Palestine Polytechnic University
Deanship of Graduate Studies and Scientific Research
Master of Mechatronics Engineering

Control of a Ball and Plate System Using Model-Based Controllers

Submitted by:

Firas "Mohammed Jawdi" Al-haddad

Thesis submitted in partial fulfillment of requirements of the degree Master of Science in Mechatronics Engineering
12, 2020

The undersigned hereby certify that they have read, examined and recommended to the Deanship of Graduate Studies and Scientific Research at Palestine Polytechnic University the approval of a thesis entitled: **Control of a Ball and Plate System Using Model-Based Controllers**, submitted by **Firas "Mohammed Jawdi" Al-haddad** in partial fulfillment of the requirements for the degree of Master in Mechtronics Engineering.

Graduate Advisory Committee:

Dr. Jasem Tamimi (Supervisor), Palestine Polytechnic University.

Signature:________________________ Date:________________

Dr. Saleh altakrouri (Internal committee member), Palestine Polytechnic University.

Signature:________________________ Date:________________

Dr. Hakam Shehadeh (External committee member), Birzeit University.

Signature:________________________ Date:________________

Thesis Approved

Dr. Murad Abu Sbieh
Dean of Graduate Studies and Scientific Research
Palestine Polytechnic University

Signature:________________________ Date:________________

DECLARATION

I declare that the Master Thesis entitled **”Control of a Ball and Plate System Using Model-Based Controllers”** is my original work, and hereby certify that unless stated, all work contained within this thesis is my own independent research and has not been submitted for the award of any other degree at any institution, except where due acknowledgement is made in the text.

Firas ”Mohammed Jawdi” Al-haddad

Signature:________________________ Date:________________

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for the master degree in Mechatronics Engineering at Palestine Polytechnic University, I agree that the library shall make it available to borrowers under rules of the library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission for extensive quotation from, reproduction, or publication of this thesis may be granted by my main supervisor, or in his absence, by the Dean of Graduate Studies and Scientific Research when, in the opinion of either, the proposed use of the material is for scholarly purposes.

Any coping or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Firas "Mohammed Jawdi" Al-haddad

Signature:________________________ Date:________________

Acknowledgement

A lot of people have contributed to the completion of my thesis. I feel so grateful to all those people who made it possible to prepare this thesis. I do value and appreciate the efforts of my supervisor Dr. Jasem Tamimi, where I had extremely fortunate to have such an advisor and mentor, who gave me the chance to work and research on my own, and at the same time to be guided and supported when needed. He taught me how to deeply think, analyze and perform to reach and achieve my goals. I have faced several obstacles, and without the help and support of my supervisor it wouldn't be possible to overcome and pass them.

I would like to express my warm thanks to all the staff members of the mechanical engineering department who formed a supportive factor in completing my thesis. In addition, the surrounding friends who helped me stay on track and overcome many difficulties have a major part of my gratitude and appreciation.

Not to forget the main reason that kept me standing and pushing me forward toward this success, which represents my family, where none of this would be achieved without them. I do dedicate my dissertation to them. They filled me with love, care and concern all the time.

الملخص

يتواجد نظام إتزان الكرة على اللوح في معظم مختبرات التحكم الجامعية، بحيث يتميز بأنه نظام غير خطي وغير مستقر ومتعدد المتغيرات، وعادة ما يتم إستخدامه للتحقق من استراتيجيات أنظمة التحكم الجديدة التي تتعامل مع الأنظمة اللاخطية. يتكون هذا النظام بشكل أساسي من كرة معدنية و شاشة لمس و محركين. توضع شاشة اللمس على اللوح المعدني الذي يتم إمالته في المحورين المتعامدين بإستخدام المحركين المؤازرين لموازنة الكرة المعدنية التي توضع فوق شاشة اللمس.

في هذه الأطروحة، تم عمل نمذجة للنظام ، ومن ثم تم تصميم وحدات تحكم خطية. تم إجراء اختبارات المحاكاة باستخدام برنامج الماتلاب لمقارنة نتائج طرق التحكم المختلفة. لكي يتم التحكم بالنظام، فقد تم تصميم خمس متحكمات مختلفة وهي : 1- نموذج تحكم تنبئي. 2- المتحكم التناسبي التكاملي. 3- المتحكم المنظم الخطي من الدرجة الثانية. 4- المتحكم المتعقب الخطي من الدرجة الثانية. 5- متحكم ردود الفعل الخاصة بحالات النظام . وقد تم تنفيذ المتحكمات الناتجة باستخدام المتحكم الدقيق (أردوينو أونو).

