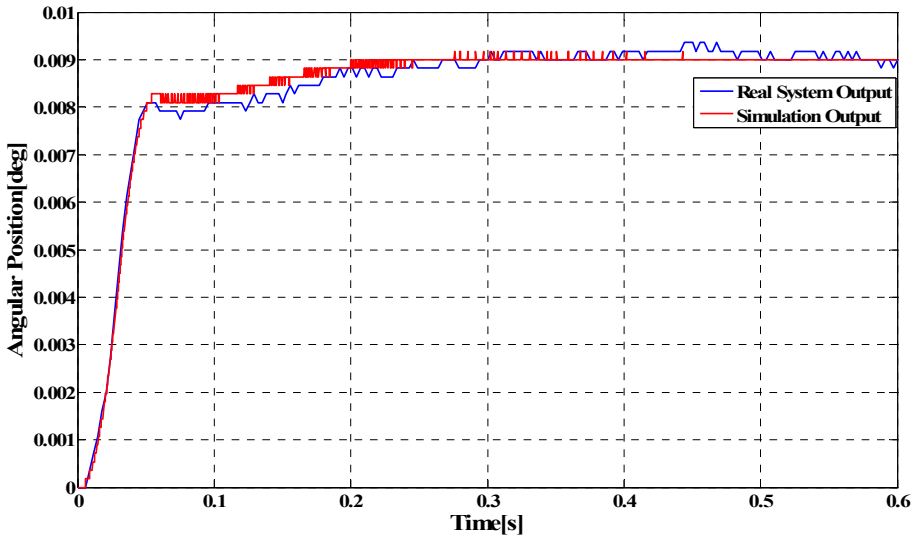

Figure 4.15. GMS azimuth axis response to 0.009 degrees input (K_p=1, K_i=1, K_d=150)

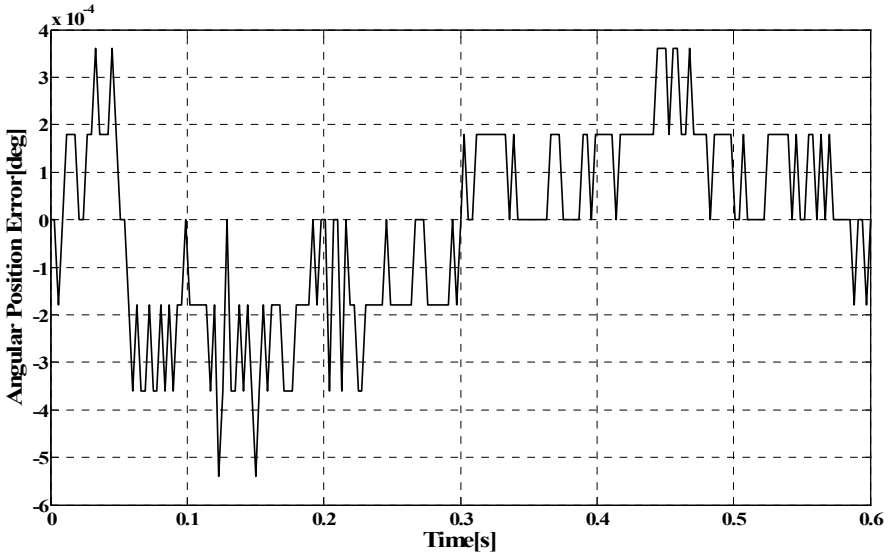

Figure 4.16. Error between real system and simulation outputs

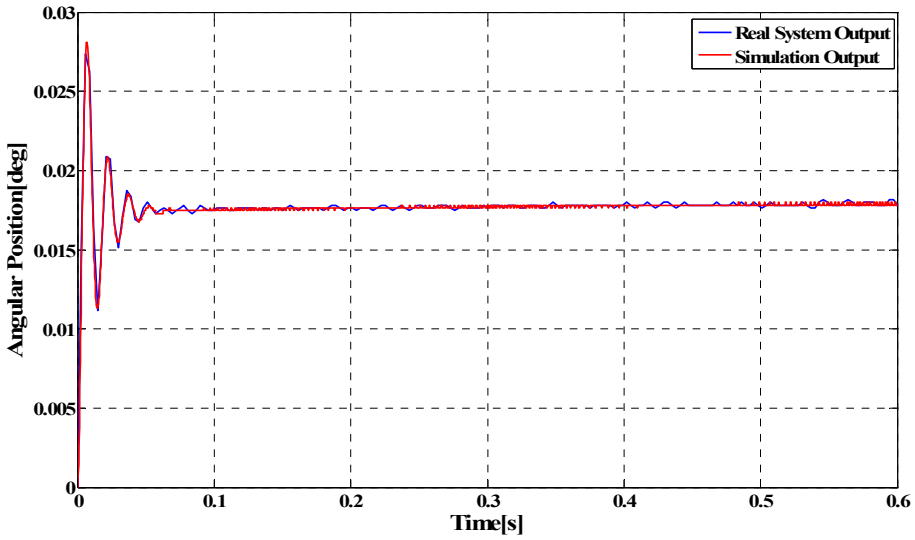

Figure 4.17. GMS elevation axis response to 0.018 degrees step input (K_p=5, K_i=1, K_d=20)

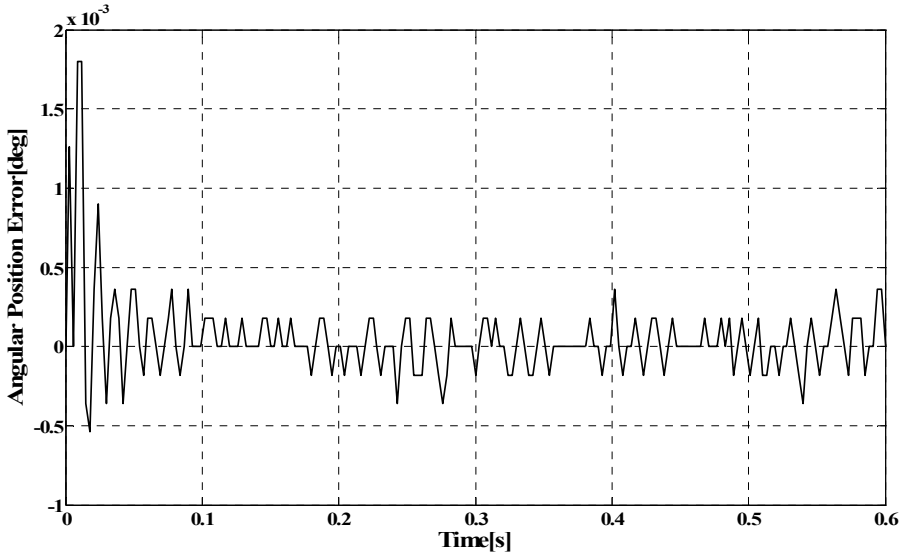

Figure 4.18. Error between real system and simulation outputs

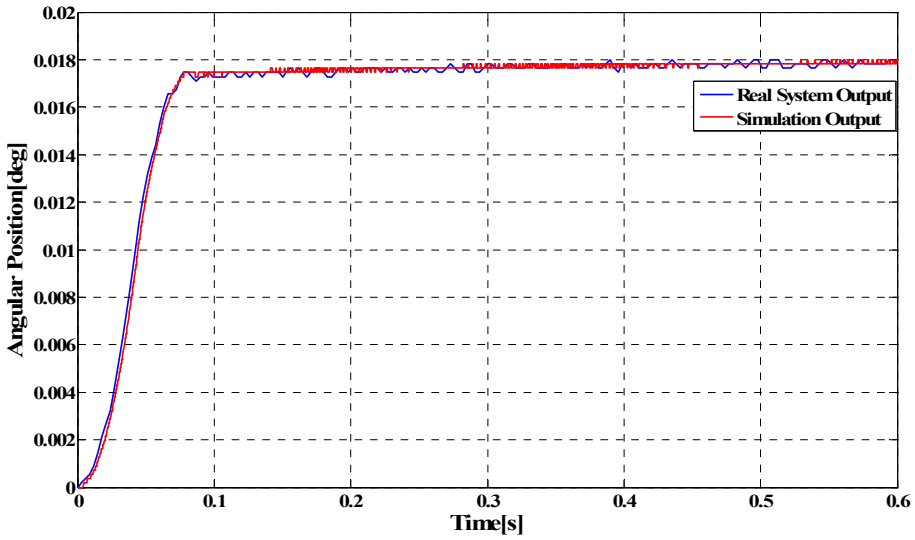

Figure 4.19. GMS elevation axis response to 0.018 degrees input (K_p=5, K_i=1, K_d=20)

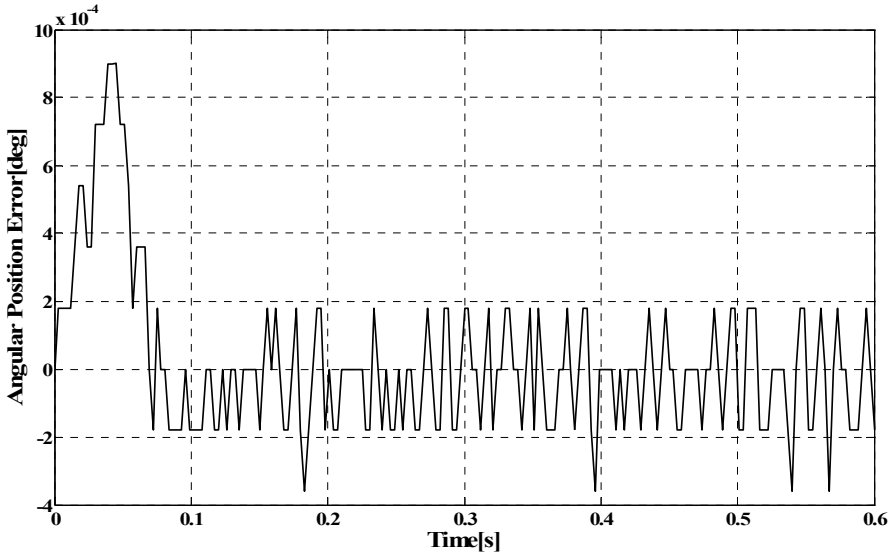

Figure 4.20. Error between real system and simulation outputs

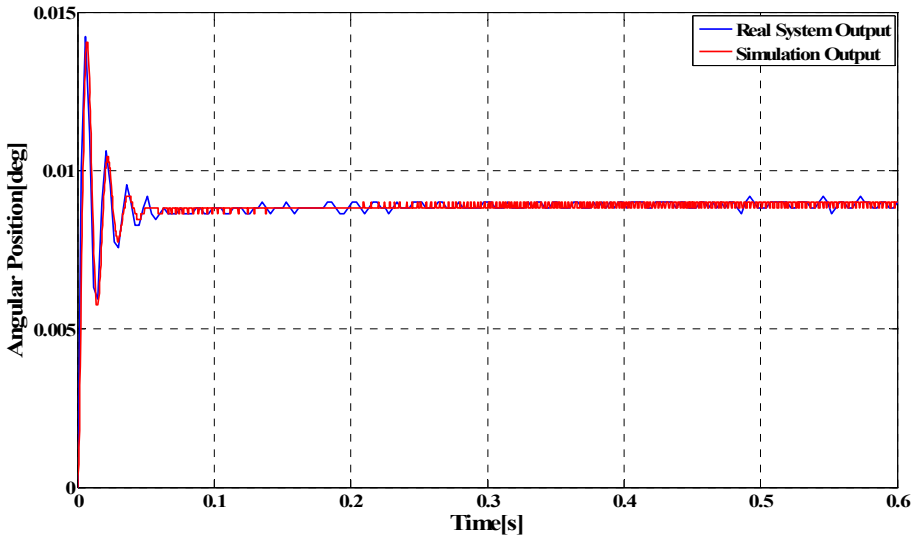

Figure 4.21. GMS elevation axis response to 0.009 degrees step input (K_p=5, K_i=1, K_d=20)

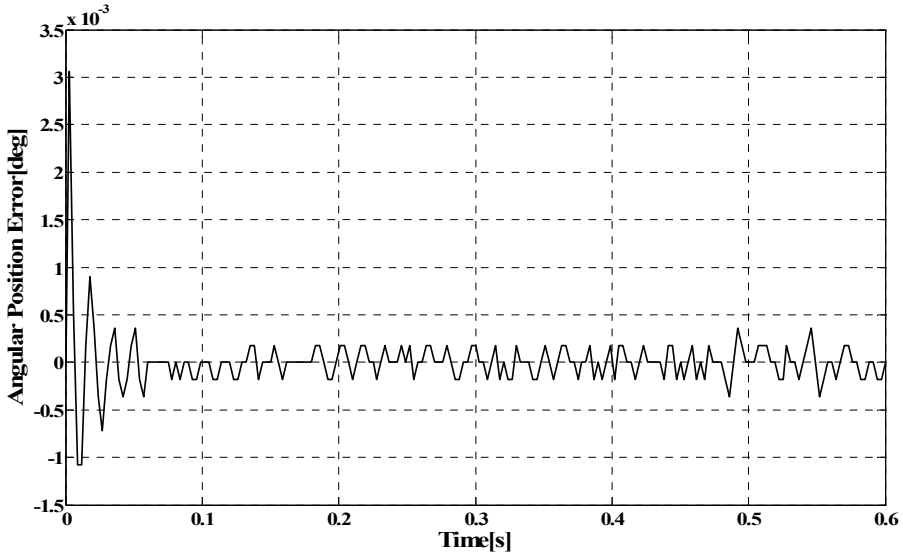

Figure 4.22. Error between real system and simulation outputs

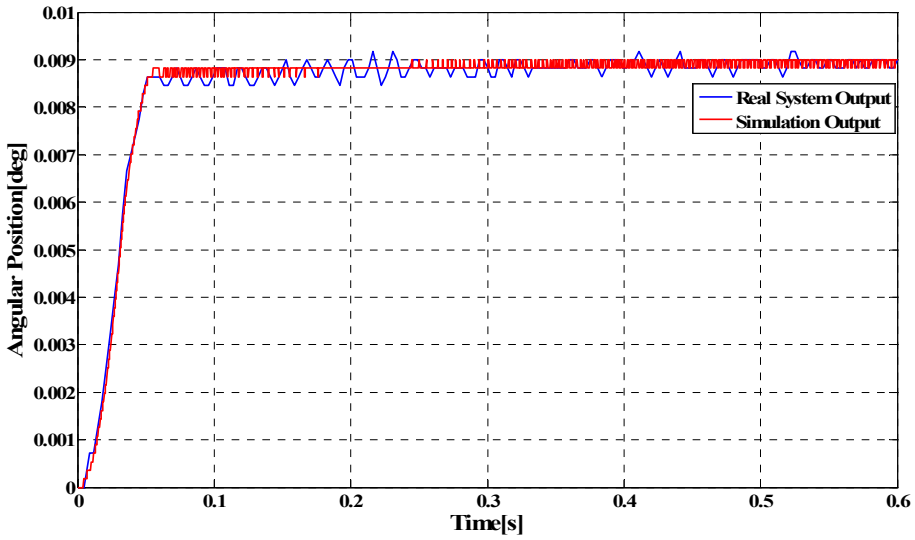

Figure 4.23. GMS elevation axis response to 0.009 degrees input (K_p=5, K_i=1, K_d=20)

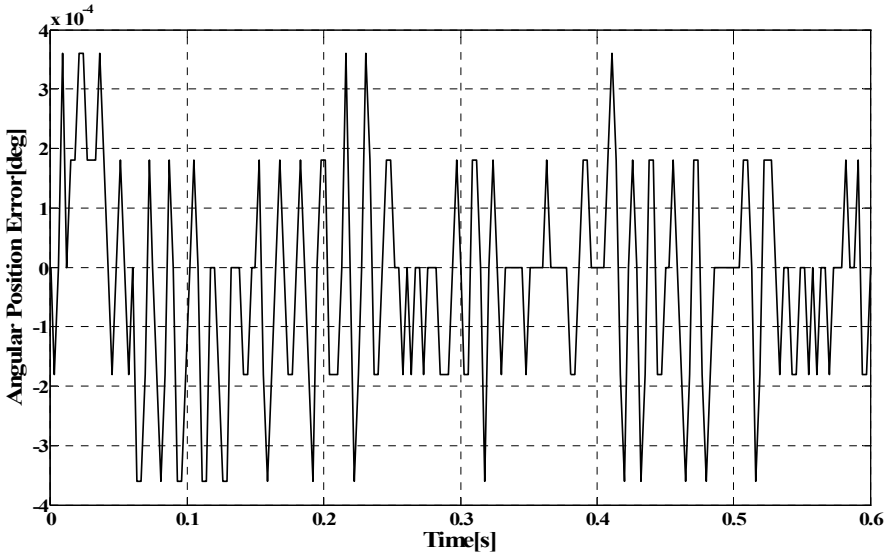

Figure 4.24. Error between real system and simulation outputs

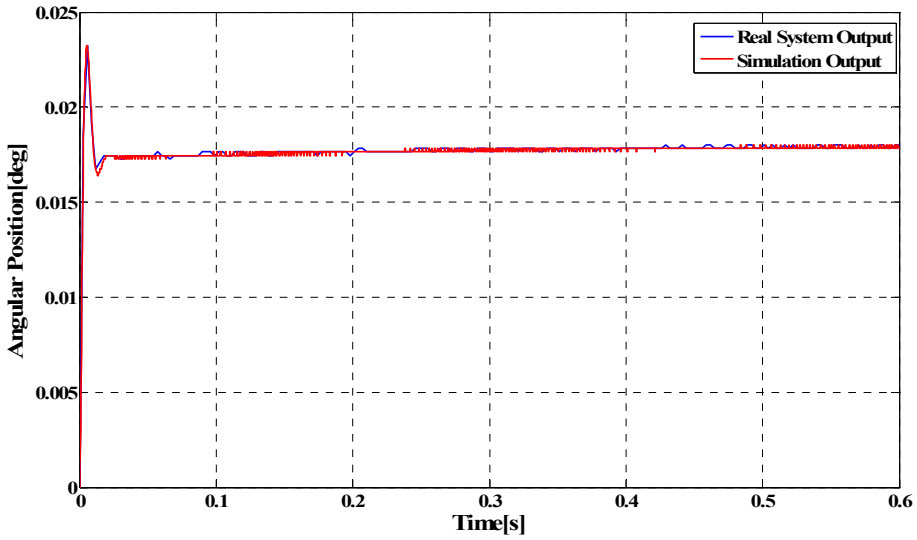

Figure 4.25. GMS elevation axis response to 0.018 degrees step input (K_p=5, K_i=1, K_d=50)

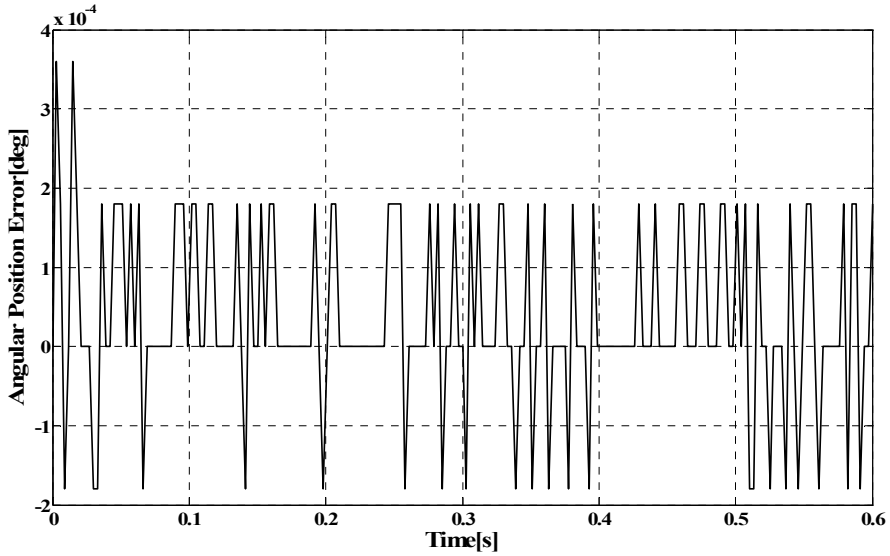

Figure 4.26. Error between real system and simulation outputs

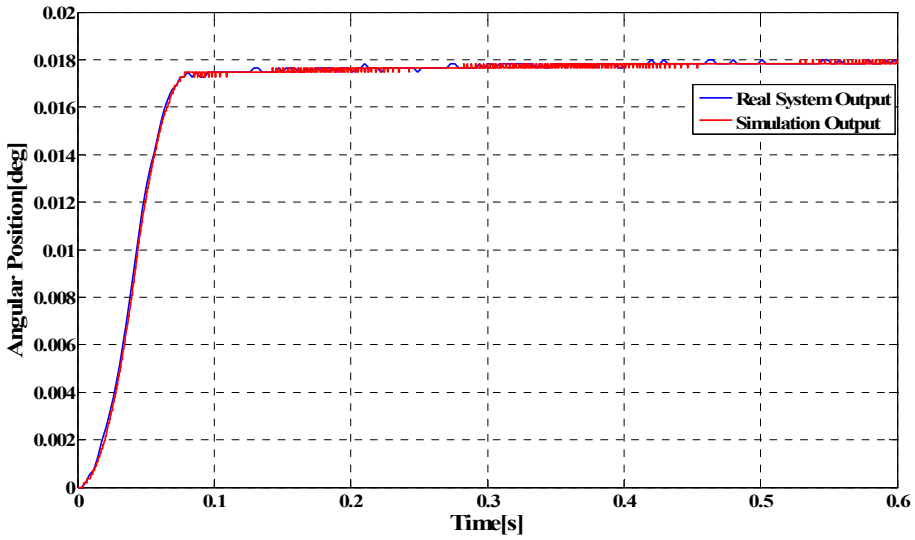

Figure 4.27. GMS elevation axis response to 0.018 degrees input (K_p=5, K_i=1, K_d=50)

