# DEVELOPMENT OF A POSITION AND TRAJECTORY TRACKING CONTROL OF BALL AND PLATE SYSTEM USING A DOUBLE FEEDBACK LOOP STRUCTURE

**By**

**Abubakar UMAR**

**Department of Electrical and Computer Engineering**
**Faculty of Engineering**
**Ahmadu Bello University Zaria, Nigeria**

**APRIL, 2017**

# DEVELOPMENT OF A POSITION AND TRAJECTORY TRACKING CONTROL OF BALL AND PLATE SYSTEM USING A DOUBLE FEEDBACK LOOP STRUCTURE

**By**

**Abubakar, UMAR, B.Eng. (ABU, 2011)**
**M.Sc/ENG/38097/2012-2013**
**abubakaru061010@gmail.com**

**A DISSERTATION SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES, AHMADU BELLO UNIVERSITY, ZARIA**

**IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF A MASTER OF SCIENCE (M.Sc) DEGREE IN CONTROL ENGINEERING**

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

**FACULTY OF ENGINEERING**

**AHMADU BELLO UNIVERSITY, ZARIA**

**NIGERIA**

**April, 2017**

## DECLARATION

I Abubakar UMAR hereby declare that the work in this dissertation entitled “Development of a Position and Trajectory Tracking Control of Ball and Plate System using a Double Feedback Loop Structure” has been carried out by me in the Department of Electrical and Computer Engineering. The information derived from literature has been duly acknowledged in the text and a list of references provided. No part of this dissertation was previously presented for another degree or diploma at this or any other institution

Abubakar UMAR (Student) _____ Signature _____ Date

## CERTIFICATION

This Dissertation entitled **DEVELOPMENT OF A POSITION AND TRAJECTORY TRACKING CONTROL OF BALL AND PLATE SYSTEM USING A DOUBLE FEEDBACK LOOP STRUCTURE** by Abubakar UMAR meets the regulations governing the award of the degree of Master of Science (MSc) in Control Engineering of the Ahmadu Bello University, and is approved for its contribution to knowledge and literary presentation.

| | | |
|---|---|---|
| Professor M. B. Mu'azu<br>(Chairman, Supervisory Committee) | Signature | Date |
| Dr. A. D. Usman<br>(Member, Supervisory Committee) | Signature | Date |
| Dr. Yusuf Jibril<br>(Head of Department) | Signature | Date |
| Professor S.Z Abubakar<br>(Dean, School of Postgraduate Studies) | Signature | Date |

## DEDICATION

This dissertation is dedicated to Almighty Allah; the Most Beneficient, the Most Gracious and the Most Merciful. Also, to my parents, Malam Jibrin Umar and Malama Amina Umar and my Siblings.

## ACKNOWLEDGEMENT

I am indeed grateful to Almighty Allah for His infinite Blessings and Guidance towards the successful completion of this work.

I wish to express my utmost gratitude to my supervisor, role model and the chairman of my supervisory committee, Prof. M. B. Mu'azu, for his time, immerse contributions and valuable guidance towards the success of this work. Indeed, the completion of this work could not have been possible without your consistent participation and assistance. I am proud to have you as my supervisor. You helped me to shape the research problem and provided valuable insight on the solution to the problem. Thank you very much Prof. My thanks also goes to my co-supervisor Dr. A.D. Usman for his valuable input and constant encouragement throughout the stages of the work. I really appreciate the training I received from you. My deep appreciation goes to the students of the Control & Computer Research Group for their valuable contributions, suggestions and constructive criticisms during the discussion stages of the work.

I acknowledge and appreciate the contribution of all the lecturers of Electrical and Computer Engineering, Ahmadu Bello University, namely: Prof. B. G. Bajoga, Prof. U.O Aliyu, Prof. B. Jimoh, Dr. A. M. S. Tekanyi, Dr. S. M. Sani, Dr. S. Man-Yahaya, Dr. I. J. Umoh and Dr. S. Garba, Dr. Y. Jibril, Dr. K.A. Abu-Bilal, Engr. M. J. Musa, Engr. A. I. Abdullahi, Engr. E. A. Gbenga, and most especially, those whose names could not be mentioned. My sincere appreciation also goes to malam Abdullahi Tukur for his administrative support towards the completion of the work.

My special thanks go to Suleiman Hussein, Salawudeen A. Tijjani, Ovie Ese, Umar Musa, Olaniyan Abdulrahman, Zaharudeen Haruna, Engr. S.M Aminu, Ajayi Ore-Ofe, Gideon Atuman Joel and Shittu Danraka for their valuable contributions and support towards the success of this work. I am indeed grateful to you guys, your handsome reward is with God. I

am very much thankful to all my course mates, especially Nwajiaku Ndubisi, Busayo Adebisi and Abdulmumin Yesufu for your encouragement and persistence. God will reward you all.

Above all, I am really indebted to my parents Malam Jibrin Umar and Malama Amina Umar for your continuous love and support. Your kind advice and understanding will always be appreciated. Thank you very much. My gratitude also goes to my siblings, Malam Muhammad Umar, Engr. Abdullahi Umar, Malama Rakiya Umar Mundir and Malama Ramatu Umar whose love, caring and prayers have kept me strong and sound all through my life.

**Abubakar Umar**
**April, 2017**

## ABSTRACT

This research work presents the development of a position and trajectory tracking control of ball and plate system. The ball and plate control system was considered as a double feedback loop structure (a loop within a loop), for effective control of the system. The inner loop was designed using linear algebraic method by solving a set of Diophantine equations. The outer loop was designed using H-infinity sensitivity approach. A virtual reality model of the ball and plate system using the virtual reality modelling language (VRML) and graphical user interface (GUI) based simulation model of the system were developed in MATLAB 2013a. The results of the simulation of the system showed that the plate was stabilized at 0.3546 seconds and the ball was able to settle at 1.7087 seconds. The trajectory tracking error of the system using the H-infinity controller was 0.0095 m. The improvements in terms of trajectory tracking error and settling time of the system when compared with the single loop H-infinity (SLH) controller are 71.8% and 60.5% respectively. The improvements when compared with the double loop structure using fuzzy sliding mode controller are 52.5% and 51.2% in terms of the trajectory tracking error and settling time respectively.

## TABLE OF CONTENTS

**TITLE PAGE**

DECLARATION i

CERTIFICATION ii

DEDICATION iii

ACKNOWLEDGEMENT iv

ABSTRACT vi

TABLE OF CONTENTS vii

LIST OF FIGURES xi

LIST OF TABLES xiii

LIST OF APPENDICES xiv

LIST OF ABBREVIATION xv

**CHAPTER ONE: INTRODUCTION**

**1.1 Background 1**

**1.2 Significance of Research 2**

**1.3 Problem Statement 3**

**1.4 Aim and Objectives 4**

**CHAPTER TWO: LITERATURE REVIEW**

**2.1 Introduction 5**

**2.2 Review of Fundamental Concepts 5**

**2.2.1 Ball and Plate System 5**

*2.2.1.1 Control system design* *11*

**2.2.2 Nonlinear Systems** **12**

**2.2.3 Controllability and Observability** **13**

*2.2.3.1 Stability* *14*

*2.2.3.2 Trajectory and motion tracking* *16*

*2.2.3.3 Path following* *17*

**2.2.4 Types of Controllers** **18**

*2.2.4.1* $H_\infty$ *controller* *19*

*2.2.4.2* $H_\infty$ *Mixed sensitivity problem* *22*

**2.2.5 Linear Algebraic Method** **23**

*2.2.5.1 Transient and steady-state requirements* *25*

*2.2.5.2 Implementation by two-parameter configuration* *27*

*2.2.5.3 Actuator parameters* *31*

*2.2.5.4 Inner loop design* *33*

**2.2.6 Virtual Reality Modelling Language (VRML) as a 3-D Modelling Tool** **35**

**2.2.7 Simulink® 3D Animation** **37**

**2.2.8 Graphical User Interface (GUI)** **38**

**2.3 Review of Similar Works** **39**

**CHAPTER THREE: MATERIALS AND METHODS**

**3.1 Introduction** **55**

**3.2 Methodology** **55**

**3.2.1 Ball and Plate System Modelling** **56**

**3.2.2 Decomposition of the Ball and Plate System** **56**

**3.2.3 Linearization of the Ball and Plate System** **57**

**3.2.4 Controllability and Observability Test for the Ball and Plate System** **58**

**3.3 Selection of the Actuator Parameters** **59**

**3.3.1 Two-Port Parameter Configuration** **63**

**3.4 Determination of the $H_\infty$ Controller** **64**

**3.5 Development of the Virtual Reality (VR) Model of the Ball and Plate System** **65**

**3.6 Development of the Simulation Environment in MATLAB Simulink** **66**

**3.6.1 Development of the Inner Loop Controller** **66**

**3.6.2 Development of the Outer Loop Controller** **67**

**3.6.3 Development of the Ball Dynamics of the Ball and Plate System** **68**

**3.6.4 Development of the Reference Signal for the Trajectory Tracking** **70**

**3.7 Graphical User Interface (GUI) of the Ball and Plate System** **70**

**3.8 Performance Evaluation** **71**

**3.8.1 Trajectory Tracking Error** **71**

**3.8.2 Transient Response** **71**

**3.9 Comparison of Results** **71**

**CHAPTER FOUR: RESULTS AND DISCUSSION**

**4.1 Introduction** **72**

**4.2 Result of the Controllability and Observability Test on the System** **72**

**4.3 Result of the Actuator Parameter** **72**

**4.4 Result of Two-Port Parameter Configuration** **73**

**4.5 Result of the $H_\infty$ Controller** **75**

**4.6 Result of the Virtual Reality (VR) Model** **76**

**4.7 Result of the Trajectory Tracking of the Ball and Plate System** **78**

**4.8 Result of the Graphical User Interface (GUI) for the Circular Trajectory Tracking** **79**

**4.9 Result of the Circular Trajectory Tracking Using H-infinity Controller Considering the Ball and Plate System as a Single Loop System** **79**

**4.10 Comparison of the Results** **80**

**4.10.1 Comparison of the controllers based on the Step Response Performance Index** **81**

**4.10.2 Comparison of the Developed Controller with that of Negash and Singh (2015)** **82**

**CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS**

**5.1 Conclusion** **83**

**5.2 Limitation** **83**

**5.3 Significant Contributions** **83**

**5.4 Recommendations for Further work** **84**

**REFERENCES** **85**

## LIST OF FIGURES

Figure 2.1: The Ball and Plate System by HUMUSOFT 6

Figure 2.2: Schematic Diagram of the Ball and Plate System 8

Figure 2.3 Control Scheme 11

Figure 2.4: Root Locus Plot of System Indicating Region of Stability 14

Figure 2.5: Mixed Sensitivity Functions 22

Figure 2.6: Unity Feedback System 27

Figure 2.7: Two-Port Parameter Configuration 29

Figure 2.8: Model of a Separately Excited DC Motor 31

Figure 2.9: Transfer Function Block Diagram between the Load Angle and Armature Voltage 33

Figure 2.10: Developed Double Feedback Loop Structure of the Control Scheme 34

Figure 2.11: Implementation of the Two-Port Parameter Configuration of the Inner Loo Loop 34

Figure 2.12: Hierarchical Tree Structure of Nodes 35

Figure 2.13: VRML Coordinate System 36

Figure 2.14: VRML Rotation Angles 37

Figure 3.1: Controllability and Observability Test Code 59

Figure 3.2: Screenshot of the Virtual Reality Model 66

Figure 3.3: Simulink Model of the Two-Port Parameter 67

Figure 3.4: Simulink model of the Outer loop 67

Figure 3.5: Simulink Model of the Ball Dynamics of the Ball and Plate System 68

Figure 3.6: Complete Simulink Model of the Ball and Plate System 69

Figure 3.7: Snippet of the Graphical User Interface Code 70

Figure 4.1: Step Response of the Actuator 73

Figure 4.2: Step Response of the Actuator with Open Parameters 74

Figure 4.3: Step Response of the H-infinity Controller 76

Figure 4.4: Virtual Reality (VR) Model of the Ball and Plate System 77

Figure 4.5: Circular Trajectory Tracking Performance with H-infinity Controller 78

Figure 4.6: Circular Trajectory Tracking of the Ball using H-infinity Controller 79

Figure 4.7: Circular Trajectory Tracking of the Ball Considering the System as a Single Controller 80

## LIST OF TABLES

Table 2.1: DC Motor Variables and Parameters 32

Table 3.1: Parameters of the Ball and Plate System 59

Table 3.2: Actuator Parameter Types 61

Table 3.3 Some Parameters of Actuator RFS-20-6012 62

Table 3.4: Iteration Results of the H-infinity Controller 65

Table 4.1: Properties of the Actuator 73

Table 4.2: Properties of the Actuator with Open Parameters 74

Table 4.3: Properties of the H-infinity Controller 76

Table 4.4: Comparison of the developed controller and that of Ghiasi and Jafari (2012) 81

Table 4.5: Comparison of the developed controller and that of Negash and Singh (N&S) (2015) 82

## LIST OF APPENDICES

**APPENDIX A1**

MATLAB CODE OF THE ACTUATOR 92

**APPENDIX A2**

MATLAB CODE OF THE ACTUATOR WITH OPEN PARAMETERS 92

**APPENDIX A3**

MATLAB CODE OF THE H-INFINITY FUNCTION 92

**APPENDIX B1**

VIRTUAL REALITY (VR) PARAMETER WINDOW OF THE BALL AND PLATE SYSTEM 93

**APPENDIX C1**

MATLAB CODE OF THE GRAPHICAL USER INTERFACE (GUI) OF THE CIRCULAR TRAJECTORY TRACKING USING H-INFINITY CONTROLLER 94

## LIST OF ABBREVIATIONS

| Acronyms | Definition |
|---|---|
| 3D | Three Dimension |
| AI | Artificial Intelligence |
| ARE | Algebraic Ricatti Equation |
| BB | Ball and Beam |
| BP | Back Propagation |
| BPS | Ball and Plate System |
| CCD | Charged Coupled Device |
| CMAC | Cerebeliar Model Articulation Controller |
| DC Motor | Direct Current Motor |
| DOF | Degree of Freedom |
| EFLC | Enforced Fuzzy Logic Controller |
| FA | Firefly Algorithm |
| FLC | Fuzzy Logic Controller |
| FRBF | Fuzzy Online Radius Basic Function |
| FNNC | Fuzzy Neural Network Controller |
| FOPD | Fractional Order Proportional Derivative |
| GUI | Graphical User Interface |
| GUIDE | Graphical User Interface Development Environment |
| HFCMAC | Hierarchical Fuzzy Cerebeliar Model Articulation Controller |
| ITAE | Integral of Time Multiplied by Absolute Error |
| KYP | Kalman-Yakubovich-Popov |
| LHP | Left Hand Plane |
| LFT | linear Fractional Transformation |
| LQ | Linear Quadratic |
| LQG | Linear Quadratic Gaussian |
| LQR | Linear Quadratic Regulator |
| LQT | Linear Quadratic Tracking |
| PD | Proportional-Differential |

| | |
|---|---|
| PID | Proportional-Integral-Differential |
| PSO | Particle Swarm Optimization |
| PSO-FNNC | Particle Swarm Optimization algorithm Fuzzy Neural Network Controller |
| MATLAB | Matrix Laboratory |
| M-FILE | MATLAB File |
| MIMO | Multiple-Input-Multiple-Output |
| MPC | Model Predictive Control |
| MS | Magnetic Suspension |
| NARMA | Nonlinear Auto Regressive Moving Average Model |
| NRCF | Normalized Right Coprime Factor |
| RBF | Radius Basic Function |
| RHP | Right Hand Plane |
| RRS | Revolute-Revolute-Spherical |
| SISO | Single-Input-Single-Output |
| TITO | Two-Input-Two-Output |
| TS | Takagi Sugeno |
| UICONTROL | User Interface Controls |
| UIMENUS | User Interface Menus |
| VR | Virtual Reality Model |
| VRML | Virtual Reality Modelling Language |
| ZN | Ziegler Nichols |

# CHAPTER ONE

# INTRODUCTION

## 1.1 Background

Balancing systems are one of the most popular and challenging test platforms for control engineers. Such systems are like the traditional cart-pole system (inverted pendulum), the ball and beam system, double and multiple inverted pendulums (Mohajerin *et al.,* 2010). The ball and plate system is a generalization of the famous ball and beam benchmark system. The latter is a two degree of freedom (DOF) system consisting of a ball that can roll on a rigid beam, while the former is a four DOF system consisting of a ball that can roll freely on a rigid plate (Moarref *et al.,* 2008). However, it is more complicated than the ball and beam system due to its coupling of multi-variables. This under-actuated system has only two actuators and is stabilized by just two control inputs (Ghiasi & Jafari, 2012). Since the movement of the ball over the plate can reach high speeds, the design of a suitable controller for this system is a major challenge; therefore, these systems are not commonly used in laboratories (Galvan-Colmenares *et al.,* 2014). The system consists of a plate pivoted at its centre such that the slope of the plate can be manipulated in two perpendicular directions (Dong *et al.,* 2011).

A servo system consists of motor controller card and two servo motors to tilt the plate. Intelligent vision system is used for measurement of a ball position from a CCD camera. The problem of the motion control of this system is to control the position of a ball on a plate for both static positions and desired paths. The slope of the plate can be manipulated in two perpendicular directions, so that the tilting of the plate will make the ball move on the plate (Dong *et al.,* 2011).

The ball and plate system finds application in areas like humanoid robot, satellite control, rocket system and unmanned aerial vehicle (UAV) (Mukherjee *et al.,* 2002) in

the fields of path planning, trajectory tracking and friction compensation (Oriolo & Vendittelli, 2005).

Various control methods have been introduced in the recent years for the ball and plate system. A controller design for two dimensional electro-mechanical ball and plate system based on the classical and modern control theory was proposed by (Knuplež *et al.,* 2003). A supervisory fuzzy controller for studying motion control of the system included the set-point problem and the tracking problem along desired trajectory which was composed of two layers as described by (Bai *et al.,* 2006). A nonlinear velocity observer for output regulation of the ball and plate system where the ball velocities were estimated by state observer was proposed by (Wang *et al.,* 2008). In the work of (Hongrui *et al.,* 2008), the position of the ball was regulated with a double feedback loop system, in which recursive back-stepping design was employed for the external loop, while switching control scheme was employed in the inner feedback loop. Also, proportional-integral-differential neural network controller based on genetic algorithm is another piece of work done on ball and plate system by (Dong *et al.,* 2009).

However, previous research works considered the ball and plate as a single loop structure. For an effective control of the ball and plate system, a double feedback loop structure, that is, a loop within a loop is considered (Liu & Liang, 2010).

### 1.2 Significance of Research

The ball and plate system is one of the most popular and important models in control education, which is used at undergraduate and post graduate studies in teaching and testing of control algorithms and also it is a benchmark nonlinear plant, this is because it is more complex than the traditional ball and beam system due to the coupling between the variables. The ball is able to move freely and has no ability to recognise the environment, as a result, the ball cannot control its behaviour by itself. However, the

control problems of the system consist of the ball's position control, trajectory tracking and obstacle avoidance. The ball's position control problem is to make the ball arrive at a point accurately and as soon as possible, the trajectory tracking problem is to make the ball follow some defined path accurately and at high speed, and the obstacle avoidance control problem is to find an optimum path for the ball in a complicated environment based on some certain criteria. All these problems are good benchmarks to proof-test the capability of the different control schemes. As a result, designing a suitable controller that will solve these problems is a major challenge.

In this research, the study of the first two problems, which is position and trajectory tracking has been carried out, considering the system as a double loop structure.

### 1.3 Problem Statement

The ball and plate system apparatus is a two dimensional electromechanical device which can be considered as a nonlinear, multivariable (with two inputs and two outputs) and unstable system. The system is under-actuated as it possesses more degrees of freedom than the number of available actuators. For an effective control of the ball and plate system, a double feedback loop structure, that is, a loop within a loop is considered. However, due to the existence of uncertainties due to friction, parameter uncertainties, measurement time delays, practical applications requires nonlinear control methods that will be adopted in the design of the inner and outer loops. In order to fulfil these requirements, the inner loop is designed as an actuator (angular) position controller for the plate inclination, while the outer loop is designed so as to control the balls (linear) position on the plate.

This research is aimed at designing the inner loop of the ball and plate system based on linear algebraic method. An overall transfer function is chosen that minimizes the integral of time multiplied by absolute error (ITAE), and a two parameter configuration

is used for the implementation of the compensators that will be obtained from the solution of a Diophantine equation. A robust controller technique, based on H-infinity control technique is implemented in the outer control loop of the ball and plate system, due to the problem of model parameter uncertainties and external disturbances, such as friction between the ball and the plate and parameter variations on the ball, it is necessary to design the outer loop with a controller that will take care of the above problem.

### 1.4 Aim and Objectives

The aim of this research is the development of a position and trajectory tracking control scheme for the ball and plate system using a double feedback loop structure.

The objectives of the research are as follows:

i) To design the inner loop using linear algebraic method and the outer loop using H-infinity sensitivity function for a more effective control of the ball and plate system.

ii) To develop a virtual reality model of the ball and plate system using the virtual reality modelling language (VRML) and its graphical user interface (GUI) based simulation model using MATLAB 2013a.

iii) Validation of the performance of the developed model by comparison with the work of Ghiasi and Jafari (2012) and Negash and Singh (2015) based settling time, trajectory tracking error and maximum overshoot as performance metrics.

# CHAPTER TWO
# LITERATURE REVIEW

## 2.1 Introduction

The literature review comprises an overview of fundamental concept and the review of similar works. In the review of fundamental concepts, most of the pertinent works and the fundamental theories that have been used for the success of this research are reviewed, after which similar works are reviewed.

## 2.2 Review of Fundamental Concepts

In this section, concepts that are fundamental and pertinent to understanding ball and plate control system are reviewed. These concepts includes the theory of the ball and plate system, nonlinear system, stability, position and trajectory tracking control, etc. there is also a review of various control schemes.

### 2.2.1 Ball and Plate System

The ball and plate system, a two-dimensional extension of the ball and beam system, which has strong coupling and nonlinear dynamics. It is highly unstable plant with apparent friction effects, which is popularly used as a benchmark to test the effectiveness of specific control algorithms, especially in the field of friction compensation. The plate with two rotating axes is the control target. A digital camera is placed above the plate to measure the relative position of a copper ball with regard to the center of the plate. The control objective, by rotating the plate, is to stabilize the ball on a specific position or to make it go along a certain trajectory (Wang *et al.*, 2014). Stabilization control of the ball and plate is to hold the ball in a specific position on the plate. Trajectory tracking control demands the ball follow the given position reference (Wang *et al.,* 2007).

Static and dynamic position tracking control could be studied simultaneously on the ball and plate system. Briefly, the control procedure is described as follows (Yuan, 2009):

i) The position sensor obtains the position of the ball on the plate. In most applications, the sensor is either a camera or a touch-screen device.

ii) Position data is fed back to the control system where the reference angle is calculated either for position command or trajectory tracking problems. If a camera is used as the sensor, the position data is obtained by image processing techniques that identify the target object from its background and determine its position from the analogue frame of picture.

iii) The rotation of the plate along the two mutually orthogonal directions are facilitated by two actuators and thus, balancing the ball to the command position or along the desired trajectory. The actuators often used for this application are either DC-motors or stepper motors and in only one instance a pneumatic device is reported.

Figure 2.1 shows a typical laboratory version of the ball and plate system by HUMUSOFT (Andinet, 2011).