وهدفت هذه الاطروحة للمقارنة بين أداء وحدات التحكم لموازنة كرة تتدحرج بحرية لكي تثبت في موضع معين على اللوح، أو لتحريكها في مسار معين على اللوحة بأقل وقت إستقرار وأقل خطأ محتمل .

Abstract

A ball and plate system (BPS) is a benchmark system in control engineering. BPS is known to be nonlinear, a multivariable and an unstable system, has been widely used to investigate and demonstrate new control strategies that can deal with nonlinearities. The BPS consists of a metal ball, a plate which can be a resisitive touch screen and two servo motors with a linkage mechanism to move the plate. A resistive touch screen is placed over the plate, a plate is pivoted at its center such that the slope of the plate can be manipulated in two perpendicular directions with two servo motors to tilte the plate. In this thesis, the modeling of our BPS is based on the Euler-Lagrange approach, which is represented in the state space form with plate angles as inputs to the system. Then, the obtained model is linearized to be able to design linear controllers. Matlab and simulink programs are used for simulation tests to evaluate the closed loop system response and to determine the parameters and gains for different controllers. Moreover, the effect of the disturbances in the measurement is analyzed. Five control stratigies are selected for static and dynamic position tracking: model predictive control (MPC), proportional-integral-derivative (PID), state feedback, linear quadratic regulator (LQR) and linear quadratic tracker (LQT) controllers. These controllers have been implemented using the Arduino Uno ATmega328P. Therefore, the aim of this project can be summarized as to compare between the performance of the five different controllers for balancing a freely rolling ball in a specific position or to move it in a circle or square trajectory on the plate with the smallest settling time and the least possible error achieved for the dynamics of the real-time system.

Table of Contents

List of Figures

List of Tables

List of Important Abbreviations

BPS	Ball and plate system
PID	Proportional integral derivative
LQR	Linear quadratic regulator
LQT	Linear quadratic tracker
m	Meter
mm	Millimeter
MIMO	Multi input multi output
SISO	Single input single output
K_D	Derivative gain
K_I	Integral gain
K_P	Proportional gain
e_{ss}	Steady state error
T_s	Settling time
sec.	Second
Kg	Kilogram
rad	Radian
PWM	Pulse width modulation
MPC	Model predictive control

Chapter 1

Introduction

Modern control engineering theory is a sub-field of applied mathematics that is based on using of new control design strategies to improve the system performance as well as the system efficiency [34, 25]. Balancing systems are one of most challenging problems in Control engineering field, which are used for testing a new control design strategies. There are lots of platforms for this inverted pendulum, double and multiple inverted pendulums and ball-beam system [47].

A ball and plate system (BPS) is one of the popular and important systems that is a generalized of the traditional ball and beam benchmark [9, 27]. The BPS has four degrees of freedom in which a ball can roll freely on a rigid platform which its inclination can be manipulated in two independent directions [71]. The rigid plate can consist of a resistive or capacitive touch panel that measures the position of the ball regards to the centre or a predetermined point of the plate. This system is inherently unstable since even a small disturbance causes that the ball will roll far away from the stationary point. In addition, it cannot be restored to an equilibrium or to a predetermined point on the plate without substituting force. Thus, some sort of control is necessary to maintain a balanced ball on a plate. Therefore, BPS is considered a well-suited system for testing designed different control methods for unstable system in real environment. In other words, the control objective is to balance the ball on a plate within a boundary or at the predetermined points. When the ball moves outside the the boundary or a predetermined points, a proposed controller will move the two independent coordinates that are needed to stabilize the ball to its designated location.

Generally, the system is a two-dimensional, multivariable and nonlinear system, which has a dynamic that includes a second order nonlinear differential equations [21, 18]. Since the BPS has a complex dynamics, an accurate nonlinear model must be developed for studying model-based controllers. To obtain the BPS dynamics, a Lagrange -Euler method can be used to derive the nonlinear equations, so the kinetic and potential energies of the system must be found. The kinetic energy is comprised of the energy due to both linear and angular motions in the system. While the potential energies can be computed for the all rigid bodies in the system [61].

1.1 Literature Review

The BPS has been used and studied in several control methods that ranged from linear control method like an optimal controller, and nonlinear control such as sliding mode controller. Some of the recent available researches on the BPS are reviewed in this section.