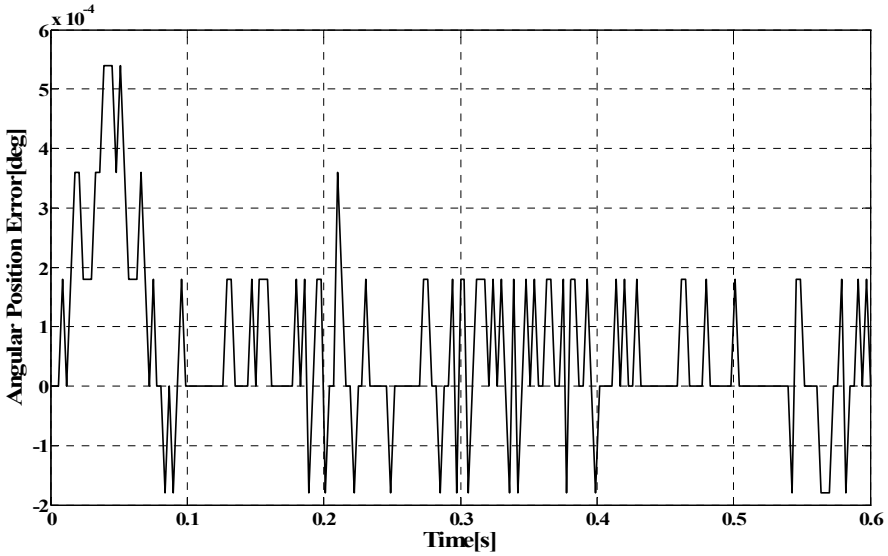

Figure 4.28. Error between real system and simulation outputs

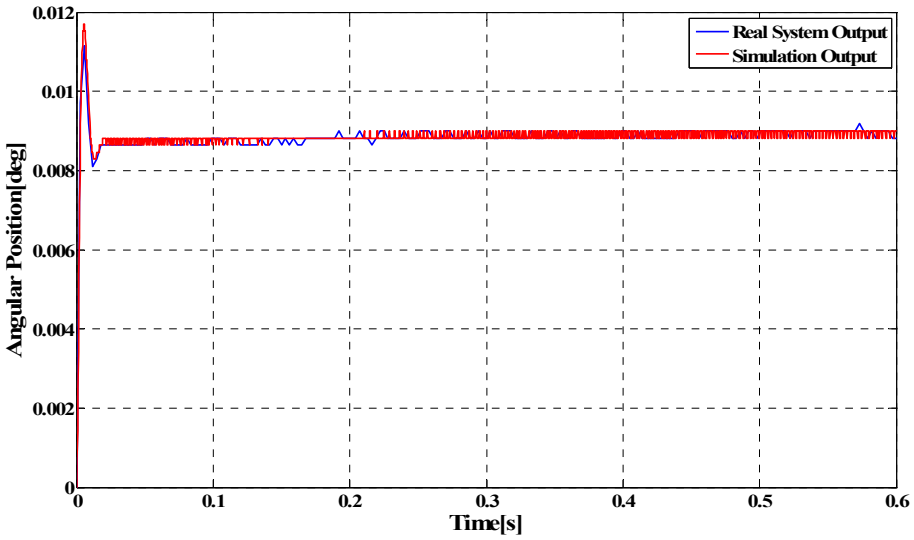

Figure 4.29. GMS elevation axis response to 0.009 degrees step input (K_p=5, K_i=1, K_d=50)

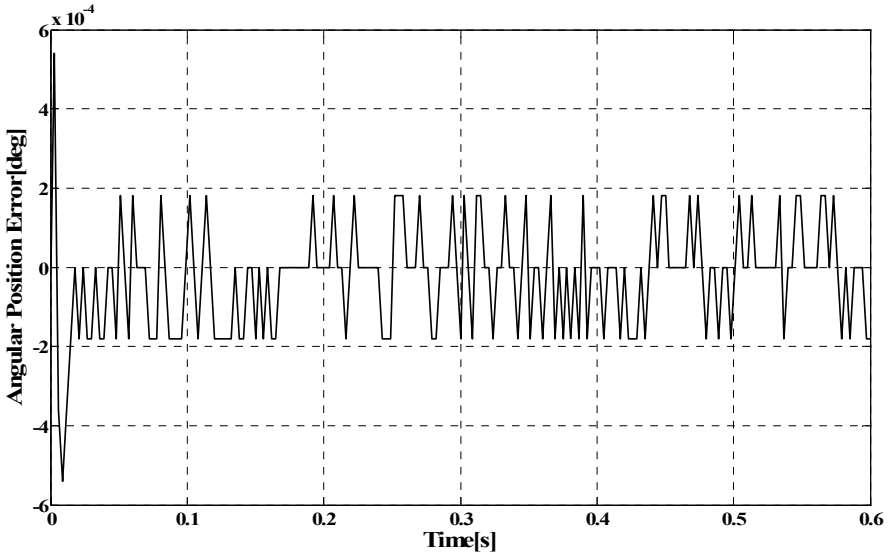

Figure 4.30. Error between real system and simulation outputs

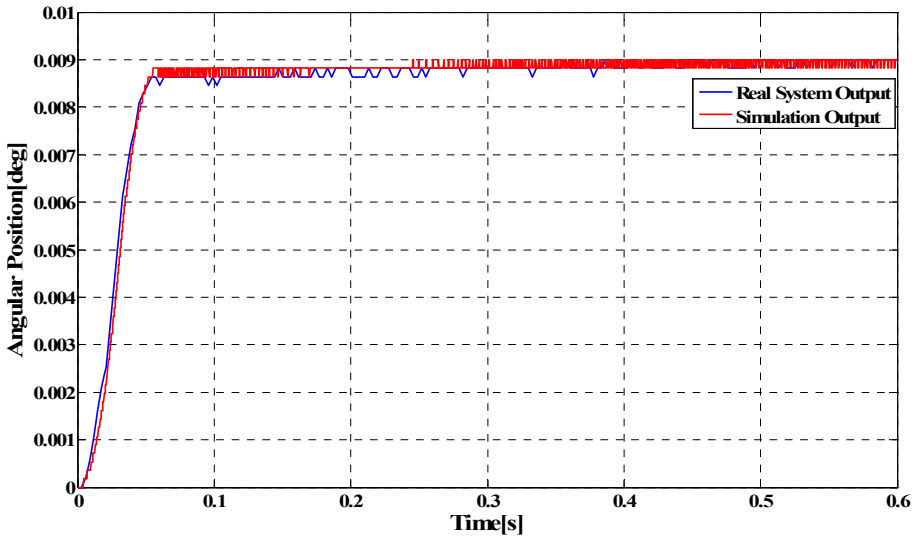

Figure 4.31. GMS elevation axis response to 0.009 degrees step input (K_p=5, K_i=1, K_d=50)

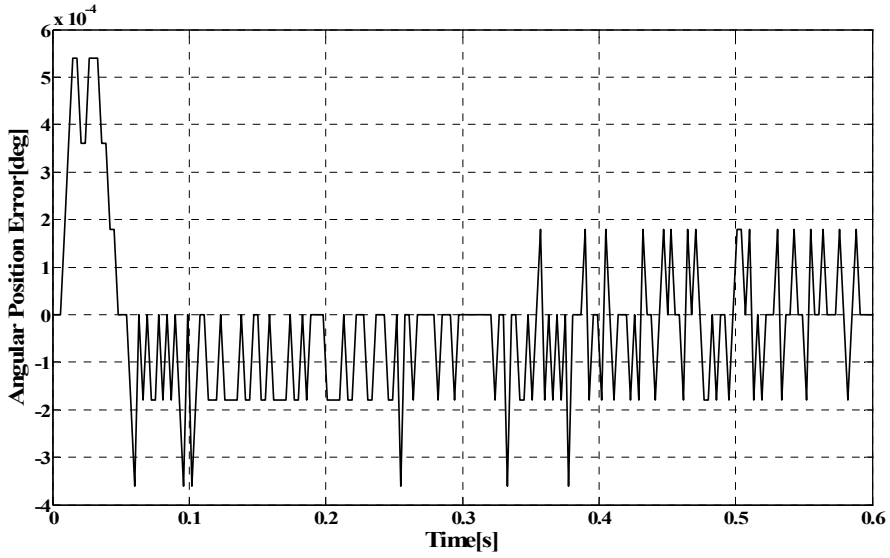

Figure 4.32. Error between real system and simulation outputs

Table 4.1. Estimated Values for GMS Azimuth Axis

Estimated Parameter	Estimated Value
J_{azm}	5.57×10^{-4} [kg.m^2]
b_{azm}	0.0203 [N.m.s/rad]
k_{azm}	3.30 [N.m/rad]
$K_{i_{azm}}$	1.28×10^{3} [V/V.s]
$K_{p_{azm}}$	4.87 [V/V]
$K_{t_{azm}}$	0.117 [N.m/A]
$K_{e_{azm}}$	0.113 [V.s/rad]
$T_{c_{azm}}$	2.27×10^{-5} [N.m]
$T_{preload_{azm}}$	1.21×10^{-4} [N.m]

Table 4.2. Estimated Values for GMS Elevation Axis

Estimated Parameter	Estimated Value
J_{elv}	7.45×10^{-5} [kg.m^2]
b_{elv}	2.02×10^{-3} [N.m.s/rad]
k_{elv}	0.529 [N.m/rad]
$K_{p_{elv}}$	1.07 [V/V]
$K_{t_{elv}}$	0.136 [N.m/A]
$K_{e_{elv}}$	0.141 [V.s/rad]
$T_{c_{elv}}$	2.37×10^{-6} [N.m]
$T_{preload_{elv}}$	1.80×10^{-5} [N.m]

The estimated values given in Table 4.1 and Table 4.2 are logical as:

- The inertia of azimuth axis is greater than the inertia of the elevation axis as expected.

- The spring constant and viscous friction coefficient values for both axes are in agreement with the rough values obtained as a result of the free oscillation tests performed for characterizing the cables.

- $K_{p_{azm}}$ value is greater than $K_{p_{elv}}$ value as expected. This was expected because the proportional gain values of the PI current regulators of the amplifiers are set as 'Decreased' for the elevation axis and set as 'Increased' for azimuth axis, using the related dipswitches on the amplifiers.

- $K_{t_{azm}}$, $K_{e_{azm}}$, $K_{t_{elv}}$ and $K_{e_{elv}}$ values are compared with these of motors of same size from the catalogues of different motor manufacturers. The values are in agreement.

- Actually, if one uses SI units, where the unit of the torque constant is N.m/A and the unit of the voltage constant is V.s/rad, numerical values of the torque constant and voltage constant should be the same. The values estimated for these constants are not the same but there exist some small differences between them about 3% for both axes. This is a reasonable outcome of the parameter estimation process, which could be taken as its success.

CHAPTER 5

CONTROLLER PARAMETER OPTIMIZATION

Since the motion control card has an embedded digital PID controller for each axis, the control algorithm that is used for position control of both GMS axes is PID control. Hence, the main task is to optimize the PID parameters of the digital PID controllers for each GMS axis. Firstly, the requirements for the motion of GMS axes should be determined.

5.1 Requirements for position control of GMS axes

The requirements for the real time position control of the GMS axes are derived from the HIL requirements related to motion of the IR spot in azimuth and elevation axes. These requirements are:

- The maximum angular velocity of the spots in both azimuth and elevation axes should not be smaller than 5°/s.
- The position accuracy of the spots in both axes should be less than 0.01°.
- Update rate of the position commands to the electromechanical components of IRSGS is 100 Hz.

When a mirror is turned by an angle θ, the deflected beam turns by an angle 2θ. So when any of the GMS axes is turned by an angle θ, then the IR spot will be moved by 2θ in the corresponding axis on the created IR scene (Figure 5.1).

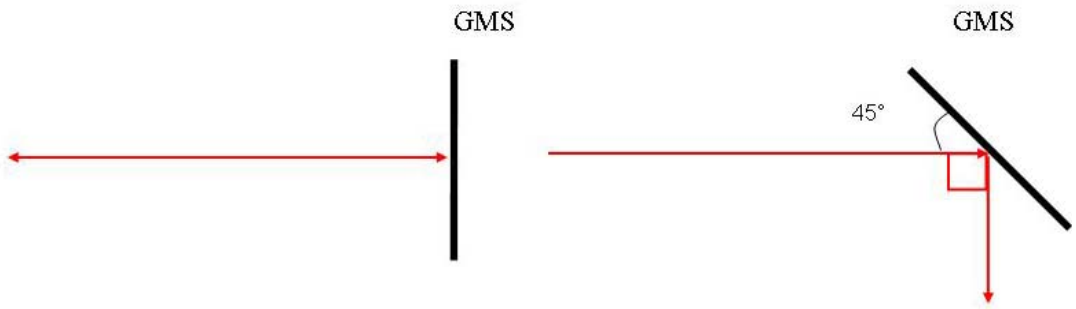

Figure 5.1. Reflection of a beam from GMS

The requirements for the GMS axes position control are determined as,

- The maximum angular velocity of GMS axes should not be smaller than 2.5°/s.
- The settling time for GMS axes should be less than 10 ms.
- The position accuracy of the GMS axes should be better than 0.005°.
- Since the maximum angular velocity requirement is 2.5°/s, the maximum position command that should be achieved in 10 ms is 0.025°. Since a position accuracy of 0.005° should be satisfied for a 0.025° position command the steady state error for a step input could not be higher than 20%. To be on the safe side, it is decided that the steady state error to a step input should be less than 10% of the input magnitude.

As long as GMS axes achieve a settling time of 10 ms, the overshoot of the position response is not very important, but still an excessive overshoot is naturally not preferred. In order to avoid large overshoots in the GMS axes position response, the following requirement for overshoot is also added:

- Maximum percent overshoot for GMS axes should be less than 25%.

5.2 Obtaining the open loop transfer functions for GMS axes in z-domain

Since the system is a hybrid system, which means it has both continuous states and discrete states, the transfer functions in the s-domain should be transformed to z-domain in order to be able to design the controller in z-domain. Transformation from s-domain to z-domain by using ZOH (Zero Order Hold) method is given in detail in the following sections. For the controller design of both GMS axes, nonlinear effects (Coulomb friction, preloading of the cables, limiters such as the integral limit and PID output limit of the motion control card or amplifier saturation voltage) are neglected.

The direct digital design method is used for this study. However, design by emulation method could also be used to design discrete time controllers as well as the direct digital design method, since the sampling rate is high and the quantization interval is small. The direct digital design method is selected since the controllers to be designed are digital controllers and designing them in their true domain seems more appropriate.

Azimuth axis:

The conceptual block diagram of the control system for GMS axes is shown in Figure 5.2 and analytical block diagram is shown in Figure 5.3.

To obtain the transfer function between the controller output, $M(s)$ and the angular position of the azimuth axis, $\theta(s)$, following equations are used.

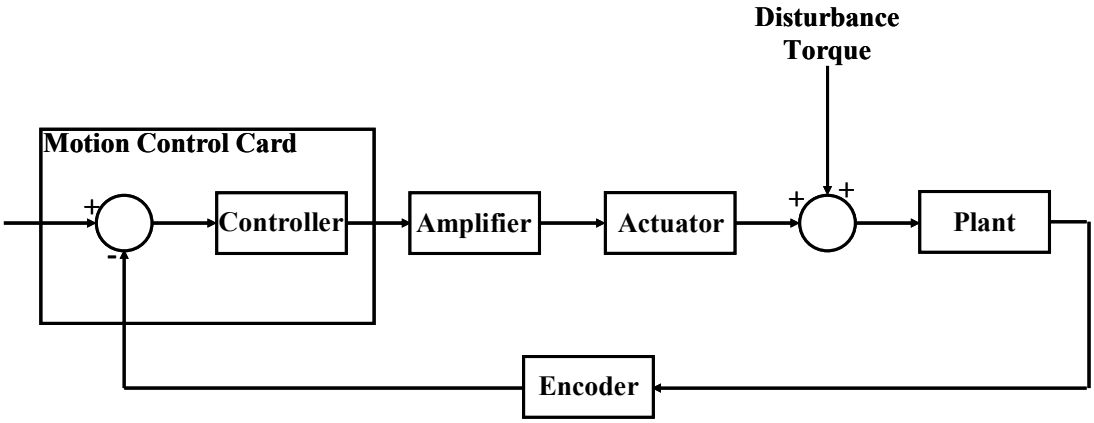

Figure 5.2. Conceptual block diagram of the control system

$$\left(K_a M(s) - I(s)\right)\left(K_p + \frac{K_i}{s}\right) = V(s) \tag{5.1}$$

$$V(s) = V_{bemf}(s) + RI(s) + LsI(s) \tag{5.2}$$

$$V_{bemf}(s) = K_e s\theta(s) \tag{5.3}$$

$$T(s) = K_t I(s) \tag{5.4}$$

$$T(s) = Js^2\theta(s) + bs\theta(s) + k\theta(s) \tag{5.5}$$

The numerical values of the parameters for GMS azimuth axis are,

$$K_{dac} = \frac{20}{2^{16}} \cong 0.000305 \text{ [V/LSB]}$$

$$K_a = 1.6 \text{ [A/V]}$$

$$K_t = 0.117 \text{ [N.m/A]}$$

$$K_e = 0.113 \text{ [V.s/rad]}$$

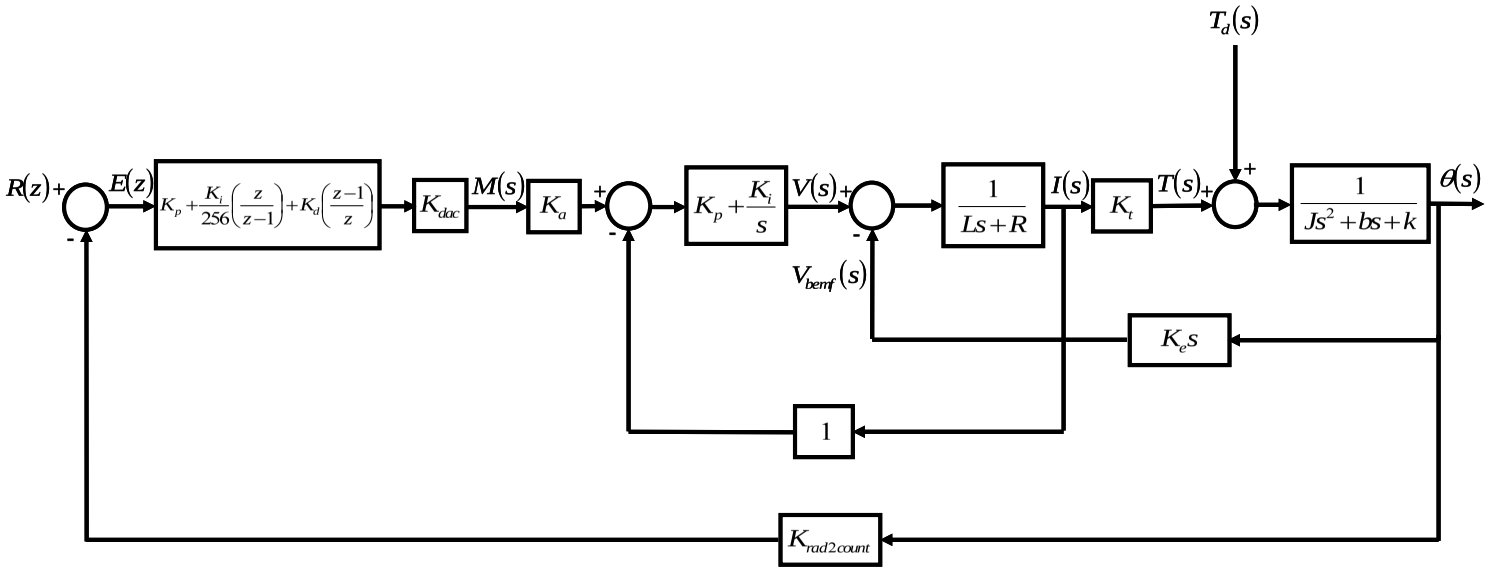

Figure 5.3. Analytical block diagram of the control system

$$K_{rad\,2count} = \frac{2000000}{2\pi} \cong 318310 \text{ [encoder count/rad]}$$

$J = 5.57 \times 10^{-4}$ [kg.m^2]

$b = 0.0203$ [N.m.s/rad]

$k = 3.30$ [N.m/rad]

$K_p = 4.87$ [V/V]

$K_i = 1.28 \times 10^3$ [V/V.s]

$L = 0.0033$ [H]

$R = 10.7$ [Ω]

Solving (5.1), (5.2), (5.3), (5.4) and (5.5) together, the transfer function between the controller output, $M(s)$ and the angular position of the azimuth axis, $\theta(s)$ is obtained as,

$$\frac{\theta(s)}{M(s)} = G_{\theta M}(s) = \frac{(K_t K_a K_p)s + K_t K_a K_i}{(JL)s^4 + (bL + JK_p + JR)s^3 + (kL + JK_i + bK_p + bR + K_t K_e)s^2 + (kK_p + kR + bK_i)s + kK_i} \tag{5.6}$$

When the numerical values of the parameters are used in (5.6);

$$\frac{\theta(s)}{M(s)} = G_{\theta M}(s) = \frac{0.912s + 240.1}{1.825 \times 10^{-6} s^4 + 0.008744s^3 + 1.056s^2 + 77.43s + 4230} \text{ [rad/V]} \tag{5.7}$$

Open loop transfer function can be obtained as follows;

$$G_{OL}(s) = G_{\theta M}(s) K_{rad\,2count} K_{dac} = \frac{88.59s + 23320}{1.825 \times 10^{-6} s^4 + 0.008744s^3 + 1.056s^2 + 77.43s + 4230}$$
[encoder count/LSB] (5.8)

The discrete open loop transfer function is obtained by transforming the continuous time domain open loop transfer function into discrete time domain by using 'ZOH' (Zero Order Hold) method.

