

**MODELING AND STATISTICAL CONTROL OF
A GIMBALED LASER TARGET SYSTEM**

A Thesis
Submitted to
The Temple University Graduate Board

In Partial Fulfillment
of the requirements for the Degree
MASTER OF SCIENCE
in ELECTRICAL ENGINEERING

by
Firdous Saleheen
January, 2014

Thesis Approvals:

Chang-Hee Won, PhD, Thesis Advisor, Electrical and Computer Engineering Department
Saroj Biswas, PhD, Electrical and Computer Engineering Department
Li Bai, PhD, Electrical and Computer Engineering Department

UMI Number: 1552298

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 1552298

Published by ProQuest LLC (2014). Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against unauthorized copying under Title 17, United States Code



ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

ABSTRACT

The space-based solar power system is an alternative to the ground-based solar power system because of its round-the-clock availability. For the space-based solar power transmission, the accurate pointing of a laser from space to ground poses a challenging control task. A gimbaled laser target system, which is used for pointing laser to a target, is a test bench for such a transmission system. The objective of this research is to determine the optimal controller for the gimbaled laser target system in terms of pointing error and error variation. In order to achieve the objective, we modeled the gimbaled laser target system, simulated the model with the controllers, and tested them on the test bench.

In this thesis, we developed a mathematical model of a two-axis gimbaled laser target system. The model consists of a pitch-yaw gimbal for the dynamic laser motion, brushless dc motors for actuating the gimbal, and an image-based position sensor. We used a Proportional-Integral-Derivative (PID) controller as the basis for the performance comparison since it is the most commonly used control method in the industry. Then we compared the PID controller with two statistical control methods Linear Quadratic Gaussian (LQG), and Minimal Cost Variance (MCV) optimal controllers.

We evaluated the pointing performance of the controllers by measuring the mean and the standard deviation of the pointing error. The simulation results indicated that the statistical controllers perform better than the PID controller under Gaussian disturbances. Between the statistical controllers, the LQG method had the smaller pointing error, while the MCV method had the smaller standard deviation of the pointing error. We then implemented the PID, LQG, and MCV controllers in an off-the-shelf dSPACE digital signal processing controller board, and tested the controllers on the test bench in a real time environment. The experimental results

showed that the LQG method decreased the mean pointing error by 46.28% compared to the PID method. The LQG method reduced the standard deviation of pointing error by 47.85% compared to the PID method. The MCV method reduced the standard deviation of the pointing error by 53.09% compared to the LQG method.

Both the simulation and experimental results showed that the MCV controller improved the pointing error variation performance over the LQG controller significantly, while slightly degrading the pointing error performance of the gimbaled laser target system. Experimental results indicate that the statistical controllers will provide a design parameter either to improve the mean pointing error or the standard deviation of the pointing error for the gimbaled laser target system. Subsequently, we believe that the statistical controllers will improve the space-based solar power transmission efficiency.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my adviser, Dr. Chang-Hee Won of the Electrical and Computer Engineering Department at Temple University for his continuous support, patience, motivation, and enthusiasm. His guidance helped me in all times of research and writing this thesis.

I would like to thank the rest of my thesis committee members: Dr. Saroj Biswas and Dr. Li Bai for their insightful comments and suggestions.

In addition, I would like to thank my labmates in CSNAP. I really appreciate the help of my colleague and friend Salvatore Giorge for discussing the problems in this work, conducting the experiments, and reviewing my writing. Also, I would like to thank my good friend Ashfaqur Rahman for reviewing my writing.

I am eternally indebted to my parents Md. Alamgir Hussain and Ayesha Begum Irani for having faith in me and encouraging me to fulfil my ambition. I would like to thank all of my friends, specially my roommate Shoumik Roychoudhury. Most of all, I want to thank my fiancée, Nancy Dutta, for loving and supporting me throughout the development of this work.

Finally, I would like to extend my thanks for the financial support from National Science Foundation Grant AIS ECCS-0969430.

TABLE OF CONTENTS

	Page
ABSTRACT	ii
ACKNOWLEDGEMENTS	iv
LIST OF FIGURES	ix
LIST OF TABLES	xiv
 CHAPTER	
1 INTRODUCTION	1
1.1 Motivations	1
1.2 Research Objectives	3
1.3 Background and Literature Review	3
1.3.1 Modeling of Pointing and Tracking System	3
1.3.2 Brushless dc Motor for the Gimbaled Laser Target System	5
1.3.3 Disturbances to the Gimbaled Laser Target System	6
1.3.4 Control Methods for Gimbaled Laser Target System	11
1.4 Contributions	16
1.5 Thesis Organization	17
2 NON-LINEAR MATHEMATICAL MODEL OF GIMBALED LASER TARGET SYSTEM	19
2.1 Overview	19
2.2 Subsystem Description	22
2.2.1 Gimbal	22
2.2.2 Brushless DC Motor	43

2.2.3	Servo Drive	50
2.2.4	Laser Pointer Model	61
2.2.5	Image-based Position Sensor	61
2.3	Derivation of a Reduced order Gimbaled Laser Target System	70
2.3.1	Derivation of a Linear Servo Drive and Brushless DC Motor Model	70
2.3.2	Nonlinear Gimbaled Laser Target System Model	80
3	LINEAR GIMBALED LASER TARGET SYSTEM	93
3.1	Overview	93
3.2	Derivation of Linear Gimbaled Laser Target Model	93
3.2.1	Assumptions	93
3.2.2	Nonlinear Torque term due to Base Angular Rate, $T_{D,base}$	94
3.2.3	Non-linear Terms in State Matrix $A(t)$	96
3.2.4	Non-linear Term $F(X, t)$	97
3.2.5	Disturbance Torque term T_{dist}	97
3.2.6	Linear Model of Gimbaled Laser Target System	97
4	CONTROL METHODOLOGIES	105
4.1	Overview	105
4.2	PID Control Approach	105
4.2.1	PID Controller Structure	105
4.2.2	PID Controller Tuning	106
4.3	Statistical Control Approach	107
4.3.1	Full State Feedback Case	107
4.3.2	Output Feedback Case	110
5	SIMULATIONS	117
5.1	Overview	117
5.2	Simulation Parameters	117

5.3	PID Control Simulations	117
5.3.1	Parameter Selection Process	117
5.3.2	PID Simulation Results	122
5.4	Full State Feedback LQG Control Simulations	126
5.5	Full State Feedback MCV Control Simulations	130
5.5.1	LQG Control as a special case of MCV Control	130
5.6	Performance Metrics	130
5.7	Comparison of Full State Feedback LQG and MCV Controller for GLTS	131
5.7.1	Pointing Responses and Pointing Error	131
5.7.2	Angular Velocity Responses	136
5.7.3	RMS Pointing Error with varying γ	141
5.7.4	Standard Deviation of Pointing Error with varying γ	142
5.8	Output Feedback LQG Control	143
5.9	Comparison of Output Feedback LQG and MCV Control	143
5.9.1	Pointing Responses	143
5.9.2	RMS Pointing Error with varying γ	146
5.9.3	Standard Deviation of Pointing Error with varying γ	147
6	HARDWARE IMPLEMENTATION AND EXPERIMENT OF CONTROLLERS	149
6.1	Overview	149
6.2	A Brief Overview of Hardware and Software Set-up	149
6.2.1	Gimbal Assembly	151
6.2.2	Laser	152
6.2.3	Motors, Amplifiers, and Power Supply	152
6.2.4	Host Computers for Controller and Image Processing	152
6.2.5	dSPACE Controller Board	154
6.2.6	Image-based Position Sensor	156

6.2.7	Linear Positioner for Disturbance Generation	157
6.3	System with Open Loop Position Control	157
6.4	System with PID Control: Output Feedback	161
6.5	System with LQG Control: Output Feedback	167
6.6	System with MCV Control: Output Feedback	167
6.7	Experiments for Comparison of System with PID, LQG, and MCV Control	169
6.7.1	For Control of Yaw Axis Only	169
6.7.2	For Control Yaw and Pitch Axis Simultaneously	175
7	CONCLUSIONS AND FUTURE WORKS	183
7.1	Conclusions	183
7.2	Future Works	184
	REFERENCES	185

LIST OF FIGURES

1.1	Solar power satellite system conceptual diagram	1
1.2	Pointing and tracking system: inter-spacecraft optical communication	4
1.3	Model of Coulomb friction torque	7
1.4	Model of static friction torque	7
1.5	Model of viscous friction torque	8
1.6	Model of Coulomb plus viscous friction torques	8
1.7	Model of Coulomb plus static plus viscous friction torques	9
1.8	Model of Coulomb plus static plus viscous plus Stribeck friction torques	9
1.9	Friction interface between two surfaces is thought of as a contact between bristles	10
2.1	Hardware implementation of a gimbaled laser target system	20
2.2	Architecture of the control system of a two-axis gimbaled laser target system	20
2.3	Two-axis gimbal system coordinates	23
2.4	Rotation of frames	24
2.5	Two-axis Newmark GM-6 gimbal with angle rotation among frames	25
2.6	On the coordinate transformation for infinitesimal rotation	29
2.7	Block diagram of pitch gimbal	34
2.8	Exponential model of nonlinear friction torque	35
2.9	Cutway view of permanent magnet brushless DC motor assembly	43
2.10	Equivalent circuit of a brushless DC motor	44
2.11	Permanent magnet DC motor waveform at six different rotor position	46

2.12	Brushless DC motor model	50
2.13	Speed controller of servo drive	51
2.14	Current command generator of servo drive	53
2.15	PWM current controller of servo drive	54
2.16	Three-phase MOSFET inverter for derivation of voltage relation	57
2.17	Three-phase MOSFET inverter waveform from Table 2.4	61
2.18	Speed controlled permanent magnet brushless DC motor servo drive scheme	62
2.19	Image-based position sensor hardware and software block diagram	63
2.20	Image tracking block diagram	63
2.21	Laser dot coordinates in camera frame and after projection in image plane	64
2.22	Relation between laser pointer frame and camera frame	66
2.23	Coordinate system on white receiver screen	68
2.24	Architecture of the control system of a two-axis gimbaled laser target system	71
2.25	Block diagram of linear pitch gimbal	73
2.26	Block diagram of mechanical part of motor	73
2.27	Block diagram of pitch gimbal and motor with gear	74
2.28	Block diagram of brushless DC motor with inner current control loop	75
2.29	Block diagram of pitch gimbal and motor	76
2.30	Redrawn block diagram of pitch gimbal and motor	76
2.31	Redrawn Block diagram of brushless DC motor with inner current control loop	77
2.32	Block diagram of simplified current control loop	79
3.1	Side view of the laser pointing device and the screen	93
5.1	Initial Tuning (Matlab PID tuner)	120
5.2	Reference tracking response after tuning (Matlab PID tuner)	120
5.3	Tuning for disturbance rejection (Matlab PID tuner)	121

5.4	Reference tracking response after tuning for disturbance rejection	121
5.5	Pointing response with PID controller without noise	123
5.6	Pointing response with PID controller under Gaussian white noise	124
5.7	Pointing response with PID controller under Gaussian white noise with optimized parameters . .	124
5.8	Pointing responses with PID controller for three cases	125
5.9	Pointing responses with LQG and PID controller under Gaussian white noise	129
5.10	Vertical pointing responses with MCV controllers	132
5.11	Horizontal pointing responses with MCV controllers	132
5.12	Vertical pointing error with MCV controllers	133
5.13	Horizontal pointing error with MCV controllers	134
5.14	Average vertical pointing responses with MCV controllers	134
5.15	Average horizontal pointing responses with MCV controllers	135
5.16	Average vertical pointing error with MCV controllers	135
5.17	Average horizontal pointing error with MCV controllers	136
5.18	Angular velocity ω_{ey} responses with MCV controllers	137
5.19	Angular velocity ω_{ey} responses with MCV controllers (zoomed view)	137
5.20	Angular velocity ω_{ez} responses with MCV controllers	138
5.21	Angular velocity ω_{ez} responses with MCV controllers (zoomed view)	138
5.22	Average angular velocity ω_{ey} responses with MCV controllers	139
5.23	Average angular velocity ω_{ey} responses with MCV controllers (zoomed view)	140
5.24	Average angular velocity ω_{ez} responses with MCV controllers	140
5.25	Average angular velocity ω_{ez} responses with MCV controllers (zoomed view)	141
5.26	RMS pointing error with MCV controllers (full-state feedback)	142
5.27	Standard Deviation of pointing error with MCV controllers (full-state feedback)	143
5.28	Vertical pointing responses (measured and estimated) with LQG controllers	144
5.29	Horizontal pointing responses (measured and estimated) with LQG controllers	145

5.30	Vertical pointing responses with MCV controllers (output feedback)	145
5.31	Horizontal pointing responses with MCV controllers (output feedback)	146
5.32	RMS pointing error with MCV controllers (output feedback)	147
5.33	Standard Deviation of pointing error with MCV controllers (output feedback)	148
6.1	Hardware implementation of a gimbaled laser target system	150
6.2	Block diagram of hardware and software of gimbaled laser target system	150
6.3	Newmark GM-6 gimbal (front view)	151
6.4	Newmark GM-6 gimbal (rear view)	151
6.5	Laser pointer	152
6.6	Brushless DC servo motor	153
6.7	Copley controls amplifier	153
6.8	DS1104 controller card	155
6.9	CP1104 connector panel	155
6.10	dSPACE RTI Blockset	155
6.11	Simulink block diagram for image processing	156
6.12	Screen coordinate system measured in centimeter	157
6.13	Linear positioner for disturbance generation	158
6.14	Simulink block diagram of yaw axis open loop position controller	158
6.15	Experiment results from yaw axis for open loop position control	159
6.16	Open loop position control for pitch gimbaled laser target system	160
6.17	Open loop position control for yaw gimbaled laser target system	161
6.18	Simulink block diagram of yaw axis PID controller	162
6.19	Duty cycle and yaw motor angular velocity	162
6.20	Duty cycle and pitch motor angular velocity	163
6.21	PID control for pitch gimbaled laser target system	164

6.22	PID control for yaw gimbled laser target system	165
6.23	Simulink block diagram of two-axis PID controller	166
6.24	Simulink block diagram of yaw axis LQG controller	166
6.25	Simulink block diagram of Kalman estimator for yaw axis LQG controller	167
6.26	Simulink block diagram of two-axis LQG controller	168
6.27	Simulink block diagram of yaw axis MCV controller	168
6.28	Simulink block diagram of Kalman estimator for yaw axis MCV controller	169
6.29	Simulink block diagram of two-axis MCV controller	170
6.30	Single axis disturbance	170
6.31	PID, LQG, MCV controller comparison	172
6.32	RMS pointing error for PID, LQG, and MCV	172
6.33	Standard Deviation of pointing error for PID, LQG, and MCV	173
6.34	Controller comparison for PID, LQG, and MCV ($\gamma = 0.5$)	174
6.35	Disturbance for x axis	176
6.36	Disturbance for y axis	176
6.37	Horizontal RMS pointing error for PID, LQG, and MCV controllers	178
6.38	Standard Deviation of horizontal pointing error for PID, LQG, and MCV controllers	178
6.39	Vertical RMS pointing error for PID, LQG, and MCV controllers	179
6.40	Standard Deviation of vertical pointing error for PID, LQG, and MCV controllers	179
6.41	PID, LQG, and MCV controller comparison for horizontal pointing	180
6.42	PID, LQG, and MCV controller comparison for vertical pointing	181

LIST OF TABLES

2.1	Rotor position dependent function with θ_r in radian	47
2.2	Magnitude of phase currents based on rotor position θ_r in radian	53
2.3	Switching states possibilities for three phase inverter	58
2.4	Midpoint voltage and phase voltage waveform for three phase inverter	60
5.1	Simulation parameters of servo motor drive in gimbaled laser target system	118
5.2	Simulation parameters of gimbal and position sensor of laser target system	119
5.3	PID tuned parameters	122
6.1	Open loop position control results	161
6.2	PID Control Results	165
6.3	RMS and Standard Deviation of pointing error for yaw gimbaled laser target system	173
6.4	RMS and Standard Deviation of pointing error for PID, LQG, and MCV controllers	177
6.5	Pointing Performances of PID, LQG, and MCV controllers	182

CHAPTER 1

INTRODUCTION

1.1 Motivations

Solar energy is a renewable energy source. It is clean and environmentally friendly. However, the sun shines only part of each 24 hours, and sometimes it is covered by clouds. Hence, the ground-based solar energy suffers from irregularities in supply and lack of dependability. A 24-hour operation is possible by placing a solar energy collector in space. This technique is called space based solar power.

Fig. 1.1 shows a conceptual diagram of a space-based solar power system. The sunlight hits the concentrating mirrors in the satellite. The electricity generated by the photo-voltaic cells is converted to microwave beam or laser beam by a beam generator, and transmitted to the ground-based receiver.

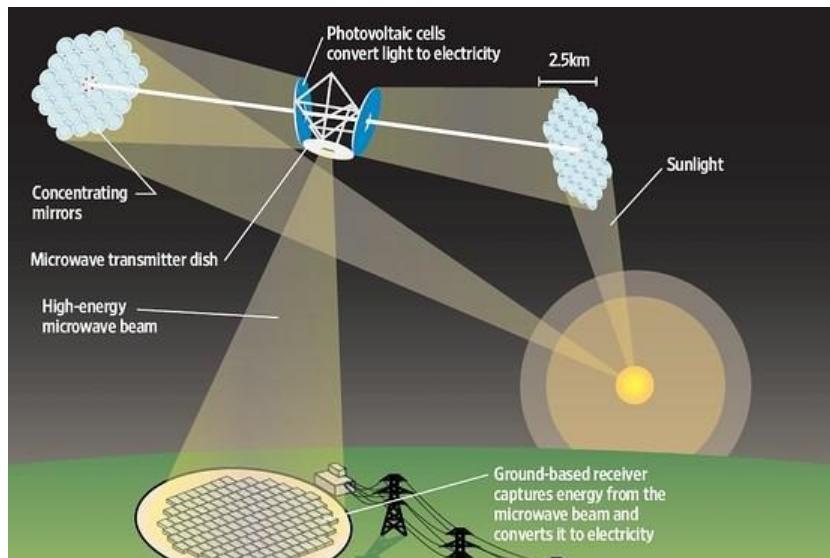


Figure 1.1: Solar power satellite system conceptual diagram

Peter Glaser (1968) first suggested such a space-based solar power system, and has been studied by NASA (The National Aeronautics and Space Administration) since 1972. In early 1976 the NASA and DOE (Department of Energy) initiated an SPS (Solar Power Satellite) Concept Development and Evaluation Program (Dietz et al., 1981). Under this program, scientists developed a reference system (called as 1979 SSPS Reference System). The reference system featured a very large solar array platform of ($5.3\text{km} \times 10.7\text{km}$) area and a gimbaled, microwave beam transmitting antenna (1 km diameter) in space for transmission. On the receiving side, there is a ground-based rectenna ($10\text{km} \times 13\text{km}$) for delivering up to 5 GW electricity. NASA also explored the possibility of using a laser instead of a microwave beam because of public safety and health issues.

A major technical concern associated with the space-based solar power system is the power efficiency of the transmission. In space, satellite attitude or orientation is affected by gravity torque, magnetic torque, aerodynamic torque, and solar radiation torque. The effects of these torques have been explained in details by Hughes (2012). Efficient power transmission requires accurate beam alignment or pointing. Accurate pointing is also important for health and public welfare. The heat caused by a laser beam can be hazardous to public safety. It is also a potential danger to aircraft which may fly through the beam. Therefore, accurate pointing of the laser beam is crucial for this application.

The gimbaled laser targeting system is considered as a model for the space-based solar power transmission system. In the gimbaled laser target system, a laser pointer is used to point at a specific point on an image-based position sensor. For the optimal control of the gimbaled laser target system, first we need an accurate model of the system. Then we require a control method that provides optimal pointing error and error variation. Thus the main motivation behind this thesis is the opportunity to formulate a safer, efficient control technique to implement in the space-based solar power transmission system by determining an optimal controller for the laser targeting system.

1.2 Research Objectives

The main objective of this research is to determine the optimal controller for the gimbaled laser target system in terms of pointing error and error variation. In order to achieve the objective, the first step is to develop an accurate model of the gimbaled laser target system. We shall develop a non-linear model, then linearize it for the control purpose. We shall add disturbance to the linear model to make it stochastic. The disturbance will be modelled with Gaussian white noise.

The next step is to evaluate and compare the performance of different control methods for the developed model through simulations. We consider the statistical controllers for the optimal control of the system under Gaussian white noise. The first statistical controller is LQG (minimal mean), and the second is MCV (minimal variance). These two statistical controllers will be compared with the simulation results of the PID controller.

The third step is to implement the PID, LQG, and MCV controllers in a dSPACE digital signal processing controller board, test the controllers on the hardware in a real time environment, and compare the pointing error and error variation performances.

1.3 Background and Literature Review

In this section, we present a background and literature review on modeling and control of a gimbaled laser target system.

1.3.1 Modeling of Pointing and Tracking System

A gimbaled laser target system is a type of pointing and tracking system. The pointing and tracking system consists of an electro-opto-mechanical assembly which provides physical interface between the optical sensor and the tracker, a control system, and auxiliary equipment, which measures target location relative to the reference system (Hilkert, 2008). This system is widely used in ground based telescope stabilization, missile guidance system, antenna assemblies, and laser communications (Wang and Williams, 2008). Whatever the application, all pointing and tracking systems have a common goal of controlling the inertial orientation

of its payload or Line-of-Sight (LOS). The LOS can be the center of the field of view (FOV) of a telescope, the aim-point of a beam or weapon, or a direction the sensor is pointed. The operating principle depends on how the electromechanical assembly is assembled. Several approaches can be used such as platform stabilization, steering stabilization, momentum wheel stabilization, and feedforward technique (Masten, 2008).

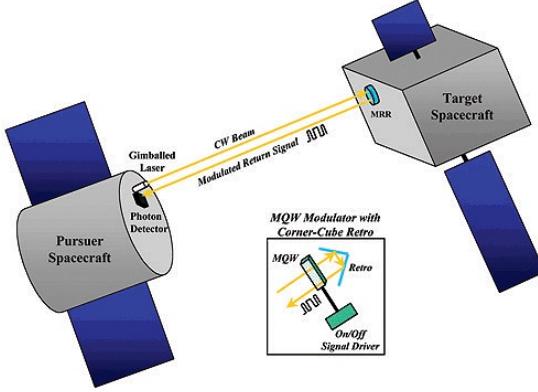


Figure 1.2: Pointing and tracking system: inter-spacecraft optical communication

Fig. 1.2 shows an example of a pointing and tracking system. This is an inter-spacecraft optical communication and navigation where the pursuer spacecraft illuminates the target platform carrying the modulating retro-reflectors with a laser beam and the laser beam is reflected back. Thus it achieves laser pointing and tracking.

Similarly, our gimbaled laser target system consists of a two-axis gimbal with laser as plant, brushless dc motor with servo drives as actuator, a dSPACE controller board as a controller, and an image-based position sensor for measuring laser dot location on sensor screen. The goal of the control system is to control the orientation of laser in such a way that it always points at the center of the sensor screen.

Rue (1969; 1974) introduced mathematical modeling of electro-optical elements as well as gimbal assembly. Ekstrand (2001) established the equation of motion for a two-axis yaw-pitch gimbal configuration assuming the gimbals are rigid bodies and have no mass imbalance. Kennedy (2003) also formulated the equation of motion where he investigated the effect of the sensor position at different places of the gimbal assemblies. Although many attempts were

made concerning the accurate dynamic modeling of the gimbal, it is still not a trivial problem in practical applications.

In this thesis, we followed the papers of Kennedy (2003) and Ekstrand (2001) for the modelling of the two axis gimbal part of the system. Ekstrand derived the dynamic model of the two axis gimbal. He considered the product of inertia in inertia tensor element. His model is more general. Kennedy considers the product of inertia negligible which simplifies the derivation of the model. We develop our model following the general model of Ekstrand, then simplify it following Kennedy's assumptions. In the next section, we discuss about the choice of actuator to drive this gimbal.

1.3.2 Brushless dc Motor for the Gimbaled Laser Target System

We found in the literature the dc motor is considered as the gimbal. In this thesis, we used a three phase brushless dc motor with a servo drive as the actuator. Krishnan (2009) developed a model for a permanent magnet brushless dc motor drive.

Brushless dc motor has three phase windings on the stator and permanent magnet rotor (Krishnan, 2001). The induced back electromotive force (emf) of this type of motor is trapezoidal. One reason of choosing this motor is the simplicity of its control. To initiate the onset and commutation of current in the phase of a motor, the beginning and end of the constant flat portion of the induced emf have to be tracked. So there are only six discrete positions for a three-phase motor in each of the electrical cycle. These signals are generated using three Hall sensors displaced from each other by 120 electrical degrees. The Hall sensors are mounted facing a small magnet wheel fixed to the rotor having the same number of poles as in the rotor of the brushless dc motor. Such an arrangement tracks the absolute position of the rotor magnets and hence the shape and position of the induced emfs in all the motor phases. The other reasons for choosing this type of motor for its reliability, reduced noise and longer lifetime.

Another motor of this family is permanent magnet synchronous motor. It requires continuous and instantaneous absolute rotor position since the induced emf is sinusoidal (Ramu, 2009). Therefore, if brushless dc motor is compared to synchronous counterpart, it saves major cost

in the feedback sensor. Also, since synchronous motors involve vector operation, brushless DC motor enjoys simpler control since it does not require that type of operation.

In brushed DC motor, the induced emf is proportional to the speed of the motor, and the torque is proportional to the current (Krishnan, 2001). When the speed increases, the induced emf increases. Effectively, it decreases the net applied voltage to the motor winding. Consequently, the current decreases, so does the motor torque. The brushless DC motor offers more torque per weight and more torque per watt, hence the efficiency than the brushed DC motor. Also, the absence of brush in brushless DC motor gives it longer lifetime.

1.3.3 Disturbances to the Gimbaled Laser Target System

The gimbaled laser target system utilizes a two-axis gimbal which is one of the major sources of noise in the system. To develop a good model of gimbal dynamics, we must model major disturbances that the gimbal experiences. The gimbal experiences friction torque, cable restraint torque, inertial disturbance torques, imbalance torque, kinematic and intergimbal coupling, vibration, and environmental disturbances (Hilkert, 2008; Masten, 2008). Out of those friction torques, cable restraint torque, and inertial disturbance torques have the dominant effects. The effects due to the rest of the disturbances can be considered negligible. Here we present a literature review on friction torque modeling.

Friction torque is typically the dominant disturbances in gimbaled laser target system. Classic friction model maps between velocity and friction force (Armstrong-Hélouvy, 1991). For rotational motion, angular velocity and torque are equivalent to velocity and friction force. Classical models are different combinations of static friction phenomena: Coulomb Friction, Static Friction (Stiction), Viscous Friction, Stribeck Friction (Stribeck effect). Coulomb friction is independent of magnitude of the velocity and discontinuous at zero velocity (Olsson et al., 1998). Fig. 1.3 shows the Coulomb friction torque-angular velocity relationship $T_{efc} - \beta$. Static friction describes the friction force at rest. Morin (1833) introduced the idea of a friction force at rest that is higher than the Coulomb friction level. Static friction counteracts external forces below a certain level and thus keeps an object from moving. Fig. 1.4 shows the static friction

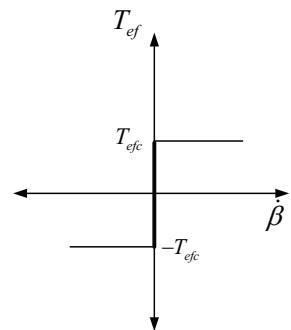


Figure 1.3: Model of Coulomb friction torque

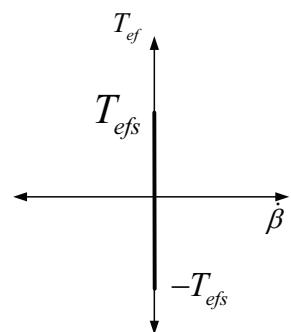


Figure 1.4: Model of static friction torque

model where T_{ef_s} is the static friction torque, T_{ef} is the friction torque, and $\dot{\beta}(t)$ is the angular velocity. Viscous friction is linearly proportional to the velocity (Armstrong-Hélouvy, 1991). Fig. 1.5 shows the viscous friction model where K_{ef} is the viscous friction coefficient, T_{ef} is the friction torque, and $\dot{\beta}(t)$ is the angular velocity. Fig. 1.6 shows a model that combines

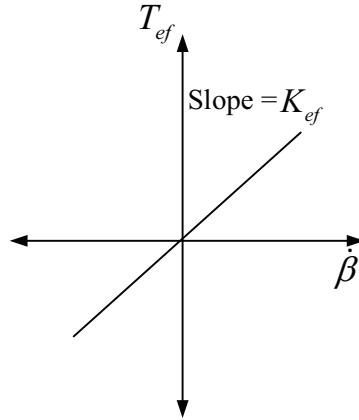


Figure 1.5: Model of viscous friction torque

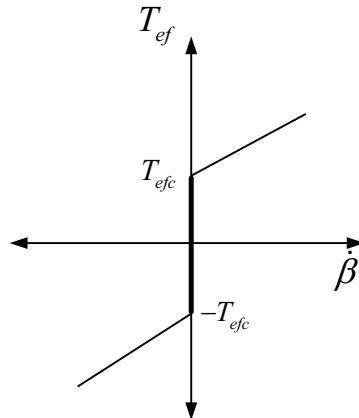


Figure 1.6: Model of Coulomb plus viscous friction torques

Coulomb and Viscous friction torque.

Fig. 1.7 shows a model that combines Coulomb, Static and Viscous friction torques. Stribeck (Stribeck, 1902) effect arises from the use of lubrication which demonstrates decreasing friction with increasing velocity at a lower range of velocity (Armstrong-Hélouvy, 1991). Fig. 1.8 shows a

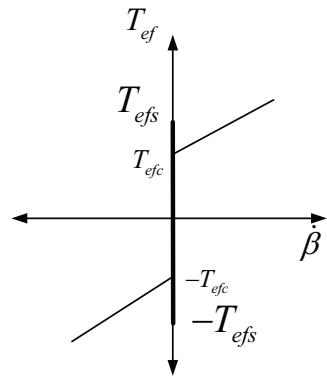


Figure 1.7: Model of Coulomb plus static plus viscous friction torques

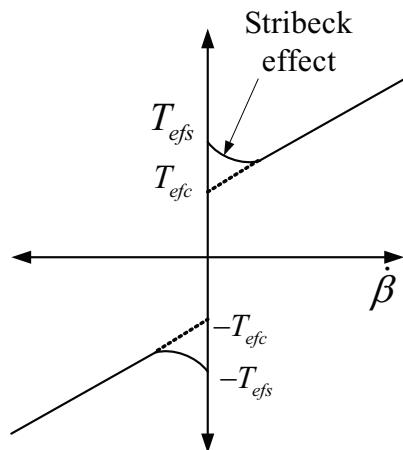


Figure 1.8: Model of Coulomb plus static plus viscous plus Stribeck friction torques

model that combines Coulomb, Static, Viscous and Stribeck friction torques. In this work, we used this static model.

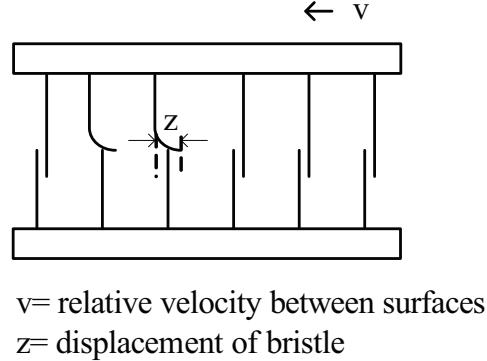


Figure 1.9: Friction interface between two surfaces is thought of as a contact between bristles

There are other friction models that capture the dynamic characteristics of friction. One of such models consider the behavior of the microscopical contact points between two surfaces. This is known as Bristle model introduced by Haessig and Friedland (1991). At microscopic level surfaces are very irregular. Thus two surfaces make contact through elastic bristles. This is viewed as two rigid bodies contacting through elastic bristles, as shown in Fig. 1.9. When the tangential force is applied, the bristles will deflect and generate the elastic friction force. For sufficiently large force, some of the bristles deflect too much which result in the slip of surface.

Canudas de Wit et al. (1995) proposed a dynamic model based on bristle model. The model supports hysteresis behavior due to frictional lag, spring-like behavior in stiction and gives a varying break-away force depending on the rate of change of the applied force. The model also considers the low velocity friction. All these phenomena are unified into a first order non-linear differential equation. Although this model looks attractive because of its dynamic nature, the parameter estimation requires several experiments. To keep the gimbaled laser target system model simpler we used the static model.

1.3.4 Control Methods for Gimbaled Laser Target System

In this section, we review the control methods previously used in pointing and tracking system like gimbaled laser target system.

Proportional-Integral (PI) and Proportional-Integral-Derivative (PID) control methods are the popular methods to control gimbaled platform. Hilkert discussed about inertially stabilized platforms which are used to stabilize and point a broad array of sensors, cameras, telescope, and weapon system (Hilkert, 2008). The platforms utilized two-axis gimbals. To control the platform, Proportional-Integral (PI) control technique was used. Masten (2008) discussed the classical control methods Proportional (P), Proportional-Integral (PI), and Proportional-Integral-Derivative (PID) control for inertially stabilized platform for optical imaging systems with the application of tracking dynamic targets with mobile sensors. He showed that PID controllers provide the best low frequency disturbance rejection performance among P, PI, and PID controllers. At high frequency, the three controllers show similar disturbance rejection performances.

Earlier Masten (1985) suggested that the complete knowledge of the states of the system may allow the use of modern control techniques. Stochastic optimal control is a family of control methods that utilizes the knowledge of the states of the system and disturbances. Lee and Yoo (2008) designed a lead PI controller and a Linear Quadratic Gaussian (LQG) controller with Loop Transfer Recovery for a two-axis gimbaled stabilized system.

Won (1994) developed Minimal Cost Variance (MCV) control, and applied it on a seismic structure (Chung et al., 1988; Dyke et al., 1996). Also he (2001) used this control scheme in satellite attitude control system. MCV control has not yet been used in two-axis gimbaled laser target system.

In this thesis, we used the statistical controllers LQG and MCV. PID control was used as a basis for comparison. Next, we shall review these three control methods: PID control, LQG control, and MCV control.

PID Control

Proportional-Integral-Derivative (PID) control is a classical control method. It is the mostly used feedback control method in which the controller minimizes error between the process variable and the desired set point. The most important development of PID took place in 1930 to 1950 from contributions of Black, Nyquist, Bode, Nichols, and Evans (Bennett, 1993). As the name implies, PID control comprises of three terms: Proportional, Integral, and Derivative. The controller has an input of error signal, $e(t)$ and the output of control signal, $u(t)$ for the plant. One form of the controller is as follows (Nise, 2004):

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t), \quad (1.1)$$

where K_p , K_i , and K_d are proportional, integral, and derivative terms. By tuning these parameters, improvement of certain system properties can be achieved. The system properties are: steady state response such as steady state error; transient response such as rise time, settling time, and overshoot; robustness such as gain and phase margins, and disturbance response.

There are several controller choices in classical control approach. This includes proportional (P) controller, proportional-integral (PI) controller and proportional-integral-derivative (PID) controller. P is easy to design while PI and PID are harder. In terms of command responsiveness PID shows oscillatory behavior, P and PI shows acceptable command responsiveness. However, PID has the best low-frequency disturbance rejection. Generally, PI or PID controller design techniques are used for stabilization loop of the gimbaled laser target system (Hilkert, 2008; Masten, 2008).

LQG Control

Linear Quadratic Gaussian control is a type of optimal control method (Dorato et al., 2000). In this method, the mean of a quadratic cost function is minimized. The plant and the controller are both assumed to be linear. The assumption of the linear plant provides several advantages. In many engineering problem, the plant is assumed to be linear. LQG provides a linear control law. The linear control law is easier to implement physically than the non-linear control law.

Also nearly all linear optimal control problems have computable solutions. In case of non-linear system, it is possible to use the linear optimal control results on a small signal basis. Furthermore, it provides unified framework for designing controller for the previously studied classical control problems.

In 1960, Kalman (1960) introduced the notion of controllability and observability, along with the formulation of linear quadratic regulator problem and linear quadratic Gaussian problem. The controllability is defined as the possibility of transferring any initial state to the desired terminal state by a suitable control action. It includes the necessary and sufficient condition for controllability in terms of the positive definiteness of the Controllability Grammian, and the fact that the linear time-invariant system with n states,

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1.2)$$

is controllable if and only if the matrix $[B, AB, \dots, A^{n-1}B]$ has rank n . Here $x(t)$ are state variables, and $u(t)$ are control inputs. A and B are state and input matrices. Therefore, the linear optimal control design starts with the time-invariant state-space model of the plant. Then a cost function is defined considering one of the cases: finite horizon or infinite horizon cases. The quadratic cost function is a function of state variables and control inputs. For finite horizon case, there is a terminal condition. For infinite horizon case the terminal condition is assumed to be zero. Now in the cost functions, there are two constant weighting matrices that must be selected. For infinite horizon case, it requires to solve algebraic Riccati equation. For finite horizon case, it requires to solve a differential Riccati equation with terminal conditions. Kalman si the first to introduce the Riccati Differential Equation as an algorithm for computing the state feedback gain of the optimal controller. The system was assumed to be linear with a quadratic performance criterion (Athans, 1971).

Up to this point, we consider a system without any noise. The uncertainty of process and sensor introduces error in design process. Also it is difficult to model uncertainty. To model the uncertainty, a probabilistic approach can be used. From a mathematical point of view, a continuous time white noise process is a Gaussian one.

In 1963, Gunckel and Franklin (1963) used Kalman-Bucy filter to design state-estimate feedback systems that minimized the expected value of a quadratic cost function of a sampled data systems. The linear quadratic problem with noisy output observation was reduced to the solution of two decoupled Riccati equations. This is achieved through a separation theorem which basically divides the the problem into the problem of optimal control with state feedback and the problem of optimal state estimation. Since the stochastic linear control problem was based on the assumption that all the stochastic disturbance signals were Gaussian, it became popularly known as linear quadratic Gaussian problem. However, this control method possesses poor robustness properties and requires excessive bandwidths (Athans, 1971; Rosenbrock and McMorran, 1971).

MCV Control

In some applications, it is required that the mean and variance are kept at a specified level. One example can be a structure under seismic disturbances. MCV offers an attractive solution for such requirement. One of the major challenges in the gimbaled laser target system is to keep the variation of state variables due to noise at a minimum to achieve better accuracy. However, MCV is a potential control method for this type of pointing and tracking configuration. Minimal Cost Variance control is a type of statistical control. In statistical control method, the cost function is viewed as a random variable. The objective of this stochastic optimal control method is to optimize the cumulants of the cost function. The first control method in this family is LQG control which minimizes the first cumulant, i.e. mean of the cost function. Then comes the idea of minimizing the second cumulant i.e. variance of the cost function keeping the first cumulant constant. Eventually the idea extends to k -th Cost Cumulant Control which minimizes a linear combination of first k cumulants.

Starting in 1965, Sain (1965) first examined minimal cost variance for the open-loop situation in a dissertation at the University of Illinois. In 1971, Sain and Liberty (1971) considered minimizing the performance variance with a pre-constrained mean for an open loop system. In that paper, new mathematical representations were devised, and for the first time analytical

procedures were used to produce and display the cost densities associated with such control laws, as well as their effects upon the state and control variables of the systems. Liberty continued to investigate characteristic functions of integral quadratic forms, further developing MCV control idea. He with Hartwig (1976), developed methods of generating cumulants in the time domain. In 1994, Won (1994) solved the full-state-feedback minimum-cost-variance (MCV) problem for the closed-loop case. MCV problem formulation is carried out for a class of non-linear stochastic differential equations, associated with an appropriate class of non-quadratic performance indices. A Hamilton-Jacobi framework is adopted. The solution gave rise to a MCV control law that is composed of two matrix functions that satisfy a system of two coupled Riccati equations. The two Riccati equations are coupled by a Lagrange parameter called γ which can be considered as a design parameter. This statistical control method is also known as cost cumulant control or cost distribution shaping method. In LQG control, only one cumulant i.e. the first cumulant is minimized, but in MCV control, there are two cumulants of interest: mean and variance of the cost function. MCV minimizes the second cumulant keeping the first cumulant at a specified level. Therefore, MCV control offers one extra design freedom at the designer's disposal to shape the cost distribution. In 2004, Pham generalized the MCV control with his kCC (k -th Cost Cumulant) control which minimizes a linear combination of first k cumulants (2004). He explored the cumulants of finite-horizon integral quadratic cost associated with a linear stochastic system. The cumulants were obtained by the generalized Karhunen-Loeve expansion (Liberty and Hartwig, 1976). Pham (2004) showed that if these cost cumulants resulted from the series expansion, are further expressed in terms of the iterated-integral kernels then the numerical values of these cost cumulants of any order can be calculated exactly by solving the corresponding set of backward-in-time differential equations.

Won (2002) showed the simulation results that it is possible to improve the efficiency of the modern satellite attitude control by using MCV control method. He showed MCV performs better than LQG for satellite attitude control application. MCV has not been yet applied for a gimbaled laser target system hardware. For the gimbaled laser target system, minimizing the error variation an important requirement. We expect that MCV (minimal variance) will be able

to reduce pointing error variation in gimbaled laser target system compared to LQG.

1.4 Contributions

The major contributions of this thesis are as follows:

- A non-linear mathematical model of a gimbaled laser target system was developed. The system uses two-axis gimbal model that characterizes the gimbal dynamics and its associated disturbances (Ekstrand, 2001; Kennedy and Kennedy, 2003). For the brushless DC motor and servo drive, the models were selected and approximated to a lower order model (Ramu, 2009). To model our image-based position sensor, we utilized the image Jacobian or interaction matrix concept used in visual servo control literature (Hutchinson et al., 1996). Then all these components were put together and simplified to obtain a set of non-linear differential equations for the two-axis gimbaled laser target system.
- The non-linear model was linearized around the hardware operating points, and then modified taking disturbance into account for the optimal control. The disturbance was modeled as Gaussian white noise.
- The statistical control methods – LQG and MCV–were adopted as the control schemes. PID control was used as a basis for comparison. The mean and the standard deviation of pointing error were measured to evaluate the pointing performance of each of the controllers.
- Both full-state feedback and output feedback cases were explored for LQG and MCV in computer simulations. The output feedback case uses Kalman filter for state estimation. Since the laser target system hardware was not equipped to measure all the states, the implementation of LQG and MCV utilized the simulated Kalman filter to estimate the states, and control the system.
- PID, LQG, and MCV controllers were implemented in an off-the-shelf dSPACE controller board in a real time environment. This marked the first time that the MCV controller was

implemented in dSPACE board.

- The mean and the standard deviation of pointing error of PID, LQG, and MCV controllers were calculated from the dSPACE board measurement. The experimental results showed that the LQG method decreased the mean pointing error by 46.28% compared to the PID method. The LQG method reduced the standard deviation of pointing error by 47.85% compared to the PID method. The MCV method reduced the standard deviation of the pointing error by 53.09% compared to the LQG method.

1.5 Thesis Organization

This document is composed of seven chapters, this being the first. Chapter 2 describes of the main subsystems of the gimbaled laser target system. First we present an overview of the architecture. Then the non-linear model of the each subsystem is described. The major subsystems are yaw and pitch gimbal, brushless DC motor, servo drive, and computer vision based position sensor. We linearize brushless DC motors, servo drives, and position sensor blocks. Then we integrate all the subsystem and derive a non-linear mathematical model of the system.

Chapter 3 discusses the assumptions and approximations used to linearize the non-linear model of the system. We got rid of the non-linear coupling terms, and derive state-space models for the gimbaled laser target system.

Chapter 4 describes the theory of PID, LQG, and MCV control methodology. We discuss full-state feedback LQG, full-state feedback MCV, output feedback LQG, and output feedback MCV. We use Kalman filter to estimate states for the output feedback case.

Chapter 5 provides the simulation results for the linear gimbaled laser target system. First we simulate PID controllers with and without white noise. We tune the PID parameter with the best noise rejection capacity. Then we simulate LQG controller. We show LQG is a special case of MCV. Next, we simulate MCV controllers and observe the effect of MCV parameter γ on the pointing performance. We explored both full-state feedback and output feedback cases.

Chapter 6 describes the hardware implementation of the simulated controllers. We imple-

ment open loop position controller and PID controller in hardware. Then we compared those two systems with the simulations. The comparison serves as model verification. After that we build PID, LQG, and MCV controllers. We perform experiments with these controllers under random disturbance. Then we compare the experimental results for PID, LQG, and MCV controllers.

Finally, Chapter 7 provides conclusions with a summary of the major results and discussions on the future work.

CHAPTER 2

NON-LINEAR MATHEMATICAL MODEL OF GIMBALED LASER TARGET SYSTEM

2.1 Overview

In this section, we develop a non-linear model of a two-axis gimbaled laser target system. First, we describe a high-level architecture of the system from the hardware set up. Then, we model different subsystems of the system and simplify them. Finally, we present a model of the gimbaled laser target system as a set of non-linear ordinary differential equations and in state-space form.

This system includes a two-axis gimbal. In a two-axis gimbal, there are an outer gimbal, and an inner gimbal. In this thesis, the outer gimbal will be alternatively termed as yaw gimbal, or azimuth gimbal. The inner gimbal will be termed as pitch gimbal or elevation gimbal. The gimbals are driven by two brushless dc motor-servo drive setup. A laser is mounted on the pitch gimbal, coupled to the line-of-sight (LOS). The laser pointer shoots laser on the screen of the position sensor. The position sensor outputs the position of the laser dot to the controller. The controller acts on the position difference, and generates a reference speed for the servo drive. The servo drive moves the gimbal with the help of a motor. This is how the gimbaled laser target system works. Fig. 2.1 shows a hardware block diagram of a gimbaled laser target system. The objective of the control scheme is to keep the laser dot at the center of the screen of the position sensor, despite disturbances. From the hardware block diagram shown in Fig. 2.1, a conceptual block diagram of the gimbaled laser target system is drawn as in Fig. 2.2.

In Fig. 2.2, The system consists of both analog and digital subsystems. Here t indicates an analog signal while k indicates a digital signal. The system inputs are desired horizontal and vertical (x, y) co-ordinates on the position sensor screen denoted by (f_x^*, f_y^*) . The system

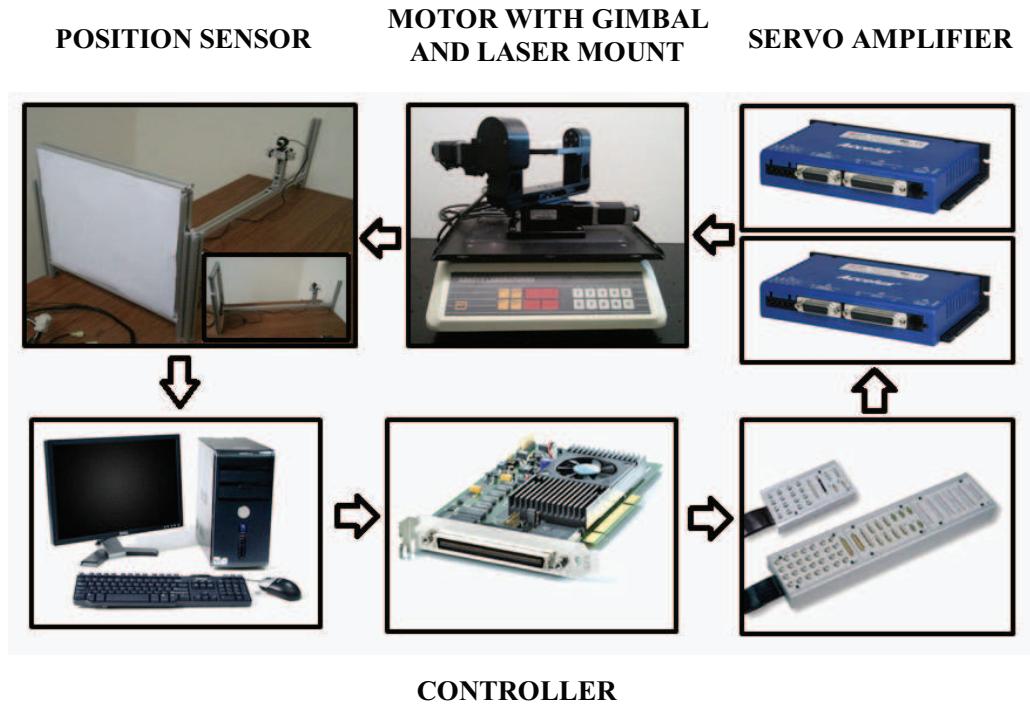


Figure 2.1: Hardware implementation of a gimbaled laser target system

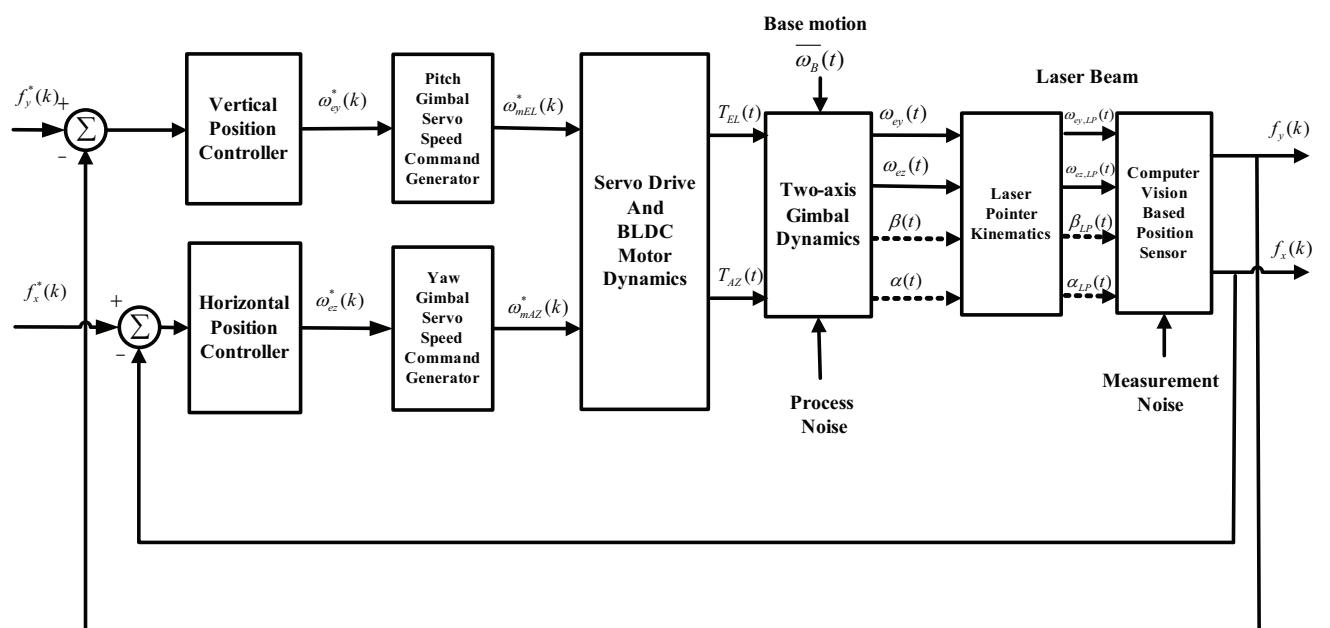


Figure 2.2: Architecture of the control system of a two-axis gimbaled laser target system

(f_x^*, f_y^*)	Desired horizontal and vertical position on white screen derived from corresponding image features of laser dot,
f_x, f_y	Measured horizontal and vertical position on white screen,
$\omega_{ez}^*, \omega_{ey}^*$	Reference yaw and pitch angular rate for gimbal,
$\omega_{mAZ}^*, \omega_{mEL}^*$	Reference yaw and pitch duty cycle command,
T_{AZ}, T_{EL}	Yaw and pitch control torque supplied by motor,
ω_{ez}, ω_{ey}	Yaw and pitch angular rate of gimbal,
α, β	Euler or LOS angles for yaw and gimbal,
$\omega_{ezLP}, \omega_{eyLP}$	Yaw and pitch angular rate of laser pointer,
α_{LP}, β_{LP}	Euler angles for laser pointer.

output co-ordinates are denoted by (f_x, f_y) . The plant model includes servo drives, brushless dc motors, and two-axis gimbal dynamics. There is an image-based position sensor in the feedback loop. The webcam inside the position sensor estimates the position of the laser dot on the screen in terms of (x, y) co-ordinates. These estimated co-ordinates are sent to a summing junction for the comparison with desired (x, y) co-ordinates. The position controller receives the error signal (position co-ordinate (x, y) difference). Then it generates a reference angular rate signal for the gimbal. However, before the signal propagates to the gimbal, it passes through servo drive and brushless dc motor. Therefore, the reference angular rate signal for the gimbal is converted into required reference motor speed signal $(\omega_{mEL}^*, \omega_{mAZ}^*)$ for the motor . A speed command generator is used for the conversion. This reference speed signal goes through the servo drive, and the brushless dc motor electronics and is changed to required torque to move the gimbal. This is a brief overview of the gimbaled laser target system architecture. Next we shall describe its subsystems.

2.2 Subsystem Description

2.2.1 Gimbal

Reference frames and Notations

We consider a two-axis gimbal configuration, which consists of an outer gimbal and an inner gimbal. The outer gimbal is called yaw or azimuth gimbal. The inner gimbal is called pitch or elevation gimbal. In this section, following topics will be covered:

- Three reference frames to express the parameters such as the angle, the angular rates, the angular accelerations will be introduced.
- The transformation matrices between the frames will be introduced. These will be used to express a parameter defined in one frame in terms of another frame.
- The relationship among the angular rates defined in different frames will be expressed.
- The inertia tensors will be described.

We introduce three reference frames or co-ordinate frames to define the gimbal motion. The outer gimbal or yaw gimbal is constructed on top of the gimbal base or platform. The inner gimbal or pitch gimbal is constructed inside the outer gimbal or yaw gimbal. There is one reference frame is defined each for the base, the pitch gimbal, and the yaw gimbal. They are denoted by b , a , e in Fig. 2.3. There are three different origins O_b , O_a , O_e are shown for three frames. We assume that the center of rotation is the same point. The center of the rotation is O , which is shown later in Fig. 2.4 and Fig. 2.5. We denote x , y , z components of three frames as follows:

$$\text{Frame } b : \begin{bmatrix} x_b & y_b & z_b \end{bmatrix}^T, \quad (2.1)$$

$$\text{Frame } a : \begin{bmatrix} x_a & y_a & z_a \end{bmatrix}^T, \quad (2.2)$$

$$\text{Frame } e : \begin{bmatrix} x_e & y_e & z_e \end{bmatrix}^T. \quad (2.3)$$

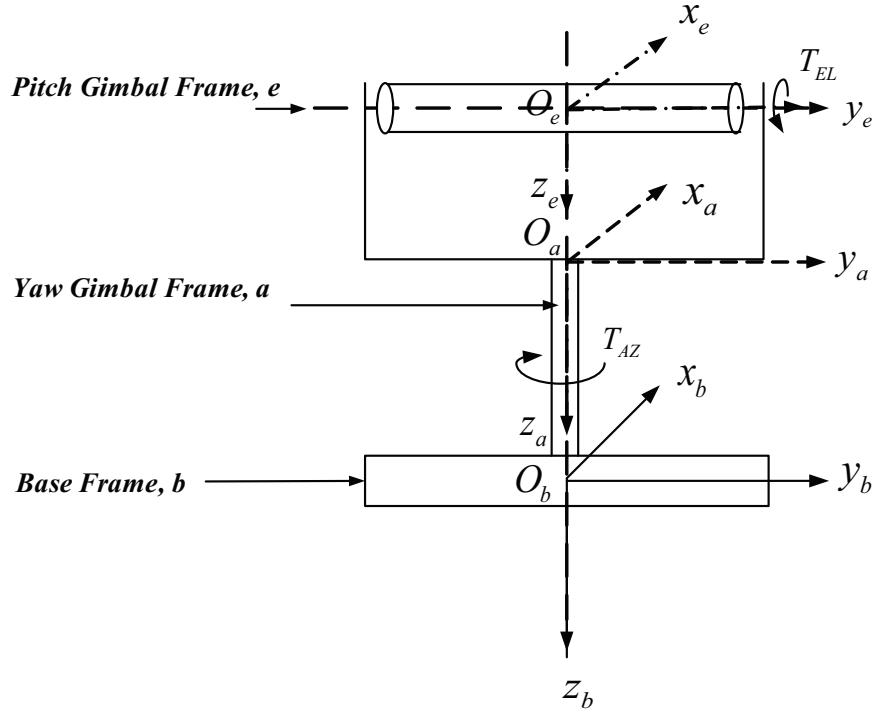


Figure 2.3: Two-axis gimbal system coordinates

Frame b is fixed to the gimbal base or platform. The gimbal base does not move even if yaw and pitch gimbal rotate. In Fig. 2.3, the origin O_b is the center of mass of the base, x_b is along the width of the base and the positive direction is the direction to which laser system is pointed. The y_b is along the length of the base and perpendicular to x_b . According to right hand rule, z_b points downwards.

Frame a is fixed to the yaw gimbal where origin O_a is the center of mass of yaw gimbal. The x_ay_a plane can rotate with reference to the z_a axis which is the “yaw axis”. The z_a coincides with z_b . Yaw gimbal is the outer gimbal frame. Therefore, when the yaw gimbal rotates, the inner gimbal frame or, the pitch gimbal also rotates. The gimbal base remains stationary.