Proportional integral derivative (PID) controller is the most widely used method to control the BPS. However, it is difficult to obtain an excellent performance response with using traditional PID controller. For example, Gharieb and Nagib [27] demonstrated that for the BPS, the PID controller has a lot of restrictions and limitations such as a long settling time with a high value of overshoot than what is designed to stabilize the ball. Jadlovska *et al.* [33] designed three controllers to compare them; PID, proportional-derivative (PD) , and proportional-sum-derivative (PSD) controllers. These controllers are designed for trajectory tracking of the BPS. The result showed that the PD controller is much better than PID and PSD controllers with a small overshoot. Integer order proportional derivative (IOPD) and fractional order proportional derivative (FOPD) controllers are designed by Borah *et al.* [14] for the trajectory tracking of a ball. Both controllers are designed using different three algorithms to minimise the integral square error. The result verify that, that FOPD outperforms IOPD when tuned by the same algorithm.

A linear state feedback regulator control is presented by Awtar *et al.* [9] and Acosta *et al.* [52]. These controllers were designed by linearizing the BPS dynamics around the central operating point. In [9], the result showed that, the accuracy of the controller for the stabilization test was 5 mm, and for the circle following, an average steady state error was 18 mm while the tracking velocity was less than 4.2 mm/s, while in [52], the result showed that the accuracy of the controller for the stabilization test was 3 mm with 1 s settling time.

A linear quadratic regulator (LQR) control method for a BPS visual servo control was presented by Cheng *et al.* in [17], the proposed controller was implemented on a real system for positioning the ball at central point of the plate with steady state error less than 7 mm. This controller also allow the ball to follow a circular path with an average steady state error was 15 mm.

A linear model predictive control (LMPC) for a BPS was presented using

Matlab simulation by Oravec *et al.* [55] for testing the circle and square reference trajectory tracking including constraints. However, in the simulation results, the MPC strategy was compared to the optimal linear quadratical controller (LQ) which is presented in [54]. The simulation result showed that, the MPC strategy has shown better results for the reference trajectory tracking than the LQ controller.

On the other hand, for a model-free controllers, a fuzzy logic controller was designed for circular tracking experiments in [72] by Yubazaki *et al.* using the single input rule module (SIRM) dynamically connected fuzzy inference model. The experimental results of the proposed controller showed a good performance with maximum tracking error of less than 50 mm. Wang *et al.* [67] have designed a trajectory tracking controller with a double loop strcture, where the inner loop has a servo controller, and the outer loop was designed for a position regulator. The outer loop consist of a single rule fuzzy logic controller with a plate angle as an output , while the position, velocity and acceleration as an input. A fuzzy logic controller was designed by Bai *et al.* [10] for trajectory tracking, where the steady-error of the trajectory tracking is unable to be eliminated. In order to solve the problem of trajectory tracking of the BPS, Han *et al.* [30] and Amin *et al.* [5] have designed a neural network PID controller (PIDNN). Han *et al.* [30] used two PIDNN controllers for two axes. To train a neural network , a differential evaluation PSO (DEPSO) has been used. On the other hand, Amin *et al.* [5] used two parallel sub controllers: Base linear controller with input-output feedback linearization method and NN-based PID compensator for compensation the ignored nonlinear effects that result from base linear controller, Where the final torque input of the real system consisted from the control inputs of the two subsystem.

A fuzzy PD controller for stabilization and trajectory tracking of the BPS with an adaptive integral action to eliminate the steady-state error was designed by Pattanapong and Deelertpaiboon [56]. However, the result showed that, the proposed algorithm of ball detection under various lighting conditions is not robust enough. Gozde [28] presented an evolutionary computation based gain scheduling controller to examine its control performances on a stabilization system. For tuning controller parameters, swarm intelligence-based particle swarm optimization (PSO) algorithm, evolutionary algorithm based differential evolution (DE) algorithms and the classical tuning algorithm are used. The results are compared to the original proportional (P) and probational derivative (PD) controllers that are given on the real BPS. The results showed that, the stability performance of PSO algorithm is better than DE and classical algorithm due to smaller maximum overshoot and greater damping ratio. An observer integrated back-stepping control is designed by Ma *et al.* [44]. A linear extended state observer and a tracking differentiator are used to estimate the uncertainties of the model and derivatives of the virtual controls in the back-stepping design. The results of circle and square trajectory experiments show that the proposed control has the ability of decoupling and suppressing uncertainty while achieving excellent tracking performance.