The discrete time domain open loop transfer function is expressed as,

$$G_{OL}(z) = Z\left\{\left(\frac{\left(1-z^{-1}\right)}{s}\right)G_{OL}(s)\right\} \tag{5.9}$$

This transformation is performed by using an m-file named *'oltf_for_gms_axes.m'* written in MATLAB. The code is given in Appendix C. The discrete time open loop transfer function is obtained as,

$$G_{OL}(z) = \frac{9.759\times10^{-5}\, z^3 + 0.0002109\, z^2 - 0.0002274 z - 5.206\times10^{-5}}{z^4 - 3.281 z^3 + 3.864 z^2 - 1.885 z + 0.3019} \tag{5.10}$$

Elevation axis:

The block diagram of the control system for GMS elevation axis is same as the one for GMS azimuth axis and shown in Figure 5.2. The transfer function between the controller output for the elevation axis and the angular position of the elevation axis is nearly the same with the transfer function between the controller output for the azimuth axis and the angular position of the azimuth axis. The only difference is that, the integral control of the PI current regulator of the amplifier used for the elevation axis is deactivated.

So by setting $K_i = 0$ in (5.6), the transfer function between the controller output for the elevation axis, $M(s)$ and the angular position of the elevation axis, $\theta(s)$ is obtained as,

$$\frac{\theta(s)}{M(s)} = G_{\theta M}(s) = \frac{\left(K_t K_a K_p\right)}{(JL)s^3 + \left(bL + JK_p + JR\right)s^2 + \left(kL + bK_p + bR + K_t K_e\right)s + \left(kK_p + kR\right)} \quad (5.11)$$

The numerical values of the parameters for the elevation axis are:

$K_{dac} = \frac{20}{2^{16}} \cong 0.000305$ [V/LSB]

$K_a = 1.6$ [A/V]

$K_t = 0.136$ [N.m/A]

$K_e = 0.141$ [V.s/rad]

$K_{rad\,2count} = \frac{2000000}{2\pi} \cong 318310$ [encoder counts/rad]

$J = 7.45 \times 10^{-5}$ [kg.m^2]

$b = 2.02 \times 10^{-3}$ [N.m.s/rad]

$k = 0.529$ [N.m/rad]

$K_p = 1.07$ [V/V]

$L = 0.0027$ [H]

$R = 8.5$ [Ω]

When these numerical values of the parameters are used in (5.11);

$$\frac{\theta(s)}{M(s)} = G_{\theta M}(s) = \frac{0.2323}{2.012 \times 10^{-7} s^3 + 0.0007187\, s^2 + 0.03992\, s + 5.06} \text{ [rad/V]} \quad (5.12)$$

So the open loop transfer function can be obtained as follows;

$$G_{OL}(s) = G_{\theta M}(s) K_{rad\,2count} K_{dac} = \frac{22.57}{2.012 \times 10^{-7} s^3 + 0.0007187\, s^2 + 0.03992\, s + 5.06}$$

[encoder count/LSB] (5.13)

The transformation from s-domain to z-domain by using 'ZOH' method is again performed by using the m-file named *'oltf_for_gms_axes.m'* written in MATLAB, whose code is given in Appendix C. The discrete time open loop transfer function is obtained as,

$$G_{OL}(z) = \frac{0.0002368z^2 + 0.0007691z + 0.0001518}{z^3 - 2.401z^2 + 1.811z - 0.4095} \tag{5.14}$$

5.3 Optimization of digital PID controller parameters by using MATLAB/ SISO Design Tool

To design the discrete PID controllers for GMS axes MATLAB/SISO Design Tool is used. Firstly, obtained discrete time open loop transfer functions are imported by MATLAB/SISO Design Tool. Then using the root locus, the placements of the closed loop poles are decided. Stages of the controller design by using MATLAB/SISO Design Tool are given in detail in the following sections.

Azimuth axis:

The root locus of the discrete time domain open loop transfer function is plotted and shown in Figure 5.4.

The controlled system should have a maximum 10 ms settling time when a step input is applied and the maximum overshoot should not be greater than 25%. If these requirements are mapped to the z-plane, it is concluded that the dominant closed loop poles should lie in the white region shown in Figure 5.5 to achieve time domain specifications.

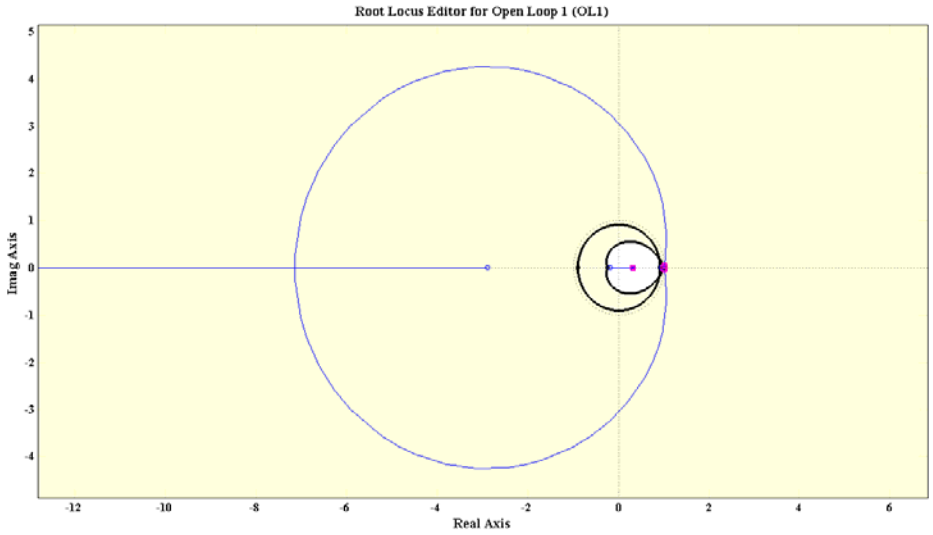

Figure 5.4. Root Locus for GMS azimuth axis (1/3)

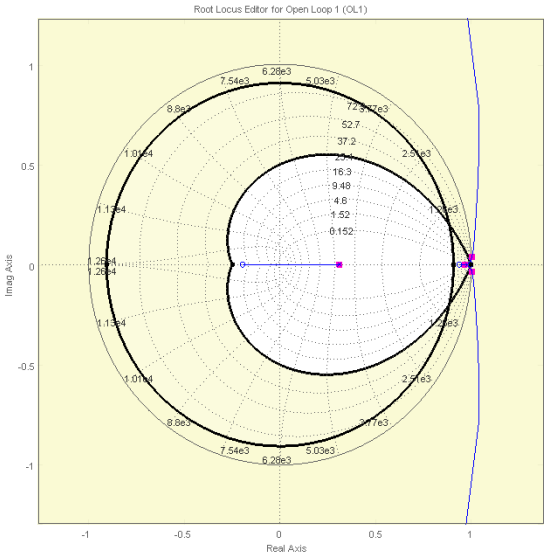

Figure 5.5. Root Locus for GMS azimuth axis (2/3)

The transfer function of the PID controller was obtained in Chapter 3 and it is given by (3.11). By equating the denominators and simplifying,

$$G_c(z)=\left(K_p+\frac{K_i}{256}+K_d\right)\frac{\left(z^2-\frac{\left(K_p+2K_d\right)}{\left(K_p+\frac{K_i}{256}+K_d\right)}z+\frac{K_d}{\left(K_p+\frac{K_i}{256}+K_d\right)}\right)}{(z-1)z} \tag{5.15}$$

As can be seen from (5.15), two poles of the controller are at z=0 and z=1. So there are three free parameters of the controller which are the location of the two zeros and the gain of the controller. Firstly, poles of the PID controller which are at z=0 and z=1 is added to the system. After the poles of the controller are placed at their location, the root locus takes the form in Figure 5.6.

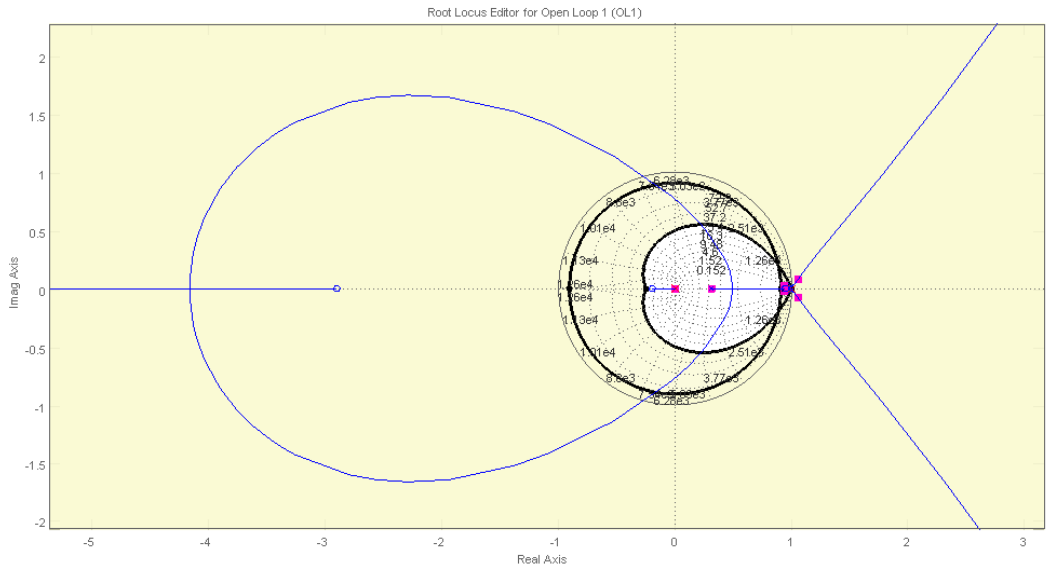

Figure 5.6. Root Locus for GMS azimuth axis (3/3)

From Figure 5.6 it can be seen that the two complex conjugate open loop poles make the system unstable. To make the system stable and to be able to make the two branches, which go to infinity, to pass through the region of design where the closed loop poles should be placed, the zeros of the controller should be placed near to the open loop poles at z=1 and z=0.979. This way these open loop poles end up at the two zeros placed near to them and the complex conjugate open loop poles would move towards the inside of the unit circle and one of them would end up at the open loop zero at z=0.936.

When the two zeros of the controller are placed at z=0.999 and z=0.978, as expected, the complex conjugate poles move towards inside of the unit circle. Since the effects of the open loop poles at z=1 and z=0.979 are canceled by the zeros of the controller, the complex conjugate poles dominate the system behavior of the system. The last thing to do is placing the two dominant closed loop poles inside the predefined area of success by adjusting the gain.

With the gain adjusted to 290 and the zeros of the PID controller placed at z=0.999 and z=0.978, the closed loop poles are located inside the predefined area as shown in Figure 5.7. The designed PID controller is;

$$G_c(z) = 290\left(\frac{(z-0.999)(z-0.978)}{(z-1)z}\right) \tag{5.16}$$

By equating (5.16) to (5.15), the parameters of the PID controller are obtained as, $K_p = 6.66$, $K_i = 1.63$ and $K_d = 283.34$. However, since the PID parameters of the digital controller of the motion control card can only be set as integers, the values are rounded up to the nearest integers. So the PID parameters for the azimuth axis are set as, $K_p = 7$, $K_i = 2$ and $K_d = 283$. With the designed controller, the unit step response of the closed loop system is as shown in Figure 5.8. As can be seen from

Figure 5.8, the maximum percent overshoot, settling time and steady state error specifications are satisfied.

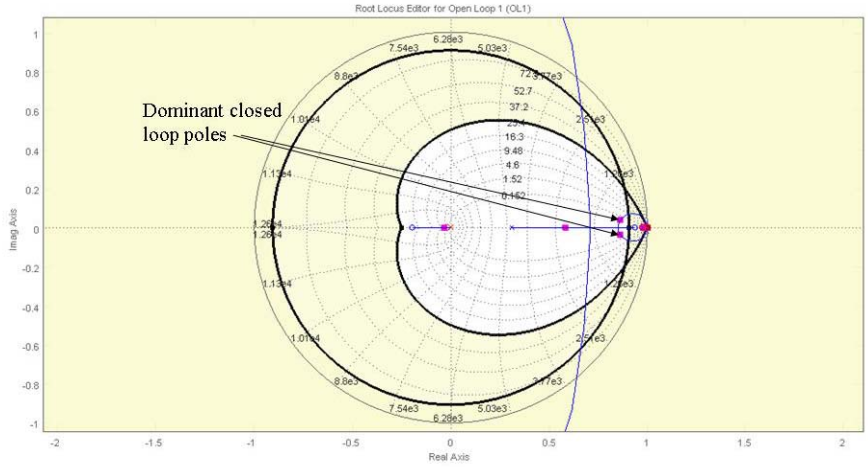

Figure 5.7. Location of the dominant closed loop poles for GMS azimuth axis

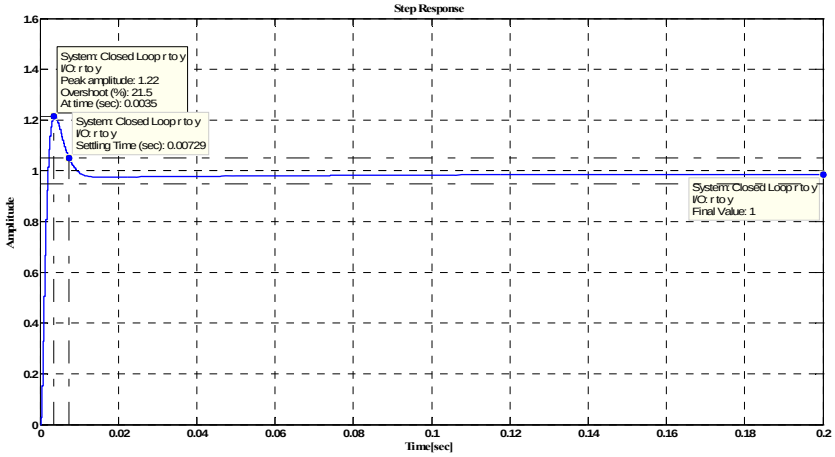

Figure 5.8. Unit step response of the GMS azimuth axis with the designed controller

Elevation axis:

The root locus of the discrete time domain open loop transfer function is plotted and shown in Figure 5.9. The controlled system should have a maximum 10 ms settling time when a step input is applied and the maximum overshoot should not be greater than 25%. If these requirements are mapped on to the z-plane, it is concluded that the dominant closed loop poles should lie in the white region shown in Figure 5.10 to achieve time domain specifications.

Again, the first thing to do is to add the poles of the PID controller which are at z=0 and z=1 is to the system. After the poles of the controller are placed at their location, the root locus takes the form shown in Figure 5.11.

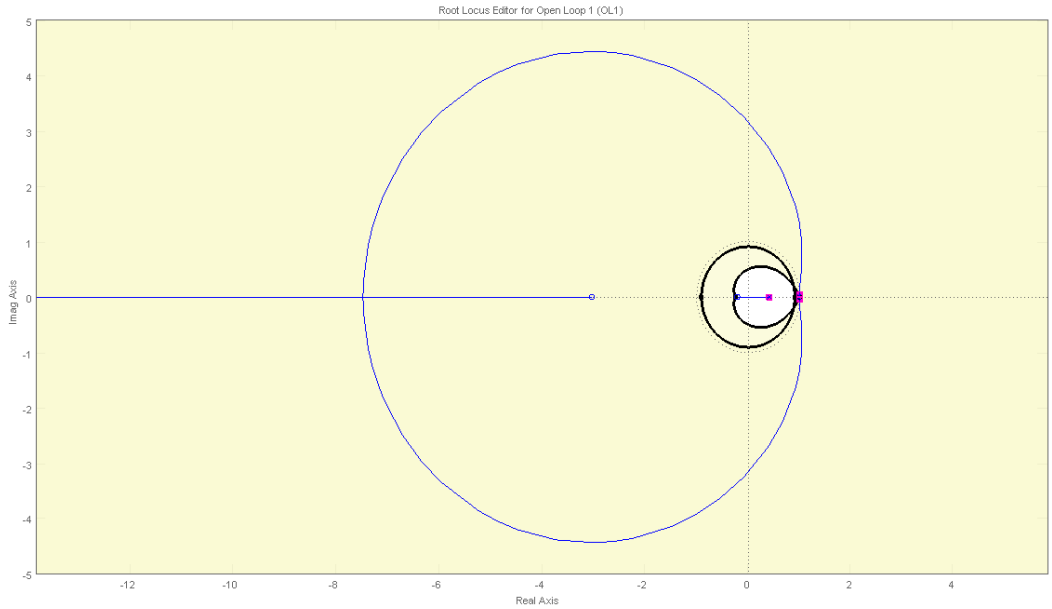

Figure 5.9. Root locus for GMS elevation axis (1/3)

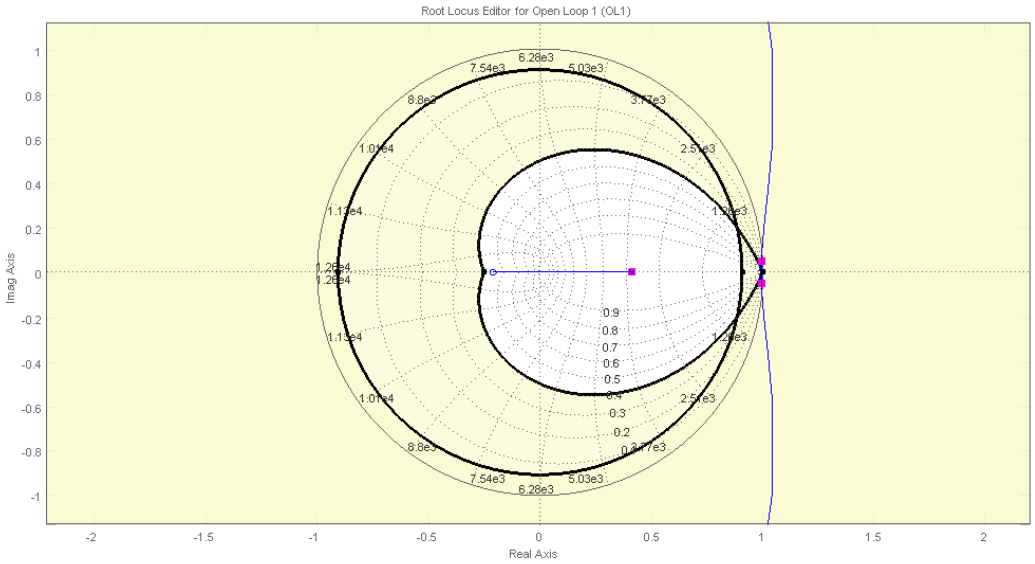

Figure 5.10. Root locus for GMS elevation axis (2/3)

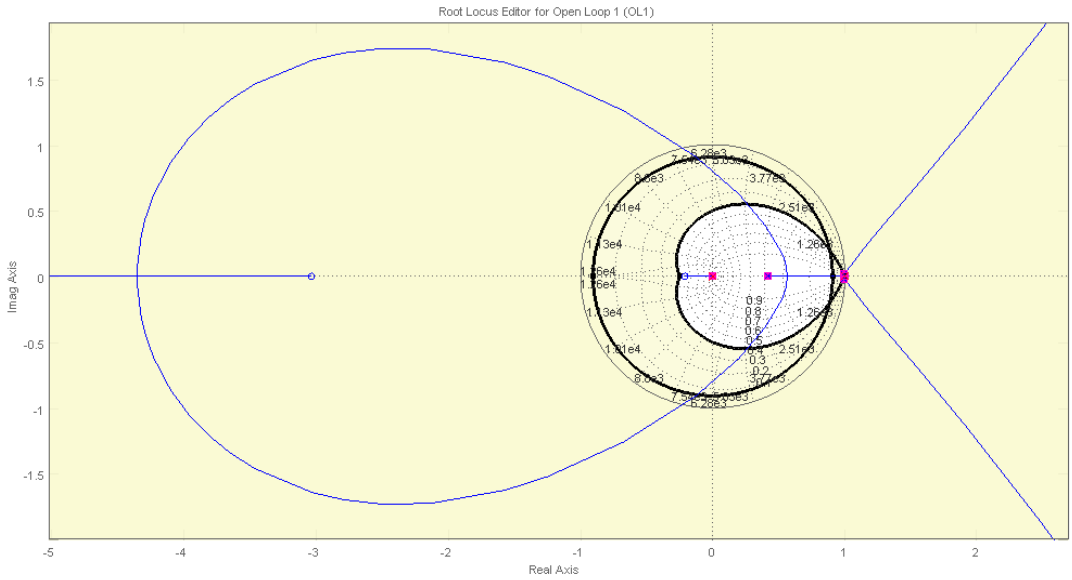

Figure 5.11. Root Locus for GMS elevation axis (3/3)

From Figure 5.11 it can be seen that the two complex conjugate open loop poles make the system unstable. To make the system stable and to be able to make the two branches, which go to infinity, to pass through the region where the closed loop poles should be placed, one of the zeros of the controller should be placed near to the open loop pole at z=1 while the other zero should be placed between z=0 and z=1. This way the open loop pole at z=1 end up at the zero placed near to it and the complex conjugate open loop poles would move towards the inside of the unit circle and one of them would end up at the open zero of the controller, which is placed between z=0 and z=1.