Figure 2.1: The Ball and Plate System by HUMUSOFT (Andinet, 2011)

The nonlinear model of the ball and plate system presented in equations (2.1) to (2.4) as described in (Andinet, 2011):

$$\left(m_b + \frac{J_b}{R_b^{\,2}}\right)\ddot{x} - m_b x\left(\dot{\theta}_x\right)^2 - m_b y\dot{\theta}_x\dot{\theta}_y + m_b g\sin\theta_x = 0 \quad (2.1)$$

$$\left(m_b + \frac{J_b}{R_b^{\,2}}\right)\ddot{y} - m_b y\left(\dot{\theta}_y\right)^2 - m_b x\dot{\theta}_x\dot{\theta}_y + m_b g\sin\theta_y = 0 \quad (2.2)$$

$$\left(m_b x^2 + J_b + J_{Px}\right)\ddot{\theta}_x + 2m_b x\dot{x}\dot{\theta}_x + m_b xy\ddot{\theta}_y + m_b\left(\dot{x}y + x\dot{y}\right)\dot{\theta}_y + m_b gx\cos\theta_x = \tau_x \quad (2.3)$$

$$\left(m_b y^2 + J_b + J_{Py}\right)\ddot{\theta}_y + 2m_b y\dot{y}\dot{\theta}_y + m_b xy\ddot{\theta}_x + m_b\left(\dot{x}y + x\dot{y}\right)\dot{\theta}_x + m_b gy\cos\theta_y = \tau_y \quad (2.4)$$

In equations (2.1) to (2.4) , $m_b$ (kg) is the mass of the ball; $J_b$ $\left(kgm^2\right)$ is the rotational moment of inertia of the ball; $J_{P_{x,y}}\left(kgm^2\right)$ is the rotational moment of inertia of the plate; $R_b\left(m\right)$ is the radius of the ball; $x\left(m\right)$ is the ball position along X-axis; $y\left(m\right)$ is the ball position along Y-axis; $\dot{x}\left(m/s\right)$ and $\ddot{x}\left(m/s^2\right)$ are the velocity and acceleration respectively along X-axis; $\dot{y}\left(m/s\right)$ and $\ddot{y}\left(m/s^2\right)$ are the velocity and acceleration respectively along Y-axis; $\theta_x\left(rad\right)$ is the plate deflection angle about X-axis; $\dot{\theta}_x\left(rad/\sec\right)$ is the plate deflection angular velocity about X-axis; $\theta_y\left(rad\right)$ is the plate deflection angle about Y-axis; $\dot{\theta}_y\left(rad/\sec\right)$ is the deflection angular velocity about Y-axis; $\tau_x\left(Nm\right)$ is the torque exerted on the plate in X-axis direction and $\tau_y\left(Nm\right)$ is the torque exerted on the plate in Y-axis direction.

Equations (2.1) and (2.2) describe the movement of the ball on the plate, they show how the effects of the ball acceleration rely on the plate deflection angle, and plate deflection angular velocity. Equations (2.3) and (2.4) show how the dynamics of the plate deflection rely on external driving forces and ball position (Duan *et al.,* 2009).

To simplify the mathematical modelling of the ball and plate system, the following assumptions are made (HUMUSOFT Ltd, 2012):

i) Ball-plate contact is not lost under any circumstances.

ii) No sliding of the ball on the plate is allowed.

iii) All friction forces and torques are neglected.

iv) Plate angles and area limitations are not considered.

Based on the above mentioned assumptions, Figure 2.2 shows the rigid body model for the ball and plate system.

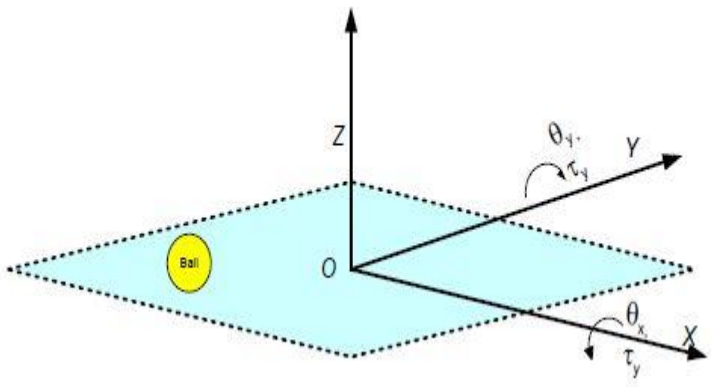



Figure 2.2: Schematic Diagram of the Ball and Plate System (Andinet, 2011)

For mathematical analysis, the ball and plate system can be simplified into a particle system made by two rigid bodies. The plate has three geometry limits in the translation along the X-axis, Y-axis and Z-axis. It also has a geometry limit in rotation about the Z-axis. The plate has two degree of freedom (DOF) in the rotation about the X-axis and Y-axis. The ball has a geometry limit in the translation along the Z-axis, and it has two degree of freedom (DOF) in translation along X-axis and Y-axis. The model has four degree of freedom (DOF) where the generalized coordinates are chosen as: $q_1 = x$, $q_2 = y$, $q_3 = \theta_x$, and $q_4 = \theta_y$ (Hongrui *et al.*, 2008).

Consider the state variable assignment (Fan *et al.*, 2004)

$$X = \left(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\right)^T = \left(x, \dot{x}, \theta_x, \dot{\theta}_x, y, \dot{y}, \theta_y, \dot{\theta}_y\right)^T \tag{2.5}$$

From equations (2.3) and (2.4), since the mass and the moment of inertia of the ball are negligible compared to the moment of inertia of the plate, then,

$$J_{Px}\ddot{\theta}_x = \tau_x \tag{2.6}$$

Or

$$\ddot{\theta}_x = u_x \tag{2.7}$$

And

$$J_{Py}\ddot{\theta}_y = \tau_y \tag{2.8}$$

Or

$$\ddot{\theta}_y = u_y \tag{2.9}$$

The state space equations of the ball and plate system can be expressed as (Fan *et al.*, 2004):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} x_2 \\ B\left(x_1 x_4{}^2 + x_4 x_5 x_8 - g\sin x_3\right) \\ x_4 \\ 0 \\ x_6 \\ B\left(x_5 x_8{}^2 + x_1 x_4 x_8 - g\sin x_7\right) \\ x_8 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{2.10}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} [X] \tag{2.11}$$

where:

$$B = \frac{m_b}{\left(m_b + \frac{J_b}{R_b{}^2}\right)} \tag{2.12}$$

In order to obtain a linearized ball and plate system model, the following assumptions are considered (Andinet, 2011).

i) The motor can control each step of the ball. This implies that skipping between steps does not occur and that the magnitude of the plant moment does not affect

the position of the rotor. In fact, the system inputs are $\theta_x$ and $\theta_y$, and not the torque moments $\tau_x$ and $\tau_y$, because the magnitude of the load moments does not affect the position of the motors. Due to this reason, equations (2.3) and (2.4) are neglected in the study of the ball and plate system.

ii) In the steady state, the plate should be in the horizontal position, where both inclination angles are equal to zero. If one assumes that the angle does not change beyond $\pm 5^0$, the sine function can be replaced by its argument.

iii) In equations (2.1) and (2.2), the velocities $\theta_x$ and $\theta_y$ are small, and hence, have negligible effects when squared or multiplied together.

Based on the above assumptions, the linearized, simplified and uncoupled ordinary differential equations are (Andinet, 2011):

$$\left\{ m_b + \frac{J_b}{R_b^{\,2}} \right\} \ddot{x} = -m_b g \theta_x \tag{2.13}$$

$$\left\{ m_b + \frac{J_b}{R_b^{\,2}} \right\} \ddot{y} = -mg \theta_y \tag{2.14}$$

$$J_b = \frac{2}{5} m_b R_b^{\,2} \tag{2.15}$$

Substituting the value of the moment of inertia of the ball in equation (2.15) into equations (2.13) and (2.14), and noting that the state space description is now split into two parts, owing to the simplified assumptions that led to the independence of motions in the X-axis, and Y-axis, then (Andinet, 2011):

$$\frac{7}{5} \ddot{x} = -g \theta_x \tag{2.16}$$

$$\frac{7}{5} \ddot{y} = -g \theta_y \tag{2.17}$$

*2.2.1.1 Control system design*

The system can be treated as two different systems operating simultaneously. Hence, similar but independent controllers can be used for controlling each coordinate of the ball motion (Awtar *et al.*, 2002).

Based on the linear model, a preliminary controller is designed with the scheme of a 'loop within a loop'. The first step involves the design of an inner loop where the encoder feedback is sent to the dc motors to achieve a servo position control.

The inner loop is then placed in an outer loop that controls the ball position. The next step in the control system design is to obtain a controller for the outer loop, based on the transfer function between ball position and the corresponding plate angle (Awtar *et al.*, 2002).

The overall control scheme can be explained as follows. While the controller in the outer loop computes the angle by which the plate should move to balance the ball, the inner loop controller actually moves the plate by that angle. Ideally, the inner loop should do this instantaneously, which is not possible in reality (Awtar *et al.*, 2002). This is shown in Figure 2.3

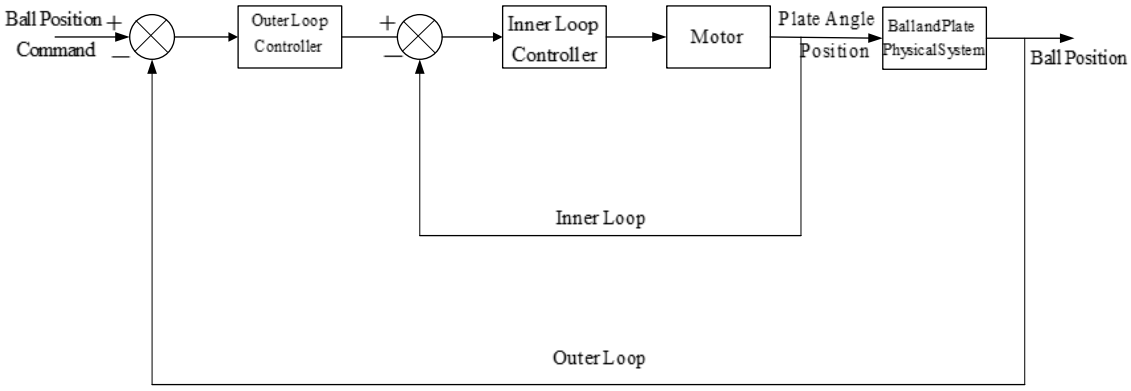



Figure 2.3 Control Scheme (Awtar *et al.*, 2002)

It is desirable to keep the speed of the inner loop much higher than that of the outer loop. This simple scheme is extremely effective in achieving the desired objective of balancing the ball (Awtar *et al.*, 2002).

### 2.2.2 Nonlinear Systems

Most physical systems are inherently nonlinear in nature. A common engineering practice in analysing nonlinear system is to linearize it about a nominal operating point and analyse the resulting linear model. A nonlinear system would be defined in the context of a linear system. Linear systems must satisfy two principles: superposition and homogeneity (Khalil & Grizzle, 1996).

The principle of superposition states that for two different inputs, $x$ and $y$, in the domain of the function $f$, (Khalil & Grizzle, 1996).

$$f(x+y) = f(x) + f(y) \tag{2.18}$$

The principle of homogeneity states that for a given input, $x$, in the domain of the function $f$, and for any real number $k$, (Khalil & Grizzle, 1996).

$$f(kx) = kf(x) \tag{2.19}$$

Any function that does not satisfy superposition and homogeneity principle is nonlinear. It is worth noting that there is no unifying characteristic of nonlinear systems, except for not satisfying the two above-mentioned principles (Khalil & Grizzle, 1996).

In order to ease the analysis of the ball and plate system, which is a nonlinear system, it is expected that the system should be linearized about its operating point. The operating point is that point at which all state and input variables are initialized to zero (Khalil & Grizzle, 1996).

### 2.2.3 Controllability and Observability

A system is said to be controllable if a control vector $u(t)$ exist that will transfer the system from any initial state $x(t_0)$ to some final state $x(t)$ in a finite time interval (Burns, 2001).

A system is said to be completely state controllable if the Kalman controllability matrix (Burns, 2001):

$$M_c = \left[ B \vdots AB \vdots \cdots \vdots A^{n-1}B \right] \tag{2.20}$$

is of full rank, that is it contains *n* linearly independent column or row vectors.

A system is said to be observable if at time $t_0$ , the system state $x(t_0)$ can be exactly determined from observation of the output $y(t)$over a finite time interval (Burns, 2001).

A system is said to be completely state observable if the kalman observability matrix (Burns, 2001):

$$M_0 = \left[ C^T \vdots A^T C^T \vdots \left(A^T\right)^{n-1} C^T \right] \tag{2.21}$$

is of full rank, that is, it contains *n* linearly independent rows or columns vectors.

The concept of controllability and observability from the control system perspective plays an important role in verifying whether a control solution exists or not from the beginning. This is important as most physical systems are controllable (stabilizable) and observable (detectable), whereas the corresponding mathematical models may not possess the property of controllability and observability. However, if the model is found not to be controllable and observable, it implies the mathematical model is not the true representation of the physical system, and as such may not yield an optimal control solution (Ogata, 2002).

*2.2.3.1 Stability*

The condition for stability is that all the poles must lie in the left hand of the s-plane (LHP), as shown in Figure 2.4.

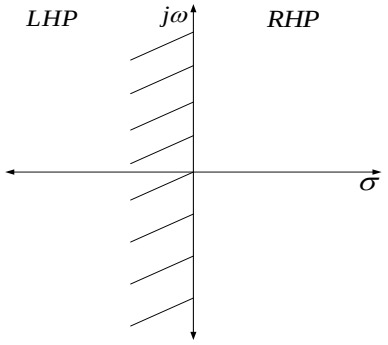



Figure 2.4: Root Locus Plot of System Indicating Region of Stability (Dingyu *et al.,* 2007)

If a system is unstable (that is its poles exists on the right hand of the s-plane (RHP)) but controllable, it means the poles can be transferred to the left hand plane, but if the system is also observable, the designer has the added flexibility of placing the poles at any location in the left hand plane (Friedland, 2012).

The requirements for the dynamics of a closed loop linear time varying system is stability, because the feedback control of the closed loop system tends to overcorrect errors that was caused by the oscillations of constant or changing amplitude. Stability can be defined in two categories (Friedland, 2012):

i) Ability of a system to return to equilibrium after an arbitrary displacement away from the equilibrium.

ii) The ability of the system to produce a bounded output from a bounded input.

The basic stability criterion is directly determined by location of the system poles or eigenvalues i.e. the roots of the characteristic equation of the system.

The ability of a system to return to equilibrium to a forced system is related as (Friedland, 2012):

$$\dot{x} = Ax + Bu \tag{2.22}$$

The solution of the first order $\dot{x}(t)$ is given as (Friedland, 2012):

$$\dot{x}(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau \tag{2.23}$$

Over an initial state condition $x(0)$

where:

$e^{At}$ is the state-transition matrix.

$x(t)$ is the state solution which captures all the state variables.

The properties of state transition matrix as related to stability are as follows (Friedland, 2012):

i) If the real parts of all the characteristics roots are strictly negative (i.e. not zero or positive), then $e^{At}$ tends asymptotically to zero. Hence, no matter how large the initial state $x(0)$ is, $x(t) \to 0$ as $t \to \infty$. The system is said to be asymptotically stable.

ii) If any characteristic root has a strictly positive real part, the state transition matrix will have at least one term which will tend to infinity as $t \to \infty$. In this case, it is possible to find some initial state which will cause $x(t)$ to become infinite. This system is said to be unstable.

iii) If all the characteristic roots have non-positive real part, and has any characteristic roots that has a zero real part which is repetitive, then, there is at least one or more term that will tend to infinity as $t \to \infty$. This system is said to be unstable.

If the model under study has been determined to be controllable and observable, then, there is a need to explore the various types of controllers that will be used in effecting

control and stabilization of the system in order to achieve a specified performance specification.

*2.2.3.2 Trajectory and motion tracking*

Stabilization is difficult and challenging, but tracking is practically important. Tracking of nonlinear systems is usually specified by a control objective to track a trajectory with the exponential convergence rate in order to guarantee performance and robustness. (Jarzebowska, 2012).

Trajectory tracking of nonlinear system is achieved based on two kinds of models. The first model considers the velocities of the system as control inputs and uses the system kinematic model. It addresses tracking at a kinematic control level and ignores the system dynamics. The second model uses the system dynamics, where control forces and torques as well as velocities can be control inputs. For a nonlinear system, a dynamic control model is usually integrated with a kinematic control model. A control system developed in such a way has two-level architecture. The lower control level operates within a kinematic model of the nonlinear system to stabilize its motion to a desired trajectory. The upper control level uses a dynamic model and stabilizes feedback obtained on the lower control level. This allows steering commands, which are velocities at the kinematic level, to be converted to forces and torques. The control system developed within this two-level architecture assumes a perfect knowledge of the system dynamics in most cases (Jarzebowska, 2012).

In classical nonlinear control, tracking usually means trajectory tracking, and it is defined as follows (Jarzebowska, 2012):

Given a nonlinear system whose dynamics is described by (Jarzebowska, 2012):

$$\dot{q}(t) = f\left(q(t), u(t), t\right) \tag{2.24}$$

where $f$ is a nonlinear vector function, $q$ is a state variable vector, and $u$ is the input vector.

With

$$q(t_0) = q_0 \tag{2.25}$$

$$z(t) = g\left(q(t), u(t), t\right) \tag{2.26}$$

where $z(t)$ is the system output.

And a desired output $z_d(t)$, find a control law $u$ such that, starting from anywhere in a bounded region. The tracking errors

$$e = z(t) - z_d(t) \tag{2.27}$$

goes to zero, and the whole system remains bounded (Jarzebowska, 2012).

*2.2.3.3 Path following*

In the tracking problem, a system is to track a desired motion, typically a trajectory, which means that it is expected to reach a sequence of desired positions in specified time instants. It means that the reference is time parameterized. In path following, a system is required to reach and follow some path without any temporal specification. The basic property of this control task is that a controller acts on a system orientation to drive it to the specified path (Jarzebowska, 2012).

The path following control problem can be formulated then as the stabilization to zero of a scalar path error function, which is a distance to the path. Path following can be

treated either as a sub-problem of trajectory tracking or stabilization. In trajectory tracking, the separation of the geometric and timing information along the trajectory is needed, while in the stabilization, the controlled output, which is a distance to the path, is to be made zero (Jarzebowska, 2012).

### 2.2.4 Types of Controllers

Some common types of controllers are as follows (Burns, 2001):

i) **Proportional Integral Differential (PID) Controllers:** These are the most common controllers that find application in various industrial control systems. The basic importance of controllers is associated with its operational and functional simplicity. PID controllers are attributed by three parameters: proportional gain $K_p$ , integral gain $K_i$ , and derivative gain $K_d$ . In a heuristic sense, these parameters are defined in terms of time. $'P'$ is dependent on the present error.$'I'$ on the accumulation of past errors, and $'D'$ on the prediction of future errors (Singh & Yadav, 2012). PID controllers could be referred to as a compensator, in many control analysis, this is due to its ability to correct errors in a given control system. However, most PID controllers do not yield optimal control performance.

ii) **Artificial Intelligence (AI) Controllers**: Intelligent controllers are controllers that employ artificial intelligence paradigm (neural network, fuzzy logic, neuro-fuzzy, genetic algorithm, swarm techniques etc.) as control algorithms. They are generally popular in the control of nonlinear systems because of their universal function approximation abilities.

iii) **Optimal Controllers:** A linear optimal controller is a linear quadratic controller that provides an optimal control law for a linear system with quadratic

performance with respect to certain measure of performance specification. Typical linear optimal controllers include: Linear quadratic regulator (LQR), Linear Quadratic Tracking (LQT) and Linear Quadratic Gaussian (LQG) (Singh & Yadav, 2012).

**iv) Robust Controllers:** Robust controllers are controllers that provide a control law which maintains system response and error signals within prescribed tolerances despite the effects of uncertainty on the system. Typical robust controllers include: $H_2$-, $H_\infty$- and $\mu$-synthesis.

Forms of uncertainty include (Dingyu *et al.*, 2007):

a) Disturbance effects on the plant

b) Measurement noise

c) Modelling errors due to nonlinearities

d) Modelling errors due to time-varying parameters

*2.2.4.1 $H_\infty$ controller*

The augmented plant model for $H_\infty$ controller can be constructed as (Dingyu *et al.*, 2007):

$$P(s)=\left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] \tag{2.28}$$

With the augmented state space description as follows:

$$\dot{x} = Ax + \begin{bmatrix} B_1 & B_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.29}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} x + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.30}$$

Straightforward manipulations give the following closed loop transfer function:

$$Ty_1u_1(s) = P_{11}(s) + P_{12}(s)\left[I - F(s)P_{22}(s)\right]^{-1} F(s)P_{21}(s) \tag{2.31}$$

The above expression is also known as the linear fractional transformation (LFT) of the interconnected system. The objective of robust control is to find a stabilizing controller

$$u_2(s) = F(s)y_2(s) \tag{2.32}$$

Such that

$$\left\|Ty_1u_1\right\|_\infty < 1 \tag{2.33}$$

Based on equation (2.30), the following three robust control problems are particularly interesting in control engineering practice (Dingyu *et al.*, 2007):

i) The $H_2$ optimal control problem: $\min_{F(s)} \| Ty_1u_1 \|_2$ ;

ii) The $H_\infty$ optimal control problem: $\min_{F(s)} \| Ty_1u_1 \|_\infty$ ;

iii) The standard $H_\infty$ robust control: $\| Ty_1u_1 \|_\infty < 1$.

The design objective is to find a robust controller $F_c(s)$ guaranteeing the closed-loop system with an $H_\infty$-norm bounded by a given positive number $\gamma$, i.e. (Dingyu *et al.*, 2007)

$$\| Ty_1u_1 \|_\infty < \gamma \tag{2.34}$$

Then the controller can be represented by (Dingyu *et al.*, 2007):

$$F_c(s) = \left[\begin{array}{c|c} A_f & -ZL \\ \hline F & 0 \end{array}\right] \tag{2.35}$$

where:

$$A_f = A + \gamma^{-2}B_1B_1^T X + B_2F + ZLC_2 \tag{2.36}$$

$$F = -B_2^T X \tag{2.37}$$

$$L = -YC_2^T \tag{2.38}$$

$$Z = \left( I - \gamma^{-2} YX \right)^{-1} \tag{2.39}$$

And X and Y are, respectively the solutions of the following two Algebraic Riccati Equations (AREs) (Dingyu *et al.,* 2007):

$$A^T X + XA + X\left( \gamma^{-2} B_1 B_1^T - B_2 B_2^T \right) X + C_1 C_1^T = 0 \tag{2.40}$$

$$AY + YA^T + Y\left( \gamma^{-2} C_1^T C_1 - C_2^T C_2 \right) Y + B_1^T B_1 = 0 \tag{2.41}$$

The conditions for the existence of an $H_\infty$ controller are as follows (Dingyu *et al.,* 2007):

i) $D_{11}$ is small enough such that $D_{11} < \gamma$ ;

ii) The solution $X$ of the controller ARE is positive-definite;

iii) The solution $Y$ of the observer ARE is positive-definite;

iv) $\lambda_{\max}\left( XY \right) < \gamma^2$, which indicates that the eigenvalues of the product of the two Riccati equation solution matrices are all less than $\gamma^2$.

A MATLAB function $\text{hinf}( )$ is provided in the robust control toolbox for $H_\infty$ controller design. The syntax is $\left[ F_c, G_{cl} \right] = h\text{inf}\left( G \right)$ , the returned tree variable $F_c$ is the designed $H_\infty$ controller in the state space form, and $G_{cl}$ is the state space object of the closed loop system.