Frame e is fixed to the pitch gimbal with the coordinates where the origin O_e is the center of mass of pitch gimbal. x_e coincides with the laser system optical axis. The laser system optical axis means the direction at which the laser is pointed to the screen y_e coincides with y_a . y_e is the axis of rotation for frame e or “pitch axis”. The x_ey_e plane can rotate with reference to y_e axis. z_e can be found using the right hand rule. When the pitch gimbal rotates, neither the

yaw gimbal nor the gimbal base rotates. We assume that the gimbals are rigid bodies. Then

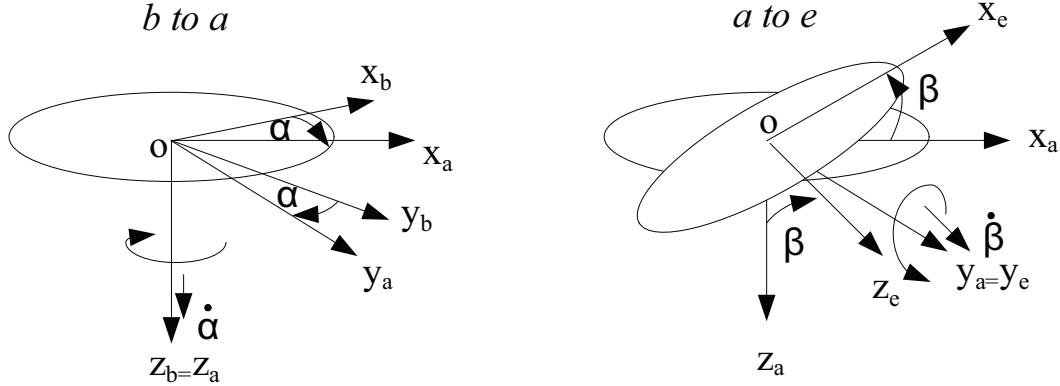


Figure 2.4: Rotation of frames

instead of the three separate origins O_b , O_a , O_e , we can consider the same origin O as the origin three frames . All the frames defined is fixed to the body of base, yaw, and pitch gimbal. A vector expressed in one frame will not have the same form in other frame. Because of the gimbal configuration, there is a relation or transformation matrix between these frames. We can find the transformation matrices by rotating one frame by some angle to coincide with other frame (Etkin and Reid, 1996). The rotation is done as follows:

- The base-fixed frame b is transferred to the yaw gimbal frame a by a positive angle of rotation $\alpha(t)$ about the z_b -axis (see Fig. 2.4). $\alpha(t)$ is called the yaw LOS or euler angle.

The transformation matrix associated with this rotation is:

$$R_{ab}(\alpha) = \begin{bmatrix} \cos \alpha(t) & \sin \alpha(t) & 0 \\ -\sin \alpha(t) & \cos \alpha(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.4)$$

where $R_{ab}(\alpha)$ is the transformation matrix from b to a .

- The yaw gimbal frame a is transferred to the pitch gimbal frame e by a positive angle of rotation $\beta(t)$ about the y_a -axis (see Fig.2.4). $\beta(t)$ is called the pitch LOS angle. The

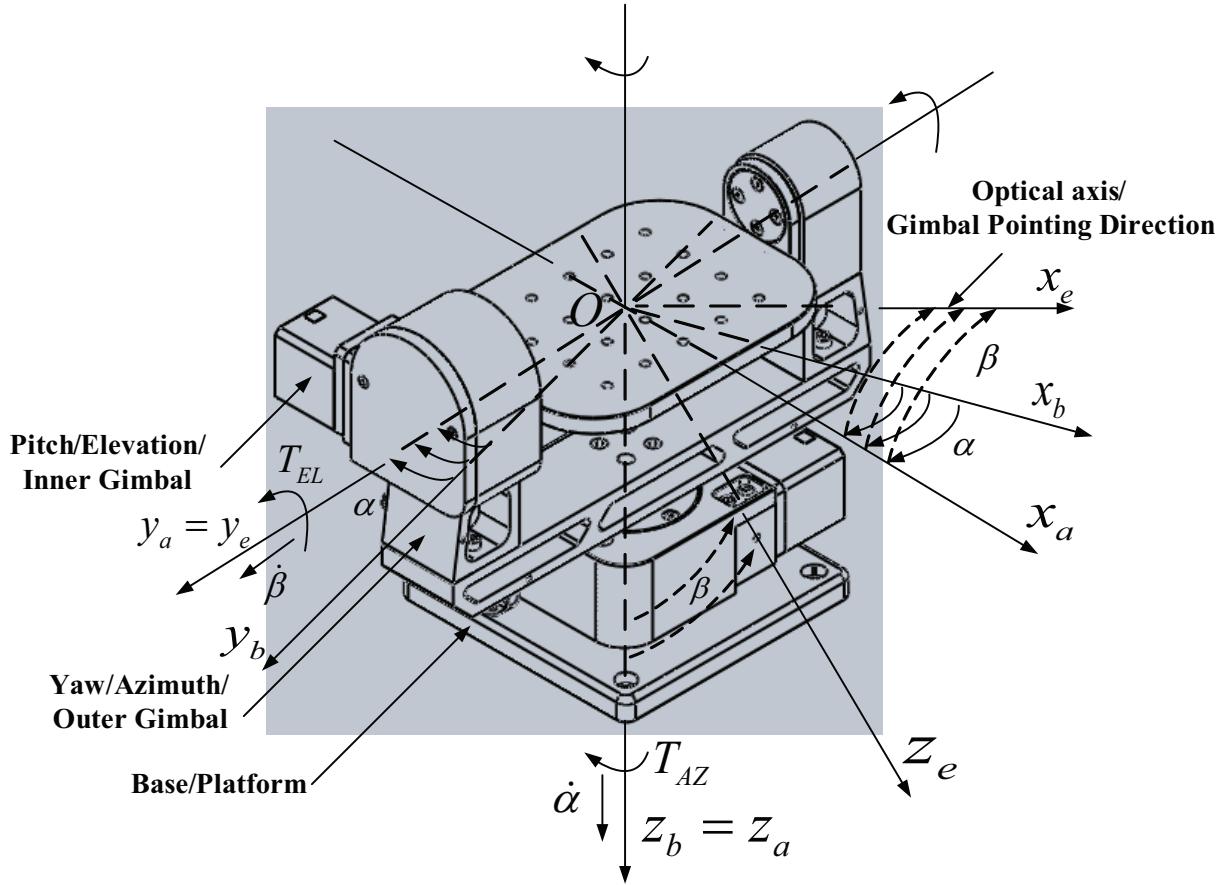


Figure 2.5: Two-axis Newmark GM-6 gimbal with angle rotation among frames

transformation matrix associated with this rotation is:

$$R_{ea}(\beta) = \begin{bmatrix} \cos \beta(t) & 0 & -\sin \beta(t) \\ 0 & 1 & 0 \\ \sin \beta(t) & 0 & \cos \beta(t) \end{bmatrix}. \quad (2.5)$$

The Euler angle and two-axis gimbal attitude are given in Fig. 2.5. For the angular velocities of frame b , a , and e respectively are,

$$\text{Frame } b : \bar{\omega}_b(t) = \begin{bmatrix} \omega_{bx}(t) & \omega_{by}(t) & \omega_{bz}(t) \end{bmatrix}^T, \quad (2.6)$$

$$\text{Frame } a : \bar{\omega}_a(t) = \begin{bmatrix} \omega_{ax}(t) & \omega_{ay}(t) & \omega_{az}(t) \end{bmatrix}^T, \quad (2.7)$$

$$\text{Frame } e : \bar{\omega}_e(t) = \begin{bmatrix} \omega_{ex}(t) & \omega_{ey}(t) & \omega_{ez}(t) \end{bmatrix}^T. \quad (2.8)$$

where $\omega_{bx}(t)$, $\omega_{by}(t)$, $\omega_{bz}(t)$ the components of angular velocity $\bar{\omega}_b(t)$ in frame b , and similarly for the other vectors. In the traditional flight dynamics the notations $\omega_{bx}(t)$, $\omega_{by}(t)$, $\omega_{bz}(t)$ are for the roll, pitch and yaw components respectively.

The yaw gimbal angular velocity vector is related to the base angular velocity vector by Euler angle relationship (Ekstrand, 2001) as follows,

$$\bar{\omega}_a(t) = R_{ab}(\alpha)\bar{\omega}_b(t) + \begin{bmatrix} 0 & 0 & \dot{\alpha}(t) \end{bmatrix}^T, \quad (2.9)$$

where $\dot{\alpha}(t) = \omega_{az}(t) - \omega_{bz}(t)$. Thus using (2.4) and (2.8) we write as follows,

$$\begin{bmatrix} \omega_{ax}(t) \\ \omega_{ay}(t) \\ \omega_{az}(t) \end{bmatrix} = \begin{bmatrix} \cos \alpha(t) & \sin \alpha(t) & 0 \\ -\sin \alpha(t) & \cos \alpha(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_{bx}(t) \\ \omega_{by}(t) \\ \omega_{bz}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha}(t) \end{bmatrix}, \quad (2.10)$$

$$= \begin{bmatrix} \omega_{bx}(t) \cos \alpha(t) + \omega_{by}(t) \sin \alpha(t) \\ -\omega_{bx}(t) \sin \alpha(t) + \omega_{by}(t) \cos \alpha(t) \\ \omega_{bz}(t) + \dot{\alpha}(t) \end{bmatrix}. \quad (2.11)$$

Similarly, the pitch gimbal angular velocity vector is related to the yaw gimbal angular velocity vector by (Ekstrand, 2001),

$$\bar{\omega}_e(t) = R_{ea}(\beta)\bar{\omega}_a(t) + \begin{bmatrix} 0 & \dot{\beta}(t) & 0 \end{bmatrix}^T, \quad (2.12)$$

$$= R_{ea}(\beta)\bar{\omega}_a(t) + \hat{\dot{\beta}}(t), \quad (2.13)$$

where $\dot{\beta}(t) = \omega_{ey}(t) - \omega_{ay}(t)$ and $\hat{\dot{\beta}}(t) = \begin{bmatrix} 0 & \dot{\beta}(t) & 0 \end{bmatrix}^T$. Thus using (2.5) and (2.8),

$$\begin{bmatrix} \omega_{ex}(t) \\ \omega_{ey}(t) \\ \omega_{ez}(t) \end{bmatrix} = \begin{bmatrix} \cos \beta(t) & 0 & -\sin \beta(t) \\ 0 & 1 & 0 \\ \sin \beta(t) & 0 & \cos \beta(t) \end{bmatrix} \begin{bmatrix} \omega_{ax}(t) \\ \omega_{ay}(t) \\ \omega_{az}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\beta}(t) \\ 0 \end{bmatrix},$$

$$= \begin{bmatrix} \omega_{ax}(t) \cos \beta(t) - \omega_{az}(t) \sin \beta(t) \\ \omega_{ay}(t) + \dot{\beta}(t) \\ \omega_{ax}(t) \sin \beta(t) + \omega_{az}(t) \cos \beta(t) \end{bmatrix}. \quad (2.14)$$

The pitch gimbal angular velocity vector is related to the base angular velocity vector by Euler angular relationship (Ekstrand, 2001). Thus using (2.4), (2.5) and (2.8) we write,

$$\bar{\omega}_e(t) = R_{ea}(\beta)R_{ab}(\alpha)\bar{\omega}_b(t) + R_{ea}(\beta) \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\beta}(t) \\ 0 \end{bmatrix}, \quad (2.15)$$

$$\begin{aligned} \begin{bmatrix} \omega_{ex}(t) \\ \omega_{ey}(t) \\ \omega_{ez}(t) \end{bmatrix} &= \begin{bmatrix} \cos \beta(t) & 0 & -\sin \beta(t) \\ 0 & 1 & 0 \\ \sin \beta(t) & 0 & \cos \beta(t) \end{bmatrix} \begin{bmatrix} \cos \alpha(t) & \sin \alpha(t) & 0 \\ -\sin \alpha(t) & \cos \alpha(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_{bx}(t) \\ \omega_{by}(t) \\ \omega_{bz}(t) \end{bmatrix} \\ &+ \begin{bmatrix} \cos \beta(t) & 0 & -\sin \beta(t) \\ 0 & 1 & 0 \\ \sin \beta(t) & 0 & \cos \beta(t) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\beta}(t) \\ 0 \end{bmatrix}, \\ &= \begin{bmatrix} \omega_{bx}(t) \cos \alpha(t) \cos \beta(t) + \omega_{by}(t) \sin \alpha(t) \cos \beta(t) - \omega_{bz}(t) \sin \beta(t) - \dot{\alpha}(t) \sin \beta(t) \\ -\omega_{bx}(t) \sin \alpha(t) + \omega_{by}(t) \cos \alpha(t) + \dot{\beta}(t) \\ \omega_{bx}(t) \cos \alpha(t) \sin \beta(t) + \omega_{by}(t) \sin \alpha(t) \sin \beta(t) - \omega_{bz}(t) \cos \beta(t) - \dot{\alpha}(t) \cos \beta(t) \end{bmatrix}. \end{aligned} \quad (2.16)$$

The relationship between angular acceleration of the pitch and yaw gimbal coordinate frame is obtained by differentiating (2.12),

$$\dot{\omega}_e(t) = R_{ea}(\beta)(t)\dot{\omega}_a(t) + Q_{ea}(\beta)\dot{\omega}_a(t)\dot{\beta}(t) + \begin{bmatrix} 0 & \ddot{\beta}(t) & 0 \end{bmatrix}^T, \quad (2.17)$$

$$= R_{ea}(\beta)(t)\dot{\omega}_a(t) + Q_{ea}(\beta)\dot{\omega}_a(t)\dot{\beta}(t) + \hat{\dot{\beta}}(t), \quad (2.18)$$

where

$$Q_{ea}(\beta) = \begin{bmatrix} -\sin \beta(t) & 0 & -\cos \beta(t) \\ 0 & 0 & 0 \\ \cos \beta(t) & 0 & -\sin \beta(t) \end{bmatrix}, \quad (2.19)$$

and

$$\hat{\dot{\beta}}(t) = \begin{bmatrix} 0 & \ddot{\beta}(t) & 0 \end{bmatrix}^T. \quad (2.20)$$

The inertia tensors of the yaw and pitch gimbals are defined with respect to the yaw gimbal (a) and pitch gimbal fixed coordinate systems (e) respectively,

$$J_a = \begin{bmatrix} J_{ax} & d_{xy} & d_{xz} \\ d_{xy} & J_{ay} & d_{yz} \\ d_{xz} & d_{yz} & J_{az} \end{bmatrix}, \quad (2.21)$$

$$J_e = \begin{bmatrix} J_{ex} & D_{xy} & D_{xz} \\ D_{xy} & J_{ey} & D_{yz} \\ D_{xz} & D_{yz} & J_{ez} \end{bmatrix}. \quad (2.22)$$

Here the moment of inertia are denoted by J and products of inertia by D and d (with appropriate indices). Note that the inertia tensor is a symmetric matrix. Whenever the principle axis is not aligned with the rotational axis, there exists the product of inertia terms. For the sign of the products of inertia, we follow the definition in (Goldstein, 1980); For example, for the elevation axis gimbal the elements of the inertia tensor is defined as:

$$J_{ex} = \sum m_e(x_e^2 + y_e^2), \quad (2.23)$$

$$J_{ey} = \sum m_e(y_e^2 + z_e^2), \quad (2.24)$$

$$J_{ez} = \sum m_e(z_e^2 + x_e^2), \quad (2.25)$$

$$D_{xy} = -\sum m_e x_e y_e, \quad (2.26)$$

$$D_{yz} = -\sum m_e y_e z_e, \quad (2.27)$$

$$D_{zx} = -\sum m_e z_e x_e. \quad (2.28)$$

where m_e is the mass of an arbitrary point on the elevation gimbal. In this thesis, no negative signs are explicitly used in the matrices following the convention in (Goldstein, 1980).

The angular moment of a rigid body can be defined as the product of body's inertia tensor with respect to the body-fixed frame and the instantaneous angular velocity. If the yaw and pitch gimbals are considered as rigid bodies then the angular momentums for the yaw and pitch

gimbals are respectively,

$$\bar{H}_a(t) = J_a \bar{\omega}_a(t), \quad (2.29)$$

$$\bar{H}_e(t) = J_e \bar{\omega}_e(t). \quad (2.30)$$

Space-fixed Frame and Body-fixed Frame

To derive the gimbal dynamics, we need to use the concept of the space-fixed frame and the body-fixed frame. Let us start with some definitions. A rigid body is defined as a system of mass points subject to the constraint that the distances between any pair of points remain constant throughout the motion (Goldstein, 1980). To specify the configuration of a rigid body we need coordinate systems. Let us consider two coordinate systems: one is the space-fixed (inertial) coordinate system and another is the body-fixed coordinate system. The space-fixed (inertial) coordinate system is the system of the observer. This is the coordinate system where Newton's first law of inertia holds (Goldstein, 1980). An infinitesimal rotation is an orthogonal transformation of coordinate axes in which the components of a vector are almost the same in both sets of axes – the change is infinitesimal (Goldstein, 1980).

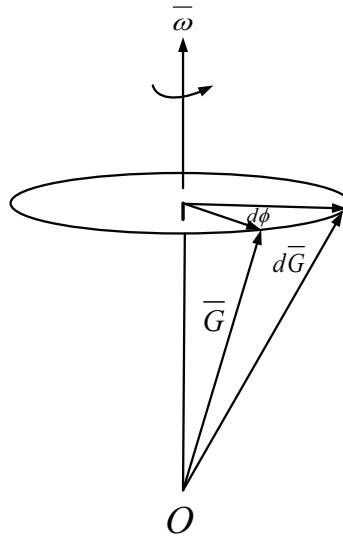


Figure 2.6: On the coordinate transformation for infinitesimal rotation

We consider an arbitrary vector \bar{G} , as shown in Fig. 2.6. \bar{G} will vary in time as the body moves, but the change will depend upon the coordinate system to which the observations are

referred. From an observer on the body-fixed coordinate system the vector \bar{G} appears constant. However, from an observer on a space-fixed coordinate system, the components of the arbitrary vector will change in time if the body is in motion. Considering an infinitesimal rotation of the rigid body, the relation between two differential changes in \bar{G} is described by,

$$(d\bar{G})_{SPACE} = (d\bar{G})_{BODY} + (d\bar{G})_{ROTATIONAL}. \quad (2.31)$$

We assume that both space-fixed and body-fixed coordinate systems have the same origin. If the body-fixed coordinate system rotates counter-clockwise by an angle $d\phi$ in an infinitely small time interval dt , then $(d\bar{G})_{ROTATIONAL}$ is given by (Goldstein, 1980),

$$(d\bar{G})_{ROTATIONAL} = d\phi \times \bar{G}. \quad (2.32)$$

Then we write Eq. (2.31) as,

$$(d\bar{G})_{SPACE} = (d\bar{G})_{BODY} + d\phi \times \bar{G}. \quad (2.33)$$

The time rate of change of \bar{G} can be obtained by dividing Eq. (2.33) by the differential time element dt ;

$$\left(\frac{d\bar{G}}{dt} \right)_{SPACE} = \left(\frac{d\bar{G}}{dt} \right)_{BODY} + \bar{\omega} \times \bar{G}, \quad (2.34)$$

where $\bar{\omega}$ is the instantaneous angular velocity of the body defined by,

$$\bar{\omega}dt = d\phi, \quad (2.35)$$

where $\bar{\omega}$ lies along the axis of infinitesimal rotation occurring between t and $t + dt$. This direction is known as the instantaneous axis of the rotation. In magnitude, it measures the instantaneous rate of rotation of the body.

Eq. (2.34) is applicable for any arbitrary vector. Therefore the following relation is used as an operator, known as rotation operator,

$$\left(\frac{d}{dt} \right)_{SPACE} = \left[\left(\frac{d}{dt} \right)_{BODY} + \bar{\omega} \times \right]. \quad (2.36)$$

Now by Newton's second law, the vector sum of kinematic torques applied to a body is the rate of change of angular momentum with respect to inertial space or space-fixed coordinate system (Rue, 1969, p.818). This can be written as,

$$\bar{L}(t) = \left(\frac{d\bar{H}(t)}{dt} \right)_{SPACE}, \quad (2.37)$$

where

$\bar{L}(t)$ = vector sum of kinematic torques applied to body,

$\bar{H}(t)$ = angular momentum of the rigid body.

We shall derive the gimbal model using Newton's second law.

Pitch Gimbal Model

We assume that the gimbal assembly consists of rigid elements, the pitch gimbal dynamics can be derived from the torque relationships about the inner gimbal body axis by following (2.37). Now we can write,

$$\bar{L}_e(t) = \left(\frac{d\bar{H}_e(t)}{dt} \right)_{SPACE}, \quad (2.38)$$

where $\bar{L}_e(t)$ = vector sum of kinematic torques about the inner (pitch) gimbal.

In two-axis gimbal case, there involves relative motion between frames or co-ordinate systems. Using (2.36) (Goldstein, 1980, pp.172-174) we can say, the total angular momentum change equals the angular momentum change relative to the rotating frame, plus the angular momentum change of the rotating frame fixed to relative to the inertial space frame. If $\bar{\omega}_e(t)$ is the instantaneous angular velocity of inner (pitch) gimbal, $\bar{H}_e(t)$ is the angular momentum of the inner gimbal with respect to coordinate system fixed to the pitch gimbal, then we can write from (2.36) and (2.38),

$$\bar{L}_e(t) = \left(\frac{d\bar{H}_e(t)}{dt} \right)_{SPACE} = \frac{d\bar{H}_e(t)}{dt} + \bar{\omega}_e(t) \times \bar{H}_e(t). \quad (2.39)$$

Now, the angular momentum in a body-fixed coordinate system is a product of inertia tensor and instantaneous angular velocity. Thus $\bar{H}_e(t) = J_e \bar{\omega}_e(t)$ which we use in (2.39), we get

$$\bar{L}_e(t) = J_e \frac{d}{dt} \bar{\omega}_e(t) + \bar{\omega}_e(t) \times J_e \bar{\omega}_e(t). \quad (2.40)$$

Combining (2.38) and (2.40) the expression for the pitch gimbal dynamics can be found. This can be found also in (Kennedy and Kennedy, 2003, p.4) as,

$$J_e \dot{\omega}_e(t) + (\bar{\omega}_e(t) \times J_e \bar{\omega}_e(t)) = \bar{L}_e(t), \quad (2.41)$$

where $\bar{L}_e(t)$

$$\bar{L}_e(t) = \begin{bmatrix} T_{ex}(t) \\ T_{EL}(t) \\ T_{ez}(t) \end{bmatrix} - \begin{bmatrix} T_{Ul_x}(t) \\ T_{Ul_y}(t) \\ T_{Ul_z}(t) \end{bmatrix} - \begin{bmatrix} 0 \\ T_{ef}(t) \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ T_{e\omega}(t) \\ 0 \end{bmatrix}, \quad (2.42)$$

with

$T_{ef}(t)$ = Friction torque,

$T_{e\omega}(t)$ = Cable restraint torque,

$T_{Ul_x}(t), T_{Ul_y}(t), T_{Ul_z}(t)$ = Mass imbalance torque about each gimbal axis,

$T_{ex}(t), T_{ez}(t)$ = Reaction torques exerted by pitch gimbal on yaw gimbal

$T_{EL}(t)$ = Pitch axis control torque.

Now, substituting the pitch gimbal angular rate with (2.12), the inertia tensor with (2.22), we rewrite (2.41) as,

$$\begin{aligned} \bar{L}_e(t) &= \begin{bmatrix} J_{ex} & D_{xy} & D_{xz} \\ D_{xy} & J_{ey} & D_{yz} \\ D_{xz} & D_{yz} & J_{ez} \end{bmatrix} \begin{bmatrix} \dot{\omega}_{ex}(t) \\ \dot{\omega}_{ey}(t) \\ \dot{\omega}_{ez}(t) \end{bmatrix} + \begin{bmatrix} \omega_{ex}(t) \\ \omega_{ey}(t) \\ \omega_{ez}(t) \end{bmatrix} \times \begin{bmatrix} J_{ex} & D_{xy} & D_{xz} \\ D_{xy} & J_{ey} & D_{yz} \\ D_{xz} & D_{yz} & J_{ez} \end{bmatrix} \begin{bmatrix} \omega_{ex}(t) \\ \omega_{ey}(t) \\ \omega_{ez}(t) \end{bmatrix}, \\ &= \begin{bmatrix} J_{ex}\dot{\omega}_{ex} + D_{xy}\dot{\omega}_{ey} + D_{xz}\dot{\omega}_{ez} \\ D_{xy}\dot{\omega}_{ex} + J_{ey}\dot{\omega}_{ey} + D_{yz}\dot{\omega}_{ez} \\ D_{xz}\dot{\omega}_{ex} + D_{yz}\dot{\omega}_{ey} + J_{ez}\dot{\omega}_{ez} \end{bmatrix} + \begin{bmatrix} 0 & -\omega_{ez} & \omega_{ey} \\ \omega_{ez} & 0 & -\omega_{ex} \\ -\omega_{ey} & \omega_{ex} & 0 \end{bmatrix} \begin{bmatrix} J_{ex} & D_{xy} & D_{xz} \\ D_{xy} & J_{ey} & D_{yz} \\ D_{xz} & D_{yz} & J_{ez} \end{bmatrix} \begin{bmatrix} \omega_{ex} \\ \omega_{ey} \\ \omega_{ez} \end{bmatrix}.. \end{aligned} \quad (2.43)$$

Thus from (2.42) and (2.43) we can write three equations from three rows,

$$\begin{aligned} T_{ex}(t) - T_{Ul_x}(t) &= J_{ex}\dot{\omega}_{ex}(t) + \omega_{ey}(t)\omega_{ez}(t)(J_{ez} - J_{ey}) + D_{xy}(\dot{\omega}_{ey}(t) + \omega_{ex}(t)\omega_{ez}(t)) \\ &\quad + D_{yz}(\omega_{ey}^2(t) - \omega_{ez}^2(t)) + D_{xz}(\dot{\omega}_{ez}(t) - \omega_{ex}(t)\omega_{ey}(t)). \end{aligned} \quad (2.44)$$

$$T_{EL}(t) - T_{Uly}(t) - T_{ef}(t) - T_{e\omega}(t) = J_{ey}\dot{\omega}_{ey}(t) + \omega_{ez}(t)\omega_{ex}(t)(J_{ex} - J_{ez}) \\ + D_{xy}(\dot{\omega}_{ex}(t) + \omega_{ey}(t)\omega_{ez}(t)) + D_{yz}(\dot{\omega}_{ez}(t) - \omega_{ex}(t)\omega_{ey}(t)) + D_{xz}(\omega_{ez}^2(t) - \omega_{ex}^2(t)). \quad (2.45)$$

$$T_{ez}(t) - T_{Ulz}(t) = J_{ez}\dot{\omega}_{ez}(t) + \omega_{ex}(t)\omega_{ey}(t)(J_{ey} - J_{ex}) \\ + D_{xy}(\omega_{ex}^2(t) - \omega_{ey}^2(t)) + D_{yz}(\dot{\omega}_{ey}(t) + \omega_{ex}(t)\omega_{ez}(t)) + D_{xz}(\dot{\omega}_{ex}(t) - \omega_{ey}\omega_{ez}(t)). \quad (2.46)$$

Eq. (2.45) expresses the relation between the pitch axis control torque T_{EL} and the angular rate ω_{ey} . We want to control this angular rate ω_{ey} since y co-ordinate of laser dot will depend on this. We call (2.45) the pitch gimbal equation of motion.

Now the last four terms on the right hand side of (2.45) are non-linear terms in the differential equation. We consider these as the disturbance due to the inertia terms (Ekstrand, 2001, p.1086), and replace with $T_D(t)$,

$$T_D(t) = \omega_{ez}(t)\omega_{ex}(t)(J_{ex} - J_{ez}) \\ + D_{xy}(\dot{\omega}_{ex}(t) + \omega_{ey}(t)\omega_{ez}(t)) + D_{yz}(\dot{\omega}_{ez}(t) - \omega_{ex}(t)\omega_{ey}(t)) \\ + D_{xz}(\omega_{ez}^2(t) - \omega_{ex}^2(t)). \quad (2.47)$$

Then (2.45) becomes,

$$T_{EL}(t) - T_{Uly}(t) - T_{ef}(t) - T_{e\omega}(t) = J_{ey}\dot{\omega}_{ey}(t) + T_D(t), \quad (2.48)$$

which can be rearranged as,

$$J_{ey}\dot{\omega}_{ey}(t) = T_{EL}(t) - T_{Uly}(t) - T_{ef}(t) - T_{e\omega}(t) - T_D(t). \quad (2.49)$$

For the relation between the angular rate and LOS angle, we write y component of (2.16),

$$\dot{\beta}(t) = \omega_{ey}(t) - (\omega_{by}(t) \cos \alpha(t) + \omega_{bx}(t) \sin \alpha(t)). \quad (2.50)$$

Now for the pitch gimbal dynamics we have a nonlinear model from (2.49) and (2.50). Eq. (2.49) is the torque-angular rate relationship. Eq. (2.50) is the angular rate-LOS angle

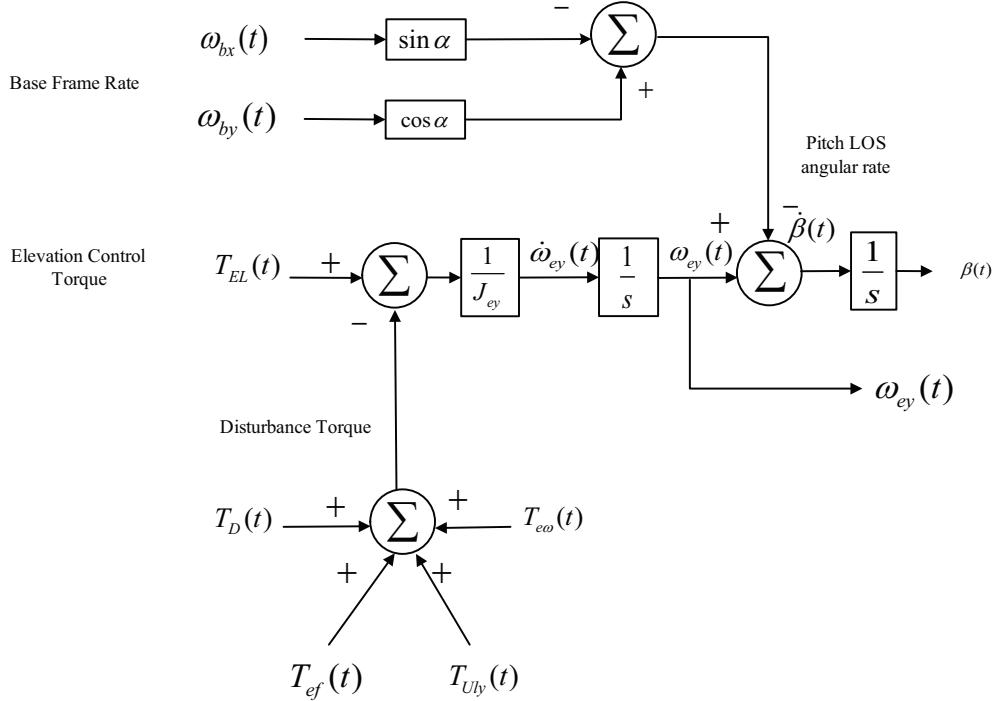


Figure 2.7: Block diagram of pitch gimbal

relationship. In this pitch gimbal dynamic model, the torques $T_{EL}(t)$, $T_{Uly}(t)$, $T_{ef}(t)$, $T_{eo}(t)$, $T_D(t)$ and base angular rates $\omega_{bx}(t)$, $\omega_{by}(t)$ are the inputs and relative pitch angular rate $\dot{\beta}(t)$ is the output as shown in Fig. 2.7. In the next subsections, we shall obtain the expressions for gimbal inertial disturbance torque, friction torque, and cable restraint torque for pitch gimbal. We shall get the pitch gimbal model after incorporating those torques.

Pitch Gimbal Inertial Disturbance Torques

Equation (2.47) is the inertial disturbance torque equation for pitch gimbal as in:

$$T_D = \omega_{ez}\omega_{ex}(J_{ex} - J_{ez}) + D_{xy}(\dot{\omega}_{ex} + \omega_{ey}\omega_{ez}) + D_{yz}(\dot{\omega}_{ez} - \omega_{ex}\omega_{ey}) + D_{xz}(\omega_{ez}^2 - \omega_{ex}^2). \quad (2.51)$$

Equation (2.51) is the expression for disturbance torque due to moment of inertia and product of inertia of pitch gimbal. Equation (2.51) represents a nonlinear system which has inputs: x and z components of pitch frame angular acceleration $\dot{\omega}_{ex}$, $\dot{\omega}_{ez}$, y component of pitch frame angular velocity ω_{ey} , and output of inertial disturbance torque T_D . In the next section, we describe pitch

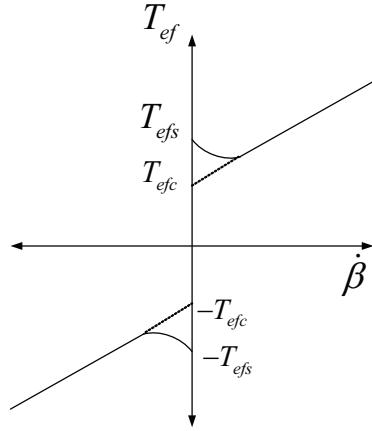


Figure 2.8: Exponential model of nonlinear friction torque

gimbal friction torque.

Pitch Gimbal Friction Torque

For modeling friction, we use the exponential friction model as shown in Fig. 2.8. The static friction torque in the direction of increasing velocity can be expressed (Iwasaki et al., 1999, p.4), using the steady-state velocity

$$T_{efss} = [T_{efc} + (T_{efs} - T_{efc})e^{-\left(\frac{\dot{\beta}(t)}{\dot{\beta}_S}\right)^2}] \operatorname{sgn}(\dot{\beta}(t)) + K_{ef}\dot{\beta}(t), \quad (2.52)$$

where

T_{efc} = Coulomb friction torque,

T_{efs} = Stiction torque,

K_{ef} = Viscous friction coefficient,

$\dot{\beta}_S$ = Stribeck angular velocity,

$\operatorname{sgn}(\dot{\beta}(t))$ is defined as

$$\operatorname{sgn}(\dot{\beta}(t)) = \begin{cases} \dot{\beta}(t) & \text{if } \dot{\beta}(t) > 0, \\ -\dot{\beta}(t) & \text{if } \dot{\beta}(t) < 0, \\ 0 & \text{if } \dot{\beta}(t) = 0. \end{cases}$$

Eq. (2.52) is valid when the relative angular velocity $\dot{\beta}$ is constant. We write the friction torque model in (2.52) in a concise form as follows

$$T_{ef}(t) = K_{ef}\dot{\beta}(t) + T_{e,fric}, \quad (2.53)$$

where $T_{e,fric}$ includes non-linear friction torque terms. We express (2.53) by removing $\dot{\beta}(t)$. We use (2.50) to replace $\dot{\beta}(t)$ and get,

$$\begin{aligned} T_{ef}(t) &= K_{ef}(\omega_{ey}(t) - (\omega_{by}(t) \cos \alpha(t) + \omega_{bx}(t) \sin \alpha(t))) + T_{e,fric}, \\ &= K_{ef}\omega_{ey}(t) - K_{ef}(\omega_{by}(t) \cos \alpha(t) + \omega_{bx}(t) \sin \alpha(t)) + T_{e,fric}. \end{aligned} \quad (2.54)$$

Eq. (2.54) is the expression for friction torque. We shall combine this to the pitch gimbal model later. In next section, we describe pitch gimbal cable restraint torque.

Pitch Gimbal Cable Restraint Torque

Due to spring-like behavior of pitch gimbal, cable restraint torque is generated. Cable restraint torque is proportional to the pitch LOS angle. Kennedy (2003) introduced a cable restraint torque model with linear and non-linear terms. For the linear term, they assume linear torsional spring model. Therefore, cable restraint torque is linearly proportional to pitch LOS angle β . However, they do not provide a model for non-linear term.

$$T_{e\omega}(t) = K_{e\omega}\beta(t) + T_{e,CR}, \quad (2.55)$$

where

$K_{e\omega}$ = Cable restraint co-efficient,

$T_{e,CR}$ = Nonlinear cable restraint torque.

Eq. (2.55) represents a system which has inputs pitch LOS angle β and the output is cable restraint torque $T_{e\omega}$.

$$\begin{aligned} T_{e\omega}(t) &= K_{e\omega}\left(\int_0^t \omega_{ey}(\tau)d\tau + \int_0^t (\omega_{by}(\tau) \cos \alpha(\tau) - \omega_{bx}(\tau) \sin \alpha(\tau))d\tau\right) + T_{e,CR}, \\ &= K_{e\omega} \int_0^t \omega_{ey}(\tau)d\tau + K_{e\omega} \int_0^t (\omega_{by}(\tau) \cos \alpha(\tau) - \omega_{bx}(\tau) \sin \alpha(\tau))d\tau + T_{e,CR}. \end{aligned} \quad (2.56)$$

Pitch Gimbal Model with Friction and Cable Restraint Torque Expression

Now putting Eq. (2.51), (2.54), and (2.56) into (2.49) we write,

$$\begin{aligned} J_{ey}\dot{\omega}_{ey}(t) = & T_{EL}(t) - T_{Uly}(t) - (K_{ef}\omega_{ey}(t) - K_{ef}(\omega_{by}(t) \cos \alpha(t) + \omega_{bx}(t) \sin \alpha(t))) \\ & + T_{e,fric} - (K_{ew} \int_0^t \omega_{ey}(\tau) d\tau + K_{ew} \int_0^t (\omega_{by}(\tau) \cos \alpha(\tau) - \omega_{bx}(\tau) \sin \alpha(\tau)) d\tau + T_{e,CR}) \\ & - (\omega_{ez}\omega_{ex}(J_{ex} - J_{ez}) + D_{xy}(\dot{\omega}_{ex} + \omega_{ey}\omega_{ez}) + D_{yz}(\dot{\omega}_{ez} - \omega_{ex}\omega_{ey}) + D_{xz}(\omega_{ez}^2 - \omega_{ex}^2)), \end{aligned} \quad (2.57)$$

and then by rearranging we write,

$$\begin{aligned} J_{ey}\dot{\omega}_{ey}(t) = & -K_{ew} \int_0^t \omega_{ey}(\tau) d\tau - K_{ef}\omega_{ey}(t) - (J_{ex} - J_{ez})\omega_{ax}(t)\omega_{ez}(t) \\ & + (J_{ex} - J_{ez})\omega_{ez}^2(t) \tan \beta + T_{EL} + K_{ef}(\omega_{by} \cos \alpha - \omega_{bx} \sin \alpha) + \\ & K_{ew} \int_0^t (\omega_{by}(\tau) \cos \alpha(\tau) - \omega_{bx}(\tau) \sin \alpha(\tau)) d\tau - T_{Uly} - T_{e,fric} - T_{e,CR}, \end{aligned} \quad (2.58)$$

Now let us define two terms: $T_{e,Dbase}$ is the torque generated from the base motion, and $T_{e,dist}$ is the disturbance torque generated from mass imbalance, non-linear friction torque terms, non-linear cable restraint torque terms.

$$T_{eD,base} = K_{ef}(\omega_{by} \cos \alpha - \omega_{bx} \sin \alpha) + K_{ew} \int_0^t (\omega_{by}(\tau) \cos \alpha(\tau) - \omega_{bx}(\tau) \sin \alpha(\tau)) d\tau, \quad (2.59)$$

and

$$T_{e,dist} = -T_{Uly} - T_{e,fric} - T_{e,CR}. \quad (2.60)$$

Then Eq. (2.58) becomes,

$$\begin{aligned} J_{ey}\dot{\omega}_{ey}(t) = & -K_{ew} \int_0^t \omega_{ey}(\tau) d\tau - K_{ef}\omega_{ey}(t) - (J_{ex} - J_{ez})\omega_{ax}(t)\omega_{ez}(t) \\ & + (J_{ex} - J_{ez})\omega_{ez}^2(t) \tan \beta + T_{EL} + T_{eD,base} + T_{e,dist}. \end{aligned} \quad (2.61)$$

We shall use Eq. (2.61) for the pitch gimbal part of the laser target system. In the next section, we shall model the yaw gimbal.

Yaw Gimbal Model

The rigid body torque dynamics for the yaw axis gimbal are derived from:

$$J_a \dot{\bar{\omega}}_a(t) + (\bar{\omega}_a(t) \times J_a \bar{\omega}_a(t)) + (\bar{L}_e(t))_a = \bar{L}_a(t), \quad (2.62)$$

where $\bar{L}_a(t)$

$$\bar{L}_a(t) = \begin{bmatrix} T_{ax}(t) \\ T_{ay}(t) \\ T_{AZ}(t) \end{bmatrix} - \begin{bmatrix} T_{Uox}(t) \\ T_{Uoy}(t) \\ T_{Uoz}(t) \end{bmatrix} - \begin{bmatrix} 0 \\ T_{af}(t) \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ T_{a\omega}(t) \\ 0 \end{bmatrix}, \quad (2.63)$$

with

$T_{af}(t)$ = Friction torque,

$T_{a\omega}(t)$ = Cable restraint torque,

$T_{Uox}(t), T_{Uoy}(t), T_{Uoz}(t)$ = Yaw gimbal mass unbalance torques,

$T_{ax}(t), T_{ay}(t)$ = Reaction torques exerted by yaw gimbal on base,

$T_{AZ}(t)$ = Azimuth axis stabilization control torque,

$(\bar{L}_e(t))_a$ = Sum of kinematic torques of pitch gimbal with respect to yaw gimbal frame,

$\bar{L}_a(t)$ = Sum of kinematic torques of yaw gimbal.

From (2.18) we write,

$$\begin{aligned} \dot{\bar{\omega}}_e(t) &= R_{ea}(\beta)(t) \dot{\bar{\omega}}_a(t) + Q_{ea}(\beta) \dot{\bar{\omega}}_a(t) \dot{\beta}(t) + \hat{\ddot{\beta}}(t), \\ \Rightarrow \dot{\bar{\omega}}_a(t) &= R_{ea}^T(\beta)(t) \dot{\bar{\omega}}_e(t) - R_{ea}^T(\beta)(t) Q_{ea}(\beta) \dot{\bar{\omega}}_a(t) \dot{\beta}(t) - R_{ea}^T(\beta)(t) \hat{\ddot{\beta}}(t). \end{aligned} \quad (2.64)$$

Taking the first term on the left of Eq. (2.62) we write,

$$\begin{aligned} J_a \dot{\bar{\omega}}_a(t) &= J_a (R_{ea}^T(\beta)(t) \dot{\bar{\omega}}_e(t) - R_{ea}^T(\beta)(t) Q_{ea}(\beta) \dot{\bar{\omega}}_a(t) \dot{\beta}(t) - R_{ea}^T(\beta)(t) \hat{\ddot{\beta}}(t)), \\ &= J_a R_{ea}^T(\beta)(t) \dot{\bar{\omega}}_e(t) - J_a R_{ea}^T(\beta)(t) Q_{ea}(\beta) \dot{\bar{\omega}}_a(t) \dot{\beta}(t) - J_a R_{ea}^T(\beta)(t) \hat{\ddot{\beta}}(t). \end{aligned} \quad (2.65)$$

Now, we express the last term of Eq. (2.62) or the sum of kinematic torques of pitch gimbal with respect to yaw-axis multiplying the transformation matrix with (2.41) and write

$$(\bar{L}_e(t))_a = R_{ea}^T(\beta) \bar{L}_e(t), \quad (2.66)$$

$$= R_{ea}^T(\beta) J_e \dot{\bar{\omega}}_e(t) + R_{ea}^T(\bar{\omega}_e(t) \times J_e \bar{\omega}_e(t)). \quad (2.67)$$

To obtain the dynamics in terms of pitch axis z component, (2.18) can be solved for $\dot{\omega}_a(t)$.

Substituting the resulting expression and (2.63) into (2.62) the following relation can be found:

Substituting (2.65) and (2.67) into (2.62) we write,

$$(J_a R_{ea}^T(\beta) + R_{ea}^T(\beta) J_e) \dot{\omega}_e(t) + (\bar{\omega}_a(t) \times J_a \bar{\omega}_a(t)) + R_{ea}^T(\beta) (\bar{\omega}_e(t) \times J_e \bar{\omega}_e(t)) \\ - J_a R_{ea}^T(\beta) Q_{ea}(\beta) \bar{\omega}_a(t) \dot{\beta} - J_a R_{ea}^T(\beta) \hat{\dot{\beta}} = L_a(t). \quad (2.68)$$

To simplify the analysis, Kennedy (2003) assumed that the gimbal rotation axes are aligned with the principal axes of inertia so that the inertia tensor is diagonal. This assumption makes the product of inertia elements of inertia tensor of (2.21) and (2.22) zero, i.e.:

$$D_{xy} = D_{yz} = D_{xz} = 0, \quad (2.69)$$

$$d_{xy} = d_{yz} = d_{xz} = 0. \quad (2.70)$$

Then we rewrite (2.21) and (2.22) as follows:

$$J_a = \begin{bmatrix} J_{ax} & 0 & 0 \\ 0 & J_{ay} & 0 \\ 0 & 0 & J_{az} \end{bmatrix}, \quad (2.71)$$

and

$$J_e = \begin{bmatrix} J_{ex} & 0 & 0 \\ 0 & J_{ey} & 0 \\ 0 & 0 & J_{ez} \end{bmatrix}. \quad (2.72)$$

Here the objective is to stabilize the angular velocity $\omega_{ez}(t)$. This can be found from the third element of (2.68) as follows:

$$[(J_a R_{ea}^T(\beta) + R_{ea}^T(\beta) J_e) \dot{\omega}_e(t)]_z + [(\bar{\omega}_a(t) \times J_a \bar{\omega}_a(t))]_z + [R_{ea}^T(\beta) (\bar{\omega}_e(t) \times J_e \bar{\omega}_e(t))]_z \\ [-J_a R_{ea}^T(\beta) Q_{ea}(\beta) \bar{\omega}_a(t) \dot{\beta}]_z - [J_a R_{ea}^T(\beta) \hat{\dot{\beta}}]_z = [L_a(t)]_z, \quad (2.73)$$

where $[]_z$ indicates the third element. Expanding the cross product terms and substituting the

kinematic torques leads to as shown in (Kennedy and Kennedy, 2003, p.5),

$$\begin{aligned}
& [J_{az} + J_{ez} \cos^2 \beta(t) + J_{ex} \sin^2 \beta(t)] \dot{\omega}_{ez}(t) = (J_{az} + J_{ex}) \dot{\omega}_{ax}(t) \sin \beta(t) \\
& + (J_{az} \omega_{ex}(t) - J_{ex} \omega_{az}(t) \sin \beta) \dot{\beta}(t) - [\omega_{ax}(t) \omega_{ay}(t) (J_{ay} - J_{ax}) - \omega_{ey}(t) \omega_{ez}(t) J_{ez} \sin \beta(t) \\
& + \omega_{ey}(t) \omega_{ax}(t) J_{ey} - \omega_{ex}(t) \omega_{ey}(t) J_{ex} \cos \beta(t)] \cos \beta(t) \\
& + [T_{AZ}(t) - T_{Uoz}(t) - T_{a,fric}(t) - T_{a\omega,CR}(t)] \cos \beta(t) \\
& - K_{af} \dot{\alpha}(t) \cos \beta(t) - K_{a\omega} \alpha(t) \cos \beta(t),
\end{aligned} \tag{2.74}$$

where K_{af} = Yaw gimbal viscous friction coefficient,

$K_{a\omega}$ = Yaw gimbal cable restraint coefficient,

$T_{a,fric}$ = Nonlinear friction torque,

$T_{a\omega,CR}$ = Nonlinear cable restraint torque.

Eq. (2.74) is the pitch axis dynamics equation. It can be modified by expressing the variables ω_{ex} and ω_{ax} in terms of the controlled variables ω_{ey} and ω_{ez} , and base disturbances. We use the following two relationships from Eq. (2.14):

$$\omega_{ex}(t) = \frac{1}{\cos \beta(t)} (\omega_{ax}(t) - \omega_{ez}(t) \sin \beta(t)), \tag{2.75}$$

$$\omega_{az}(t) = \frac{1}{\cos \beta(t)} (\omega_{ez}(t) - \omega_{ax}(t) \sin \beta(t)). \tag{2.76}$$

Using (2.75) and (2.76) the first term on the right hand side of Eq. (2.74) can be expressed as:

$$\begin{aligned}
& (J_{az} + J_{ex}) \dot{\omega}_{ax}(t) \sin \beta(t) = (J_{az} + J_{ex}) (\dot{\omega}_{bx}(t) \cos \alpha(t) - \dot{\omega}_{by}(t) \sin \alpha(t) \\
& + \omega_{ay}(t) \omega_{ax}(t) \tan \beta(t) + \omega_{ay}(t) \omega_{bz} \sin \beta(t) - (J_{az} + J_{ex}) \omega_{ez}(t) \omega_{ey}(t) \tan \beta(t)),
\end{aligned} \tag{2.77}$$

the second term as

$$\begin{aligned}
& (J_{az} \omega_{ex}(t) - J_{ex} \omega_{az}(t) \sin \beta(t)) \dot{\beta}(t) = (J_{az} + J_{ex}) \omega_{ay}(t) \omega_{az}(t) \tan \beta(t) \\
& - (J_{az} + J_{ex}) \omega_{ez}(t) \omega_{ey}(t) \tan \beta(t) + \frac{1}{\cos \beta(t)} (J_{az} + J_{ex} \sin^2 \beta(t)) \omega_{ax}(t) \omega_{ey}(t) \\
& - \frac{1}{\cos \beta(t)} (J_{az} + J_{ex} \sin^2 \beta(t)) \omega_{ax}(t) \omega_{ay}(t),
\end{aligned} \tag{2.78}$$

the third term as

$$\begin{aligned}
& [\omega_{ax}(t)\omega_{ay}(t)(J_{ay} - J_{ax}) - \omega_{ey}(t)\omega_{ez}(t)J_{ez}\sin\beta(t) \\
& + \omega_{ey}(t)\omega_{ax}(t)J_{ey} - \omega_{ex}(t)\omega_{ey}(t)J_{ex}\cos\beta(t)]\cos\beta(t) \\
& = (J_{ey} - J_{ex})\omega_{ax}(t)\omega_{ey}(t)\cos\beta(t) - (J_{ez} - J_{ex})\omega_{ey}(t)\omega_{ez}(t)\cos\beta(t)\sin\beta(t) \\
& + (J_{ay} - J_{ax})\omega_{ax}\omega_{ay}\cos\beta(t),
\end{aligned} \tag{2.79}$$

finally, the friction and cable restraint terms,

$$K_{af}\dot{\alpha}(t)\cos\beta(t) = K_{af}\omega_{ez}(t) - K_{af}\omega_{ax}(t)\sin\beta(t) - K_{af}\omega_{bz}(t)\cos\beta(t), \tag{2.80}$$

$$K_{a\omega} = K_{a\omega}(\cos\beta(t)) \int_0^t \left(\frac{1}{\cos\beta(t)} (\omega_{ez}(\tau) - \omega_{ax}(\tau)\sin\beta(\tau)) - \omega_{bz}(\tau) \right) d\tau. \tag{2.81}$$

From the left hand side of Eq. (2.74), we write the coefficient of $\dot{\omega}_{ez}(t)$ as

$$J_s = J_{az} + J_{ez}\cos^2\beta(t) + J_{ex}\sin^2\beta(t), \tag{2.82}$$

Using Eq. (2.77)-(2.82), we write (2.74) as,

$$\begin{aligned}
J_s\dot{\omega}_{ez}(t) &= (J_{az} + J_{ex})(\dot{\omega}_{bx}(t)\cos\alpha(t) - \dot{\omega}_{by}(t)\sin\alpha(t) \\
& + \omega_{ay}(t)\omega_{ax}(t)\tan\beta(t) + \omega_{ay}(t)\omega_{bz}\sin\beta(t) - (J_{az} + J_{ex})\omega_{ez}(t)\omega_{ey}(t)\tan\beta(t) \\
& + (J_{az} + J_{ex})\omega_{ay}(t)\omega_{az}(t)\tan\beta(t) - (J_{az} + J_{ex})\omega_{ez}(t)\omega_{ey}(t)\tan\beta(t) \\
& + \frac{1}{\cos\beta(t)}(J_{az} + J_{ex}\sin^2\beta(t))\omega_{ax}(t)\omega_{ey}(t) - \frac{1}{\cos\beta(t)}(J_{az} + J_{ex}\sin^2\beta(t))\omega_{ax}(t)\omega_{ay}(t) \\
& - ((J_{ey} - J_{ex})\omega_{ax}(t)\omega_{ey}(t)\cos\beta(t) - (J_{ez} - J_{ex})\omega_{ey}(t)\omega_{ez}(t)\cos\beta(t)\sin\beta(t) \\
& + (J_{ay} - J_{ax})\omega_{ax}\omega_{ay}\cos\beta(t)) + [T_{AZ}(t) - T_{Uoz}(t) - T_{a,fric}(t) - T_{a\omega,CR}(t)]\cos\beta(t) \\
& - (K_{af}\omega_{ez}(t) - K_{af}\omega_{ax}(t)\sin\beta(t) - K_{af}\omega_{bz}(t)\cos\beta(t)) \\
& - (K_{a\omega}(\cos\beta(t)) \int_0^t \left(\frac{1}{\cos\beta(t)} (\omega_{ez}(\tau) - \omega_{ax}(\tau)\sin\beta(\tau)) - \omega_{bz}(\tau) \right) d\tau).
\end{aligned} \tag{2.83}$$

Now, we write the coefficients of $\omega_{ax}(t)\omega_{ey}(t)$ as $g_y(t)$ and the coefficients of $\omega_{ey}(t)\omega_{ez}(t)$ as $g_{yz}(t)$ which is shown as follows:

$$g_y(t) = (J_{ey} - J_{ex})\cos\beta(t) - \frac{1}{\cos\beta(t)}(J_{az} + J_{ex}\sin^2\beta(t)), \tag{2.84}$$

$$g_{yz}(t) = (J_{az} + J_{ex}) \tan \beta(t) - (J_{ez} - J_{ex}) \cos \beta(t) \sin \beta(t). \quad (2.85)$$

Also, we define two terms: $T_{a,Dbase}$ is the torque generated from the base motion, and $T_{a,dist}$ is the disturbance torque generated from mass imbalance, non-linear friction torque terms, non-linear cable restraint torque terms.

$$\begin{aligned} T_{aD,base} &= (J_{az} + J_{ex})(\dot{\omega}_{bx}(t) \cos \alpha(t) - \dot{\omega}_{by}(t) \sin \alpha(t) + \omega_{ay}(t) \omega_{ax}(t) \tan \beta(t) \\ &\quad + \omega_{ay}(t) \omega_{bz}(t)) \sin \beta(t) + K_{af}(\omega_{ax}(t) \sin \beta(t) + \\ &\quad \omega_{bz}(t) \cos \beta(t)) + K_{a\omega}(\cos \beta(t)) \int_0^t (\omega_{ax}(\tau) \tan \beta(\tau) + \omega_{bz}(\tau)) d\tau. \end{aligned} \quad (2.86)$$

$$T_{a,dist} = -T_{Uoz} - T_{a,fric} - T_{a,CR}. \quad (2.87)$$

Using (2.84), (2.85), (2.86), and (2.87), we write (2.83) in a concise form as follows:

$$\begin{aligned} J_s \dot{\omega}_{ez}(t) &= -K_{a\omega}(\cos \beta(t)) \int_0^t \frac{1}{\cos \beta(\tau)} \omega_{ez}(\tau) d\tau - K_{af} \omega_{ez}(t) \\ &\quad - g_y(t) \omega_{ey}(t) \omega_{ax}(t) - g_{yz}(t) \omega_{ey}(t) \omega_{ez}(t) + T_{AZ} \cos \beta(t) + T_{aD,base} \cos \beta(t) + T_{a,dist} \cos \beta(t). \end{aligned} \quad (2.88)$$

We apply integration by parts rule on the term $-K_{a\omega}(\cos \beta) \int_0^t \frac{1}{\cos \beta} \omega_{ez}(\tau) d\tau$ and found

$$-K_{a\omega}(\cos \beta(t)) \int_0^t \frac{1}{\cos \beta(\tau)} \omega_{ez}(\tau) d\tau = -K_{a\omega} \int_0^t \omega_{ez}(\tau) d\tau + \cos \beta(t) K_{a\omega} \int_0^t \frac{\tan \beta(\tau)}{\cos \beta(\tau)} \omega_{ez}(\tau) d\tau. \quad (2.89)$$

Now Eq. (2.88) becomes,

$$\begin{aligned} J_s \dot{\omega}_{ez}(t) &= -K_{a\omega} \int_0^t \omega_{ez}(\tau) d\tau + K_{a\omega}(\cos \beta(t)) \int_0^t \frac{\tan \beta(\tau)}{\cos \beta(\tau)} \omega_{ez}(\tau) d\tau - K_{af} \omega_{ez}(t) \\ &\quad - g_y(t) \omega_{ey}(t) \omega_{ax}(t) - g_{yz}(t) \omega_{ey}(t) \omega_{ez}(t) + T_{AZ} \cos \beta(t) + T_{aD,base} \cos \beta(t) + T_{a,dist} \cos \beta(t). \end{aligned} \quad (2.90)$$

Equation (2.90) is the yaw gimbal non-linear model. Next, we shall model the brushless dc motor.