When one of the zeros of the controller is placed at z=0.9970 and the other zero is placed at z=0.9608, the complex conjugate poles move towards inside of the unit circle as expected. Lastly, the two dominant closed loop poles should be placed inside the predefined area of success by adjusting the gain.

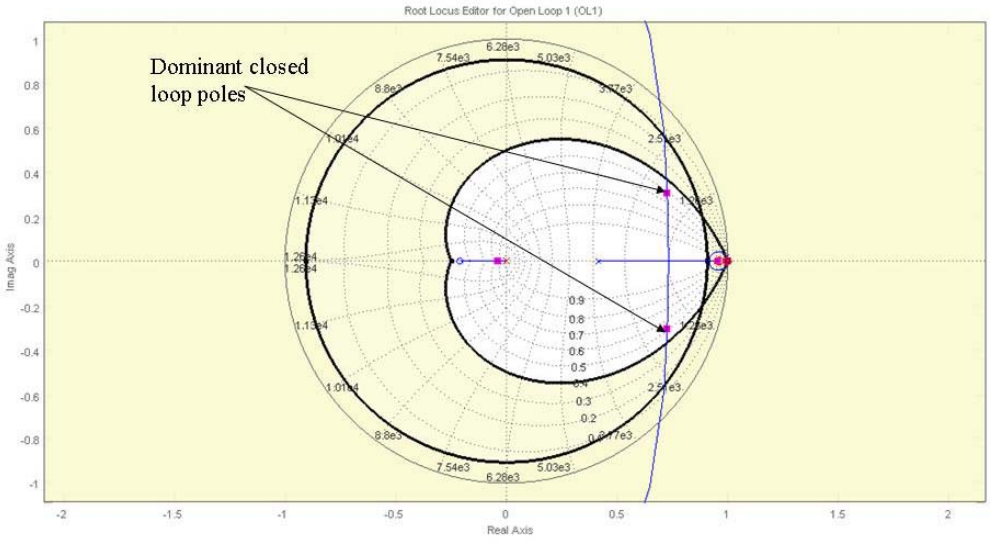

Figure 5.12. Location of the dominant closed loop poles for GMS elevation axis

With the gain adjusted to 167.02 and the zeros of the PID controller placed at z=0.9970 and z=0.9608, the closed loop poles are located inside the predefined area as shown in Figure 5.12. The designed PID controller becomes;

$$G_c(z) = 167.02\left(\frac{(z-0.9970)(z-0.9608)}{(z-1)z}\right) \tag{5.17}$$

By equating (5.17) to (5.15), the parameters for the PID controller of the GMS elevation axis are obtained as, $K_p = 7.01$, $K_i = 5.03$ and $K_d = 159.99$. However, since the PID parameters of the digital controller of the motion control card can only be set as integers, the values are rounded up to the nearest integers. Hence the PID parameters for the azimuth axis are set as, $K_p = 7$, $K_i = 5$ and $K_d = 160$. With the designed controller, the unit step response of the closed loop system is as shown in Figure 5.13. As can be seen from the plot, maximum percent overshoot, settling time and steady state error specifications are satisfied.

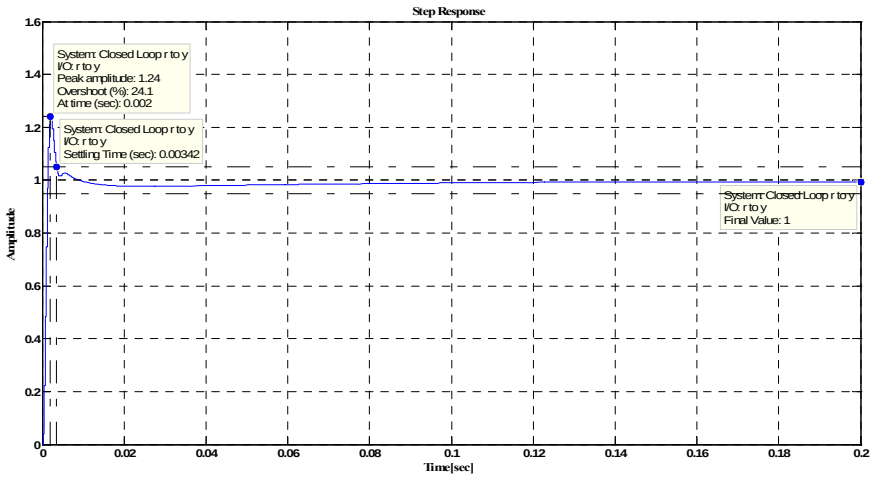

Figure 5.13. Unit step response of the GMS elevation axis with the designed controller

CHAPTER 6

CONTROLLER PERFORMANCE EVALUATION

The performance of the digital PID controllers optimized for the GMS axes, both on a stationary platform and on FMS are to be observed and evaluated. Two methods used for this purpose are running simulation models and performing tests with the real system.

The performances on a stationary base are observed both by simulation runs and real system tests. However, since the 5 axis FMS that would be used for the HIL simulation system is not available yet, performances of the controllers on moving FMS axes for both GMS axes are observed by simulation runs only. A fixed step size of 250 microseconds is used for the simulations while the solver is selected as Dormand-Prince.

6.1 Performances of controllers on a stationary platform

The case, where the GMS is mounted on a stationary plate is considered in this section. The simulation model is run for different step inputs with magnitudes of 0.005, 0.010, 0.025 and 0.050 degrees for both GMS axes while the FMS axes are motionless. The same step inputs are also sent to the motion control card. The real system responses and simulation outputs plotted on the same graphs and also the error graphs showing the errors between the real system and simulation outputs are shown in Figure 6.1 through Figure 6.16.