Moreover, Liu *et al.* [41] have designed a sliding mode controller for the BPS. To deal with disturbances and uncertainties, two scenarios have been used. In the first scenario, the limitation boundary (upper and lower) has been assumed for disturbances and uncertainties. In the second scenario, an observer has been used to estimate it which gives a better response time with a good disturbances rejection when compared to the first one. A sliding mode controller (SMC) with error integration was designed by Bang *et al.* [11] to compare it with linear quadratic (LQ) controller results. This controller was implemented to a stewart platform with rotary actuators to manipulate the plate. From the experiments results for circle trajectory, SMC normally works better than LQ control for a ball and plate system with 4 mm and 12 mm error steady state for SMC and LQ controller respectively. Ali et *et al.* [4] design a new procedure of an optimal nonlinear controller, the proposed procedure depnds on the invasive weed optimization (IWO) to obtain the optimal parameters of the nonlinear controller. However, the controller has proven its effectiveness in tracking the desired trajectories with the presence of these uncertainties and disturbances.

Debono and Bugeja [19] have compared between a designed integral sliding mode and state feedback controllers for the BPS, where both controllers are designed to follow a certain trajectory. The result showed that the sliding mode controller is very quick and managed to obtain a more precise response at much higher speeds, moreover, it is able to track a sinusoidal signal with a magnitude of ± 0.11 m.

On the other hand, the state feedback controller is found to be precise but slow to follow the predefined trajectory. A sliding mode controller with a fuzzy logic controller have been used to control the BPS by Negash and Singh [49], where the controller consisted of a double loop structure. It was noticed that, the sliding mode chattering factor has been reduced by using fuzzy logic. Moezi *et al.* [46] have designed an optimal adaptive interval type-2 fuzzy fractional-order backstepping sliding mode control method to improve the performance of closed loop control of BPS. A fractional-order back-stepping sliding surface is designed to reduce the error , while an adaptive interval type-2 fuzzy compensator is used to compensate the nonlinear effects. However, the experimental results showed the superiority of the proposed method in comparison to other type of SMC.

A BPS controller which presented by Yuan *et al*. [71] and Awtar *et al*. [9] consist from two-loop cascaded control strategy, where the inner loop is used to control the motor angle. The outer loop is used for adjusting the angle of the plates to controls the ball's position. The outer loop must adjust the plat's angle to the desired angle before the ball's position is measured again. It means that the ball position readings sensor must be available before any control action. The inner loop controller consisted of PID controller. For the outer loop, a different strategies have been implemented, a fuzzy logic controller supervisor for a sliding mode and PD controller are used by Yuan *et al*. [71]. For stabilizing the ball on a desired position, each controller takes a long time (roughly 10 second). The result of trajectory tracking showed that the controller takes a long time to measure the ball's position, with high overshoot and error. Awtar *et al*. [9] have designed a standard pole placement and LQR design procedures for the outer loop.

Awtar *et al*. [9], Rastin *et al*. [57] and Moarref [45] present different mechanisms of the mechanical designs that can used for the real BPS. Awtar *et al*. [9] designed an L-shaped mechanism to transmit the required torque from the motor to the plate. This is a most popular mechanical structure which is used to couple the motors to the plate and provide the necessary torque. The main limitation of the L-shaped structure ignores the small angle plate deflection which is lead to the ball slipping on the plate. Rastin *et al*. [57] presented a different actuation of the mechanical structure approach where the system has three actuators. This approach is more complicated than L-shaped approach where the axes of rotation of the plate are not independent. The design used by Moarref *et al*. [45] uses a long arm to transmit the torque from the motor to the plate. The main limitation of the both motors when they move, the controller must be slow as possible to avoid the damage for the system. The design of Debono, and Bugeja [19] allows the plate of the system to be rotated independently by each motor, where the system was designed with an inner plate and external frame. The external plate is moved by one motor which is fixed on the base and the inner plate is moving by a fixed motor on the external frame.

For sensing the position of the ball on the plate, the two most common approaches are using a resistive touch screen in [9, 72], or an overhead camera in [27, 52, 52, 10, 19, 57]. Both sensing method convey signals that are used for a driver circuit to process the information and then convert it to the coordinates (x,y). An array of phototransistors [73] is arranged on the plate to sense the position of ball by interruption of light.