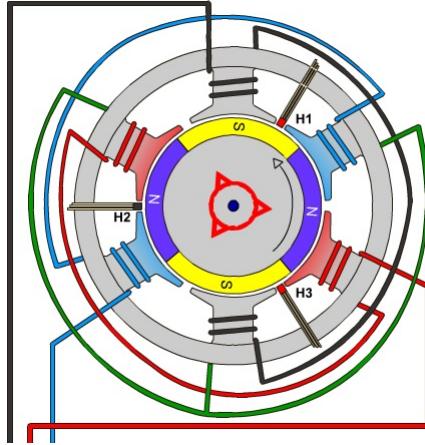


Figure 2.9: Cutway view of permanent magnet brushless DC motor assembly

2.2.2 Brushless DC Motor

In our system, one three phase permanent magnet brushless dc motor is used for producing torque to drive each gimbal. A three phase brushless DC motor (BLDCM) has three phase windings on the stator and permanent magnet rotor (Ramu, 2009). The induced back electromotive force (emf) of this type of motor is trapezoidal. To initiate the onset and commutation of current in the phase of a motor, the beginning and end of the constant flat portion of the induced emf have to be tracked. That amounts to only six discrete positions for a three-phase motor in each of the electrical cycle. These signals are generated using three Hall sensors displaced from each other by 120 electrical degrees. The Hall sensors are mounted facing a small magnet wheel fixed to the rotor having the same number of poles as in the rotor of BLDCM. Such an arrangement tracks the absolute position of the rotor magnets and hence the shape and position of the induced emfs in all the motor phases.

Fig. 2.9 shows the assembly of a permanent magnet brushless dc motor with a four pole permanent magnet as rotor, six coils as stator, and three hall sensors for rotor position feedback.

The derivation of the BLDCM model is based on the following assumptions:

- The motor is unsaturated.
- Eddy current and hysteresis effect in motor have negligible influence on winding current.

- The three phases are symmetric and balanced.
- The stator resistances per phase are equal.
- The mutual inductances are considered negligible.
- The stator self-inductance are constant and independent of rotor position.
- The stator self-inductances per phase are equal.
- The induced electromotive force (emf) are of trapezoidal shape.
- The devices in power switching circuit are ideal.

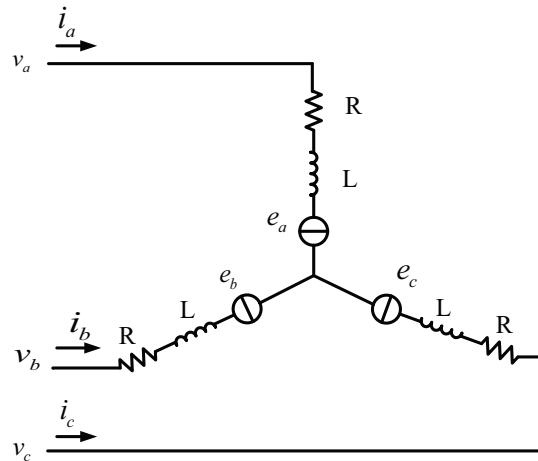


Figure 2.10: Equivalent circuit of a brushless DC motor

The equivalent circuit of BLDCM is shown in Fig. 2.10 (Krishnan, 2001). The coupled circuit equations of the stator windings in terms of motor electrical constants are,

$$v_a(t) = Ri_a(t) + L \frac{d}{dt} i_a(t) + e_a(t), \quad (2.91)$$

$$v_b(t) = Ri_b(t) + L \frac{d}{dt} i_b(t) + e_b(t), \quad (2.92)$$

$$v_c(t) = Ri_c(t) + L \frac{d}{dt} i_c(t) + e_c(t), \quad (2.93)$$

where $v_a(t), v_b(t), v_c(t)$ = phase voltage of stator windings for phase a , phase b and phase c , $i_a(t), i_b(t), i_c(t)$ = stator phase current for phase a , phase b and phase c ,

R = stator resistance,

L = stator self inductance,

$e_a(\theta_r(t)), e_b(\theta_r(t)), e_c(\theta_r(t))$ = instantaneous induced emfs for phase a , phase b and phase c .

Now, Eq. (2.91), (2.92), and (2.93) are written in the matrix form as,

$$\begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} + \begin{bmatrix} L & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & L \end{bmatrix} \begin{bmatrix} \frac{d(i_a(t))}{dt} \\ \frac{d(i_b(t))}{dt} \\ \frac{d(i_c(t))}{dt} \end{bmatrix} + \begin{bmatrix} e_a(\theta_r(t)) \\ e_b(\theta_r(t)) \\ e_c(\theta_r(t)) \end{bmatrix}. \quad (2.94)$$

The instantaneous-induced emfs are defined as (Krishnan, 2001),

$$\begin{bmatrix} e_a(\theta_r(t)) \\ e_b(\theta_r(t)) \\ e_c(\theta_r(t)) \end{bmatrix} = \lambda_p \omega_m(t) \begin{bmatrix} f_a(\theta_r(t)) \\ f_b(\theta_r(t)) \\ f_c(\theta_r(t)) \end{bmatrix}, \quad (2.95)$$

where

λ_p = modified flux linkage,

$\omega_m(t)$ = instantaneous motor angular velocity,

θ_r = rotor position,

$f_a(\theta_r(t)), f_b(\theta_r(t)), f_c(\theta_r(t))$ = functions with the same shape as $e_a(\theta_r(t)), e_b(\theta_r(t)), e_c(\theta_r(t))$ with maximum magnitude of ± 1 .

The definition of λ_p is found from the definition of the peak value of instantaneous induced emf E_p . E_p is derived as (Krishnan, 2001) :

$$E_p = (Blv)N = N(Blr\omega_m) = N\phi_a\omega_m = \lambda_p\omega_m, \quad (2.96)$$

where

N = the number of conductors in series per phase,

v = the velocity,

l = the length of the conductor,

r = the radius of the rotor bore,

ω_m = is the angular velocity,

B = the flux density of the field in which the conductors are placed.

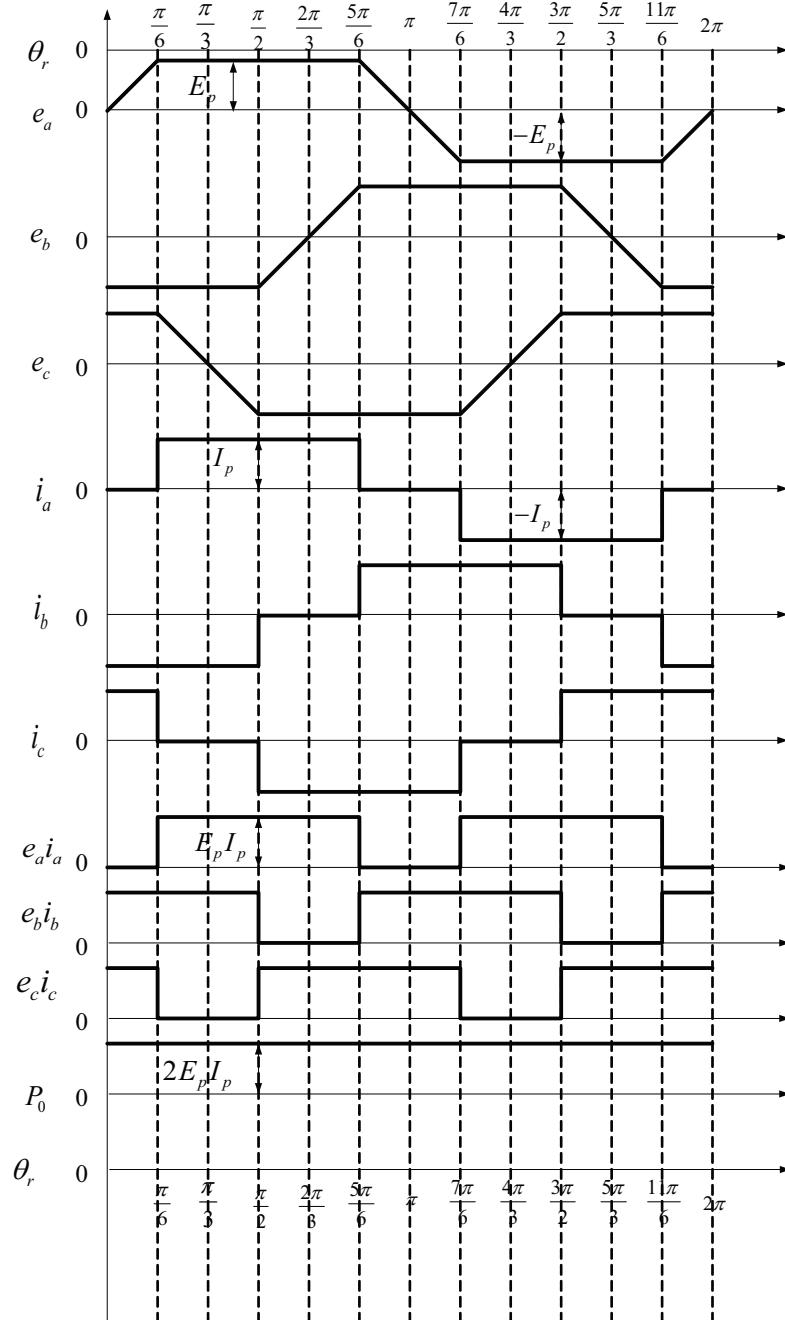


Figure 2.11: Permanent magnet DC motor waveform at six different rotor position

Fig. 2.11 shows the permanent magnet dc motor waveform (Ramu, 2009) with θ_r in radian, e_a, e_b, e_c in volts, i_a, i_b, i_c in ampere, $e_a i_a, e_b i_b, e_c i_c$ in watts, P_0 in watts. Table 2.1 lists the

Table 2.1: Rotor position dependent function with θ_r in radian

Rotor Position, θ_r	$f_a(\theta_r)$	$f_b(\theta_r)$	$f_c(\theta_r)$
$0 < \theta_r < \frac{\pi}{6}$	$\frac{6}{\pi}\theta_r$	-1	+1
$\frac{\pi}{6} < \theta_r < \frac{\pi}{2}$	+1	-1	$1 - \frac{6}{\pi}(\theta_r - \frac{\pi}{6})$
$\frac{\pi}{2} < \theta_r < \frac{5\pi}{6}$	+1	$1 + \frac{6}{\pi}(\theta_r - \frac{11\pi}{6})$	-1
$\frac{5\pi}{6} < \theta_r < \frac{7\pi}{6}$	$1 - \frac{6}{\pi}(\theta_r - \frac{5\pi}{6})$	+1	-1
$\frac{7\pi}{6} < \theta_r < \frac{3\pi}{2}$	-1	+1	$1 - \frac{2}{\pi}(\theta_r - \frac{3\pi}{2})$
$\frac{3\pi}{2} < \theta_r < \frac{11\pi}{6}$	-1	$1 - \frac{6}{\pi}(\theta_r - \frac{3\pi}{2})$	+1
$\frac{11\pi}{6} < \theta_r < 2\pi$	$-1 + \frac{6}{\pi}(\theta_r - \frac{11\pi}{6})$	-1	+1

values of $f_a(\theta_r)$, $f_b(\theta_r)$, and $f_c(\theta_r)$ depending on the rotor position θ_r .

Eq. (2.94) is re-written as follows using (2.95),

$$\begin{aligned}
 \frac{d}{dt} \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} = & \begin{bmatrix} -R/L & 0 & 0 \\ 0 & -R/L & 0 \\ 0 & 0 & -R/L \end{bmatrix} \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} + \\
 & \begin{bmatrix} -\lambda_p \omega_m(t)/L & 0 & 0 \\ 0 & -\lambda_p \omega_m(t)/L & 0 \\ 0 & 0 & -\lambda_p \omega_m(t)/L \end{bmatrix} \begin{bmatrix} f_a(\theta_r(t)) \\ f_b(\theta_r(t)) \\ f_c(\theta_r(t)) \end{bmatrix} \\
 & + \begin{bmatrix} -1/L & 0 & 0 \\ 0 & -1/L & 0 \\ 0 & 0 & -1/L \end{bmatrix} \begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix}. \quad (2.97)
 \end{aligned}$$

The equation of motion for the mechanical subsystem of the motor is described as (Ramu, 2009),

$$J_m \frac{d\omega_m}{dt} + b\omega_m = T_e(t) - T_L(t), \quad (2.98)$$

where J_m is the motor moment of inertia, b is the viscous friction coefficient of the motor, T_L

is the load torque. The electromagnetic torque is given by (Ramu, 2009),

$$T_e(t) = (e_a(\theta_r(t))i_a(t) + e_b(\theta_r(t))i_b(t) + e_c(\theta_r(t))i_c(t))\frac{1}{\omega_m(t)}. \quad (2.99)$$

Substituting (2.95) into (2.99) we get,

$$\begin{aligned} T_e &= (\lambda_p\omega_m(t)f_a(\theta_r(t))i_a(t) + \lambda_p\omega_m(t)f_b(\theta_r(t))i_b(t) + \lambda_p\omega_m(t)f_c(\theta_r(t))i_c(t))\frac{1}{\omega_m(t)}, \\ &= \lambda_p f_a(\theta_r(t))i_a(t) + \lambda_p f_b(\theta_r(t))i_b(t) + \lambda_p f_c(\theta_r(t))i_c(t), \end{aligned} \quad (2.100)$$

$$= \begin{bmatrix} \lambda_p f_a(\theta_r(t)) & \lambda_p f_b(\theta_r(t)) & \lambda_p f_c(\theta_r(t)) \end{bmatrix} \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix}. \quad (2.101)$$

For the compactness of the model, T_L is used as one of the input for the model. However, we can consider load torque $T_L = 0$. We can also assume electromagnetic torque is completely converted to mechanical torque which will drive the gimbal. With this assumption and using (2.100), Eq. (2.98) becomes,

$$J_m \frac{d\omega_m}{dt} + b\omega_m = T_e(t) = \lambda_p f_a(\theta_r(t))i_a(t) + \lambda_p f_b(\theta_r(t))i_b(t) + \lambda_p f_c(\theta_r(t))i_c(t), \quad (2.102)$$

$$\frac{d\omega_m}{dt} = \frac{\lambda_p}{J_m} f_a(\theta_r(t))i_a(t) + \frac{\lambda_p}{J_m} f_b(\theta_r(t))i_b(t) + \frac{\lambda_p}{J_m} f_c(\theta_r(t))i_c(t) - \frac{b}{J_m}\omega_m(t). \quad (2.103)$$

The angular velocity of the motor and the rotor position θ_r is related by,

$$\frac{d\theta_r(t)}{dt} = \frac{P}{2}\omega_m(t), \quad (2.104)$$

where P is the number of magnetic poles of the motor.

Combining (2.97), (2.103) and (2.104), we get the following state space form,

$$\dot{x}_m(t) = A_m(\theta_r(t))x_m(t) + B_m(t)u_m(t), \quad (2.105)$$

where

$$x_m(t) = \begin{bmatrix} i_a(t) & i_b(t) & i_c(t) & \omega_m(t) & \theta_r(t) \end{bmatrix}^T, \quad (2.106)$$

$$u_m(t) = \begin{bmatrix} v_a(t) & v_b(t) & v_c(t) & T_L(t) \end{bmatrix}^T, \quad (2.107)$$

$$A_m(\theta_r(t)) = \begin{bmatrix} -\frac{R}{L} & 0 & 0 & \frac{\lambda_p}{L} f_a(\theta_r) & 0 \\ 0 & -\frac{R}{L} & 0 & \frac{\lambda_p}{L} f_a(\theta_r) & 0 \\ 0 & 0 & -\frac{R}{L} & \frac{\lambda_p}{L} f_a(\theta_r) & 0 \\ \frac{\lambda_p}{J_m} f_a(\theta_r) & \frac{\lambda_p}{J_m} f_b(\theta_r) & \frac{\lambda_p}{J_m} f_c(\theta_r) & -\frac{b}{J_m} & 0 \\ 0 & 0 & 0 & \frac{P}{2} & 0 \end{bmatrix}, \quad (2.108)$$

$$B_m = \begin{bmatrix} \frac{1}{L} & 0 & 0 & 0 \\ 0 & \frac{1}{L} & 0 & 0 \\ 0 & 0 & \frac{1}{L} & 0 \\ 0 & 0 & 0 & -\frac{1}{J_m} \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2.109)$$

Now it is convenient to express electromagnetic torque in terms of states. Because we need to monitor states for using them as feedback to the other components of the servo drive. So the desired output from this motor will be all the states and the electromagnetic torque.

To express electromagnetic torque in terms of states, we augment (2.101) and (2.106), we found

$$T_e(\theta_r(t)) = \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \\ \omega_m(t) \\ \theta_r(t) \end{bmatrix} = \begin{bmatrix} \lambda_p f_a(\theta_r(t)) & \lambda_p f_b(\theta_r(t)) & \lambda_p f_c(\theta_r(t)) & 0 & 0 \end{bmatrix} \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \\ \omega_m(t) \\ \theta_r(t) \end{bmatrix}. \quad (2.110)$$

In state space form, the output can be written as,

$$y_m(t) = \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \\ \omega_m(t) \\ \theta_r(t) \\ T_e(t) \end{bmatrix} = \begin{bmatrix} & & & & I_{5 \times 5} \\ \lambda_p f_a(\theta_r(t)) & \lambda_p f_b(\theta_r(t)) & \lambda_p f_c(\theta_r(t)) & 0 & 0 \end{bmatrix} \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \\ \omega_m(t) \\ \theta_r(t) \end{bmatrix}, \quad (2.111)$$

$$= C_m(t)x_m(t). \quad (2.112)$$

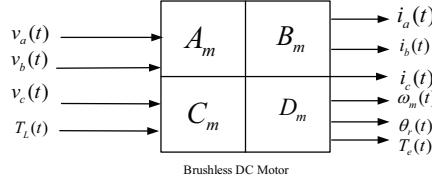


Figure 2.12: Brushless DC motor model

Fig. 2.12 shows the state diagram of brushless DC motor. Here A_m is 5×5 , B_m is 5×4 , C_m is 6×5 , D_m is 6×4 matrices and x_m is 5×1 , u_m is 4×1 vectors. This completes the modeling of brushless DC motor.

2.2.3 Servo Drive

Speed Controller

The absolute rotor position is provided by the hall sensors of the brushless dc motor and it is converted to the rotor speed through the signal processor (Krishnan, 2001). The reference rotor speed is compared with filtered actual rotor speed, and the speed error is amplified through the speed controller. The speed controller is a Proportional-Integral Controller as shown in Fig. 2.13. The output of the speed controller is the reference torque, T_e^* . The transfer function is

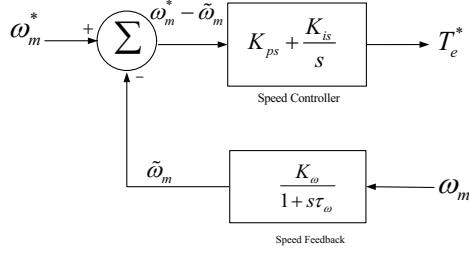


Figure 2.13: Speed controller of servo drive

given by (Krishnan, 2001),

$$\frac{T_e^*}{\omega_m^* - \tilde{\omega}_m} = K_{ps} + \frac{K_{is}}{s}, \quad (2.113)$$

where

ω_m^* = the reference speed,

$\tilde{\omega}_m$ = the filtered rotor speed,

K_{ps} = the speed controller proportional gain,

K_{is} = the speed controller integral gain.

Speed Feedback

Speed feedback is modelled with a low-pass filter with gain K_ω and time constant τ_ω as in Fig. 2.13. The transfer function of the speed feedback filter is

$$\frac{\tilde{\omega}_m}{\omega_m} = \frac{K_\omega}{1 + s\tau_\omega}. \quad (2.114)$$

Current Command Generator

Speed controller produces reference torque command T_e^* , which is the input for the reference current command generator. Only two phases conduct current at any time with the two phases being in series for a full wave inverter operation, the phase currents are equal in magnitude but opposite in sign. The rotor position-dependent functions have the same signs as the stator phase currents in the conduction mode (Ramu, 2009).

For example, from Fig. 2.11, we see when the rotor position is between $\frac{\pi}{6}$ to $\frac{\pi}{2}$ radian, phase a and b conduct. Then, $i_a = I_p$, $i_b = -I_p$, and $i_c = 0$. From Table 2.1, $f_a(\theta_r) = +1$,

$f_b(\theta_r) = -1$, and $f_c(\theta_r) = 1 - \frac{6}{\pi}(\theta_r - \frac{\pi}{6})$. Now from (2.100) we get,

$$T_e = \lambda_p f_a(\theta_r(t)) i_a(t) + \lambda_p f_b(\theta_r(t)) i_b(t) + \lambda_p f_c(\theta_r(t)) i_c(t), \quad (2.115)$$

$$= \lambda_p (1 \cdot I_p + (-1) \cdot (-I_p) + (1 - \frac{6}{\pi}(\theta_r - \frac{\pi}{6})) \cdot 0), \quad (2.116)$$

$$= 2I_p \lambda_p. \quad (2.117)$$

For example, from Fig. 2.11, we see when the rotor position is between $\frac{\pi}{2}$ to $\frac{5\pi}{6}$ radian, phase a and c conduct. Then, $i_a = I_p$, $i_b = 0$, and $i_c = -I_p$. From Table 2.1, $f_a(\theta_r) = +1$, $f_b(\theta_r) = 1 + \frac{6}{\pi}(\theta_r - \frac{11\pi}{6})$, and $f_c(\theta_r) = -1$. Now from (2.100) we get,

$$T_e = \lambda_p f_a(\theta_r(t)) i_a(t) + \lambda_p f_b(\theta_r(t)) i_b(t) + \lambda_p f_c(\theta_r(t)) i_c(t), \quad (2.118)$$

$$= \lambda_p (1 \cdot I_p + (1 + \frac{6}{\pi}(\theta_r - \frac{11\pi}{6})) \cdot 0) + (-1) \cdot (-I_p), \quad (2.119)$$

$$= 2I_p \lambda_p. \quad (2.120)$$

In this way, we can calculate for T_e at different rotor positions. At any rotor position Eq. (2.117) is true. We see that electromagnetic torque is related with current magnitude with a flux linkage factor.

Now we wish to generate the magnitude of phase currents i_a^* , i_b^* , and i_c^* . First, we shall calculate the peak magnitude of current I_p^* . The peak current command I_p^* is found from electromagnetic torque T_e^* (Ramu, 2009).

$$T_e^* = \begin{bmatrix} \lambda_p f_a(\theta_r) & \lambda_p f_b(\theta_r) & \lambda_p f_c(\theta_r) \end{bmatrix} \begin{bmatrix} i_a^* \\ i_b^* \\ i_c^* \end{bmatrix}, \quad (2.121)$$

where i_a^* , i_b^* and i_c^* are reference phase currents of three phases a , b , c . At any time the torque command is described as (2.117). Therefore,

$$T_e^* = 2\lambda_p I_p^*. \quad (2.122)$$

The current command is then

$$I_p^* = \frac{T_e^*}{2\lambda_p}. \quad (2.123)$$

Table 2.2: Magnitude of phase currents based on rotor position θ_r in radian

Rotor Position, $\theta_r(t)$	$i_a^*(t)$	$i_b^*(t)$	$i_c^*(t)$
$0 < \theta_r(t) < \frac{\pi}{6}$	0	$-I_p^*$	$+I_p^*$
$\frac{\pi}{6} < \theta_r(t) < \frac{\pi}{2}$	$+I_p^*$	$-I_p^*$	0
$\frac{\pi}{2} < \theta_r(t) < \frac{5\pi}{6}$	$+I_p^*$	0	$-I_p^*$
$\frac{5\pi}{6} < \theta_r(t) < \frac{7\pi}{6}$	0	$+I_p^*$	$-I_p^*$
$\frac{7\pi}{6} < \theta_r(t) < \frac{3\pi}{2}$	$-I_p^*$	$+I_p^*$	0
$\frac{3\pi}{2} < \theta_r(t) < \frac{11\pi}{6}$	$-I_p^*$	0	$+I_p^*$
$\frac{11\pi}{6} < \theta_r(t) < 2\pi$	0	$-I_p^*$	$+I_p^*$

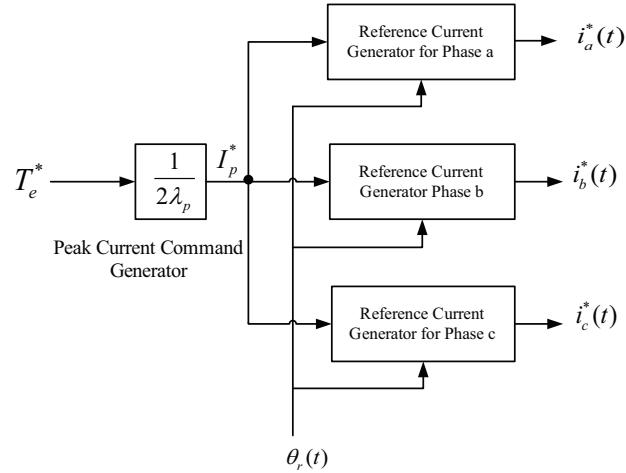


Figure 2.14: Current command generator of servo drive

The individual phase current commands are generated from the current magnitude command and absolute rotor position as shown in Table 2.2. Therefore, the current command generator outputs current command based on reference torque, and based on rotor position individual phase current is generated from current command generator as shown in Fig. 2.14.

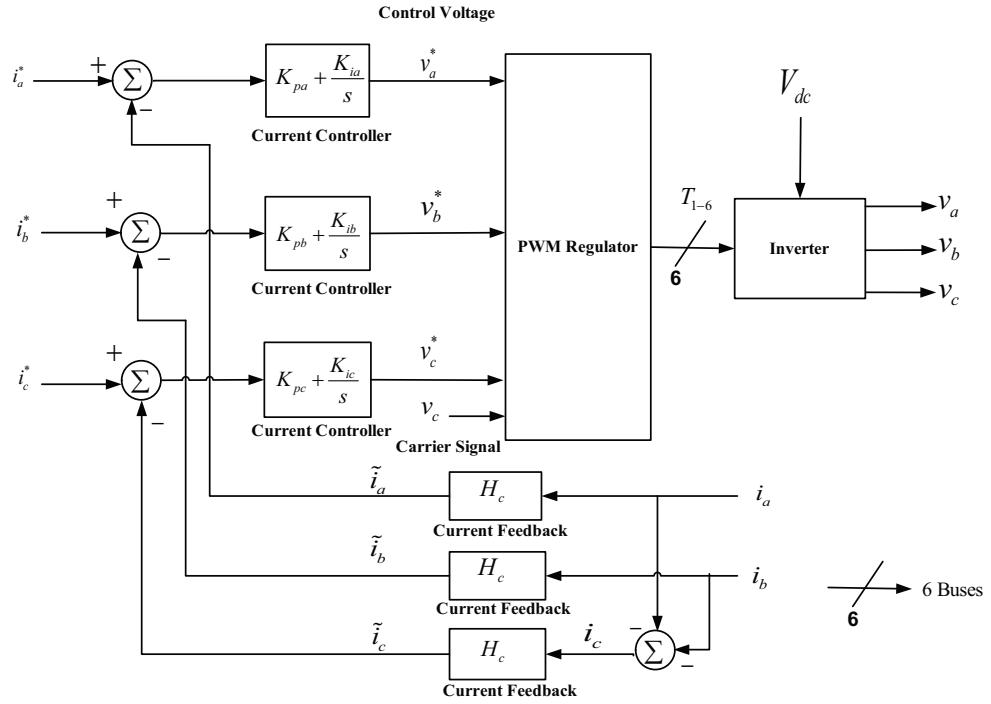


Figure 2.15: PWM current controller of servo drive

Pulse Width Modulated (PWM) Current Controller

PWM current controller takes inputs of individual phase currents and produces gate pulses for the three phase inverter (Ramu, 2009) as shown in Fig. 2.15. To amplify these current references through the inverter, the phase currents are sensed and compared to their respective references to generate errors in phase currents.

For each of the phase, there is a PI controller which forces the current error to zero. The outcome of the current controller is a signal that demands the phase voltage to follow in such a way that the current error in that phase is reduced to zero. Therefore, the current controller output is a voltage signal and it will vary very fast. This is the reference signal to interact with the PWM carrier signal to generate gate pulses with variable duty cycles for the devices in a phase leg of the inverter.

Similar is the case for each and every phase of the motor drive. Only two-phase currents are sensed. In a three-phase three-wire system; the sum of the phase currents being zero, the third phase current is constructed using this fact. The processing circuit provides this option and is

intended to include the filters for all individual phase currents too.

Current Feedback

The gain of the current feedback is H_c . The gain is same for all three phases. For most of the practical applications, no significant filtering is required (Krishnan, 2001). The input of the each current feedback is phase currents i_a, i_b, i_c respectively. The outputs are $\tilde{i}_a, \tilde{i}_b, \tilde{i}_c$.

Current Controller

The reference phase current and the current from the feedback are compared, and the current error is amplified through the PI current controller. The output of the current controller is a control voltage signal for the PWM regulator. The transfer function of the current controller for phase a, b , and c are given respectively,

$$\frac{v_a^*}{i_a^* - \tilde{i}_a} = K_{pa} + \frac{K_{ia}}{s}, \quad (2.124)$$

$$\frac{v_b^*}{i_b^* - \tilde{i}_b} = K_{pb} + \frac{K_{ib}}{s}, \quad (2.125)$$

$$\frac{v_c^*}{i_c^* - \tilde{i}_c} = K_{pc} + \frac{K_{ic}}{s}, \quad (2.126)$$

where

i_a^*, i_b^*, i_c^* = the reference current for phase a, b , and c ,

$\tilde{i}_a, \tilde{i}_b, \tilde{i}_c$ = the filtered actual current from current feedback for phase a, b , and c ,

K_{pa}, K_{pb}, K_{pc} = the current controller proportional gain for phase a, b , and c , and

K_{ia}, K_{ib}, K_{ic} = the current controller integral gain for phase a, b , and c .

Equations (2.124), (2.125), and (2.126) are PI current controller models for phase a, b , and c respectively.

PWM Regulator

The control voltage generated by the PI controller is compared with a sawtooth shaped carrier signal to generate switching signal for the base drive of the inverter switches (Pillay and

Krishnan, 1989). It is achieved by varying the pulse width to the gate signals of the inverter and it is known as PWM. Its realization is as follows. If the control voltage signal is larger than the sawtooth, the voltage is switched positively, while if the control voltage signal is smaller than the sawtooth, the voltage is switched negatively. The switching logic for one phase is summarized as:

$$T1 = \frac{1}{2}V_{dc}, \quad v_a^* > v_c, \quad (2.127)$$

$$T4 = -\frac{1}{2}V_{dc}, \quad v_a^* > v_c. \quad (2.128)$$

For example, if T1 is conducting, v_a is equal to $+\frac{V_{dc}}{2}$, where V_{dc} is the dc supply voltage and the reference is taken as the midpoint of the supply. By switching T1 off, the free-wheeling diode across T4 immediately starts conducting to maintain the current flow through the motor inductance. This automatically forces v_a to equal to $-\frac{V_{dc}}{2}$, even though T4 is not yet conducting. This is called a PWM current controller because of the pulse width modulation of the voltage.

The switching logic for the other two phases are summarized as:

$$T5 = \frac{1}{2}V_{dc}, \quad v_b^* > v_c, \quad (2.129)$$

$$T2 = -\frac{1}{2}V_{dc}, \quad v_b^* > v_c, \quad (2.130)$$

$$T3 = \frac{1}{2}V_{dc}, \quad v_c^* > v_c, \quad (2.131)$$

$$T6 = -\frac{1}{2}V_{dc}, \quad v_c^* > v_c. \quad (2.132)$$

The positive voltage means logical one and the negative voltage means logical zero. Fig. 2.15 shows the PWM regulator input and output. The inverters on the same leg will not have only positive or only negative voltages because of the power supply polarity. Therefore, the switching logic follows certain sequence.

The Three-phase Inverter

The three-phase MOSFET (Metal-Oxide-Semiconductor Field Effect Transistor) inverter produces three-phase voltages for brushless dc motor. The gates of the MOSFETs in the inverter are triggered in such a manner that it regulates the input supply voltage to the output of the

inverter. Therefore, the three-phase inverter has input dc supply voltage and six input voltages generated from PWM regulator.

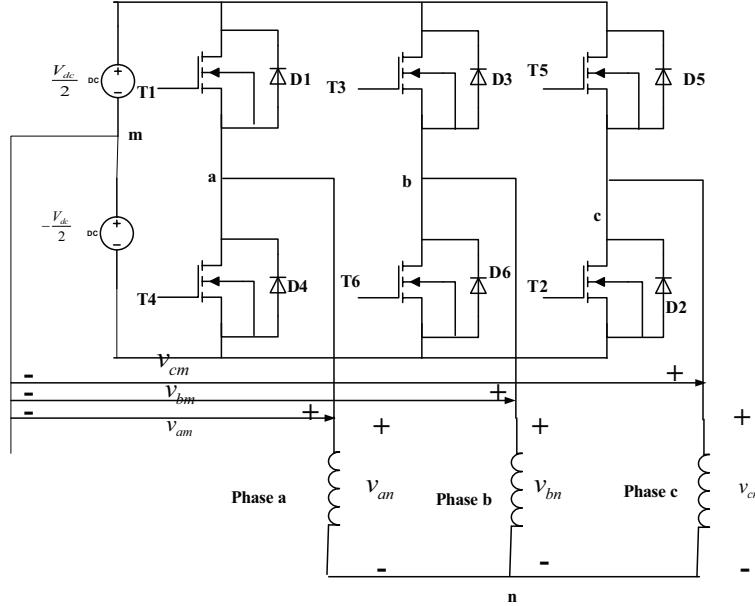


Figure 2.16: Three-phase MOSFET inverter for derivation of voltage relation

To find the relationship between the gate trigger voltages, dc supply voltages and the output voltages we need to assume:

- The dc supply voltage is constant in the three phase inverter circuit shown in Fig. 2.16.
- Each of the phase leg is operated independently.
- One switch in each phase leg is turned on at a given time.
- When a switch (top/bottom) is turned off, the current in that phase is transferred to the other device's (bottom/top) anti parallel or free-wheeling diode. This is to provide sufficient delay for switching , and hence avoid a short circuit in the dc supply.
- The top/bottom switches' complement is bottom/top switch.
- The dc supply voltage is split into two halves at the midpoint of dc voltage source.

Table 2.3: Switching states possibilities for three phase inverter

Mode	T1	T3	T5	v_{am}	v_{bm}	v_{cm}
I	1	0	0	+0.5 V_{dc}	-0.5 V_{dc}	-0.5 V_{dc}
II	1	0	1	+0.5 V_{dc}	-0.5 V_{dc}	+0.5 V_{dc}
III	1	1	0	+0.5 V_{dc}	+0.5 V_{dc}	-0.5 V_{dc}
IV	1	1	1	+0.5 V_{dc}	+0.5 V_{dc}	+0.5 V_{dc}
V	0	1	1	-0.5 V_{dc}	+0.5 V_{dc}	+0.5 V_{dc}
VI	0	1	0	-0.5 V_{dc}	+0.5 V_{dc}	-0.5 V_{dc}
VII	0	0	1	-0.5 V_{dc}	-0.5 V_{dc}	+0.5 V_{dc}
VIII	0	0	0	-0.5 V_{dc}	-0.5 V_{dc}	-0.5 V_{dc}

Let,

m = the midpoint of the source voltage,

a, b, c = The midpoint of the inverter phase legs,

v_{am}, v_{bm}, v_{cm} = The inverter midpoint voltages with respect to dc source midpoint,

v_{ab}, v_{bc}, v_{ca} = The line voltages,

v_{an}, v_{bn}, v_{cn} = The phase voltages,

The three phase legs are with transistor T1 and T4, T3 and T6, and T5 and T2.

T4, T6, T2 are complementary switch of T1, T3, and T5.

They vary depending on the state of the transistors T1 and T4, T3 and T6, and T5 and T2, respectively.

Table 2.3 shows the possible switching states of transistor and midpoint voltage corresponding to the states. In Table 2.3, 1 means the switch is ON, 0 means the switch is OFF.

We also assume the brushless dc motor as only reactive load. We assume the switches are ideal which means we are ignoring free-wheeling diode for now. The relationship between the

line voltages and the midpoint voltages are

$$v_{ab} = v_{am} - v_{bm}, \quad (2.133)$$

$$v_{bc} = v_{bm} - v_{cm}, \quad (2.134)$$

$$v_{ca} = v_{cm} - v_{am}. \quad (2.135)$$

We assume that this is a balanced three-phase system. Therefore,

$$v_{an} + v_{bn} + v_{cn} = 0. \quad (2.136)$$

The relationship between phase voltages and midpoint voltages are given by,

$$v_{an} = v_{am} - v_{nm}, \quad (2.137)$$

$$v_{bn} = v_{bm} - v_{nm}, \quad (2.138)$$

$$v_{cn} = v_{cm} - v_{nm}. \quad (2.139)$$

Summing up Eq. (2.137), (2.138), and (2.139) we get

$$v_{an} + v_{bn} + v_{cn} = v_{am} + v_{bm} + v_{cm} - 3v_{nm}. \quad (2.140)$$

Substituting (2.136) into (2.140), we get

$$v_{am} + v_{bm} + v_{cm} - 3v_{nm} = 0, \quad (2.141)$$

$$v_{nm} = \frac{1}{3}(v_{am} + v_{bm} + v_{cm}). \quad (2.142)$$

Using Table 2.3 we get for mode I, $v_{nm} = \frac{1}{6}V_{dc}$. For Eq. (2.137) we can write,

$$v_{an} = v_{am} - v_{nm}, \quad (2.143)$$

$$= v_{am} - \frac{1}{3}(v_{am} + v_{bm} + v_{cm}), \quad (2.144)$$

$$= \frac{2}{3}v_{am} - \frac{1}{3}(v_{bm} + v_{cm}). \quad (2.145)$$

Likewise, we can write

$$v_{bn} = \frac{2}{3}v_{bm} - \frac{1}{3}(v_{am} + v_{cm}), \quad (2.146)$$

$$v_{cn} = \frac{2}{3}v_{cm} - \frac{1}{3}(v_{am} + v_{bm}). \quad (2.147)$$

Table 2.4: Midpoint voltage and phase voltage waveform for three phase inverter

Mode	T1	T4	T3	T6	T5	T2	v_{nm}	v_{an}	v_{bn}	v_{cn}
I	1	0	0	1	0	1	$+\frac{V_{dc}}{6}$	$+\frac{2V_{dc}}{3}$	$-\frac{V_{dc}}{3}$	$-\frac{V_{dc}}{3}$
II	1	0	0	1	1	0	$-\frac{V_{dc}}{6}$	$\frac{V_{dc}}{3}$	$\frac{V_{dc}}{3}$	$-\frac{2V_{dc}}{3}$
III	1	0	1	0	0	1	$+\frac{V_{dc}}{6}$	$-\frac{V_{dc}}{3}$	$+\frac{2V_{dc}}{3}$	$-\frac{V_{dc}}{3}$
IV	1	0	1	0	1	0	$-\frac{V_{dc}}{6}$	$-\frac{2V_{dc}}{3}$	$\frac{V_{dc}}{3}$	$\frac{V_{dc}}{3}$
V	0	1	1	0	1	0	$+\frac{V_{dc}}{6}$	$-\frac{V_{dc}}{3}$	$-\frac{V_{dc}}{3}$	$+\frac{2V_{dc}}{3}$
VI	0	1	1	0	0	1	$-\frac{V_{dc}}{6}$	$\frac{V_{dc}}{3}$	$-\frac{2V_{dc}}{3}$	$+\frac{V_{dc}}{3}$
VII	0	1	0	1	1	0	$+\frac{V_{dc}}{6}$	$\frac{2V_{dc}}{3}$	$-\frac{V_{dc}}{3}$	$-\frac{V_{dc}}{3}$
VIII	0	1	0	1	0	1	$-\frac{V_{dc}}{6}$	$\frac{V_{dc}}{3}$	$\frac{V_{dc}}{3}$	$-\frac{2V_{dc}}{3}$

Let us give an example of voltages during mode I. In mode I, T1 is ON, T3, T5 are OFF. Thus their complementary switch T4 is OFF, and T6, T2 are ON. In this case, from Table 2.3, $v_{am} = 0.5V_{dc}$ and $v_{bm} = v_{cm} = -0.5V_{dc}$. Using these values into Eq. (2.145), (2.146), and (2.147), we get $v_{an} = \frac{2V_{dc}}{3}$ and $v_{bn} = v_{cn} = -\frac{V_{dc}}{3}$. The same calculation is applicable for other modes.

Table 2.4 lists the modes, switching states (logical 1 and 0), and corresponding phase voltages in terms of dc supply voltage.

For the rest of the document, for phase voltage v_{an}, v_{bn}, v_{cn} we shall use v_a, v_b, v_c .

In summary, speed controlled permanent magnet brushless DC Motor servo drive scheme has PI speed controller, PWM current controller, three-phase inverter and permanent magnet brushless DC motor.

Fig. 2.18 shows the diagram of permanent magnet brushless DC motor servo drive scheme.

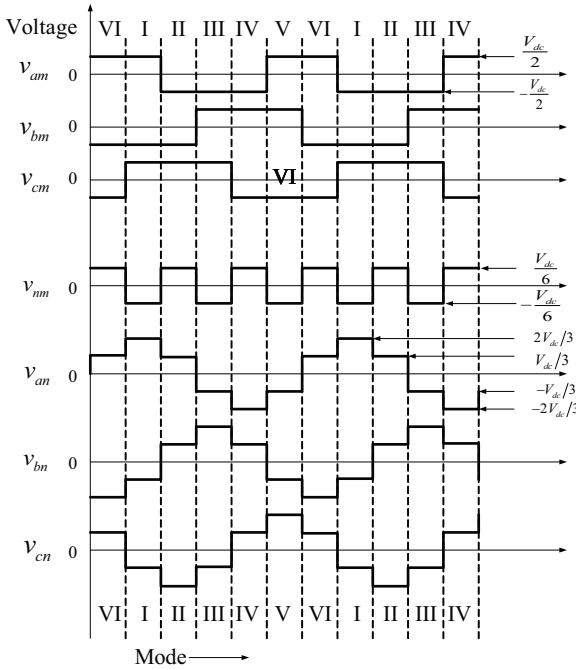


Figure 2.17: Three-phase MOSFET inverter waveform from Table 2.4

2.2.4 Laser Pointer Model

The laser pointer generates a laser beam. The pointer is mounted on the pitch gimbal payload mounting plate. It is assumed that the pointer is rigidly attached to the pitch gimbal. We define the pointer co-ordinate frame is parallel with the pitch gimbal frame. It follows that the laser pointer co-ordinate frame can be expressed by $\begin{bmatrix} x_{e,LP} & y_{e,LP} & z_{e,LP} \end{bmatrix}^T = \begin{bmatrix} x_e & y_e & z_e \end{bmatrix}^T$. With this definition, it can be concluded that the laser beam horizontal motion depends on the yaw LOS angle and vertical motion depends on pitch gimbal LOS angle. This means that the laser beam angular displacement from its previous position can be considered equal to the change of yaw and pitch gimbal LOS angles. Also the angular rates of both pitch gimbal and the laser pointer will be the same. Therefore, it is sufficient to model the laser pointer kinematics with a unit gain.

2.2.5 Image-based Position Sensor

The input of the image-based position sensor is the angular rate of the laser pointer, i.e., laser beam and the output is the position of laser beam on a white receiver screen expressed

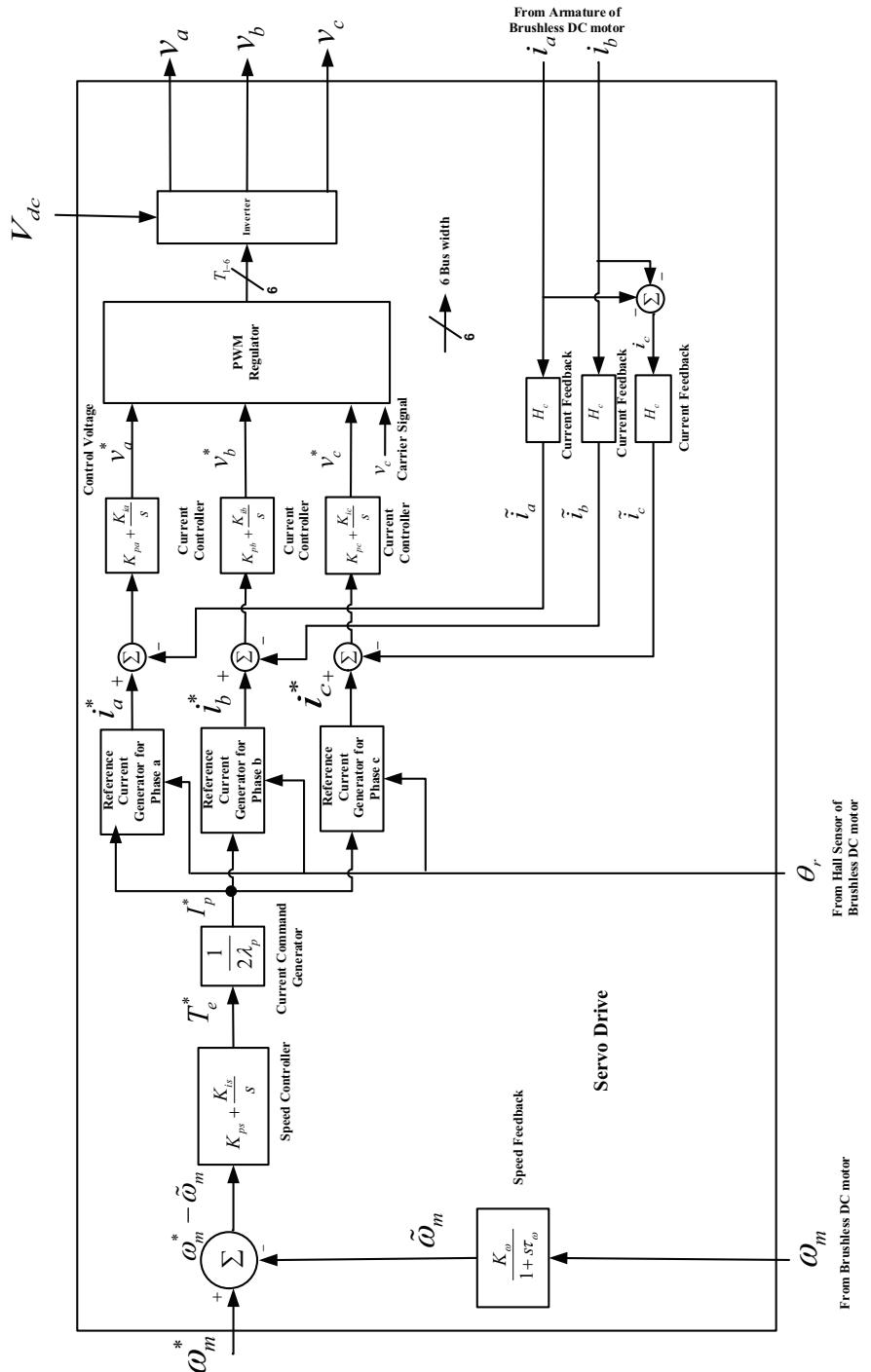


Figure 2.18: Speed controlled permanent magnet brushless DC motor servo drive scheme

in centimeter. The hardware and software part of image-based position sensor is shown in Fig. 2.19:

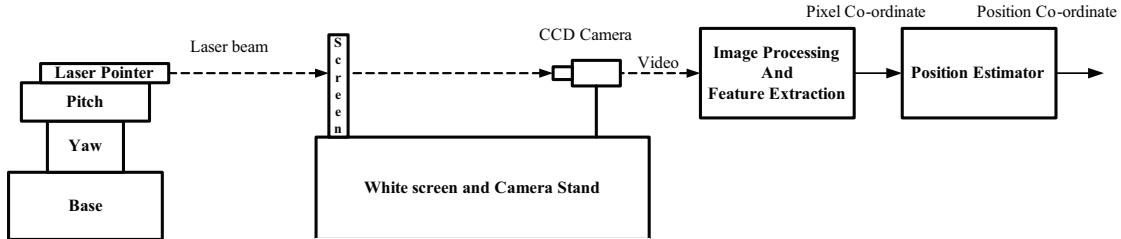


Figure 2.19: Image-based position sensor hardware and software block diagram

Image Tracker Algorithm

The image tracking algorithm block diagram is shown in Fig. 2.20. An RGB (Red, Green,

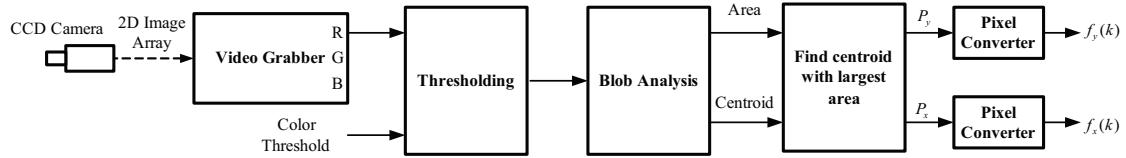


Figure 2.20: Image tracking block diagram

Blue) webcam using CMOS sensor is used to acquire two dimensional array of brightness values from three dimensional scene. A video grabber block in Simulink is used to interface with the hardware. The laser beam color is red. So we extract only red color information and using a threshold the image is enhanced. Then we use a blob analysis block in Simulink which outputs the area and centroid of the desired image. The centroid with maximum area is chosen as the image from which we use the image feature, pixel. In this way, the pixel information of the laser dot on the screen is extracted.

Camera Projection Model:Perspective Projection

The object to be observed is located in the full 3-D world, whereas the camera can only record its 2-D image. The full perspective transformation between world and image plane co-

ordinate frame is conventionally analysed using the pinhole camera model (Hutchinson et al., 1996). We assign the camera co-ordinate system with x_c and y_c axes forming a basis for the image plane as in Fig. 2.21. z_c is perpendicular to the image plane, along the optical axis (line-of-sight). The origin O is located at λ behind the image plane. Here,

λ = focal length of the camera,

$P(x_c, y_c, z_c)$ = co-ordinate of the object to be observed in the world co-ordinate,

$Q(v, w, \lambda)$ = co-ordinate of the point of intersection of the line connecting the object with the origin O_c .

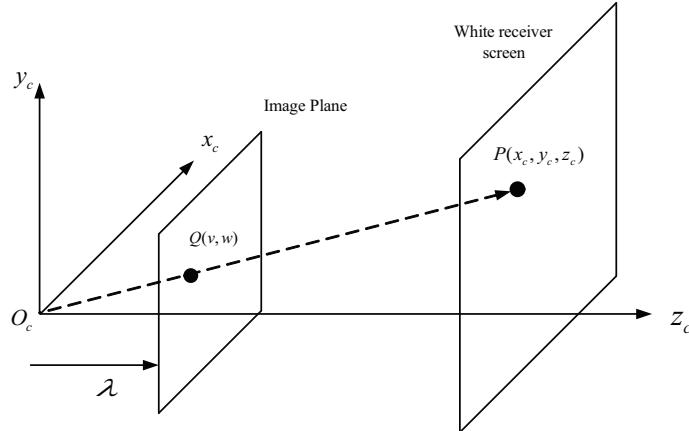


Figure 2.21: Laser dot coordinates in camera frame and after projection in image plane

The pinhole camera assumption allows us to write,

$$k \begin{bmatrix} x_c & y_c & z_c \end{bmatrix}^T = \begin{bmatrix} v & w & \lambda \end{bmatrix}^T, \quad (2.148)$$

and

$$\begin{bmatrix} v \\ w \end{bmatrix} = \frac{\lambda}{z_c} \begin{bmatrix} x_c \\ y_c \end{bmatrix}. \quad (2.149)$$

The co-ordinates in the image plane should then be quantized and the origin should be moved into the lower left corner to obtain pixel co-ordinates (P_x, P_y) . That gives,

$$\begin{bmatrix} -\frac{v}{s_x} \\ -\frac{w}{s_y} \end{bmatrix} = \begin{bmatrix} P_x - o_r \\ P_y - o_c \end{bmatrix}, \quad (2.150)$$

where s_x and s_y are the pixel dimensions and o_r and o_c are half the width and height of the image frame in pixels. To make the above equations simple, the pixelized information will be considered centered. Thus

$$\begin{bmatrix} -\frac{v}{s_x} \\ -\frac{w}{s_y} \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \end{bmatrix}. \quad (2.151)$$

Now we get the relation among pixel co-ordinates, pixel dimensions, and image plane co-ordinates.

Laser Pointer Motion and Interaction Matrix

In image-based visual servo control, the motion of the laser pointer changes to the image observed by the vision system. Thus, the specification of an image-based visual servo task involves determining an appropriate error function, such that when the task is achieved the error function will be zero. Although the error is defined in the image parameter space, the laser pointer control input is typically defined in pitch gimbal co-ordinates here. Therefore, it is necessary to relate the changes in the image feature parameter changes to the changes in the position of the laser pointer. This relationship is captured by image Jacobian or interaction matrix or feature sensitivity matrix. This matrix is derived for a general case in (Hutchinson et al., 1996) where the camera mounted on a robot hand observes a moving target. Therefore, both translational and rotational motion are involved in the derivation. This matrix is derived in (Hutchinson et al., 1996) as

$$\begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -\frac{\lambda}{z_c} & 0 & \frac{v}{z_c} & \frac{vw}{\lambda} & -\frac{\lambda^2+v^2}{\lambda} & w \\ 0 & -\frac{\lambda}{z_c} & \frac{w}{z_c} & \frac{\lambda^2+w^2}{\lambda} & -\frac{vw}{\lambda} & -v \end{bmatrix} \begin{bmatrix} v_{cx} \\ v_{cy} \\ v_{cz} \\ \omega_{cx} \\ \omega_{cy} \\ \omega_{cz} \end{bmatrix}, \quad (2.152)$$

where v and ω denote the translational and angular rate of the target expressed in camera frame. In our case, the laser pointer and the camera do not change position with respect to each other. Therefore, following the above relation, we shall first relate the image co-ordinate vector to the laser pointer angular rate vector expressed in camera frame. Then we shall find the rotational matrix between the laser pointer frame and the camera co-ordinate frame. Finally, we shall relate the image co-ordinate vector to the laser pointer angular rate vector expressed in laser pointer co-ordinate frame or pitch gimbal frame.

The image Jacobian or interaction matrix in our case would be with x and y components of angular rate in the camera frame,

$$\begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \frac{vw}{\lambda} & -\frac{\lambda^2+v^2}{\lambda} \\ \frac{\lambda^2+w^2}{\lambda} & -\frac{vw}{\lambda} \end{bmatrix} \begin{bmatrix} \omega_{cx} \\ \omega_{cy} \end{bmatrix}. \quad (2.153)$$

Now, the three components ω_{cx} , ω_{cy} , and ω_{cz} define the inertial angular rate of laser pointer expressed in camera frame, which is rotated with respect to pitch gimbal frame using a fixed rotation matrix. From Fig. 2.22 we write,

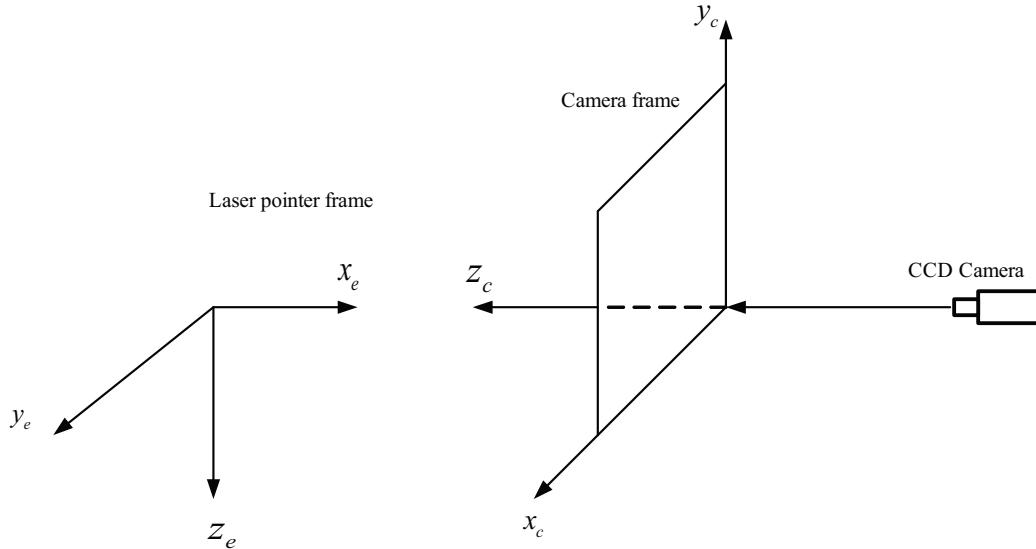


Figure 2.22: Relation between laser pointer frame and camera frame

$$\begin{bmatrix} \omega_{ex} \\ \omega_{ey} \\ \omega_{ez} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \omega_{cx} \\ \omega_{cy} \\ \omega_{cz} \end{bmatrix}. \quad (2.154)$$

If we want to express a point expressed in pitch gimbal frame into camera frame, then the transformation matrix is:

$$\begin{bmatrix} \omega_{cx} \\ \omega_{cy} \\ \omega_{cz} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_{ex} \\ \omega_{ey} \\ \omega_{ez} \end{bmatrix}. \quad (2.155)$$

Therefore interaction matrix is as follows using (2.153) and (2.155) :

$$\begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \frac{vw}{\lambda} & -\frac{\lambda^2+v^2}{\lambda} \\ \frac{\lambda^2+w^2}{\lambda} & -\frac{vw}{\lambda} \end{bmatrix} \begin{bmatrix} \omega_{ey} \\ -\omega_{ez} \end{bmatrix}. \quad (2.156)$$

That gives,

$$\begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \frac{vw}{\lambda} & \frac{\lambda^2+v^2}{\lambda} \\ \frac{\lambda^2+w^2}{\lambda} & \frac{vw}{\lambda} \end{bmatrix} \begin{bmatrix} \omega_{ey} \\ \omega_{ez} \end{bmatrix}. \quad (2.157)$$

By using brushless dc motor servo drive it is possible, at least partially, to affect the pitch gimbal angular rate by commanding its two components ω_{ey} and ω_{ez} . The relation between image features and the commanding velocity can be written as follows using (2.157):

$$\begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \frac{vw}{\lambda} & \frac{\lambda^2+v^2}{\lambda} \\ \frac{\lambda^2+w^2}{\lambda} & \frac{vw}{\lambda} \end{bmatrix} \begin{bmatrix} \omega_{ey}^* \\ \omega_{ez}^* \end{bmatrix}. \quad (2.158)$$

Linearization around the operating point

In order to develop insight to the model (2.157), we consider a situation where $w = 0$ (the observed object is vertically centered on the screen), the equation simplifies to:

$$\dot{v} = \frac{\lambda^2 + v^2}{\lambda} \omega_{ez}. \quad (2.159)$$

This is a nonlinear relationship between the angle and the line segment in the image plane. The focal length of the camera is far greater than the maximum value of v . The linear dynamics is

then

$$\dot{v} = \lambda \omega_{ez}. \quad (2.160)$$

The similar relationship is applicable for other co-ordinate:

$$\dot{w} = \lambda \omega_{ey}. \quad (2.161)$$

Co-ordinate frame for white receiver screen

We interpret pixel position into linear displacement on the white screen from the center of the white screen. For that, we need to define a coordinate system on the white screen first. Fig.