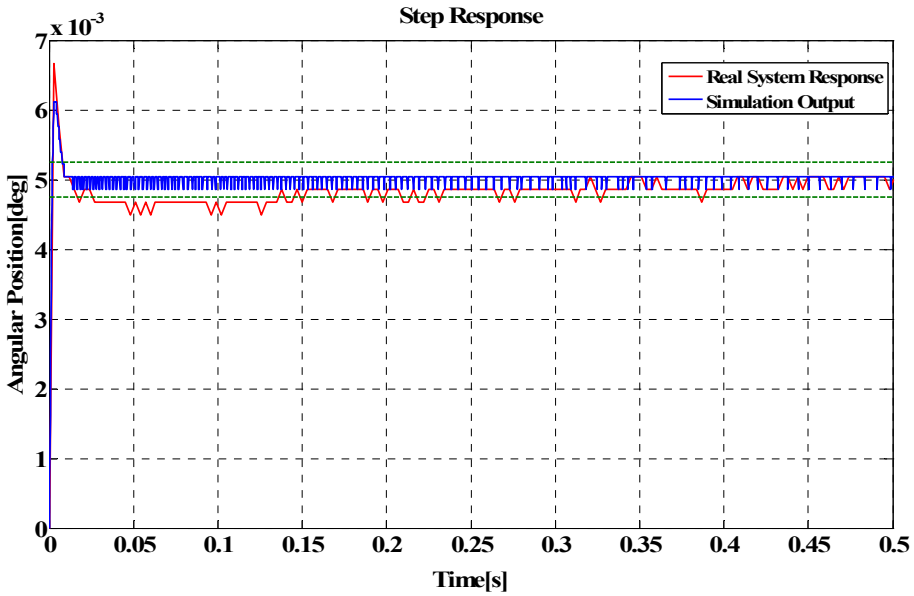

Figure 6.1. GMS azimuth axis response to 0.005 degree step input

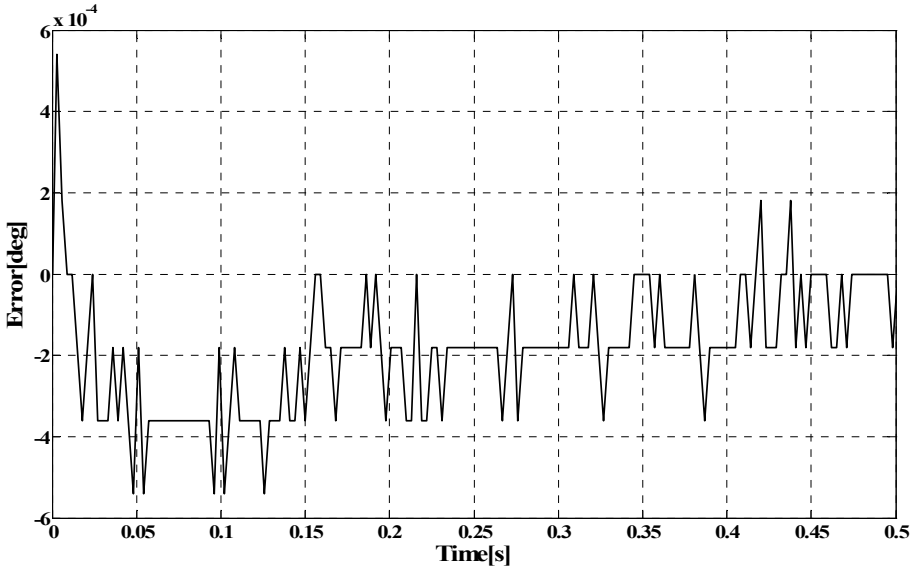

Figure 6.2. Error between real system and simulation outputs

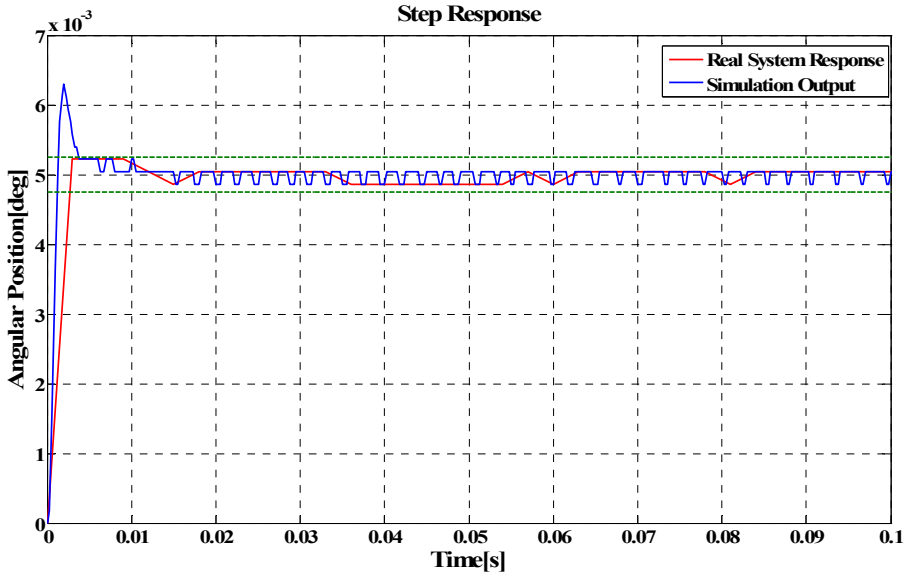

Figure 6.3. GMS elevation axis response to 0.005 degree step input

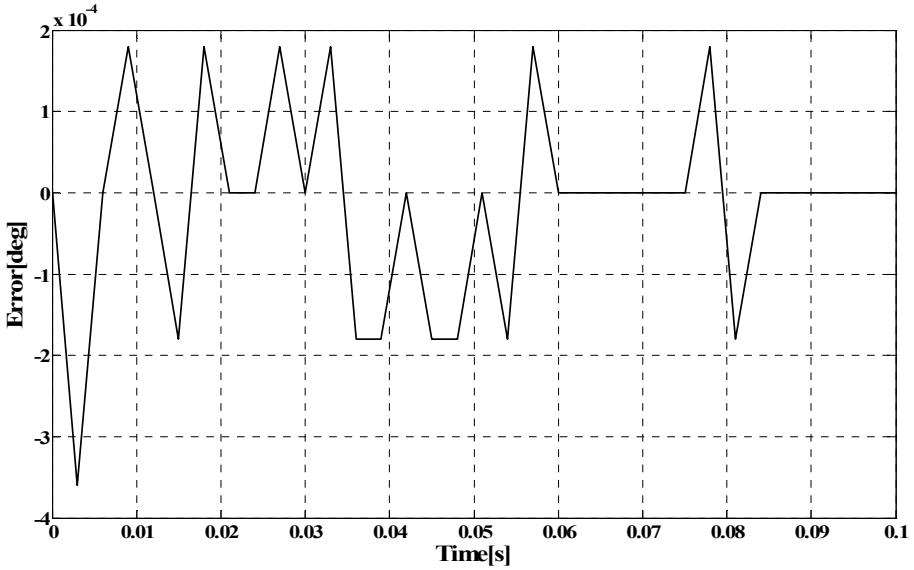

Figure 6.4. Error between real system and simulation outputs

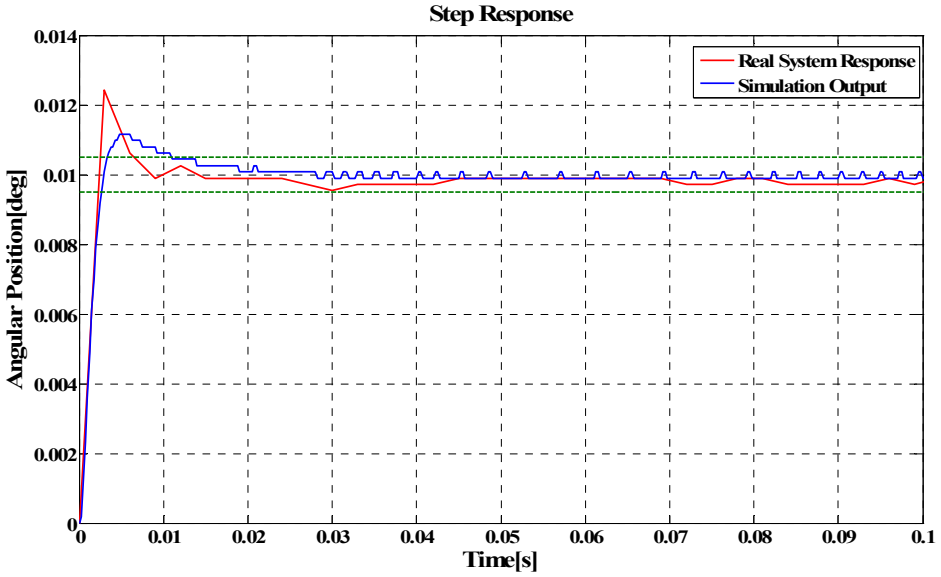

Figure 6.5. GMS azimuth axis response to 0.010 degree step input

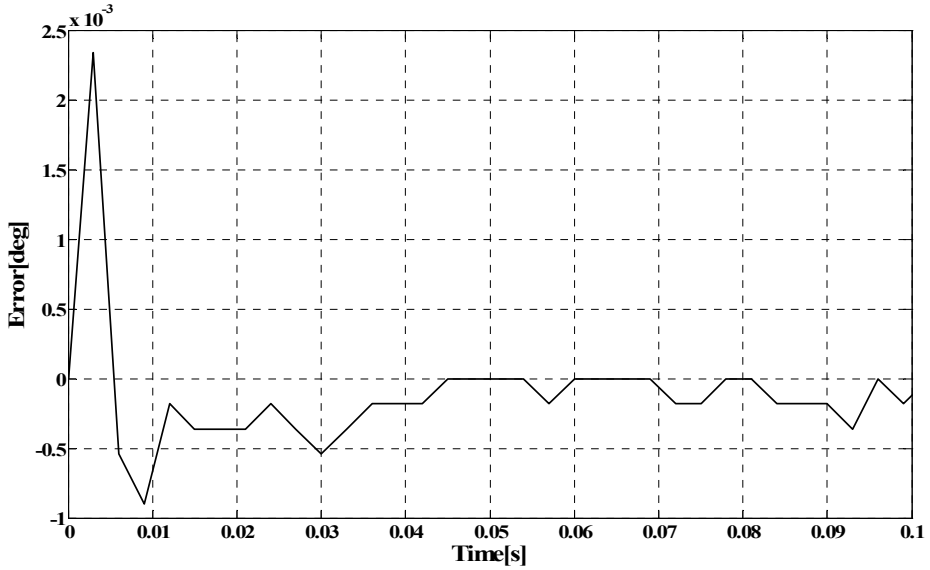

Figure 6.6. Error between real system and simulation outputs

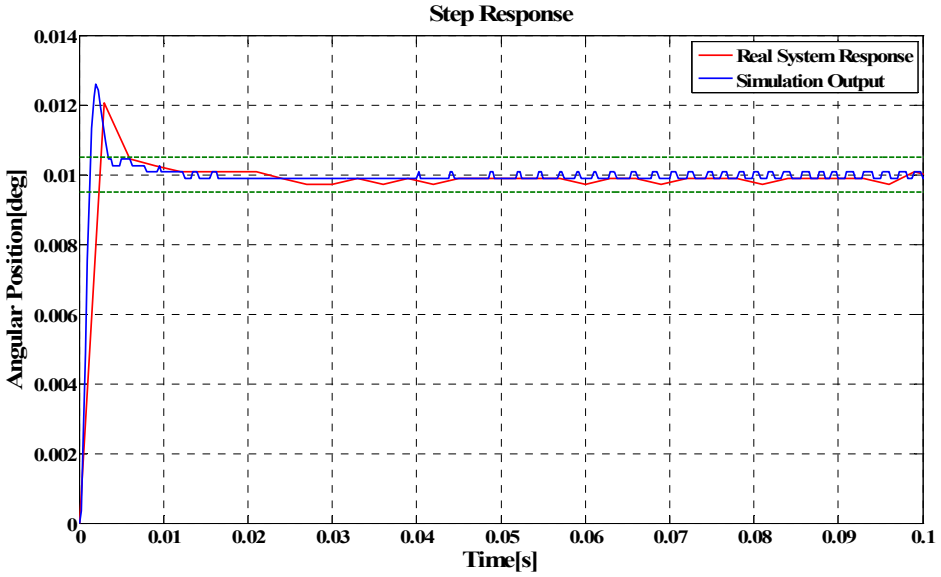

Figure 6.7. GMS elevation axis response to 0.010 degree step input

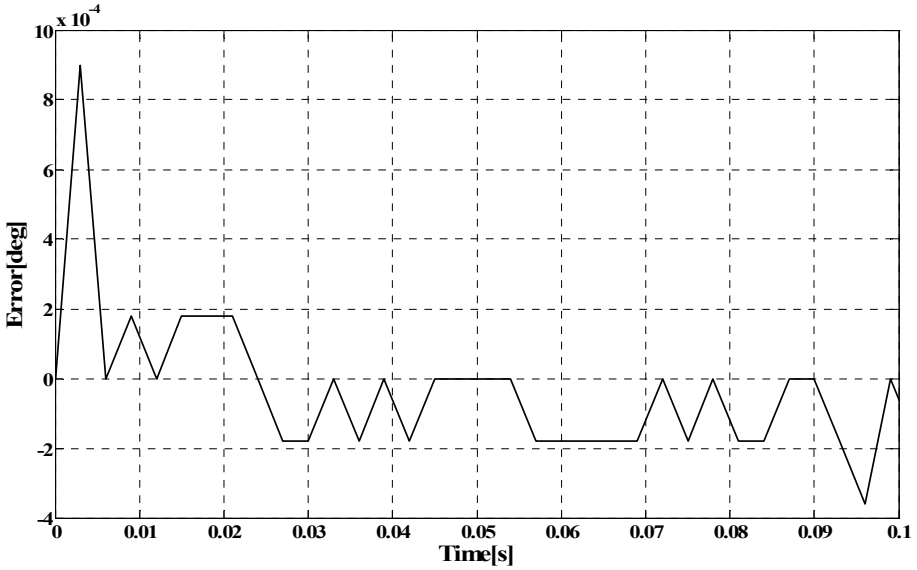

Figure 6.8. Error between real system and simulation outputs

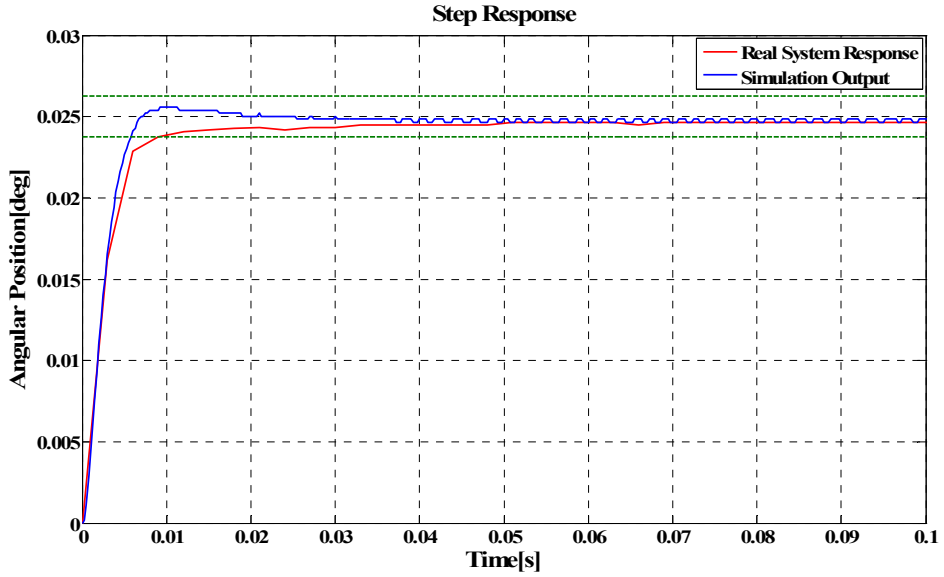

Figure 6.9. GMS azimuth axis response to 0.025 degree step input

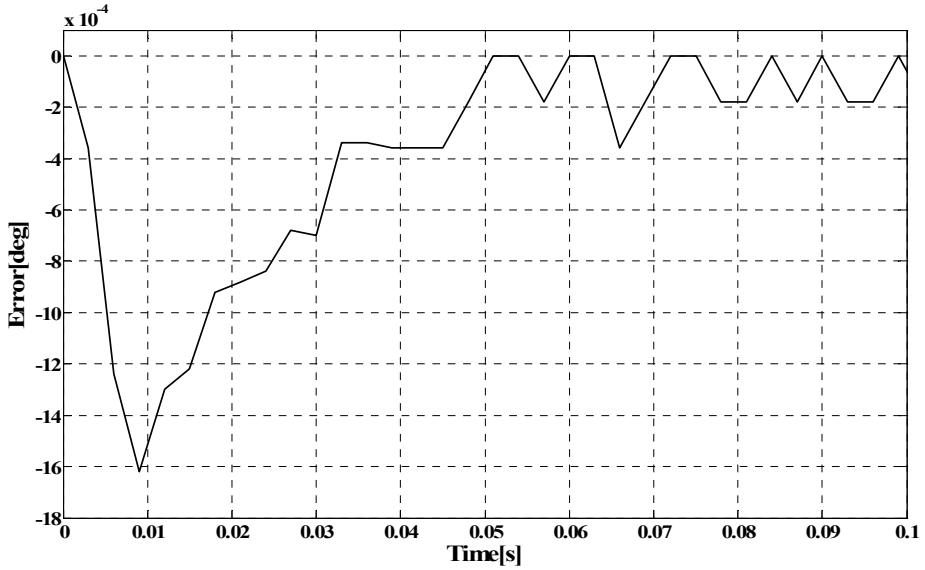

Figure 6.10. Error between real system and simulation outputs

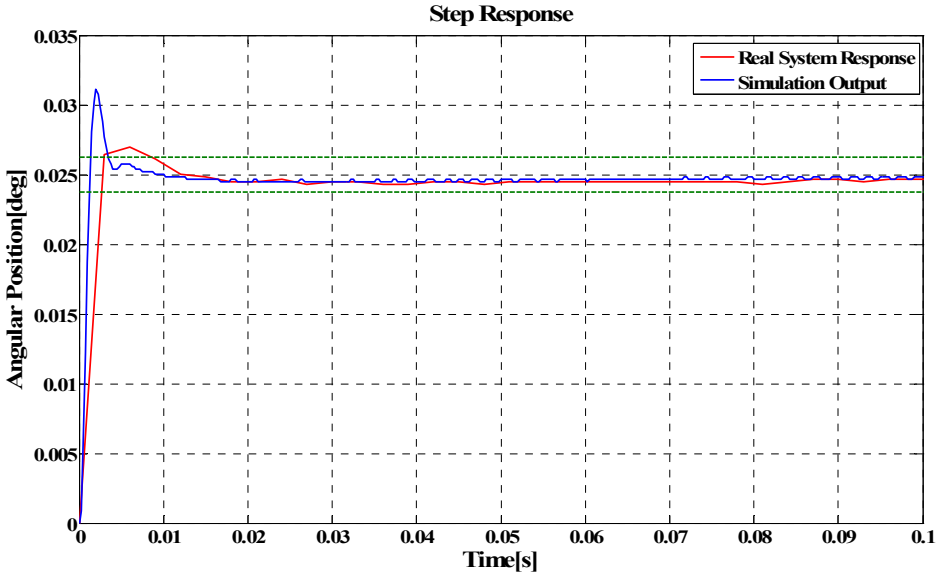

Figure 6.11. GMS elevation axis response to 0.025 degree step input

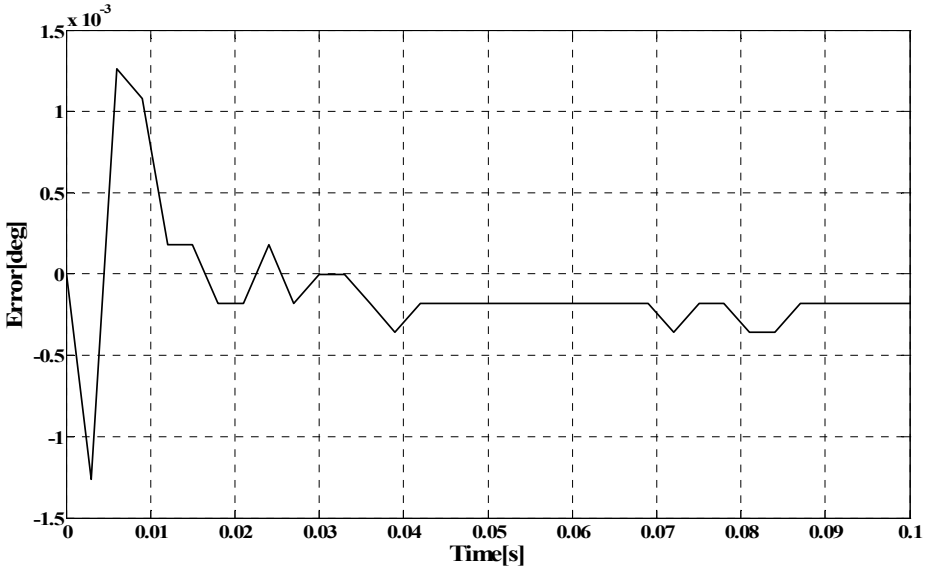

Figure 6.12. Error between real system and simulation outputs

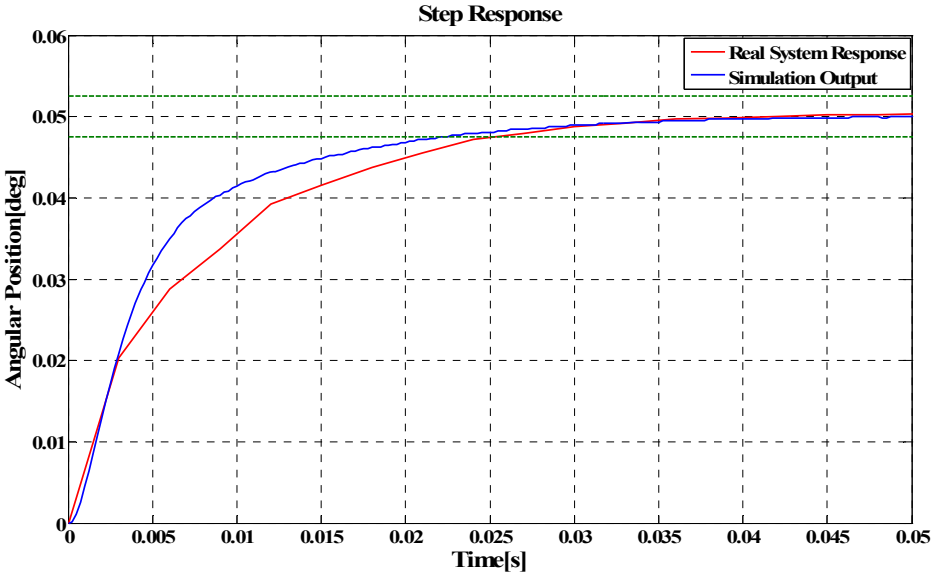

Figure 6.13. GMS azimuth axis response to 0.050 degree step input

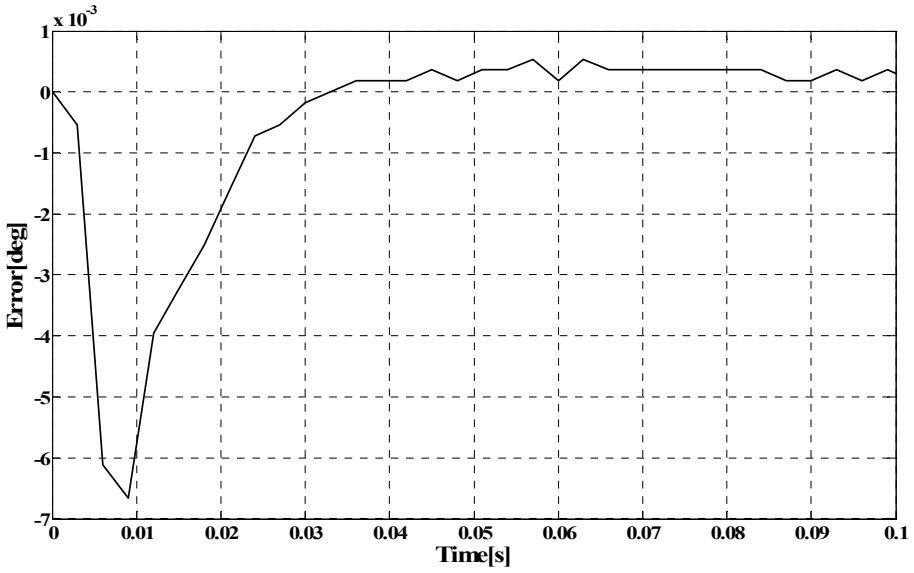

Figure 6.14. Error between real system and simulation outputs

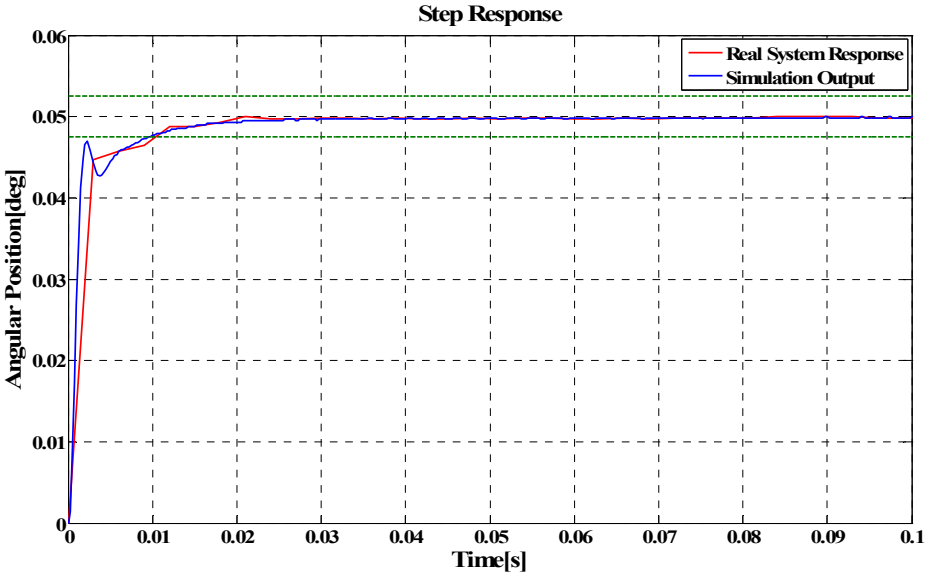

Figure 6.15. GMS elevation axis response to 0.050 degree step input

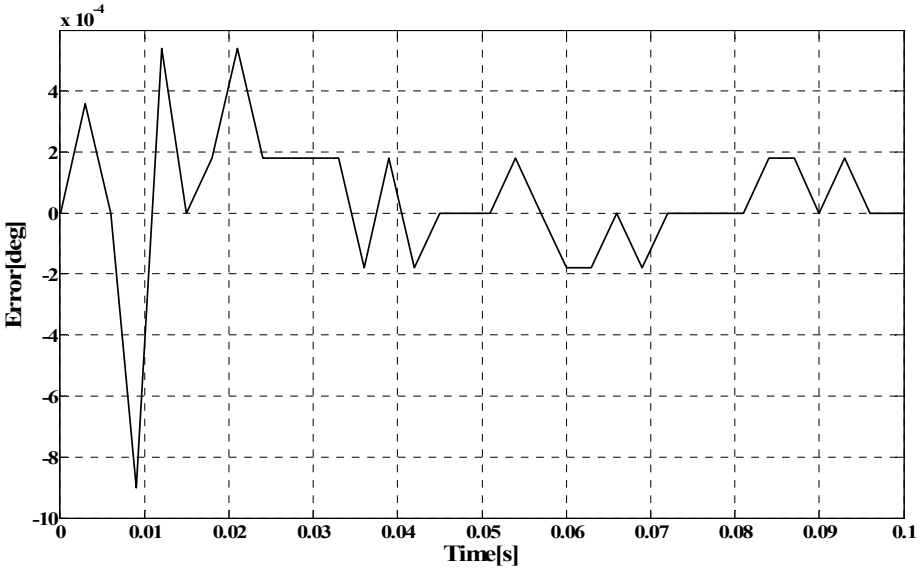

Figure 6.16. Error between real system and simulation outputs

As can be seen from the figures, the error between real system responses and simulation outputs are generally in the order of 10^{-4}. Both real axes responses and simulation outputs to 0.005 degree step input are as expected and meet time domain specs. However, when the simulation output for azimuth axis to 0.010 degree step input is examined, it is seen that the maximum overshoot value is decreased, and the settling time is increased. The reason is the saturation of the amplifier. The modulated supply voltage is 24 Volts. For 100% duty cycle the PWM blocks output a maximum voltage of 24 Volts. While the submodel outputs are examined, it is noticed that the voltage output of PWM block for the azimuth axis is limited by the voltage limiter block to 24 Volts. This behavior is not clear for the real system response. On the other hand, both the real system response and simulation output of the elevation axis to 0.010 degree step input is as expected and satisfies the requirements.

When the 0.025 degrees step input response plot for the azimuth axis is examined, it is seen that there is no overshoot. Maximum overshoot value is expected to decrease further since the amplifier which saturates for 0.010 degree step input is expected to saturate again for 0.025 degree step input but the decrease in the overshoot value is more drastic than expected. It is suspected that the other limits in the control system, which are the PID output limit and integral term limit, may be reached. When the PID output is observed, it is seen that the PID output reaches its positive limit indeed. The elevation axis still does not reach either the saturation limit for amplifier or the PID output limit for 0.025 degree step input so that it behaves as expected. Real system responses and simulation outputs for both axes to 0.025 degree step input satisfy the requirements.

Although step inputs greater than 0.025 degrees are not expected as axis position commands in the system, responses to a 0.050 degree step input are also observed in order to check the performance limits of the controllers. For 0.050 degree step input, the PID output of the elevation axis also reaches its positive limit. Hence, the time

domain characteristics of the response are also changed for elevation axis response but it still satisfies the requirements. On the other hand, the settling time for the azimuth axis increases to 22 ms for 0.050 degree step input, which does not satisfy the requirement.

6.2 Controllers' performances in dynamic environment

Some realistic scenarios were simulated utilizing a simulation model of the whole HIL system which had been constructed at Roketsan Inc. Three sets of position commands for the FMS and GMS axes are extracted from the results of these scenario runs and used as inputs to the developed MATLAB/Simulink model in order to observe the performance of the controllers under the effect of disturbance torques imposed on GMS axes.

The FMS position commands from the scenario runs are not directly fed as kinematic inputs to FMS axes. The transfer functions for the FMS axes are available so the position commands are fed to these FMS axes transfer functions first and then supplied as kinematic inputs to the FMS axes in the model. This way the fidelity of the simulation to real life increases.

For each scenario, simulation is run for both cases where the FMS is moving and not moving. The position commands for the GMS axes, the GMS axes position responses while FMS is also moving and GMS axes position responses while the FMS axes are stationary are plotted on the same graph for each scenario. The error between the commanded trajectories and axes responses are given by graphs. On the other hand, to observe the effect of FMS motion on system performance, the differences between the outputs of the simulations with and without FMS motion are presented via graphs.

First scenario:

The position commands for FMS axes are shown in Figure 6.17 and Figure 6.18 while the position commands for GMS axes, position responses for stationary and moving FMS cases, and the error graphs are given in Figure 6.19 through Figure 6.24.

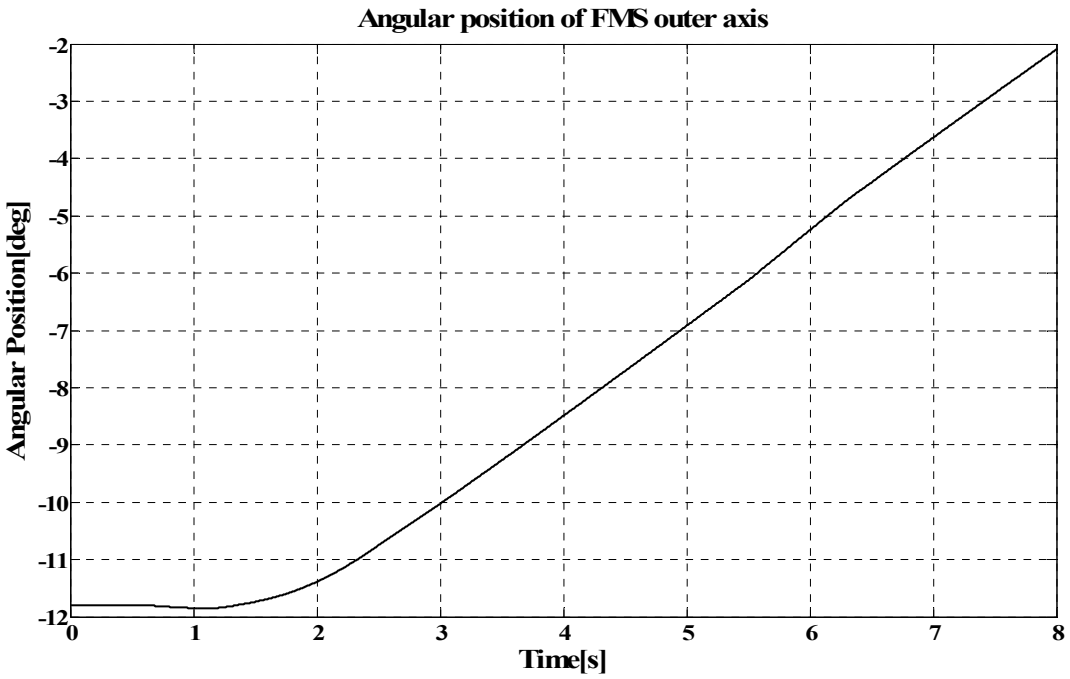

Figure 6.17. Position command for the outer FMS axis

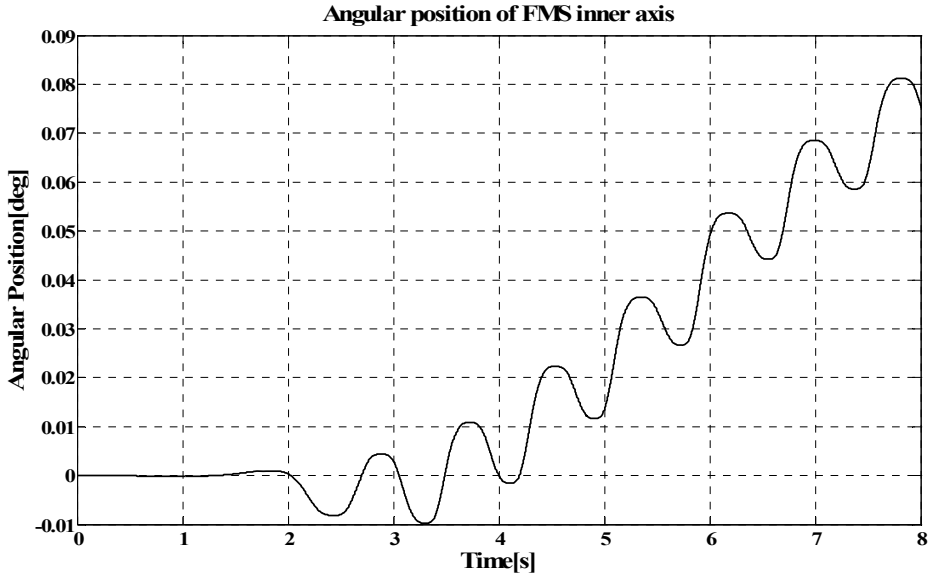

Figure 6.18. Position command for the inner FMS axis

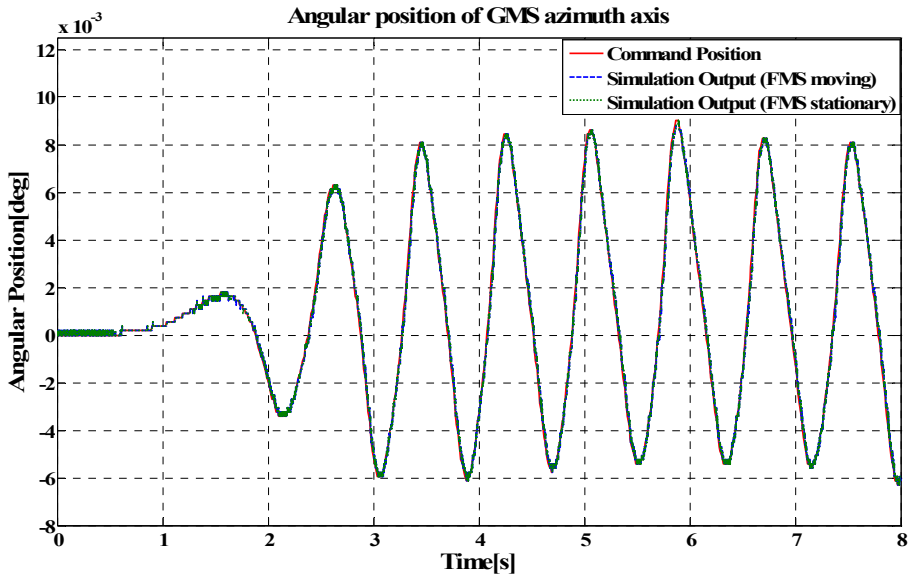

Figure 6.19. Position command and response of GMS azimuth axis

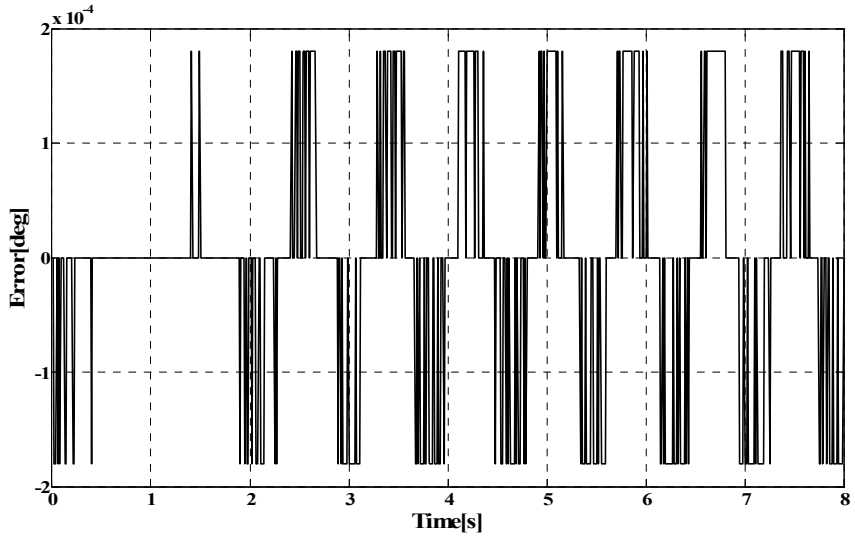

Figure 6.20. Error between commanded position and simulation output with FMS motion for azimuth axis