In this thesis, the stability of the five selected algorithim was checked in [40, 65, 74, 70] based on lyapunov stability, where Lewis in [40] proved the stability of LQR and LQT controller based on lyapunov function. Valluri *et al.* in [65] proved the stability of liner and nonlinear system with MPC controller. For a state feedback controller, Zhao *et al.* in [74] present a state feedback switching parameters controller and proved the stability of the controller. Antonito *et al.* in [70] checked and proved the stability of the Classical PID Controller via Lyapunov stability theory.

1.2 Thesis Objectives and Contributions

The main objectives of this thesis are to compare between different types of linear controller stratigies of a position and trajectory tracking control approaches for the BPS .

Therefore, the following analysis and experiments will be done to achieve our objectives:

- Analyze and model of the BPS using Euler-lagrangian method.
- Analyze the controllability and observability of the system using the obtained model.
- Apply different control strategies for the BPS for regulation and tracking and then compare between them.
- Design observers for different controller, and compare between methods to obtain the best results in terms of performance and robustness.
- Simulate the system controllers with observer using the exact nonlinear model in Simulink.
- Design a real time control system for real time simulation and implementation of the LMPC algorithm.
- Perform experimental stability and tracking tests of the control system, and compare between the results of different controllers.

1.3 Thesis Organization

The remaining parts of the thesis are organized as follows:

Chapter 2 describes the mechanical and electrical setup of the BPS that are used in this thesis. Specifically, the main characteristics of the linkage mechanism that are used for motion transmission from the servo motor to the system plate as well as the main characteristics and restrictions of the hardware components, such as the touch screen, servo system and different drivers. It also presents the physical configuration with electrical connection diagram of these components in order to have a better understanding of the system operation .

Chapter 3 presents the considerations and assumption that are taken for the mathematical modeling of the BPS. This chapter also presents the calculation of the kinetic and potential energies, kinematics with the deduction of the dynamics equations using the Euler-Lagrange approach to derive the nonlinear model. Also, this chapter presents the BPS using state space representation and the assumption that are used to linearize the system dynamics.

Chapter 4 presents the design process of linear controllers including the theory of these controllers, controller design in the continuous and discrete time fashions. In the same way for the observer, the design process is presented using linear approaches, such as the Luenberger observer. The pole placement and LQR method are considered in the designing of the different controllers.

In Chapter 5, simulations result are presented and the performance of the different controllers are compared.

Chapter 6 presents the experimental results of each approach that are raised in Chapter 5 using different reference trajectories.

Conclusions and outlook of the thesis are presented in Chapter 7.

Chapter 2

Structure of the BPS

2.1 Mechanical Structure of the BPS

Since our BPS must be designed such that the ball is free to roll on the plate, two servo motors that are controlled independently must be linked to the plate through a two-linkage mechanism. Each linkage mechanism has a universal joint which enables the plate to rotate in both directions at the same time. A third universal joint is attached to the center rod and the plate. The mechanical parts of the BPS is illustrated in Fig. 2.1,which contains;

1. The BPS base which caries all the other components of the system.
2. Plate holder which is used to hold the resistive touch screen.
3. The central shaft which is used between structure base and the middle of plate holder. Its job is holding the plate holder from the center point.
4. The servo motor holder which fits the main base, and it will be fixed by nuts and screw.

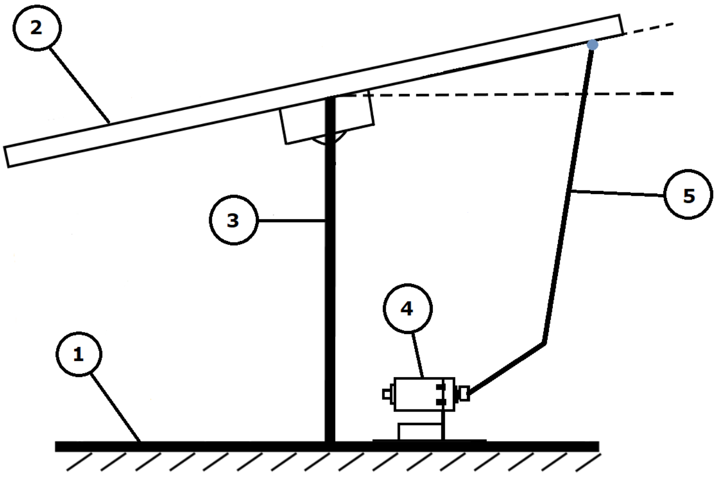


Figure 2.1: 2D view of BPS.