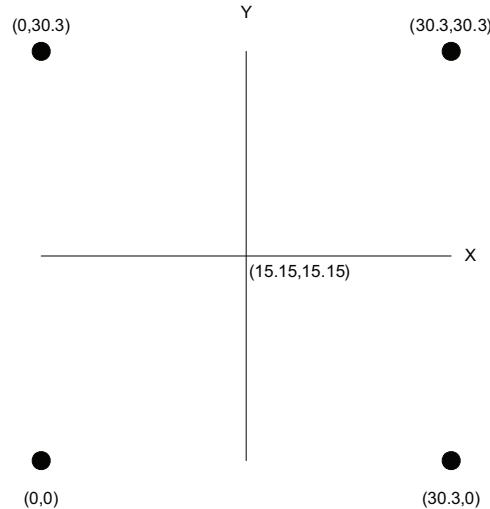


Figure 2.23: Coordinate system on white receiver screen

2.23 shows the defined coordinate system. The receiver white screen is a square screen of $30.3 \text{ cm} \times 30.3 \text{ cm}$ dimension and the screen resolution is 480×480 . The pixel coordinate in image plane is expressed into white receiver screen frame into cm unit. The lower leftmost corner is the point with co-ordinate $(0, 0)$ and the upper rightmost corner is the point is $(30.3, 30.3)$. The point of interest where the desired laser red dot on the receiver screen is $(15.15, 15.15)$. This is the origin of the co-ordinate system for this application. Therefore, in this case, $f_x^* = f_y^* = 15.15$. The maximum pixel error on the both sides of origin for both axes are 15.15 and -15.15 . The conversion factor between pixel and linear displacement from the center of the screen was

found using,

$$K_{pixel} = \frac{\delta D_y}{\delta P_y}, \quad (2.162)$$

where, K_{pixel} =conversion factor,

δD_y = distance between the two edge points on the screen along vertical axis,

δP_y =pixel difference between the two edge points.

$$f_y = K_{pixel} \times P_y \quad (2.163)$$

where P_y = new pixel position,

f_y = new pixel position for pitch axis expressed in cm

The similar rule is applied for the yaw gimbal.

Linear Model of Image-based Position Sensor

From (2.151), we get

$$\begin{bmatrix} -\frac{v}{s_x} \\ -\frac{w}{s_y} \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \end{bmatrix}. \quad (2.164)$$

Now, we shall interpret the image plane co-ordinate on the white receiver screen using (2.151) and (2.163) :

$$f_y = K_{pixel} \times P_y = -K_{pixel} \times \frac{w}{s_y}. \quad (2.165)$$

Taking derivative of (2.165), we get

$$\dot{f}_y(t) = -\frac{K_{pixel}}{s_y} \dot{w}(t) = -\frac{K_{pixel} \lambda}{s_y} \omega_{ey}(t) = C_y \omega_{ey}(t), \quad (2.166)$$

where C_y is a scale factor to convert angular velocity to position co-ordinate on the screen, and $C_y = -K_{pixel} \frac{\lambda}{s_y}$. This is for vertical position co-ordinate. For the horizontal position co-ordinate, we can write,

$$\dot{f}_x(t) = C_x \omega_{ez}(t), \quad (2.167)$$

where $C_x = -K_{pixel} \frac{\lambda}{s_x}$. Eq. (2.166) and (2.167) are the linearized dynamics of image-based position sensor.

We need to convert the reference position co-ordinate into reference speed command with some conversion factor. Then we can write:

$$f_y^* = C_{\omega EL} \omega_{mEL}^*, \quad (2.168)$$

$$f_x^* = C_{\omega AZ} \omega_{mAZ}^*, \quad (2.169)$$

where $C_{\omega EL}$ and $C_{\omega AZ}$ are the conversion factor between reference position co-ordinate command and reference speed command for both axis.

In the next section, we derive a reduced order gimbaled laser target system.

2.3 Derivation of a Reduced order Gimbaled Laser Target System

We described all the system components by transfer functions, state-space models, and look-up tables. It is possible to simulate this system in computer software such as in Matlab and Simulink. However, that would be a complex non-linear system to work with. To design a controller for that system would be difficult. A linear model will help utilize different control strategies to be applied on the system. Therefore, we shall reduce the complexity of the model by reducing the order, and derive a linear version of the model. We shall first obtain a simple first order approximation of a servo drive and brushless dc motor model. Then with that we shall cascade a two-axis gimbal model. We shall consider the sensors here as ideal sensor, i.e., the gain of the sensor will be unity.

Going back to our control design objective, we shall design the controller for the inner stabilization loop as shown in 2.24. The goal is to stabilize the angular velocity of the gimbaled laser target system so that the laser can point the target accurately.

2.3.1 Derivation of a Linear Servo Drive and Brushless DC Motor Model

We shall start by the equation describing the pitch gimbal. The moment of inertia of both the motor and the gimbal will be lumped together. Although, this derivation use pitch gimbal moment of inertia for building the model, later we shall see we can use the obtained model for yaw gimbal also.

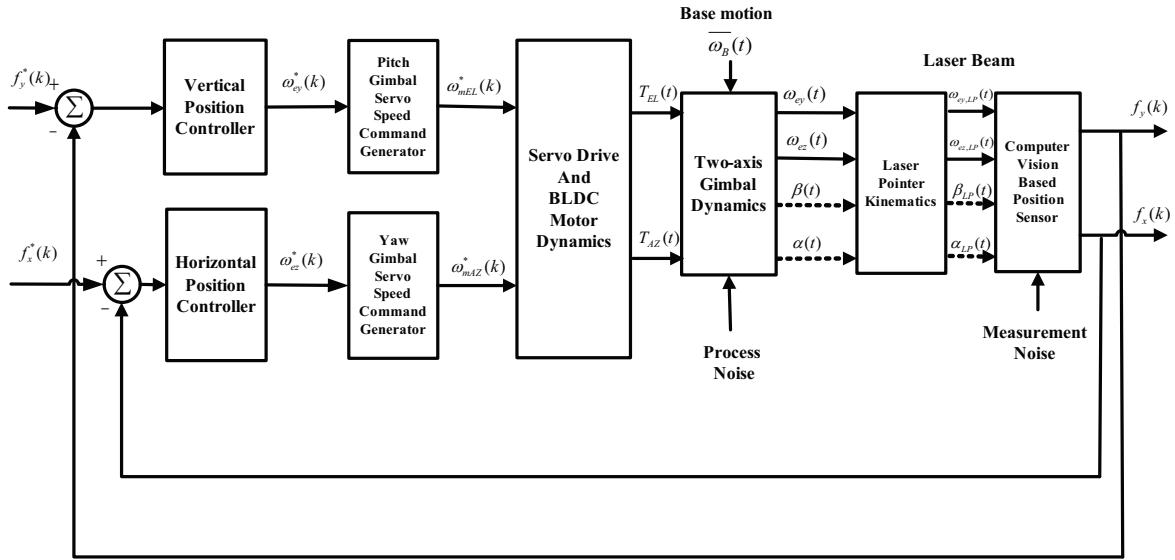


Figure 2.24: Architecture of the control system of a two-axis gimbaled laser target system

Pitch gimbal equation is given by (2.49)

$$J_{ey}\dot{\omega}_{ey}(t) = T_{EL}(t) - T_{Uly}(t) - T_{ef}(t) - T_{ew}(t) - T_D(t), \quad (2.170)$$

where $T_{ef}(t)$ = Friction torque,

$T_{ew}(t)$ = Cable restraint torque,

$T_{Ulx}(t), T_{Uly}(t), T_{Ulz}(t)$ = Mass imbalance torque about each gimbal axis,

$T_{ex}(t), T_{ez}(t)$ = Reaction torques exerted by pitch gimbal on yaw gimbal,

$T_{EL}(t)$ = Pitch axis control torque.

Friction torque is given by following (2.52),

$$T_{ef} = [T_{efc} + (T_{efs} - T_{efc})e^{-\left(\frac{\dot{\beta}(t)}{\dot{\beta}_S}\right)^2}]sgn(\dot{\beta}(t)) + K_{ef}\dot{\beta}(t), \quad (2.171)$$

where

T_{efc} = Coulomb friction torque,

T_{efs} = Stiction torque,

$\dot{\beta}_S$ = Stribeck angular velocity,

$sgn(\dot{\beta}(t))$ is defined as

$$sgn(\dot{\beta}(t)) = \begin{cases} \dot{\beta}(t) & \text{if } \dot{\beta}(t) > 0, \\ -\dot{\beta}(t) & \text{if } \dot{\beta}(t) < 0, \\ 0 & \text{if } \dot{\beta}(t) = 0. \end{cases}$$

From (2.171) we can write,

$$T_{ef,nonlinear}(t) = [T_{efc} + (T_{efs} - T_{efc})e^{-\left(\frac{\dot{\beta}(t)}{\dot{\beta}_S}\right)^2}] sgn(\dot{\beta}(t)), \quad (2.172)$$

$$\dot{\beta}(t) = \omega_{ey}(t) - \omega_{ay}(t), \quad (2.173)$$

$$\omega_{ay}(t) = -\omega_{bx}(t) \cos \alpha(t) + \omega_{by}(t) \sin \alpha(t). \quad (2.174)$$

The base is assumed to be non-rotating, then base angular velocities and angular acceleration become zero.

$$\dot{\beta}(t) = \omega_{ey}(t) + \omega_{bx}(t) \cos \alpha(t) - \omega_{by}(t) \sin \alpha(t), \quad (2.175)$$

$$\dot{\beta}(t) = \omega_{ey}(t). \quad (2.176)$$

Now (2.171) can be written as,

$$\begin{aligned} T_{ef} &= [T_{efc} + (T_{efs} - T_{efc})e^{-\left(\frac{\dot{\beta}(t)}{\dot{\beta}_S}\right)^2}] sgn(\dot{\beta}(t)) + K_{ef}\dot{\beta}(t), \\ &= T_{ef,nonlinear}(t) + K_{ef}\omega_{ey}(t). \end{aligned} \quad (2.177)$$

The inertial disturbance is given by (2.51),

$$T_D = \omega_{ez}\omega_{ex}(J_{ex} - J_{ez}) + D_{xy}(\dot{\omega}_{ex} + \omega_{ey}\omega_{ez}) + D_{yz}(\dot{\omega}_{ez} - \omega_{ex}\omega_{ey}) + D_{xz}(\omega_{ez}^2 - \omega_{ex}^2). \quad (2.178)$$

We assume that the pitch gimbal is axially symmetrical such that its spin axis is its principle axis of inertia. Also we assume that the gimbals are balanced. Then we can write, $D_{xy} = D_{yz} = D_{xz} = 0$. With this relation, (2.178) becomes,

$$T_D(t) = \omega_{ez}(t)\omega_{ex}(t)(J_{ex} - J_{ez}). \quad (2.179)$$

Further assuming that $J_{ex} = J_{ez}$, we find that $T_D(t) = 0$ (Ekstrand, 2001). Eq. (2.170) becomes,

$$J_{ey}\dot{\omega}_{ey}(t) = T_{EL}(t) - T_{Uly}(t) - [T_{ef,nonlinear}(t) + K_{ef}\dot{\beta}(t)] + T_{e\omega}(t).$$

For the non-linear terms we write, $T_{dist}(t) = -T_{Uly}(t) - T_{ef,nonlinear}(t) - T_{e,CR}(t) - T_{ew}(t)$.

Now we can write for the pitch gimbal,

$$J_{ey}\dot{\omega}_{ey}(t) + K_{ef}\omega_{ey}(t) = T_{EL}(t) + T_{dist}(t). \quad (2.180)$$

To deal with only linear terms, from (2.180) we neglect $T_{dist}(t)$ which consists of nonlinear inner axis disturbance terms. That gives us,

$$J_{ey}\dot{\omega}_{ey}(t) + K_{ef}\omega_{ey}(t) = T_{EL}(t). \quad (2.181)$$

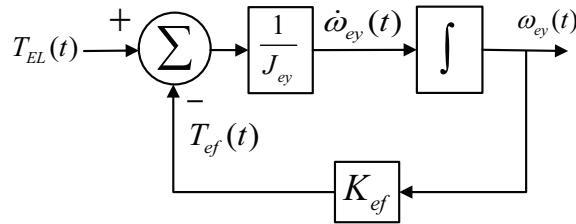


Figure 2.25: Block diagram of linear pitch gimbal

$$J_{ey}\frac{d}{dt}(\omega_{ey}(t)) = T_{EL}(t) - K_{ef}\omega_{ey}(t). \quad (2.182)$$

For brushless DC motor mechanical part, Eq. (2.103) can be written as,

$$J_m\frac{d}{dt}(\omega_m(t)) = T_e(t) - b\omega_m(t). \quad (2.183)$$

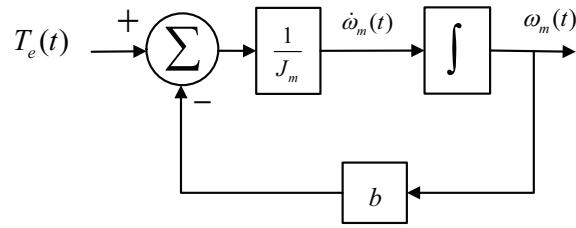


Figure 2.26: Block diagram of mechanical part of motor

Eq. (2.182) and Eq. (2.183) were derived assuming direct drive. However, The motor is connected to the load through a set of gears. The gears have a teeth ratio or gear ratio and can be treated as torque transformers. The gears are primarily used to amplify the torque on the

load side that is at a lower speed compared to the motor speed. The motor is designed to run at high speeds because it has been found that the higher the speed, the lower is the volume and size of the motor. The gears can be modelled from the two facts: firstly, the power handled by the gear is the same on both the sides; secondly, speed on each side is inversely proportional to its tooth number or the radius of the gear, i.e. $\omega_m r = \omega_{ey} R$, where r and R are the radii of gear attached to the motor and the gimbal. The gear ratio N is defined as: $N = \frac{R}{r}$. Then we can write $\omega_m = N\omega_{ey}$.

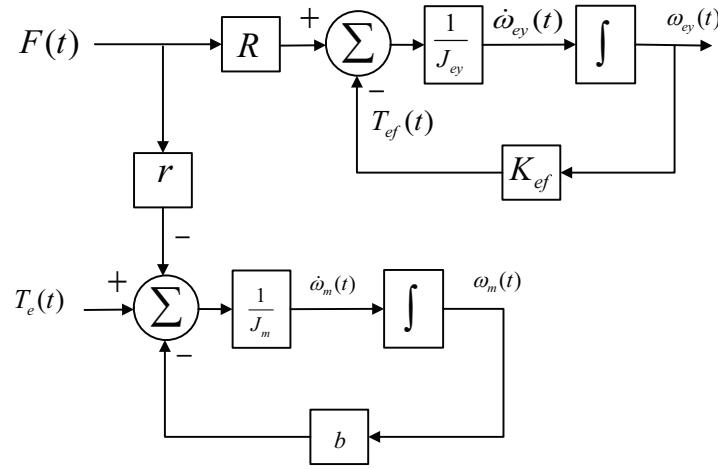


Figure 2.27: Block diagram of pitch gimbal and motor with gear

Therefore, we need to modify Eq. (2.182) and Eq. (2.183) since for a geared driven system, there is an interacting force between the gear of the motor and that of the gimbal. The equations will be as follows:

$$J_{ey} \frac{d}{dt}(\omega_{ey}(t)) = RF(t) - K_{ef}\omega_{ey}(t). \quad (2.184)$$

$$J_m \frac{d}{dt}(\omega_m(t)) = T_e(t) - rF(t) - b\omega_m(t). \quad (2.185)$$

Equating these two equations we obtain,

$$\begin{aligned}
\frac{R}{r} &= \frac{J_{ey} \frac{d}{dt}(\omega_{ey}(t)) + K_{ef}\omega_{ey}(t)}{-J_m \frac{d}{dt}(\omega_m(t)) - b\omega_m(t) + T_e(t)}, \\
J_{ey} \frac{d}{dt}(\omega_{ey}(t)) + K_{ef}\omega_{ey}(t) &= -N^2 J_{ey} \frac{d}{dt}(\omega_{ey}(t)) - N^2 b\omega_{ey}(t) + NT_e(t), \\
(J_{ey} + N^2 J_m) \frac{d}{dt}(\omega_{ey}(t)) + (K_{ef} + N^2 b)\omega_{ey}(t) &= T_{EL}(t), \\
J_t \frac{d}{dt}(\omega_{ey}(t)) + b_t\omega_{ey}(t) &= T_{EL}(t).
\end{aligned}$$

where $J_t = J_{ey} + N^2 J_m$ and $b_t = K_{ef} + N^2 b$. Eq. (2.186) is the model for pitch gimbal model combined with mechanical part of the motor with gear. The brushless dc motor is very similar to a dc machine (Krishnan, 2001). The similarity holds when only two phases are conducting during the interval where the rotor flux linkages are constant as seen from these phases and all the phases are electrically symmetric and balanced.

During two-phase conduction as shown in Fig. 2.10, the inverter output voltage, $V_a(s)$, is applied to the two phases having an impedance of

$$Z(s) = 2(R + Ls) = sL_a + R_a; \quad (2.186)$$

where $R_a = 2R$ and $L_a = 2L$,

R = stator resistance,

L = stator self inductance.

The voltage equation for the stator is given by,

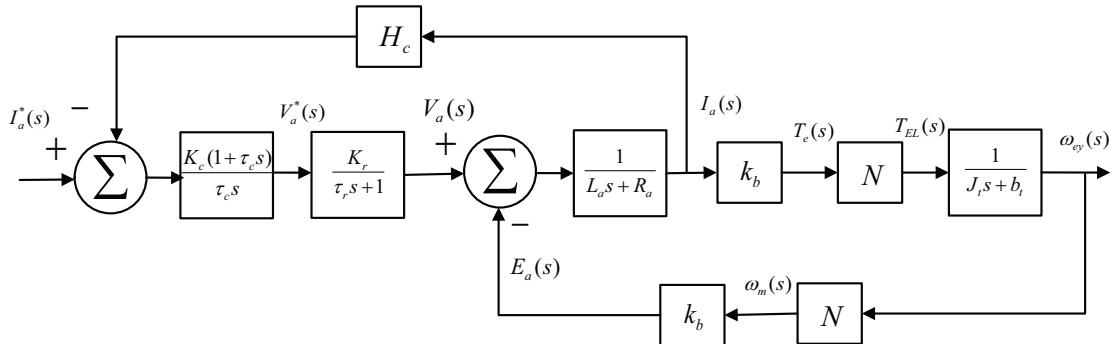


Figure 2.28: Block diagram of brushless DC motor with inner current control loop

$$V_a(s) = (sL_a + R_a)i_a(s) + E_a(s) - E_c(s), \quad (2.187)$$

where the last two terms are the induced emfs in phases a and c , respectively. But the induced emfs in phases a and c are equal and opposite during the regular operation of the drive scheme and given as,

$$E_a(s) = -E_c(s) = \lambda_p \omega_m, \quad (2.188)$$

which on substitution gives the stator voltage equation as,

$$V_a(s) = (sL_a + R_a)I_a(s) + 2\lambda_p \omega_m(s) = (sL_a + R_a)I_a(s) + k_b N \omega_{ey}(s), \quad (2.189)$$

where the emf constant for both the phases is combined into one constant as $k_b = 2\lambda_p$, V/rad/s. The electromagnetic torque for two phases combined is given by $T_e(s) = k_b N \omega_{ey}(s)$. The motor with an inner current control loop is shown in Fig. 2.28.

The machine contains an inner loop due to the induced emf. It is not physically seen as it is magnetically coupled. The inner current loop will cross this back emf loop, creating a complexity in the development of the model. The interactions of these loops can be decoupled by suitably redrawing the block diagram.

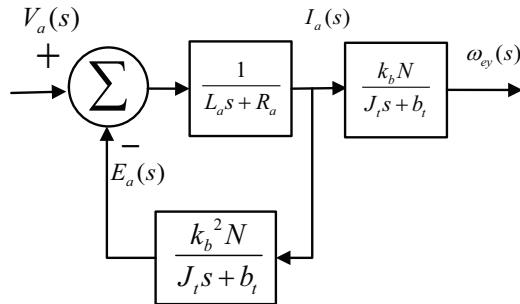


Figure 2.29: Block diagram of pitch gimbal and motor

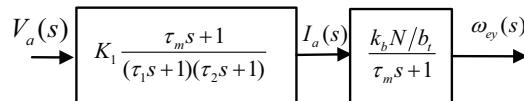


Figure 2.30: Redrawn block diagram of pitch gimbal and motor

The development of such a block diagram for the pitch gimbal and the motor is shown in Fig. 2.29 and 2.30.

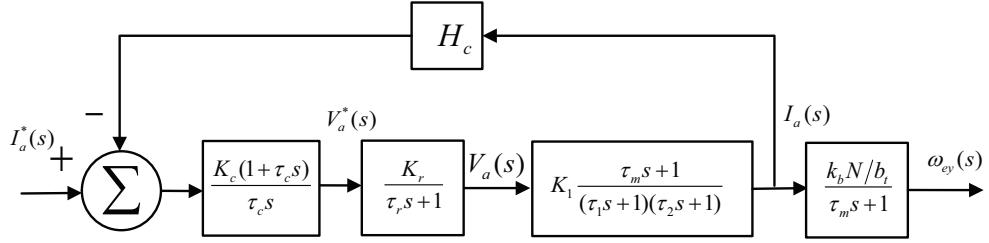


Figure 2.31: Redrawn Block diagram of brushless DC motor with inner current control loop

To decouple the inner current loop from the inherent induced emf loop, it is necessary to split the transfer function between the speed and voltage into two cascade transfer functions, between speed and phase current and between phase current and input voltage and represented as:

$$\frac{\omega_{ey}(s)}{V_a(s)} = \frac{\omega_{ey}(s)}{I_a(s)} \frac{I_a(s)}{V_a(s)}, \quad (2.190)$$

where

$$\frac{\omega_{ey}(s)}{I_a(s)} = \frac{k_b N}{J_t s + b_t} = \frac{\frac{k_b N}{b_t}}{\frac{J_t}{b_t} s + 1} = \frac{\frac{k_b N}{b_t}}{\tau_m s + 1}, \quad (2.191)$$

$$\begin{aligned} \frac{I_a(s)}{V_a(s)} &= \frac{\frac{1}{L_a s + R_a}}{1 + \frac{1}{L_a s + R_a} \frac{k_b^2 N}{J_t s + b_t}}, \\ &= \frac{J_t s + b_t}{(L_a s + R_a)(J_t s + b_t) + k_b^2 N}, \\ &= \frac{J_t s + b_t}{(J_t L_a s^2 + (J_t R_a + b_t L_a)s + (k_b^2 N + R_a b_t))}, \\ &= \frac{\frac{b_t(\tau_m s + 1)}{k_b^2 N + R_a b_t}}{\frac{1}{k_b^2 N + R_a b_t} [J_t L_a s^2 + (J_t R_a + b_t L_a)s + (k_b^2 N + R_a b_t)]}, \\ &= K_1 \frac{\tau_m s + 1}{(\tau_1 s + 1)(\tau_2 s + 1)}, \end{aligned} \quad (2.192)$$

where $\tau_m = \frac{J_t}{b_t}$, $K_1 = \frac{b_t(\tau_m s + 1)}{k_b^2 N + R_a b_t}$, $\tau_1, \tau_2 = \frac{1}{2} \frac{(J_t R_a + b_t L_a) \pm [(J_t R_a + b_t L_a)^2 - 4 J_t L_a (k_b^2 N + R_a b_t)]^{1/2}}{k_b^2 N + R_a b_t}$.

The inverter is represented as a first-order lag with a gain given by (Ramu, 2009),

$$\frac{V_a(s)}{V_a^*(s)} = \frac{K_r}{\tau_r s + 1}, \quad (2.193)$$

where $K_r = 0.65 \frac{V_{dc}}{V_{cm}}$, $\tau_r = \frac{1}{2 f_c}$. V_{dc} is the dc link voltage input to the inverter, V_{cm} is the maximum control voltage, and f_c is the switching (carrier) frequency of the inverter.

The current controller is represented as $\frac{K_c(\tau_c s + 1)}{\tau_c s}$. K_c and τ_c correspond to the gain and time constants of the controller. The gain of the current feedback is H_c . No significant filtering is required in most of the cases (Ramu, 2009).

It is seen that the current loop does not contain the inner induced emf loop because of the modification derived earlier. Now we shall design the current controller. The current control loop is shown in Figure . The loop gain function is,

$$\begin{aligned}\text{Loop Gain} &= \frac{K_c(\tau_c s + 1)}{\tau_c s} \times \frac{K_r}{\tau_r s + 1} \times K_1 \frac{\tau_m s + 1}{(\tau_1 s + 1)(\tau_2 s + 1)} \times H_c, \\ &= \left\{ \frac{K_c K_r K_1 H_c}{\tau_c} \right\} \frac{(\tau_c s + 1)(\tau_m s + 1)}{s(\tau_1 s + 1)(\tau_2 s + 1)(\tau_r s + 1)}. \end{aligned} \quad (2.194)$$

This is a fourth-order system and simplification is necessary to synthesize a controller without resorting to computer. Noting that τ_m is in the order of a second and in the vicinity of the gain crossover frequency, this approximation is valid: $1 + \tau_m s \approx \tau_m s$. Then the loop gain function becomes,

$$\text{Loop Gain} = \left\{ \frac{K_c K_r K_1 H_c \tau_m}{\tau_c} \right\} \frac{(\tau_c s + 1)}{(\tau_1 s + 1)(\tau_2 s + 1)(\tau_r s + 1)}. \quad (2.195)$$

Now the time constants in the denominator are seen to have the relationship $\tau_r < \tau_2 < \tau_1$. Eq. (2.195) can be reduced to second order to facilitate a simple controller synthesis by judiciously selecting $\tau_c = \tau_2$. Then the loop gain function is,

$$\text{Loop Gain} \approx \frac{K}{(\tau_1 s + 1)(\tau_r s + 1)}, \quad (2.196)$$

where $K = \frac{K_c K_r K_1 H_c \tau_m}{\tau_c}$.

The characteristic equation or denominator of the transfer function between the armature current and its command is:

$$(\tau_1 s + 1)(\tau_r s + 1) + K = \tau_1 \tau_r \left[s^2 + \left(\frac{\tau_1 + \tau_r}{\tau_1 \tau_r} \right) s + \frac{K + 1}{\tau_1 \tau_r} \right], \quad (2.197)$$

from which the natural frequency and the damping ratio is obtained as:

$$\omega_n^2 = \frac{K + 1}{\tau_1 \tau_r}, \quad (2.198)$$

$$\xi = \frac{\frac{\tau_1 + \tau_r}{\tau_1 \tau_r}}{2 \sqrt{\frac{K+1}{\tau_1 \tau_r}}} \quad (2.199)$$

For good dynamic performance, it is an accepted practice to have a damping ratio of 0.707.

Hence equating the damping ratio to 0.707, we obtain,

$$K + 1 = \frac{\left(\frac{\tau_1 + \tau_r}{\tau_1 \tau_r}\right)^2}{\frac{2}{\tau_1 \tau_r}}, \quad (2.200)$$

Realizing that $K \gg 1$, and $\tau_1 \gg \tau_r$, we get

$$K \approx \frac{\tau_1}{2\tau_r}. \quad (2.201)$$

This gives us $K_c = \frac{1}{2} \frac{\tau_1 \tau_c}{\tau_r} \left(\frac{1}{K_1 K_r H_c \tau_m} \right)$.

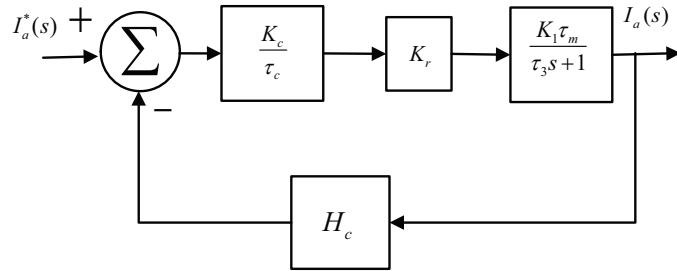


Figure 2.32: Block diagram of simplified current control loop

We shall further reduce the order of the current control loop to reduce the complexity of designing the rate controller. The current loop can be approximated by adding the time delay in the converter block to T1 of the motor and due to the cancellation of one motor pole with a zero of the current controller, the resulting current loop is shown in Fig. 2.32. Here the loop gain is $\frac{K_c K_r K_1 H_c \tau_m}{\tau_c} \frac{1}{\tau_3 s + 1}$ where $\tau_3 = \tau_1 + \tau_r$. Then the transfer function of the current I_a and the current command I_a^* is,

$$\begin{aligned} \frac{I_a(s)}{I_a^*(s)} &= \frac{\frac{K}{H_c(\tau_3 s + 1)}}{1 + \frac{K}{\tau_3 s + 1}}, \\ &= \frac{K_i}{\tau_i s + 1}, \end{aligned} \quad (2.202)$$

where current loop time constant is $\tau_i = \frac{\tau_3}{1+K}$ and current loop gain is $K_i = \frac{K}{H_c} \frac{1}{1+K}$. The resulting model of the current loop is a first-order system. Similarly, a speed control loop of the servo drive can be approximated with a first order. The transfer function between the current command I_a^* and the speed command ω_m^* is as follows:

$$\frac{I_a^*(s)}{\omega_m^*(s)} = \frac{K_s}{\tau_s + 1}, \quad (2.203)$$

where speed loop time constant is τ_s and speed loop gain is K_s . Cascading (2.202) and (2.203), we obtain the model for servo drive and brushless dc motor.

$$\frac{I_a(s)}{\omega_m^*(s)} = \frac{K_i}{\tau_i s + 1} \times \frac{K_s}{\tau_s + 1}. \quad (2.204)$$

2.3.2 Nonlinear Gimbaled Laser Target System Model

The two-axis gimbal dynamics for pitch (inner or elevation) axis and yaw gimbal (outer or azimuth) axis are based on rigid body dynamics. The detailed derivation was given in Section 2.2.1 and 2.2.1. The rigid body dynamics for pitch and yaw gimbal axes can be described as:

$$J_e \dot{\omega}_e(t) + (\bar{\omega}_e(t) \times J_e \bar{\omega}_e(t)) = \bar{L}_e(t), \quad (2.205)$$

$$J_a \dot{\omega}_a(t) + (\bar{\omega}_a(t) \times J_a \bar{\omega}_a(t)) + (\bar{L}_e(t))_A = \bar{L}_a(t), \quad (2.206)$$

where $\bar{L}_e(t)$ is

$$\bar{L}_e(t) = \begin{bmatrix} T_{ex}(t) \\ T_{EL}(t) \\ T_{ez}(t) \end{bmatrix} - \begin{bmatrix} T_{Ulx}(t) \\ T_{Uly}(t) \\ T_{Ulz}(t) \end{bmatrix} - \begin{bmatrix} 0 \\ T_{ef}(t) \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ T_{ew}(t) \\ 0 \end{bmatrix}, \quad (2.207)$$

with

$T_{ef}(t)$ = Friction torque,

$T_{e,fric}(t)$ = Non-linear friction torques,

$T_{ew}(t)$ = Cable restraint torque,

$T_{Ulx}(t), T_{Uly}(t), T_{Ulz}(t)$ = Mass imbalance torque about each gimbal axis,

$T_{ex}(t), T_{ez}(t)$ = Reaction torques exerted by pitch gimbal on yaw gimbal,

$T_{EL}(t)$ = Elevation or pitch control torque,

and $\bar{L}_a(t)$ is:

$$\bar{L}_a(t) = \begin{bmatrix} T_{ax}(t) \\ T_{ay}(t) \\ T_{AZ}(t) \end{bmatrix} - \begin{bmatrix} T_{Uox}(t) \\ T_{Uoy}(t) \\ T_{Uoz}(t) \end{bmatrix} - \begin{bmatrix} 0 \\ T_{af}(t) \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ T_{a\omega}(t) \\ 0 \end{bmatrix}, \quad (2.208)$$

with

$\bar{L}_a(t)$ = Sum of kinematic torques of yaw gimbal,

$T_{af}(t)$ = Friction torque,

$T_{a\omega}(t)$ = Cable restraint torque,

$T_{Uox}(t), T_{Uoy}(t), T_{Uoz}(t)$ = Yaw gimbal mass unbalance torques,

$T_{ax}(t), T_{ay}(t)$ = Reaction torques exerted by yaw gimbal on base,

$T_{AZ}(t)$ = Azimuth axis stabilization control torque.

$(\bar{L}_e(t))_a$ = Sum of kinematic torques of pitch gimbal with respect to yaw gimbal frame.

Friction torques are defined as (See details in Section 2.2.1):

$$T_{ef}(t) = K_{ef}\dot{\beta}(t) + T_{e,fric}(t), \quad (2.209)$$

$$T_{af}(t) = K_{af}\dot{\alpha}(t) + T_{a,fric}(t), \quad (2.210)$$

where, $\beta(t)$ = Pitch Euler angle,

K_{ef} = Viscous friction coefficient,

$T_{ef}(t)$ = Friction torque,

$T_{e,fric}$ = Non-linear friction torques,

$\alpha(t)$ = Yaw Euler angle,

K_{af} = Viscous friction coefficient,

$T_{a,fric}(t)$ = Non-linear friction torques.

Cable restraint torques are defined as (See details in Section 2.2.1):

$$T_{e\omega}(t) = K_{e\omega}\beta(t) + T_{e,CR}(t), \quad (2.211)$$

$$T_{a\omega}(t) = K_{a\omega}\alpha(t) + T_{a,CR}(t), \quad (2.212)$$

where, $K_{e\omega}$ = Cable restraint coefficient,

$T_{e\omega}(t)$ = Cable restraint torque,

$T_{e,CR}(t)$ = Nonlinear cable restraint torques,

$K_{a\omega}$ = Cable restraint coefficient,

$T_{a\omega}(t)$ = Cable restraint torque,

$T_{a,CR}(t)$ = Nonlinear cable restraint torques.

For the simplification of analysis, Kennedy (2003) assumed that the gimbal rotation axes are aligned with the principal axes of inertia so that the inertia tensor is diagonal. From (2.207), the pitch gimbal axis dynamics can be expressed as (detailed derivation in 2.2.1):

$$\begin{aligned} J_{ey}\dot{\omega}_{ey}(t) = & -K_{e\omega} \int_0^t \omega_{ey}(\tau)d\tau - K_{ef}\omega_{ey}(t) - (J_{ex} - J_{ez})\omega_{ax}(t)\omega_{ez}(t) \\ & +(J_{ex} - J_{ez})\omega_{ez}^2(t) \tan \beta(t) + T_{EL} + T_{eD,base}(t) + T_{e,dist}(t), \end{aligned} \quad (2.213)$$

with τ is the dummy time variable. $T_{eD,base}$ is pitch gimbal torque generated from the base motion is given by,

$$\begin{aligned} T_{eD,base} = & K_{ef}(\omega_{by}(t) \cos \alpha(t) - \omega_{bx}(t) \sin \alpha(t)) \\ & + K_{e\omega} \int_0^t (\omega_{by}(\tau) \cos \alpha(\tau) - \omega_{bx}(\tau) \sin \alpha(\tau)) d\tau, \end{aligned} \quad (2.214)$$

with τ is the dummy time variable. $T_{e,dist}(t)$ is pitch gimbal disturbance torque generated from mass imbalance, non-linear friction torque terms,

$$T_{e,dist}(t) = -T_{Uly}(t) - T_{e,fric}(t) - T_{e,CR}(t). \quad (2.215)$$

From (2.208), the yaw gimbal axis dynamics can be expressed as (detailed derivation in Section 2.2.1):

$$\begin{aligned} J_s\dot{\omega}_{ez}(t) = & -K_{a\omega} \int_0^t \omega_{ez}(\tau)d\tau + \cos \beta(t)K_{a\omega} \int_0^t \frac{\tan \beta(\tau)}{\cos \beta(\tau)} \omega_{ez}(\tau)d\tau - K_{af}\omega_{ez}(t) \\ & - g_y(t)\omega_{ey}(t)\omega_{ax}(t) - g_{yz}(t)\omega_{ey}(t)\omega_{ez}(t) + T_{AZ} \cos \beta(t) + T_{aD,base} \cos \beta(t) + T_{a,dist} \cos \beta(t). \end{aligned} \quad (2.216)$$

with τ is the dummy time variable. J_s includes moment of inertia terms with the functions of Euler angle $\beta(t)$.

$$J_s = J_{az} + J_{ez} \cos^2 \beta(t) + J_{ex} \sin^2 \beta(t), \quad (2.217)$$

$g_y(t)$ and $g_{yz}(t)$ are two nonlinear terms depends on β and inertia tensor elements:

$$g_y(t) = (J_{ey} - J_{ex}) \cos \beta(t) - \frac{1}{\cos \beta(t)} (J_{az} + J_{ex} \sin^2 \beta(t)), \quad (2.218)$$

$$g_{yz}(t) = (J_{az} + J_{ex}) \tan \beta(t) - (J_{ez} - J_{ex}) \cos \beta(t) \sin \beta(t). \quad (2.219)$$

$T_{aD,base}(t)$ is yaw gimbal torque generated from the base motion,

$$\begin{aligned} T_{aD,base}(t) &= (J_{az} + J_{ex})(\dot{\omega}_{bx}(t) \cos \alpha(t) - \dot{\omega}_{by}(t) \sin \alpha(t) + \omega_{ay}(t) \omega_{ax}(t) \tan \beta(t) \\ &\quad + \omega_{ay}(t) \omega_{bz}(t)) \sin \beta(t) + K_{af}(\omega_{ax}(t) \sin \beta(t) + \omega_{bz}(t) \cos \beta(t)) \\ &\quad + K_{a\omega}(\cos \beta(t)) \int_0^t (\omega_{ax}(\tau) \tan \beta(\tau) + \omega_{bz}(\tau)) d\tau, \end{aligned} \quad (2.220)$$

$T_{a,dist}(t)$ includes yaw gimbal disturbance torque generated from mass imbalance, non-linear friction torque terms, non-linear cable restraint torque terms,

$$T_{a,dist}(t) = -T_{Uoz}(t) - T_{a,fric}(t) - T_{a,CR}(t). \quad (2.221)$$

Now, the gimbal is driven by servo drive with servo drive with a brushless dc motor. Therefore, we shall link servo drive variable with gimbal variable. Using subscript for pitch and yaw axis, we write the relationship between control torques (gimbal variable) and armature currents (servo drive variable) as:

$$T_{EL}(t) = k_b N i_{aEL}(t), \quad (2.222)$$

$$T_{AZ}(t) = k_b N i_{aAZ}(t). \quad (2.223)$$

With above relationships, we write (2.213) and (2.216) as:

$$\begin{aligned} J_{ey}\dot{\omega}_{ey}(t) &= -K_{e\omega} \int_0^t \omega_{ey}(\tau) d\tau - K_{ef}\omega_{ey}(t) - (J_{ex} - J_{ez})\omega_{ax}(t)\omega_{ez}(t) \\ &\quad + (J_{ex} - J_{ez})\omega_{ez}^2(t) \tan \beta(t) + k_b N i_{aEL}(t) + T_{eD,base}(t) + T_{e,dist}(t), \end{aligned} \quad (2.224)$$

and

$$\begin{aligned}
J_s \dot{\omega}_{ez}(t) = & -K_{a\omega} \int_0^t \omega_{ez}(\tau) d\tau + \cos \beta(t) K_{a\omega} \int_0^t \frac{\tan \beta(\tau)}{\cos \beta(\tau)} \omega_{ez}(\tau) d\tau \\
& - K_{af} \omega_{ez}(t) - g_y(t) \omega_{ey}(t) \omega_{ax}(t) - g_{yz}(t) \omega_{ey}(t) \omega_{ez}(t) \\
& + k_b N i_{aAZ}(t) \cos \beta(t) + T_{aD,base}(t) \cos \beta(t) + T_{a,dist}(t) \cos \beta(t).
\end{aligned} \quad (2.225)$$

We shall use (2.224) and (2.225) to describe two-axis gimbal dynamics in our system. The inputs to the model are armature currents to the motor $i_{aEL}(t)$, $i_{aAZ}(t)$ and the outputs are the gimbal angular rates $\omega_{ey}(t)$ and $\omega_{ez}(t)$.

A first order model was derived earlier for current loop of the servo drive. The transfer function is:

$$\frac{I_a(s)}{I_a^*(s)} = \frac{K_i}{1 + s\tau_i}, \quad (2.226)$$

where I_a is the armature current of the motor and the I_a^* is the reference current command. K_i is the gain and τ_i is the time constant. Using subscript for pitch and yaw axis, we write (2.226) in differential equation form as:

$$\frac{d}{dt}(i_{aEL}(t)) = \left(-\frac{1}{\tau_i} \right) i_{aEL}(t) + \left(\frac{K_{iEL}}{\tau_{iEL}} \right) i_{aEL}^*(t). \quad (2.227)$$

$$\frac{d}{dt}(i_{aAZ}(t)) = \left(-\frac{1}{\tau_i} \right) i_{aAZ}(t) + \left(\frac{K_{iAZ}}{\tau_{iAZ}} \right) i_{aAZ}^*(t). \quad (2.228)$$

A first order model is used to model speed filter to convert reference speed command to the reference current command for the servo drive. The transfer function is:

$$\frac{I_a^*(s)}{\omega_m^*(s)} = \frac{K_s}{1 + s\tau_s}, \quad (2.229)$$

where ω_m^* is the reference speed of the motor and the I_a^* is the reference current command. K_s is the gain and τ_s is the time constant. Using subscript for pitch and yaw axis, we write (2.229) in differential equation form as:

$$\frac{d}{dt}(i_{aEL}^*(t)) = \left(-\frac{1}{\tau_s} \right) i_{aEL}^*(t) + \left(\frac{K_{sEL}}{\tau_{sEL}} \right) \omega_{mEL}^*(t). \quad (2.230)$$

$$\frac{d}{dt}(i_{aAZ}^*(t)) = \left(-\frac{1}{\tau_s}\right) i_{aAZ}^*(t) + \left(\frac{K_{sAZ}}{\tau_{sAZ}}\right) \omega_{mAZ}^*(t). \quad (2.231)$$

We express the reference speed command $\omega_{mEL}^*(t), \omega_{mAZ}^*(t)$ in terms of vertical and horizontal co-ordinates. Therefore we use (2.168) and (2.169) with (2.230) and (2.231) and get,

$$i_{aEL}^*(t) = \left(-\frac{1}{\tau_s}\right) i_{aEL}^*(t) + \left(\frac{K_{sEL} C_{\omega EL}}{\tau_{sEL}}\right) f_y^*(t), \quad (2.232)$$

$$i_{aAZ}^*(t) = \left(-\frac{1}{\tau_s}\right) i_{aAZ}^*(t) + \left(\frac{K_{sAZ} C_{\omega AZ}}{\tau_{sAZ}}\right) f_x^*(t). \quad (2.233)$$

The laser pointer is mounted on the gimbal. There is a linear relationship between laser pointer angular velocity and position co-ordinate on image-based sensor. For detailed derivation see Section 2.2.5. The linear relationships are from (2.166) and (2.167) as follows:

$$\dot{f}_y(t) = C_y \omega_{ey}(t), \quad (2.234)$$

$$\dot{f}_x(t) = C_x \omega_{ez}(t), \quad (2.235)$$

where $f_x(t), f_y(t)$ are the horizontal and vertical position co-ordinates, C_x, C_y are the conversion scale factors from angular rate to position co-ordinates, and $\omega_{ey}(t), \omega_{ez}(t)$ are the gimbal angular rate.

Integrating both sides of (2.234) and (2.235), assuming zero initial condition, we find,

$$\int_0^t \omega_{ey}(\tau) d\tau = \frac{1}{C_y} f_y(t). \quad (2.236)$$

$$\int_0^t \omega_{ez}(\tau) d\tau = \frac{1}{C_x} f_x(t). \quad (2.237)$$

Now from (2.224), (2.225), (2.227), (2.228), (2.232), (2.233), (2.234), (2.235), (2.236), (2.237), we write following equations and then we shall present those in state-space form:

$$\dot{f}_y(t) = C_y \omega_{ey}(t), \quad (2.238)$$

$$\begin{aligned}\dot{\omega}_{ey}(t) &= \left(-\frac{K_{e\omega}}{C_y J_{ey}}\right) f_y(t) + \left(-\frac{K_{ef}}{J_{ey}}\right) \omega_{ey}(t) + \left(-\frac{J_{ex} - J_{ez}}{J_{ey}} \omega_{ax}(t)\right) \omega_{ez}(t) \\ &+ \left(\frac{J_{ex} - J_{ez}}{J_{ey}}\right) \omega_{ez}^2(t) \tan \beta(t) + \left(\frac{k_b N}{J_{ey}}\right) i_{aEL}(t) + \frac{1}{J_{ey}} T_{eD,base}(t) + \frac{1}{J_{ey}} T_{e,dist}(t), \quad (2.239)\end{aligned}$$

$$\dot{i}_{aEL}(t) = \left(-\frac{1}{\tau_{iEL}}\right) i_{aEL}(t) + \left(\frac{K_{iEL}}{\tau_{iEL}}\right) i_{aEL}^*(t), \quad (2.240)$$

$$\dot{i}_{aEL}^*(t) = \left(-\frac{1}{\tau_{sEL}}\right) i_{aEL}^*(t) + \left(\frac{K_{sEL} C_{\omega EL}}{\tau_{sEL}}\right) f_y^*, \quad (2.241)$$

$$\dot{f}_x(t) = C_x \omega_{ez}(t), \quad (2.242)$$

$$\begin{aligned}\dot{\omega}_{ez}(t) &= \left(\frac{-K_{a\omega}}{C_x J_s}\right) f_x(t) + \left(\frac{-K_{af}}{J_s}\right) \omega_{ez}(t) - \cos \beta(t) K_{a\omega} \int_0^t \frac{\tan \beta(\tau)}{\cos \beta(\tau)} \frac{f_x(\tau)}{C_x} d\tau + \\ &\left(\frac{-g_y(t)}{J_s} \omega_{ax}(t)\right) \omega_{ey}(t) + \left(\frac{-g_{yz}(t)}{J_s}\right) \omega_{ey}(t) \omega_{ez}(t) + \left(\frac{k_b N}{J_s}\right) \cos \beta(t) i_{aAZ} \\ &+ \frac{1}{J_s} T_{aD,base}(t) \cos \beta(t) + \frac{1}{J_s} T_{a,dist}(t) \cos \beta(t), \quad (2.243)\end{aligned}$$

$$\dot{i}_{aAZ}(t) = \left(-\frac{1}{\tau_{iAZ}}\right) i_{aAZ}(t) + \left(\frac{K_{iAZ}}{\tau_{iAZ}}\right) i_{aAZ}^*(t), \quad (2.244)$$

$$\dot{i}_{aAZ}^*(t) = \left(-\frac{1}{\tau_{sAZ}}\right) i_{aAZ}^*(t) + \left(\frac{K_{sAZ} C_{\omega AZ}}{\tau_{sAZ}}\right) f_x^*(t). \quad (2.245)$$

Writing (2.238)-(2.245), we obtain the non-linear time varying model of the Gimbaled Laser Target System as follows:

$$\dot{X}(t) = A(t)X(t) + B(t)U(t) + F(X, t) + G(T_{D,base}(t) + T_{dist}(t)), \quad (2.246)$$

where, $A(t)$ is the state matrix,

$$A(t) = \begin{bmatrix} 0 & C_y & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{K_{e\omega}}{C_y J_{ey}} & -\frac{K_{ef}}{J_{ey}} & \frac{k_b N}{J_{ey}} & 0 & -J_{\Delta z}\omega_{ax}(t) & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_{iEL}} & \frac{K_{iEL}}{\tau_{iEL}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_{sEL}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_x & 0 & 0 \\ 0 & -g_y(t)\frac{\omega_{ax}(t)}{J_s} & 0 & 0 & -\frac{K_{a\omega}}{J_s} & -\frac{K_{af}}{J_s} & \frac{k_b N}{J_s} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_{iAZ}} & \frac{K_{iAZ}}{\tau_{iAZ}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_{sAZ}} \end{bmatrix}, \quad (2.247)$$

J_s is the moment of inertia term including Euler angle $\beta(t)$,

$$J_s = J_{az} + J_{ez} \cos^2 \beta(t) + J_{ex} \sin^2 \beta(t). \quad (2.248)$$

$g_y(t)$ is a nonlinear terms depends on $\beta(t)$ and inertia tensor elements:

$$g_y(t) = (J_{ey} - J_{ex}) \cos \beta(t) - \frac{1}{\cos \beta(t)} (J_{az} + J_{ex} \sin^2 \beta(t)), \quad (2.249)$$

$\beta(t)$ is defined from (2.14) as:

$$\dot{\beta}(t) = \omega_{ey}(t) - (\omega_{by}(t) \cos \alpha(t) + \omega_{bx}(t) \sin \alpha(t)). \quad (2.250)$$

$\alpha(t)$ is defined from (2.11) as:

$$\dot{\alpha}(t) = \frac{1}{\cos \beta(t)} \omega_{ez}(t) - \omega_{ax}(t) \tan \beta(t). \quad (2.251)$$

$\omega_{ax}(t)$ defined from (2.11) as:

$$\omega_{ax}(t) = \omega_{bx}(t) \cos \alpha(t) + \omega_{by}(t) \sin \alpha(t). \quad (2.252)$$

C_y, C_x	scale factor for the conversion of the pitch and yaw angular velocity to vertical and horizontal position co-ordinate of laser dot on the position sensor screen,
$K_{e\omega}, K_{a\omega}$	pitch and yaw cable restraint co-efficient,
K_{ef}, K_{af}	pitch and yaw viscous friction co-efficient,
J_{ex}, J_{ey}, J_{ez}	pitch inertia tensor diagonal elements,
J_{ax}, J_{ay}, J_{az}	yaw inertia tensor diagonal elements,
K_{iEL}, K_{iAZ}	servo-motor current loop gain for pitch and yaw gimbal,
τ_{iEL}, τ_{iAZ}	servo-motor current loop time constant for pitch and yaw gimbal,
K_{sEL}, K_{sAZ}	servo-motor speed loop gain for pitch and yaw gimbal,
τ_{sEL}, τ_{sAZ}	servo-motor speed loop time constant for pitch and yaw gimbal,
$\omega_{ax}(t)$	x component of yaw gimbal angular rate,
k_b	damping co-efficient of motor,
N	gear ratio.,
$J_{\Delta z} = \frac{J_{ex}-J_{ez}}{J_{ey}}$	the moment of inertia ratio written in concise form.

$X(t)$ is the state vector,

$$X(t) = \begin{bmatrix} f_y(t) & \omega_{ey}(t) & i_{aEL}(t) & i_{aEL}^*(t) & f_x(t) & \omega_{ez}(t) & i_{aAZ}(t) & i_{aAZ}^*(t) \end{bmatrix}^T, \quad (2.253)$$

with The initial conditions are chosen arbitrarily, the horizontal and vertical position co-ordinates are at (2, 2); all other states are zero initially.

$$X(0) = \begin{bmatrix} 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}^T, \quad (2.254)$$

$f_y(t)$	vertical position co-ordinate on screen,
$\omega_{ey}(t)$	y component of pitch gimbal angular rate,
$i_{aEL}(t)$	armature current for servo-motor drive for pitch gimbal,
$i_{aEL}^*(t)$	reference current command for servo-motor drive for pitch gimbal,
$f_x(t)$	horizontal position co-ordinate on screen,
$\omega_{ez}(t)$	cross-elevation gimbal angular rate,
$i_{aAZ}(t)$	reference current command for servo-motor drive for pitch gimbal,
$i_{aAZ}^*(t)$	reference current command for servo-motor drive for yaw gimbal,
$X(0)$	the initial condition.

B is the input matrix,

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{K_{sEL}C_{\omega EL}}{\tau_{sEL}} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{K_{sAZ}C_{\omega AZ}}{\tau_{sAZ}} \cos \beta(t) \end{bmatrix}, \quad (2.255)$$

with K_{sEL}, K_{sAZ} = pitch and yaw gimbal servo drive speed filter gain,

τ_{sEL}, τ_{sAZ} = pitch and yaw gimbal servo drive speed filter time constant,

$C_{\omega EL}, C_{\omega AZ}$ = scale factor for converting vertical and horizontal position co-ordinate into reference speed command for pitch and yaw gimbal servo drive,

U is the input,

$$U(t) = \begin{bmatrix} f_y^*(t) \\ f_x^*(t) \end{bmatrix}, \quad (2.256)$$

with $f_y^*(t), f_x^*(t)$ = vertical and horizontal reference position co-ordinates,
state-dependent and time-varying non-linear term,

$$F(X, t) = \begin{bmatrix} 0 \\ J_{\Delta z} \omega_{ez}^2(t) \tan \beta(t) \\ 0 \\ 0 \\ 0 \\ -K_{aw}(\cos \beta(t)) \int_0^t \frac{\tan \beta(\tau)}{\cos \beta(\tau)} \frac{f_x(\tau)}{C_x} d\tau + \frac{g_{yz}(t) \omega_{ey}(t)}{J_s} \\ 0 \\ 0 \end{bmatrix}, \quad (2.257)$$

$g_{yz}(t)$ includes non-linear terms $\beta(t)$ and inertia tensor elements.

$$g_{yz}(t) = (J_{az} + J_{ex}) \tan \beta(t) - (J_{ez} - J_{ex}) \cos \beta(t) \sin \beta(t), \quad (2.258)$$

G matrix includes the inverse of moment of inertia terms,

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{J_{ey}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{J_s} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.259)$$

$T_{D,base}(t)$ is the torque vector generated in pitch and yaw gimbal due to base motion,

$$T_{D,base}(t) = \begin{bmatrix} 0 & T_{eD,base}(t) & 0 & 0 & 0 & T_{aD,base}(t) \cos \beta(t) & 0 & 0 \end{bmatrix}^T, \quad (2.260)$$

$T_{eD,base}$ is the pitch gimbal torque generated from the base motion,

$$T_{eD,base} = K_{ef}(\omega_{by}(t) \cos \alpha(t) - \omega_{bx}(t) \sin \alpha(t)) + K_{ew} \int_0^t (\omega_{by}(\tau) \cos \alpha(\tau) - \omega_{bx}(t)(\tau) \sin \alpha(\tau)) d\tau, \quad (2.261)$$

$T_{aD,base}$ is the yaw gimbal torque generated from the base motion,

$$\begin{aligned} T_{aD,base}(t) = & (J_{az} + J_{ex})(\dot{\omega}_{bx}(t) \cos \alpha(t) - \dot{\omega}_{by}(t) \sin \alpha(t) + \omega_{ay}(t)\omega_{ax}(t) \tan \beta(t) \\ & + \omega_{ay}(t)\omega_{bz}(t)) \sin \beta(t) + K_{af}(\omega_{ax}(t) \sin \beta(t) + \omega_{bz}(t) \cos \beta(t)) \\ & + K_{aw}(\cos \beta(t)) \int_0^t (\omega_{ax}(\tau) \tan \beta(\tau) + \omega_{bz}(\tau)) d\tau, \end{aligned} \quad (2.262)$$

$T_{dist}(t)$ is disturbance torque vector due to non-linear friction, cable restraint, and mass imbalance torques,

$$T_{dist} = \begin{bmatrix} 0 & T_{e,dist}(t) & 0 & 0 & 0 & T_{a,dist}(t) \cos \beta(t) & 0 & 0 \end{bmatrix}^T, \quad (2.263)$$

$T_{e,dist}(t)$ includes pitch gimbal disturbance torque generated from mass imbalance, non-linear friction torque terms, non-linear cable restraint torque terms,

$$T_{e,dist}(t) = -T_{Uoy}(t) - T_{e,fric}(t) - T_{e,CR}(t). \quad (2.264)$$

$T_{a,dist}(t)$ includes yaw gimbal disturbance torque generated from mass imbalance, non-linear friction torque terms, non-linear cable restraint torque terms,

$$T_{a,dist}(t) = -T_{Uoz}(t) - T_{a,fric}(t) - T_{a,CR}(t). \quad (2.265)$$

The outputs are $(f_y(t), f_x(t))$ vertical and horizontal position co-ordinate on screen. The output equation is,

$$Y(t) = CX(t), \quad (2.266)$$

where,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (2.267)$$

Eq. (2.246) and (2.266) are the non-linear model time-varying model of the Gimbaled Laser Target System.

In summary, we developed a non-linear model of the Gimbaled Laser Target System. We started with a high level diagram, and then described each component of the system. Finally, we integrated the components and simplified the model. In the next chapter, we shall linearize the non-linear model, and develop a stochastic linear model of the Gimbaled Laser Target System.