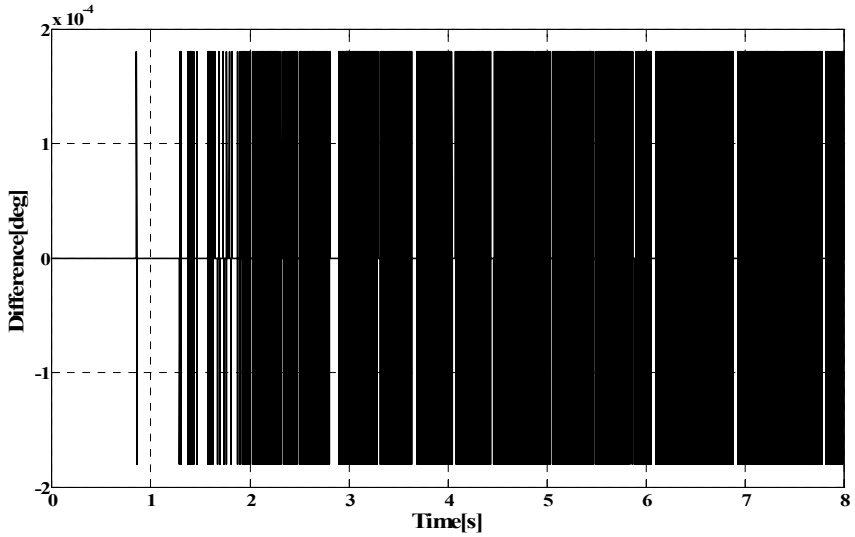

Figure 6.21. Difference between the simulation outputs with and without FMS motion for azimuth axis

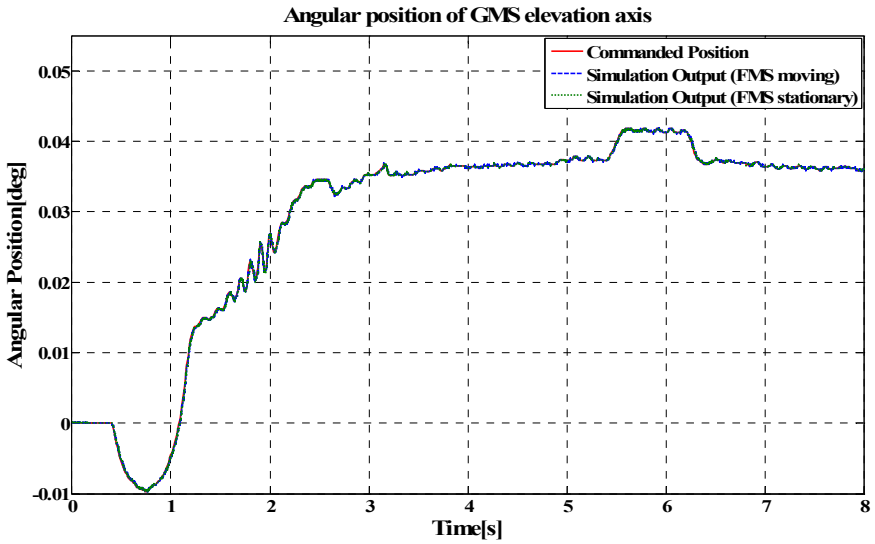

Figure 6.22. Position command and response of GMS elevation axis

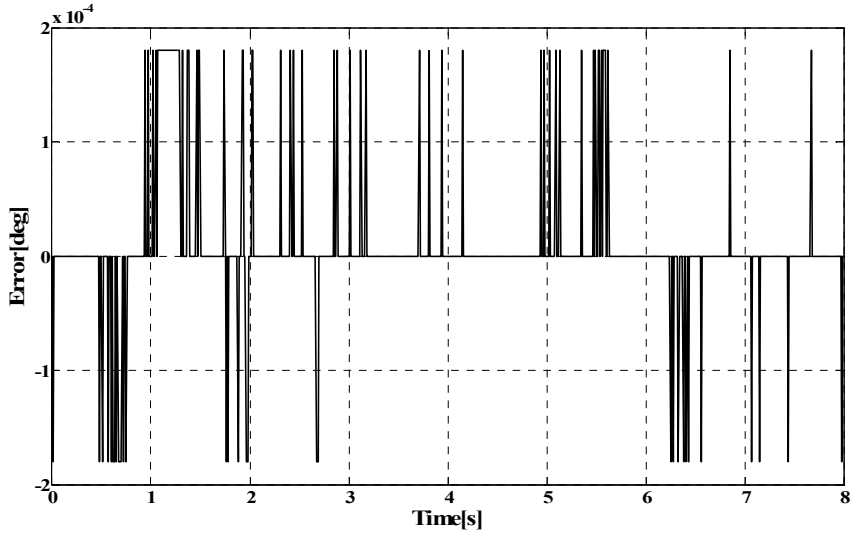

Figure 6.23. Error between commanded position and simulation output with FMS motion for elevation axis

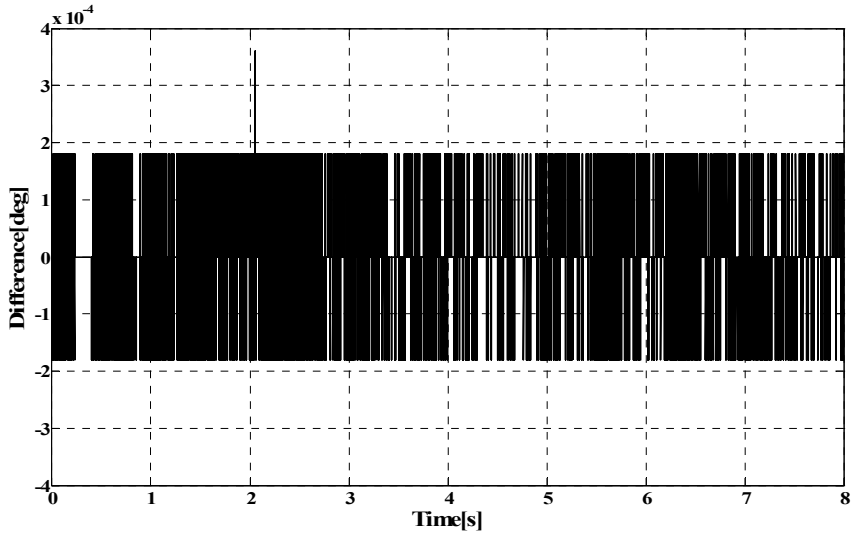

Figure 6.24. Difference between the simulation outputs with and without FMS motion for elevation axis

Results show that both GMS axes track given position commands accurately. The maximum error between the commanded trajectory and the axis trajectory is 0.00018 degrees, which is the position resolution value, for both GMS axes. On the other hand, the maximum difference between the outputs of the simulations with and without FMS motion is 0.00018 degrees for the azimuth axis and 0.00036 degrees for the elevation axis. Such small differences show that the effect of FMS axes movements on GMS axes are effectively rejected by the control systems of both axes for this scenario.

Second scenario:

The position commands for FMS axes are shown in Figure 6.25 and Figure 6.26 while the position commands for GMS axes, position responses for stationary and moving FMS cases, and error graphs are given in Figure 6.27 through Figure 6.32.

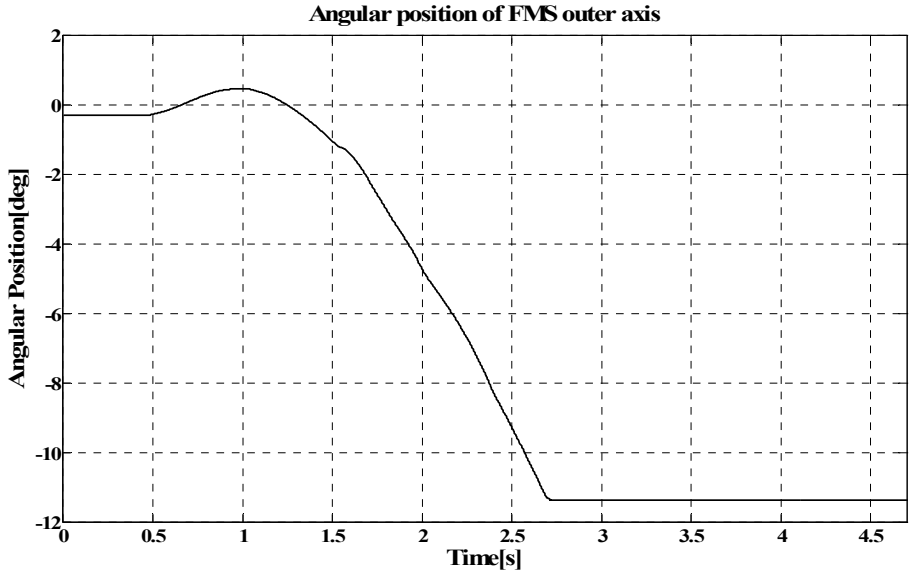

Figure 6.25. Position command for the outer FMS axis

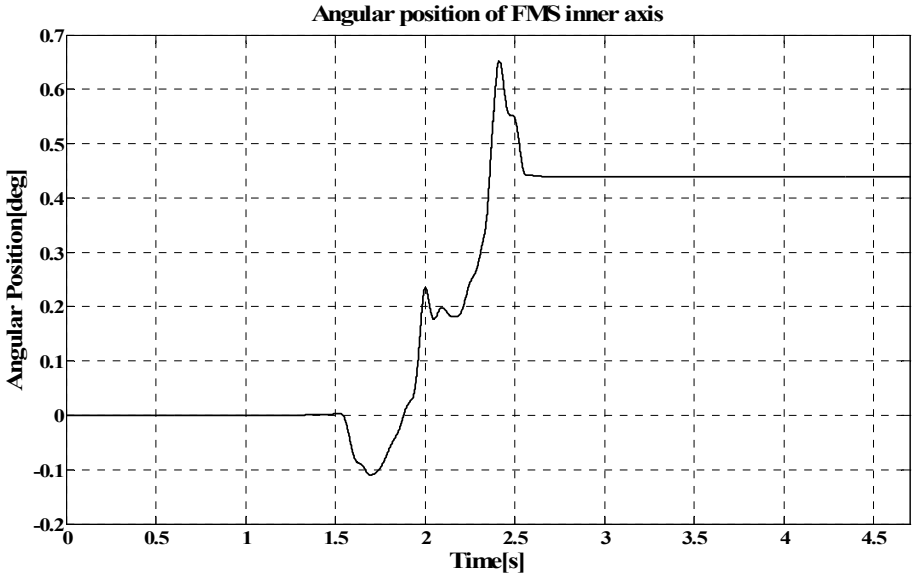

Figure 6.26. Position command for the inner FMS axis

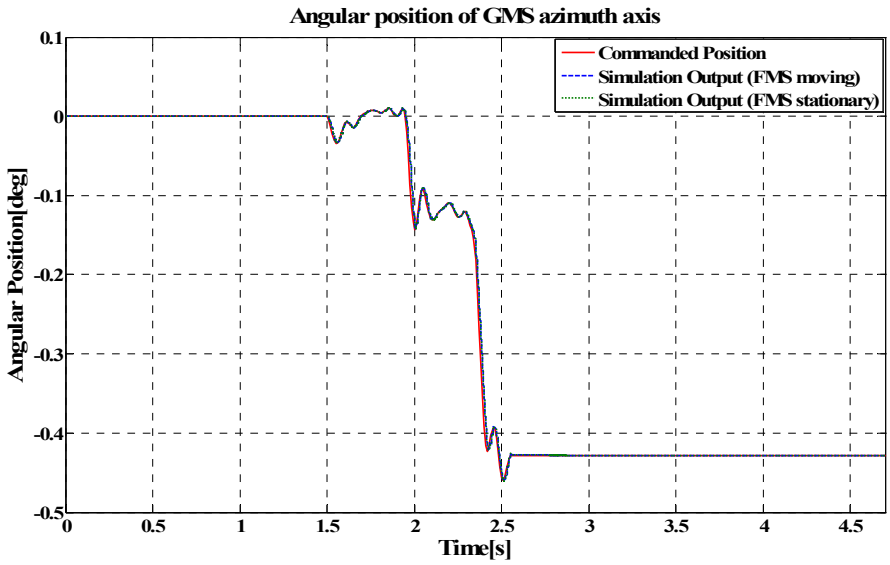

Figure 6.27. Position command and response of GMS azimuth axis

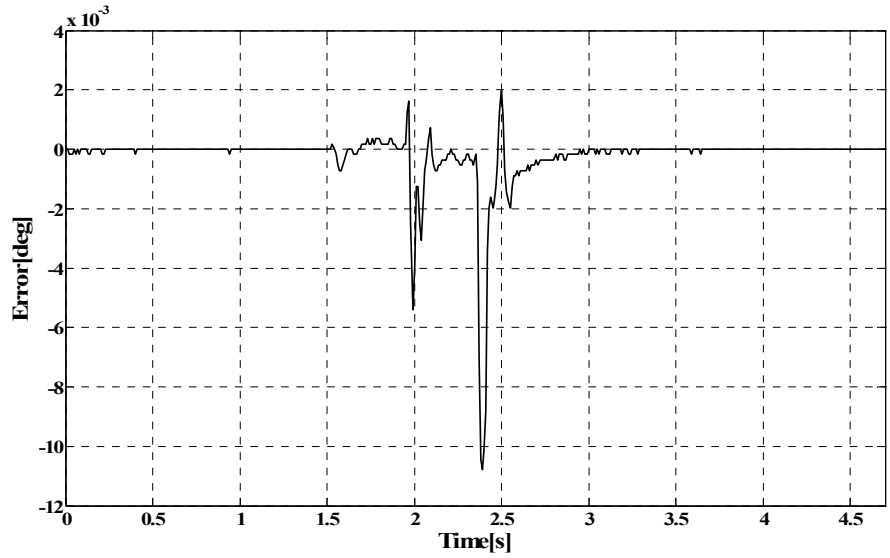

Figure 6.28. Error between commanded position and simulation output with FMS motion for azimuth axis

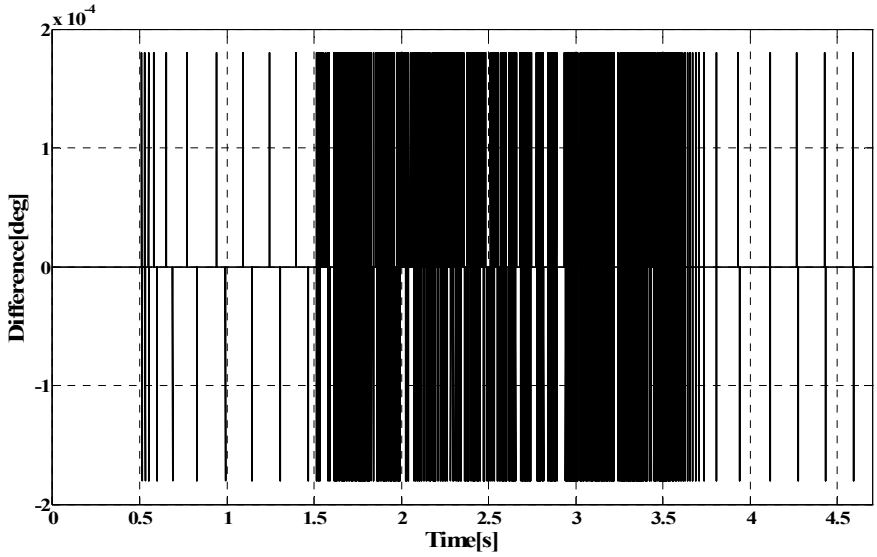

Figure 6.29. Difference between the simulation outputs with and without FMS motion for azimuth axis

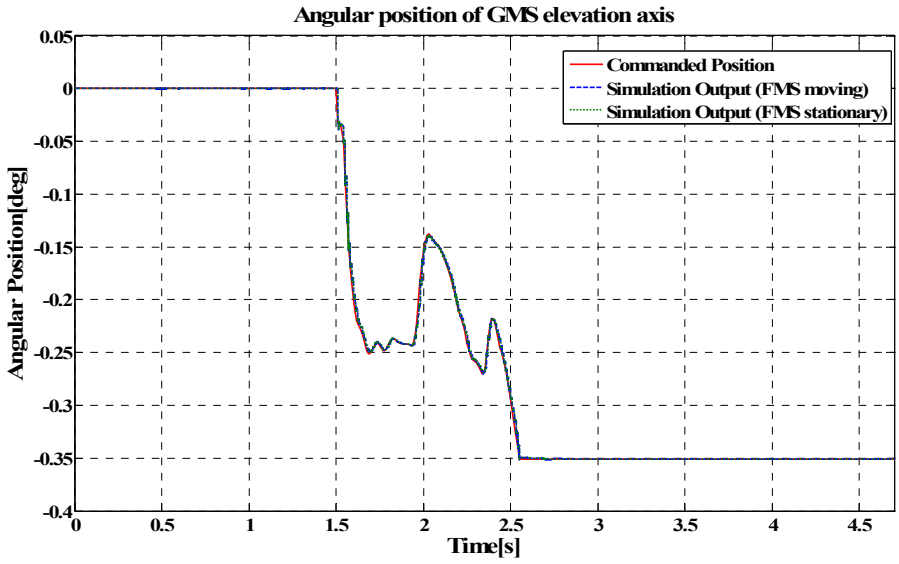

Figure 6.30. Position command and response of GMS elevation axis

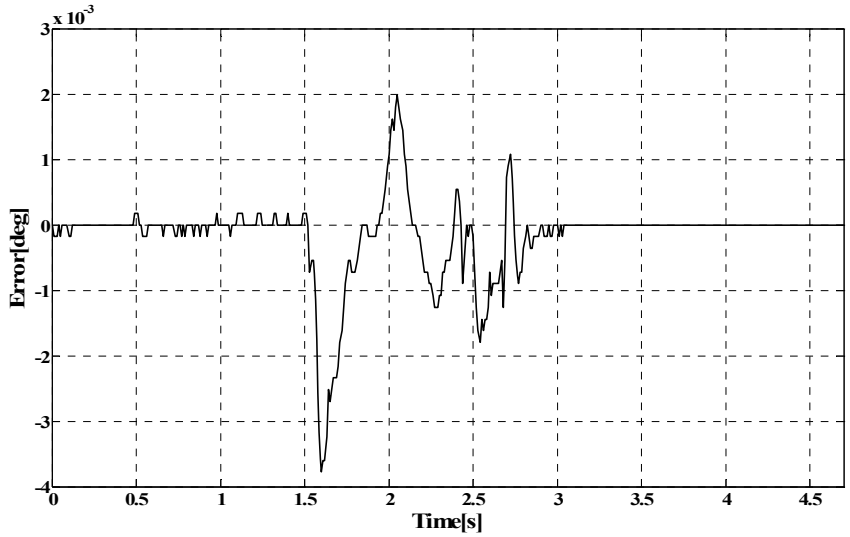

Figure 6.31. Error between commanded position and simulation output with FMS motion for elevation axis

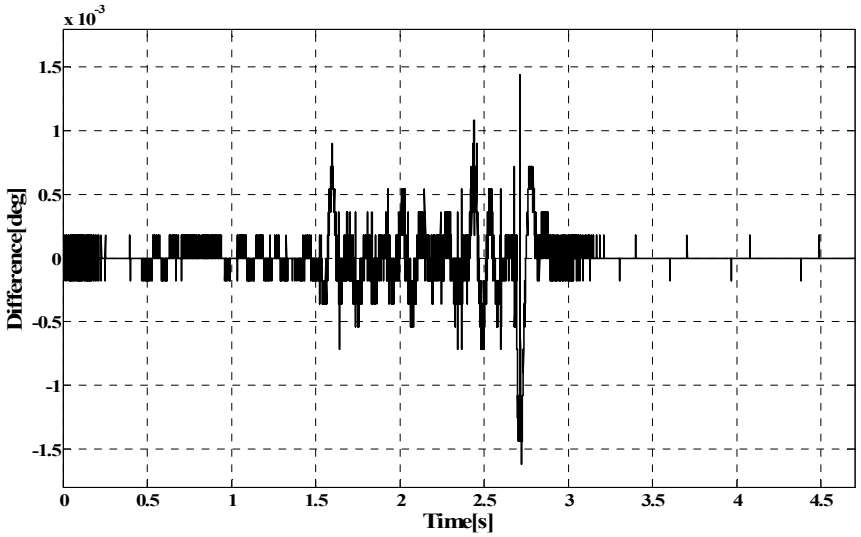

Figure 6.32. Difference between the simulation outputs with and without FMS motion for elevation axis

From the graphs, it is seen that although both GMS axes track given position commands accurately for the major part of the scenario, between 2.37 and 2.42 seconds the error between the commanded and axis trajectories for azimuth axis does not satisfy the position accuracy requirement. When the commanded trajectory for the azimuth axis is examined, it can be seen that at 2.34 seconds, the trajectory is at -0.15 degrees while at 2.42 seconds the trajectory is at -0.43 degrees. This corresponds to an average movement of 0.035 degrees per 10 milliseconds, which is the command update rate. From the controller performance results given in previous section, it was observed that amplifier saturation and limiting of PID output occurs for azimuth axis when relatively large position commands are applied. This causes position commands greater than 0.025 degrees not to settle in 10 milliseconds. Relatively large position commands sent to the azimuth axis between 2.34 and 2.42 seconds result in an accumulated error because of amplifier saturation and PID output limitation, and this error exceeds the accuracy limit after some point. The maximum error between the commanded trajectory and the axis trajectory is 0.0037 degrees for GMS elevation axis which is smaller than the required position accuracy. On the other hand, the maximum difference between the outputs of the simulations with and without FMS motion is 0.00018 degrees for the azimuth axis and 0.00144 degrees for the elevation axis. Since these differences are smaller than the position accuracy required, it can be said that the effect of FMS axes movements on GMS axes are effectively rejected by the control systems of both axes also for this scenario.