5. Two independent two-linkage mechanism which is used to convert rotation motion from servo motor to linear motion in the plate.

To analyze our linkage mechanism of our system, Fig. 2.2, can be used to show all dimensions and angles.

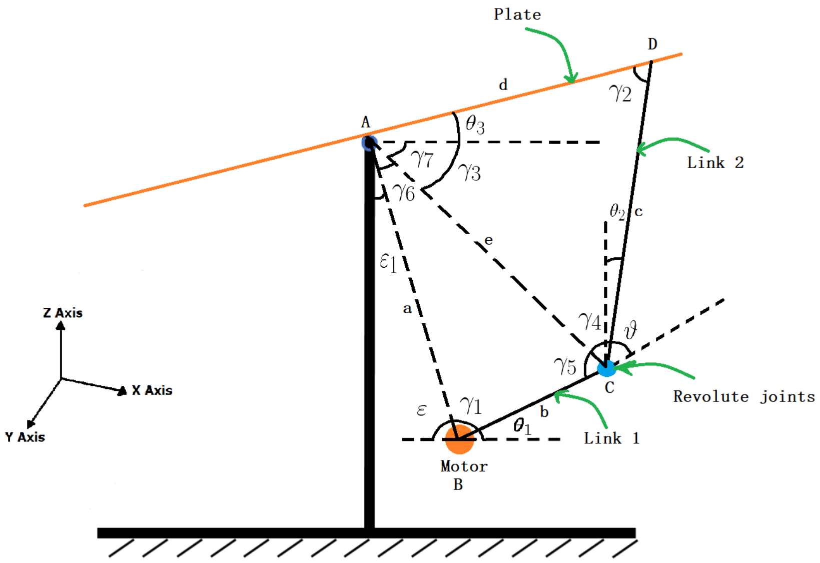


Figure 2.2: Linkage mechanism for the plate rotation.

(3.11)-(3.12)

To understand the linkage mobility, initially, the value of the angles (ε and ε_1) and the lengths of all straight lines (a, b, c and d) in Fig. 2.2 are supposed to be known. When the motor arm (Link 1) is moved with a specific angle θ_1, then the value of the angle γ_1 can be found by:

$$\theta_1 + \varepsilon + \gamma_1 = 180 \tag{2.1}$$

Then, based on the value of the angle γ_1, the length 'e' can be computed by :

$$e = \sqrt{a^2 + b^2 - 2ab\cos\gamma_1} \tag{2.2}$$

Therefore, the values of angles γ_3, γ_6 and γ_7 can be computed as :

$$\gamma_3 = \cos^{-1}\left[\frac{d^2 + e^2 - c^2}{2de}\right] \tag{2.3}$$

$$\gamma_6 = \cos^{-1}\left[\frac{a^2 + e^2 - b^2}{2ae}\right] \tag{2.4}$$

$$\gamma_7 = 90 - (\gamma_6 + \varepsilon_1) \tag{2.5}$$

Accordingly, when the servo motor is moved with a specific angle (θ_1), then the value of angle that will deliverd to the plate (θ_3) is as follows:

$$\theta_3 = \gamma_3 - \gamma_7 \tag{2.6}$$

These equations (2.1)-(2.6) will be used within the Arduino programming code to calculate the specific final angle that the servo motor will move to balance the rolling ball on a specific point on the plate.

The mechanical values and parameters of the BPS that are used in this thesis can be summarized in Table 2.1, where the x, y, z axes are shown in Fig. 2.2 and the mass moment of inertia of the plate and ball were calculated according to [24] .

Table 2.1: Parameters of the BPS

Parameters	Description	Value	Unit
m	Ball mass	0.260	Kg
r	Radius of the ball	0.02	m
I_{px}	Mass moment of inertia of the plate about x axis	0.4	$Kg.m^2$
I_{py}	Mass moment of inertia of the plate about y axis	0.676	$Kg.m^2$
I_{pz}	Mass moment of inertia of the plate about z axis	1.076	$Kg.m^2$
I_b	Mass moment of inertia of the ball	10.4104×10^{-4}	$Kg.m^2$
g	Gravitational acceleration	9.89	m/s^2