In the design of optimal $H_\infty$ controller, the optimal criterion is defined as (Dingyu *et al.,* 2007):

$$\max_{\gamma} \| Ty_1 u_1 \|_\infty < \frac{1}{\gamma} \tag{2.42}$$

and in general,

$$\max_{\gamma} \begin{bmatrix} W_1 S \\ W_2 FS \\ W_3 T \end{bmatrix} \leq \frac{1}{\gamma} \tag{2.43}$$

*2.2.4.2 $H_\infty$ Mixed sensitivity problem*

In the design of the $H_\infty$ optimal control, using the mixed sensitivity problem, the weighting functions which are $W_1(s)$, $W_2(s)$ and $W_3(s)$ are used for shaping the plant model $G(s)$. The weighting function $W_1(s)$, penalises the error signal, $W_2(s)$ penalises the input signal and $W_3(s)$ penalises the output signal (Hossain, 2007). This is shown in Figure 2.5.

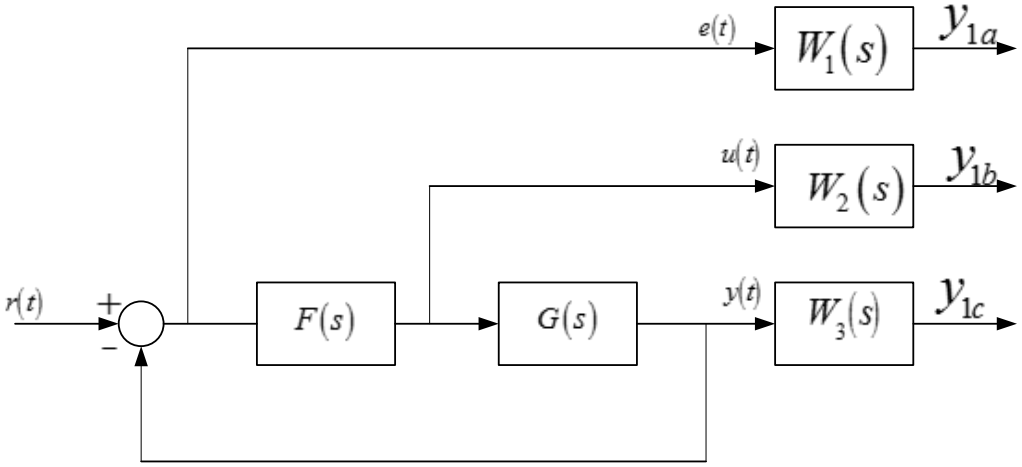



Figure 2.5: Block Diagram of Mixed Sensitivity Functions (Dingyu *et al.*, 2007)

However, these weighting functions are regarded as filters. They correct the errors that are being sent to the plant, and try to shape the plant to a certain specifications.

The augmented plant model $P(s)$ is written as (Dingyu *et al.*, 2007):

$$P(s) = \left[\begin{array}{c|c} W_1 & -W_1 G \\ 0 & W_2 \\ 0 & W_3 G \\ \hline I & -G \end{array}\right] \tag{2.44}$$

The linear fractional transformation (LFT) of the mixed sensitivity problem $Ty_1u_1$, the sensitivity transfer function $S(s)$ and the complementary sensitivity transfer function $T(s)$ are given as (Dingyu *et al.,* 2007):

$$Ty_1u_1 = \begin{bmatrix} W_1S \\ W_2FS \\ W_3T \end{bmatrix} \tag{2.45}$$

$$S(s) = \left[1 + F(s)G(s)\right]^{-1} \tag{2.46}$$

$$T(s) = 1 - S(s) = F(s)G(s)\left[1 + F(s)G(s)\right]^{-1} \tag{2.47}$$

where:

$F(s)$ is the controller, $S(s)$ is the sensitivity transfer function and $T(s)$ is the complementary sensitivity function respectively.

The choice of the weighting functions is a trial and error method, which is not an easy task, one has to embed some certain design experience, intuition and domain knowledge of the robust controller design of the plant (Hossain, 2007).

### 2.2.5 Linear Algebraic Method

The linear algebraic approach or the inward approach is such that the overall transfer function is first determined to meet the design specifications and then implemented using a suitable configuration (Chen, 1995). Then, the search for parameters such that the resulting overall system will meet design specifications is carried out. The approach is essentially a trial-and-error method; therefore, one usually chooses the simplest possible feedback configuration namely, a unity feedback configuration, and start from the simplest possible compensator namely, a gain. If the design objective cannot be met by searching the gain, a different configuration or a compensator of degree 1 (phase-lead or phase-lag network) is chosen, and the search is repeated (Chen, 1995). The

approach starts from the internal compensators, and then to the overall system to meet the design specifications. (Chen, 1995). This is in contrast to the design of control systems using the root-locus method or the frequency-domain methods, in which the configuration and a compensator are first chosen with open parameters.

The choice of an overall transfer function is not entirely arbitrary. Consider a proper transfer function

$$G(s) = \frac{N(s)}{D(s)} \tag{2.48}$$

And an overall transfer function

$$G_0(s) = \frac{N_0(s)}{D_0(s)} \tag{2.49}$$

$G_0(s)$ is said to be implementable if there exists a configuration with no plant leakage, such that $G_0(s)$ can be built using only proper compensators. The no plant leakage constraint implies that all power must pass through the plant, and that no compensator be introduced in parallel with the plant.

However, $G_0(s)$ is said to be well-posed and totally stable. By well-posed, this implies that the closed loop transfer function of every possible input-output pair of the system is proper and stable to avoid amplification of high frequency noise and to avoid unstable pole-zero cancellation (Chen, 1995).

The necessary and sufficient conditions for $G_0(s)$ to be implementable are (Chen, 1995):

i) $G_0(s)$ must be stable

ii) $G_0(s)$ contains the non-minimum phase zeros of $G(s)$

iii) The pole-zero excess of $G_0(s)$ is equal to or larger than that of $G(s)$

From the above conditions, if the roots of $N(s)$ does not appear in $N_0(s)$, the only way to achieve this is to introduce the same root by feedback to cancel it. This cancellation is an unstable pole-zero cancellation if the root of $N(s)$ is in the closed right half $S$-plane. Therefore, all non-minimum-phase zeros of $G(s)$ must appear in $N_0(s)$. The poles of $G(s)$ or the roots of $D(s)$ are shifted by feedback, and it is immaterial whether $D(s)$is Hurwitz or not. Therefore, unstable poles of $G(s)$ do not impose any constraint on$G_0(s)$, but non-minimum phase zeros of $G(s)$ do. If the pole-zero excess inequality is met, then all poles and all minimum-phase zeros of $G_0(s)$ can be arbitrarily assigned (Chen, 1995).

The overall implementable transfer function $G_0(s)$ shall be chosen to fulfil both the steady state, and transient behaviours of the system $G(s)$.

*2.2.5.1 Transient and steady-state requirements*

The performance of a control system is generally specified in terms of two criteria, its transient response and its steady state requirements. This is because it renders a simple and straight forward procedure to compute the overall transfer function of a system (Chen, 1995).

i) **Transient Response**

The transient response of a control system consists of the settling time, rise time, overshoot and the steady state error. An optimal system is the system that has the smallest transient response and a better steady state performance (Chen, 1995). Integral of time multiplied by absolute error (ITAE) has the largest changes as $\zeta$ varies, therefore, has the best selectivity compared to integral of absolute error

(IAE) and integral of square error (ISE) performance indices, in order for the system to track the reference input with a better transient performance. ITAE also yields a system with a faster response than the other criteria (Chen, 1995).

A constraint on the actuator response or on the bandwidth of the resulting systems must be imposed; otherwise, it is possible to design an overall system to have a performance index as small as desirable, and the resulting actuator response will approach infinity (Andinet, 2011).

The problem is to find an optimal implementable $G_0(s)$ with the desired tracking performance as in (Chen, 1995):

$$\lim_{t\to\infty}\left|y(t)-r(t)\right|=0 \tag{2.50}$$

Where $y(t)$ is the plant output, and $r(t)$ is the reference input. ITAE is defined by (Chen, 1995):

$$J=\int_0^{\infty} t\left|e(t)\right|dt \tag{2.51}$$

Where $e(t)$ is the error between the reference input and the plant output at time $t$.

ii) **Steady State Requirement**

A control system with the overall transfer function must be designed so that the plant output $y(t)$ tracks eventually a reference input $r(t)$ without error. If the chosen transfer function $G_0(s)$ is given by (Chen, 1995)

$$G_0(s)=\frac{\beta_0+\beta_1 s+\beta_2 s^2+\cdots+\beta_m s^m}{\alpha_0+\alpha_1 s+\alpha_2 s^2+\cdots+\alpha_n s^n} \tag{2.52}$$

With $\alpha_n>0$ and $n\geq m$ is said to achieve asymptotic tracking if it satisfies equation (2.50). The conditions for $G_0(s)$ to achieve asymptotic tracking depend on the type

of $r(t)$ to be tracked. If $r(t)$ is a step function, the conditions for $G_0(s)$ to achieve tracking are $G_0(s)$ is stable, $\alpha_0 = \beta_0$ at $t \geq 0$. If $r(t)$ is a ramp function, the conditions are $G_0(s)$ stable, $\alpha_0 = \beta_0$ and $\alpha_1 = \beta_1$ at $t \geq 0$ (Chen, 1995).

*2.2.5.2 Implementation by two-parameter configuration*

Given a plant transfer function $G(s)$ and an implementable model $G_0(s)$, an overall system will be designed so that the overall transfer function matches or equal to $G_0(s)$.

If the denominator $D(s)$ and the numerator $N(s)$ of a plant transfer function have no common factors, then it is possible to achieve any model matching. The conditions of achieving matching depend on the degree of the compensators. The larger the degree of the compensators, the more parameters, we have for matching (Chen, 1995).

The two parameter configuration is used to implement the overall transfer function $G_0(s)$ and to calculate the desired compensators, considering the unity feedback system shown in Figure 2.6.

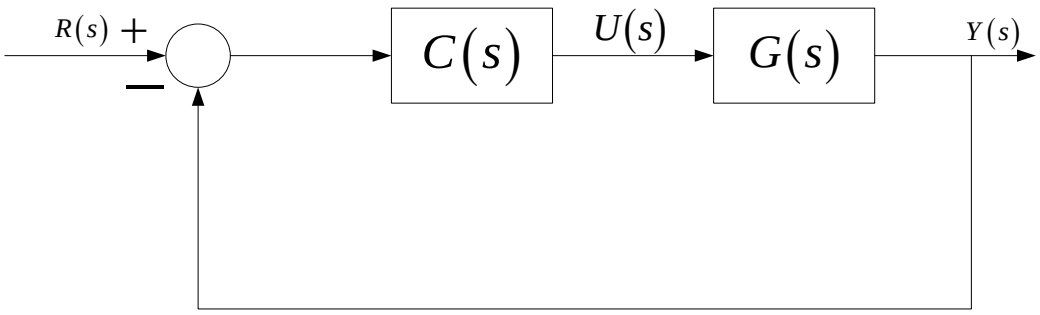



Figure 2.6: Unity Feedback System (Chen, 1995)

The actuator response is of the form (Chen, 1995)

$$U(s) = C(s)\big(R(s) - Y(s)\big) = C(s)R(s) - C(s)Y(s) \tag{2.53}$$

The same compensator is applied to the reference input and plant output to generate the actuator response, thus the generic form is

$$U(s) = C_1(s)R(s) - C_2(s)Y(s) \tag{2.54}$$

Where $C_1(s)$ is the feedforward compensator, and $C_2(s)$ is feedback compensator

$$C_1(s) = \frac{L(s)}{A_1(s)} \tag{2.55}$$

$$C_2(s) = \frac{M(s)}{A_2(s)} \tag{2.56}$$

Where $L(s)$, $M(s)$, $A_1(s)$, $A_2(s)$ are polynomials. In general, $A_1(s)$ and $A_2(s)$ need not be the same. It turns out that even if they are chosen to be the same, the two compensators can be used to achieve any model matching. Therefore, $A_1(s) = A_2(s) = A(s)$, and the compensators become (Chen, 1995):

$$C_1(s) = \frac{L(s)}{A(s)} \tag{2.57}$$

$$C_2(s) = \frac{M(s)}{A(s)} \tag{2.58}$$

$$U(s) = \frac{L(s)}{A(s)}R(s) - \frac{M(s)}{A(s)}Y(s) = \left[\frac{L(s)}{A(s)} - \frac{M(s)}{A(s)}\right]\begin{bmatrix} R(s) \\ Y(s) \end{bmatrix} \tag{2.59}$$

Thus, the compensator is written as (Chen, 1995)

$$C(s) := \left[C_1(s) - C_2(s)\right] = \left[\frac{L(s)}{A(s)} - \frac{M(s)}{A(s)}\right] = A^{-1}(s)\left[L(s) - M(s)\right] \tag{2.60}$$

However, the compensator will be implemented as a unit, in order to avoid the problem of possible unstable pole-zero cancellation. Furthermore, the configuration of Figure

2.7 that requires the minimum number of integrators will be used to design the controller (Chen, 1995).

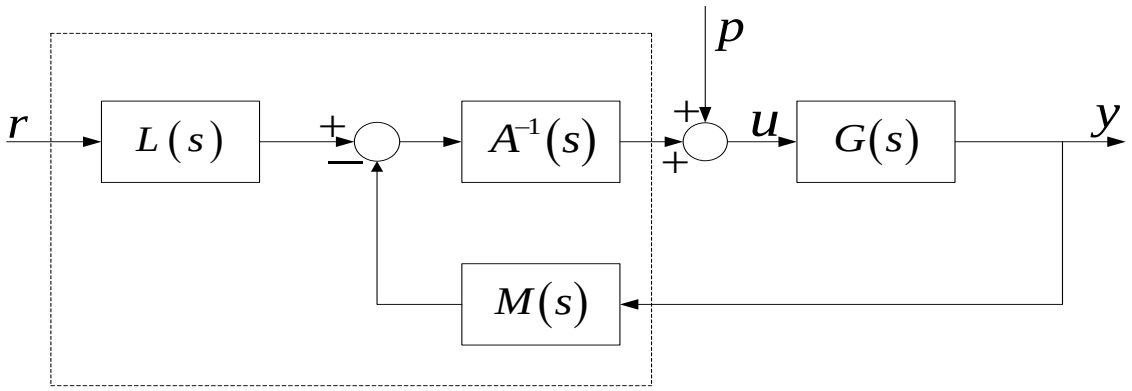



Figure 2.7: Two-Port Parameter Configuration (Chen, 1995)

From Mason's formula, the transfer function from $r$ to $y$ is given as (Chen, 1995)

$$\frac{Y(s)}{R(s)} = \frac{L(s)A^{-1}(s)G(s)}{1+A^{-1}(s)M(s)G(s)} \tag{2.61}$$

substituting $G(s) = \frac{N(s)}{D(s)}$, equation (2.61) becomes,

$$\frac{Y(s)}{R(s)} = \frac{L(s)N(s)}{A(s)D(s)+M(s)N(s)} \tag{2.62}$$

The problem of how to find appropriate compensator can be stated as follows (Chen, 1995):

If $G(s) = N(s)/D(s)$, $N(s)$ and $D(s)$ are coprime, $\deg N(s) < \deg D(s) = n$ and given an implementable $G_0(s) = N_0(s)/D_0(s)$, find proper compensators $L(s)/A(s)$ and $M(s)/A(s)$ such that

$$G_0(s) = \frac{N_0(s)}{D_0(s)} = \frac{L(s)N(s)}{A(s)D(s)+M(s)N(s)} \tag{2.63}$$

The procedure of finding the above solution is as follows (Chen, 1995):

***Step 1:*** Compute

$$\frac{G_0(s)}{N(s)} = \frac{N_0(s)}{D_0(s)N(s)} =: \frac{N_p(s)}{D_p(s)} \tag{2.64}$$

Where $N_p(s), D_p(s), N_0(s)$ and $D_0(s)$ are coprime.

***Step 2:*** Introduce an arbitrary Hurwitz polynomial $\bar{D}_p(s)$ so that the degree of $D_p(s)\bar{D}_p(s)$ is atleast $2n-1$.

***Step 3:*** Rewrite equation (2.60) as

$$G_0(s) = \frac{N(s)N_p(s)}{D_p(s)} = \frac{N(s)\left[N_p(s)\bar{D}_p(s)\right]}{D_p(s)\bar{D}_p(s)} = \frac{N(s)L(s)}{A(s)D(s)+M(s)N(s)} \tag{2.65}$$

Let

$$L(s) = N_p(s)\bar{D}_p(s) \tag{2.66}$$

And solve $A(s)$ and $M(s)$ from

$$A(s)D(s)+M(s)N(s) = D_p(s)\bar{D}_p(s) =: F(s) \tag{2.67}$$

If:

$$A(s) = A_0 + A_1 s + ... + A_m s^m \tag{2.68}$$

$$M(s) = M_0 + M_1 s + ... + M_m s^m \tag{2.69}$$

And

$$F(s) =: D_p(s)\bar{D}_p(s) = F_0 + F_1 s + F_2 s^2 + ... + F_{n+m} s^{n+m} \tag{2.70}$$

With $m \geq n-1$ , then $A(s)$ and $M(s)$ in equations (2.70), can be solved from the following linear algebraic equation (Chen, 1995).

$$\begin{bmatrix} D_0 & N_0 & 0 & 0 & \cdots & 0 & 0 \\ D_1 & N_1 & D_0 & N_0 & & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdots & 0 & 0 \\ D_n & N_n & D_{n-1} & N_{n-1} & & D_0 & N_0 \\ 0 & 0 & D_n & N_n & & D_1 & N_1 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & & D_n & N_n \end{bmatrix} \begin{bmatrix} A_0 \\ M_0 \\ \cdots \\ A_1 \\ M_1 \\ \cdots \\ \vdots \\ \cdots \\ A_m \\ M_m \end{bmatrix} = \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{n+m} \end{bmatrix} \tag{2.71}$$

In conclusion, its shows that the compensators are proper, and the two-parameter configuration is totally stable and posed. The configuration has no plant leakage, which implies that the configuration can be used to implement any implementable overall transfer function (Chen, 1995).

*2.2.5.3 Actuator parameters*

The actuator with a permanent DC motor and gear is considered in the design of the inner loop. DC motor which are extensively used in control systems exhibits backlash, due to its loose fit between two or more of the meshed gears (Golnaraghi & Kuo, 2010). Assuming there is no backlash and considering Figure 2.8.

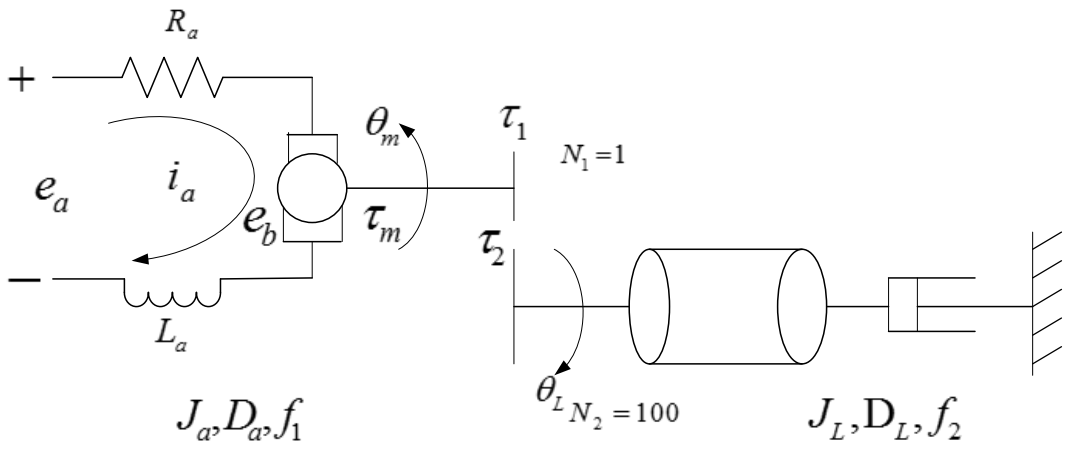



Figure 2.8: Model of a Separately Excited DC Motor (Golnaraghi & Kuo, 2010)

The DC motor is modelled as a circuit with resistance $R_a$ connected in series with an inductance $L_a$, and a voltage source $e_a$, the back electromotive force (emf) $e_b$, when the motor rotates. The motor variables and parameters are defined in Table 2.1

Table 2.1: DC Motor Variables and Parameters (Golnaraghi & Kuo, 2010)

| S/N | Description | Symbol | Unit |
|---|---|---|---|
| 1 | Applied Voltage | $e_a$ | $V$ |
| 2 | Back emf | $e_b$ | $V$ |
| 3 | Load Torque | $T_L$ | $Nm/A$ |
| 4 | Input Torque | $\tau_1$ | $Nm/A$ |
| 5 | Output Torque | $\tau_2$ | $Nm/A$ |
| 6 | Torque Constant | $K_m$ | - |
| 7 | Back emf Constant | $K_e$ | - |
| 8 | Rotor Inertia | $J_a$ | $Kgm^2$ |

The electrical equations relating to the input torque $\tau_1$ and the output torque $\tau_2$ are expressed as (Golnaraghi & Kuo, 2010):

$$\tau_1 = -J_a \frac{d^2\theta_m}{dt^2} - f_1 \frac{d\theta_m}{dt} + \tau_m \tag{2.72}$$

$$\tau_2 = J_L \frac{d^2\theta_L}{dt^2} + f_2 \frac{d\theta_L}{dt} \tag{2.73}$$

$$R_a I_a + L_a I_a s = e_a - K_e \theta_m s \tag{2.74}$$

$$\tau_m = K_m I_a \tag{2.75}$$

From equations (2.71) to (2.74), the relationship between $\theta_L$ and $e_a$ can be written as (Golnaraghi & Kuo, 2010):

$$\frac{\theta_L}{e_a} = \frac{K_t \frac{N_1}{N_2}}{\left[ J_{eq} L_a s^3 + \left( J_{eq} R_a + D_{eq} L_a \right) s^2 + \left( D_{eq} R_a + K_t K_e \right) s \right]} = T_L \tag{2.76}$$

Where:

$$J_{eq} = J_a + J_L \left( \frac{N_1}{N_2} \right)^2 \tag{2.77}$$

$$D_{eq} = D_a + D_L \left( \frac{N_1}{N_2} \right)^2 \tag{2.78}$$

From equation (2.75), the transfer function block diagram is given in Figure 2.9.

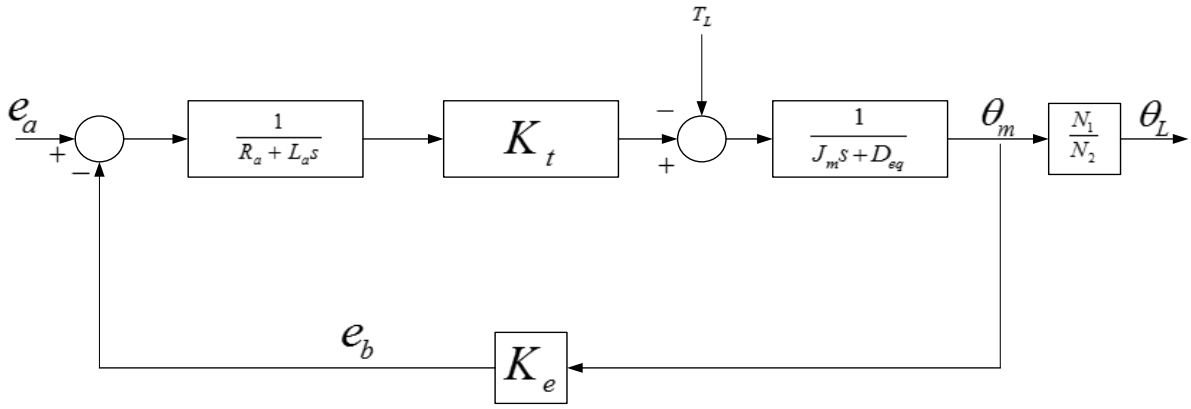



Figure 2.9: Transfer Function Block Diagram between the Load Angle and Armature Voltage (Golnaraghi & Kuo, 2010)

*2.2.5.4 Inner loop design*

The inner loop serves as a DC motor servo angular position controller. The inner loop controller analysis is carried out using two-port parameter configuration. This is shown in Figure 2.10.