Third scenario:

The position commands for FMS axes are shown in Figure 6.33 and Figure 6.34 while the position commands for GMS axes, position responses for stationary and moving FMS cases, and the error graphs are given in Figure 6.35 through Figure 6.40.

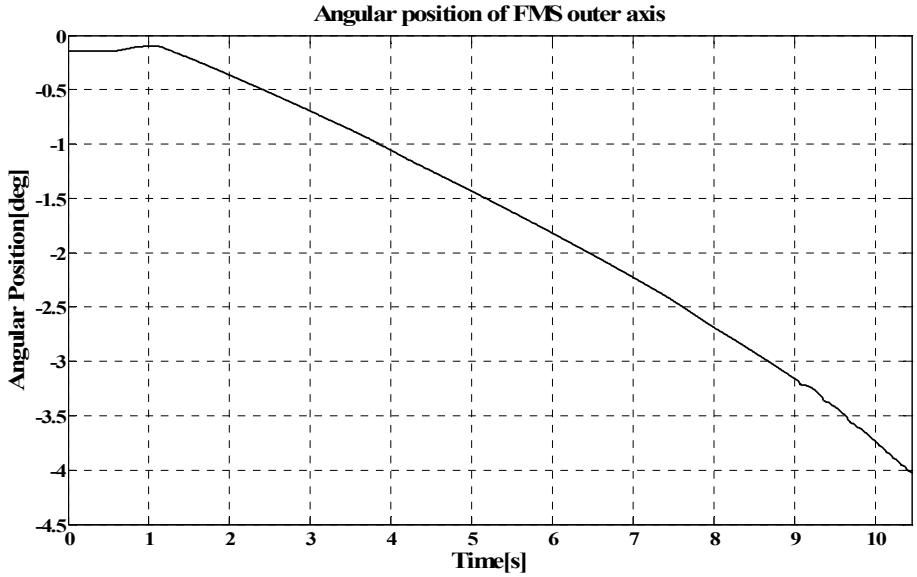

Figure 6.33. Position command for the outer FMS axis

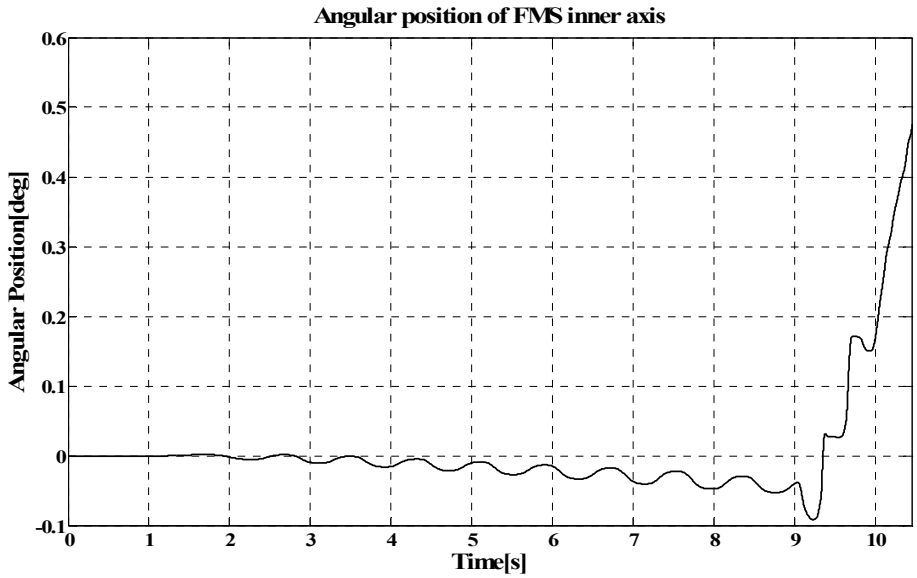

Figure 6.34. Position command for the inner FMS axis

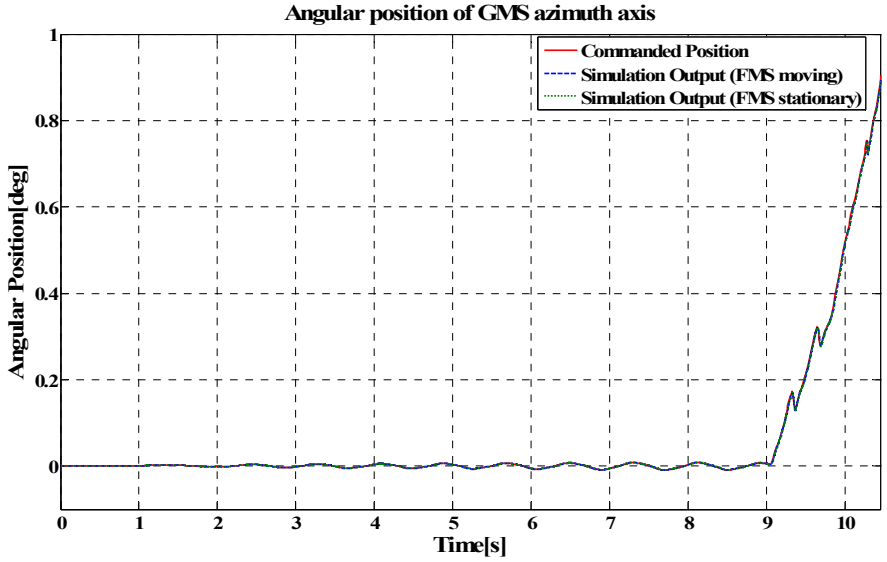

Figure 6.35. Position command and response of GMS azimuth axis

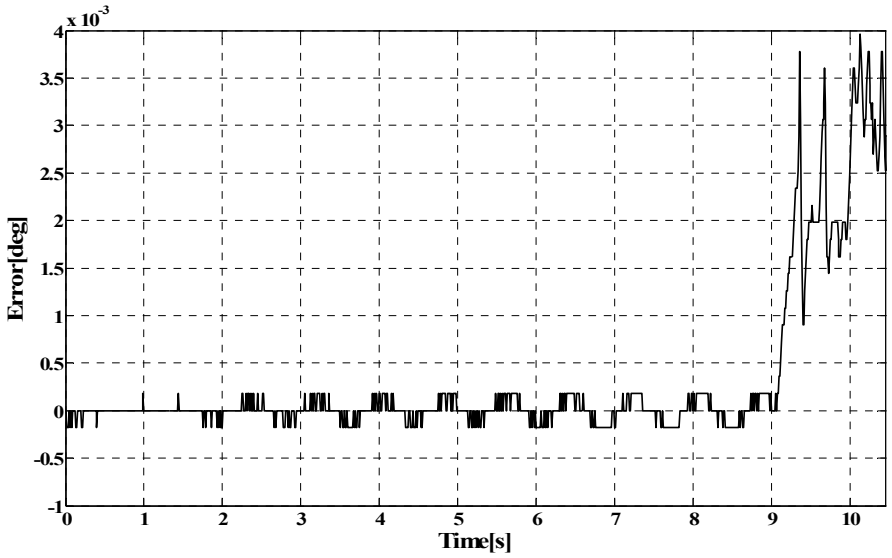

Figure 6.36. Error between commanded position and simulation output with FMS motion for azimuth axis

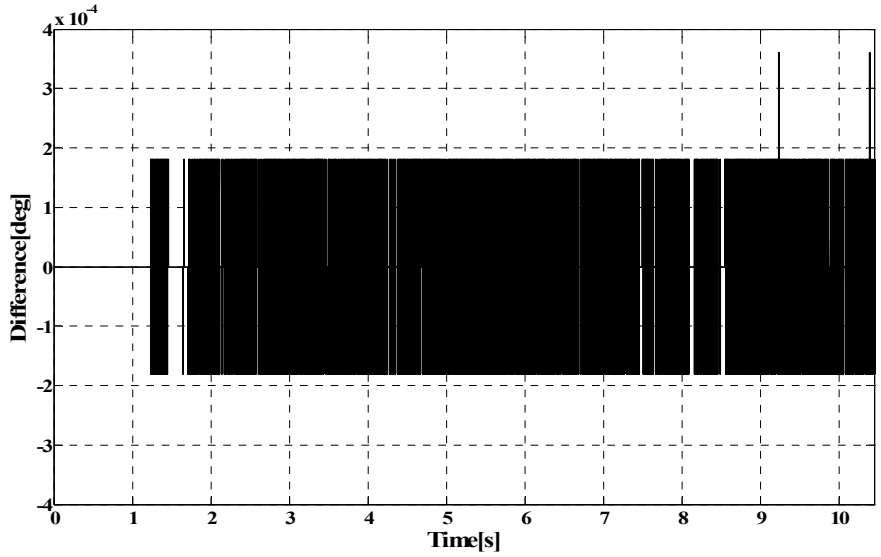

Figure 6.37. Difference between the simulation outputs with and without FMS motion for azimuth axis

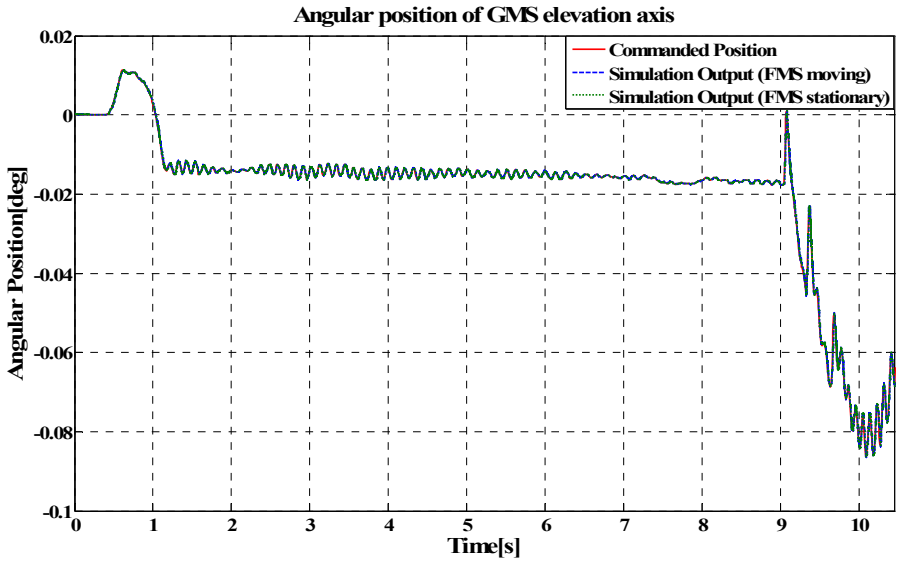

Figure 6.38. Position command and response of GMS elevation axis

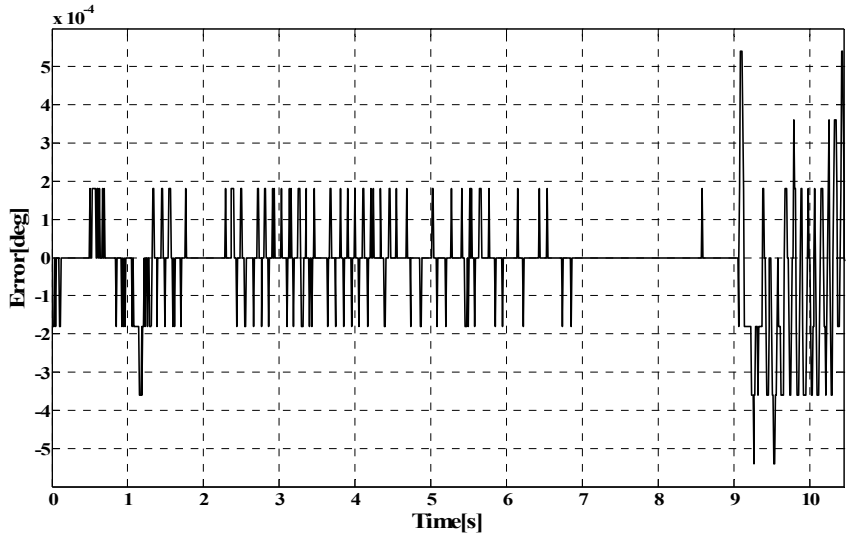

Figure 6.39. Error between commanded position and simulation output with FMS motion for elevation axis

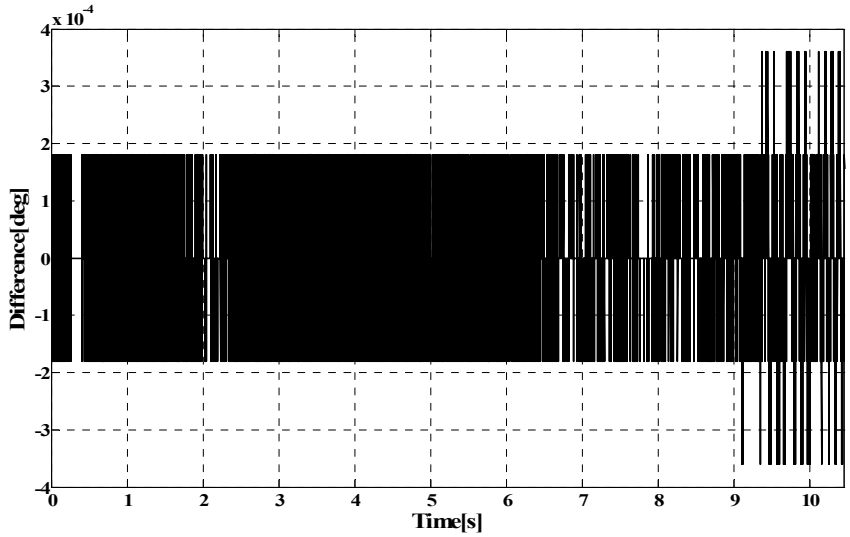

Figure 6.40. Difference between the simulation outputs with and without FMS motion for elevation axis

Results show that both GMS axes track given position commands accurately. The maximum error between the commanded trajectory and the axis trajectory is 0.004 degrees for azimuth axis and 0.00054 degrees for elevation axis. On the other hand, the maximum difference between the outputs of the simulations with and without FMS motion is 0.00036 degrees for both GMS axes. Such small differences show that for this scenario, the disturbance torques imposed on GMS axes because of FMS axes movements are effectively rejected by the control systems of both axes.

CHAPTER 7

SUMMARY, CONCLUSION AND RECOMMENDATIONS

7.1 Summary

The focus of this thesis work is on modeling, parameter identification, real-time control system design, and implementation of a two axis gimbaled mirror system.

Roketsan Inc., is developing a HIL system, which will be used as a tool for optimizing flare dispensing programs to effectively counter various kinds of IR guided missiles. An IRSGS is also being developed by Roketsan Inc. to simulate the radiometric properties and motion of the target. The aforementioned two axis gimbaled mirror system is the component that will be used for moving the generated IR spot in azimuth and elevation axes to simulate the motion of the target w.r.t. the seeker of the missile.

The mathematical models are obtained and realized in MATLAB/Simulink environment for each component of the real-time control system, mainly, the motion control card, amplifiers and the GMS axes.

The motion control card can generate trajectories for the axes considering the velocity and acceleration values set for the axes and the position commands sent to the card. These command generators are modeled. The discrete time transfer

function of the digital PID controllers that the motion control card has for each axis is first assumed and then verified by performing tests on the motion control card.

The amplifiers used are current (torque) mode amplifiers. They have PI current regulators in order to make a current value, which is proportional to the voltage value that is applied to their input pins, pass through the motor windings. Of all the components of the amplifiers, these PI current regulators have the most important impact on the dynamics of the system. Modeling the amplifiers as proportionality constants between the input voltage and the output current would decrease the fidelity of the developed model to the real system much. Consequently, PI current regulator models are also included.

The electrical dynamics of actuators of the GMS axes are modeled with the resistance and inductance values of the motor coils. The power flow from the electrical system to the mechanical system is modeled through the torque constants while its reverse flow is modeled by the back emf effect.

Both viscous friction and Coulomb friction at the bearings are included in the model. It is observed that the dynamics of the cables coming from the connectors and attached to the axes has significant effect on the dynamics of the axes. The cables cause position depending torques to act on the axes. The cables are represented as if they are springs.

Disturbance torques that would be developed on the GMS axes because of the motion of the FMS axes are modeled using two different methods. In the first method, Euler equations for rigid body motion for both GMS axes are solved. For this method, the kinematics analysis of the five link FMS-GMS mechanism is performed by using Denavit-Hartenberg convention. In the second method, the FMS-GMS mechanism is modeled in the MATLAB/Simulink/SimMechanics environment. The outputs of these two models are compared for verification.

Values of some parameters of the constructed MATLAB/Simulink model were unknown. The MATLAB/Simulink Parameter Estimation Tool is used to estimate these missing parameters. Various position commands are sent to the motion control card and the responses of the GMS axes to these commands are collected via the motion control card. These input-output data sets are entered to MATLAB/Simulink Parameter Estimation Tool and the parameters to be estimated are left as unknowns in the developed MATLAB/Simulink model. The parameter values obtained as the result of estimation are deemed logical. The real system outputs and the simulation outputs are compared to validate the estimation results. With the unknown values estimated, the model development is completed.

The optimization on the PID parameters of the digital controllers is achieved by the pole placement method. Since the control system has both continuous and discrete states, a transformation from s-domain to z-domain is performed to obtain the discrete time transfer functions of systems for both axes. The transfer functions obtained are imported to MATLAB/SISO Design Tool and the optimization on controller parameters are performed using the root locus of the system in MATLAB/SISO Design Tool. The nonlinear effects in the model such as friction torque, preloading of the cables, PID output limiter, PID controller integral term limiter and PWM modulator block output voltage limiter are neglected during controller optimization.

The performances of the controllers are observed both by running the simulation model and conducting tests with the real system. The real system outputs and the simulation results are compared.

The performances of the controllers under the effect of the disturbance torques, which are imposed on GMS axes by the motion of FMS axes, are simulated using the MATLAB/Simulink model.

7.2 Conclusions

The main objectives of this thesis work are to form a detailed high fidelity model for the whole real-time control system, and optimizing the PID parameters of the motion control card digital controllers in order to achieve the position control of the GMS axes considering the time domain requirements determined and implementing the optimized controllers.

A good understanding of the real-time control system components and a detailed analysis on their characteristics enable the construction of true and detailed models of each component.

Perfect agreement between the two different models developed for the torques acting on GMS axes because of the motion of FMS axes shows that the mathematical modeling of disturbance torques is highly reliable. Hence, using any one of these models in simulations to observe the performance of the system under the effect of disturbances would give a good idea about how the real system would behave on a FMS in motion.

Using MATLAB/Simulink Parameter Estimation Tool is a convenient selection for parameter estimation. It can perform the parameter estimation for multiple input-output datasets at the same time which greatly increases the reliability of the values obtained as the result of estimation. All the parameter values obtained as the result of the estimation were meaningful. From the parameter estimation results it is concluded that, although the parameter estimation results are deemed satisfactory, the time-varying effects of the cables on GMS axes dynamics degrades the success and reliability of the estimation especially for the azimuth axis. This is because the cable coming from the encoder connector and attached to the azimuth axis deforms more because of its structure. The whole MATLAB/Simulink model completed with

the estimation of the unknown parameters is highly reliable and detailed which increases the possibility of the controllers optimized using this model to perform as expected in the real system.