CHAPTER 3

LINEAR GIMBALED LASER TARGET SYSTEM

3.1 Overview

In the previous chapter, a non-linear model of the Gimbaled Laser Target System (GLTS) in Eq. (2.246) was developed. In this chapter, the non-linear model will be linearized around nominal operating points. Many modern methods of control system design are based on linear algebra. A linear model is suitable for the application of those methods. The statistical control method approached in this thesis also requires a linear model. Finally, we present a stochastic linear model for the gimbaled laser target system.

3.2 Derivation of Linear Gimbaled Laser Target Model

3.2.1 Assumptions

This test setup replicates a transmission from a satellite to a ground receiver on the earth. Fig. 3.1 shows a scenario where the laser pointer imitates the transmission equipment on the satellite and the screen imitates the ground receiving antenna on earth.

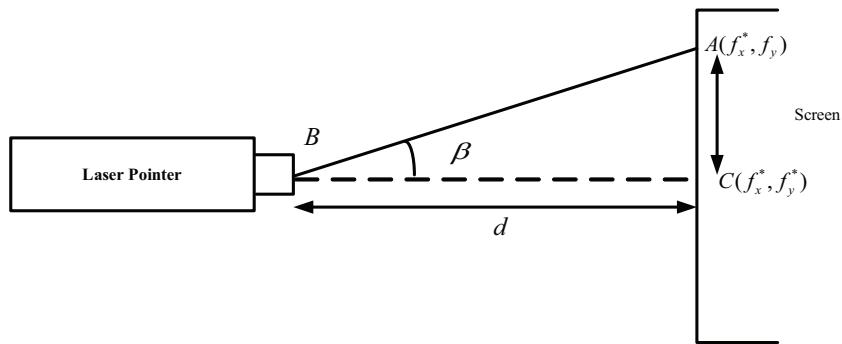


Figure 3.1: Side view of the laser pointing device and the screen

In Fig. 3.1, d is the distance from point B on laser pointer to $C(f_x^*, f_y^*)$ on screen. C is

the center of the screen with co-ordinates (f_x^*, f_y^*) . Let, A be a point with co-ordinates (f_x^*, f_y) . The use of the notations A, B, C should not be confused with the state-space form A, B, C notations. The interpretation of these notations are only for this section. A tangent relationship can be found from the distance of the laser pointer and the screen BC , and the distance travelled by laser dot from point C to point A , CA .

$$\begin{aligned}\tan \beta &= \frac{CA}{BC}, \\ \beta &= \tan^{-1} \frac{f_y - f_y^*}{d}.\end{aligned}\quad (3.1)$$

The maximum value of $f_y - f_y^*$ can be 15.15 cm since the distance from the center to the edge of the position sensor is 15.15 cm. The value of d can be varied. Geo-stationary orbit is approximately 36,000 km above the earth's equator. With a scale-down factor of 10^7 meters, this distance becomes 3.6 m or 360 cm. This distance is physically realizable in a laboratory settings. Using this value of the distance, β is found to be 2.5° or 0.0436 radian from (3.1). Even if the half of 360 cm is considered, β is 4.8° or 0.0839 radian. This is why, the angle β is assumed to be small (less than 5°).

3.2.2 Nonlinear Torque term due to Base Angular Rate, $T_{D,base}$

The details about the reference frames and notation were given in Section 2.2.1. However, for the ease of following the derivation, we repeat some notations here. For the angular velocities of frame b (base), a (yaw) and e (pitch) respectively are from (2.8),

$$Frame\ b : \bar{\omega}_b(t) = \begin{bmatrix} \omega_{bx}(t) & \omega_{by}(t) & \omega_{bz}(t) \end{bmatrix}^T, \quad (3.2)$$

$$Frame\ a : \bar{\omega}_a(t) = \begin{bmatrix} \omega_{ax}(t) & \omega_{ay}(t) & \omega_{az}(t) \end{bmatrix}^T, \quad (3.3)$$

$$Frame\ e : \bar{\omega}_e(t) = \begin{bmatrix} \omega_{ex}(t) & \omega_{ey}(t) & \omega_{ez}(t) \end{bmatrix}^T. \quad (3.4)$$

where $\omega_{bx}, \omega_{by}, \omega_{bz}$ are the components of base angular velocity ω_b in frame b , and similarly for the other vectors.

The relationship between base angular velocity and yaw angular velocity are from (2.11):

$$\begin{bmatrix} \omega_{ax}(t) \\ \omega_{ay}(t) \\ \omega_{az}(t) \end{bmatrix} = \begin{bmatrix} \omega_{bx}(t) \cos \alpha(t) + \omega_{by}(t) \sin \alpha(t) \\ -\omega_{bx}(t) \sin \alpha(t) + \omega_{by}(t) \cos \alpha(t) \\ \omega_{bz}(t) + \dot{\alpha}(t) \end{bmatrix}. \quad (3.5)$$

The relationship between yaw angular velocity and pitch angular velocity are from (2.11):

$$\begin{bmatrix} \omega_{ex}(t) \\ \omega_{ey}(t) \\ \omega_{ez}(t) \end{bmatrix} = \begin{bmatrix} \omega_{ax}(t) \cos \beta(t) - \omega_{az}(t) \sin \beta(t) \\ \omega_{ay}(t) + \dot{\beta}(t) \\ \omega_{ax}(t) \sin \beta(t) + \omega_{az}(t) \cos \beta(t) \end{bmatrix}. \quad (3.6)$$

Now, the two-axis gimbal is mounted on rigid base platform with no angular motion. Therefore,

$$\omega_{bx} = \omega_{by} = \omega_{bz} = 0. \quad (3.7)$$

Using (3.7), we obtain from (3.5),

$$\begin{bmatrix} \omega_{ax}(t) \\ \omega_{ay}(t) \\ \omega_{az}(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha} \end{bmatrix}. \quad (3.8)$$

and from (3.6) we get,

$$\begin{bmatrix} \omega_{ex}(t) \\ \omega_{ey}(t) \\ \omega_{ez}(t) \end{bmatrix} = \begin{bmatrix} -\omega_{az}(t) \sin \beta(t) \\ \dot{\beta}(t) \\ \omega_{az}(t) \cos \beta(t) \end{bmatrix}. \quad (3.9)$$

Now, $T_{D,base}(t)$ is the torque vector generated in pitch and yaw gimbal due to base motion in the non-linear model (2.246). From (2.263), (2.214), and (2.220) we know,

$$T_{D,base}(t) = \begin{bmatrix} 0 & T_{eD,base}(t) & 0 & 0 & 0 & T_{aD,base}(t)(\cos \beta(t)) & 0 & 0 \end{bmatrix}^T. \quad (3.10)$$

$T_{eD,base}(t)$ is pitch gimbal torque generated from the base motion,

$$T_{eD,base}(t) = K_{ef}(\omega_{by} \cos \alpha(t) - \omega_{bx}(t) \sin \alpha(t)) + K_{ew} \int_0^t (\omega_{by}(\tau) \cos \alpha(\tau) - \omega_{bx}(\tau) \sin \alpha(\tau)) d\tau, \quad (3.11)$$

$T_{aD,base}$ is yaw gimbal torque generated from the base motion,

$$\begin{aligned} T_{aD,base}(t) = & (J_{az} + J_{ex})(\dot{\omega}_{bx}(t) \cos \alpha(t) - \dot{\omega}_{by}(t) \sin \alpha(t) + \omega_{ay}(t)\omega_{ax}(t) \tan \beta(t) \\ & + \omega_{ay}(t)\omega_{bz}(t)) \sin \beta(t) + K_{af}(\omega_{ax}(t) \sin \beta(t) + \\ & \omega_{bz}(t) \cos \beta(t)) + K_{a\omega}(\cos \beta(t)) \int_0^t (\omega_{ax}(\tau) \tan \beta(\tau) + \omega_{bz}(\tau)) d\tau, \end{aligned} \quad (3.12)$$

Using (3.7), we find $T_{eD,base} = T_{aD,base} = 0$ which in turn make $T_{D,base}$,

$$T_{D,base}(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \quad (3.13)$$

3.2.3 Non-linear Terms in State Matrix $A(t)$

$A(t)$ is the state matrix in the non-linear model (2.246). From (2.247), we know,

$$A(t) = \begin{bmatrix} 0 & C_y & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{K_{e\omega}}{C_y J_{ey}} & -\frac{K_{ef}}{J_{ey}} & -\frac{k_b N}{J_{ey}} & 0 & -J_{\Delta z}\omega_{ax}(t) & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_{iEL}} & \frac{K_{iEL}}{\tau_{iEL}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_{sEL}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_x & 0 & 0 \\ 0 & -g_y(t)\frac{\omega_{ax}(t)}{J_s} & 0 & 0 & -\frac{K_{a\omega}}{J_s} & -\frac{K_{af}}{J_s} & -\frac{k_b N}{J_s} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_{iAZ}} & \frac{K_{iAZ}}{\tau_{iAZ}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_{sAZ}} \end{bmatrix}, \quad (3.14)$$

where the non-linear terms are $-J_{\Delta z}\omega_{ax}(t)$ and $g_y(t)\frac{\omega_{ax}(t)}{J_s}$. Now from (3.8), we know $\omega_{ax} = 0$. That makes these non-linear terms $-J_{\Delta z}\omega_{ax}(t)$ and $g_y(t)\frac{\omega_{ax}(t)}{J_s}$ in $A(t)$ zero. In this way, the

non-linear time-varying $A(t)$ becomes linear time-invariant A .

$$A = \begin{bmatrix} 0 & C_y & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{K_{e\omega}}{C_y J_{ey}} & -\frac{K_{ef}}{J_{ey}} & -\frac{k_b N}{J_{ey}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_{iEL}} & \frac{K_{iEL}}{\tau_{iEL}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_{sEL}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_x & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{K_{a\omega}}{J_s} & -\frac{K_{af}}{J_s} & -\frac{k_b N}{J_s} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_{iAZ}} & \frac{K_{iAZ}}{\tau_{iAZ}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_{sAZ}} \end{bmatrix}. \quad (3.15)$$

3.2.4 Non-linear Term $F(X, t)$

$F(X, t)$ is state-dependent and time-varying non-linear term from (2.257),

$$F(X, t) = \begin{bmatrix} 0 & J_{\Delta z} \omega_{ez}^2(t) \tan \beta(t) & 0 & 0 & 0 & -\cos \beta(t) K_{a\omega} \int_0^t \frac{\tan \beta(\tau)}{\cos \beta(\tau)} \frac{f_x(\tau)}{C_x} d\tau + \frac{g_{yz}(t) \omega_{ey}(t)}{J_s} & 0 & 0 \end{bmatrix}^T. \quad (3.16)$$

Around this operating point, $\sin \beta \approx \tan \beta \approx 0$. This non-linear term has little effect on the system dynamics around the operating point. Therefore, $F(X, t)$ is ignored.

3.2.5 Disturbance Torque term T_{dist}

T_{dist} is the disturbance torque generated due to friction torques, cable restraint torques, and mass imbalance torques for pitch and yaw gimbal. We model this disturbance torque as stationary Weiner process $w(t)$.

3.2.6 Linear Model of Gimbaled Laser Target System

Now with all the assumptions made in the previous section, we change nonlinear model in (2.246) in a linear state-space form as follows:

$$\dot{x}(t) = Ax(t) + Bu(t) + Gw(t), \quad (3.17)$$

Then we write (3.17) in stochastic differential vector form,

$$dx(t) = (Ax(t) + Bu(t))dt + Gdw(t), \quad (3.18)$$

where, the state matrix,

$$A = \begin{bmatrix} 0 & C_y & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{K_{ew}}{C_y J_{ey}} & -\frac{K_{ef}}{J_{ey}} & \frac{k_b N}{J_{ey}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_{iEL}} & \frac{K_{iEL}}{\tau_{iEL}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_{sEL}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_x & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{K_{aw}}{J_s} & -\frac{K_{af}}{J_s} & \frac{k_b N}{J_s} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_{iAZ}} & \frac{K_{iAZ}}{\tau_{iAZ}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_{sAZ}} \end{bmatrix}, \quad (3.19)$$

the state vector,

$$x(t) = \begin{bmatrix} f_y(t) & \omega_{ey}(t) & i_{aEL}(t) & i_{aEL}^*(t) & f_x(t) & \omega_{ez}(t) & i_{aAZ}(t) & i_{aAZ}^*(t) \end{bmatrix}^T, \quad (3.20)$$

with

$f_y(t)$ = vertical position co-ordinate on screen,

$\omega_{ey}(t)$ = pitch gimbal angular rate,

$i_{aEL}(t)$ = armature current for servo-motor drive for pitch gimbal,

$i_{aEL}^*(t)$ = reference current command for servo-motor drive for pitch gimbal,

$f_x(t)$ = horizontal position co-ordinate on screen,

$\omega_{ez}(t)$ = cross-elevation gimbal angular rate,

$i_{aAZ}(t)$ = reference current command for servo-motor drive for pitch gimbal,

$i_{aAZ}^*(t)$ = reference current command for servo-motor drive for yaw gimbal,

and $x(0)$ is the initial condition. The initial conditions are chosen arbitrarily, the horizontal and vertical position co-ordinates are at (2,2); all other states are zero initially. the initial conditions,

$$x(0) = \begin{bmatrix} 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}^T, \quad (3.21)$$

the input matrix,

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{K_{sEL}C_{\omega EL}}{\tau_{sEL}} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{K_{sAZ}C_{\omega AZ}}{\tau_{sAZ}} \end{bmatrix}, \quad (3.22)$$

with K_{sEL}, K_{sAZ} = pitch and yaw gimbal servo drive speed filter gain,

τ_{sEL}, τ_{sAZ} = pitch and yaw gimbal servo drive speed filter time constant,

$C_{\omega EL}, C_{\omega AZ}$ = scale factor for converting vertical and horizontal position co-ordinate into reference speed command for pitch and yaw gimbal servo drive,

$u(t)$ is the input,

$$u(t) = \begin{bmatrix} f_y^*(t) \\ f_x^*(t) \end{bmatrix}, \quad (3.23)$$

with $f_y^*(t), f_x^*(t)$ = vertical and horizontal reference position co-ordinates. $w(t)$ is the process noise. The noise is introduced by the disturbance torque present in system dynamics. It is a vector of Wiener process or Brownian motion. $dw(t)$ is a Gaussian random process with zero

mean and covariance matrix Wdt .

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{J_{ey}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{J_s} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.24)$$

The outputs are $(f_y(t), f_x(t))$ vertical and horizontal position co-ordinate on screen. The output equation is,

$$y(t) = Cx(t) + v(t), \quad (3.25)$$

where,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \quad (3.26)$$

$v(t)$ is the measurement noise. The noise is introduced during image processing. We model $v(t)$ as Weiner process. Eq. (3.18) and (3.25) are the linear model of the Gimbaled Laser Target System. We shall now decouple this model for pitch and yaw gimbal. The pitch Gimbaled Laser Target System model will be indicated by ‘EL’ subscript, and the yaw Gimbaled Laser Target System model will be indicated by ‘AZ’ subscript.

Linear Pitch GLTS Model

$$dx_{EL}(t) = (A_{EL}x_{EL}(t) + B_{EL}u_{EL}(t))dt + G_{EL}dw_{EL}(t),$$

(3.27)

where, the state matrix,

$$A_{EL} = \begin{bmatrix} 0 & C_y & 0 & 0 \\ -\frac{K_e\omega}{C_y J_{ey}} & -\frac{K_e f}{J_{ey}} & \frac{k_b N}{J_{ey}} & 0 \\ 0 & 0 & -\frac{1}{\tau_{iEL}} & \frac{K_{iEL}}{\tau_{iEL}} \\ 0 & 0 & 0 & -\frac{1}{\tau_{sEL}} \end{bmatrix}, \quad (3.28)$$

the state vector,

$$x_{EL}(t) = \begin{bmatrix} f_y(t) & \omega_{ey}(t) & i_{aEL}(t) & i_{aEL}^*(t) \end{bmatrix}^T, \quad (3.29)$$

with

$f_y(t)$ = vertical position co-ordinate on screen,

$\omega_{ey}(t)$ = pitch gimbal angular rate,

$i_{aEL}(t)$ = armature current for servo-motor drive for pitch gimbal,

$i_{aEL}^*(t)$ = reference current command for servo-motor drive for pitch gimbal.

$x_{EL}(0)$ is the initial condition. The initial condition is chosen arbitrarily. The vertical position co-ordinate is chosen as 2 cm; all other states are zero initially. the initial conditions,

$$x_{EL}(0) = \begin{bmatrix} 2 & 0 & 0 & 0 \end{bmatrix}^T, \quad (3.30)$$

the input matrix,

$$B_{EL} = \begin{bmatrix} 0 & 0 & 0 & \frac{K_{sEL} C_{\omega EL}}{\tau_{sEL}} \end{bmatrix}^T. \quad (3.31)$$

with K_{sEL} = pitch servo drive speed filter gain,

τ_{sEL} = pitch gimbal servo drive speed filter time constant,

$C_{\omega EL}$ = scale factor for converting vertical position co-ordinate into reference speed command for pitch gimbal servo drive,

$u_{EL}(t)$ is the input,

$$u_{EL}(t) = \begin{bmatrix} f_y^*(t) \end{bmatrix}, \quad (3.32)$$

with $f_y^*(t)$ = vertical reference position co-ordinate. $w_{EL}(t)$ is the process noise. It is a scalar of Wiener process or Brownian motion. $dw_{EL}(t)$ is a Gaussian random process with zero mean and covariance matrix $W dt$.

$$G_{EL} = \begin{bmatrix} 0 & \frac{1}{J_{ey}} & 0 & 0 \end{bmatrix}^T, \quad (3.33)$$

The output is $f_y(t)$ vertical position co-ordinate on screen. The output equation is,

$$\boxed{y_{EL}(t) = C_{EL}x_{EL}(t) + v_{EL}(t)}, \quad (3.34)$$

where,

$$C_{EL} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}. \quad (3.35)$$

$v_{EL}(t)$ is the measurement noise. The noise is introduced during image processing. We model $v_{EL}(t)$ as Weiner process. The statistics (expectation and covariance) of the initial condition, process noise and measurement noise are as follows:

$$\begin{aligned} E[x_{EL}(0)] &= \hat{x}_{EL0}, E\{[x_{EL}(0) - \hat{x}_{EL}(0)][x_{EL}(0) - \hat{x}_{EL}(0)]^T\} = P(0), \\ E[w_{EL}(t)] &= 0, E[w_{EL}(t)w_{EL}^T(\tau)] = W\delta(t - \tau), \\ E[v_{EL}(t)] &= 0, E[v_{EL}(t)v_{EL}^T(\tau)] = V\delta(t - \tau), \\ E[w_{EL}(t)v_{EL}^T(\tau)] &= 0. \end{aligned}$$

Eq. (3.27) and (3.34) are the linear model of the Pitch Gimbaled Laser Target System.

Linear Yaw GLTS Model

$$\boxed{dx_{AZ}(t) = (A_{AZ}x_{AZ}(t) + B_{AZ}u_{AZ}(t))dt + G_{AZ}dw_{AZ}(t)}, \quad (3.36)$$

where, the state matrix,

$$A_{AZ} = \begin{bmatrix} 0 & C_x & 0 & 0 \\ -\frac{K_a\omega}{C_x J_s} & -\frac{K_af}{J_s} & \frac{k_bN}{J_s} & 0 \\ 0 & 0 & -\frac{1}{\tau_{iAZ}} & \frac{K_{iAZ}}{\tau_{iAZ}} \\ 0 & 0 & 0 & -\frac{1}{\tau_{sAZ}} \end{bmatrix}, \quad (3.37)$$

the state vector,

$$x_{AZ}(t) = \begin{bmatrix} f_x(t) & \omega_{ez}(t) & i_{aAZ}(t) & i_{aAZ}^*(t) \end{bmatrix}^T, \quad (3.38)$$

with

$f_x(t)$ = horizontal position co-ordinate on screen,

$\omega_{ez}(t)$ = yaw gimbal angular rate,

$i_{aAZ}(t)$ = armature current for servo-motor drive for yaw gimbal,

$i_{aAZ}^*(t)$ = reference current command for servo-motor drive for yaw gimbal,

and $x_{AZ}(0)$ is the initial condition. The initial condition is chosen arbitrarily, the horizontal 2 cm; all other states are zero initially. the initial condition,

$$x_{AZ}(0) = \begin{bmatrix} 2 & 0 & 0 & 0 \end{bmatrix}^T, \quad (3.39)$$

the input matrix,

$$B_{AZ} = \begin{bmatrix} 0 & 0 & 0 & \frac{K_{sAZ}C_{\omega AZ}}{\tau_{sAZ}} \end{bmatrix}^T, \quad (3.40)$$

with K_{sAZ} = yaw gimbal servo drive speed filter gain,

τ_{sAZ} = yaw gimbal servo drive speed filter time constant,

$C_{\omega AZ}$ = scale factor for converting horizontal position co-ordinate into reference speed command for yaw gimbal servo drive,

$u_{AZ}(t)$ is the input,

$$u_{AZ}(t) = \begin{bmatrix} f_x^*(t) \end{bmatrix}, \quad (3.41)$$

with $f_x^*(t)$ = horizontal reference position co-ordinates. $w_{AZ}(t)$ is the process noise. It is a scalar of Wiener process or Brownian motion. $dw_{AZ}(t)$ is a Gaussian random process with zero mean and covariance matrix Wdt .

$$G_{AZ} = \begin{bmatrix} 0 & \frac{1}{J_s} & 0 & 0 \end{bmatrix}^T, \quad (3.42)$$

The output is $f_x(t)$ horizontal position co-ordinate on screen. The output equation is,

$$\boxed{y_{AZ}(t) = C_{AZ}x_{AZ}(t) + v_{AZ}(t)}, \quad (3.43)$$

where,

$$C_{AZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \quad (3.44)$$

$v_{AZ}(t)$ is the measurement noise. The noise is introduced during image processing. We model $v_{AZ}(t)$ as Weiner process. The statistics of initial condition, process noise, and measurement noise are as follows:

$$E[x_{AZ}(0)] = \hat{x}_{EL0}, E\{[x_{AZ}(0) - \hat{x}_{AZ}(0)][x_{AZ}(0) - \hat{x}_{AZ}(0)]^T\} = P(0),$$

$$E[w_{AZ}(t)] = 0, E[w_{AZ}(t)w_{AZ}^T(\tau)] = W\delta(t - \tau),$$

$$E[v_{AZ}(t)] = 0, E[v_{AZ}(t)v_{AZ}^T(\tau)] = V\delta(t - \tau),$$

$$E[w_{AZ}(t)v_{AZ}^T(\tau)] = 0.$$

Eq. (3.36) and (3.43) are the linear model of the Yaw Gimbaled Laser Target System.

In summary, in this chapter, we developed a stochastic linear model for gimbaled laser target system. In the next chapter, we discussed PID, LQG, and MCV control methods.

CHAPTER 4

CONTROL METHODOLOGIES

4.1 Overview

In the previous chapter, a linear model for Gimbaled Laser Target system was derived. In this chapter, we shall discuss the control methodologies we are going to adopt to control the model. The control methodologies are PID (Proportional-Integral-Derivative) Control, LQG (Linear Quadratic Gaussian) Control, and MCV (Minimal Cost Variance) Control for full-state feedback and output-feedback case.

4.2 PID Control Approach

PID controllers have been most widely used in industry for their simplicity and good control performance (Sung et al., 2009). In this section, the structure of PID controller is explained and PID controller tuning is discussed.

4.2.1 PID Controller Structure

The PID controller input is the error signal, and the output is the control signal. The error signal is the difference between the set-point and the process output. The control signal is the input signal for the process model or plant model. PID controllers are composed of three parts: Proportional (P) part, Integral (I) part, Derivative (D) part. They are described mathematically as follows:

$$u_P(t) = K_p e(t), \quad (4.1)$$

$$u_I(t) = K_I \int e(\tau) d\tau, \quad (4.2)$$

$$u_D(t) = K_D \frac{d}{dt}(e(t)), \quad (4.3)$$

where

$u_P(t)$ =proportional controller output,

$u_I(t)$ =integral controller output,

$u_D(t)$ =derivative controller output,

$e(t)$ = error input,

K_P = proportional controller gain,

K_I = integral controller gain,

K_D = derivative controller gain.

The output of the PID controller is the sum of the above mentioned three parts:

$$\begin{aligned} u(t) &= u_P(t) + u_I(t) + u_D(t) \\ &= K_p e(t) + K_I \int e(\tau) d\tau + K_D \frac{d}{dt}(e(t)). \end{aligned} \quad (4.4)$$

The transfer function of the PID controller $G_c(s)$ is:

$$G_c(s) = K_P + \frac{K_I}{s} + K_D s. \quad (4.5)$$

Therefore, PID controller has three tuning parameters, K_P , K_I , and K_D .

4.2.2 PID Controller Tuning

There are numerous strategies to tune the parameters of PID controllers (Åström and Hägglund, 2006). However, the discussion of those strategies is out of scope of this thesis. In this section, Matlab's PID Tuner is discussed, which is used for PID control simulation. Typical tuning objective includes:

- Closed-loop stability: The closed-loop system output remains bounded for bounded input.
- Adequate performance: The closed-loop system tracks reference changes and suppresses disturbances as rapidly as possible. The larger the loop bandwidth (the first frequency at which the open-loop gain is unity), the faster the controller responds to changes in the reference or disturbances in the loop.

- Adequate robustness: The loop design has enough phase margin and gain margin to allow for modeling errors or variations in system dynamics.

The algorithm used in Matlab for tuning PID controllers meets these objectives by automatically tuning the PID gains to balance performance (response time) and robustness (stability margins). By default, the algorithm chooses a crossover frequency (loop bandwidth) based upon the plant dynamics, and designs for a target phase margin of 60 degree. If the bandwidth or phase margin is changed using the sliders in the PID Tuner Graphical User Interface, the algorithm computes PID gains that best meet those targets.

Matlab PID Tuner uses a slightly different form of PID controller. In practical applications, the pure derivative action is never used, due to the “derivative kick” produced in the control signal for a step input, and to the undesirable noise amplification (Xue et al., 2007). It is usually replaced by a first-order low pass filter. Thus, the transfer function of the approximate PID controller can be written as:

$$G_c(s) = K_P + \frac{K_I}{s} + K_D \frac{s}{\frac{s}{N_p} + 1}, \quad (4.6)$$

where N_p is the time constant of the low pass filter used with derivative gain. For a large value of N_p , the above controller becomes PID controller. Now, there are four tuning parameters K_P , K_I , K_D , and N_p in the PID controller.

4.3 Statistical Control Approach

4.3.1 Full State Feedback Case

Preliminaries

We consider a stochastic linear time invariant dynamic system modelled on $[t_0, t_f]$. The dynamic process model is given by,

$$dx(t) = Ax(t)dt + Bu(t)dt + Gdw(t), \quad x(t_0) = x_0, \quad t \in [t_0, t_f] \quad (4.7)$$

The statistics of the disturbance inputs, and the initial conditions are assumed to be known.

$x(t) \in \mathbb{R}^n$	an n -dimensional state vector at time t ,
$u(t) \in \mathbb{R}^m$	an m -dimensional control action vector at time t ,
$w(t) \in \mathbb{R}^p$	an p -dimensional disturbance input vector at time t ,
x_0	the initial condition,
$A \in \mathcal{C}([t_0, t_F]; \mathbb{R}^{n \times n})$	a real continuous matrix of $n \times n$,
$B \in \mathcal{C}([t_0, t_F]; \mathbb{R}^{n \times m})$	a real continuous matrix of $m \times n$,
$G \in \mathcal{C}([t_0, t_F]; \mathbb{R}^{n \times p})$	a real continuous matrix of $n \times p$.

- $w(t)$ is a white, zero-mean Gaussian random process with the expected value and covariance of

$$E[w(t)] = 0, \quad (4.8)$$

$$E[w(t)w^T(\tau)] = W\delta(t - \tau). \quad (4.9)$$

W is the spectral density matrix which is symmetric and positive semi-definite. The Dirac delta function $\delta(t - \tau)$ guarantees the whiteness property.

- $x(t_0)$ has a known deterministic value. The expected value and the correlation of increment of the initial conditions are given by,

$$E[x(t_0)] = \tilde{x}_0, \quad (4.10)$$

$$E[(x(t_0) - \tilde{x}_0)(x(t_0) - \tilde{x}_0)^T] = P_0. \quad (4.11)$$

$x(t_0)$ is independent of $w(t)$ which means,

$$E[x(t_0)w(t)] = 0, \quad \forall t. \quad (4.12)$$

An integral quadratic form random cost $J : \mathcal{C}([t_0, t_F]; \mathbb{S}^n) \mapsto \mathbb{R}^+$ such that

$$J = \int_{t_0}^{t_F} (x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau))d\tau, \quad (4.13)$$

where $Q \in \mathcal{C}([t_0, t_F]; \mathbb{R}^{n \times n})$ is positive semi-definite; $R \in \mathcal{C}([t_0, t_F]; \mathbb{R}^{m \times m})$ is positive definite.

Linear Quadratic Gaussian Control

LQG method optimizes the mean, which is the first cumulant cost criterion (Anderson and Moore, 1990). The cost criterion for full state feedback LQG control for infinite time horizon case is given by,

$$J_{LQG} = \lim_{t_F \rightarrow \infty} \frac{1}{t_F} E \{ J(t_F) \}, \quad (4.14)$$

Assuming $R > 0$, (A, B) is stabilizable, and (\sqrt{Q}, A) is detectable. Then the full state optimal LQG controller for the infinite-time horizon is given by

$$u(t) = -R^{-1}B^T \mathcal{P}x(t) = -K_{LQG}x(t), \quad (4.15)$$

where \mathcal{P} is the solution of the algebraic Riccati equation:

$$A^T \mathcal{P} + \mathcal{P}A - \mathcal{P}BR^{-1}B^T \mathcal{P} + Q = 0. \quad (4.16)$$

In our case, we want the output to track a constant step command $r(t) = r_0 \cdot 1(t)$, where r_0 is the center of position sensor (f_x^*, f_y^*) . Therefore, we modify the control law of (4.15) to

$$u(t) = -R^{-1}B^T \mathcal{P}x(t) + r(t) = -K_{LQG}x(t) + r(t), \quad (4.17)$$

For horizontal and vertical case the control laws are respectively,

$$u_{AZ}(t) = -K_{LQG,AZ}x_{AZ}(t) + f_x^* \cdot 1(t), \quad (4.18)$$

$$u_{EL}(t) = -K_{LQG,EL}x_{EL}(t) + f_y^* \cdot 1(t). \quad (4.19)$$

Minimal Cost Variance Control

MCV control minimizes the second cumulant, variance, while the first cumulant is kept at a pre-specified level. The cost criterion for full-state feedback MCV control for infinite time horizon case is

$$J_{MCV} = \lim_{t_F \rightarrow \infty} \frac{1}{t_F} [E \{ J(t_F)^2 \} - (E \{ J(t_F) \})^2]. \quad (4.20)$$

In infinite time horizon, the full-state-feedback linear MCV control has the form (Won and Gunaratne, 2002),

$$u(t) = -R^{-1}B^T(\mathcal{M} + \gamma \mathcal{V})x(t) = -K_{MCV}x(t), \quad (4.21)$$

where the positive semi-definite \mathcal{M} and \mathcal{V} are solutions of the coupled algebraic Riccati equations:

$$A^T \mathcal{M} + \mathcal{M} A + Q - \mathcal{M} B R^{-1} B^T \mathcal{M} + \gamma^2 \mathcal{V} B R^{-1} B^T \mathcal{V} = 0, \quad (4.22)$$

and

$$4\mathcal{M} G W G^T \mathcal{M} + A^T \mathcal{V} + \mathcal{V} A - \mathcal{M} B R^{-1} B^T \mathcal{V} - \mathcal{V} B R^{-1} B^T \mathcal{M} - 2\gamma \mathcal{V} B R^{-1} B^T \mathcal{V} = 0, \quad (4.23)$$

If $\gamma = 0$, the MCV control problem becomes LQG control problem.

In our case, we want the output to follow a step command. Therefore, following the LQG case we can rewrite the control law for horizontal and vertical case respectively as,

$$u_{AZ}(t) = -K_{MCV,AZ}x(t) + f_x^*.1(t), \quad (4.24)$$

$$u_{EL}(t) = -K_{MCV,EL}x(t) + f_y^*.1(t). \quad (4.25)$$

4.3.2 Output Feedback Case

In practical situations, the states are not always available for measurement. Therefore, the statistical control law described in previous section cannot be implemented directly. The solution is to estimate the states of the system using the available measured states and noise statistics. For the gimbaled laser target system, only the laser dot position measurement is available. This section discusses building a state estimator from the measured output and noise statistics. Then the statistical control law is implemented in full state feedback fashion as discussed in the previous section.

Preliminaries

We consider a stochastic linear time invariant dynamic system modelled on $[t_0, t_f]$. The dynamic process model is given by,

$$dx(t) = Ax(t)dt + Bu(t)dt + Gdw(t), \quad x(t_0) = x_0, \quad t \in [t_0, t_f] \quad (4.26)$$

The observation process model is given by,

$$dy(t) = Cx(t)dt + dv(t), \quad (4.27)$$

$x(t) \in \mathbb{R}^n$	an n -dimensional state vector at time t ,
$u(t) \in \mathbb{R}^m$	an m -dimensional control action vector at time t ,
$w(t) \in \mathbb{R}^p$	an p -dimensional disturbance input vector at time t ,
x_0	the initial condition,
$A \in \mathcal{C}([t_0, t_F]; \mathbb{R}^{n \times n})$	a real continuous matrix of $n \times n$,
$B \in \mathcal{C}([t_0, t_F]; \mathbb{R}^{n \times m})$	a real continuous matrix of $m \times n$,
$G \in \mathcal{C}([t_0, t_F]; \mathbb{R}^{n \times p})$	a real continuous matrix of $n \times p$.
$y(t) \in \mathbb{R}^s$	an s -dimensional output vector at time t ,
$v(t) \in \mathbb{R}^s$	an s -dimensional measurement error vector at time t ,
$C \in \mathcal{C}([t_0, t_F]; \mathbb{R}^{s \times n})$	a real continuous matrix of $s \times n$.

The statistics of the disturbance inputs, measurement errors, and the initial conditions are assumed to be known.

- $w(t)$ is a white, zero-mean Gaussian random process with the expected value and covariance of

$$E[w(t)] = 0, \quad (4.28)$$

$$E[w(t)w^T(\tau)] = W\delta(t - \tau). \quad (4.29)$$

W is the spectral density matrix which is symmetric and positive semi-definite. The Dirac delta function $\delta(t - \tau)$ guarantees the whiteness property.

- $v(t)$ is a white, zero-mean Gaussian random process with the expected value and covariance of

$$E[v(t)] = 0, \quad (4.30)$$

$$E[v(t)v^T(\tau)] = V\delta(t - \tau). \quad (4.31)$$

V is the spectral density matrix which is symmetric and positive definite.

- $w(t)$ and $v(t)$ are uncorrelated which means,

$$E[w(t)v^T(\tau)] = 0, \quad \forall t, \tau. \quad (4.32)$$

- $x(t_0)$ has a known deterministic value. The expected value and the correlation of increment of the initial conditions are given by,

$$E[x(t_0)] = \tilde{x}_0, \quad (4.33)$$

$$E[(x(t_0) - \tilde{x}_0)(x(t_0) - \tilde{x}_0)^T] = P_0. \quad (4.34)$$

$x(t_0)$ is independent of $w(t)$ and $v(t)$ which means,

$$E[x(t_0)w(t)] = 0, \quad \forall t. \quad (4.35)$$

$$E[x(t_0)v(t)] = 0, \quad \forall t. \quad (4.36)$$

LQG and MCV Control with Estimated States

The output feedback LQG problem is to devise a feedback control law which minimizes the expected value of the cost function,

$$J = \int_{t_0}^{t_F} (x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau))d\tau. \quad (4.37)$$

The infinite horizon cost function is

$$J_{LQG} = \lim_{t_F \rightarrow \infty} \frac{1}{t_F} E[J(t_F)] \quad (4.38)$$

The solution to the LQG problem is given by the **separation principle** (Maciejowski and Vinnicombe, 2000). This principle divides the problem into two sub-problems.

- Obtain an optimal estimate $\tilde{x}(t)$ of the states $x(t)$ (optimal in the sense that the error variance $E[(x(t) - \tilde{x}(t))(x(t) - \tilde{x}(t))^T]$ is minimized).

$Q \in \mathcal{C}([t_0, t_F]; \mathbb{R}^{n \times n})$	a symmetric, positive semi-definite matrix (also called weighting matrix),
$R \in \mathcal{C}([t_0, t_F]); \mathbb{R}^{m \times m})$	a symmetric, positive definite matrix (also called weighting matrix),

- Use this estimate as if it were an exact measurement of the states to solve deterministic linear quadratic control problem.

The solution to the first sub-problem, is given by Kalman filter theory (Kalman, 1960). Fig shows the structure of the Kalman filter. The inputs to the Kalman filter are the control input and output vectors, u and y . The output of the filter is the state estimate vector $\tilde{x}(t)$. The Kalman filter gain matrix K_f is given by

$$K_f = \mathcal{P}_f C^T V^{-1}, \quad (4.39)$$

where \mathcal{P}_f satisfy the following algebraic Riccati equation:

$$\mathcal{P}_f A^T + A\mathcal{P}_f - \mathcal{P}_f C^T V^{-1} C \mathcal{P}_f + G W G^T = 0, \quad (4.40)$$

and \mathcal{P}_f is symmetric and positive semi-definite.

The second sub-problem is to find the control signal which will minimize the deterministic cost,

$$J = \int_{t_0}^{t_F} (x^T(\tau) Q x(\tau) + u^T(\tau) R u(\tau)) d\tau, \quad (4.41)$$

on the assumption that

$$\dot{x}(t) = Ax(t) + Bu(t). \quad (4.42)$$

The solution to this is to let the control signal $u(t)$ be a linear function of the state:

$$u(t) = -K_c x(t). \quad (4.43)$$

The optimal state-feedback matrix K_c is given by

$$K_c = R^{-1} B^T \mathcal{P}_c, \quad (4.44)$$

where P_c satisfies the algebraic Riccati equation

$$A^T \mathcal{P}_c + \mathcal{P}_c A - \mathcal{P}_c B R^{-1} B^T \mathcal{P}_c + Q = 0, \quad (4.45)$$

and \mathcal{P}_c is symmetric and positive semi-definite.

The matrices K_f and K_c exist, and the closed loop system is internally stable, provided the systems with state-space realizations $(A, G\sqrt{W}, C)$ and (A, B, \sqrt{Q}) are stabilizable and detectable. Now we shall look at the internal stability characteristic of the LQG compensated system. We start with the dynamic process model

$$dx(t) = Ax(t)dt + Bu(t)dt + Gdw(t). \quad (4.46)$$

We want to implement the control law as in

$$u(t) = -K_c \tilde{x}(t) + r(t), \quad (4.47)$$

where $r(t)$ is the step command that our system should track.

Inserting this control law in the dynamic process model we get the closed loop model

$$\begin{aligned} dx(t) &= Ax(t)dt + Bu(t)dt + Gdw(t), \\ &= Ax(t)dt + B(-K_c \tilde{x}(t) + r(t))dt + Gdw(t), \\ &= Ax(t)dt - BK_c \tilde{x}(t)dt + Br(t)dt + Gdw(t), \end{aligned} \quad (4.48)$$

$$\begin{aligned} &= Ax(t)dt - BK_c x(t)dt + BK_c x(t)dt - BK_c \tilde{x}(t)dt + Br(t)dt + Gdw(t), \\ &= (A - BK_c)x(t)dt + BK_c(x(t) - \tilde{x}(t))dt + Br(t)dt + Gdw(t), \\ &= (A - BK_c)x(t)dt + BK_c \epsilon(t)dt + Br(t)dt + Gdw(t), \end{aligned} \quad (4.49)$$

where the estimation error is defined by $\epsilon(t) = x(t) - \tilde{x}(t)$. The Kalman filter is governed by the model

$$d\tilde{x}(t) = (A - K_f C)\tilde{x}(t)dt + Bu(t)dt + K_f dy(t). \quad (4.50)$$

Subtracting (4.50) from (4.46) we get,

$$\begin{aligned}
d\epsilon(t) &= Ax(t)dt + Bu(t)dt + Gdw(t) - (A - K_f C)\tilde{x}(t)dt - Bu(t)dt - K_f dy(t), \\
&= Ax(t)dt + Gdw(t) - (A - K_f C)\tilde{x}(t)dt - K_f(Cx(t) + dv(t)), \\
&= (A - K_f C)(x(t) - \tilde{x}(t))dt + Gdw(t) - K_f dv(t), \\
d\epsilon(t) &= (A - K_f C)(\epsilon(t))dt + Gdw(t) - K_f dv(t).
\end{aligned} \tag{4.51}$$

Combining (4.49) and (4.51), we get the LQG compensated augmented system,

$$\begin{bmatrix} dx(t) \\ d\epsilon(t) \end{bmatrix} = \begin{bmatrix} A - BK_c & BK_c \\ 0 & A - K_f C \end{bmatrix} \begin{bmatrix} x(t) \\ \epsilon(t) \end{bmatrix} dt + \begin{bmatrix} B \\ 0 \end{bmatrix} r(t)dt + \begin{bmatrix} G & 0 \\ G & -K_f \end{bmatrix} \begin{bmatrix} dw(t) \\ dv(t) \end{bmatrix}. \tag{4.52}$$

Eq. (4.53) can be written as

$$d\tilde{x}(t) = K_f Cx(t)dt + (A - K_f C - BK_c)\tilde{x}(t)dt + Br(t)dt + K_f dv(t). \tag{4.53}$$

Combining (4.46), (4.47) and (4.53), we get another form of the LQG compensated augmented system

$$\begin{bmatrix} dx(t) \\ d\tilde{x}(t) \end{bmatrix} = \begin{bmatrix} A & -BK_c \\ K_f C & A - K_f C - BK_c \end{bmatrix} \begin{bmatrix} x(t) \\ \tilde{x}(t) \end{bmatrix} dt + \begin{bmatrix} B \\ B \end{bmatrix} r(t)dt + \begin{bmatrix} G & 0 \\ 0 & K_f \end{bmatrix} \begin{bmatrix} dw(t) \\ dv(t) \end{bmatrix}. \tag{4.54}$$

For MCV control, we shall build an estimator using Kalman Filter. Then the full-state feedback control strategy will be implemented. The difference from LQG control would be the state-feedback gain matrix K_c in (4.44). In this case, K_c will be computed from the full-state feedback case in (4.21):

$$K_c = -R^{-1}B^T(\mathcal{M} + \gamma\mathcal{V}). \tag{4.55}$$

For MCV, the compensated augmented system takes the same form as in (4.52) and (4.54).

In summary, we described PID, LQG, and MCV control schemes. For PID controller tuning, we used Matlab tool for PID tuning. LQG and MCV schemes were discussed for the full-state

feedback and the output feedback case. We modified the LQG and MCV control laws for the step command tracking. Next, we shall describe the simulations of the system with the control methodologies described in this chapter.

CHAPTER 5

SIMULATIONS

5.1 Overview

In this chapter, we present the simulations for PID, LQG, MCV control of the linear model of the gimbaled laser target system developed in Chapter 3. The PID controllers are simulated for noiseless, with Gaussian white noise, and noise rejection cases. Then the LQG controller is simulated and compared to the PID controller in terms of noise rejection. Finally, the LQG controller and the MCV controllers are simulated with the comparison of pointing error and pointing error variations.

5.2 Simulation Parameters

Table 5.1 and 5.2 list the parameter values used in the simulation.

5.3 PID Control Simulations

The linear model of the Pitch Gimbaled Laser Target System is given by Eq. (3.27) and (3.34). The input to the model is vertical position set-point $f_y^*(t)$. The output of the model is the vertical position co-ordinate on the screen $f_y(t)$.

Eq. (3.36) and (3.43) are the linear model of the Yaw Gimbaled Laser Target System. The input to the model is horizontal position set-point $f_x^*(t)$. The output of the model is the horizontal position co-ordinate on the screen $f_x(t)$.

5.3.1 Parameter Selection Process

We went through following steps for PID control simulations:

- The PID tuner was used to obtain PID control parameters for the linear models assuming that there is no noise present.

Table 5.1: Simulation parameters of servo motor drive in gimbaled laser target system

Parameter	Description	Value	Unit	Source
R	Resistance per phase	2.60	Ohm	Newmark Inc.
L	Inductance per phase	2.8×10^{-3}	Henry	Newmark Inc.
b	Damping coefficient	3.65×10^{-5}		Krishnan (2001)
k_b	back emf constant	0.0678	V/rad/s	Newmark Inc.
V_{dc}	DC voltage	42	V	experiment
J_m	Rotor inertia	1.13×10^{-4}	kgm^2	Newmark Inc.
	Triangular wave frequency	1000	Hz	experiment
v_c	Triangular wave peak value	5	V	Copley control
H_c	Current feedback	8		arbitrary
K_{sEL}, K_{sAZ}	Speed filter gain	10, 40		arbitrary
$\tau_{sEL} = \tau_{sAZ}$	Speed filter time constant	0.0052	s	arbitrary

- A Gaussian white noise was applied to the model, and the PID controllers were simulated.
- Then the PID tuner was used to find another set of parameters for the best noise rejection performance.

Here we show an example how we used PID tuner to choose parameter values. This example considers a plant model of

$$G(s) = \frac{6(s+5)e^{-s}}{(s+1)(s+2)(s+3)(s+4)},$$

which has to be controlled by a PI controller. The PID tuner selects the parameter values initially considering closed loop stability, adequate performance, adequate robustness.

Fig. 5.1 shows the PID tuner for reference tracking. For step reference tracking, the settling time is about 12 seconds and the overshoot is about 6.3 percent, which is acceptable for this example.

Table 5.2: Simulation parameters of gimbal and position sensor of laser target system

Parameter	Description	Value	Unit	Source
J_{ex}, J_{ey}, J_{ez}	pitch diagonal element(inertia)	0.000565, 0.036725, 0.0226		Kennedy (2003)
J_{ax}, J_{ay}, J_{az}	yaw diagonal element(inertia)	0.0113, 0.0452, 0.0678		Kennedy (2003)
$K_{ef} = K_{af}$	viscous friction coefficient	0.00063	Nm/rad/s	Iwasaki et al. (1999)
N	gear ratio	90	Nm	Kennedy (2003)
$K_{e\omega} = K_{a\omega}$	Cable restraint co-efficient	0.000011	rad/s	Kennedy (2003)
δD	Length of the screen	30.30	cm	Experiment
δP	Pixel Length of the screen	480		Experiment
λ	focal length of camera	0.38	cm	Gigaware
$s_x = s_y$	pixel dimension	0.000056	cm	Gigaware

Then we assume that a step disturbance occurs at the plant input and the main purpose of the PI controller is to reject this disturbance quickly. If we “Switch Response” to “Input disturbance rejection”, The peak deviation is about 1 and it settles to less than 0.1 in about 9 seconds. This is shown in Fig. 5.2.

We move the response time slider to the right to increase the response speed (open loop bandwidth). The K_I gain in the Controller parameters table first increases and then decreases, with the maximum value occurring at 0.3. When K_I is 0.3, the peak deviation is reduced to 0.9 (about 10% improvement) and it settles to less than 0.1 in about 6.7 seconds (about 25% improvement). This is shown in Fig. 5.3. We switch the response back to “Reference tracking”. Because we increase the bandwidth, the step reference tracking response becomes more oscillated. Additionally the overshoot exceeds 15 percent, which is usually unacceptable.

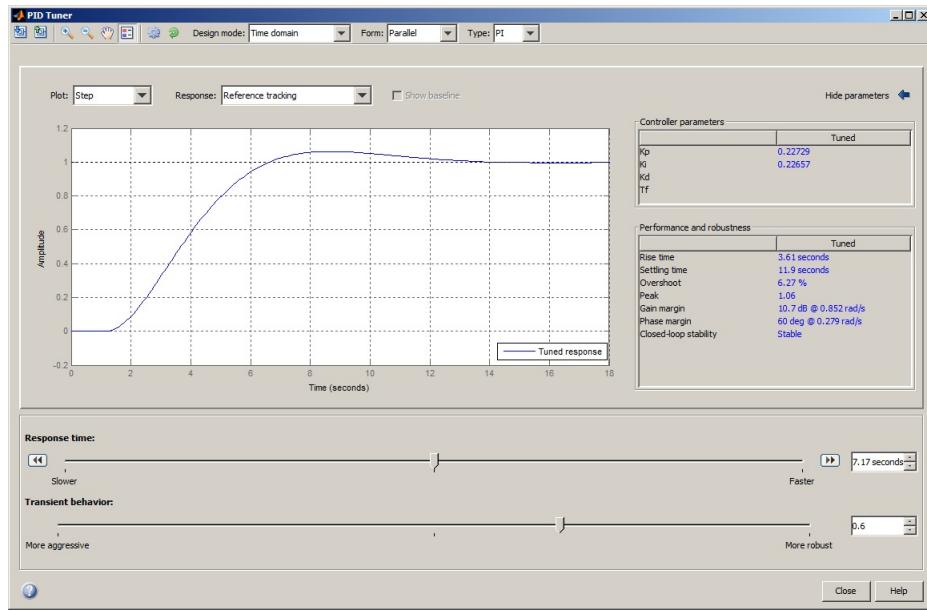


Figure 5.1: Initial Tuning (Matlab PID tuner)

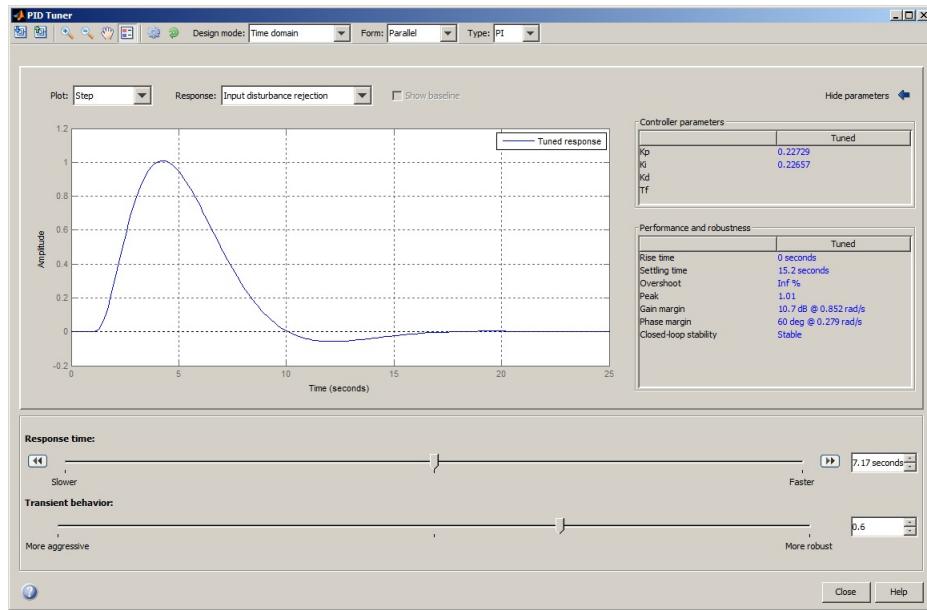


Figure 5.2: Reference tracking response after tuning (Matlab PID tuner)

This is shown in Fig. 5.4. This type of performance trade off between reference tracking and disturbance rejection often exists because a single PID controller is not able to satisfy both design goals at the same time.

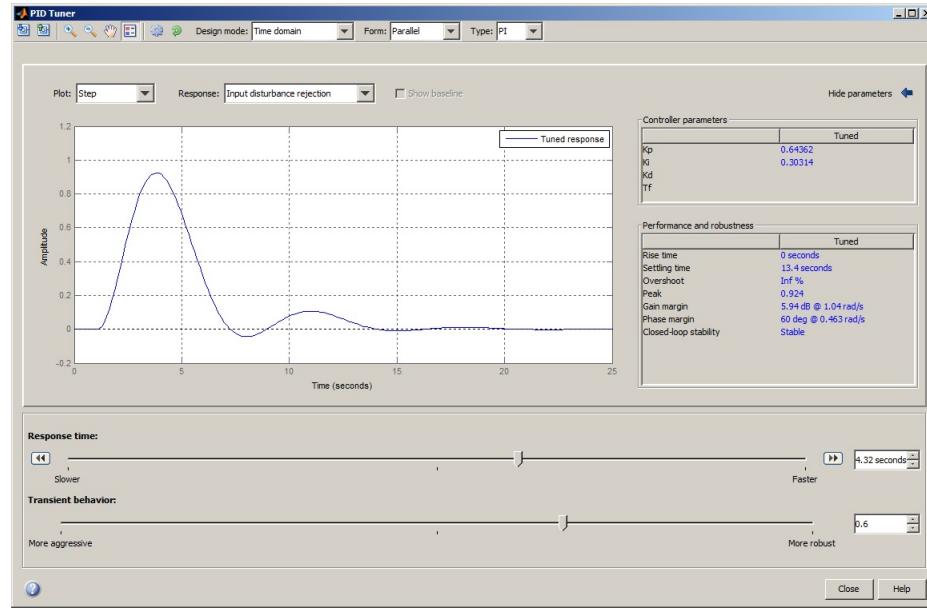


Figure 5.3: Tuning for disturbance rejection (Matlab PID tuner)

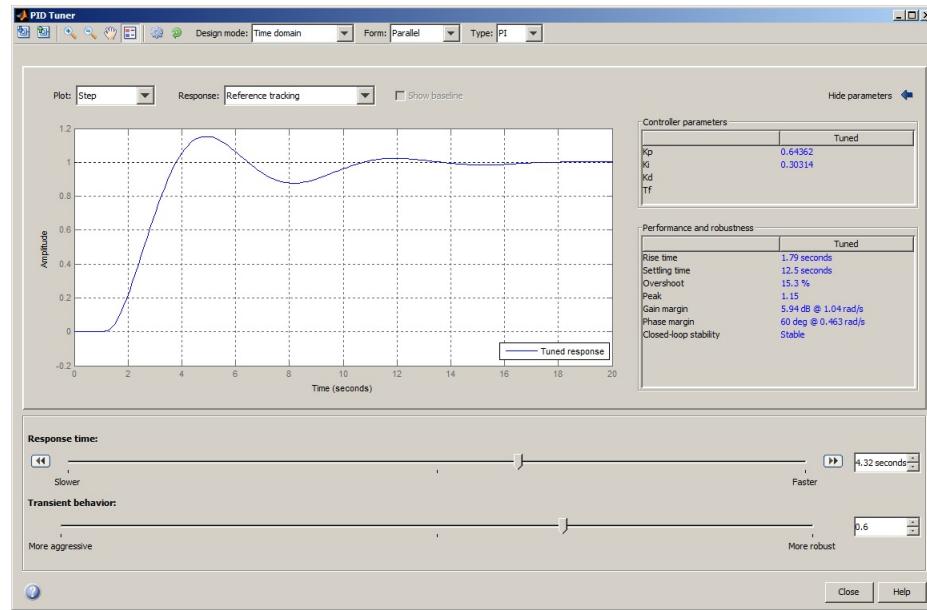


Figure 5.4: Reference tracking response after tuning for disturbance rejection

From this example we see that the designer can specify some baseline criteria for designing PID controller, and use Matlab tuner to design it.

Table 5.3: PID tuned parameters

Case	Model	K_p	K_I	K_D	N_p	Fig.
PID without noise	Pitch	0.0058	7.009	0.0422	3.2773	5.5
	Yaw	0.0065	0.00012	0.0357	4.8923	5.5
PID with noise	Pitch	0.0058	7.009	0.0422	3.2773	5.6
	Yaw	0.0065	0.00012	0.0357	4.8923	5.6
PID optimized for disturbance rejection (High noise variance)	Pitch	0.0315	0.001	0.118	9.02	5.7
	Yaw	0.0104	0.0002	0.0473	6.4492	5.7

For our case, we specify that there should be little or no overshoot. This is the most important criteria of our control objective. In space-based solar power application, it is not desirable that laser beam hit outside the receiver antenna. The settling time should be less than 20 seconds. The settling time considered here is arbitrary. The gain and phase margin should be 10 dB and 60° . We used PID tuner to view whether these criteria fulfilled, and then selected the parameters for the simulation.

Table 5.3 lists all the tuned parameters used for the simulations.

5.3.2 PID Simulation Results

Both the vertical and horizontal set-points are 15.15 cm. The initial position co-ordinate is (2 cm, 2cm). The simulations were run for 200 seconds. 200 seconds were considered long enough to observe the steady-state characteristics of the system with the PID controllers. The form of PID controllers were described in (4.6).

Fig. 5.5 shows the position response of PID controlled GLTS without noise. The vertical position reached 95 percent of the desired position at 11.6 seconds; the horizontal position took 9.01 seconds. In both cases, an overdamp characteristic was observed.