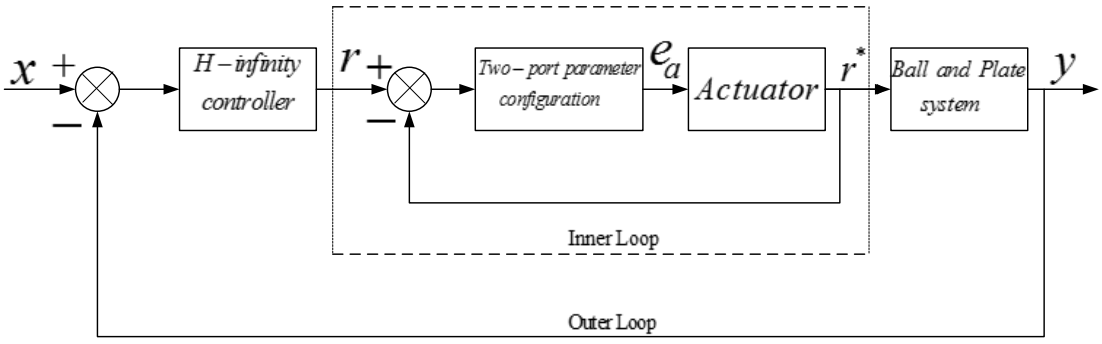



Figure 2.10: Developed Double Feedback Loop Structure of the Control Scheme

From equation (2.75), and Figure 2.9, the inner control loop is drawn using the two-port parameter configuration, which is shown in Figure 2.11.

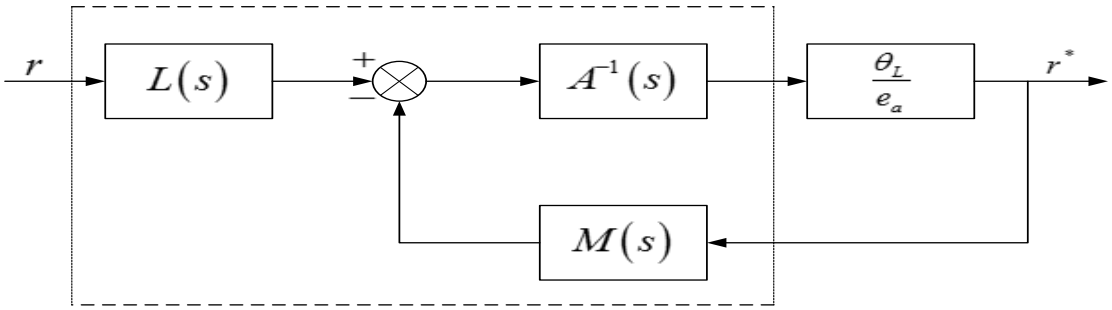



Figure 2.11: Implementation of the Two-Port Parameter Configuration of the Inner Loop

The ITAE optimal overall transfer function with zero position error of the system $G_0(s)$ is given as (Chen, 1995):

$$G_0(s) = \frac{\omega_0^{\,2}}{s^2 + 1.4\omega_0 s + \omega_0^{\,2}} \tag{2.79}$$

Also, the equations describing the step response of the actuator are given as (Chen, 1995):

$$G_0(s) = \frac{r^*}{r} = \frac{L(s)A^{-1}(s)G(s)}{1 + A^{-1}(s)G(s)M(s)} = \frac{L(s)G(s)}{A(s) + G(s)M(s)} \tag{2.80}$$

$$\frac{U(s)}{R(s)} = \frac{L(s)A^{-1}(s)}{1 + A^{-1}(s)G(s)M(s)} = \frac{L(s)}{A(s) + G(s)M(s)} = \frac{G_0(s)}{G(s)} \tag{2.81}$$

### 2.2.6 Virtual Reality Modelling Language (VRML) as a 3-D Modelling Tool

Virtual Reality Modelling Language (VRML) is a file format for describing interactive 3-D object and worlds. It is designed to be used on the Internet, intranets and local client systems. It is also intended to be a universal interchange format for the integrated 3-D graphics and multimedia. It is used in many application areas such as engineering and scientific visualization, entertainment and educational titles, web pages, multimedia presentations and shared virtual worlds (Andinet, 2011).

In the VRML model, a 3-D scene is described by a hierarchical tree structure of objects (nodes). Each node in the tree represents some functionality of the scene. There are 54 different types of nodes. Some of the nodes are *shapes nodes*, representing real 3-D objects, and some of them are *grouping nodes*, used for holding child nodes (MathWorks, 2015). Figure 2.12 shows the hierarchical structure of the nodes.

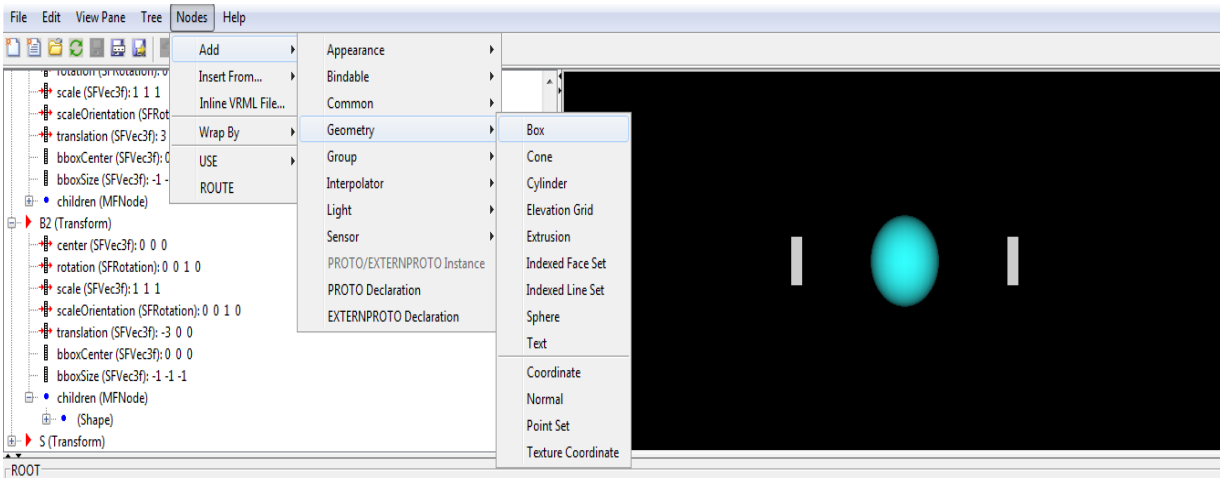



Figure 2.12: Hierarchical Tree Structure of Nodes (MathWorks, 2015)

Examples of nodes includes (MathWorks, 2015):

**i)** Box node: Represents a box in a scene.

**ii)** Transformation node: Defines the position, the scale, the scale orientation, rotation translation and the children of its subtree (grouping nodes).

iii) Material node: Corresponds to material in a scene.

iv) DirectionalLight node: Represents lighting in a scene.

v) Fog node: Allows you to modify the environment optical properties

vi) ProximitySensor node: Brings interactivity to VRML97. This node generates events when the user enters, exists, and moves within the defined region in space.

Each node contains a list of fields that holds values defining the parameters for its function. The nodes can be placed in the top level of a tree or as children of other nodes in the hierarchy of the tree. The position and orientation of child objects are specified, relative to the parent object. The parent object has its local coordinate space defined by its own position and orientation. Moving the parents object also moves the child object relative to the parent object (MathWorks, 2015).

In VRML, all lengths and distances are measured in metres, while all angles are measured in radians. It also uses the world coordinate system, the $x-axis$ represents the horizontal axis, the $y-axis$ points upwards, and the $z-axis$ places objects nearer or father from the front of the screen (MathWorks, 2015). This is shown in Figure 2.13.

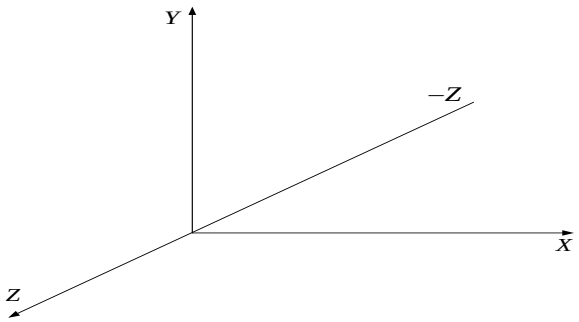



Figure 2.13: VRML Coordinate System (MathWorks, 2015)

The rotation angles are defined using the right-hand rule. The counterclockwise direction is the positive rotation angle of an object moving around that axis. This is shown in Figure 2.14 (MathWorks, 2015).

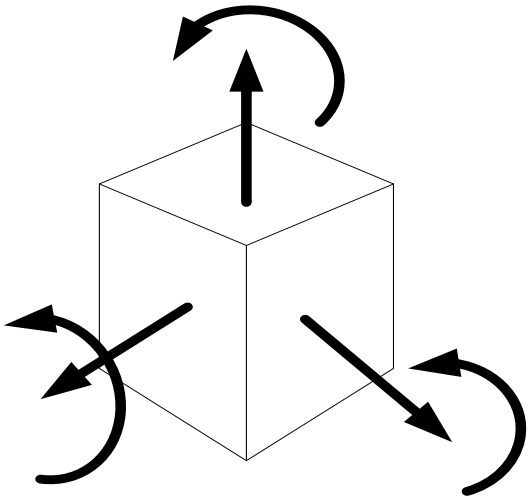


Figure 2.14: VRML Rotation Angles (MathWorks, 2015)

### 2.2.7 Simulink® 3D Animation

Simulink®3D Animation™ product uses VRML97 technology to deliver a 3-D visualization solution for MATLAB users. The Simulink®3D Animation™ product provides an interface to connect an existing virtual world, defined with VRML, to the Simulink® and MATLAB® environments. It is a useful contribution to a wide use of VRML97 in the field of technical and scientific computation and interactive 3-D animation (MathWorks, 2015).

The Simulink®3D Animation™ product is a solution for interacting with virtual reality models of dynamic systems over time. It extends the capabilities of the MATLAB and Simulink software into the world of virtual reality graphics. The necessary steps to use the Simulink®3D Animation™ product are (MathWorks, 2015):

***Step 1:*** Create virtual worlds or three-dimensional scenes using standard Virtual Reality Modelling Language (VRML).

***Step 2:*** Create and define dynamic systems with the MATLAB and Simulink products.

***Step 3:*** View moving three-dimensional scenes driven by signals from the Simulink environment.

Simulink®3D Animation™ library provides blocks to directly connect Simulink signals with virtual worlds. This connection lets us to visualize the Simulink model as a three-dimensional animation. Most of the software features can be implemented with Simulink blocks. Once these blocks are included in the Simulink diagram, one can select a virtual world and connect Simulink signals to the virtual world. It automatically scans a virtual world for available VRML nodes that the Simulink can drive. The VRML node properties are listed in a hierarchical tree-style viewer. One select the degrees of freedom to control from within the Simulink interface. After connecting these inputs to appropriate Simulink signals, the simulation is then viewed with a VRML viewer (MathWorks, 2015).

It contains a viewer that is the default viewing method for the virtual worlds. To create virtual worlds that are desired to be connected to the Simulink block diagrams, the VRML editor is used. It also includes one of the classic VRML authoring tools, V-Realm Builder by Ligos. With the addition of this VRML authoring tools, VRML products provides a complete authoring development, and working environment for carrying out 3-D visual simulations (MathWorks, 2015).

### 2.2.8 Graphical User Interface (GUI)

Graphical User Interface (GUI), refers to the universal idea of icons, buttons, etc., that are visually presented to a user as a "front-end" of a software application (Patrick & Thomas, 2003). However, GUI consist of a figure window containing menus, buttons, text, graphics etc. that a user can manipulate interactively with the mouse and keyboard. There are two main steps in creating a GUI (Hunt *et al.*, 2014):

i) One is designing its layout, and

ii) The other is writing callback functions that perform the desired operations when the user selects different features.

The strength of GUIs is that they provide a means through which individuals can communicate with the computer without programming commands. This components have become quite standardized and developed into a user friendly and intuitive set of tools. These tools can be used to increase the productivity of a user or to provide a window to the sophistication and power of a MATLAB application for those with little or no MATLAB programming experience (Hunt *et al.,* 2014).

The graphics objects components of GUI come in two classes (Hunt *et al.,* 2014):

i) User interface controls *(uicontrols)*

ii) User interface menus *(uimenus)*

The *uicontrols* and *uimenus* can be combined with other graphic objects to create informative, intuitive and aesthetically pleasing GUIs. Access to them is done using two approaches, the first approach is a low-level, bottom-up approach where the use of skills to handle graphics to write M-files that implement the GUI. The second approach is the use of MATLAB's Graphical User interface Development Environment (GUIDE), which is high-level, yet powerful and extremely easy to use; an excellent tool for quickly developing GUIs that takes care of much of the "bookkeeping" usually associated with GUI development (Hunt *et al.,* 2014).

## 2.3 Review of Similar Works

A summary of some of the relevant literature regarding the ball and plate system are described in this section.

Liu and Liang (2010) proposed a position tracking control scheme for the ball and plate system. At first, the property of the ball and plate control system was analysed, and the path tracking control problem was divided into sub-problems of ball position control and plate angle control. The double feedback loop structure, a loop within a loop was

employed in the design of the control scheme of the ball and plate. Sliding mode control algorithm was proposed for the ball position (outer loop) control algorithm based on dynamic model of the ball motion system and basic principle of sliding mode control. Asymptotical stability of the closed-loop system was proved mathematically through Lyapunov stability theory. Simulation was done in MATLAB, the results demonstrated that the proposed sliding mode control algorithm eliminates the effect of time-varying uncertain friction while the ball was rolling on the plate surface, thus, the path tracking precision was improved. However, the work did not state the control algorithm that was employed in the design of the inner loop, and did not consider the path trajectory tracking error of the ball on the plate, which reduced the accuracy of the path trajectory tracking of the ball on the plate.

Moreno-Armendáriz *et al.,* (2010) reported a novel approach of an indirect adaptive control using hierarchical fuzzy cerebeliar model articulation controller (CMAC) neural networks. The proposed method provided a simple control architecture that merges hierarchical structure, CMAC neural network and fuzzy logic. The input space dimension in CMAC is a time consuming task especially when the number of inputs is huge, this would overload the memory and make the neuro-fuzzy system very hard to implement. This can be simplified using a number of low dimensional fuzzy CMAC in a hierarchical form. Simulation and implementation result showed good stability of the ball on the plate, the hierarchical fuzzy CMAC (HFCMAC) controller solved the problem of overshoot, position and velocity error of the ball. However, they did not consider the path trajectory tracking of the ball on the plate, because their work was only limited to the stability analysis of the ball on the plate.

Dong *et al.*, (2011) proposed and tested a fuzzy online radius basic function (FRBF) tuning method optimized by particle swarm optimization algorithm (PSO) for the control problem of the ball and plate system. Particle swarm optimization algorithm fuzzy neural network control (PSO-FNNC) scheme was introduced for the ball and plate system. The fuzzy neural network control (FNNC) was optimized by the offline particle swarm optimization (PSO) of global searching ability, and the online radius basic function (RBF) algorithm ability of local searching. Application of the technology of PSO in fuzzy online radius basic function-proportional integral derivative (FRBF-PID) system of ball and plate system was helpful to improve the dynamic properties and stability. Design and simulation was done in MATLAB, and it showed that multi-layer neural network was suitable to process large amounts of information fuzzy control in real time system. Based on the fuzzy logic for overcoming the shortcomings in information processing and control, the PSO-FRBF-PID was more quickly, accurately and efficiently in control of the angle in the ball and plate system. PSO-FRBF-PID controller does stabilize the ball and plate system given set-point control and trajectory experiments. The parameters and rules optimized by PSO were excellent to adapt to varying system, which makes the controlled performance optimal or near optimal. However, they consider the ball and plate system as a single feedback loop system, which gave them a higher trajectory tracking error, when tracking the path trajectory tracking of the ball on the plate.

Zheng *et al.*, (2011) proposed the position control of the ball and plate system based on switching mechanism. The mathematical model and the corresponding simplified model of the ball and plate system were derived according to its dynamic characteristics. The analysis of the applications of different types of proportional-integral-differential (PID) controllers were given. Based on these analyses, different PID controllers including

integral action separation, anti-windup, neural network PID and switching mechanism were used respectively. These algorithms were tested on the GPB2001 ball and plate system experimental platform. Experimental results showed that the switching controller had better settling time and transient response than the other PID controllers. The work suggested that future researchers should study on how to design a controller with good robustness for position control. However, they did not consider the path trajectory tracking of the system, because their work was on stability analysis of the ball and plate system.

Ghiasi and Jafari (2012) proposed an optimal robust controller design via H-infinity approach. Simulation results showed that the proposed controller had strong robustness, satisfactory and eliminated the effect of linearization problems, unknown external disturbances and time varying uncertain friction while the ball rolling on the plate surface, thus, the trajectory tracking position was improved. However, they considered the ball and plate system as a single feedback loop system, which resulted in a higher trajectory tracking error.

Jafari *et al.,* (2012) proposed a linear quadratic Gaussian controller. The simulation result showed that the proposed controller had strong robustness, satisfactory and eliminated the effect of noises in the plant, and output measurement of uncertain friction, while the ball was rolling on the surface, thus, the trajectory tracking process was improved. However, they considered the ball and plate system as a single feedback loop system, which gave them a higher trajectory tracking error.

Wang *et al.,* (2012) described the identification of the ball and plate system using multiple neural networks. Based on the dynamic characteristics of the ball and plate

system, the mathematical description of the system was derived by using Lagrange equation. The nonlinear autoregressive moving average model (NARMA), which is a standard model for the characterization of general discrete nonlinear system, and NARMA-L2 which is another nonlinear model that is always used for identification of complex nonlinear system were used to describe the nonlinear ball and plate system. Back propagation (BP) neural network which has strong ability of approximation for nonlinear systems was used to train the approximate nonlinear function of these two models. Simulation was done in MATLAB, and it showed that NARMA-L2 and NARMA had better modelling results. The modelling error of NARMA-L2 model was larger than the NARMA model, it was still located in the allowed error range. NARMA-L2 model was preferred to be used than NARMA model because less calculation was needed. However, they did not consider the path trajectory tracking of the system because the work was on stability analysis of the ball on the plate.

Mochizuki and Ichihara (2013) proposed a design method of proportional-integral-derivative (PID) control based on the generalized Kalman-Yakubovich-Popov (KYP) lemma for the ball and plate. The design method had two features: the first one is that a structure of the controller was called integral-proportional differential (I-PD) to suppress a large input signal against a major change of the reference signal; the second one is that a filter was introduced into the feedback loop to reduce an influence of the noise measurement from the camera. Both the simulation and experimental results showed the effectiveness of the design method. However, they did not consider the trajectory tracking error of the ball on the plate, which reduced the accuracy of the path trajectory tracking of the ball on the plate.

Rastin *et al.,* (2013) proposed a fuzzy approach to control the ball and plate on a given path. It consists of two level Takagi, Sugeno (TS) type fuzzy type controller; the first level tries to lead the ball to the desired goal, and the second one prevents collision between the ball and obstacles. Three-revolute-revolute-spherical 3RRS parallel mechanism was used to control the angles of the plate. The plant had three symmetry RRS legs and a circular plate of diameter 52cm, a digital camera and a real time image processing was used for locating the ball. The location is sent to the controller, then the output of the controller operates using both inverse kinematics and dynamic of the mechanism. Parameters of the controllers were determined using linear quadratic regulator (LQR) and Bryson's rule. Simulation results showed good tracking and obstacle avoidance. However, their work was on trajectory tracking and obstacle avoidance of the ball on the plate and they did not consider the path trajectory tracking error of the ball on the plate, which reduced the accuracy of the ball tracking the desired path on the plate.

Pattanapong and Deelertpaiboon (2013) proposed a control technique for ball and plate system using fuzzy logic with adaptive integral control action to balance and track the ball for different types of trajectories. The adaptive integral can automatically adjust its values linearly and only active when the ball's position is within the specified distance error. The position controller was placed in the outer loop, while the servo controller was placed in the inner loop. Two inputs were used for the fuzzy controller design, the first one is the position error, while the second one is the difference of position error, known as the velocity error. The adaptive integral controller has one input with automatic gain adjustment. Experiments were conducted on four different cases to show the effectiveness of fuzzy controller with adaptive integral gain. The first one was to balance the ball at the specified point on the plate, called the balancing mode, the

second one was to allow the ball to follow any point on the plate specified by a mouse in real time, called the mouse tracking mode. The third one was a square tracking mode, which made the ball to track along the edge of the square. The last one was to track along the circle trajectory, called the circle tracking mode. However, they did not consider the trajectory tracking error of the ball on the plate, which reduced the oscillation and accuracy of the ball in tracking a desired path on the plate.

Wang *et al.*, (2014) reported a linear control law based on a reduced-order observer to attenuate the after effects caused by the friction on the output of the traditional proportional differential (PD) type control scheme instead of compensating the primary friction mechanics. A pseudo reference command was produced to counteract the output error during the steady state phase. The describing function technique was used to test whether the proposed method could generate limit cycles or not. The describing function method validated that the limit cycle could not appear using the proposed reference compensation method. The results of the mathematical simulations and practical experiments showed that the proposed method could achieve remarkable performance improvement comparing with the original PD, proportional-integral-differential (PID) and direct compensation strategies within the framework of the traditional double loop cascade control structure. However, the harmful effect caused by the friction when the system was operating in the low-velocity regime with high precision requirements which caused high steady error, large overshoot and long settling time was not considered.

Borah *et al.,* (2014) proposed a fractional order proportional derivative (FOPD) controller tuned by firefly algorithm that has a high convergence rate, robustness and the capability to attain global optimum in less number of iterations. The firefly

algorithm (FA) when compared to Ziegler Nichols (ZN) method results in a superior system performance in terms of time domain specifications. The flashing characteristics of real butterflies are; owing to their unisexuality, the fireflies move towards the more attractive and brighter ones regardless of their sex; the degree of attractiveness of a firefly is proportional to its brightness which decreases as the distances from the other flies increases; and, if there is not brighter or more attractive firefly than a particular one, then it will move randomly. The facts and contribution distinguishing the work are: firstly, the simulation was carried out without linearizing the ball and plate system. Secondly, the integer order proportional derivative (IOPD) controller designed had been tuned by firefly algorithm, which provided an improved result compared to the existing controller inbuilt in the system. Thirdly, the model (33-240, Feedback Instruments) taken for simulation is a newer version of the ball and plate system, with not much research done on it. The simulation results showed that the proposed FOPD had a significant reduction in the settling time and overshoot. However, they did not consider the angular velocity and tracking error of the ball and plate system, which reduced the oscillation and accuracy of the ball in tracking a desired path on the plate.

Galvan-Colmenares *et al.*, (2014) presented the dual proportional derivative (PD) control with nonlinear compensation to solve the regulation problem for the ball and plate system. A new dynamic model in the manipulator robot form was obtained to use these controllers. The proposed nonlinear model was very useful to design and validate different control algorithms which can then be extrapolated to problems with the same characteristics. One advantage of working with nonlinear model is that the full dynamic can be seen, which made it possible to analyse the behaviour of the system at each equilibrium point. Lyapunov function was presented for the stability analysis; as a result of the analysis, it showed that the asymptotic stability of the closed-loop system could

be guaranteed. Experimental result showed the excellent performance of the proposed controller, obtaining the best behaviour in comparison with the other selected controllers. However, their work did not consider the path trajectory tracking of the system because the work was on the stability analysis of the ball on the plate.