With the optimized PID parameters both the real system responses and simulation outputs to different magnitude step inputs are satisfying the time domain requirements, which shows the accuracy of the modeling and success of the controller parameter optimization together.

The simulation results for the case, where FMS and GMS motion profiles taken from the realistic scenarios, show that the GMS axes track the position commands very closely. Since the accuracy of the constructed MATLAB/Simulink model is proven by the very close results between the real system and simulation outputs obtained from the stationary environment tests, and the accuracy of the disturbance torque models are proven by exactly same results of the two models, the real system is expected to behave more or less the same as the model outputs.

The motion control card used for the real-time control system has PID controller blocks for each axis by default which impedes the implementation of other control algorithms. The PID algorithm is thought to be adequate since the design and production of the GMS is high quality with negligible unbalance for the axes and very low friction values for each axis. The results show that the PID controller is suitable and adequate for the task indeed.

It is observed that for a small system like GMS used in this study, which has very small moment of inertia values for the axes and very small Coulomb friction at the bearings, the elasticity of the cables can affect the system dynamics greatly. Since the cables are non-ideal elements and their effect on the system dynamics is time-varying; presence of the cables made both modeling and position controlling more

difficult. Since the use of the cables is inevitable, they should be guided in a proper way in order to lessen their unwanted effects on system dynamics.

Among all the nonlinearities, the PID output limit of the digital PID controllers in the motion control card has the most significant impact on system dynamics. For large position commands the PID output reaches its limit and cause the position responses of the GMS axes deviate from the expected behavior.

7.3 Recommendations for future work

In order to observe the real performance of the controllers for both axes under the effect of disturbance torques, tests should be carried out on FMS.

The effect of the elasticity of the cables is a problem. They are modeled as ideal springs but this is only an assumption. It can be seen from the cable characterization test figures in Chapter 3 that the free position of the cables continuously changes and the torque values that they apply at a position depends on the previous deformation of them. For more efficient and accurate position control of the GMS axes, the effects of the cables on GMS axes should be compensated. The cables from the connectors to the axes should be guided in a proper way.

Frequency domain analysis of the system is not performed since there were no frequency domain specs for the motion of GMS axes. Tests could be conducted with GMS to get the frequency responses of the axes.

The supply voltages of the amplifiers could be increased in order to get faster response from the system at the expense of increased maximum overshoot.

Since the PID output reaches its limit for large position commands, the PID controller gains could be decreased in order to prevent the PID output to reach its limit. This way the position response of the GMS axes could satisfy the requirements for even larger position commands. Of course, while decreasing the controller gains time domain requirements should be considered. The system may not achieve the desired response for smaller gains.

The data used in all stages of this thesis work is collected via the motion control card and just because of this fact the maximum data collection rate is limited to 333 Hz. By using DAQ cards much faster rates could be achieved for data collection. This way, the parameter identification could be performed for each component of the real-time control system separately, which in turn would lead to more reliable results.

REFERENCES

[1] Ideal Aerosmith, Inc., http://www.ideal-aerosmith.com/images/3axis_large.jpg , accessed date:10/01/2010.

[2] Ideal Aerosmith, Inc., http://www.ideal-aerosmith.com/images/5axis_large.jpg, accessed date: 10/01/2010.

[3] Nutfield Technology, Inc., URL: http://www.nutfieldtech.com/frameheads/ofh7.asp, accessed date:14/01/2010.

[4] Beasley, D. B., Saylor, D. A. (1999), Application of multiple IR projector technologies for AMCOM HWIL simulations, Proc. of SPIE Conference on Technologies for synthetic environments: Hardware in the loop testing IV, Orlando, FL, USA, 5-7 April 1999, pp. 223-231.

[5] Toshiba Machine Co., Ltd., URL:http://www.toshiba machine.co.jp/English/product/high/contents/img/polygon_004.jpg, accessed date: 14/01/2010.

[6] Beasley, D. B., Cooper, J. B., Saylor, D. A., Buford, J. A. (1997), Calibration and non-uniformity correction of MICOM's diode-laser-based infrared scene projector, Proc. of SPIE Conference on Technologies for synthetic environments: Hardware in the loop testing II, Orlando, FL, USA, 21-23 April 1997, pp. 91-101.

[7] Pinsky, E., Sturlesi D. (1997), Generation of dynamic IR scene for seekers testing, Proc. of SPIE Conference on Infrared Technology and Applications XXIII, Orlando, FL, USA, 21-25 April 1997, pp. 681-691.

[8] Swarup, N. (1993), Design and Control of a Two-Axis Gimbal System for use in Active Vision, B.Sc. Thesis, Mechanical Engineering Department, Massachusetts Institute of Technology, Cambridge, MA, USA.

[9] Smith, B. J., Schrenk, W. J., Gass, W. B., Shtessel, Y. B. (1999), Sliding Mode Control in a Two Axis Gimbal System, Proc. of IEEE Aerospace Conference, Snowmass at Aspen, CO, USA, 6-13 March 1999, Vol. 5, pp. 457-470.

[10] Wongkamchang, P., Sangveraphunsiri, V. (2008), Control of inertial stabilization systems using robust inverse dynamics control and adaptive control, Thammasat International Journal of Science and Technology, Vol. 13, No. 2, pp. 20-32.

[11] Sangveraphunsiri, V., Malithong, K. (2009), Control of inertial stabilization systems using robust inverse dynamics control and adaptive control, The 23rd Conference of the Mechanical Engineering Network of Thailand, Chiang Mai, Thailand, 4-7 November 2009, DRC-011299.

[12] Skoglar, P. (2002), Modelling and control of IR/EO-gimbal for UAV surveillance applications, M.Sc. Thesis, Applied Physics and Electrical Engineering Department, Linköping Institute of Technology, Linköping, Sweden.

[13] Rzasa, J.R. (2007), Design and application of pan and tilt servo gimbals in pointing, acquisition, and tracking, M.Sc. Thesis, Electrical Engineering Department, University of Maryland, College Park, MD, USA.

[14] National Instruments Corporation, URL: http://sine.ni.com/nips/cds/view/p/lang/en/nid/13826, accessed date: 20/01/2010.

[15] Advanced Motion Controls Corporation, URL: http://www.a-m-c.com/images/ productpics/s16a8.jpg, accessed date: 20/01/2010.

[16] National Instruments Corporation (2006), NI-Motion User Manual, pp. 3-4.

[17] Craig, J.J. (1989), Introduction to Robotics Mechanics and Control, Second Edition, Addison-Wesley Publishing Company, Inc., Reading, MA, USA, pp. 74.

[18] Özgören, M.K. (2002), Topological analysis of 6 joint serial manipulators and their inverse kinematic solutions, Mechanism and Machine Theory, Vol. 37, Issue 5, pp. 511-547.

[19] National Instruments Corporation (2006), NI 7350 User Manual, pp. A1-A7.

[20] Advanced Motion Controls Corporation, URL: http://www.a-m-c.com/ download/datasheet/s16a8.pdf, accessed date: 12/02/2010.

APPENDIX A

TECHNICAL SPECIFICATIONS OF THE MOTION CONTROL CARD

Servo Performance

PID update rate range	62.5µs/sample to 5 ms/sample
Max PID update rate	62.5 µs per 2 axes
8-axis PID update rate	250 µs total
Trajectory update rate	Same as PID update rate
Multi-axis synchronization	<1 update sample
Position accuracy	
Encoder feedback	±1 quadrature count
Analog feedback	±1 LSB
Double-buffered trajectory parameters	
Absolute position range	$\pm 2^{31}$ counts
Max relative move size	$\pm 2^{31}$ counts
Velocity range	1 to ±20,000,000 counts/s
RPM range[1]	±1,200,000 revolutions/min
Acceleration/deceleration[2]	244 to 512,000,000 counts/s^2 at a PID update rate of 250 µs
RPS/s range[1]	±256,000 revolutions/s^2
S-Curve time range	1 to 32,767 samples
Following error range	0 to 32,767 counts
Gear ratio	±32,767:1 to ±1:32,767

[1] Assumes a 2,000-count encoder.

[2] Refer to the *NI-Motion User Manual* for more information.

Figure A.1. Technical specifications of the motion control card (1/7) [19]

Servo control loop modes .....PID, PIVff, S-Curve, Dual Loop
- PID (Kp, Ki, and Kd) gains .....0 to 32,767
- Integration limit (Ilim).....0 to 32,767
- Derivative sample period (Td).....1 to 63 samples
- Feedforward (Aff, Vff) gains .....0 to 32,767
- Velocity feedback (Kv) gain .....0 to 32,767

Servo command analog outputs
- Voltage range.....±10 V
- Resolution.....16 bits (0.000305 V/LSB)
- Programmable torque (velocity) limits
  - Positive limit .....±10 V (–32,768 to +32,767)
  - Negative limit.....±10 V (–32,768 to +32,767)
- Programmable offset .....±10 V (–32,768 to +32,767)

Stepper Performance

Trajectory update rate range .....62.5 to 500 µs/sample
- Max update rate .....62.5 µs per 2 axes
- 8-axis update rate.....250 µs total

Multi-axis synchronization .....<1 update sample

Position accuracy
- Open-loop stepper .....1 full, half, or microstep
- Encoder feedback .....±1 quadrature count
- Analog feedback.....±1 LSB

Double-buffered trajectory parameters
- Position range .....$\pm 2^{31}$ steps
- Max relative move size.....$\pm 2^{31}$ steps
- Velocity range .....1 to 8,000,000 steps/s
- RPM range[1] .....±1,200,000 revolutions/min
- Acceleration/deceleration[2].....244 to 512,000,000 steps/s^2 at a PID update rate of 250 µs
- RPS/s range[1] .....±256,000 revolutions/s^2

[1] Assumes a 2,000-count encoder.

[2] Refer to the *NI-Motion User Manual* for more information.

Figure A.2. Technical specifications of the motion control card (2/7) [19]

S-curve time range ..... 1 to 32,767 samples
Following error range ..... 0 to 32,767 counts
Gear ratio ..... ±32,767:1 to ±1:32,767

Stepper outputs
Max pulse rate ..... 8 MHz (full, half, and microstep)
Max pulse width ..... 6.5 µs at <40 kHz
Min pulse width ..... 40 ns at >4 MHz
Step output mode ..... Step and direction or CW/CCW
Voltage range ..... 0 to 5 V
Output low voltage ..... 0.6 V at 64 mA sink
Output high voltage ..... Totem Pole: 2V at 16 mA source; open collector: built-in 3.3 kΩ pull-up to +5 V
Polarity ..... Programmable, active high or active low

System Safety

Watchdog timer function ..... Resets board to startup state
Watchdog timeout ..... 63 or 256 ms, programmable

Shutdown input
Voltage range ..... 0 to 5 V
Input low voltage ..... 0.8 V
Input high voltage ..... 2 V
Built-in pull-up resistor ..... 3.3 kΩ to +5 V
Polarity ..... Rising edge
Control ..... Disable all axes and command outputs; resets I/O to default states
Host +5 V max current sourced from controller ..... 100 mA at 5 V

Motion I/O

Encoder inputs ..... Quadrature, incremental, single-ended
Max count rate ..... 20 MHz

Figure A.3. Technical specifications of the motion control card (3/7) [19]

Min pulse width.....Programmable; depends on digital filter settings

Voltage range.....0 to 5 V

Input low voltage.....0.8 V

Input high voltage.....2 V

Built-in pull-up resistor .....3.3 kΩ to +5 V

Min index pulse width .....Programmable; depends on digital filter settings

Forward, reverse, and home inputs

Number of inputs.....3 per axis, up to 24

Voltage range.....0 to 5 V

Input low voltage.....0.8 V

Input high voltage.....2 V

Built-in pull-up resistor .....3.3 kΩ to +5 V

Polarity .....Programmable, active high or active low

Min pulse width

Limit filters enabled .....1 ms

Limit filters disabled .....50 ns

Control.....Individual enable/disable, stop on input, prevent motion, Find Reference

Trigger (position capture) inputs

Number of inputs.....Up to 8 (Encoders 1 through 8)

Voltage range.....0 to 5 V

Input low voltage.....0.8 V

Input high voltage.....2 V

Built-in pull-up resistor .....3.3 kΩ to +5 V

Polarity .....Programmable, active high or active low

Min pulse width.....100 ns

Max capture latency .....100 ns

Capture accuracy .....1 count

Max capture rate (non-buffered) .....150 Hz

Figure A.4. Technical specifications of the motion control card (4/7) [19]

Max buffered capture rate[1] ..... 2 kHz per axis

Breakpoint (position compare) outputs

Number of outputs ..... Up to 8 (Encoders 1 through 8)

Voltage range ..... 0 to 5 V

Output low voltage ..... 0.6 V at 64 mA sink

Output high voltage ..... Totem Pole: 2 V at 16 mA source; open collector: built-in 3.3 kΩ pull-up to +5 V

Polarity ..... Programmable, active high or active low

Max trigger rate (non-buffered) ..... 150 Hz

Max buffered trigger rate[1] ..... 2 kHz per axis

Max periodic rate ..... 4 MHz per axis

Minimum pulse width (pulse mode only) ..... 200 ns

Inhibit/enable output

Number of outputs ..... 1 per axis, up to 8

Voltage range ..... 0 to 5 V

Output low voltage ..... 0.6 V at 64 mA sink

Output high voltage ..... Totem Pole: 2 V at 16 mA source; open collector: built-in 3.3 kΩ pull-up to +5 V

Polarity ..... Programmable, active high or active low

Control ..... MustOn/MustOff or automatic when axis off

Analog inputs

Control ..... Assigned to axis for analog feedback or general-purpose analog input

Number of inputs ..... Up to 8, multiplexed, single-ended

Multiplexer scan rate ..... 25 µs per enabled ADC

[1] Assumes a PID update rate of 250 µs. 2 kHz per axis for PID rates between 62.5 and 250 µs, and 1 kHz per axis for PID rates greater than 250 µs. This value is not to exceed 8 kHz total for all ongoing buffered breakpoint (position compare) and trigger (position capture) operation.

Figure A.5. Technical specifications of the motion control card (5/7) [19]

Input coupling.....DC
Input impedance .....100 MΩ min
Voltage range (programmable).....±10 V, ±5 V, 0–10 V, 0–5 V
Bandwidth.....234 kHz
Resolution.....16 bits, no missing codes
Monotonicity .....Guaranteed
Absolute accuracy
all ranges.....0.5% of full-scale

System noise
±10 V.....220 µVrms, typical
±5 V.....120 µVrms, typical
0–10 V.....130 µVrms, typical
0–5 V.....60 µVrms, typical

Maximum working voltage .....±11 V

Overvoltage protection
Powered on .....±25 V
Powered off .....±15 V

Analog outputs
Number of outputs.....Up to 8, single-ended
Output coupling.....DC
Voltage range.....±10 V
Output current.....±5 mA
Minimum load .....2 kΩ at full-scale
Resolution.....16 bits, no missing codes
Monotonicity .....Guaranteed
Absolute accuracy .....0.5% of full-scale
Noise.....100 µVrms Max
Protection.....Short-circuit to ground
Settling time.....15 µs, full-scale step
Analog reference output voltage .....7.5 V (nominal)
Analog reference output current.....5 mA

Onboard temperature sensor accuracy.....±4 °C

Figure A.6. Technical specifications of the motion control card (6/7) [19]

Digital I/O

Ports ..... Up to 8 8-bit ports
- Line direction ..... Individual bit programmable

Inputs
- Voltage range ..... 0 to 5 V
  - Input low voltage ..... 0.8 V
  - Input high voltage ..... 2.0 V
- Polarity ..... Programmable, active high or active low

Built-in pull-up resistor ..... 10 kΩ, configurable pull-up to +5 V or pull-down to GND

Outputs
- Voltage range ..... 0 to 5 V
  - Output low voltage ..... 0.45 V at 24 mA
  - Output high voltage ..... 2.4 V at 24 mA
- Max total DIO current
  - Sourced from controller ..... 1 A
- Polarity ..... Programmable, active high or active low
- PWM outputs
  - Number of PWM outputs 2
  - Max PWM frequency ..... 50 kHz
  - Resolution 8-bit
  - Duty cycle range ..... 0 to (255/256)%
  - Clock sources ..... Internal or external

RTSI

Trigger lines 8

PXI Star Trigger (PXI-7350 only) 1

Figure A.7. Technical specifications of the motion control card (7/7) [19]

APPENDIX B

AMPLIFIER SPECIFICATIONS

SPECIFICATIONS

Power Specifications		
Description	Units	Value
DC Supply Voltage Range	VDC	20 - 80
DC Bus Over Voltage Limit	VDC	86
Maximum Peak Output Current[1]	A	16
Maximum Continuous Sine Wave Current	Arms	8
Maximum Power Dissipation at Continuous Current	W	32
Minimum Load Inductance (Line-To-Line)[2]	µH	200
Switching Frequency	kHz	33
Control Specifications		
Description	Units	Value
Command Sources	-	±10 V Analog
Commutation Methods	-	External, Sinusoidal
Modes of Operation	-	Current
Motors Supported	-	Brushless
Hardware Protection	-	Over Current, Over Temperature, Over Voltage, Short Circuit (Phase-Phase & Phase-Ground)
Mechanical Specifications		
Description	Units	Value
Agency Approvals	-	CE Class A (EMC), CE Class A (LVD), cUL, RoHS, UL
Size (H x W x D)	mm (in)	129.3 x 77.7 x 38.6 (5.1 x 3.1 x 1.5)
Weight	g (oz)	280 (9.9)
Heatsink (Base) Temperature Range[3]	°C (°F)	0 - 65 (32 - 149)
Storage Temperature Range	°C (°F)	-40 - 85 (-40 - 185)
P1 Connector	-	15-pin, female D-sub
P2 Connector	-	5-port, 5.08 mm spaced, screw terminal

Notes

1. Maximum duration of peak current is ~2 seconds.
2. Lower inductance is acceptable for bus voltages well below maximum. Use external inductance to meet requirements.
3. Additional cooling and/or heatsink may be required to achieve rated performance.

Figure B.1. Specifications of three phase sinusoidal brushless amplifier [20]

HARDWARE SETTINGS

Switch Functions

Switch	Description	Setting	
		On	Off
1	Peak Current Limit. Sets the peak current limit to 50% or 100% of the maximum peak current. Must be set the same as switch 2.	100%	50%
2	Peak Current Limit. Sets the peak current limit to 50% or 100% of the maximum peak current. Must be set the same as switch 1.	100%	50%
3	Current loop proportional gain adjustment. Must be set the same as switch 5. ON by default.	Decrease	Increase
4	Current loop integral gain. Activates or deactivates integration. Must be set the same as switch 6. OFF by default.	Inactive	Active
5	Current loop proportional gain adjustment. Must be set the same as switch 3. ON by default.	Decrease	Increase
6	Current loop integral gain. Activates or deactivates integration. Must be set the same as switch 4. OFF by default.	Inactive	Active
7	Bit 0 of binary value for RMS current limit setting. See details below.	1	0
8	Bit 1 of binary value for RMS current limit setting. See details below.	1	0
9	Reserved Function	-	-
10	Inhibit logic. Sets the logic level of inhibit pins.	Active Low	Active High

Additional Details

Switches 1 & 2, switches 3 & 5, and switches 4 & 6 must be set the same. Switches 7 and 8 can be used to reduce the continuous current limit to a percentage given in the table below. 100% means no reduction.

% Of Maximum Continuous Current Limit	Switch 7	Switch 8
25	OFF	OFF
50	OFF	ON
100	ON	OFF
	ON	ON

Figure B.2. Hardware settings for three phase sinusoidal brushless amplifier [20]

APPENDIX C

MATLAB CODE TO PERFORM TRANSFORMATION FROM S-DOMAIN TO Z-DOMAIN USING 'ZOH' METHOD

```
K_dac=20/(2^16);                    %DAC converter gain
K_rad2count=1000000/pi;             %Conversion from radians to
encoder counts
Ka=1.6;                                   %Amplifier gain
Ts=0.000250;                              %Controller update period

%Azimuth Axis
%--------------------------------------

G_teta_m_azimuth=tf([(Kt_azm*Ka*Kp_azm)(Kt_azm*Ka*Ki_azm)],
[(J_azm*L_azm) (b_azm*L_azm+J_azm*Kp_azm+J_azm*R_azm)
(k_azm*L_azm+J_azm*Ki_azm+b_azm*Kp_azm+b_azm*R_azm+Kt_azm*Ke_azm)
(k_azm*Kp_azm+k_azm*R_azm+b_azm*Ki_azm) (k_azm*Ki_azm)]);

G_ol_azimuth=G_teta_m_azimuth*K_rad2count*K_dac;

G_ol_azimuth_discrete=c2d(G_ol_azimuth,Ts,'zoh');

%Elevation Axis
%--------------------------------------

G_teta_m_elevation=tf([(Kt_elv*Ka*Kp_elv)(Kt_elv*Ka*Ki_elv)],
[(J_elv*L_elv) (b_elv*L_elv+J_elv*Kp_elv+J_elv*R_elv)
(k_elv*L_elv+J_elv*Ki_elv+b_elv*Kp_elv+b_elv*R_elv+Kt_elv*Ke_elv)
(k_elv*Kp_elv+k_elv*R_elv+b_elv*Ki_elv) (k_elv*Ki_elv)]);
G_ol_elevation=G_teta_m_elevation*K_rad2count*K_dac;

G_ol_elevation_discrete=c2d(G_ol_elevation,Ts,'zoh');
```