A Simulink input noise block was used to generate Gaussian white noise over $[0, 1]$. The sample time was chosen as 0.0001 seconds. The fastest time constant in the model is closer to

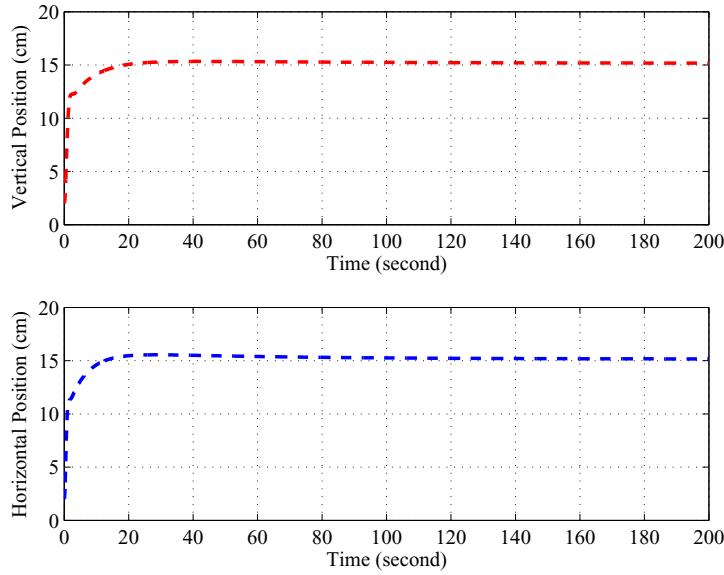


Figure 5.5: Pointing response with PID controller without noise

1 milli second. Therefore, 10 milli seconds of sampling interval were considered sufficient to capture the complete characteristics of the system. The seed used for random number generation is 24431. The same seed was used for all the simulations to ensure the repeatability of the results. To observe the average performance, the seeds were later chosen randomly. The details about this will be covered in the MCV controller section.

A Gaussian white noise with noise variance of 0.001 was applied to the model. The reason for using Gaussian noises is that many naturally occurring noise processes are Gaussian Anderson and Moore (1990). Fig. 5.6 shows the PID controlled system response under this noise. The vertical position response varied between 290.90 cm to -127.60 cm. The horizontal position response varied from 91.03 cm to -28.75 cm. This shows that the PID controllers perform poorly under Gaussian noise. Also, the range of the pointing response values are greater than 30.3 cm. This indicates that under this noise, the system requires a bigger screen than the current specification. Furthermore, the system did not settle to some steady-state value. Another set of parameters were obtained for the noise rejection. This is found by the Matlab PID tuner tool. The vertical position varied between 92.18 cm to -41.18 cm. The horizontal position

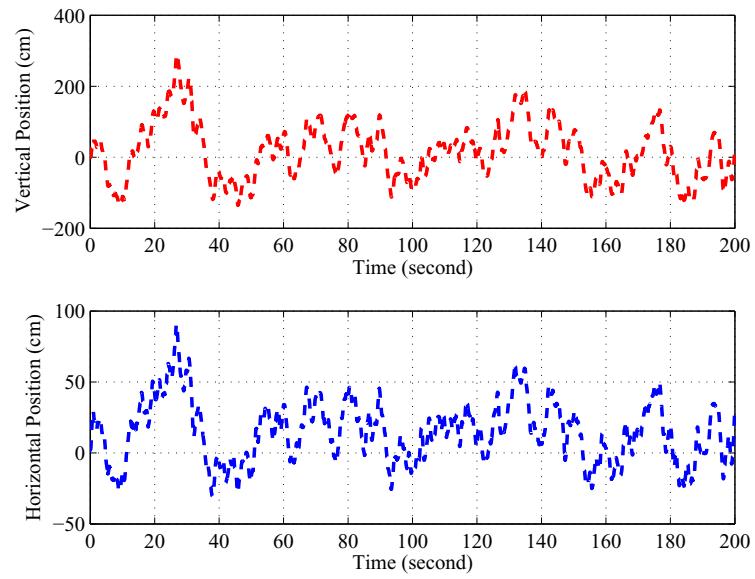


Figure 5.6: Pointing response with PID controller under Gaussian white noise

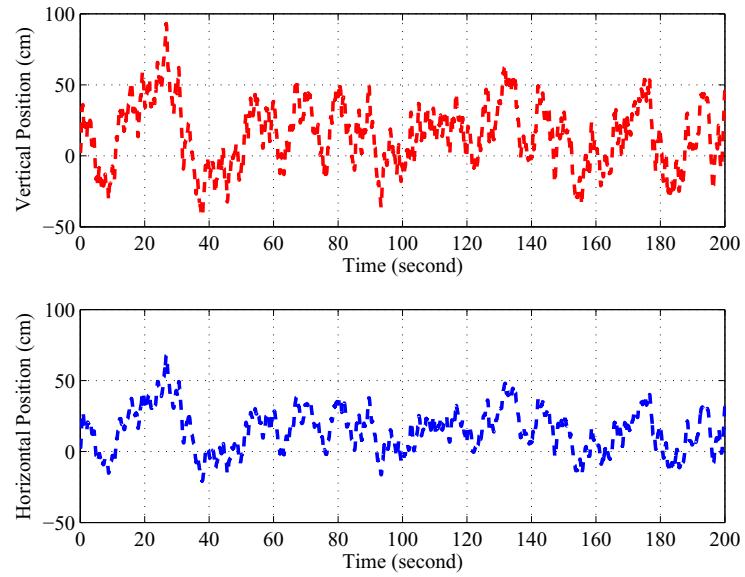


Figure 5.7: Pointing response with PID controller under Gaussian white noise with optimized parameters

varied between 68.14 cm to -18.85 cm. This shows that the new set of parameters improve the noise rejection capacity of the PID controllers. However, the pointing responses shown in Fig. 5.7 are not satisfactory.

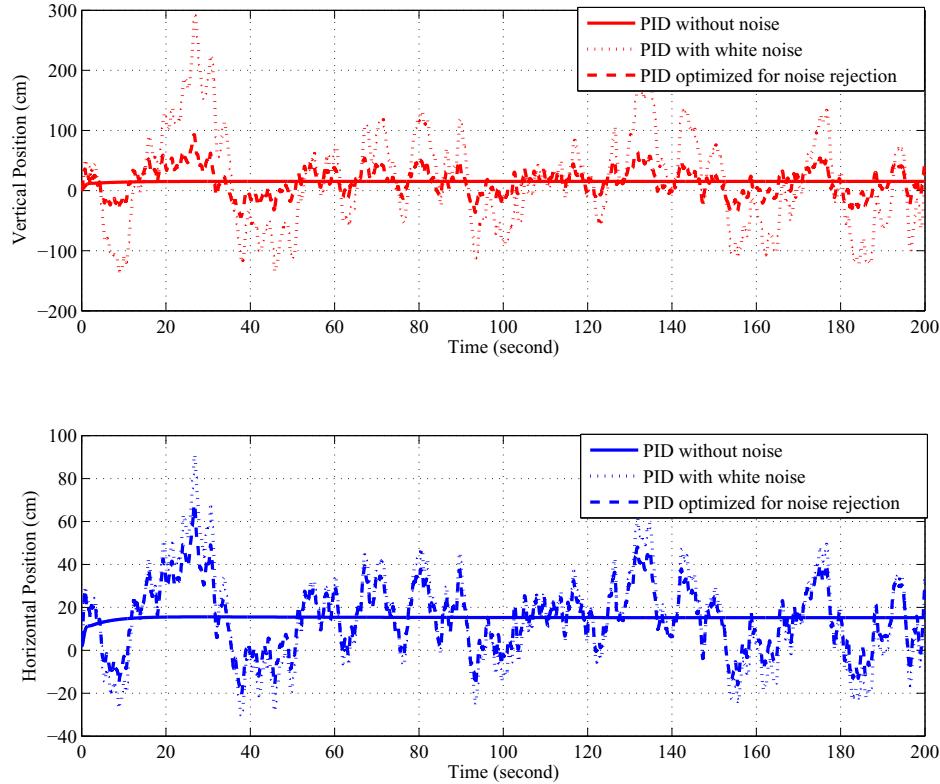


Figure 5.8: Pointing responses with PID controller for three cases

In Fig. 5.8, the pointing responses of PID controlled GLTS without noise, with the Gaussian white noise with noise variance = 0.001, with the optimized parameters for noise rejection are put together.

In the next section, the statistical controllers will be simulated. The hypothesis is that the statistical control methods such as the Linear Quadratic Gaussian Controller and the Minimum Cost Variance Controller will perform better than the PID Controllers in the presence of Gaussian white noise.

5.4 Full State Feedback LQG Control Simulations

Eq. (3.27), (3.34), (3.36), and (3.43) describe the linear model for pitch and yaw GLTS, respectively. All the eigenvalues were negative for both models. Therefore, both the models were stable. The models were found controllable and observable. Here it is assumed that all the states can be measured. A full state feedback control scheme was simulated. First the controller was derived in Matlab using the process dynamics. Then the system was simulated with the controllers in simulink. To implement the model with the set-point and the process noise, the state-space representation of the model was re-arranged. One example of the re-arrangement will be shown for the pitch GLTS model. For the ease of understanding, the pitch GLTS model description is repeated from Chapter 3.

$$dx_{EL}(t) = (A_{EL}x_{EL}(t) + B_{EL}u_{EL}(t))dt + G_{EL}dw_{EL}(t), \quad (5.1)$$

where, the state matrix,

$$A_{EL} = \begin{bmatrix} 0 & C_y & 0 & 0 \\ -\frac{K_{e\omega}}{C_y J_{ey}} & -\frac{K_{ef}}{J_{ey}} & \frac{k_b N}{J_{ey}} & 0 \\ 0 & 0 & -\frac{1}{\tau_{iEL}} & \frac{K_{iEL}}{\tau_{iEL}} \\ 0 & 0 & 0 & -\frac{1}{\tau_{sEL}} \end{bmatrix}, \quad (5.2)$$

the state vector,

$$x_{EL}(t) = \begin{bmatrix} f_y(t) & \omega_{ey}(t) & i_{aEL}(t) & i_{aEL}^*(t) \end{bmatrix}^T, \quad (5.3)$$

with

$f_y(t)$ = vertical position co-ordinate on screen,

$\omega_{ey}(t)$ = pitch gimbal angular rate,

$i_{aEL}(t)$ = armature current for servo-motor drive for pitch gimbal,

$i_{aEL}^*(t)$ = reference current command for servo-motor drive for pitch gimbal, and

The initial condition $x_{EL}(0)$ is chosen arbitrarily. The vertical position co-ordinate is chosen as 2 cm; all other states are zero initially. The initial conditions are,

$$x_{EL}(0) = \begin{bmatrix} 2 & 0 & 0 & 0 \end{bmatrix}^T, \quad (5.4)$$

the input matrix is,

$$B_{EL} = \begin{bmatrix} 0 & 0 & 0 & \frac{K_{sEL}C_{\omega EL}}{\tau_{sEL}} \end{bmatrix}^T, \quad (5.5)$$

with K_{sEL} = pitch servo drive speed filter gain,

τ_{sEL} = pitch gimbal servo drive speed filter time constant,

$C_{\omega EL}$ = scale factor for converting vertical position co-ordinate into reference speed command for pitch and yaw gimbal servo drive, and

$$u_{EL}(t) = f_y^*(t), \quad (5.6)$$

with $f_y^*(t)$ = vertical reference position co-ordinate. $w_{EL}(t)$ is the process noise. $dw_{EL}(t)$ is a Gaussian random process with zero mean and the covariance matrix $W dt$.

$$G_{EL} = \begin{bmatrix} 0 & J_{ey}^{-1} & 0 & 0 \end{bmatrix}^T, \quad (5.7)$$

The output is $f_y(t)$ vertical position co-ordinate on screen. The output equation is,

$$y_{EL}(t) = C_{EL}x_{EL}(t), \quad (5.8)$$

where,

$$C_{EL} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}. \quad (5.9)$$

Note that there is no measurement noise because of the full state feedback assumption. The statistics (expectation and covariance) of the initial condition, process noise and measurement noise are as follows:

$$\begin{aligned} E[x_{EL}(0)] &= \hat{x}_{EL0}, E\{[x_{EL}(0) - \hat{x}_{EL}(0)][x_{EL}(0) - \hat{x}_{EL}(0)]^T\} = P_0, \\ E[w_{EL}(t)] &= 0, E[w_{EL}(t)w_{EL}^T(\tau)] = W\delta(t - \tau). \end{aligned}$$

Eq. (5.1) and (5.8) are the linear model of the Pitch Gimbaled Laser Target System. The cost function is written as follows:

$$J_{EL} = \int_0^{t_F} (x_{EL}^T(t)Qx_{EL}(t) + u_{EL}^T(t)Ru_{EL}(t))dt. \quad (5.10)$$

As is customary in such cost, the matrices Q and R are both symmetric, Q being positive semidefinite, and R being positive definite.

The LQG cost function is the expectation or the first cumulant of the quadratic cost function defined in (5.10).

$$J_{LQG} = \lim_{t_F \rightarrow \infty} \frac{1}{t_F} E \{ J_{EL} \}. \quad (5.11)$$

The full-state feedback LQG control that minimizes the cost function is

$$u_{EL}(t) = -K_{LQG,EL}x_{EL}(t) + f_y^*(t), \quad (5.12)$$

where $K_{LQG,EL}$ is the full state feedback LQG gain.

In our case, we want the output to track a constant step command $f_y^*(t) = f_y^*.1(t)$. Therefore, we shall implement the control law that minimizes the cost function as

$$\begin{aligned} u_{EL}(t) &= -K_{LQG,EL}x_{EL}(t) + f_y^*(t), \\ &= -K_{LQG,EL}x_{EL}(t) + f_y^*.1(t). \end{aligned} \quad (5.13)$$

Using control law in (5.13), the state equation in (5.2) is re-written as:

$$\begin{aligned} dx_{EL}(t) &= ((A_{EL} - B_{EL}K_{LQG,EL})x_{EL}(t))dt + B_{EL}f_y^*.1(t) + G_{EL}dw_{EL}(t), \\ &= ((A_{EL} - B_{EL}K_{LQG,EL})x_{EL}(t))dt + \begin{bmatrix} B_{EL} & G_{EL} \end{bmatrix} \begin{bmatrix} f_y^*.1(t) \\ dw_{EL}(t) \end{bmatrix}, \\ &= \hat{A}_{EL}x_{EL}(t)dt + \hat{B}_{EL}\hat{u}_{EL}(t). \end{aligned} \quad (5.14)$$

Assuming all the states are measured, we write the output equation as:

$$\hat{y}_{EL}(t) = \hat{C}_{EL}x_{EL}(t) + \hat{D}_{EL}u_{EL}(t), \quad (5.15)$$

where \hat{C}_{EL} is a fourth order identity matrix, and \hat{D}_{EL} is a 4×2 zero matrix. Similar rearrangement was done for yaw gimbaled laser target system model.

Now (5.14) and (5.15) will be implemented in simulink state-space block. Similar equations are implemented for yaw gimbaled laser target system model. Q is chosen as a fourth order

identity matrix, and R is chosen as 1. These design parameters are selected arbitrarily. The noise variance of the process noise is 0.001. The simulation is run for 200 seconds. The initial position co-ordinate is (2 cm, 2 cm).

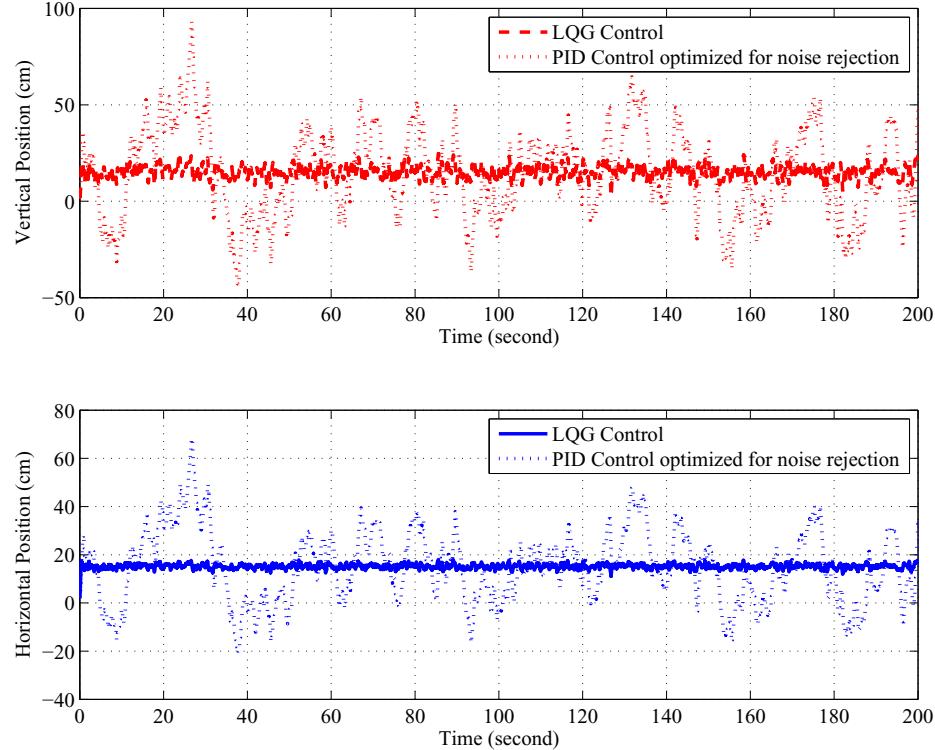


Figure 5.9: Pointing responses with LQG and PID controller under Gaussian white noise

Fig. 5.9 shows the LQG controlled GLTS pointing response. Also the PID controlled system responses with the same noise condition are plotted. From the plot we see that for the full state feedback LQG the pointing response varies around 15.15 cm with ± 5 cm variation. For the PID controlled system, the pointing varies between 92.18 cm to -41.18 cm. It is evident that the LQG control performed better than PID control in the presence of Gaussian noise.

5.5 Full State Feedback MCV Control Simulations

The cost criterion for the full-state-feedback MCV control for the pitch GLTS for the finite time horizon case (from (4.20)) is

$$J_{MCV} = \lim_{t_F \rightarrow \infty} \frac{1}{t_F} [E \{ J(t_F)^2 \} - (E \{ J(t_F) \})^2]. \quad (5.16)$$

We applied the full-state-feedback linear MCV control with a view to forcing the output to follow a step command from Eq. (4.25),

$$u_{EL}(t) = -K_{MCV,EL}x(t) + f_y^*(t) = -K_{MCV,EL}x(t) + f_y^*.1(t). \quad (5.17)$$

5.5.1 LQG Control as a special case of MCV Control

An MCV controller was developed for finite horizon case. γ is an important parameter of the MCV controller. For a special case of $\gamma = 0$, MCV control problem becomes LQG control problem. In this section, it will be shown that MCV control for finite horizon case can be used for infinite horizon case. For this, LQG control for infinite horizon case and MCV control with $\gamma = 0$ for finite horizon case will be compared. Using same design parameters, LQG and MCV ($\gamma = 0$) controllers were obtained. LQG controllers for pitch and yaw GLTS are:

$$K_{LQG,EL} = \begin{bmatrix} 0.99992 & 22.005 & 3.6345 & 2.3187 \end{bmatrix}, \quad (5.18)$$

and

$$K_{LQG,AZ} = \begin{bmatrix} 0.99998 & 18.219 & 1.2216 & 0.80839 \end{bmatrix}. \quad (5.19)$$

MCV $\gamma = 0$ controllers were obtained for a final time of 5 seconds. MCV controllers had same value as LQG controllers. Therefore, MCV $\gamma = 0$ controllers for the finite horizon case can be used for the infinite horizon case as well. From this point on, the MCV control for $\gamma = 0$ will be considered as the LQG control.

5.6 Performance Metrics

- Pointing Error and Root-Mean-Squared Pointing Error: The pointing error is the difference between the set-points and the actual position co-ordinates. The root mean squared

pointing error is defined in (5.20):

$$\begin{aligned}
\mu_\varepsilon &= \sqrt{\frac{\sum_{t=0}^{t=t_f} \varepsilon^2(t)}{N_\varepsilon}}, \\
&= \frac{\sqrt{\sum_{t=0}^{t=t_f} \varepsilon^2(t)}}{\sqrt{N_\varepsilon}}, \\
&= \frac{\|\varepsilon(t)\|_2}{\sqrt{N_\varepsilon}},
\end{aligned} \tag{5.20}$$

where μ_ε is the root mean squared error, ε is the error between the set-point position co-ordinate and the current position co-ordinate, N_ε is the number of samples.

- Standard Deviation of Pointing Error : The standard deviation of the pointing error indicates the variation of the error from the mean error value. This is defined as:

$$\sigma_\varepsilon = \sqrt{\frac{\sum_{t=0}^{t=t_f} (\varepsilon(t) - \mu)^2}{N_\varepsilon - 1}}. \tag{5.21}$$

This formula is also referred as the corrected sample standard deviation. We are interested about the steady-state pointing performance of the system. Therefore, we shall calculate the mean and standard deviation after removing the transient part of the position response.

- State Values : Won (1999) analyzed the average state values to compare the control methodologies. The interesting states here are the pointing responses and the angular velocities.

5.7 Comparison of Full State Feedback LQG and MCV Controller for GLTS

5.7.1 Pointing Responses and Pointing Error

Fig. 5.10 shows the vertical pointing responses of MCV controlled GLTS for $\gamma = 0$, $\gamma = 0.01$, $\gamma = 0.05$, $\gamma = 0.1$, $\gamma = 0.15$, and $\gamma = 0.2$. They all started from the initial condition of 2 cm. The desired position response is 15.15 cm. The vertical pointing responses do not approach to the target 15.15 cm for any value of γ . This is because of the applied Gaussian white noise. With the increasing γ values the responses shift downwards reducing the mean values, the ripple in the response decreases.

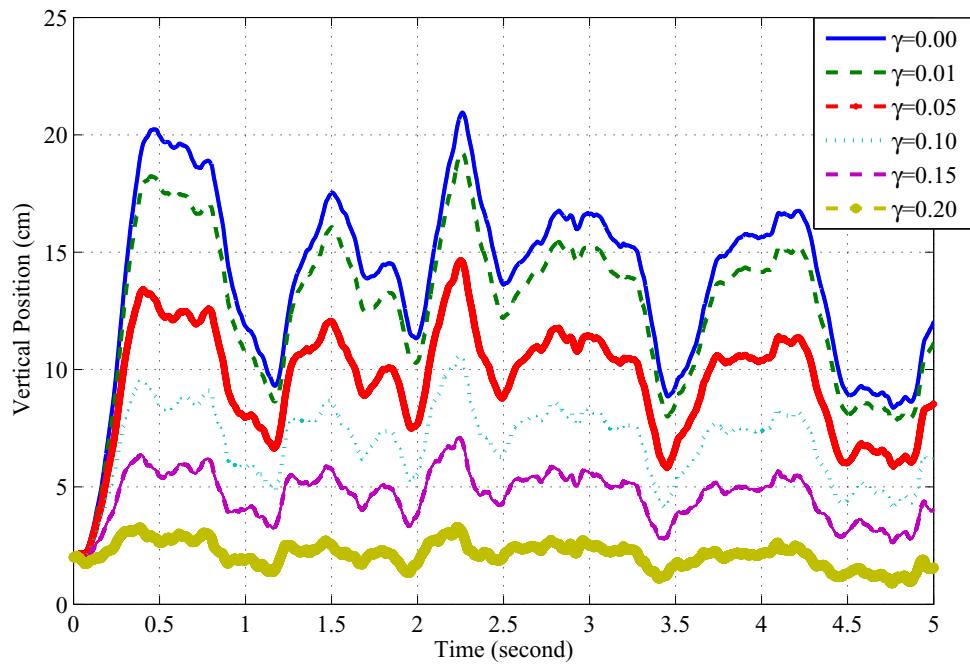


Figure 5.10: Vertical pointing responses with MCV controllers

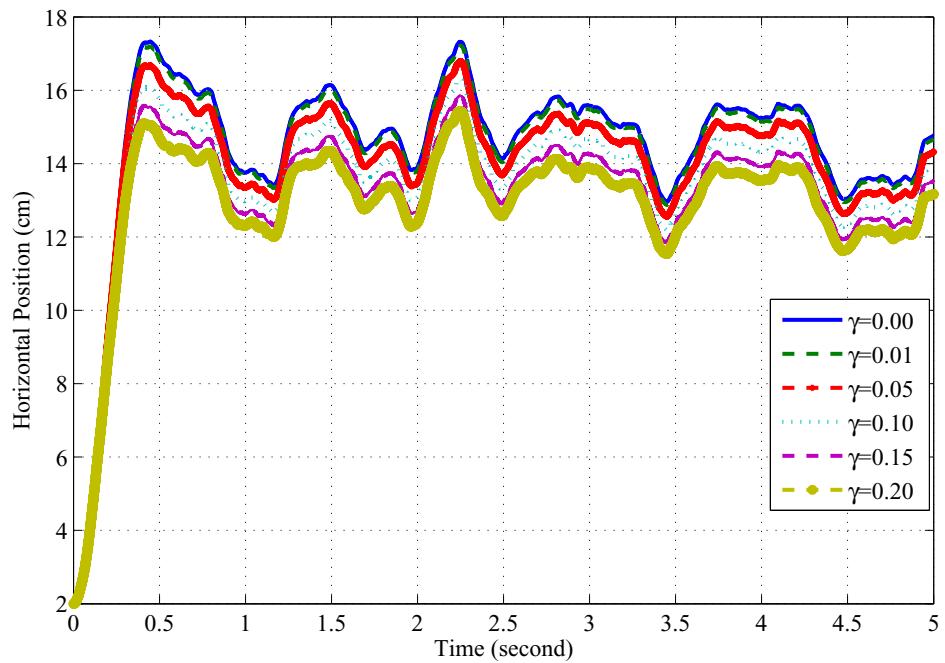


Figure 5.11: Horizontal pointing responses with MCV controllers

Fig. 5.11 shows the horizontal pointing responses of MCV controlled GLTS for $\gamma = 0$, $\gamma = 0.01$, $\gamma = 0.05$, $\gamma = 0.1$, $\gamma = 0.15$, and $\gamma = 0.2$. They show the same trend as the vertical responses. They are less affected by the applied Gaussian white noise than the vertical responses.

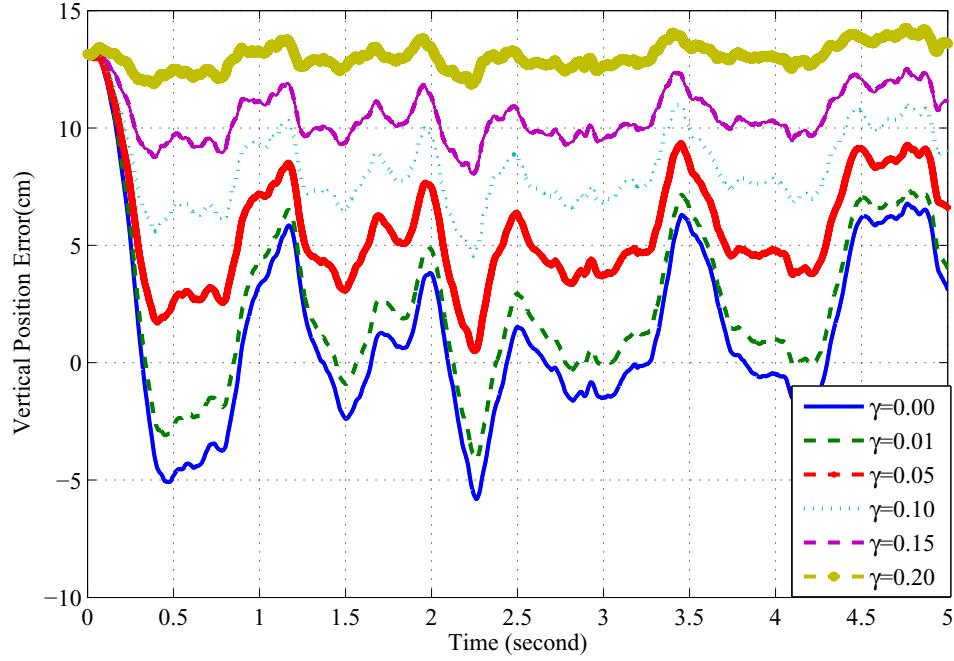


Figure 5.12: Vertical pointing error with MCV controllers

Fig. 5.12 and 5.13 show the pointing error responses for vertical and horizontal position. These two plots show with increasing γ , the pointing errors increase.

The above pointing responses are subject to a Gaussian white noise generated from a fixed seed 23341. To observe the average position response, the simulation was run 100 times with 100 different seeds chosen randomly. The averaging effect will smooth the ripples in the responses. It helps viewing the differences in the responses..

Fig. 5.14 and 5.15 show the average vertical and horizontal pointing responses of MCV controlled GLTS for $\gamma = 0$, $\gamma = 0.01$, $\gamma = 0.05$, $\gamma = 0.1$, $\gamma = 0.15$, and $\gamma = 0.2$. The averaging effect reduces the ripples in the response graphs. Also, it is clear from the average pointing error graphs of Fig. 5.16 and 5.17.

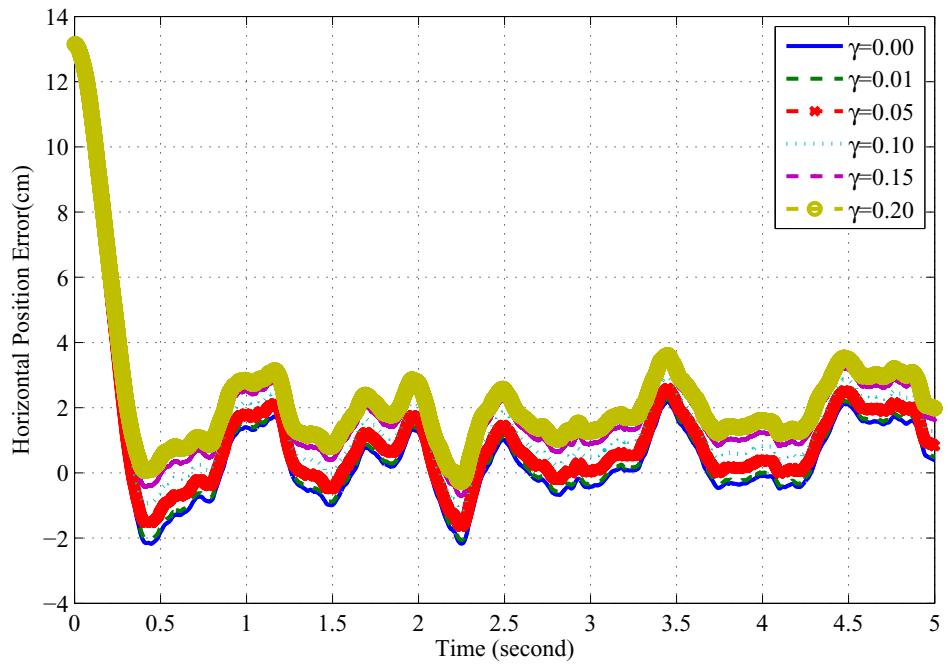


Figure 5.13: Horizontal pointing error with MCV controllers

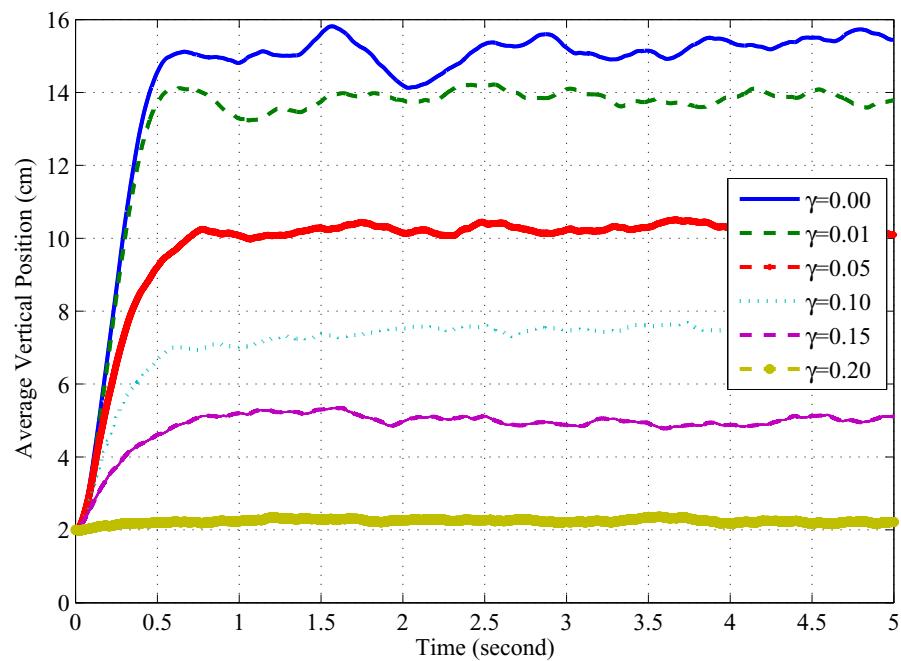


Figure 5.14: Average vertical pointing responses with MCV controllers

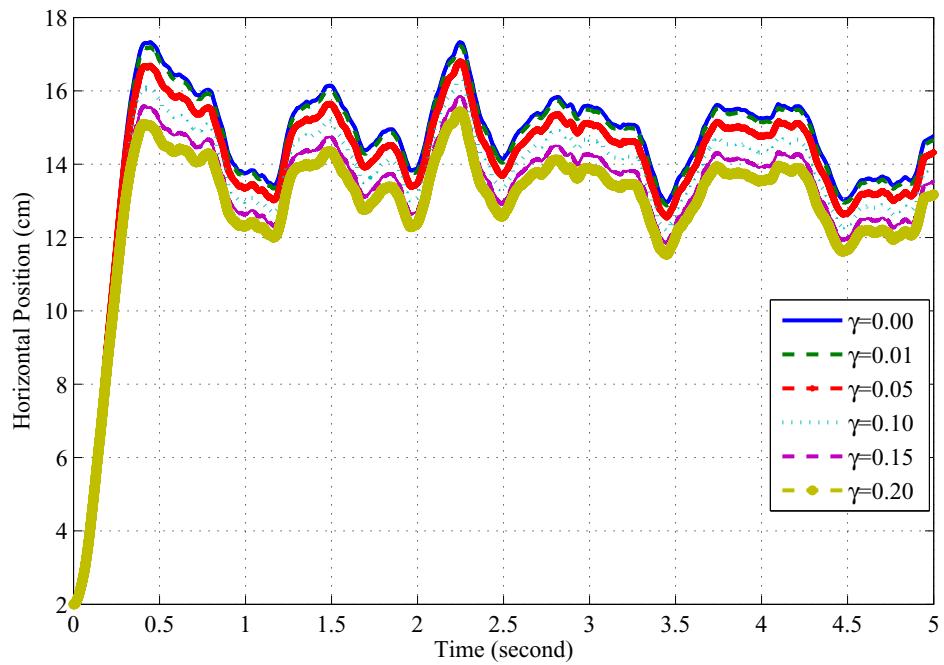


Figure 5.15: Average horizontal pointing responses with MCV controllers

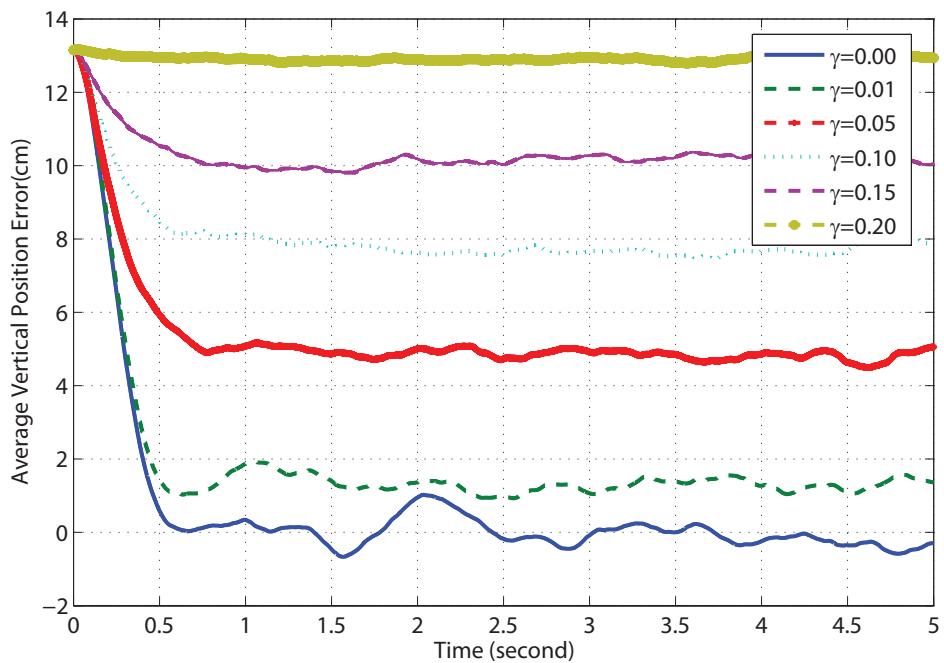


Figure 5.16: Average vertical pointing error with MCV controllers

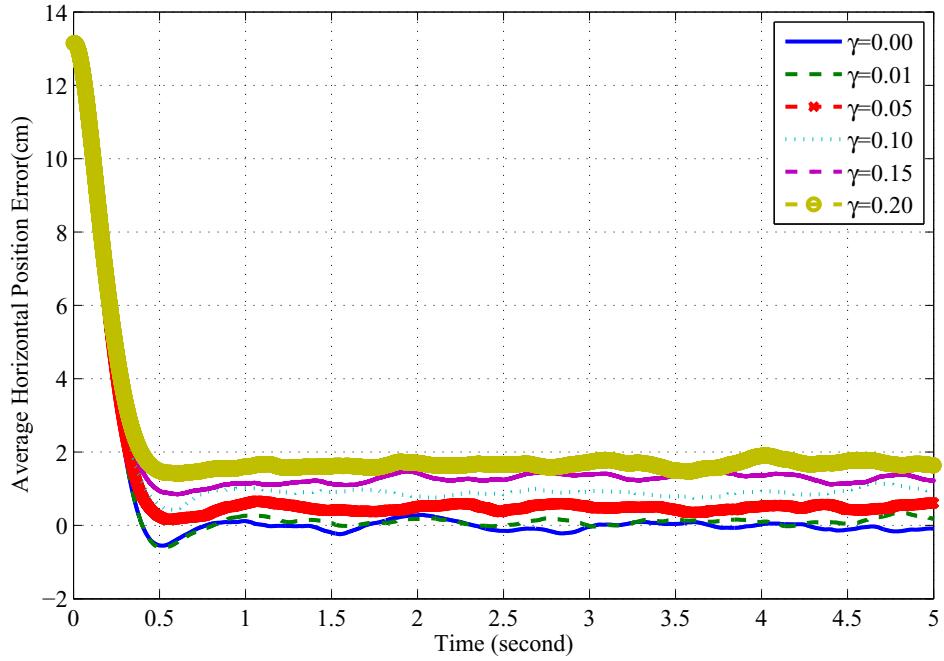


Figure 5.17: Average horizontal pointing error with MCV controllers

5.7.2 Angular Velocity Responses

Although we do not compare the controllers based on angular velocity response, it is interesting to observe the response because it is related to position. We want to see the peak angular velocity and the time it takes to settle to zero values.

Fig. 5.18 shows the responses of angular velocity ω_{ey} for $\gamma = 0$, $\gamma = 0.01$, $\gamma = 0.05$, $\gamma = 0.1$, $\gamma = 0.15$, and $\gamma = 0.2$. Fig. 5.19 is the zoomed view of Fig. 5.18. The responses show that the angular velocities vary between 0.95 to -0.7 radian per second. As the γ increases from 0, the angular velocity variation decreases.

Fig. 5.20 shows the responses of angular velocity ω_{ez} for $\gamma = 0$, $\gamma = 0.01$, $\gamma = 0.05$, $\gamma = 0.1$, $\gamma = 0.15$, and $\gamma = 0.2$. Fig. 5.21 is the zoomed view of Fig. 5.20. The responses show that the angular velocities vary between 0.55 to -0.25 radian per second. As the γ increases from 0 to 0.2, there is not much variation observed in the responses. In the zoomed view, we see the angular velocity values decrease as γ increases. This eventually affects the pointing

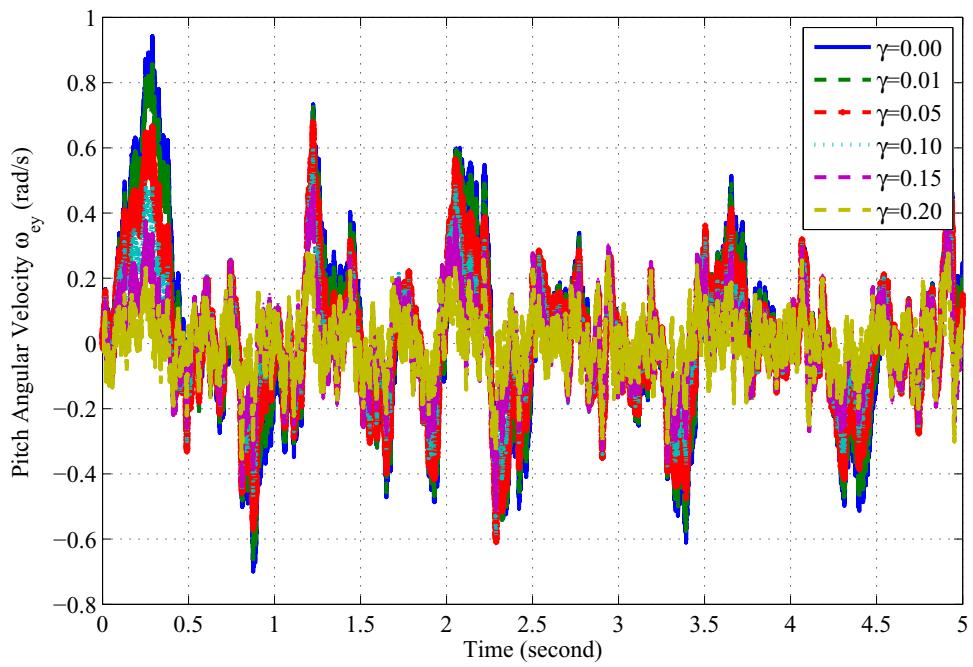


Figure 5.18: Angular velocity ω_{ey} responses with MCV controllers

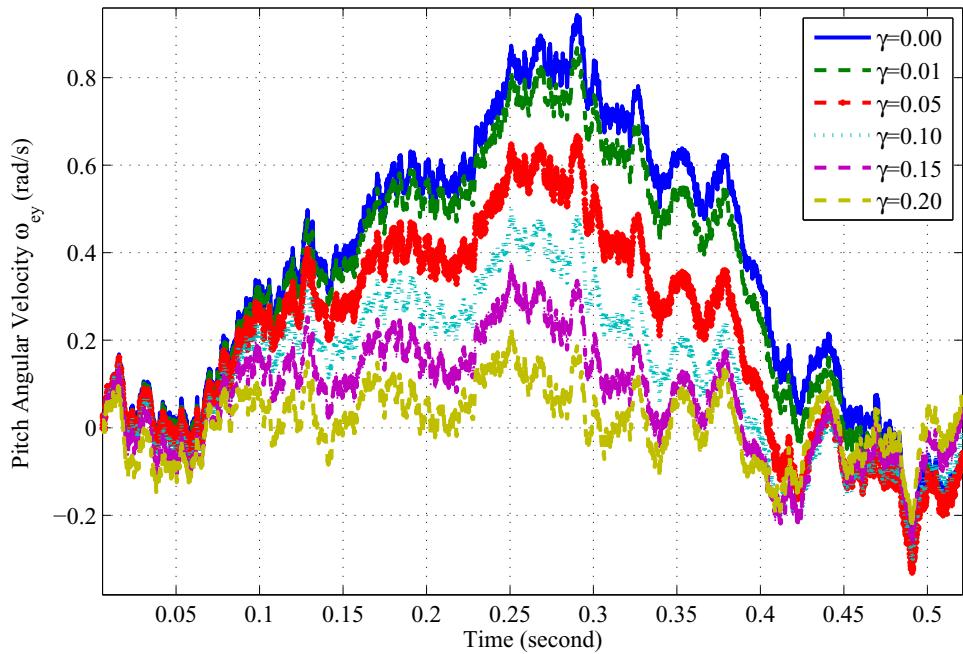


Figure 5.19: Angular velocity ω_{ey} responses with MCV controllers (zoomed view)

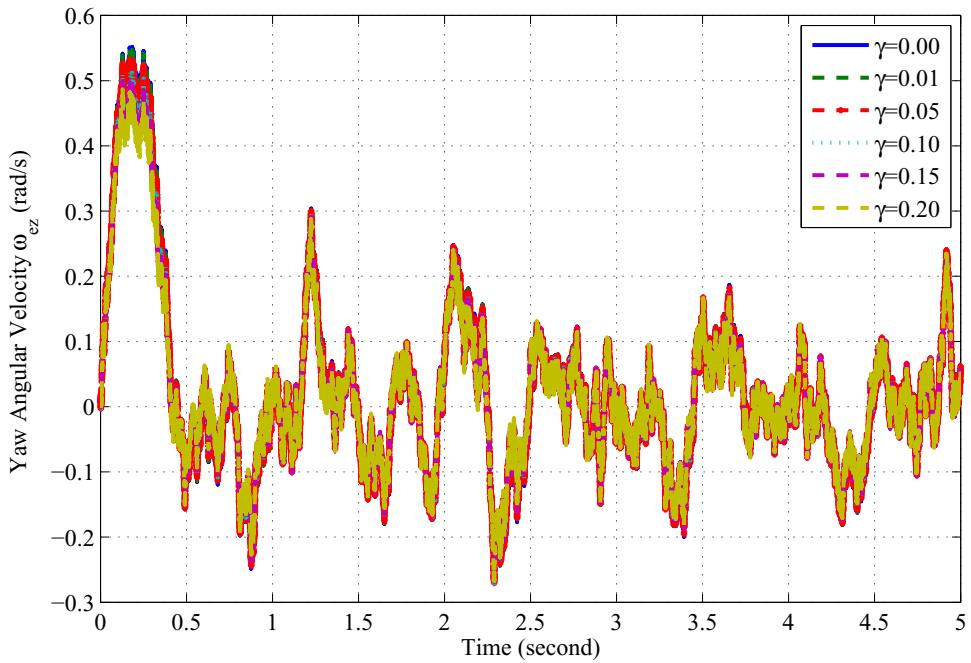


Figure 5.20: Angular velocity ω_{ez} responses with MCV controllers

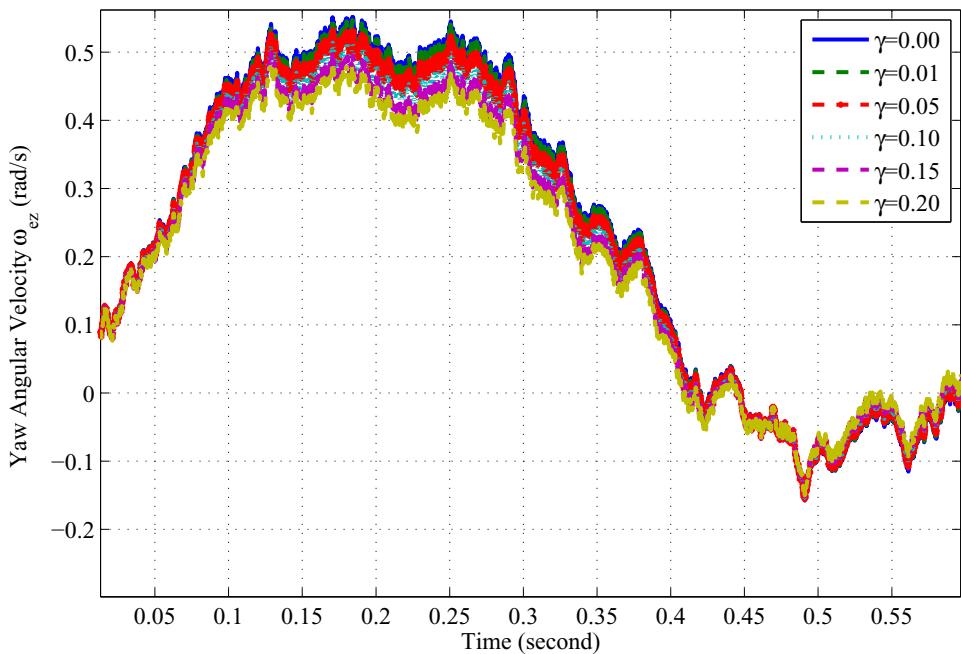


Figure 5.21: Angular velocity ω_{ez} responses with MCV controllers (zoomed view)

responses. From pointing response graphs, we see the pointing response values reduce as γ increases creating steady state pointing error.

The above angular velocity responses are subject to a Gaussian white noise generated from a fixed seed 23341. To observe the average angular velocity response, the simulation was run 100 times with 100 different seeds chosen randomly. The averaging effect will smooth the ripples in the responses. It helps viewing the differences in the responses.

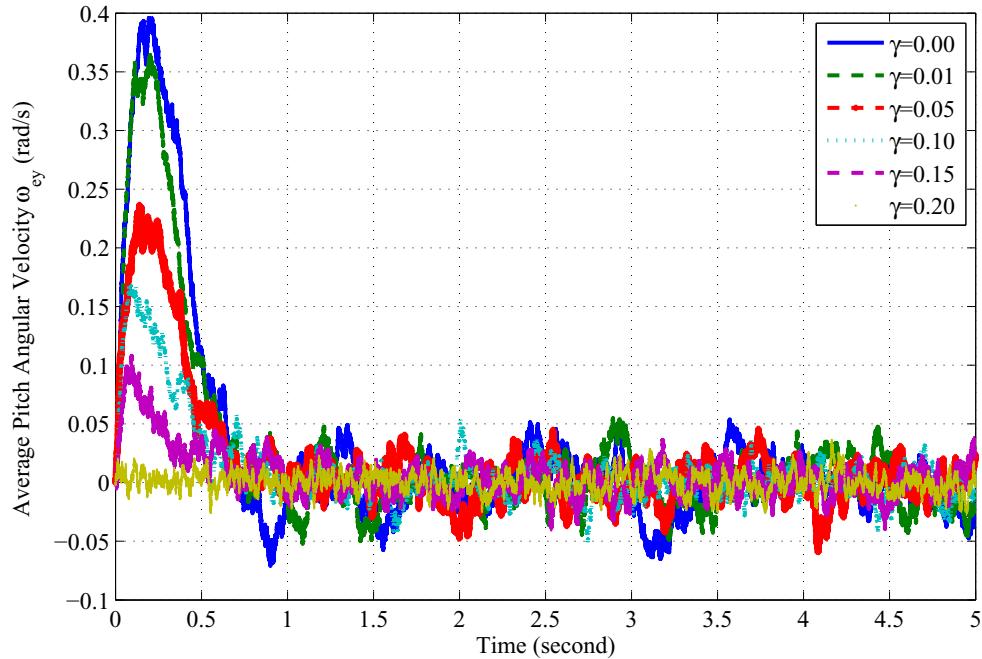


Figure 5.22: Average angular velocity ω_{ey} responses with MCV controllers

Fig. 5.22 shows the responses of average angular velocity ω_{ey} for $\gamma = 0$, $\gamma = 0.01$, $\gamma = 0.05$, $\gamma = 0.1$, $\gamma = 0.15$, and $\gamma = 0.2$. Fig. 5.23 is the zoomed view of Fig. 5.22. The responses show that the angular velocity for $\gamma = 0$ vary between 0.4 to -0.06 radian per second. For $\gamma = 0.01$ it varies between 0.36 to -0.05 radian per second. We see as the γ increases from 0, the angular velocity variation decreases.

Fig. 5.24 shows the responses of average angular velocity ω_{ez} for $\gamma = 0$, $\gamma = 0.01$, $\gamma = 0.05$, $\gamma = 0.1$, $\gamma = 0.15$, and $\gamma = 0.2$. Fig. 5.25 is the zoomed view of Fig. 5.24. The responses show that the angular velocity for $\gamma = 0$ vary between 0.49 to -0.06 radian per second. For

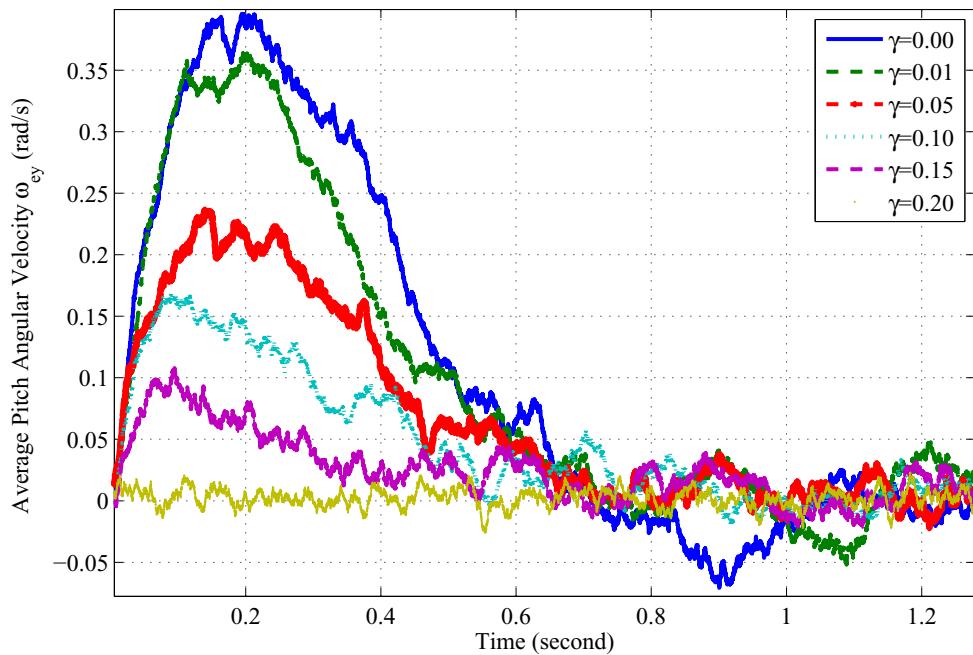


Figure 5.23: Average angular velocity ω_{ey} responses with MCV controllers (zoomed view)

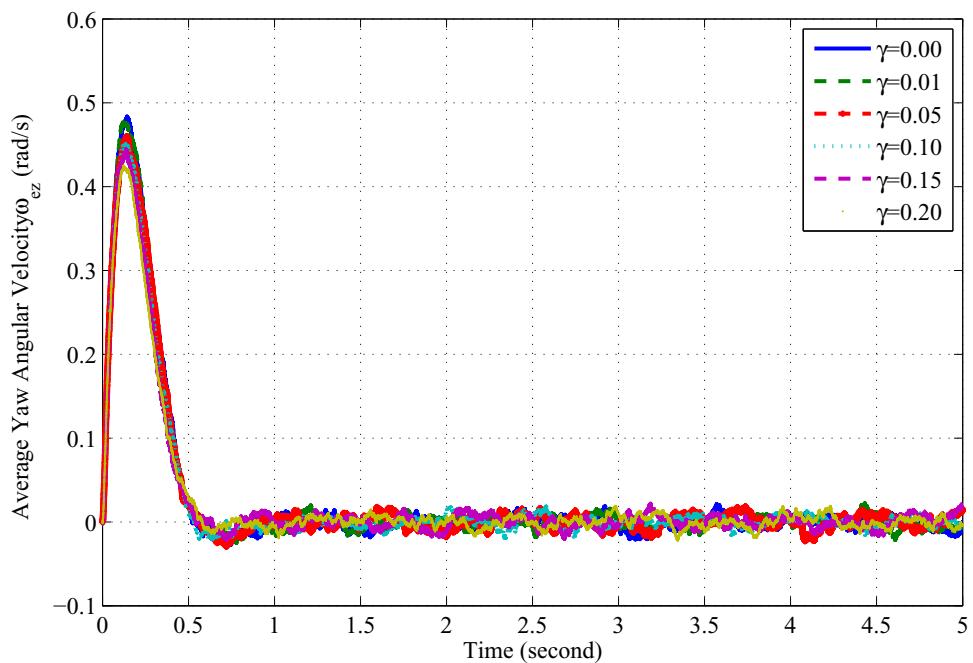


Figure 5.24: Average angular velocity ω_{ez} responses with MCV controllers

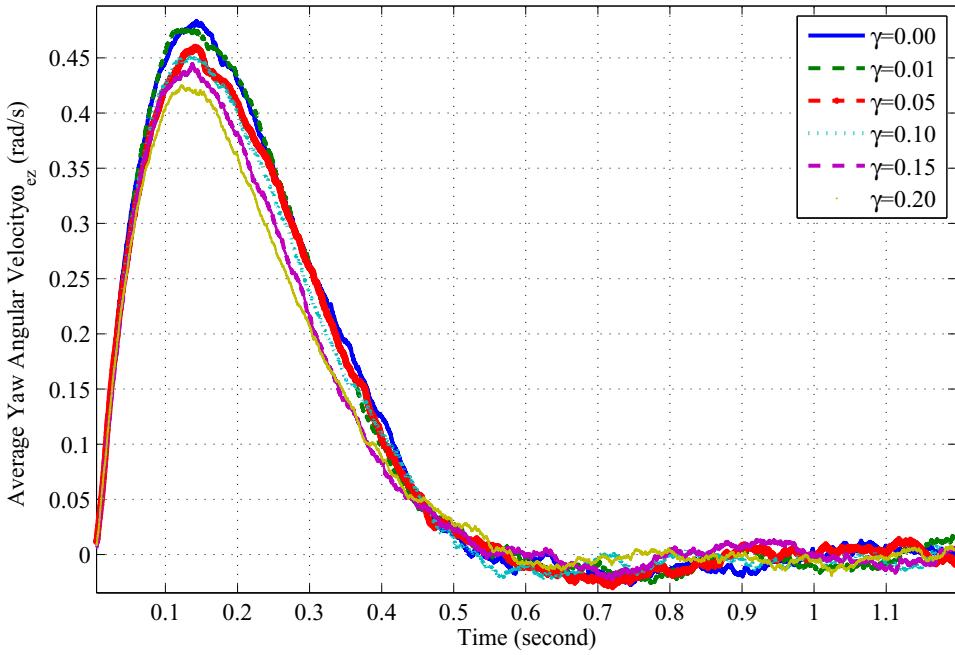


Figure 5.25: Average angular velocity ω_{ez} responses with MCV controllers (zoomed view)

$\gamma = 0.01$ it varies between 0.47 to -0.05 radian per second. We see as the γ increases from 0, the angular velocity variation decreases.