Lin *et al.*, (2014) presented a fuzzy logic control (FLC) implementation on a ball and plate system for dynamic control verification using charged-coupled device (CCD) position sensing. A two degree-of-freedom (DOF) platform was constructed for experiments using four magnetic suspension (MS) actuators. The mathematical model of the ball and plate operation was derived in Euler-Lagrange equation, the system dynamics was carefully derived by considering force, moment and inertia relationship among ball, plate and MS actuators. The enforced fuzzy logic control (EFLC) was proposed to modify and improve the tracking performance of the system than that of the basic FLC, especially in overshoot. Simulation was done in MATLAB/GUI (graphical user interface) environment. Experimental results showed that using FLC, the ball and plate control could be implemented to accomplish the desired performance by considering its complex nonlinear characteristics. The effect from the Enforced FLC was excellent in comparison to the basic FLC. However, their work did not consider the path trajectory tracking of the system, because, the work was on stability analysis of the ball on the plate.

Lin *et al.*, (2014) discussed how to synthesize a stabilizable robust controller for the ball and plate system (BPS). A loop-shaping method in terms of a normalized right co-prime factor (NRCF) uncertainty description for the BPS was employed in order to reduce the fragility in its robust controller synthesis. The mathematical model of the BPS was derived using Lagrange equation, the well-known lead-lag series compensation design

methods were innovatively adopted to obtain appropriate pre and post compensators as the weighting functions to guarantee the BPS time domain performance requirements. Simulation results verified that the weighting functions design methods directly had a good effect on the design of the robust controller, especially, on the system performance specification. However, their work did not consider the path trajectory tracking of the system, because it was on stability analysis of the ball on the plate.

Oravec and Jadlovská (2014a) presented a way of design of the simulation models of the virtual models design in tool V-Realm builder of the mechatronic systems, and a way of linking models in the MATLAB and Simulink tools. The tool V-Realm Builder is a part of the Simulink 3D animation toolbox. An application *VirtModelsKKUI* was created in MATLAB tool, and virtual models in Simulink tool. The application was used for the simulation of the virtual models ball and beam (B&B) and ball and plate (B&P). The virtual models B&B and B&P provided more illustrative view of the simulation with control algorithms in chosen control scheme. However, their simulation work did not consider path trajectory tracking error of the ball on the plate, which reduced the accuracy of the ball tracking a desired path on the plate.

Oravec and Jadlovská (2014b) described a new mechatronic laboratory model ball and plate_KYB (B&P_KYB). This model was specific in construction and used construction parts. It was focused on the modelling of mechatronic laboratory model B&P_KYB, creation, analysis and validation of the simulation model B&P_KYB. An optimal linear quadratic (LQ) control algorithm was implemented with an integrator in discrete form. The algorithm was based on state model of mechatronic system B&P_KYB. A correct scheme have to be chosen for an optimal LQ algorithm with an integrator in discrete form. Simulation was done in MATLAB/SIMULINK, and the

simulation model was created in form Grey box model, which is one of many forms of creating simulation model. The simulation model after verification was suitable replacement of real model and it was possible for use for simulations. Simulation results of optimal LQ control in discrete form were not acceptable, because the response of real or simulation model B&P_KYB on reference trajectory was changing with time. However, their simulation work did not consider path trajectory tracking error of the ball on the plate, which reduced the accuracy of the path trajectory of the ball on the plate.

Roy *et al.,* (2015) proposed a design of proportional and derivative (PD) controllers in cascade control scheme to control trajectory of a ball in a ball and plate system. A cascaded control strategy was adopted to make the ball follow a reference trajectory. Proportional derivative controllers were used for both the inner and outer loop of the control system. Parameters of the PD controllers were tuned using particle swarm optimization technique (PSO) which does not require in-depth analysis of the nonlinear plant dynamics. A decoupled nonlinear model of the system was realized in MATLAB-Simulink environment. The result showed that PSO can be successfully used to tune PD controllers (inner and outer loop) of such nonlinear inherently unstable system. However, the overshoot of the system was not considered due to variation of the system parameters and external disturbances of the PD controller. The work recommended that future work should focus on hardware implementation of the system, the robustness of the system due to variation of system parameters and external disturbances should be tested and a more complex reference signal like random signal should also be used. However, the work did not consider the path trajectory tracking error of the ball on the plate, which reduced the accuracy of the ball tracking a desired path on the plate.

Negash and Singh (2015) proposed system in which a sliding mode control, one of the tools available in designing robust controllers, was introduced in the outer loop of a double-loop feedback control of the ball and plate system. Fuzzy logic and genetic algorithm were used to design the outer loop controller in an optimal manner, while linear algebraic method was used to design the inner loop controller by solving a set of Diophantine equations. The mathematical model of the B&P system was obtained from Euler-Lagrange equations of motion. The controller was evaluated using simulation studies. However, the result showed that the ball could be stabilized anywhere on the plate in 3.5 seconds, and could also track a circular trajectory of 0.4 metre radius at 0.8 radians/seconds in 10 seconds without significant chattering. However, the work did not consider the extent and effect of the tracking error due to chattering.

Oravec and Jadlovská (2015) presented an implementation of the predictive state space control algorithm, called model predictive control (MPC), which is a modern method of nonlinear dynamic system control. This control algorithm was verified on the ball and plate laboratory model, called B&P_KYB, for the reference trajectory tracking. The chosen reference trajectories were circle and square. Simulation was done in MATLAB/SIMULINK, the simulation results showed that the MPC algorithm fulfils the requirements of the control quality of the reference trajectory. The MPC algorithm was compared to the optimal linear quadratic (LQ), and had better results of the simulation of the reference trajectory tracking than the LQ algorithm. The nonlinear simulation laboratory model had the required quality, the control was smooth, without strong oscillations, which is very important for the lifetime of servomotors. However, the work did not consider the path trajectory tracking error of the ball on the plate, which reduced the accuracy of the ball tracking the path trajectory on the plate.

Oravec *et al.,* (2015) described the design of an adaptive control algorithm and its application for simulation of a model of the ball and plate (B&P) mechanical system. The mathematical description of the system was derived from the Euler-Lagrange differential equations and the design of the adaptive control algorithm was achieved with the elements of the self-tuning. A recursive least squares method algorithm was used for identification of the system parameters, and the system description assumed ARX model. Simulation was done in MATLAB/SIMULINK, it was verified on the laboratory model B&P_KYB. The simulation results showed a better trajectory tracking reference, when compared with the results using control system based on the pole placement method without adaptation. However, the work did not consider the trajectory tracking error of the ball on the plate, which reduced the precision when tracking the desired trajectory on the plate.

Kassem *et al.,* (2015) proposed four strategies for static and dynamic position tracking: proportional-integral-differential (PID), linear quadratic regulator (LQR), sliding mode and fuzzy controller. The control problem of the ball and plate system (BPS) was tackled with six degree-of-freedom (6 DOF) Stewart platform. The BPS which is a typical multi-variable nonlinear system, which is a two dimensional expansion of the ball and beam system. The complexity of this system increased when the plate was mounted on a Stewart platform with 6 DOF which made the system a big challenge in the control field. The simulation was done in MATLAB 2009 environment, and the results showed that the comparison between the proposed strategies illuminates the capability of the sliding mode control to solve the problem of static and dynamic tracking for nonlinear systems with high complexity as it overcomes modelling imprecision and parameter variations. However, the trajectory tracking error was large, due to the problem of chattering.

Cheng and Tsai (2016) presented a skilful robotic wrist system using a visual control technique to demonstrate dexterity of the mechanical wrist from the viewpoint of the table tennis. The Euler-Lagrange method was employed to obtain the dynamic equations of the ball-plate system. The plate was operated by a two degrees-of-freedom mechanical wrist driven by two servo motors. A web camera Philips SPC300NC was employed to simulate human eye to visualize location of the ball on the plate. Intelligent control algorithm using Linear Quadratic Regulator (LQR) approach was developed to adjust the plate's attitude to guide the ball to a specific position to achieve given balancing tasks. Two balancing experiments, including positioning the ball at central point of the plate and allowing the ball to follow a circular path, were conducted to demonstrate satisfactory and promising control performance. Simulation was done in MATLAB/SIMULINK, and it showed that both simulation and experimental results demonstrated similar position and planar trajectory tracking responses. However, they did not consider the position error of the ball, which was caused when the friction dominates the ball's motion.

Xiao and Buttazzo (2016) showed how to manage time and achieve performance improvements by using adaptive strategies in sensing and control. Sensory acquisition, control and actuation of the ball-and-plate system were performed on a STM32F4 microcontroller. A camera mounted on the embedded board was used to detect the ball position, and also, to cope with the limitations of the computational speed of the embedded board, an adaptive approach that changes the capture area of the image acquisition process at every execution was proposed for improving the detection accuracy. A system simulator using a TrueTime kernel that provides multimasking environment was developed as a support tool for designing the controller. Jitter compensation techniques, which adaptively update the proportional-integral-differential

(PID) control parameters at each sampling instant, were adopted to deal with the performance degradation due to the sampling jitter, and also the conditions for finding the set of proper PID control parameters, a practical issue of applying jitter compensation techniques was given. Simulation was done in MATLAB, experimental results showed that the system became stable using the adaptive capture window, due to the higher achievable detection rate, the system response of the controller without compensation suffered a larger overshoot and a lower rising time. The work suggested that future researchers should investigate the possibility of applying similar adaptive mechanisms to other embedded control applications, especially for those whose system performance is heavily affected by the hardware constraints. However, their work did not consider the path trajectory tracking of the system, because it was on stability analysis of the ball on the plate.

Cheng and Chou (2016) presented a visual servo control scheme for a ball and plate system with a maze. The maze built on the plate formed obstacles for the ball and increases variety and complexity of its environment, in order to guide the ball passing through the maze to the designated target on the plate. Two major tasks, visual servo and path planning were implemented. The position of the ball was located by finding the extreme pixels at the ball's boundary with maximal or minimal coordinate along both X and Y axes. The approach of distance transform was chosen for path planning due to its computational efficiency. The idea behind this approach was to follow the shortest distance measured based on a distance map generated by ripple wave pattern starting at the target position. Fuzzy control was chosen to implement the controller for the ball and plate system to mimic dexterous skills of human operation. The mamdani's method and the centroid approach were employed for rule inference and de-fuzzification, respectively. Simulation was done in Borland C++ environment with the

help of Directshow SDK for image acquisition. Three experiments including balancing, circular trajectory following, and path planning for the challenging maze were conducted. Both the simulation and experiments exhibited similar and satisfactory control responses. However, the work did not consider the path trajectory tracking error of the ball on the plate, which reduced the accuracy of the ball tracking the path trajectory on the plate.

From the reviewed literatures, it is evident that for an effective control of the ball and plate system, a double feedback loop structure that is a loop within a loop should be considered. However, previous researchers like Ghiasi and Jafari (2012) and Jafari *et al.,* (2012) had considered the ball and plate system as a single loop system, which gave them a higher trajectory tracking error when tracking a defined path by the ball. In this research work, the double feedback loop structure is employed, for the design of the ball and plate system. The inner loop is designed based on linear algebraic method, by solving a set of Diophantine equations and the outer loop is designed by using H-infinity sensitivity function.

# CHAPTER THREE

# MATERIALS AND METHODS

## 3.1 Introduction

In this chapter, the methods, materials and procedures used for the successful completion of this research are explained and these involves development of a position and trajectory tracking control scheme for the ball and plate system using a double feedback loop structure. The steps of the methodology adopted for this research, towards developing a position and trajectory tracking control scheme for the ball and plate system using a double feedback loop structure are as highlighted in section 3.2

## 3.2 Methodology

The methodology adopted in this research are highlighted as follows:

i) Adopting the model of the dynamics of the ball and plate system using Lagrange's equation.
    a) Decomposition of the nonlinear equation of the ball and plate system into single-input-single-output (SISO) system.
    b) Linearization of the decoupled nonlinear equation of the ball and plate system.
    c) Performing controllability and observability test on the ball and plate system.

ii) Selection of appropriate actuator parameters for the control of the ball and plate system.

iii) Designing the inner loop angular position controller using linear algebraic method.

iv) Designing the outer loop linear position controller using H-infinity sensitivity function for the ball and plate system.

v) Development of the:
    a) Virtual Reality (VR) model for the ball and plate system.

b) Inner loop controller and outer loop controller in MATLAB/SIMULINK 2013a environment.

c) Graphical User Interface (GUI) for the circular trajectory tracking of the ball on the plate.

vi) Validation by comparing with the work of Ghiasi and Jafari (2012) in terms of settling time, trajectory tracking error and maximum overshoot of the system as well as the work of Negash and Singh (2015) that considered the ball and plate system as a double loop structure.

### 3.2.1 Ball and Plate System Modelling

This section explains the decomposition of the nonlinear equation of the ball and plate system and linearization of the decomposed nonlinear equation of the ball and plate system.

### 3.2.2 Decomposition of the Ball and Plate System

As discussed in subsection 2.2.1, the mathematical model of the ball and plate system can be decomposed into X-axis and Y-axis as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ B\left(x_1 x_4{}^2 - g\sin x_3\right) \\ x_4 \\ 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} u_x \end{bmatrix} \tag{3.1}$$

$$\begin{bmatrix} \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} x_6 \\ B\left(x_5 x_8{}^2 - g\sin x_7\right) \\ x_8 \\ 0 \end{bmatrix} \begin{bmatrix} x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} u_y \end{bmatrix} \tag{3.2}$$

However, since the ball and plate system can be decomposed into two subsystems, X-axis and Y-axis, and can be controlled independently of each other, subsequent analysis was made on the X-axis direction.

From the state space model of the ball and plate system as given in equation (3.1) and (3.2), and substituting the value of $B$ and $J_b$ as given in equation (2.15) as:

$$B = \frac{m_b R_b^{\,2}}{m_b R_b^{\,2} + J_b} \tag{3.3}$$

However, since the ball and plate system was decomposed into the X and Y-axis, the state space model along the X-axis is given as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -7.007 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_x \tag{3.4}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} [x] \tag{3.5}$$

### 3.2.3 Linearization of the Ball and Plate System

The ball and plate system was linearized by considering the approximate value for a solid ball's moment of inertia as given in equation (2.15), substituting equation (2.15) into equations (2.1) and (2.2), and considering the following assumptions:

i) The plate inclination angle is not more than $\pm 5^0, \theta_x \approx 1$ and $\theta_y \approx 1$, the sine function of $\theta$ can be replaced by its argument.

$$\left.\begin{aligned} \sin\theta_x &= \theta_x \\ \sin\theta_y &= \theta_y \end{aligned}\right\} \tag{3.6}$$

ii) The velocities $\dot{\theta}_x$ and $\dot{\theta}_y$ are small and have negligible effect when squared and multiplied together.

$$\left.\begin{aligned} \theta_x &\approx 1 \\ \theta_y &\approx 1 \\ \dot{\theta}_x \dot{\theta}_y &\approx 0 \\ \dot{\theta}_x^{\,2} &\approx 0 \\ \dot{\theta}_y &\approx 0 \end{aligned}\right\} \tag{3.7}$$

Then, equations (2.16) and (2.17) are written as:

$$\frac{7}{5}\ddot{x} + g\theta_x = 0 \tag{3.8}$$

$$\frac{7}{5}\ddot{y} + g\theta_y = 0 \tag{3.9}$$

The two differential equations for the X-axis and Y-axis were found. Equations (3.8) and (3.9) were used to estimate the ball and plate system's states $x$, $y$, $\dot{x}$ and $\dot{y}$. Also considering the system inputs as $\theta_x$ and $\theta_y$, the transfer function can be written as:

$$G_x(s) = \frac{X_b(s)}{\theta_x(s)} = \frac{g}{\frac{7}{5}s^2} \tag{3.10}$$

$$G_y(s) = \frac{Y_b(s)}{\theta_y(s)} = \frac{g}{\frac{7}{5}s^2} \tag{3.11}$$

Subsequent analysis was made using the linearized model of the ball and plate system due to its symmetrical nature.

### 3.2.4 Controllability and Observability Test for the Ball and Plate System

In order to carry out position and trajectory tracking analyses, the controllability and observability tests were carried out on the system to show whether the system was controllable and observable. The controllability and observability test code of the ball and plate system is shown in Figure 3.1.

```
% Determination of the Controllability and Observability of the Ball and Plate System
A=[0 1 0 0;0 0 -7.007 0;0 0 0 1;0 0 0 0];
B=[0;0;0;1];
C=[1 0 0 0];
Mc=ctrb(A,B);
Mo=obsv(A,C);
if rank (Mc)==size(A)
    'System is Controlable'
else
    'System is NOT Contolable'
end
CONTR_RANK=rank(Mc)
if rank (Mo)==size(A)
    'System is Observable'
else
    'System is NOT Observable'
end
OBSV_RANK=rank(Mo)
if rank (Mo)==size(A)
    'System is Observable'
else
    'System is NOT Observable'
end
```

Figure 3.1: Controllability and Observability Test Code

### 3.3 Selection of the Actuator Parameters

The parameters of the DC motor were selected based on the requirements of the load torque, moment of inertia and the angular speed of the motor. Table 3.1 shows the parameters that are related to the ball and plate system.

Table 3.1: Parameters of the Ball and Plate System (HUMUSOFT Ltd., 2012)

| S/N | Description | Symbol | Unit | Value |
|---|---|---|---|---|
| 1 | Mass of the ball | $m$ | $kg$ | 0.11 |
| 2 | Radius of the ball | R | $m$ | 0.02 |
| 3 | Dimension of the plate (square) | $lxb$ | $m^2$ | 0.16 |
| 4 | Mass moment of inertia of the plate | $J_{Px,y}$ | $kgm^2$ | 0.5 |
| 5 | Mass moment of inertia of the ball | $J_b$ | $kgm^2$ | 1.76e-5 |
| 6 | Maximum velocity of the ball | $v$ | $m/s$ | 0.04 |

Since the angle of the plate is very small, and the angular velocity and angular acceleration depends on the plate inclination, the load torque of the system was obtained from equation (2.3).

$$\left(m_b x^2 + J_b + J_{Px}\right)\ddot{\theta}_x + 2m_b x\dot{x}\dot{\theta}_x + m_b xy\ddot{\theta}_y + m_b\left(\dot{x}y + x\dot{y}\right)\dot{\theta}_y + m_b gx\cos\theta_x = \tau_L \qquad (3.12)$$

Nothing that $\theta_x = \theta_y \approx 0$ and $\cos\theta \approx 1$, and substituting the value of the mass moment of inertia of the plate, from Table 3.1, equation (3.12) becomes:

$$\tau_L = m_b gx\cos\theta_x = 0.5396Nm \qquad (3.13)$$

From Andinet (2011), the angular speed was estimated for the actuator as $n_L = 20\,rev/\min$

However, the moment of inertia of the load was calculated from Table 3.1,

$$J_L = J_P + J_B = 0.5 + 1.76e^{-5} = 0.5000176 \approx 0.5\,kgm^2 \qquad (3.14)$$

From the above values of $\tau_L$ , $n_L$ , and $J_L$ , a DC motor was chosen from a certain manufacturer's catalogue, in this case, harmonic drives LLC which fulfilled these minimal requirements. This is given in Table 3.2

Table 3.2: Actuator Parameter Types (LLC, November 2016)

| **Actuator-Type** | $J_a\left(Kgm^2\right)$ | $\tau_{average}\left(Nm\right)$ | $\tau_N\left(Nm\right)$ | $\tau_{acceleration}\left(Nm\right)$ | $\tau_{maximum}\left(Nm\right)$ | $n_{load}\left(rpm\right)$ |
|---|---|---|---|---|---|---|
| RHS-14-6003 | 0.45 | 17.84 | 5.4 | 44.27 | 18 | 60 |
| RHS-14-3003 | 1.8 | 49.27 | 7.8 | 224.91 | 28 | 30 |
| RHS-17-6006 | 0.089 | 9.64 | 9.8 | 25.37 | 34 | 60 |
| RHS-17-3006 | 0.36 | 15.76 | 20 | 74.12 | 54 | 30 |
| RHS-20-6007 | 0.29 | 14.17 | 12 | 35.89 | 57 | 60 |
| RHS-20-3007 | 1.2 | 35.26 | 24 | 162.08 | 84 | 30 |
| RHS-25-6012 | 0.53 | 19.69 | 20 | 48.46 | 100 | 60 |
| RHS-25-3012 | 2.1 | 56.29 | 40 | 256.33 | 160 | 30 |
| RHS-32-6018 | 1.4 | 39.93 | 30 | 94.01 | 220 | 60 |
| RHS-32-3018 | 5.8 | 142.88 | 60 | 643.79 | 340 | 30 |
| RFS-20-6012 | 0.47 | 18.31 | 20 | 45.32 | 57 | 60 |
| RFS-20-3012 | 1.9 | 51.61 | 30 | 235.38 | 84 | 30 |
| RFS-25-6018 | 1.1 | 32.93 | 30 | 78.30 | 100 | 60 |
| RFS-25-3018 | 4.5 | 112.44 | 60 | 507.65 | 160 | 30 |
| RFS-32-6030 | 3.1 | 79.68 | 50 | 183.02 | 220 | 60 |
| RFS-32-3030 | 12.0 | 288.04 | 100 | 1293.10 | 340 | 30 |

Given the operating conditions as: $\tau_L = 5\left(Nm\right)$, $n_L = 50\left(rpm\right)$, $J_L = 0.3\left(kgm^2\right)$, $t_1 = 0.1\sec$, $t_2 = 0.2\sec$, $t_3 = 0.1\sec$, $t_4 = 0.6\sec$. The acceleration torque $\left(\tau_{acceleration}\right)$ and the average torque $\left(\tau_{average}\right)$ was calculated using the following equation (LLC, November 2016):

$$\tau_{acceleration} = \tau_L + 2\pi\left(\frac{\left(J_a + J_L\right)n_L}{t_1}\right) \tag{3.15}$$

$$\tau_{average} = \sqrt{\frac{\tau_{acceleration}{}^2 * t_1 + \tau_L * t_2 + T_3^2 * t_3}{t_1 + t_2 + t_3 + t_4}} \tag{3.16}$$

Where:

$$T_3 = \tau_L - \left(\tau_{acceleration} - \tau_L\right) \tag{3.17}$$

$T_3$ is the braking torque and $T_L$ is the load torque. From equations (3.15) to (3.17), the actuator that meets the requirement is RFS-20-6012. The DC servo actuators are usually equipped with control units, and the safety features are incorporated to protect the motors and control units from accidental damage.

Table 3.3 gives a summary of important parameters of RFS-20-6012.

Table 3.3 Some Parameters of Actuator RFS-20-6012

| S/N | Actuator Item | Actuator Parameter | Unit | Value |
|---|---|---|---|---|
| 1 | Rated Output Power | $P_{rated}$ | $KW$ | 0.123 |
| 2 | Rated Voltage | $V_{rated}$ | $V$ | 75 |
| 3 | Rated Current | $I_{rated}$ | $A$ | 2.9 |
| 4 | Rated Output Torque | $\tau_{rated}$ | $Nm$ | 20 |
| 5 | Rated Output Speed | $\omega_{rated}$ | $rpm$ | 60 |
| 6 | Maximum Output Torque | $\tau_{\max}$ | $Nm$ | 57 |
| 7 | Maximum Output Speed | $\omega_{\max}$ | $rpm$ | 80 |
| 8 | Torque Constant | $K_t$ | $Nm/A$ | 10.5 |
| 9 | Inertia at the Output Shaft | $J_a$ | $kgm^2$ | 0.47 |
| 10 | Viscous Damping Constant | $K_e$ | $Nm/rpm$ | 0.086 |
| 11 | Gear Ratio | $n_1/n_2$ | $1:R$ | 50 |
| 12 | Armature Resistance | $R_a$ | $\Omega$ | 1.2 |
| 13 | Armature Inductance | $L_a$ | $mH$ | 1.1 |

From Table 3.3, equation (2.75) can be written as:

$$\frac{\theta_L}{e_a} = \frac{0.105}{\left[0.47005s^2 + 421.113s\right]} \tag{3.18}$$

$$= \frac{0.2234}{s(s+895.89)} \approx \frac{2.49x10^{-4}}{s} \tag{3.19}$$

### 3.3.1 Two-Port Parameter Configuration

Two-port parameter configuration was used in the inner loop design. In order to find the value of $\omega_0$ , a step response is required, which should settle at 0.4 seconds, according to the Humosoft ball and plate system manual. Through simulation, the value of $\omega_0 = 20rad/\text{secs}$ was found to give the required steady state response.

From equation (2.78), substituting the value of $\omega_0$ ,

$$G_0(s) = \frac{400}{s^2 + 28s + 400} \tag{3.20}$$

The complete MATLAB code of the step response of the actuator is given in Appendix A1.

Equation (3.20) is rewritten with open parameters as $\frac{k_m}{s}$ , in order to satisfy the condition of the parameters of Table 3.3, which is for the rated power and rated voltage, the value of $k_m = 0.123KW$ and $V = 75V$ .

The condition becomes:

$$step\left(\frac{G_0(s)}{G(s)}\right) < 75 \tag{3.21}$$

$$step\left(\frac{400}{s^2 + 28s + 400} * \frac{s}{k_m}\right) < 75 \tag{3.22}$$

The step response of the actuator with open parameters is given in Appendix A2.

Considering equation (2.47) and (2.63)

$$G(s)=\frac{N(s)}{D(s)}=\frac{0.123}{s} \tag{3.23}$$

$$\frac{G_0(s)}{N(s)}=\frac{N_p(s)}{D_p(s)}=\frac{3252}{s^2+28s+400} \tag{3.24}$$

Introduce a Hurwitz polynomial, $\bar{D}_p(s)$, with its roots which lies inside an acceptable pole-zero cancellation region, so that the degree of $D_p(s)\bar{D}_p(s)$ is at least $2n-1$. In this case, $D_p(s)=2$, therefore $\bar{D}_p(s)=1$ was chosen.

From equations (2.65) and (2.66),

$$L(s)=N_p(s)\bar{D}_p(s)=3252 \tag{3.25}$$

$$A(s)D(s)+M(s)N(s)=s^2+28s+400 \tag{3.26}$$

Solving the Diophantine equation, (equations (3.25) and (3.26)), for the values of the compensator and the DC motor actuator model, the resulting equation was formed.

$$\begin{bmatrix} 0 & 0.123 & 0 & 0 \\ 1 & 0 & 0 & 0.123 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} A_0 \\ M_0 \\ A_1 \\ M_1 \end{bmatrix}=\begin{bmatrix} 400 \\ 28 \\ 1 \end{bmatrix} \tag{3.27}$$

Letting $M_1=1;$

$$A(s)=A_0+A_1s=28+s \tag{3.28}$$

$$M(s)=M_0+M_1s=3252+s \tag{3.29}$$

### 3.4 Determination of the $H_\infty$ Controller

The weighting functions $W_1(s)$, $W_2(s)$ and $W_3(s)$ for the control of the ball on the plate were selected after extensive simulation and fine tuning as:

$$W_1(s)=\frac{100\left(1.5s^2+12.64s+18.49\right)}{100\left(1.5s^2+103.2s+18.49\right)} \tag{3.30}$$

$$W_2(s) = 1 \tag{3.31}$$

$$W_3(s) = \frac{s^2}{100} \tag{3.32}$$

The MATLAB code of the $H_\infty$ Controller is given in Appendix A3. The following messages were displayed from the simulation as presented in Table 3.4.

Table 3.4: Iteration Results of the H-infinity Controller

<< H-Infinity Optimal Control Synthesis>>

| No | Gamma | D11<=1 | P-EXIST | P>=0 | S-EXIST | S>=0 | Lam(PS)<1 | C.L |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.0000e+00 | FAIL | OK | OK | OK | FAIL | OK | STAB |
| 2 | 5.0000e-01 | OK | OK | FAIL | OK | OK | OK | UNST |
| 3 | 2.5000e-01 | OK | OK | OK | OK | OK | OK | STAB |
| 4 | 3.7500e-01 | OK | OK | OK | OK | OK | OK | STAB |
| 5 | 4.3750e-01 | OK | OK | OK | OK | OK | OK | STAB |
| 6 | 4.6875e-01 | OK | OK | OK | OK | OK | OK | STAB |
| 7 | 4.8438e-01 | OK | OK | OK | OK | OK | FAIL | UNST |
| 8 | 4.7656e-01 | OK | OK | OK | OK | OK | FAIL | UNST |
| 9 | 4.7266e-01 | OK | OK | OK | OK | OK | OK | STAB |

### 3.5 Development of the Virtual Reality (VR) Model of the Ball and Plate System

The virtual reality model (VR) of the ball and plate system was developed on V-Realm builder. The process by which the (VR) model was developed is shown in Figure 3.2.

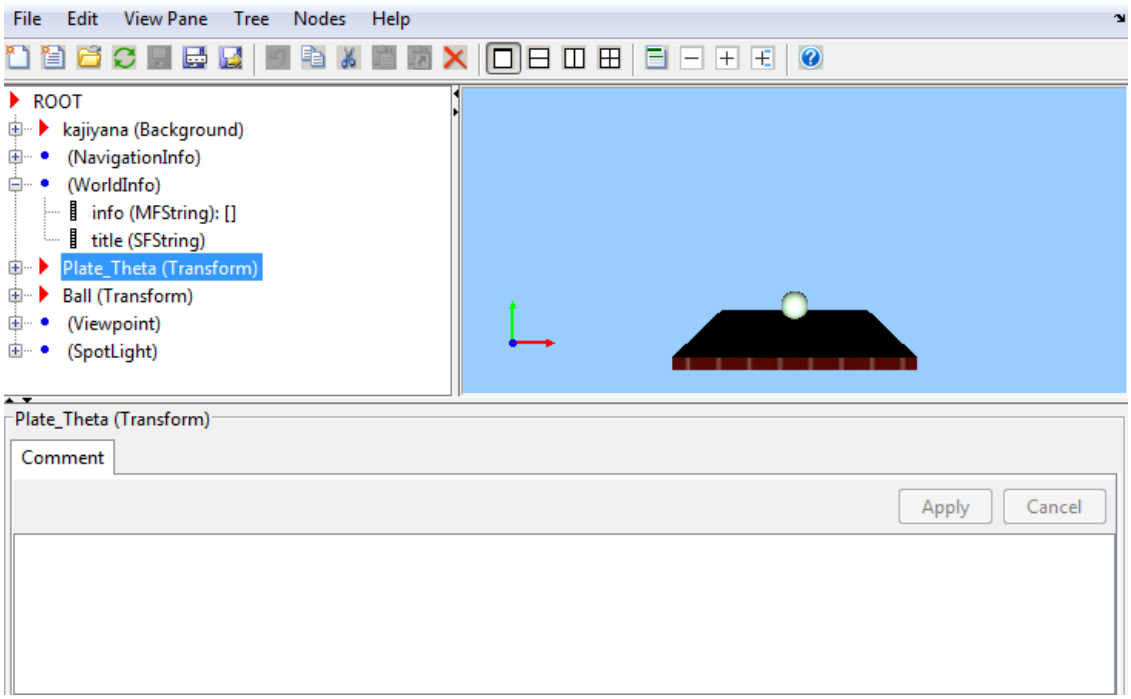



Figure 3.2: Screenshot of the Virtual Reality Model

The virtual Reality (VR) parameter window of the ball and plate system is given in Appendix B1.

### 3.6 Development of the Simulation Environment in MATLAB Simulink

The parameters were simulated in MATLAB Simulink, inner loop controller was first developed, then the outer loop controller was developed next, as discussed in subsection 2.2.1.1, then the virtual reality model block was incorporated together with the Simulink block of the controllers.

#### 3.6.1 Development of the Inner Loop Controller

The parameters of the inner loop after solving the set of Diophantine equation, by linear algebraic method were modelled in the Simulink environment as shown in Figure 3.3.

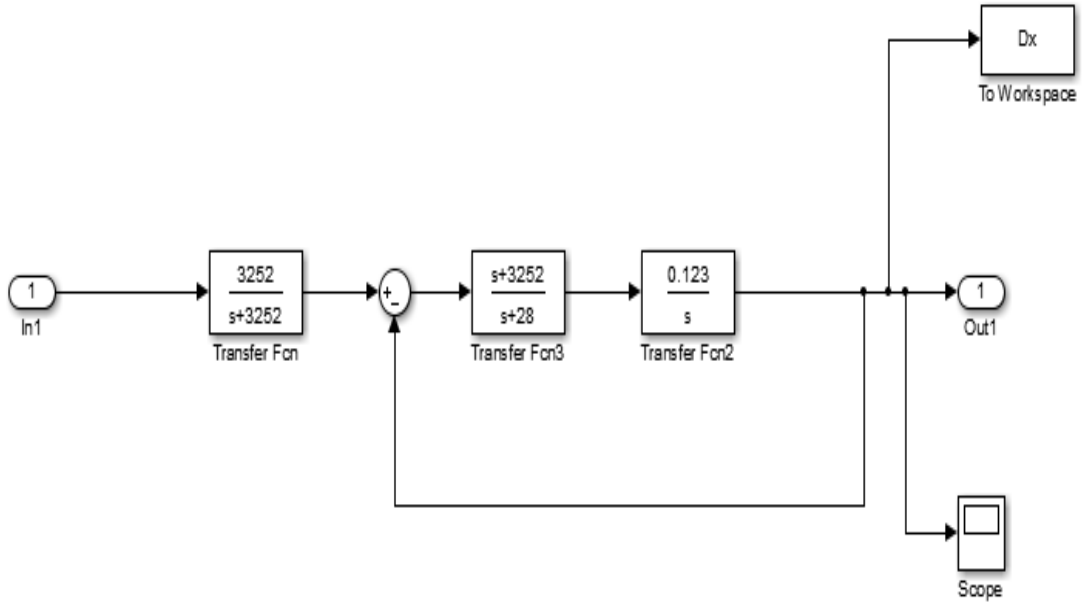



Figure 3.3: Simulink Model of the Two-Port Parameter

### 3.6.2 Development of the Outer Loop Controller

The state space equation of the $H_\infty$ controller derived from the analysis of the mixed sensitivity problem was modelled in the Simulink environment as shown in Figure 3.4.

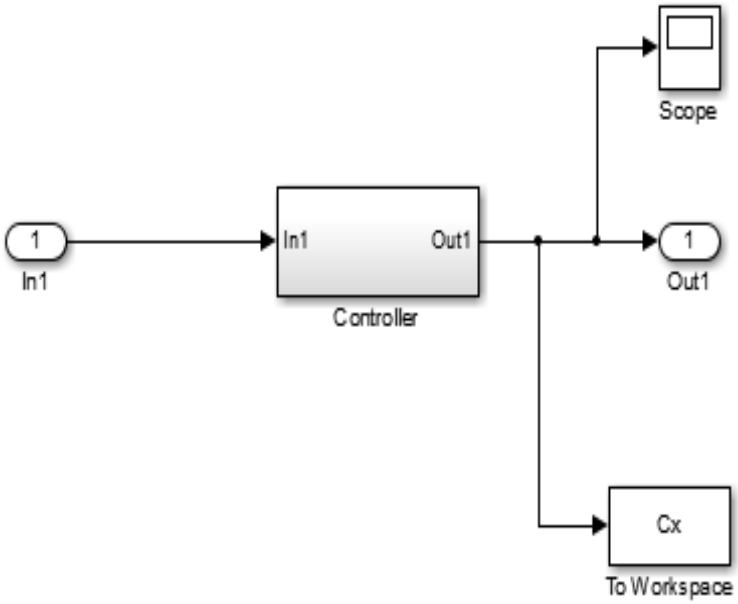



Figure 3.4: Simulink model of the Outer loop

### 3.6.3 Development of the Ball Dynamics of the Ball and Plate System

The dynamics of the ball and plate system was modelled in the Simulink environment based on equation (2.1) as shown in Figure 3.5.

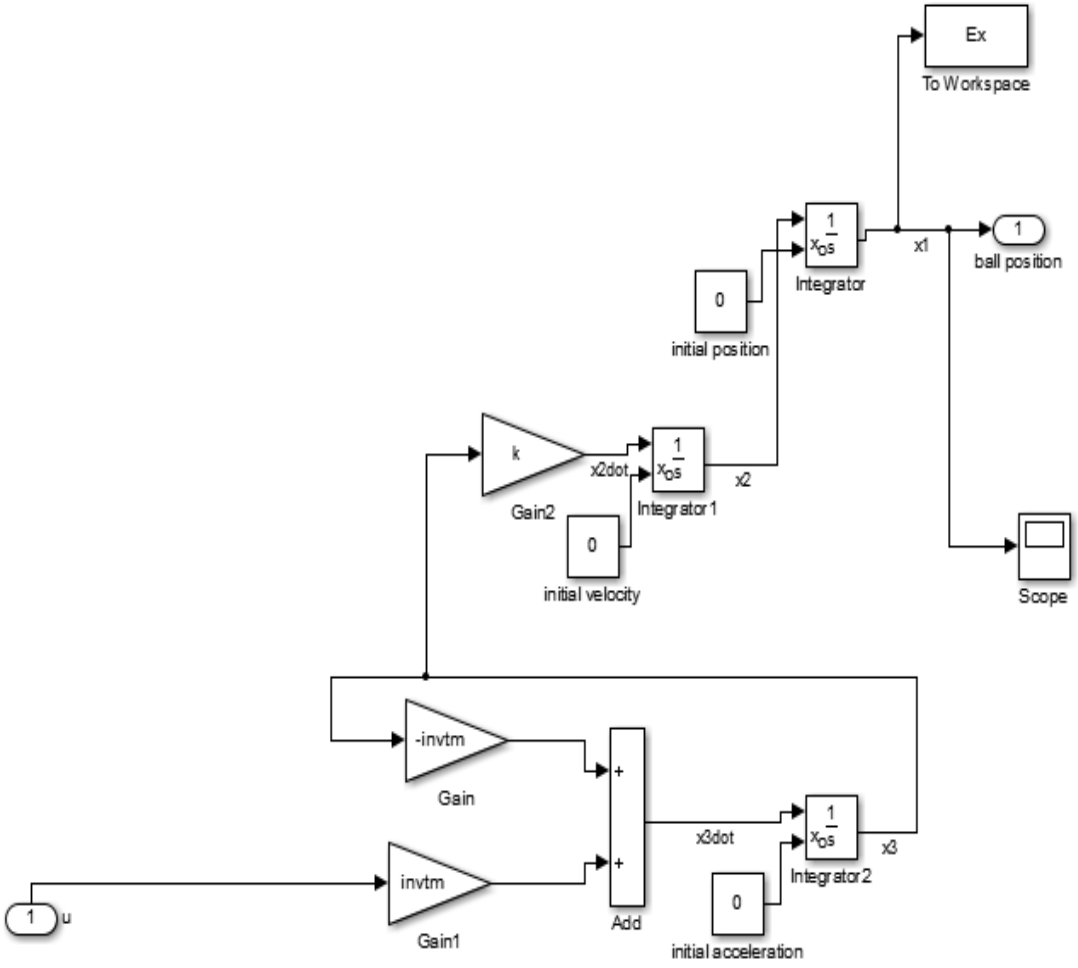



Figure 3.5: Simulink Model of the Ball Dynamics of the Ball and Plate System

The complete Simulink model of the ball and plate system is shown in Figure 3.6.

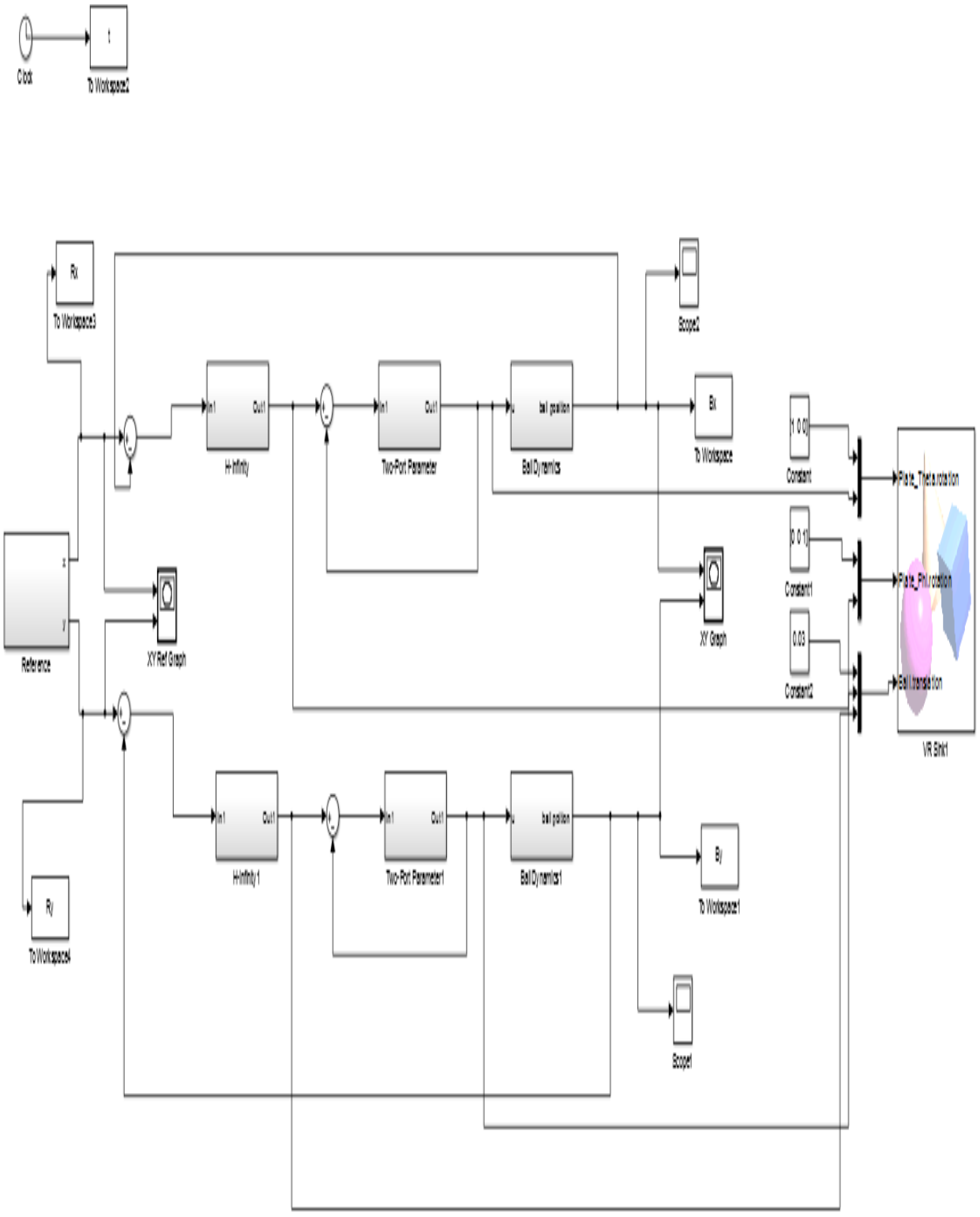



Figure 3.6: Complete Simulink Model of the Ball and Plate System

### 3.6.4 Development of the Reference Signal for the Trajectory Tracking

A circular trajectory of radius 0.4 m, and a sinusoidal reference signal with a reference input of $x = 0.4(1-\cos\omega t)$ and $y = 0.4(\sin\omega t)$ was taken into consideration, and was used to demonstrate the trajectory tracking performance of the ball. The angular frequency of the sinusoidal reference signal used was 1.57 rad/sec.

### 3.7 Graphical User Interface (GUI) of the Ball and Plate System

The graphical user interface (GUI) of the ball and plate system was developed in MATLAB environment, the GUI was used to visualize the path trajectory tracking of the ball and plate system. The process in which the GUI was developed is shown in Figure 3.7.

```
% Hint: get(hObject,'Value') returns toggle state of grid


% --- Executes on button press in Plot.
function Plot_Callback(hObject, eventdata, handles)
% hObject    handle to Plot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Rx = str2double(get(handles.Rx,'string'));
Bx = str2double(get(handles.Bx,'string'));
sim('work_1_new');
axes(handles.axes1);
plot(Rx,Ry,'r',Bx,By,'b');
xlim([-0.5 2]);
ylim([-0.5 0.5]);
legend('Reference Path','Ball Position')
xlabel('X Ball Position (m)');
ylabel('Y Ball Position (m)');
grid on
```

Figure 3.7: Snippet of the Graphical User Interface Code

The complete MATLAB code GUI is given in Appendix C1.

### 3.8 Performance Evaluation

The performance of the developed path trajectory tracking of the ball and plate system using H-infinity controller was evaluated using the performance metrics listed in subsection 2.2.3.2 and 2.2.5.1

#### 3.8.1 Trajectory Tracking Error

The ball tries to follow a specified path as defined by the type of signal that was sent to the reference signal, as such the difference between the desired signal and the reference signal as defined by equation (2.27) which is termed the trajectory tracking error.

#### 3.8.2 Transient Response

As the ball tries to track the reference signal, the transient response of the controller that controls the ball's movement on the plate as discussed in subsection 2.2.5.1 was considered, in particular the settling time and the overshoot of the controller.

### 3.9 Comparison of Results

The trajectory tracking performance of the system using the results of the controller was compared with the single loop configuration of Ghiasi and Jafari (2012) and the reported work done with fuzzy sliding mode controller based on double loop configuration of Negash and Singh (2015).

# CHAPTER FOUR

# RESULTS AND DISCUSSION

## 4.1 Introduction

In this chapter, results and discussion of the theoretical concepts from the previous two chapters are presented. These include the performance of the inner loop control design, the outer loop control design, the virtual reality (VR) model, the trajectory tracking of the ball on the plate, and the graphical user interface design of the trajectory tracking of the ball and plate system.

## 4.2 Result of the Controllability and Observability Test on the System

From the controllability and observability test on the system, it showed that a control solution exist on the system, and getting an optimal solution can be achieved since it is realizable and detectable.

```
ans =
System is Controlable

CONTR_RANK =
      4

ans =
System is Observable

OBSV_RANK =
      4

ans =
System is Observable
```

## 4.3 Result of the Actuator Parameter

The effect of the pole at $s = -895.89$ on the transient is negligible. Therefore, the transfer function can be approximated by a first order differential equation. The impact of the pole can be safely neglected, but its effect on the steady state performance of the system must be retained in order to have a model that exhibits the properties of the actuator.

### 4.4 Result of Two-Port Parameter Configuration

The step response of the actuator is given in Figure 4.1.

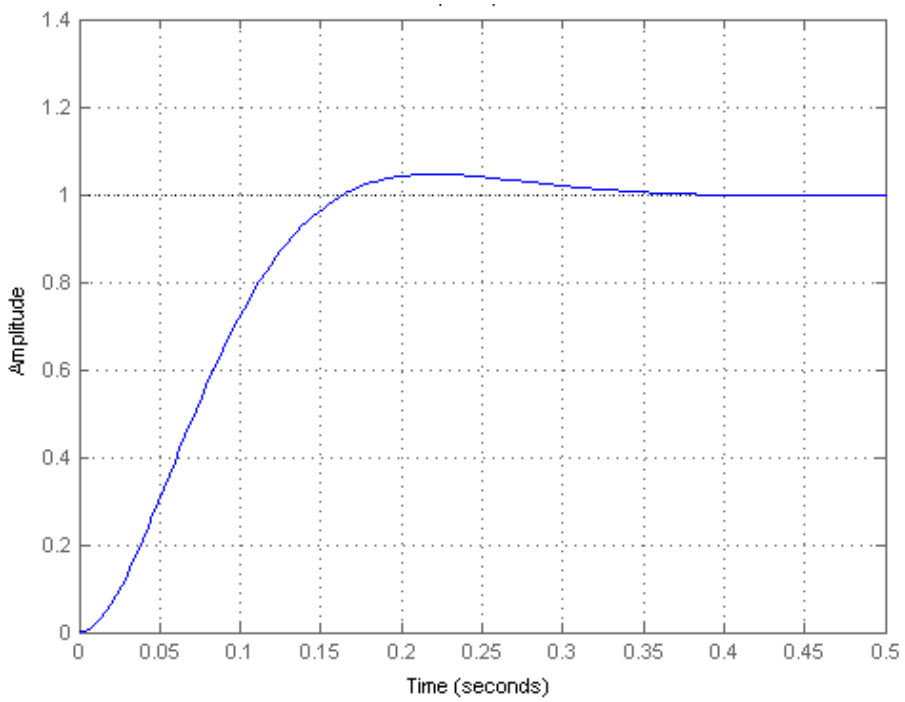



Figure 4.1: Step Response of the Actuator

Table 4.1 shows the properties of the actuator.