In summary, we see with the increasing γ the peak value of the angular velocity decreases. They settle to approximately zero steady-state value almost at the same time.

5.7.3 RMS Pointing Error with varying γ

We compared the root mean squared pointing error for the statistical controllers with the full-state feedback case. The γ values were from 0 to 0.22 with 0.01 interval. For γ , we ran the simulation 100 times with the Gaussian white noise. There were 100 different randomly generated seeds for each run. This avoids the deterministic nature of selecting a constant as a seed for the Gaussian white noise generation. Then we simulated the position response and obtained average position response. We see that the response reached approximately desired 15.15 cm within 1 second. We discarded the transient part of the position response during RMS pointing error and Standard deviation of pointing error calculation. This is to consider only the

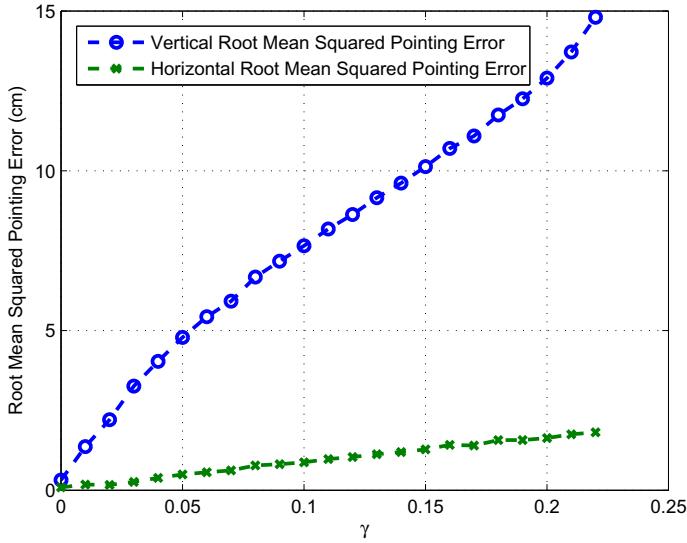


Figure 5.26: RMS pointing error with MCV controllers (full-state feedback)

steady-state part of the pointing response.

Fig. 5.26 shows the root mean squared pointing error with varying γ . We only considered the steady-state part of the position response during the error calculation. It shows that with the increasing value of γ , the vertical root mean squared pointing error $\mu_{\varepsilon,EL}$ increases from 1.73 to 14.68 . The horizontal root mean squared pointing error $\mu_{\varepsilon,AZ}$ increases from 1.28 to 2.3. We see from the graph that the γ values increase the error monotonically. The increasing γ values affect horizontal pointing response less than the vertical pointing response.

5.7.4 Standard Deviation of Pointing Error with varying γ

Fig. 5.27 shows the standard deviation of pointing error with varying γ from 0 to 0.22. It shows that with increasing value of γ , the vertical standard deviation of pointing error decreases from 1.71 to 0.30. The decrease is not monotonic. We found that for $\gamma = 0.02$, it shows larger Standard deviation than LQG case. Also, we see that the variation of γ shows little improvement in case of the horizontal standard deviation of pointing error. The standard deviation decreases from 1.27 to 1.09 with peaks and dips in the trend.

In summary, we can select γ values that can improve the standard deviation of pointing error

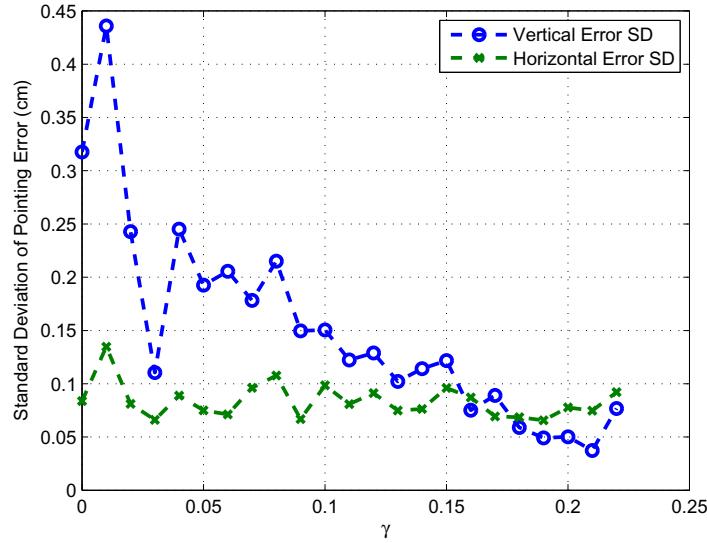


Figure 5.27: Standard Deviation of pointing error with MCV controllers (full-state feedback)

performance. However, the system may suffer from a larger pointing error than the LQG case.

5.8 Output Feedback LQG Control

Since all the states in the laser target system model is not available for measurement, we simulated the LQG and MCV control for the output feedback case. In this method, we used a Kalman filter to estimate the states, then applied the LQG and MCV control in full-state feedback fashion. The process noise variance was 0.0001 and measurement noise variance was 0.00001. They are uncorrelated with each other. Fig. 5.28 and 5.29 show the noisy measurement of the position state on the top. At the bottom, we plotted the actual response and estimated response. We see that the Kalman filter was able to estimate the states from the noisy measurement with almost no error.

5.9 Comparison of Output Feedback LQG and MCV Control

5.9.1 Pointing Responses

Fig. 5.30 shows the vertical pointing responses of MCV controlled GLTS for $\gamma = 0$, $\gamma = 0.05$, $\gamma = 0.1$, $\gamma = 0.15$, and $\gamma = 0.2$. They all started from the initial condition of 2 cm. The

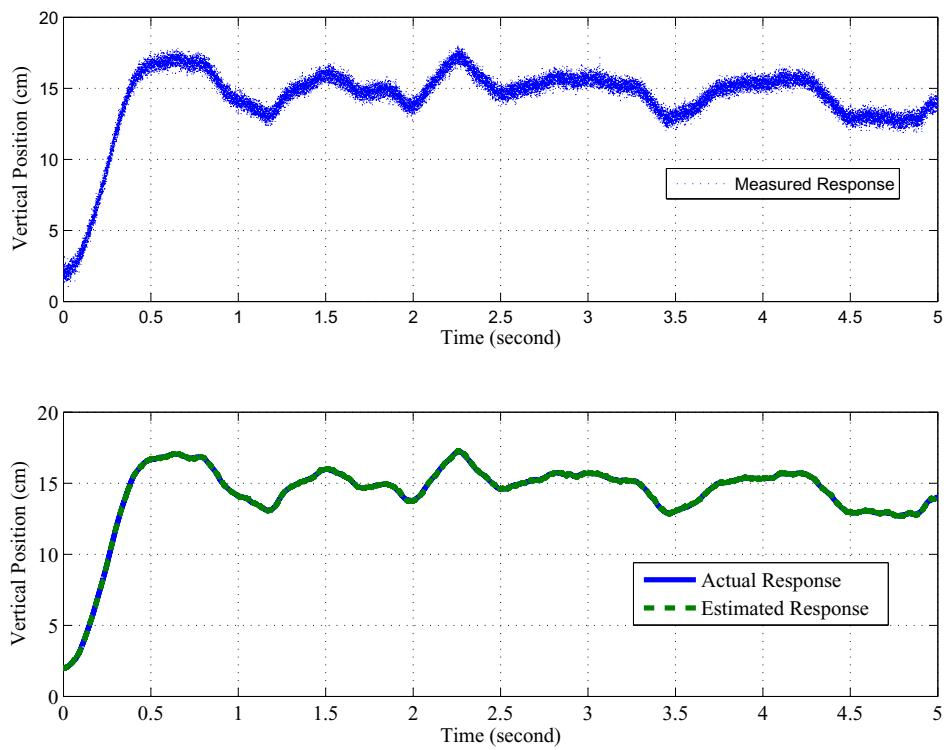


Figure 5.28: Vertical pointing responses (measured and estimated) with LQG controllers

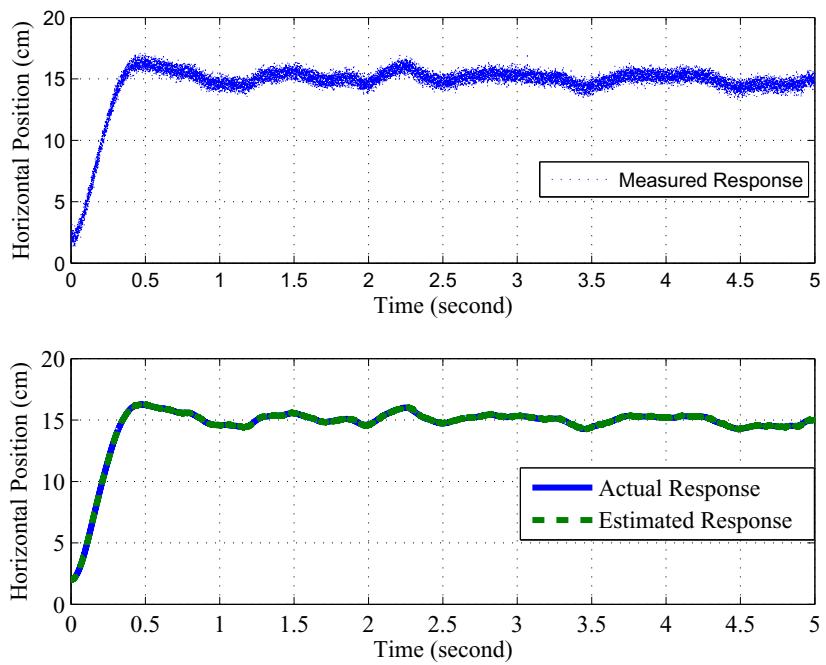


Figure 5.29: Horizontal pointing responses (measured and estimated) with LQG controllers

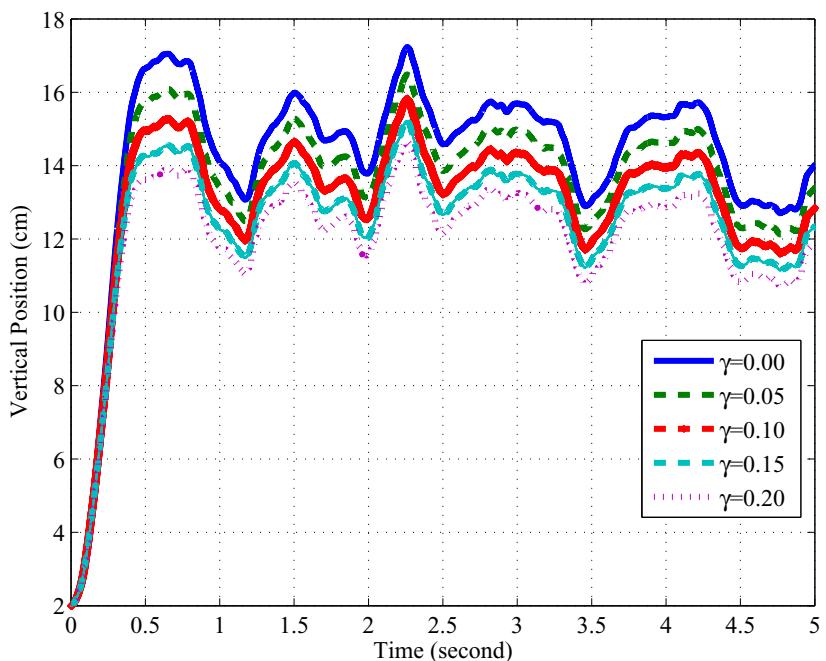


Figure 5.30: Vertical pointing responses with MCV controllers (output feedback)

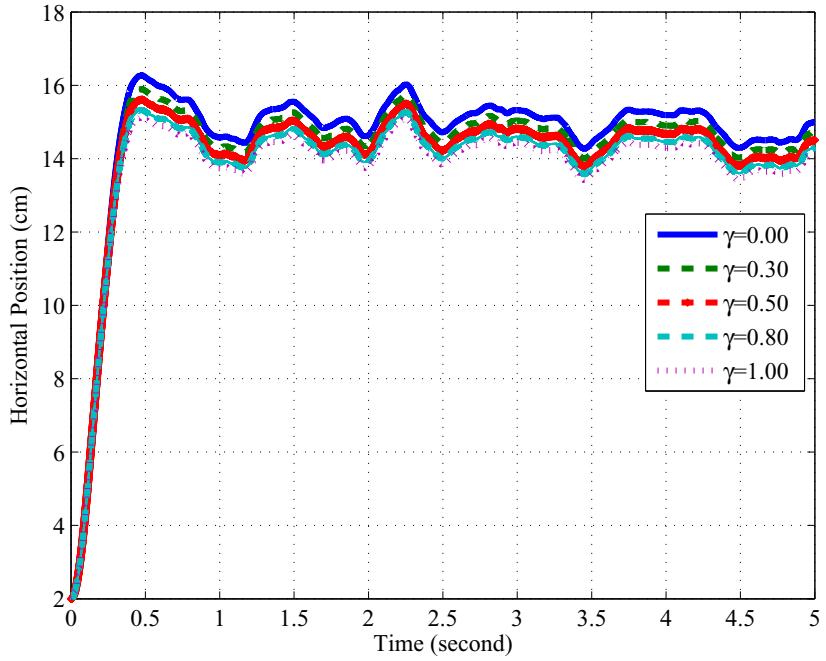


Figure 5.31: Horizontal pointing responses with MCV controllers (output feedback)

desired position response is 15.15 cm. The vertical pointing responses do not approach to the target 15.15 cm for any value of γ . This is because of the applied Gaussian white noise. With the increasing γ values the responses shift downwards reducing the mean values, the ripple in the response decreases.

Fig. 5.31 shows the horizontal pointing responses of MCV controlled GLTS for $\gamma = 0$, $\gamma = 0.3$, $\gamma = 0.5$, $\gamma = 0.8$, ,and $\gamma = 1.0$. They show the same trend as the vertical responses. They are less affected by the applied Gaussian white noise than the vertical responses. With the increasing γ , the pointing errors increase.

5.9.2 RMS Pointing Error with varying γ

We compared the root mean squared pointing error for the statistical controllers with the output feedback case. The γ values were from 0 to 1 with 0.05 interval. For γ , we ran the simulation 100 times with the Gaussian white noise. There were 100 different randomly generated seeds for each run. This avoids the deterministic nature of selecting a constant as a seed for the

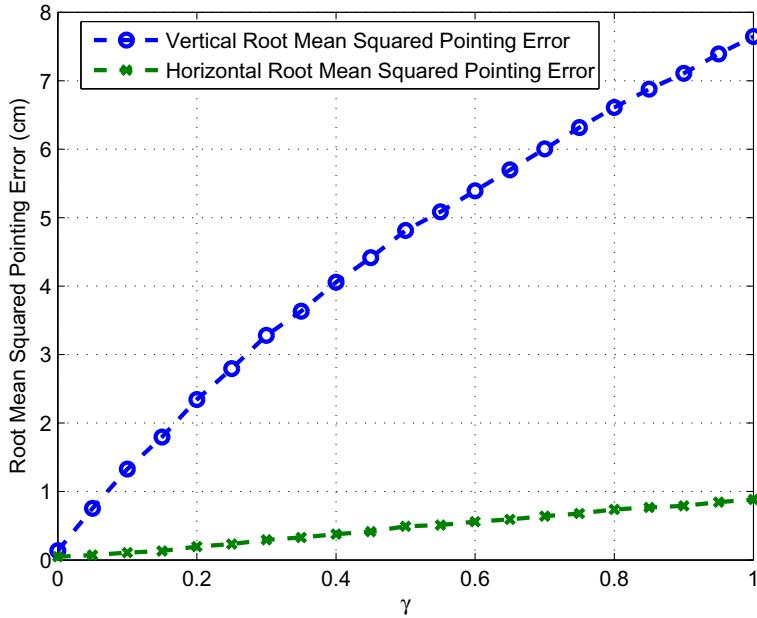


Figure 5.32: RMS pointing error with MCV controllers (output feedback)

Gaussian white noise generation. Then we simulated the position response and obtained average position response. We see that the response reached approximately desired 15.15 cm within 1 second. We discarded the transient part of the position response during RMS pointing error and Standard deviation of pointing error calculation. This is to consider only the steady-state part of the pointing response.

Fig. 5.32 shows the root mean squared pointing error with varying γ from 0 to 1. It shows that with the increasing value of γ , the vertical root mean squared pointing error increases from 0.14 to 7.64. The horizontal root mean squared pointing error increases from 0.05 to 0.88. We see a monotonic increasing trend in the mean pointing error with increasing γ values. The curve for vertical mean pointing error is steeper than the horizontal response. The increasing γ values affect horizontal pointing response less than the vertical pointing response.

5.9.3 Standard Deviation of Pointing Error with varying γ

The Standard deviation of the pointing error was found in the similar manner as in previous section. We compared the Standard deviation of pointing error for different statistical

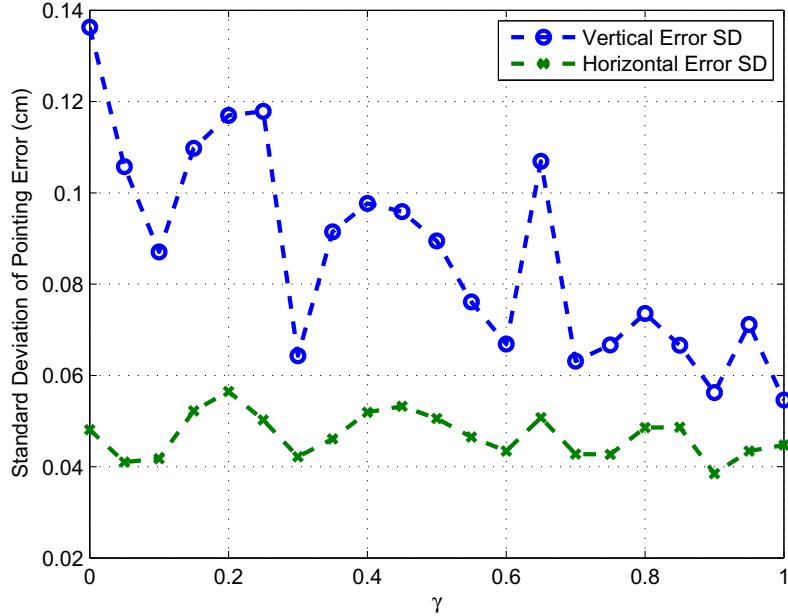


Figure 5.33: Standard Deviation of pointing error with MCV controllers (output feedback)

controllers. Fig. 5.33 shows the standard deviation of pointing error with varying γ from 0 to 1. It shows that with increasing value of γ , the vertical standard deviation of pointing error decreases from 0.15 to 0.05. The variation of γ shows little improvement in case of the horizontal standard deviation of pointing error. The standard deviation decreases from 0.05 to 0.04 for 0 and 1. In between these two values, we see the trend is not monotonic. For the vertical response, all the non-zero γ values show decrease in standard deviation than that of the LQG case. For the horizontal response, not all the non-zero γ values show the decrease than that of LQG case.

In summary, it is possible to select a γ value that can improve the standard deviation of pointing error performance compared to LQG. In that case, the system suffers from a larger pointing error than the LQG case.

Thus, in this chapter, we presented the simulation results of the system using PID, full-state feedback LQG, MCV, and output-feedback LQG, MCV. In the next chapter, we present the hardware implementation of the controllers and the hardware experiment results.

CHAPTER 6

HARDWARE IMPLEMENTATION AND EXPERIMENT OF CONTROLLERS

6.1 Overview

In this chapter, we present the hardware implementation of PID, LQG, and MCV controllers. We used a dSPACE controller board for hardware implementation of the controllers. dSPACE (Digital Signal Processing and Control Engineering) is a software and hardware intended to facilitate interfacing of Simulink models to hardware devices in real time. We shall compare the developed model and the actual physical model with an open loop position control experiment. This will be a validation experiment for the developed mathematical model. Then PID, LQG, and MCV controllers will be implemented.

The objective of the experiments conducted here is to evaluate the controllers' pointing performance of the hardware. The initial position of the laser dot is (2.02 cm, 2.02 cm) in a custom defined x and y axis. The desired position is (15.15 cm, 15.15 cm). All the controllers will be tested using these same initial conditions, and the desired position.

6.2 A Brief Overview of Hardware and Software Set-up

The hardware block diagram is shown in Fig. 6.1. A detailed Hardware and Software block diagram is given in 6.2. The major components of the system are

1. Gimbal Assembly,
2. Motors, Amplifiers, and Power Supply,
3. dSPACE Controller Board,
4. Host Computer for Controller and Image Processing,
5. Image-based Position Sensor,

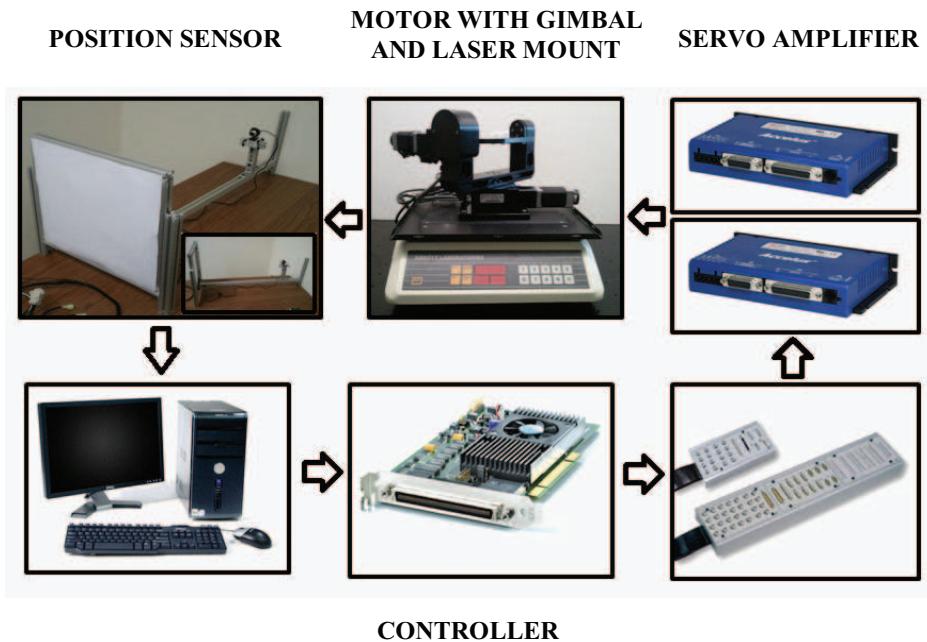


Figure 6.1: Hardware implementation of a gimbaled laser target system

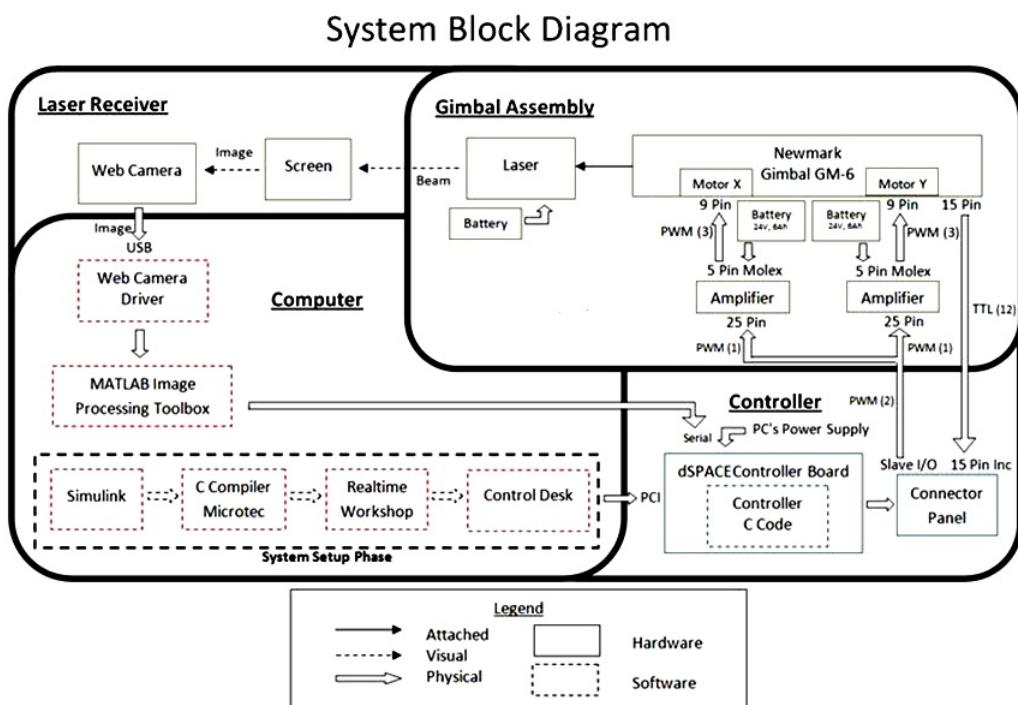


Figure 6.2: Block diagram of hardware and software of gimbaled laser target system

6. Linear Positioner for Disturbance.

6.2.1 Gimbal Assembly

The laser target system consists of a two-axis gimbal (pitch and yaw). The GM-6 gimbal system (Newmark Systems Inc., CA, USA) is equipped with rotary stages. The accuracy of the gimbal is 70 arc-sec. The travel range is $\pm 90^\circ$ for azimuth and elevation. The gear ratio 90 : 1 for both azimuth and elevation axis. Fig. 6.3 and 6.4 show the front and rear view of the gimbal.



Figure 6.3: Newmark GM-6 gimbal (front view)



Figure 6.4: Newmark GM-6 gimbal (rear view)

6.2.2 Laser

A laser is mounted on the pitch axis. This is a 50 mW red laser. The radius of the laser dot is approximately 5 mm. The laser dot position on a screen is used as the visual feedback for the system. Fig. 6.5 shows the laser pointer.



Figure 6.5: Laser pointer

6.2.3 Motors, Amplifiers, and Power Supply

The system used two brushless dc servo motors (Newmark Systems Inc., CA, USA)(Fig. 6.6). These are controlled by Pulse Width Modulated (PWM) signal generated from two amplifiers (Copley Controls Corp, Canton, MA, USA)(Fig. 6.7). The amplifiers can be configured using software called CME2 (Control Motion Explorer) provided by the hardware vendor. The system is powered by a 42 volt dc power source.

6.2.4 Host Computers for Controller and Image Processing

A PC is used for hosting dSPACE controller card, image processing, and control algorithm implementation. The PC has an Intel Pentium Processor 3.40 GHz, and 2GB of RAM. For image processing and control algorithm implementation Matlab R2010b and Simulink were used. Another PC is used for hosting CME2 software. This is used for manual jogging of the motor. There is one PC for hosting software for controlling linear positioner.



Figure 6.6: Brushless DC servor motor



Figure 6.7: Copley controls amplifier

6.2.5 dSPACE Controller Board

dSPACE (dSPACE GmbH, Germany) is a popular hardware and software system used for Rapid Control Prototyping (RCP). It is easy to implement control algorithm taken from mathematical model and use it as a real-time application. This way the control strategy can be tested with the actual control system.

dSPACE controller board comes with a DSP based DS1104 controller card (Fig. 6.8), and a connector panel CP1104 (Fig. 6.9). DS1104 controller card is a single-board system with real-time hardware and comprehensive I/O. The board can be installed in 5V PCI slot. The CP1104 I/O board has eight ADC inputs and eight DAC outputs. They can be accessed via BNC connectors. There are Digital I/O, Slave DSP I/O, Incremental Encoder Interfaces, and Serial Interfaces on the panel. They can be accessed via sub-D connectors. The CP1104 I/O board is an input/output interface between DS1104 controller card and amplifiers.

Simulink (Mathworks Inc, Natick, MA) is a block diagram environment for multidomain simulation and Model-Based Design. It supports simulation, automatic code generation, and continuous test and verification of embedded systems. It provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB. It is possible to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis. Using Simulink's Real Time Workshop (RTI) (Fig. 6.10) combined with dSPACE's Real Time Interface, the Simulink model can be compiled in C code. Then this real-time model can be downloaded into board, and started automatically. Simulink generates a *.sdf (System Description File) file when the model is converted into real-time. This file gives access to the variables of Simulink model in Control Desk software (dSPACE GmbH, Germany). This software is used for graphing, data capturing and real-time parameter tuning.

The dSPACE controller board CP1104 is connected with the computer via an interface card. Simulink is used to design a controller and interface it with the outside plant. Then using a seamless combination of Matlab Real-Time Workshop (RTW) and dSPACE Real-Time Inter-



Figure 6.8: DS1104 controller card



Figure 6.9: CP1104 connector panel

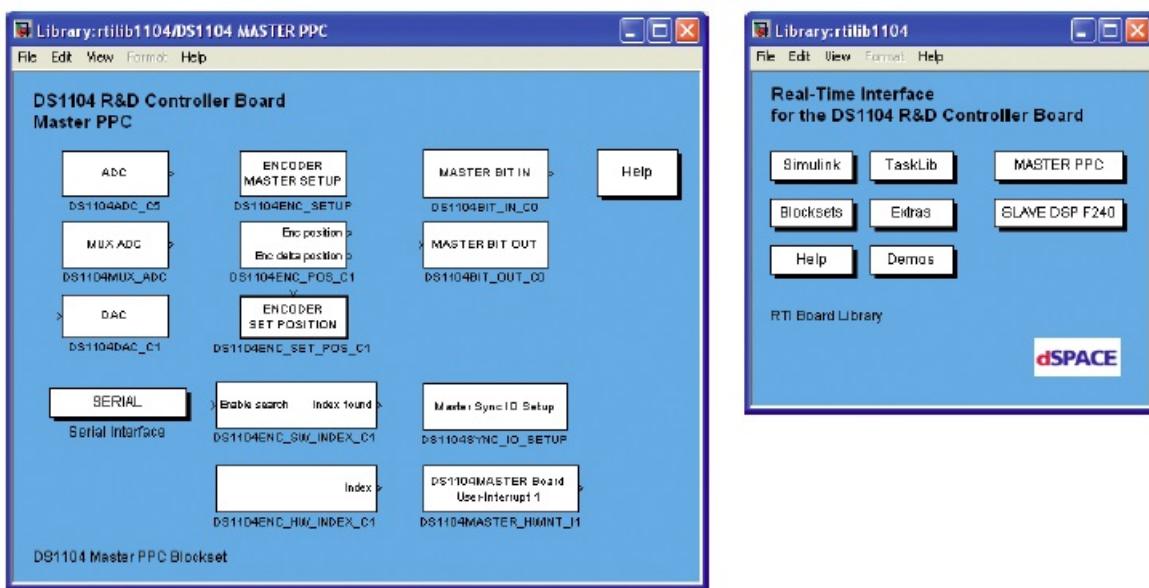


Figure 6.10: dSPACE RTI Blockset

face (RTI), the Simulink model is compiled into a C code. Then this C code is downloaded on the dSPACE 1104 card for real-time running. dSPACE ControlDesk software is used for graphing, data capturing and real-time parameter tuning.

6.2.6 Image-based Position Sensor

The system uses an image-based position sensor which consists of a white screen and webcam sitting in front of the laser. The laser dot hits the screen, the pixel is captured by webcam and sent to Matlab/Simulink image processing program as shown in Fig. 6.11.

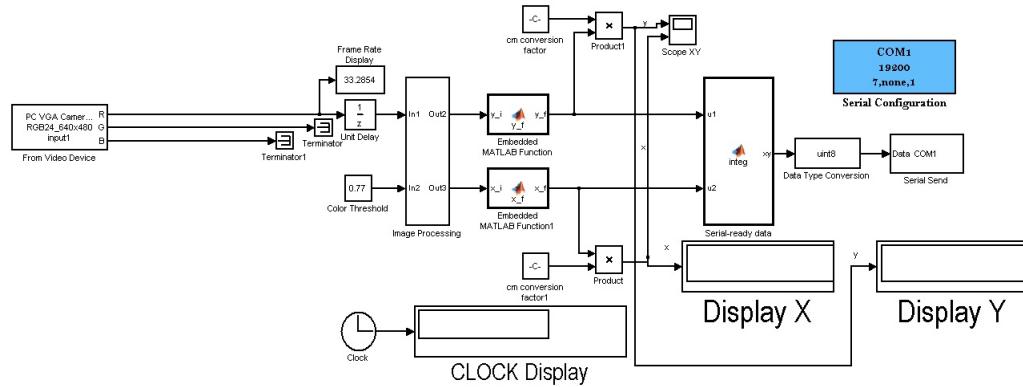


Figure 6.11: Simulink block diagram for image processing

The dSPACE controller board receives the current position of the laser dot serially from Simulink. The laser points to a white screen, measuring $30.3\text{cm} \times 30.3\text{cm}$. For this system, the yaw axis corresponds to the x-axis pictured in Fig. 6.12 and the pitch axis corresponds to the y-axis; the terms will be used interchangeably for the remainder of the paper. The laser is positioned in “front” of the screen at approximately $(0, 0, 148\text{ cm})$ and pointing towards it (i.e. shining in the negative z direction). “Behind” the screen is a digital camera, mounted at approximately $(0, 0, -76)$. From Fig. 6.12 it follows that the center of the screen is located at $(15.15, 15.15, 0)$, which will be considered “home” position.

The yaw and pitch axes have a separate controller for each axis. Each axis controller is similar to the other. Therefore, whenever the specifics of the controller are discussed below, the

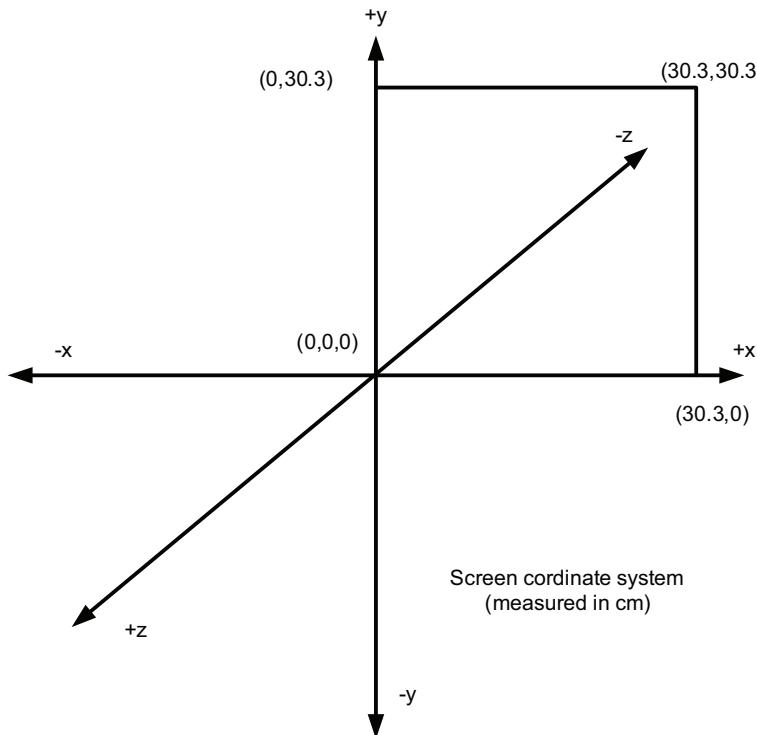


Figure 6.12: Screen coordinate system measured in centimeter

similar details apply to both the x and y axis, and, unless otherwise noted.

6.2.7 Linear Positioner for Disturbance Generation

Linear positioners are designed for a variety of applications in research and other industrial areas requiring precision positioning. We use a NLS4 Linear Stage with motion controller (Newmark Inc., CA, USA) as shown in Fig. 6.13. It has two-axis of motion in x and y direction. The two-axis gimbal base is kept on this linear stage.

6.3 System with Open Loop Position Control

An open loop position controller was first simulated in Simulink. Then the controller was implemented in dSPACE controller board.

Fig. 6.14 shows the Simulink diagram of the controller for yaw axis implemented in dSPACE. A position difference command of 13.13 cm was applied. This command is converted into “ON” time using a speed constant of 0.5536. The speed constant was found experimentally



Figure 6.13: Linear positioner for disturbance generation

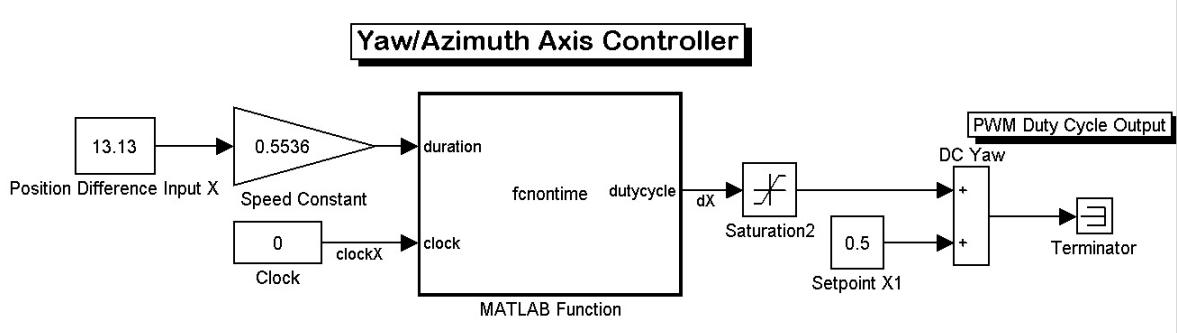


Figure 6.14: Simulink block diagram of yaw axis open loop position controller

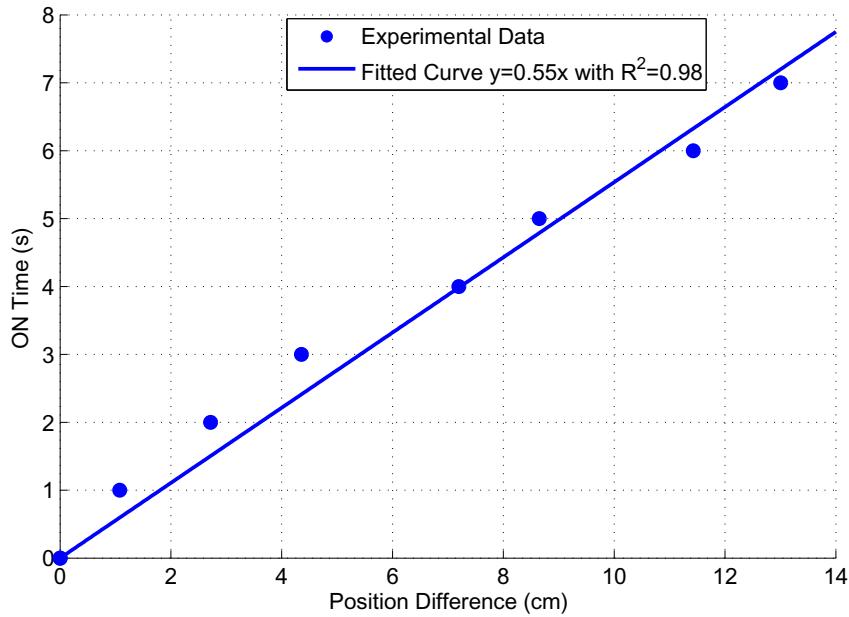


Figure 6.15: Experiment results from yaw axis for open loop position control

from position difference versus ON time graph (see Fig. 6.15). ON time means the simulation time for which the proper duty cycle is applied to move the gimbal. An ON time versus Position difference experiment was performed. In this experiment, a 0.52 duty cycle was applied to the yaw-axis servo amplifier for 1 to 7 seconds with 1 second of increment. Then the position differences were recorded. With the ON time and position difference data, a position difference versus ON time graph was plotted (see Fig. 6.15).

In *fcnontime* function, the ON time is compared to a simulation clock time. This embedded matlab function outputs 0.02 for ON time. After that it outputs 0. The output of the function is then fed into a saturation block with a lower limit of -0.5 and an upper limit of 0.5. This signal is then fed into a summing block where a value of 0.5 is added to it. The end result is an output which is always 0.52 for ON time, and 0.5 for the rest of the period. This Simulink block in Fig. 6.14 was downloaded to the dSPACE controller board to perform the experiment.

This controller ensures the “displacement” for ON time, and “no displacement” for the rest of the period. 0.5 duty cycle corresponds to 0 rpm (rev/min). 0.52 Duty cycle corresponds to 12

rpm. It is found during the experiment 0.52 duty cycle is the suitable speed command. The duty cycle was sufficiently high to move the gimbal from a “rest” position. Also, the laser dot has to be observed within a 30.30×30.30 cm² area. The higher speed would introduce much image processing noise in the captured position signal. The complete control task can be implemented in one function block. The current implementation is done considering the implementation of the other controllers.

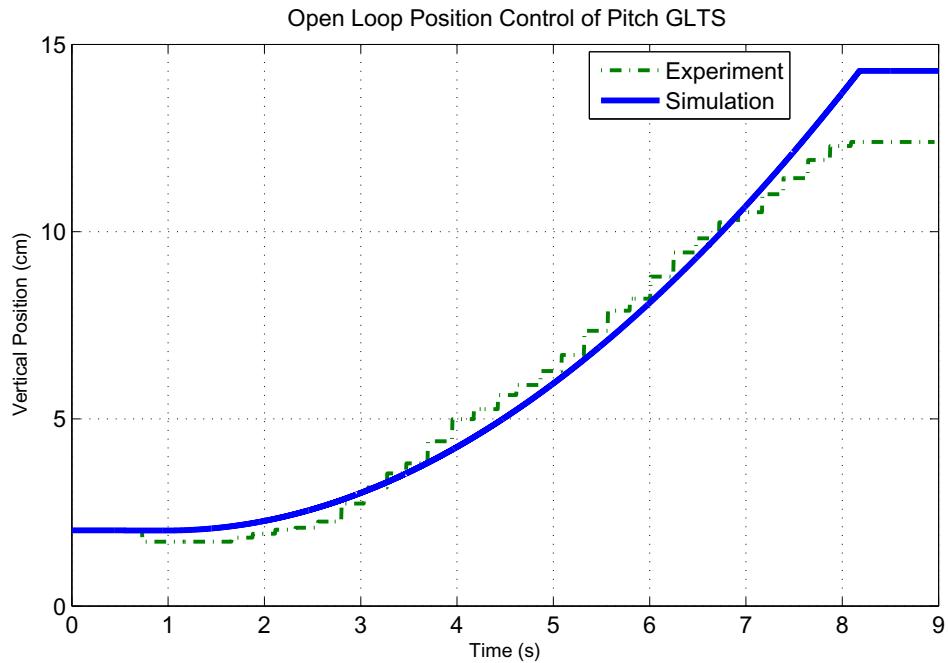


Figure 6.16: Open loop position control for pitch gimbaled laser target system

Fig. 6.16 and 6.17 show the open loop position control response of pitch and yaw Gimbaled Laser Target System from the simulation and the experiment. The experimental curve shows staircase pattern because of down sampling of the captured data from the hardware. In both simulation and experiment, the initial conditions were (2 cm, 2 cm). The pitch gimbaled laser target system reaches the final position of 14.29 cm at 8.18 seconds in simulation. In the experiment, it took 8.2 seconds to reach the final position of 12.39 cm. The yaw gimbaled laser target system reaches the final position of 17.17 cm at 6.9 seconds in simulation. In experiment, it took 8.5 seconds to reach the final position of 17.64 cm. The summary of the responses is

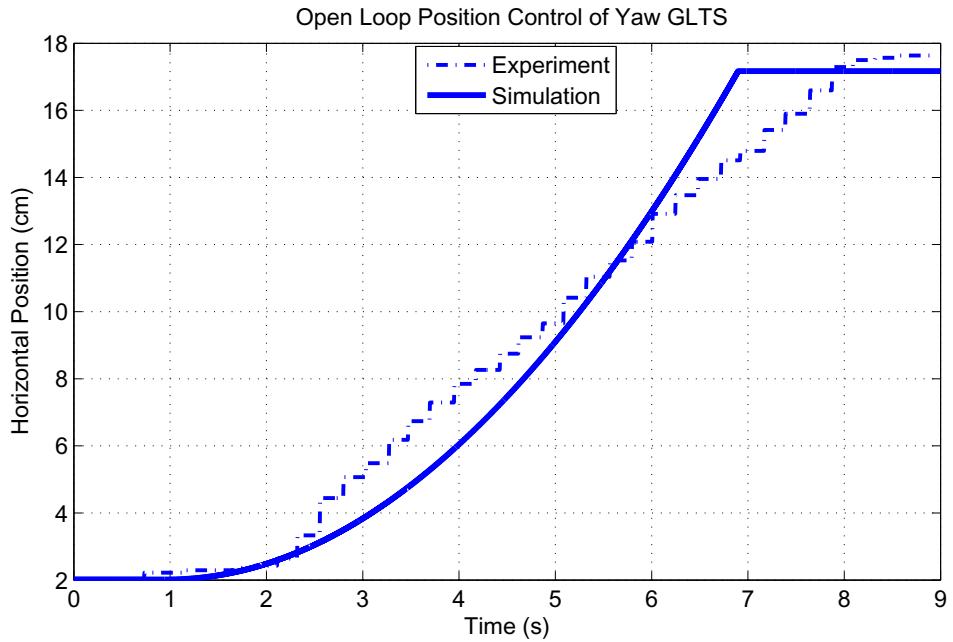


Figure 6.17: Open loop position control for yaw gimbaled laser target system

Table 6.1: Open loop position control results

	Pitch Gimbal Simulation	Pitch Gimbal Experiment	Yaw Gimbal Simulation	Yaw Gimbal Experiment
Final Position (cm)	14.29	12.39	17.17	17.64
Position Error (%)	5.68	18.22	13.33	16.44

shown in Table 6.1.

6.4 System with PID Control: Output Feedback

Fig. 6.18 shows the Simulink diagram of the PID controller for yaw axis implemented in dSPACE. The PID controller block in Simulink receives an error signal from a summing junction. The error signal is the difference between the set-point 15.15 cm and the current output position found from image processing block of position sensor. PID controller parameters K_P , K_I , and K_D used are 0.24 , 3.60×10^{-6} , and 6.40×10^{-3} for pitch gimbal. For yaw gimbal, they are 0.12 , 3.18×10^{-6} , and 3.30×10^{-3} .

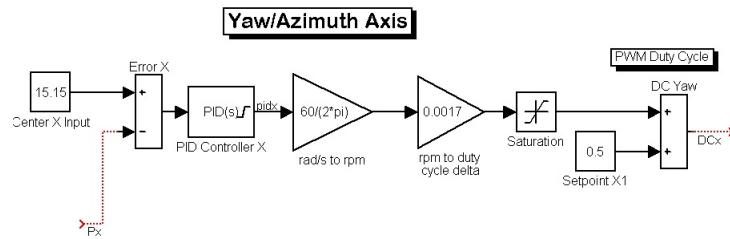


Figure 6.18: Simulink block diagram of yaw axis PID controller

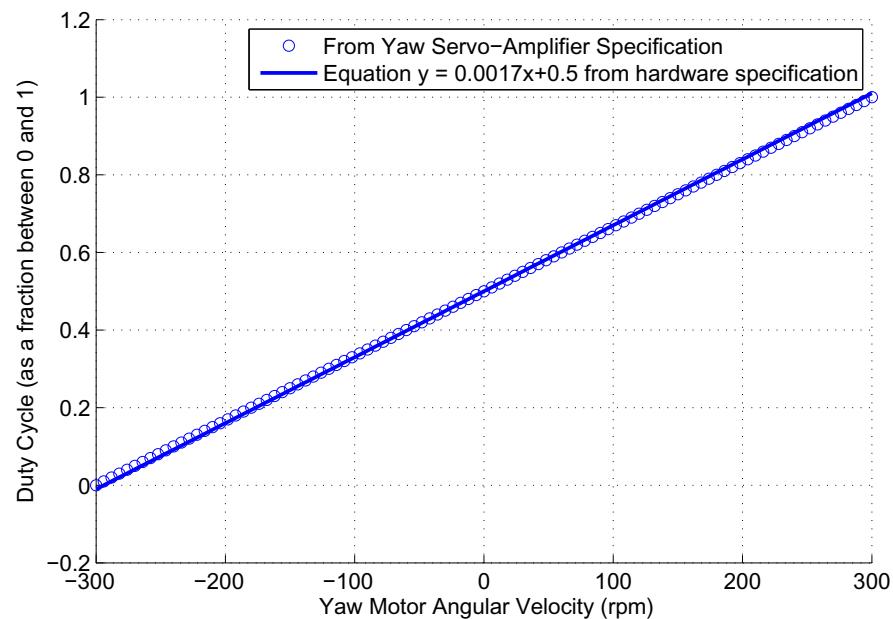


Figure 6.19: Duty cycle and yaw motor angular velocity

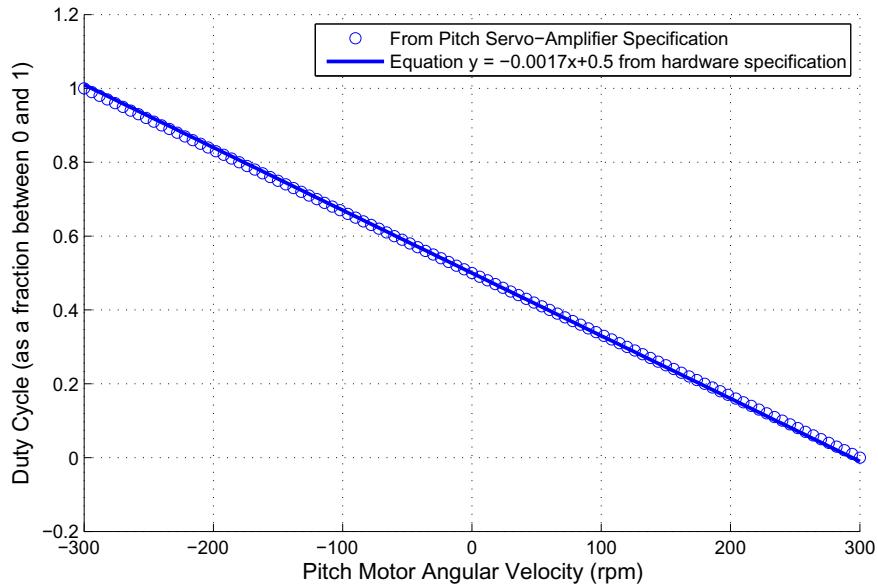


Figure 6.20: Duty cycle and pitch motor angular velocity

PID controller outputs an angular velocity command in rad/s unit. Then it drives pitch and the yaw Gimbaled Laser Target System. The dSPACE implementation of this controller requires some additional blocks. The servo-amplifier in the hardware set up receives angular velocity command in Pulse Width Modulated Signal form. PWM input of 50% type was used. In this mode, there is only one physical connection required for PWM signal going into servo-amplifier. A 50% duty cycle corresponds to zero speed command in the velocity mode of operation of the amplifier. Duty-cycles of 0%, and 100% would result in negative full-scale, or positive full-scale outputs. So, the duty-cycle controls not only the magnitude, but also the polarity of the amplifier outputs.

The relation between the angular velocity (rpm) and the duty cycle (a value between 0 and 1) for the yaw and pitch gimbal are found from the hardware specification (see Fig. 6.19 and 6.20):

$$D_x = 0.0017\omega_{mx} + 0.5, \quad (6.1)$$

$$D_y = -0.0017\omega_{my} + 0.5. \quad (6.2)$$

D_x is the duty cycle for yaw servo-motor, ω_{mx} is the angular velocity in rpm unit. D_y is the duty cycle for yaw servo-motor, ω_{my} is the angular velocity in rpm unit. These equations are implemented using gain and summing block (see Fig. 6.18). For yaw motor, duty cycle 0 corresponds to -300 rpm, 0.50 corresponds to 0 rpm, and 1 corresponds to 300 rpm. Using this information, Fig. 6.19 was drawn. For pitch motor, duty cycle 0 corresponds to 300 rpm, 0.50 corresponds to 0 rpm, and 1 corresponds to -300 rpm. Using this information, Fig. 6.20 was drawn.

A saturation block is used to ensure that the duty cycle value remains within 0.5 and -0.5. This is because of the hardware set-up requirement. This Simulink block in Fig. 6.18 was downloaded to the dSPACE controller board to perform the experiment.

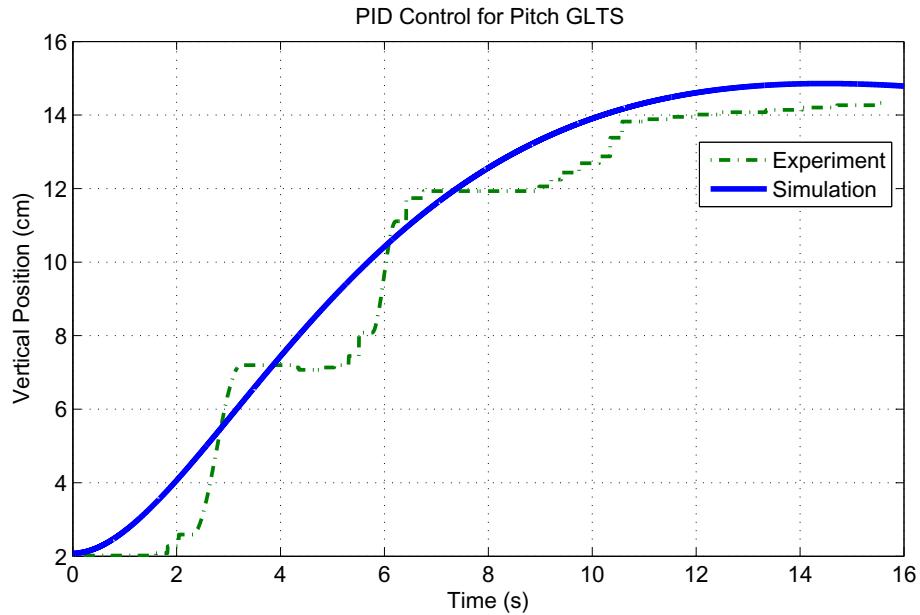


Figure 6.21: PID control for pitch gimbaled laser target system

Fig. 6.21 and 6.22 show the PID control response for pitch and yaw Gimbaled Laser Target System from the experiment and the simulation. The pitch gimbaled laser target system reaches the final position of 14.5 cm at 15.5 seconds in simulation. In the experiment, it took 15.8 seconds to reach the final position of 14.33 cm. The yaw gimbaled laser target system reaches the final position of 16.33 cm at 14 seconds in simulation. In the experiment, it took 14.5

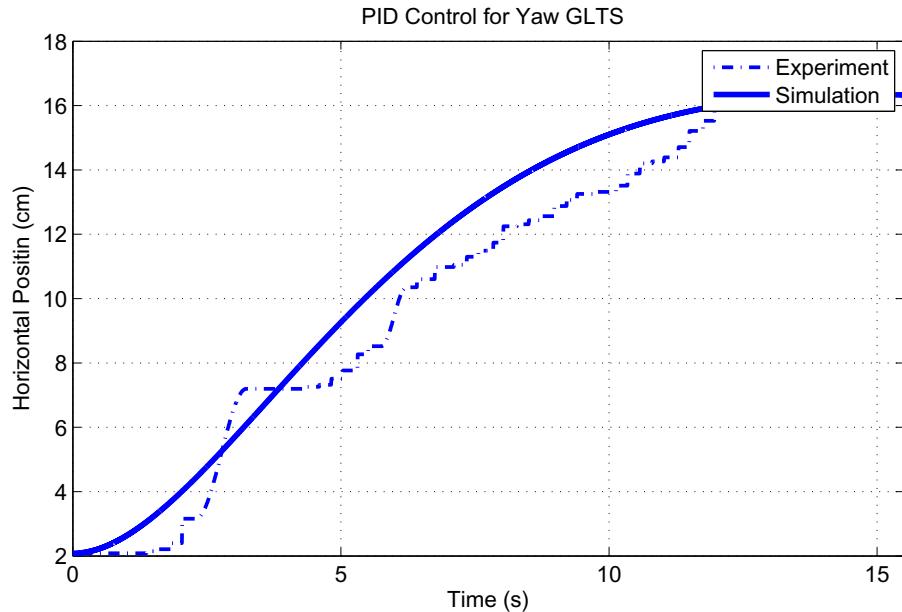


Figure 6.22: PID control for yaw gimbaled laser target system

Table 6.2: PID Control Results

	Pitch Gimbal Simulation	Pitch Gimbal Experiment	Yaw Gimbal Simulation	Yaw Gimbal Experiment
Final Position (cm)	14.5	14.33	16.33	16.22
Position Error (%)	4.29	5.41	7.78	7.06

seconds to reach the final position of 16.22 cm. We see at several points the experimental curve did not follow the simulation curve. It seems sometimes the system was only controlled by yaw-axis or pitch-axis control signal, even though the control signal application was simultaneous. This occurred because of the coupling between the axes. For this sometimes only one axis motion dominated the resultant motion. The summary of the responses is shown in Table 6.2.