Table 4.1: Properties of the Actuator

| System Response | Value |
|---|---|
| Settling Time (sec) | 0.2989 |
| Overshoot (%) | 4.5986 |

From Table 4.1, the settling time of the actuator is 0.2989 seconds, which showed that the plate will settle before 0.4 seconds that was set for it. However, the actuator $U(s)$ due to a step input should not exceed the rated DC motor voltage, which is 75V.

Figure 4.2 shows the step response of the actuator with open parameters (rated power and voltage) of the DC motor.

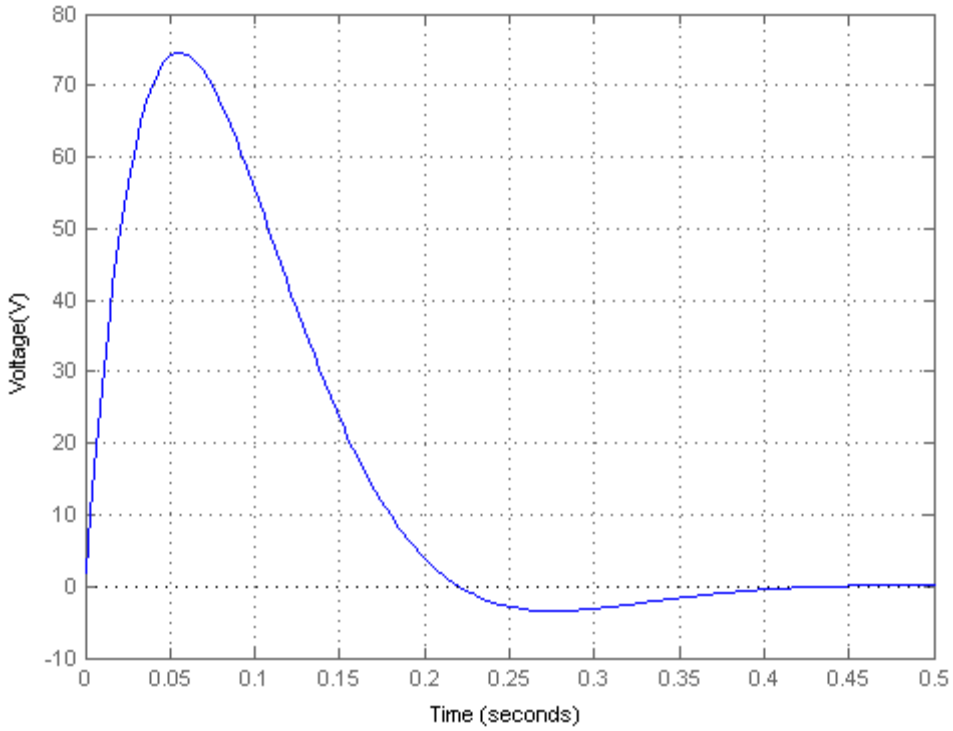



Figure 4.2: Step Response of the Actuator with Open Parameters

Table 4.2 shows the properties of the actuator with open parameters

Table 4.2: Properties of the Actuator with Open Parameters

| **Actuator System Response** | **Value** |
|---|---|
| Settling Time (sec) | 0.3546 |
| Peak Voltage (V) | 74.5631 |

From Table 4.2, the plate stabilized at 0.3546 seconds. Also, the peak voltage is 74.56V, which is closer to the rated voltage of 75V. From this, it showed that the proper design of the inner loop DC actuator has the following properties given in Table 4.2.

From the chosen DC motor RFS-20-6012, the actuator provided a gain of $2.49x10^{-4}$, the required gain $\frac{0.123}{2.49x10^{-4}} \approx 494$ for the system was obtained approximately using a preamplifier.

### 4.5 Result of the $H_\infty$ Controller

The function $W_1(s)$ is the bound for the sensitivity function $S(s)$, and it reflects to external disturbance rejection, a small-steady state error with respect to the desired equilibrium, and plant variations tolerance i.e. the performance tolerance, $W_2(s)$ reflects the control input constraints. $W_3(s)$ reflects the peak requirement on the complementary sensitivity function $T(s)$ to reduce the frequency oscillations, and to limit $T(s)$ in the frequency range for stability.

Iteration number nine (9) from Table 3.4 was displayed as the best iteration under the tolerance: 0.0100, which is 0.47266 ($\gamma = 0.47266$), which is less than 1, which satisfied the H-infinity design criterion, and the designed optimal H-infinity controller based on the selected weighting function $W_1(s)$, $W_2(s)$ and $W_3(s)$ is:

$$\mathrm{Gc} = \frac{23622(\mathrm{s}+1.631)(\mathrm{s}+0.2213)(\mathrm{s}+0.1794)}{(\mathrm{s}+1109)(\mathrm{s}+25.82)(\mathrm{s}+0.9012)(\mathrm{s}+0.3308)} \tag{4.1}$$

The step response of the H-infinity controller is given in Figure 4.3.

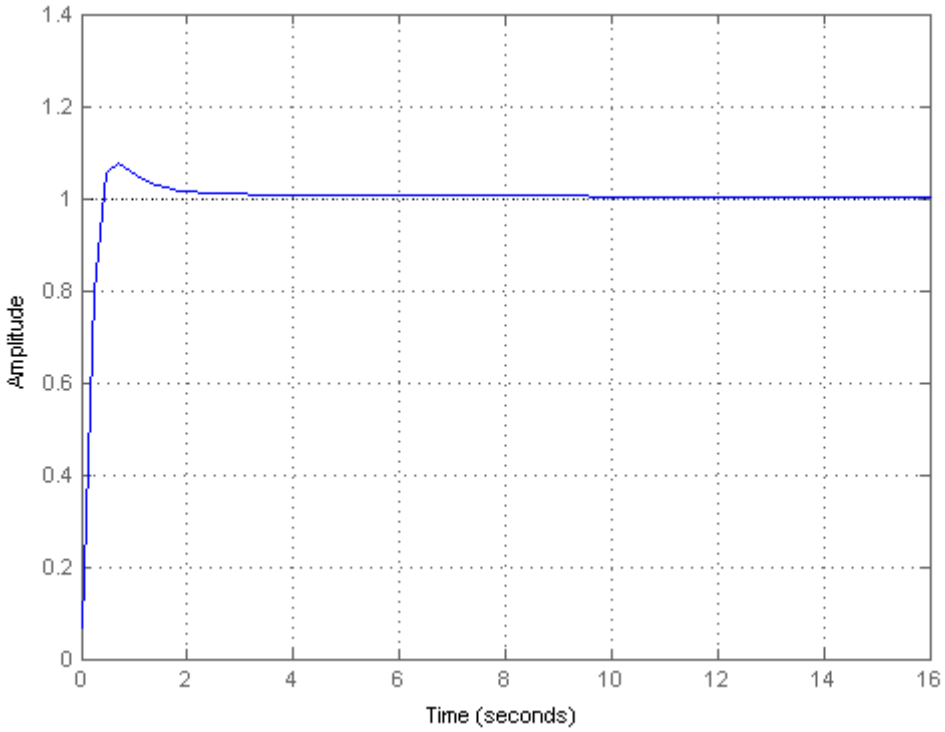



Figure 4.3: Step Response of the H-infinity Controller

From Figure 4.3, the properties of the H-infinity controller is given in Table 4.3.

Table 4.3: Properties of the H-infinity Controller

| **H-infinity Controller System Response** | **Value** |
|---|---|
| Settling Time (sec) | 1.7087 |
| Overshoot (%) | 7.7246 |

From Table 4.3, it shows that the H-infinity controller stabilized the ball at 1.7087 seconds, with an overshoot of 7.7246 %. This shows a good indication of tracking the ball on the desired path on the plate.

### 4.6 Result of the Virtual Reality (VR) Model

The virtual reality (VR) model of the ball and plate designed on V-Realm builder is shown in Figure 4.4

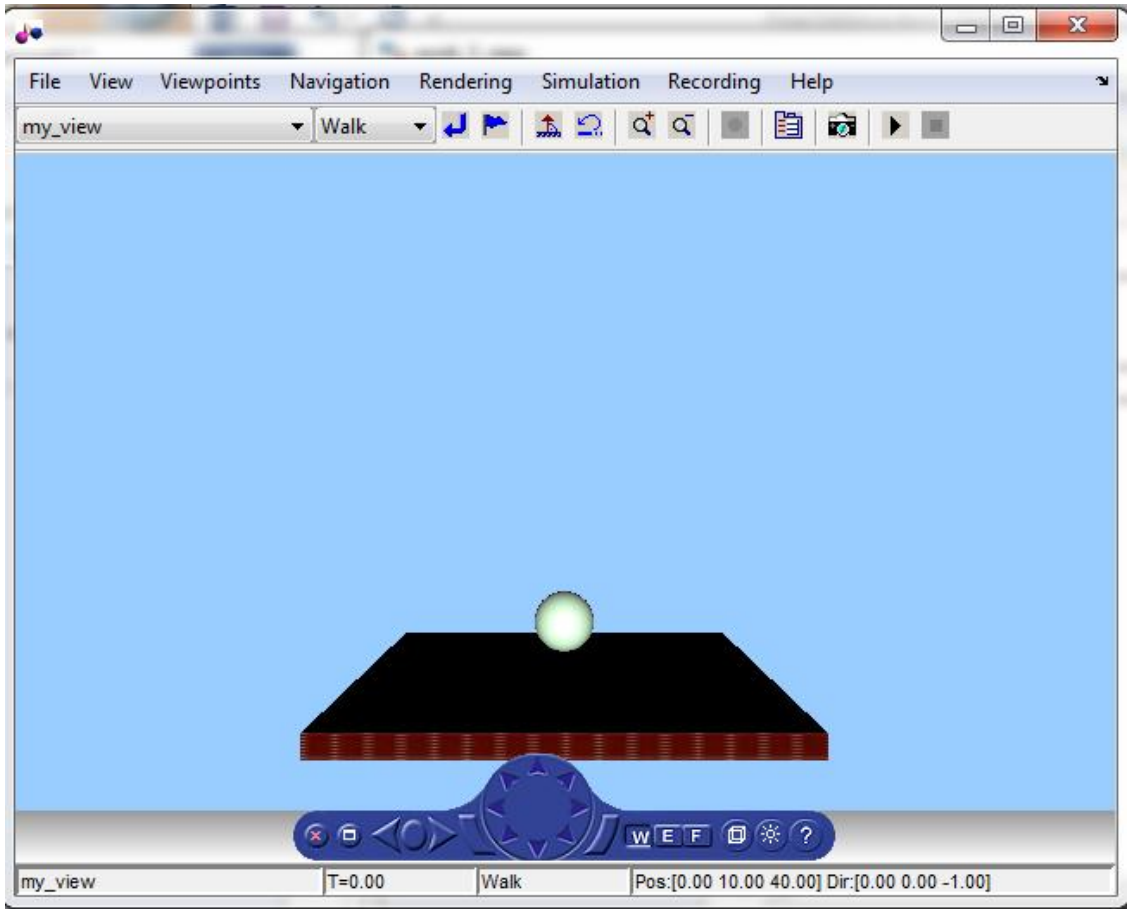



Figure 4.4: Virtual Reality (VR) Model of the Ball and Plate System

From Figure 4.4, the rotation of the plate about the X-axis and Z-axis were controlled by signals from the simulation of the ball and plate Simulink model. Rotation of the plate about the Y-axis was restricted. The translation of the ball was realized along the X-Z plane of the plate as shown in Figure 2.13, while no translation was allowed on the Y-axis.

However, since the available signals for the simulation were only for 2-D plots (as the system was decomposed into X-axis and Y-axis), the position and angle data of the model were transformed into a 3-D dimensional coordinates so as to display the virtual reality of the ball and plate system. In doing so, the differences in the coordinate system convention between virtual reality modelling language and MATLAB as shown in Figure 2.13 was considered.

### 4.7 Result of the Trajectory Tracking of the Ball and Plate System

The plot of the trajectory tracking of X-position with respect to Y-position is shown in Figure 4.5.

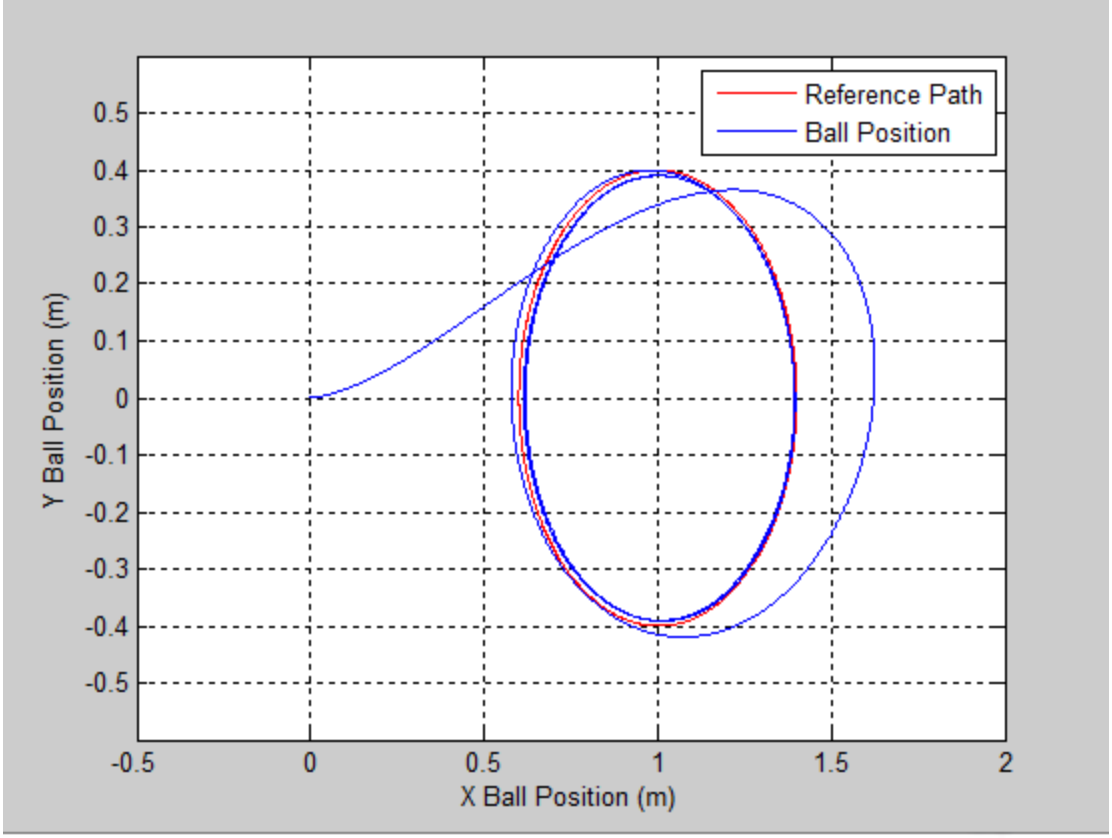



Figure 4.5: Circular Trajectory Tracking Performance with H-infinity Controller

The ball was allowed to track a circular trajectory at a frequency of 0.52 rad/sec at a complete revolution of 12 seconds. When the speed was increased to 0.9 rad/sec, at a complete revolution of 7 seconds, the trajectory tracking error increased.

However, it was observed that using the H-infinity controller, the steady state tracking error of the ball is 0.0095 m. This shows that the ball was able to track the reference signal with a trajectory tracking error of 0.0095 m.

### 4.8 Result of the Graphical User Interface (GUI) for the Circular Trajectory Tracking

The graphical user interface (GUI) for the circular trajectory tracking of the ball and plate system is shown in Figure 4.6.

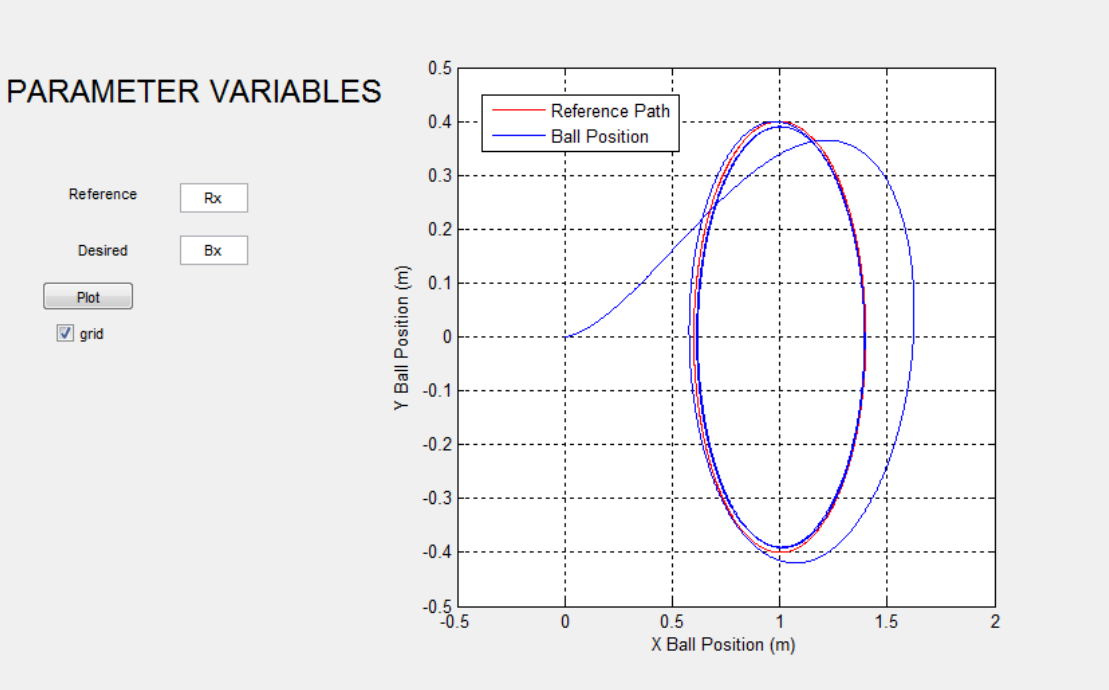



Figure 4.6: Circular Trajectory Tracking of the Ball using H-infinity Controller

From the GUI, the trajectory tracking of the ball on the plate was displayed on the figure interface of the GUI for proper visualization of the path trajectory tracking of the ball on the plate.

### 4.9 Result of the Circular Trajectory Tracking Using H-infinity Controller Considering the Ball and Plate System as a Single Loop System

The circular trajectory tracking proposed by Ghiasi and Jafari (2012) considering the system as a single loop system is shown in Figure 4.7.

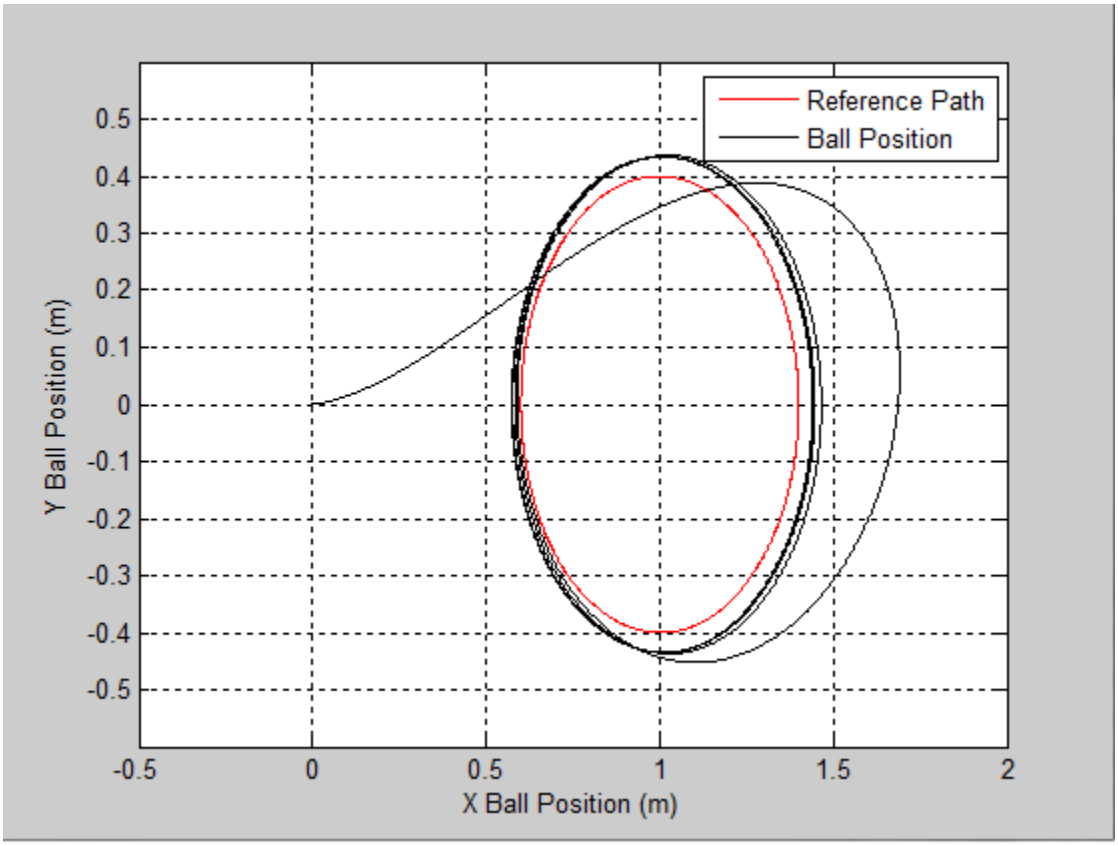



Figure 4.7: Circular Trajectory Tracking of the Ball Considering the System as a Single Loop Controller

The ball was allowed to track a circular trajectory at a frequency of 0.33 rad/sec at a complete revolution of 19 seconds. It is observed that using the H-infinity controller as a single loop system configuration, the steady state tracking error of the ball is 0.0337 m. This shows that the ball was able to track the reference signal, but with a higher tracking error of 0.0337 m.

### 4.10 Comparison of the Results

The results of the system was compared with single loop H-infinity (SLH) controller, as proposed by Ghiasi and Jafari (2012), based on settling time, overshoot and trajectory tracking error of the system, and also with the reported work of Negash and Singh (2015) that used fuzzy sliding mode controller based on settling time and trajectory tracking error of the system.

### 4.10.1 Comparison of the controllers based on the Step Response Performance Index

The developed double loop configuration was compared with that of single loop H-infinity (SLH) controller configuration based on settling time, percentage overshoot and trajectory tracking error. Table 4.4 shows the comparison of the result.

Table 4.4: Comparison of the developed controller and that of Ghiasi and Jafari (2012)

| Performance Index | Developed | SLH |
|---|---|---|
| **Settling Time (sec)** | 1.7087 | 4.3254 |
| **Percentage Overshoot (%)** | 7.7246 | 9.4302 |
| **Trajectory Tracking Error (m)** | 0.0095 | 0.0337 |

From Table 4.4, it can be observed that the developed controller has achieved a better settling time of 60.5%, a percentage overshoot of 18.1% and trajectory tracking error of 71.8% over the single loop H-infinity controller, this shows that with the use of the double loop configuration for the design of the ball and plate system, the errors that will result in an increase in terms settling time, percentage overshoot and trajectory tracking error was reduced significantly. This in turn improves the trajectory tracking performance of the ball on the plate.

### 4.10.2 Comparison of the Developed Controller with that of Negash and Singh (2015)

The developed controller was compared with that of Negash and Singh (2015) (N&S), based on settling time and trajectory tracking error. Table 4.5 shows the comparison.

Table 4.5: Comparison of the developed controller and that of Negash and Singh (2015) (N&S)

| Performance Index | Developed | (N&S) |
|---|---|---|
| **Settling Time (sec)** | 1.7087 | 3.5000 |
| **Trajectory Tracking Error (m)** | 0.0095 | 0.0200 |

From Table 4.5, it is also observed that the developed controller has achieved a better settling time of 51.2%, and a trajectory tracking error of 52.5% over the fuzzy sliding mode controller proposed by Negash and Singh (2015). This shows that H-infinity controller is a more robust controller than the fuzzy sliding mode controller, which in turn gives a better settling time and trajectory tracking error of the ball on the plate, thus improving the trajectory tracking performance of the ball on the plate.

# CHAPTER FIVE

# CONCLUSION AND RECOMMENDATIONS

## 5.1 Conclusion

The position and trajectory tracking of the ball and plate system was developed using the double feedback loop structure. The inner loop was designed using linear algebraic method, while the outer loop was designed using H-infinity mixed sensitivity function. From the analysis, it was observed that the plate was stabilized at 0.3546 seconds. For the outer loop design, it was observed that the ball was able to settle at 1.7087 seconds. The trajectory tracking of the ball was also presented by applying a sinusoidal reference signal to the system, a Virtual Reality (VR) model was developed to show the trajectory tracking in 3D animation. Graphical User Interface (GUI) was also developed to display the trajectory tracking of the system. The trajectory tracking error of the system using H-infinity controller was 0.0095 m. The results showed an improvement in terms of trajectory tracking error of 71.8%, settling time of 60.5%, and an overshoot of 18.1%, when compared with the single loop H-infinity (SLH) controller. Also, when compared with the fuzzy sliding mode controller, it showed an improvement in terms of settling time of 51.2%, and trajectory tracking error of 52.5% of the ball and plate system.

## 5.2 Limitation

The limitation that had contributed to getting a higher trajectory tracking error is:

i) The backlash effect of the DC actuator motor when analysing the inner loop controller was also not considered.

## 5.3 Significant Contributions

The significant contributions of this research are as follows:

i) Development of a MATLAB based Graphical User Interface (GUI) for path trajectory of the ball and plate system.

ii) The developed controllers for the double loop configuration showed an improvement of 71.8% in trajectory tracking error, 60.5% in settling time and 18.1% in overshoot when compared with the single loop configuration of the system and an improvement of 52.5% in trajectory tracking error and 51.2% in settling time when compared with fuzzy sliding mode controller.

### 5.4 Recommendations for Further work

The following possible areas of further work are recommended for consideration for future research:

i) The coupling terms and its nonlinear characteristics which led to the modelling of the system into a single-input-single-output (SISO) system and also its effect to the system can also be considered.

ii) Robust controllers such as $\mu - synthesis$ and $H_2$ controller can be used for outer loop analysis, due to some of the properties of the system like parameter uncertainties, measurement time delay, friction etc.

iii) Path planning and obstacle avoidance analysis for the ball and plate system can also be considered using the same controllers for the inner and outer loop configuration.

## REFERENCES


Andinet, N. H. (2011). *Design of Fuzzy Sliding Mode Controller for the Ball and Plate System*. *(Masters of Science Thesis), Addis Ababa Institute of Technology, Addis Ababa., Published*.

Awtar, S., Bernard, C., Boklund, N., Master, A., Ueda, D., & Craig, K. (2002). Mechatronic design of a ball-on-plate balancing system. *Mechatronics, 12*(2), 217-228.

Bai, M., Lu, H., Su, J., & Tian, Y. (2006). *Motion control of ball and plate system using supervisory fuzzy controller.* Paper presented at the The Sixth World Congress on Intelligent Control and Automation, WCICA 2006.

Borah, M., Majhi, L., Roy, P., & Roy, B. (2014). *Design of a Fractional Order PD Controller Tuned by Firefly Algorithm for Stability Control of the Nonlinear Ball and Plate System.* Paper presented at the IEEE International Conference on Advanced Communication Control and Computing Technologies, Ramanathapuram, India.

Burns, R. S. (2001). *Advanced Control Engineering* (Vol. 3). Linacre House, Jordn Hill, Oxford OX2 8DP 225 Wildwood Avenue, Woborn, MA 01801-2041: Butterworth-Heinemann.

Chen, C.-T. (1995). *Analog and Digital Control System Design: Transfer-function, State-space, and Algebraic Methods*. New York, USA: Oxford University Press, Inc.

Cheng, C.-C., & Chou, C.-C. (2016). *Fuzzy-Based Visual Servo with Path Planning for a Ball-Plate System. International Symposium on Intelligent Computing Systems (ISICS)*, 97-107. doi: 10.1007/978-3-319-30447-2_8

Cheng, C.-C., & Tsai, C.-H. (2016). Visual Servo Control for Balancing a Ball-Plate System. *International Journal of Mechanical Engineering and Robotics Research, 5*(1), 28.

Dingyu, X., YangQuan Chen, & Derek P., A. (2007). *Linear Feedback Control; Analysis and Design with MATLAB* (Vol. 3 ). 3600 Market Street, 6th floor, Philadelphia, PA 19104-2688, USA: Society for Industrial and Applied Mathematics.

Dong, X., Zhang, Z., & Chen, C. (2009). *Applying genetic algorithm to on-line updated PID neural network controllers for ball and plate system.* Paper presented at the Fourth International Conference on Innovative Computing, Information and Control (ICICIC), 2009.

Dong, X., Zhao, Y., Xu, Y., & Zhang, Z. (2011). Design of PSO fuzzy neural network control for ball and plate system. *International Journal of Innovative Computing, Information and Control, 7*(12), 7091-7103.

Duan, H., Tian, Y., & Wang, G. (2009). *Trajectory tracking control of ball and plate system based on auto-disturbance rejection controller.* Paper presented at the 7th Asian Control Conference, ASCC 2009.

Fan, X., Zhang, N., & Teng, S. (2004). Trajectory planning and tracking of ball and plate system using hierarchical fuzzy control scheme. *Fuzzy Sets and Systems, 144*(2), 297-312.

Friedland, B. (2012). *Control System Design: An Introduction to State-Space Methods* (2 ed.). Dover Publications, Inc. 31 East 2nd Street Mineola, New York: Courier Corporation.

Galvan-Colmenares, S., Moreno-Armendáriz, M. A., Rubio, J. d. J., Ortíz-Rodriguez, F., Yu, W., & Aguilar-Ibáñez, C. F. (2014). Dual PD Control Regulation with

Nonlinear Compensation for a Ball and Plate System. *Mathematical Problems in Engineering, 2014*, 1-10. doi: 10.1155

Ghiasi, A. R., & Jafari, H. (2012). *Optimal Robust Controller Design for the Ball and Plate System.* Paper presented at the The 9th International Conference on Electronics Computer and Computation ICECCO-2012.

Golnaraghi, F., & Kuo, B. C. (2010). *Automatic Control Systems* (9th ed. Vol. 2): John Wiley & Sons, Inc.

Hongrui, W., Yantao, T., Siyan, F., & Zhen, S. (2008). *Nonlinear control for output regulation of ball and plate system.* Paper presented at the 27th Chinese Control Conference, CCC 2008.

Hossain, S. (2007). Design of a robust controller for a magnetic levitation system. *Wichita State University*.

HUMUSOFT Ltd. (2012). User's Manual for CE151 Ball and Plate Apparatus.

Hunt, B. R., Lipsman, R. L., & Rosenberg, J. M. (2014). A guide to MATLAB: for beginners and experienced users.

Jafari, H., Rahimpour, A., Pourrahim, M., & Hashemzadeh, F. (2012). *Linear Quadratic Gaussian Control for ball and plate system.* Paper presented at the 2012 International Conference on Computer, Control, Education and Management (CCEM 2012), Dubai, United Arab Emirates.

Jarzebowska, E. (2012). *Model-Based Tracking Control of Nonlinear Systems*. Taylor & Francis Group, 6000 Broken Parkway NW, Suite 300, Boca Raton , FL 33487-2742: CRC Press.

Kassem, A., Haddad, H., & Albitar, C. (2015). Commparison Between Different Methods of Control of Ball and Plate System with 6DOF Stewart Platform. *IFAC-PapersOnLine, 48*(11), 47-52.

Khalil, H. K., & Grizzle, J. (1996). *Nonlinear systems* (Vol. 3): Prentice hall New Jersey.

Knuplež, A., Chowdhury, A., & Svečko, R. (2003). *Modeling and Control design for the ball and plate system.* Paper presented at the Proceedings of the IEEE International Conference on Industrial Technology, (ICIT'03), Maribor, Slovania.

Lin, C. E., Liou, M.-C., & Lee, C.-M. (2014). Image Fuzzy Control on Magnetic Suspension Ball and Plate System. *International Journal of Automation and Control Engineering*.

Lin, H.-Q., Cui, S.-G., Geng, L.-H., & Zhang, Y.-L. (2014). *H∞ controller design for a ball and plate system using normalized coprime factors.* Paper presented at the 26th Chinese Control and Decision Conference (CCDC), 2014.

Liu, H., & Liang, Y. (2010). *Trajectory tracking sliding mode control of ball and plate system.* Paper presented at the 2nd International Asia Conference on Informatics in Control, Automation and Robotics (IACICAR), 2010.

LLC, H. D. (November 2016). Harmonic Drive DC Servo Actuators. *http://harmonicdrive.net/media/support/catalogs/pdf/rh-rhs-rfs-dc-servo-catalog-0905.pdf*.

MathWorks. (2015). The MathWorks, Inc. Simulink® 3D Animation™ User's Guide.

Moarref, M., Saadat, M., & Vossoughi, G. (2008). *Mechatronic Design and Position of a Novel Ball and Plate System.* Paper presented at the 16th Mediterranean Conference on Control and Automation Congress Centre, Ajaccio, France.

Mochizuki, S., & Ichihara, H. (2013, July 17-19). *I-PD Controller Design based on Generalized KYP Lemma for Ball and Plate system.* Paper presented at the European Control Conference (ECC), 2013 Zurich, Switzerland.

Mohajerin, N., Menhaj, M. B., & Doustmohammadi, A. (2010). *A reinforcement learning fuzzy controller for the ball and plate system.* Paper presented at the IEEE International Conference on Fuzzy Systems (FUZZ), 2010.

Moreno-Armendáriz, M. A., Pérez-Olvera, C. A., Rodríguez, F. O., & Rubio, E. (2010). Indirect hierarchical FCMAC control for the ball and plate system. *Neurocomputing, 73*(13-15), 2454-2463. doi: 10.1016/j.neucom.2010.03.023

Mukherjee, R., Minor, M. A., & Pukrushpan, J. T. (2002). Motion Planning for a Spherical Mobile Robot: Revisiting the Classical Ball-Plate Problem. *Journal of Dynamic Systems Measurement and Control, 124*, 502-511. doi: 10.1115/1.1513177

Negash, A., & Singh, N. P. (2015). *Position Control and Tracking of Ball and Plate System Using Fuzzy Sliding Mode Controller.* Paper presented at the Afro-European Conference for Industrial Advancement.

Ogata, K. (2002). *Modern Control Engineering* (Vol. 4). Prentice-Hall, Inc. Upper Saddle River, New Jersey 07458: Pearson Education International.

Oravec, M., & Jadlovská, A. (2014a). *Design of Virtual Models of Mechatronics Systems with Simulink 3D Animation Toolbox.* Paper presented at the In: Proceedings of the international Scientific Conference-Technical Computing, Bratislava, Slovakia.

Oravec, M., & Jadlovská, A. (2014b). *Optimal control of the mechatronicalal laboratory model B&P_KYB.* Paper presented at the Proceeding of the Faculty of Electrical Engineering and Informatics of the Technical University of Kosice, Košice.

Oravec, M., & Jadlovská, A. (2015). *Model Predictive Control of a Ball and Plate laboratory model.* Paper presented at the IEEE 13th International Symposium on Applied Machine Intelligence and Informatics (SAMI), 2015 Slovakia.

Oravec, M., Jadlovská, A., & Novisedláková, D. (2015). *Design of Adaptive Control Algorithm of Mechatronic System Ball & Plate.* Paper presented at the Proceeding of the Faculty of Electrical Engineering and Informatics of the Technical University of Kosice.

Oriolo, G., & Vendittelli, M. (2005). A Framework for the Stabilization of General Nonholonomic Systems With an Application to the Plate-Ball Mechanism. *IEEE Transactions on Robotics, 21*(2), 162-175.

Patrick, M., & Thomas, H. O. (2003). *Graphics and GUIs with MATLAB* (3 ed.). CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431: Chapman & Hall CRC Press Company, London.

Pattanapong, Y., & Deelertpaiboon, C. (2013). *Ball and plate position control based on fuzzy logic with adaptive integral control action.* Paper presented at the IEEE International Conference on Mechatronics and Automation (ICMA), 2013.

Rastin, M. A., Moosavian, S., Talebzadeh, E., & Alaeddin, M. (2013). *Trajectory tracking and obstacle avoidance of a ball and plate system using fuzzy theory.* Paper presented at the 13th Iranian Conference on Fuzzy Systems (IFSC), 2013.

Roy, P., Kar, B., & Hussain, I. (2015). *Trajectory Control of a Ball in a Ball and Plate System Using Cascaded PD Controllers Tuned by PSO.* Paper presented at the Proceedings of Fourth International Conference on Soft Computing for Problem Solving.

Singh, N., & Yadav, S. K. (2012). Comparison of LQR and PD controller for stabilizing Double Inverted Pendulum System. *International Journal of Engineering Research and Development, 1*(12), 69-74

Wang, H., Tian, Y., Ding, C., Gu, Q., & Guo, F. (2008). *Output regulation of the ball and plate system with a nonlinear velocity observer.* Paper presented at the 7th World Congress on Intelligent Control and Automation, WCICA 2008.

Wang, H., Tian, Y., Sui, Z., Zhang, X., & Ding, C. (2007). Tracking Control of Ball and Plate System with a Double Feedback Loop Structure. *International Conference on mechatronics and Automation*.

Wang, Y., Li, X., Li, Y., & Zhao, B. (2012). *Identification of ball and plate system using multiple neural network models.* Paper presented at the International Conference on System Science and Engineering (ICSSE), 2012.

Wang, Y., Sun, M., Wang, Z., Liu, Z., & Chen, Z. (2014). A novel disturbance-observer based friction compensation scheme for ball and plate system. *ISA Trans, 53*(2), 671-678.

Xiao, J., & Buttazzo, G. (2016). *Adaptive Embedded Control for a Ball and Plate System.* Paper presented at the ADAPTIVE 2016: The Eighth International Conference on Adaptive and Self-Adaptive Systems and Applications.

Yuan, D.-h. (2009). *Pneumatic servo ball & plate system based on touch screen and oscillating cylinder.* Paper presented at the International Workshop on Intelligent Systems and Applications, (ISA) 2009.

Zheng, F., Li, X., Wang, S., & Ding, D. (2011). *Position control of ball and plate system based on switching mechanism.* Paper presented at the IEEE International Conference on Automation and Logistics (ICAL), 2011.

# APPENDICES

## APPENDIX A1

## MATLAB CODE OF THE ACTUATOR

```
clc
clear all
num=[0 0 400];
den=[1 28 400];
G=tf(num,den);
step(G)
grid on
stepinfo(G)
```

## APPENDIX A2

## MATLAB CODE OF THE ACTUATOR WITH OPEN PARAMETERS

```
clc
clear all
num=[0 400 0];
den=[0.123 3.444 49.2];
G=tf(num,den);
step(G)
ylabel('Voltage(V)')
grid on
stepinfo(G)
```

## APPENDIX A3

## MATLAB CODE OF THE H-INFINITY FUNCTION

```
clc
k=4.803;invtm=5.35;
s=tf('s');
G=7.007/s^2;
[a,b,c,d]=ssdata(ss(G));
p1=-0.1;
a1=a-p1*eye(size(a));
G_shift=ss(a1,b,c,d);
beta=100;alpha=1.5;w1c=4.3;zeta1=1.2;zeta2=1.2;
W1=tf((beta*((alpha*s^2)+(2*zeta1*w1c*(sqrt(alpha))*s+(w1c*w1c))))/((b
eta*s^2)+(2*zeta2*w1c*(sqrt(beta))*s+(w1c*w1c))));
W2=[];
W3=(s^2)/100;
TSS_shift=augtf(G_shift,W1,W2,W3);
[gg,ss_Fopt_shift]=hinfopt(TSS_shift);
[a2,bf,cf,df]=branch(ss_Fopt_shift);
af=a2+p1*eye(size(a2));
Gc_=ss(af,bf,cf,df);
Gc=zpk(Gc_);
G_o=G*Gc;
G_c=feedback(G_o,1);
step(G_c)
grid on
figure
```

```
nichols(G_o)
grid
axis([-360,0,-40,40])
```

# APPENDIX B1

## VIRTUAL REALITY (VR) PARAMETER WINDOW OF THE BALL AND PLATE SYSTEM

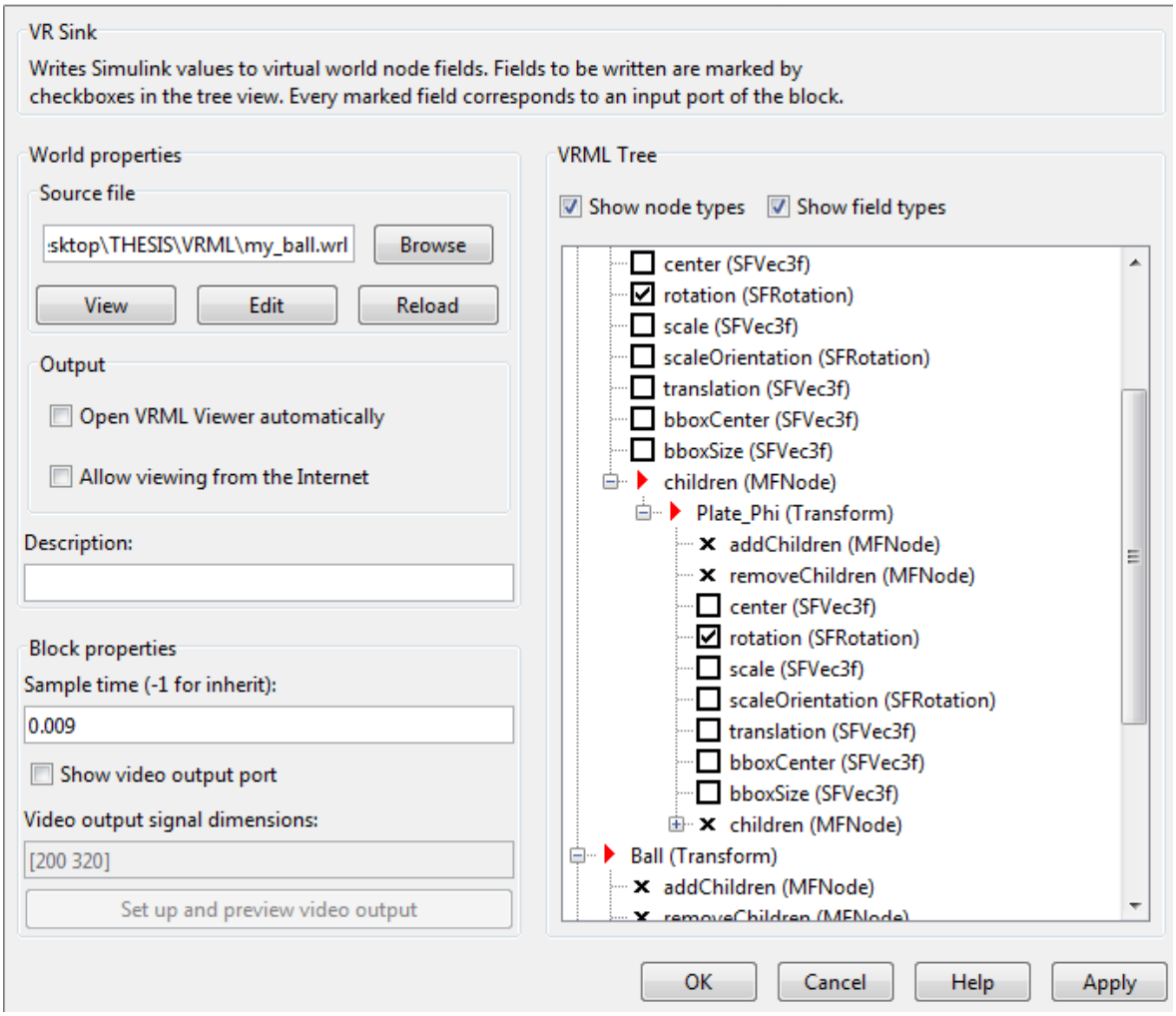



**Figure B1: Virtual Reality (VR) Parameter Window of the Ball and Plate System**

## APPENDIX C1

## MATLAB CODE OF THE GRAPHICAL USER INTERFACE (GUI) OF THE CIRCULAR TRAJECTORY TRACKING USING H-INFINITY CONTROLLER

```
function varargout = plot_tr(varargin)
% PLOT_TR MATLAB code for plot_tr.fig
%      PLOT_TR, by itself, creates a new PLOT_TR or raises the
existing
%      singleton*.
%
%      H = PLOT_TR returns the handle to a new PLOT_TR or the handle
to
%      the existing singleton*.
%
%      PLOT_TR('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in PLOT_TR.M with the given input
arguments.
%
%      PLOT_TR('Property','Value',...) creates a new PLOT_TR or raises
the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before plot_tr_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to plot_tr_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help plot_tr

% Last Modified by GUIDE v2.5 15-Oct-2016 19:03:59

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @plot_tr_OpeningFcn, ...
                   'gui_OutputFcn',  @plot_tr_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before plot_tr is made visible.
```

```
function plot_tr_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to plot_tr (see VARARGIN)

% Choose default command line output for plot_tr
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes plot_tr wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = plot_tr_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function Rx_Callback(hObject, eventdata, handles)
% hObject    handle to Rx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Rx =str2double(get(handles.Rx,'string'));

% Hints: get(hObject,'String') returns contents of Rx as text
%        str2double(get(hObject,'String')) returns contents of Rx as a
double


% --- Executes during object creation, after setting all properties.
function Rx_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Rx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function Bx_Callback(hObject, eventdata, handles)
% hObject    handle to Bx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Bx as text
%        str2double(get(hObject,'String')) returns contents of Bx as a
double
Bx = str2double(get(handles.Bx,'string'));


% --- Executes during object creation, after setting all properties.
function Bx_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Bx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in grid.
function grid_Callback(hObject, eventdata, handles)
% hObject    handle to grid (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of grid


% --- Executes on button press in Plot.
function Plot_Callback(hObject, eventdata, handles)
% hObject    handle to Plot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Rx = str2double(get(handles.Rx,'string'));
Bx = str2double(get(handles.Bx,'string'));
sim('work_1_new');
axes(handles.axes1);
plot(Rx,Ry,'r',Bx,By,'b');
xlim([-0.5 2]);
ylim([-0.5 0.5]);
legend('Reference Path','Ball Position')
xlabel('X Ball Position (m)');
ylabel('Y Ball Position (m)');
grid on
```