For two-axis PID control the simulink block diagram used is shown in Fig. 6.23

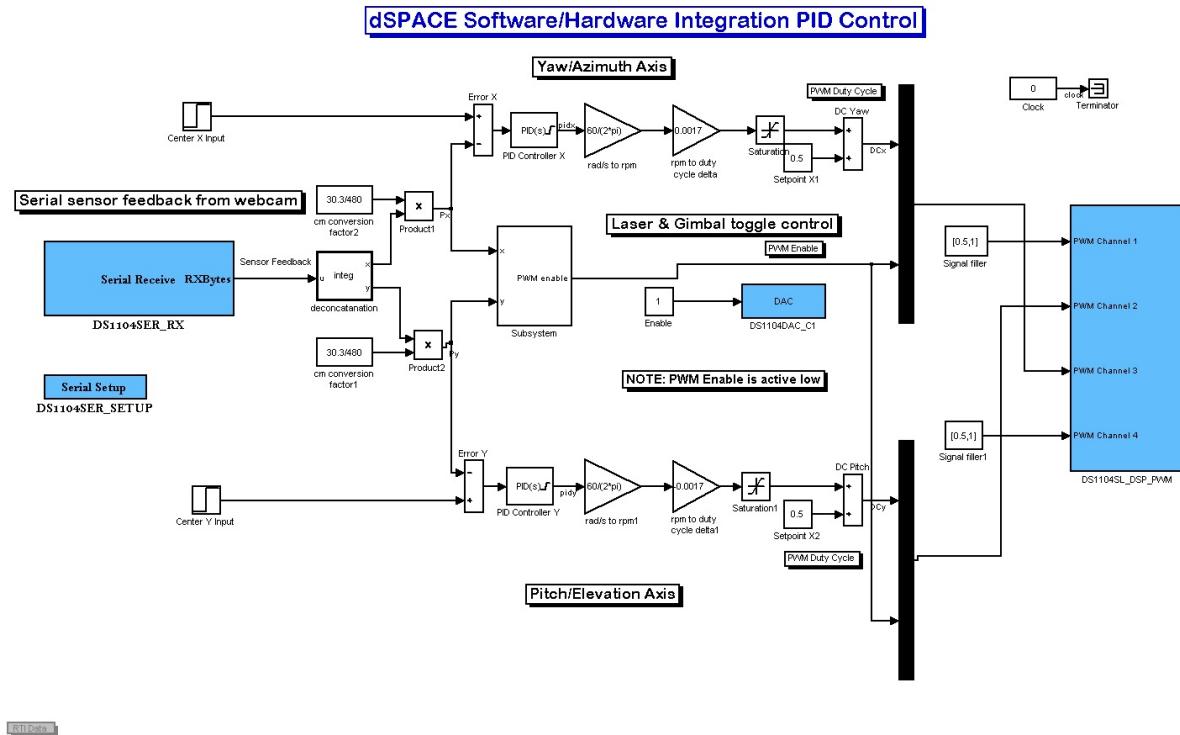


Figure 6.23: Simulink block diagram of two-axis PID controller

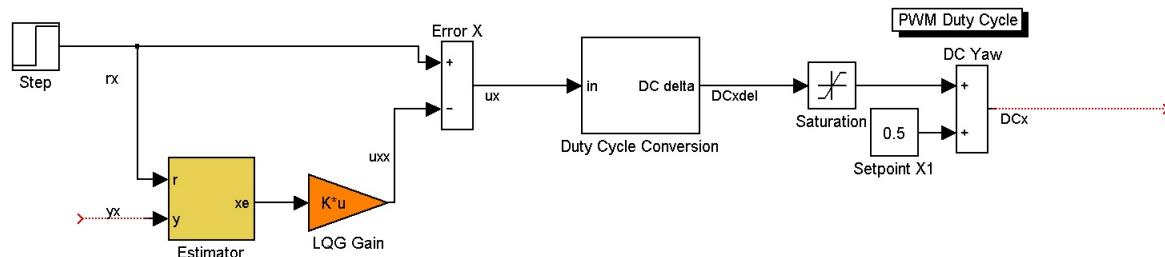


Figure 6.24: Simulink block diagram of yaw axis LQG controller

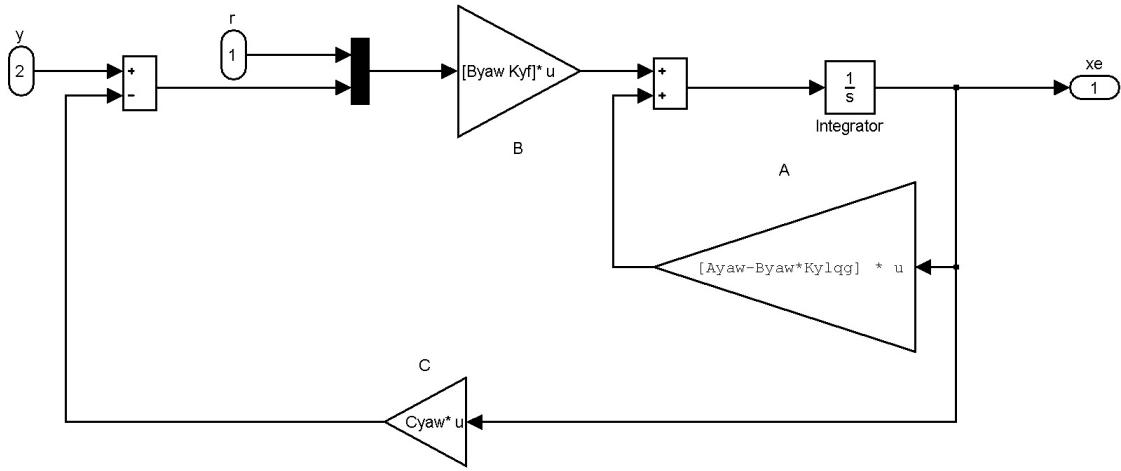


Figure 6.25: Simulink block diagram of Kalman estimator for yaw axis LQG controller

6.5 System with LQG Control: Output Feedback

Fig. 6.24 shows the Simulink diagram of the LQG controller for yaw axis implemented in dSPACE. The estimator block receives two inputs: the desired set-point position and the position output measured by the webcam and the image processing block. The estimator uses a Kalman filter to compute the estimated states. Fig. 6.25 shows the simulink block diagram of the estimator. The full-state feedback LQG gain is used to construct state feedback control. Then this control signal is subtracted from the set-point. This signal is converted to duty cycle using the speed command and duty cycle relationship. Then it is applied to the amplifier. We calculated the Kalman filter gain and the full-state feedback LQG gain using the values used in the output feedback LQG simulation.

For two-axis LQG control the simulink block diagram used is shown in Fig. 6.26

6.6 System with MCV Control: Output Feedback

Fig. 6.27 shows the Simulink diagram of the MCV controller for yaw axis implemented in dSPACE. The estimator block receives two inputs: the desired set-point position and the position output measured by the webcam and the image processing block. The estimator uses a Kalman filter to compute the estimated states. Fig. 6.28 shows the Simulink block diagram

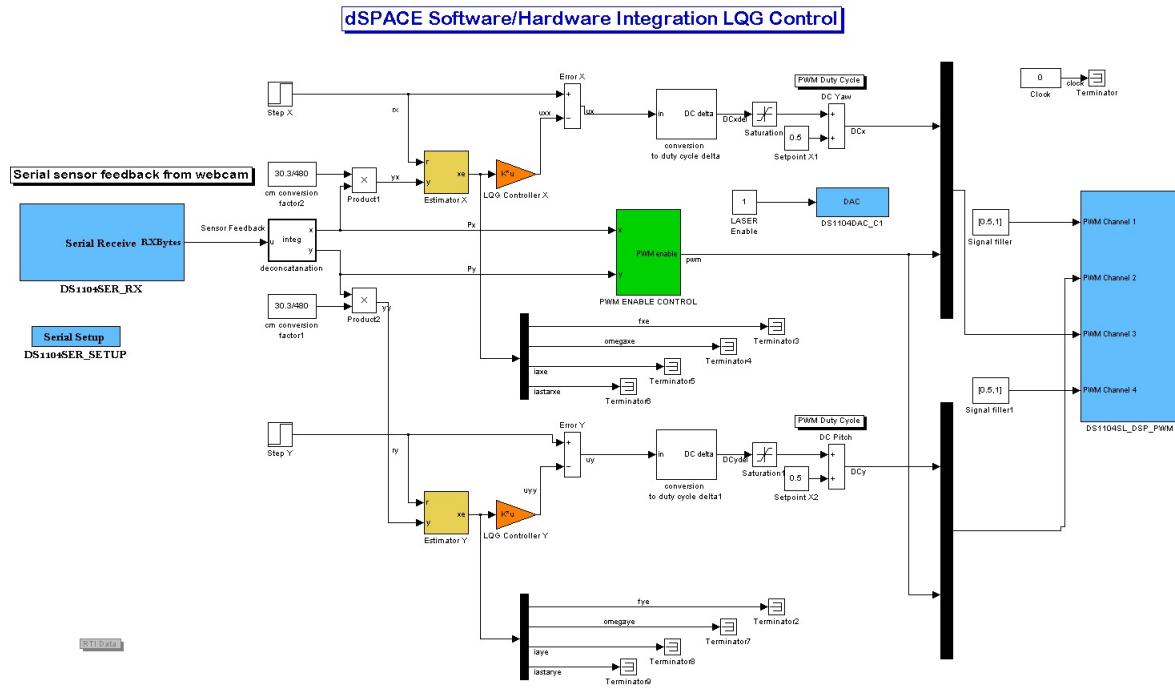


Figure 6.26: Simulink block diagram of two-axis LQG controller

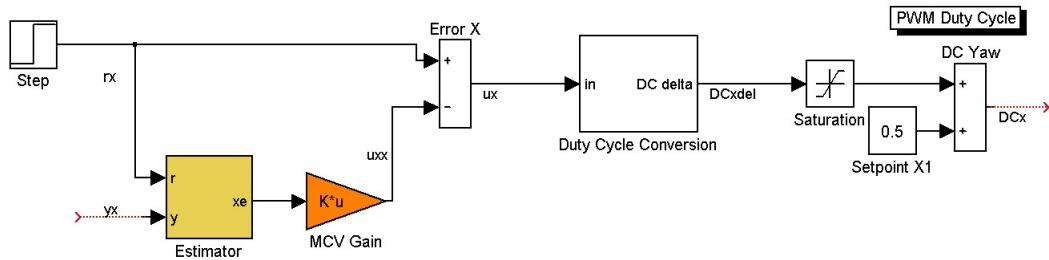


Figure 6.27: Simulink block diagram of yaw axis MCV controller

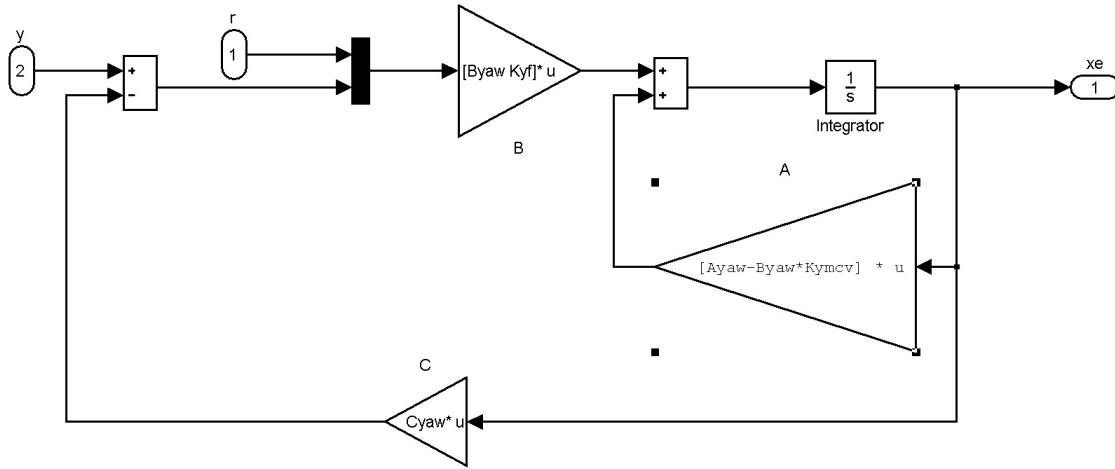


Figure 6.28: Simulink block diagram of Kalman estimator for yaw axis MCV controller

of the estimator. The full-state feedback MCV gain is used to construct state feedback control. Then this control signal is subtracted from the set-point. This signal is converted to duty cycle using the speed command and duty cycle relationship. Then it is applied to the amplifier. We calculated the Kalman filter gain and the full-state feedback MCV gain for different γ using the values used in the output feedback MCV simulation.

For two-axis MCV control the simulink block diagram used is shown in Fig. 6.29

6.7 Experiments for Comparison of System with PID, LQG, and MCV Control

6.7.1 For Control of Yaw Axis Only

The objective of this experiment is to compare the pointing performances of different controllers for yaw axis under Gaussian white noise. We used a PID, LQG, and MCV ($\gamma = 0.1, 0.5, 1.0, 2.0$) controllers.

For $\gamma = 0$ the MCV controller converts to the LQG controller. For $\gamma > 2$, the ordinary differential equation solver fails to calculate state-feedback gain for the developed model and noise statistics. The rest of the values of γ were chosen arbitrarily in between 0 and 2.

The lighting of the room has an effect on the performance of the webcam and the image processing block. We kept the same lighting condition for all the controllers. A darker room

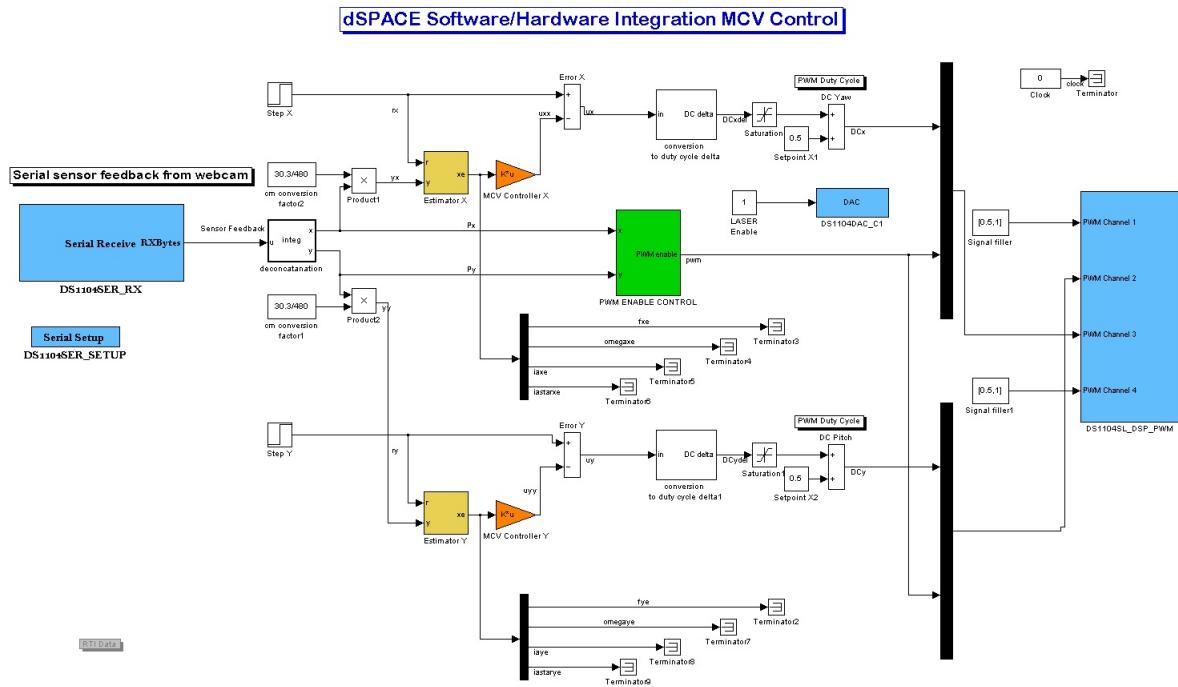


Figure 6.29: Simulink block diagram of two-axis MCV controller

enhances the contrast of the red laser dot on the screen. The contrast enhancement improves the image processing block performance. Therefore, we kept the room relatively dark during the experiment. The experiment duration was 60 seconds for all the controllers.

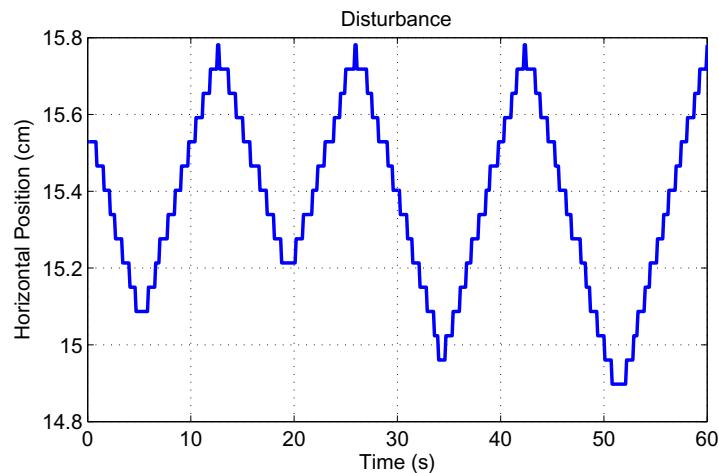


Figure 6.30: Single axis disturbance

Noise sources are categorized into two major categories: process noise and measurement noise. The process noise comes from friction, cable restraint torques, and base or platform motion. The measurement noise comes from the image processing block in Matlab. Friction torques and cable restraint torques are associated with gimbal structure, angular velocity, and angles. For the base or platform motion, we used a linear positioner (Newmark Inc., CA, USA). A random triangular shape disturbance signal command was applied to the positioner. The positioner starts from an absolute position at one velocity, then moves to different relative position at different velocity. We chose the lower peaks of the disturbance signal randomly using a value generated from the real time CPU clock. This makes the signal non-deterministic for the experiment period. Fig. 6.30 shows the disturbance signal used in the experiment.

We initialized the laser dot close to $x = 2$ cm position. Because of the linear positioner motion, we could not keep the laser dot at exactly 2 cm position. We disabled the pitch axis amplifier to avoid coupling effect. The desired final position was $x = 15.15$ cm. For PID control of yaw gimbal, the parameters K_P , K_I , and K_D were 0.12 , 3.18×10^{-6} , and 3.30×10^{-3} . For LQG and MCV, the tuning parameters were,

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and $R = 1$. Then we ran the experiment. Fig. 6.31 shows the PID, LQG, and MCV controllers pointing responses.

It is difficult to distinguish the controller performance from Fig. 6.31. So we plotted RMS pointing error and the standard deviation of the pointing error. Fig. 6.32 and Fig. 6.33 show the RMS pointing error and the standard deviation of the pointing error. From Fig. 6.32 and Fig. 6.33, we selected the best MCV controller. The “best” controller should exhibit the lowest standard deviation of pointing error keeping the RMS pointing error at a tolerable level.

We found that the RMS pointing error for LQG is 2.31 cm. The RMS pointing error for MCV $\gamma = 0.1$ is 2.48 cm, which is larger than the LQG case. But RMS pointing errors for

Comparison of PID, LQG, and MCV Controlled Response for Yaw Gimbaled Laser Target System

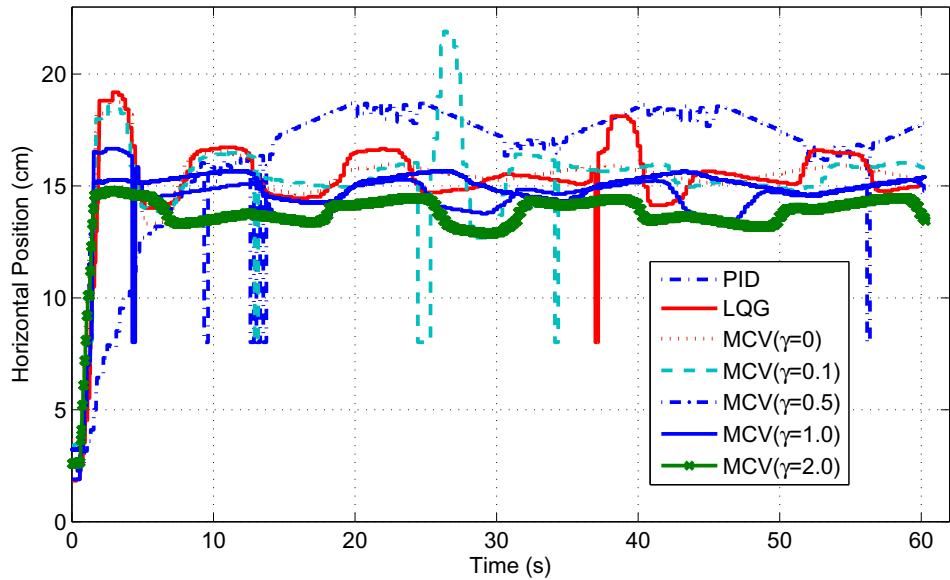


Figure 6.31: PID, LQG, MCV controller comparison

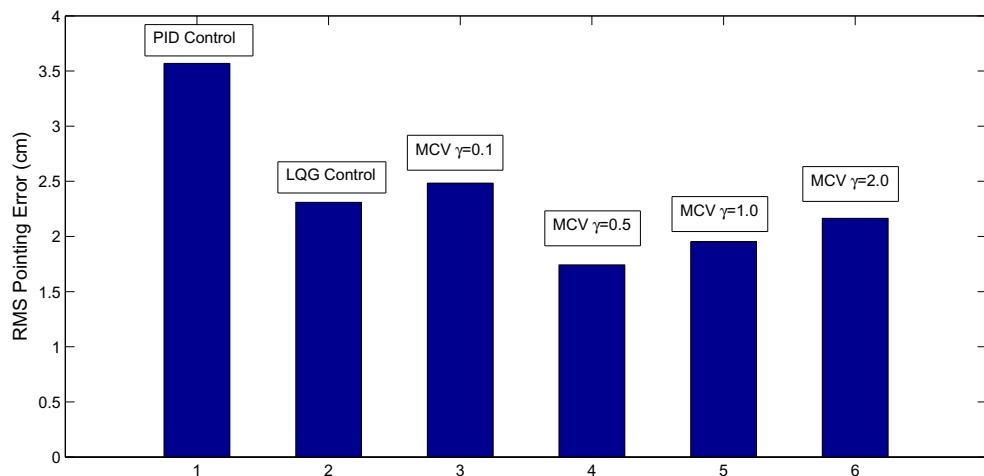


Figure 6.32: RMS pointing error for PID, LQG, and MCV

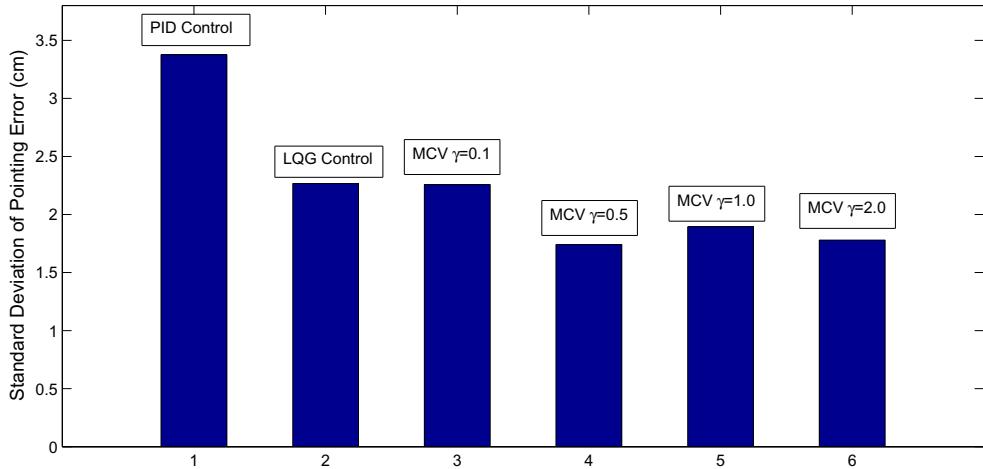


Figure 6.33: Standard Deviation of pointing error for PID, LQG, and MCV

Table 6.3: RMS and Standard Deviation of pointing error for yaw gimbaled laser target system

	RMS Pointing Error (cm)	SD of Pointing Error (cm)
PID	3.57	3.37
LQG	2.31	2.27
MCV ($\gamma = 0.5$)	1.74	1.74

MCV $\gamma = 0.5, 1.0, 2.0$ are less than LQG. Among MCV $\gamma = 0.5, 1.0, 2.0$, MCV $\gamma = 0.5$ RMS error is the lowest – 1.74 cm. MCV $\gamma = 2.0$ RMS error is the highest – 2.46 cm.

Standard deviation of pointing error for MCV $\gamma = 0.1, 0.5, 1.0, 2.0$ are less than LQG (2.27). Among MCV $\gamma = 0.5, 1.0, 2.0$, $\gamma = 0.5$ standard deviation (1.74) is the lowest, $\gamma = 1.0$ error of 1.89 cm is the highest.

In this experiment, all MCV controllers show better performance than LQG both in terms of RMS pointing error and standard deviation of pointing error. MCV ($\gamma = 0.5$) was the best among MCV controllers. From this experiment, we cannot conclude that the increasing gamma will increase the RMS pointing error and decrease the standard deviation of pointing error monotonically.

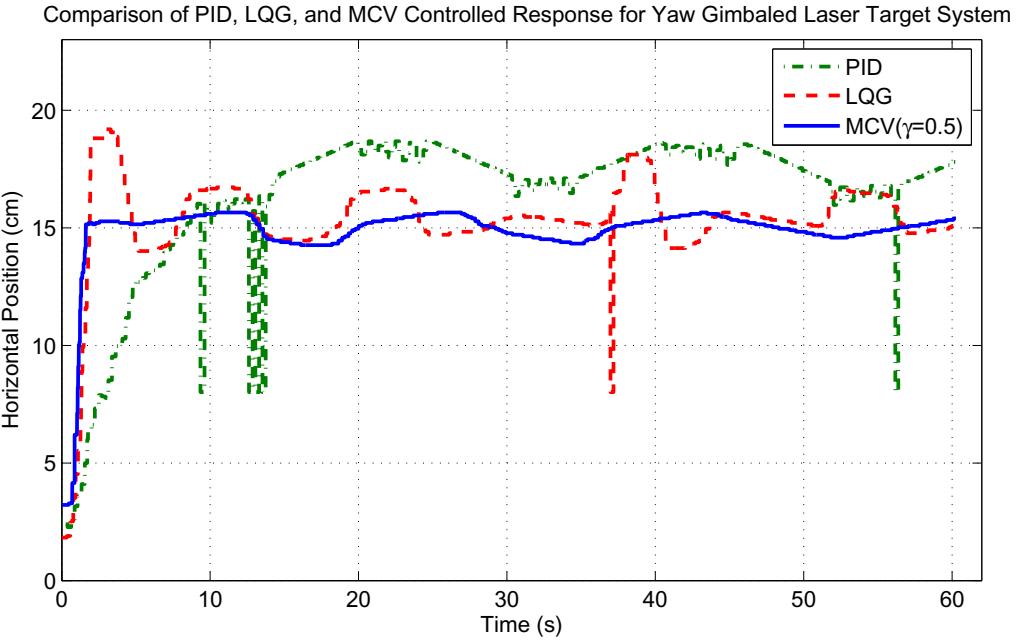


Figure 6.34: Controller comparison for PID, LQG, and MCV ($\gamma = 0.5$)

Fig. 6.34 shows the comparative plot for PID, LQG, and MCV ($\gamma = 0.5$). We see that PID controller shows the larger variation from the desired position. Also, we notice that PID controlled response shows some spikes of 8 cm value. When laser dot speed was high, sometimes the image processing block in Matlab detect a false pixel position. This occurs for a few sample times. It does not affect the controller performance much. It is interesting to notice that PID control suffers more than LQG and MCV for this. Table 6.3 shows the RMS pointing error and standard deviation of pointing error among PID, LQG, and MCV ($\gamma = 0.5$) controllers. We see that MCV performs better than PID and LQG. The ratio of RMS pointing error for PID, LQG, and MCV is 2.05 : 1.33 : 1.0. The ratio of standard deviation of pointing error for PID, LQG, and MCV is 1.94 : 1.3 : 1.0.

In conclusion, it is possible to find a MCV controller that provides the better standard deviation of pointing error performance than the LQG case. The MCV controller may or may not produce the smaller pointing error than the LQG case. The conclusion applies for the single axis gimbaled laser target system.

6.7.2 For Control Yaw and Pitch Axis Simultaneously

The objective of this experiment is to compare the pointing performances of different controllers for yaw and pitch axis under Gaussian white noise. We used a PID, LQG, and MCV ($\gamma = 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0$) controllers. The controllers for yaw and pitch axis are different. In this experiment, they work simultaneously. The two axis of the gimbal affect each others' motion. Therefore, this experiment will give more general idea than the single axis experiment about the controllers' performance.

We described the γ value selection rationale in the previous section. Also, the lighting condition was similar as the previous experiment. The linear positioner used for disturbance motion generation works for x axis (horizontal direction) only. For two-axis disturbance with the available positioner, we took the following approach.

We tilted one edge of the positioner with 10° degree angle with respect to the other edge. The positioner was run from one edge to another. Because of the tilting, there will be disturbance in y axis (vertical direction). The y axis motion was a component of x axis motion. The y axis disturbance magnitude is smaller than that of x axis in this approach.

Fig. 6.35 and 6.36 show x and y components of the disturbance signal used in the experiment. We generated the disturbance signal using the real time CPU clock. The signal was random, but it is not the same as the single axis experiment.

We initialized the laser dot close to $x = 2$ cm position. Because of the linear positioner motion, we could not keep the laser dot at exactly 2 cm position. We disabled the pitch axis amplifier to avoid coupling effect. The desired final position was $x = 15.15$ cm. The PID control parameters K_P , K_I , and K_D used were 0.24 , 3.60×10^{-6} , and 6.40×10^{-3} for pitch gimbal. For yaw gimbal, they were 0.12 , 3.18×10^{-6} , and 3.30×10^{-3} .

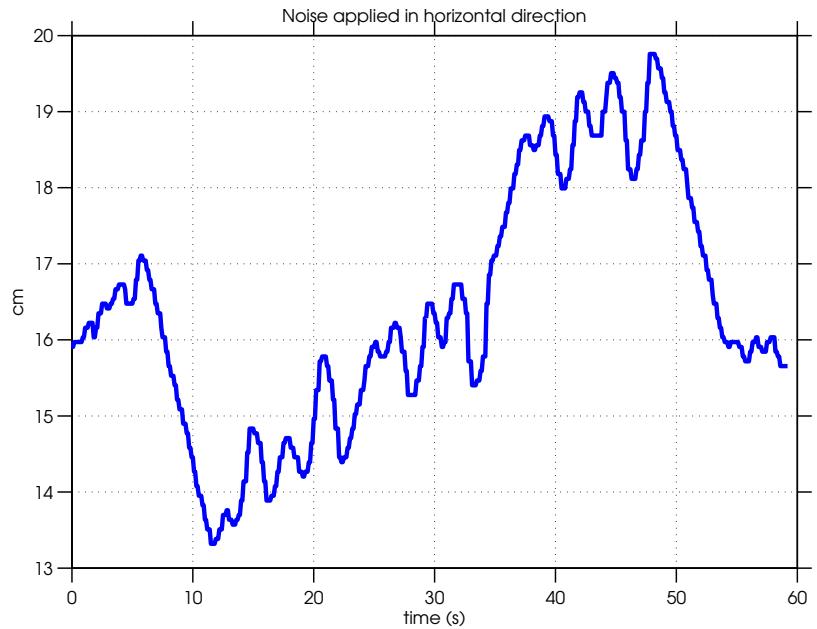


Figure 6.35: Disturbance for x axis

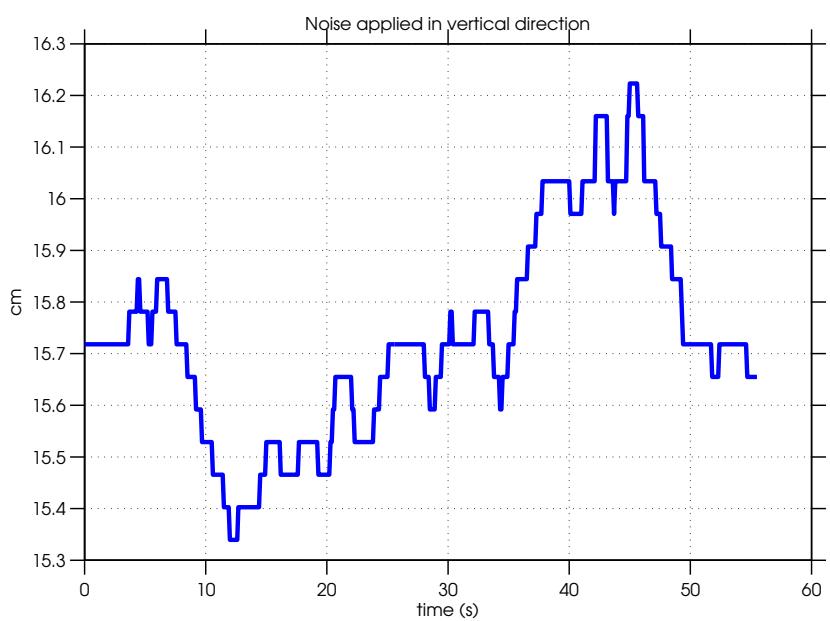


Figure 6.36: Disturbance for y axis

Table 6.4: RMS and Standard Deviation of pointing error for PID, LQG, and MCV controllers

	Horizontal		Vertical	
	RMS Pointing Error (cm)	SD of Pointing Error (cm)	RMS Pointing Error (cm)	SD of Pointing Error (cm)
PID	3.81	3.8	3.76	3.72
LQG	2.1	2.57	2.02	1.94
MCV ($\gamma = 0.01$)	2.59	2.57	2.15	2.11
MCV ($\gamma = 0.02$)	2.36	2.34	2.06	0.91
MCV ($\gamma = 0.05$)	2.43	2.41	2.41	2.02
MCV ($\gamma = 0.1$)	2.2	2.19	2.80	2.23
MCV ($\gamma = 0.2$)	2.29	2.29	3.26	1.55
MCV ($\gamma = 0.5$)	1.92	1.62	3.2	1.01
MCV ($\gamma = 1.0$)	2.03	1.92	7.98	0.91
MCV ($\gamma = 2.0$)	2.38	1.95	13	0.42

For LQG and MCV, the tuning parameters were,

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and $R = 1$. The experiment duration was approximately 65 seconds for all the controllers.

Next we calculated the RMS pointing errors and the standard deviations of the pointing error for horizontal and vertical directions. Table 6.4 shows the RMS pointing error and the standard deviation of the pointing error.

Fig. 6.37 and Fig. 6.38 show the horizontal RMS pointing error and the standard deviation of the horizontal pointing error. PID shows the highest RMS pointing error of 3.81 cm and the highest standard deviation of the pointing error of 3.80 cm. LQG controller reduces this

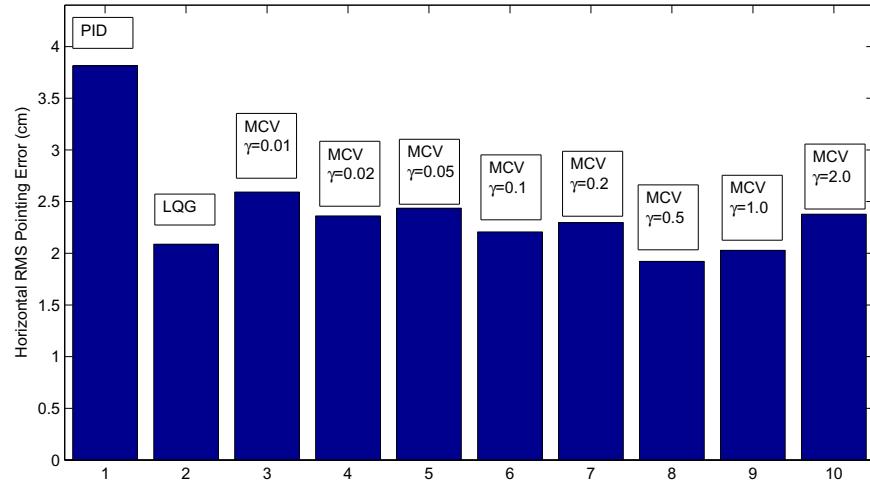


Figure 6.37: Horizontal RMS pointing error for PID, LQG, and MCV controllers

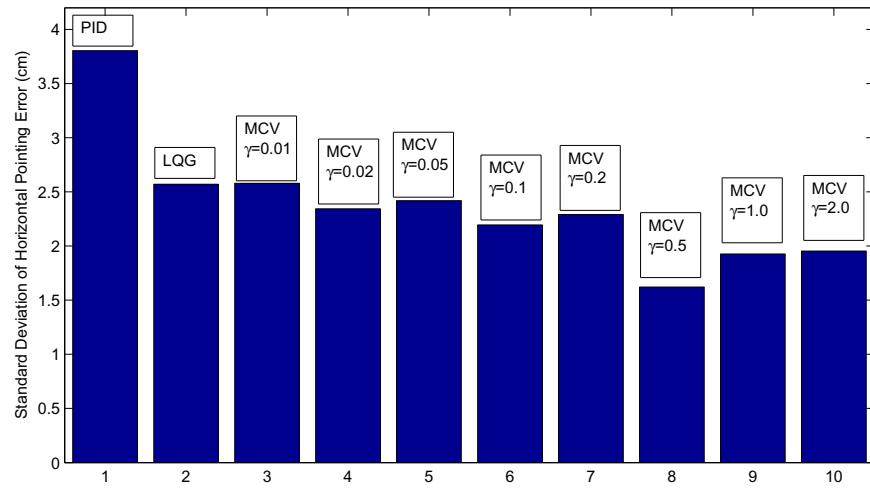


Figure 6.38: Standard Deviation of horizontal pointing error for PID, LQG, and MCV controllers

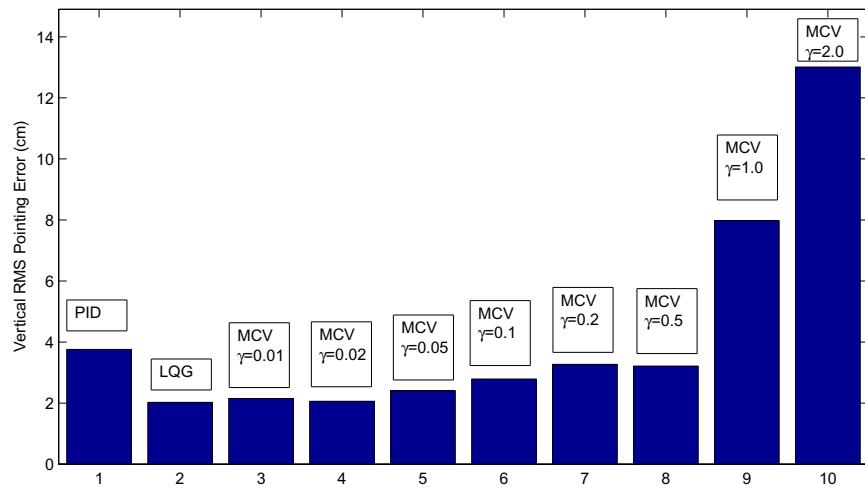


Figure 6.39: Vertical RMS pointing error for PID, LQG, and MCV controllers

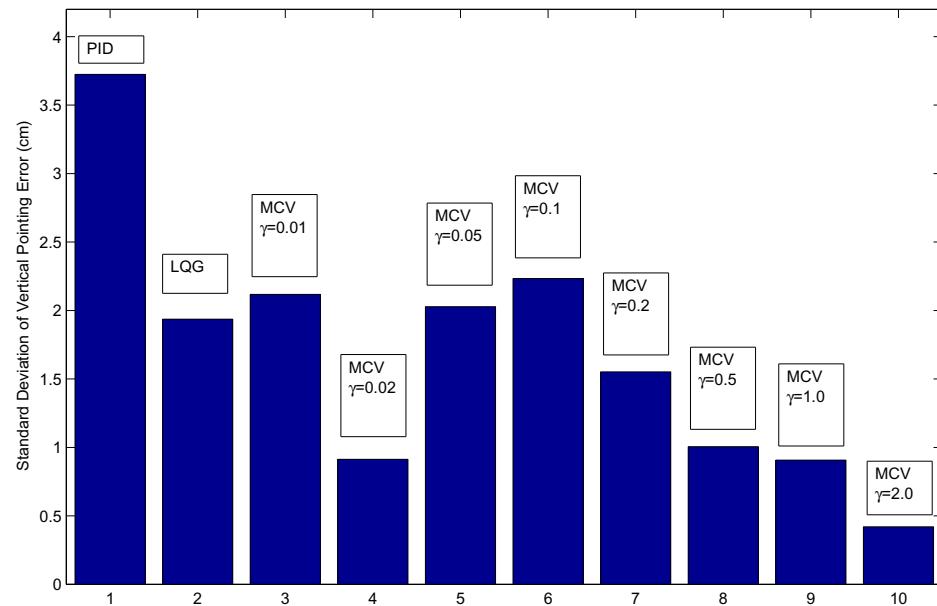


Figure 6.40: Standard Deviation of vertical pointing error for PID, LQG, and MCV controllers

RMS pointing error to 2.10 cm. It reduces the standard deviation of the pointing error to 2.57 cm. MCV ($\gamma = 0.01, 0.02, 0.05, 0.1, 2$) controllers give larger error than LQG controller. MCV ($\gamma = 0.5, 1$) controllers shows less RMS pointing error than the LQG case. MCV ($\gamma = 0.5$) shows the least pointing error of 1.92 cm. For standard deviation of the pointing error we see the similar pattern. The lowest standard deviation of the pointing error was 1.62 cm by MCV ($\gamma = 0.5$). Therefore, we select MCV for $\gamma = 0.5$ is the best MCV controller for yaw axis.

Fig. 6.39 and Fig. 6.40 show the vertical RMS pointing error and the standard deviation of the vertical pointing error. PID shows the RMS pointing error of 3.76 cm and the standard deviation of the pointing error of 3.72 cm. LQG controller reduces this RMS pointing error to 2.02 cm. It reduces the standard deviation of the pointing error to 1.94 cm. All the MCV controllers give larger error than LQG controller. Among them, MCV ($\gamma = 0.02$) shows a pointing pointing error of 2.06 cm. It gives the standard deviation of pointing error of 0.91 cm. MCV ($\gamma = 2$) gives the best standard deviation. However, it shows the largest pointing error of 13 cm. Therefore, we select MCV for $\gamma = 0.02$ is the best MCV controller for pitch axis.

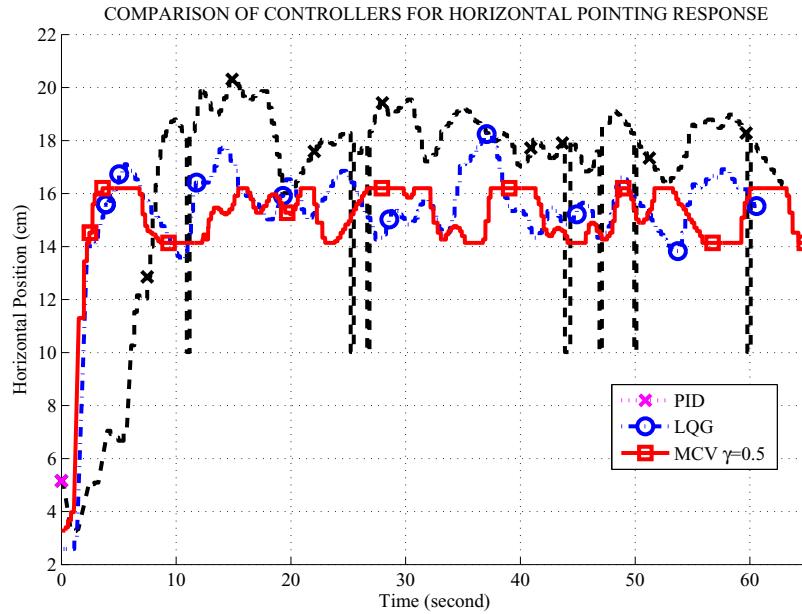


Figure 6.41: PID, LQG, and MCV controller comparison for horizontal pointing

From this experiment, we cannot conclude that the increasing gamma will increase the RMS

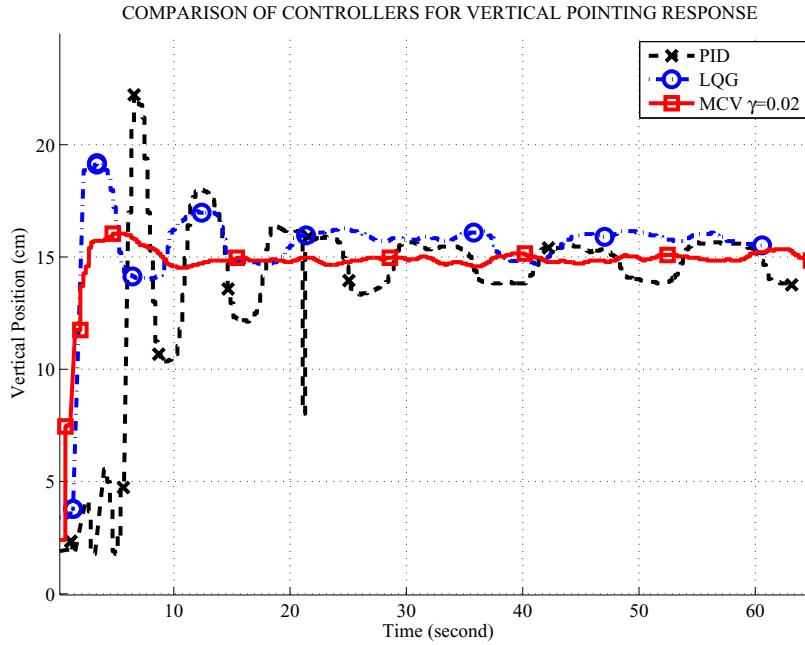


Figure 6.42: PID, LQG, and MCV controller comparison for vertical pointing

pointing error and decrease the standard deviation of pointing error monotonically. MCV ($\gamma = 0.5$) was the best among the MCV controllers for yaw axis and $\gamma = 0.02$ for pitch axis.

Table 6.5 shows the RMS pointing error and the standard deviation of pointing error among three controllers: PID, LQG, and MCV. Fig. 6.41 shows the comparative plot for PID, LQG, and MCV ($\gamma = 0.5$) for yaw axis. The ratio of RMS pointing error for PID, LQG, and MCV is 1.98 : 1.09 : 1.0. The ratio of standard deviation of pointing error for PID, LQG, and MCV is 2.34 : 1.59 : 1.0. Fig. 6.42 shows the comparative plot for PID, LQG, and MCV ($\gamma = 0.02$) for pitch axis. The ratio of RMS pointing error for PID, LQG, and MCV is 1.86 : 1 : 1.02. The ratio of standard deviation of pointing error for PID, LQG, and MCV is 4.09 : 2.13 : 1.0.

Compared to PID method, LQG shows 44.48% and 46.28% smaller mean pointing error for horizontal and vertical case. MCV shows 49.61% and 45.21% smaller mean pointing error for horizontal and vertical case. LQG shows 32.37% and 47.85% smaller standard deviation of pointing error for horizontal and vertical case. MCV shows 57.37% and 75.54% smaller standard deviation of pointing error for horizontal and vertical case.

Table 6.5: Pointing Performances of PID, LQG, and MCV controllers

	Horizontal		Vertical	
	RMS Pointing Error (cm)	SD of Pointing Error (cm)	RMS Pointing Error (cm)	SD of Pointing Error (cm)
PID	3.81	3.80	3.76	3.72
LQG	2.10	2.57	2.02	1.94
MCV	1.92	1.62	2.06	0.91

Compared to LQG, MCV method shows 8.57% larger and 1.98% smaller mean pointing error for horizontal and vertical case. MCV method shows 36.96% and 53.09% smaller standard deviation of pointing error for horizontal and vertical case.

In conclusion, MCV controller yields better pointing performance than PID and LQG controller on the basis of standard deviation of the pointing error. If we consider RMS pointing error, MCV ($\gamma = 0.5$) controller shows better performance than PID and LQG for the horizontal case.

In summary, we described a short overview of the gimabaled laser target system hardware. We compared the developed mathematical model and the hardware through open loop position control and PID control experiment. Then we implemented the PID, LQG, and MCV controllers in dSPACE hardware. Finally, we performed a single-axis experiment and a two-axis experiment for comparing the PID, LQG, and MCV controllers' pointing performances.

CHAPTER 7

CONCLUSIONS AND FUTURE WORKS

7.1 Conclusions

This thesis presented an approach to evaluate and compare the performances of different control methods for a laser target system. We developed linear models for the system while taking random disturbances into account, ran simulations with three different control methods, and implemented the controllers on an experimental test bench.

In this thesis, a non-linear mathematical model was developed for a two-axis gimbaled laser target system. The system uses computer vision to control the two-axis gimbal dynamics variables so that the laser always points to the center of the position sensor. The non-linear model was linearized around its hardware operating points. The disturbance was modeled as Gaussian white noise and added to the linearized model to obtain a stochastic linear model for the optimal control. As for verification of the model, we compared the position response from the simulation and experimental results of an open loop position control and a closed loop PID control schemes. We found that the simulated position responses followed the experimental responses closely.

Two statistical control methods were considered in this thesis – the LQG and the MCV control. Being the most commonly used controller in industry, the PID controller was used as the benchmark for the performance comparison. As for the performance metrics, we used the mean and the standard deviation of the pointing error. The controllers performances were evaluated and compared through the computer simulations first. Then all these controllers were implemented and tested in a dSPACE board in a real time environment. The LQG and MCV controllers were implemented for the output feedback case since all the states were not available for measurement. This is the first time that the MCV control method has been implemented for

a laser target system on hardware.

The simulation results showed that the statistical controllers performed better compared to the PID controller in terms of pointing error and error variation. Later, the experimental measurements also agreed with the simulation results.

The MCV controller performance varied depending on the values of the design parameter γ . The simulation results show a monotonic degradation of the mean pointing error with respect to the change of γ . From the hardware experiment we found that the trend was not always true. The experimental results showed that there are certain γ values such as 0.5 for yaw gimbal for which the MCV was a better option than the LQG. Therefore, the pointing error performance is not monotonic with respect to γ for the gimbaled laser target system.

In general, the MCV controllers improved the pointing error variation performance over the LQG controller. The MCV controller reduced the standard deviation of the pointing error compared to the LQG controller, both in simulations and experiments. However, the MCV controller also suffered from a slightly degraded mean pointing error.

In conclusion, the statistical controllers provide the designer with the flexibility of improving either the pointing error or error variation performance. This study shows the potential benefits of using statistical controllers in the gimbaled laser target system applications, subsequently, the space-based solar power transmission.

7.2 Future Works

The work presented in this thesis takes the initial step towards implementing and testing higher order cost cumulant control. A higher order cumulant such as third and fourth cumulant may improve the pointing performance. In order to improve the hardware model, system identification experiment can be done. This will estimate the parameter values of the hardware. Simulating the developed model with the estimated parameter values will be a validation of the model. Nonlinear statistical control can be developed in the future. This will enable us to control the developed nonlinear system directly. It would be interesting to compare the performance of controllers developed from the nonlinear and linear systems.

REFERENCES

- Anderson, B. D. O. and Moore, J. B. (1990). *Optimal control: linear quadratic methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Armstrong-Hélouvry, B. (1991). *Control of machines with friction*. Kluwer international series in engineering and computer science: Robotics. Kluwer Academic Publishers.
- Åström, K. and Hägglund, T. (2006). *Advanced Pid Control*. ISA-The Instrumentation, Systems, and Automation Society.
- Athans, M. (1971). The role and use of the stochastic linear-quadratic-gaussian problem in control system design. *Automatic Control, IEEE Transactions on*, 16(6):529 – 552.
- Bennett, S. (1993). *A History of Control Engineering: 1930-1955*. Iee Control Engineering Series. Peter Peregrinus.
- Canudas de Wit, C., Olsson, H., Astrom, K., and Lischinsky, P. (1995). A new model for control of systems with friction. *Automatic Control, IEEE Transactions on*, 40(3):419 –425.
- Chung, L., Reinhorn, A., and Soong, T. (1988). Experiments on active control of seismic structures. *Journal of Engineering Mechanics*, 114(2):241–256.
- Dietz, R., Center, G. C. M. S. F., Center, L. B. J. S., Aeronautics, U. S. N., Scientific, S. A., and Branch, T. I. (1981). *Satellite power system: concept development and evaluation program*. Number v. 3 in NASA reference publication. National Aeronautics and Space Administration, Scientific and Technical Information Branch.
- Dorato, P., Abdallah, C., and Cerone, V. (2000). *Linear Quadratic Control: An Introduction*. Krieger.

- Dyke, S., Spencer, B., Quast, P., Sain, M., Jr., D., and Soong, T. (1996). Acceleration feedback control of mdof structures. *Journal of Engineering Mechanics*, 122(9):907–918.
- Ekstrand, B. (2001). Equations of motion for a two-axes gimbal system. *IEEE Transactions on Aerospace and Electronic Systems*, 37(3):1083 –1091.
- Etkin, B. and Reid, L. (1996). *Dynamics of Flight Stability and Control*. Wiley.
- Glaser, P. (1968). Power from the sun: Its future. *Science*, 162:857–861.
- Goldstein, H. (1980). *Classical mechanics*. Addison-Wesley series in physics. Addison-Wesley Pub. Co.
- Gunckel, T. L., II, and Franklin, G. F. (1963). A general solution for linear, sampled-data control. *Journal of Basic Engineering*, 85(2):197–201.
- Haessig, D. A., Olsson, and Friedland, B. (1991). On the modeling and simulation of friction. *Journal of dynamic systems, measurement, and control*, 113(3):354.
- Hilkert, J. (2008). Inertially stabilized platform technology concepts and principles. *Control Systems, IEEE*, 28(1):26 –46.
- Hughes, P. (2012). *Spacecraft Attitude Dynamics*. Dover Publications, Incorporated.
- Hutchinson, S., Hager, G., and Corke, P. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12:651–670.
- Iwasaki, M., Shibata, T., and Matsui, N. (1999). Disturbance-observer-based nonlinear friction compensation in table drive system. *Mechatronics, IEEE/ASME Transactions on*, 4(1):3 –8.
- Kalman, R. (1960). Contributions to the theory of optimal control.
- Kennedy, P. and Kennedy, R. (2003). Direct versus indirect line of sight (los) stabilization. *Control Systems Technology, IEEE Transactions on*, 11(1):3 – 15.
- Krishnan, R. (2001). *Electric motor drives: modeling, analysis, and control*. Prentice Hall.

- Lee, H.-P. and Yoo, I.-E. (2008). Robust control design for a two-axis gimbaled stabilization system. In *Aerospace Conference, 2008 IEEE*, pages 1 –7.
- Liberty, S. R. and Hartwig, R. C. (1976). On the essential quadratic nature of lqg control-performance measure cumulants. *Information and Control*, 32(3):276 – 305.
- Maciejowski, J. and Vinnicombe, G. (2000). *Multivariable Feedback Design*. Pearson Education Canada.
- Masten, M. (2008). Inertially stabilized platforms for optical imaging systems. *Control Systems, IEEE*, 28(1):47 –64.
- Masten, M. K. (1985). Applications of control theory to design of line-of-sight stabilization systems. In *American Control Conference, 1985*, pages 1219–1222.
- Morin, A. (1833). New friction experiments carried out at Metz in 18311833. In *Proceedings of the French Royal Academy of Sciences*, 4:1–128.
- Nise, N. (2004). *Control Systems Engineering*. Wiley.
- Olsson, H., Astrom, K. J., de Wit, C. C., Gafvert, M., and Lischinsky, P. (1998). Friction models and friction compensation. *Eur. J. Control*, 4(3):176–195.
- Pham, K. (2004). *Statistical Control Paradigms for Structural Vibration Supression*. PhD thesis, University of Notre Dame, Notre Dame, Indiana.
- Pillay, P. and Krishnan, R. (1989). Modeling, simulation, and analysis of permanent-magnet motor drives. ii. the brushless dc motor drive. *Industry Applications, IEEE Transactions on*, 25(2):274 –279.
- Ramu, K. (2009). *Permanent Magnet Synchronous and Brushless DC Motors*. Mechanical Engineering Series. CRC Press/Taylor & Francis.
- Rosenbrock, H. and McMorran, P. (1971). Good, bad, or optimal? *Automatic Control, IEEE Transactions on*, 16(6):552 – 554.

- Rue, A. (1969). Stabilization of precision electrooptical pointing and tracking systems. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-5(5):805 –819.
- Rue, A. (1974). Precision stabilization systems. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-10(1):34 –42.
- Sain, M. (1965). *On Minimal-Variance Control of Linear Systems with Quadratic Loss*. PhD thesis, University of Illinois, Urbana, Illinois.
- Sain, M. and Liberty, S. (1971). Performance-measure densities for a class of lqg control systems. *Automatic Control, IEEE Transactions on*, 16(5):431 – 439.
- Stribeck, R. (1902). Die wesentlichen Eigenschaften der Gleit- und Rollenlager The key qualities of sliding and roller bearings. *Zeitschrift des Vereines Deutscher Ingenieure*, 46:1342–1348.
- Sung, S., Lee, J., and Lee, I. (2009). *Process Identification and PID Control*. Wiley.
- Wang, H. and Williams, T. (2008). Strategic inertial navigation systems - high-accuracy inertially stabilized platforms for hostile environments. *Control Systems, IEEE*, 28(1):65 –85.
- Won, C. (1994). *Cost Cumulants In Risk-Sensitive and Minimum Cost Variance Control*. PhD thesis, Department of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana.
- Won, C., Schrader, C., Michel, A., and Sain, M. (2008). *Advances in Statistical Control, Algebraic Systems Theory, and Dynamic Systems Characteristics: A Tribute to Michael K. Sain*. Systems & control. Birkhäuser.
- Won, C.-H. (1999). Comparative study of various control methods for attitude control of a {LEO} satellite. *Aerospace Science and Technology*, 3(5):323 – 333.

Won, C.-H. and Gunaratne, K. (2002). Performance study of lqg, mcv, and risk-sensitive control methods for satellite structure control. In *American Control Conference*, volume 3, pages 2481 – 2486.

Won, C.-H., Sain, M., and Liberty, S. (2001). Full-state-feedback minimal cost variance control on an infinite time horizon: the risk-sensitive approach. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 1, pages 819–824 vol.1.

Xue, D., Chen, Y., and Atherton, D. (2007). *Linear Feedback Control: Analysis and Design with MATLAB*. Advances in Design and Control. Society for Industrial and Applied Mathematics.