2.2 Electrical Structure of the BPS

To stabilize the ball on the desired position on the plate, the system must be able to alter the inclination of the plate around the two orthogonal x and y axis. This is achieved by two servo motors which is connected to the plate by two pairs of universal linkage rod. A four wire resistive touch screen which acts as a plate, will provide the ball position in two dimensions (x and y) as an input to the Arduino microcontroller. Then the Arduino microcontroller will be used to calculate the necessary output signals from a control law. Finally, the servo motors receive these control signals and rotate the plate with angles α and β to result the ball movement on the plate in x and y directions, respectively. The state variables of the BPS are sent from the Arduino microcontroller to the personal computer (PC) via serial port to draw the state variables outputs. For the MPC strategy, A new Arduino library was created to connect Arduino uno microcontroller with PC, where an Arduino uno is being used as a central control to manage the input and output signals between BPS system and PC. Firstly, the resistive touch screen will provide the ball position (x, y) to the Arduino microcontroller, then the arduino will sent the data of ball position directly via serial port to the PC. The PC contains a Matlab program that is used for designing and implementation a real-time MPC strategy and to calculate the control signals, then the PC will sent the control signals serially to the Arduino microcontroller. The servo motors receives these signals from Arduino to rotate the plate with α and β. Fig. 2.3 below visualizes the basic elements and flow of information through the system.

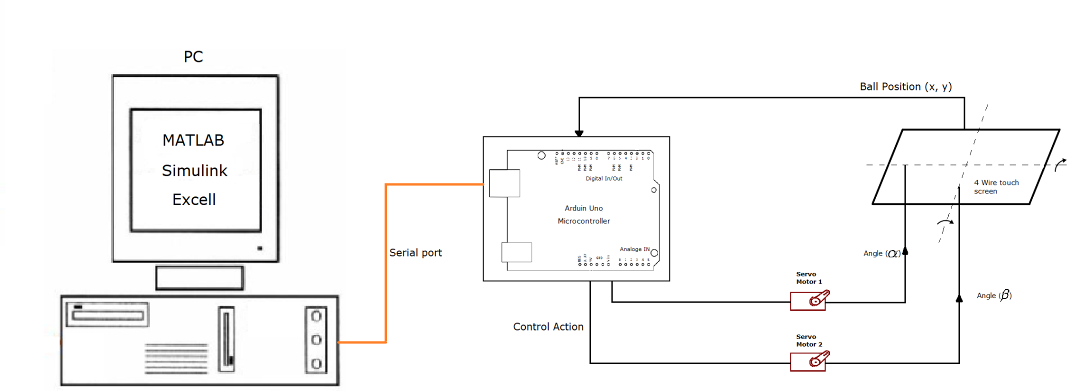


Figure 2.3: Flow of information in BPS.

The electrical parts of the BPS include a touch screen to determine the ball position, servo motors to change the title angles of the plate and the interfacing circuits that connect touch screen and servo motors with the controller. Fig. 2.4 demonstrates the electrical connection between different parts of BPS.

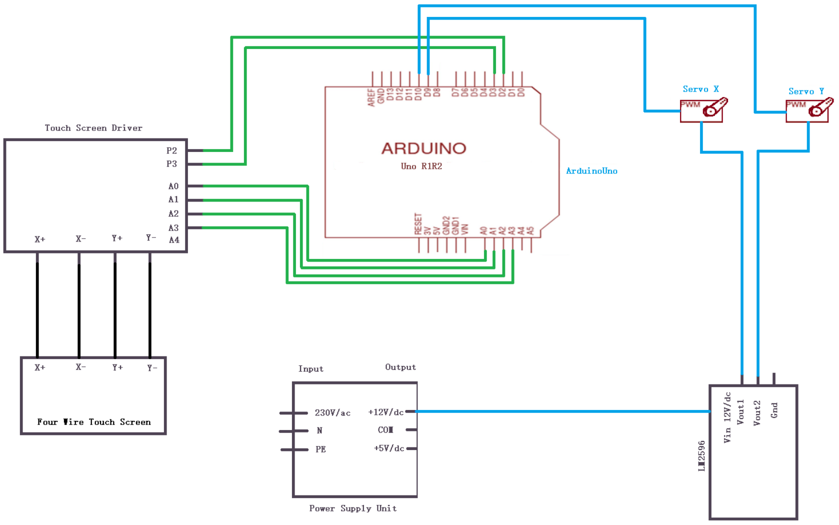


Figure 2.4: Electrical schimatic diagram of the BPS.

2.2.1 Resistive Touch Screen.

A resistive touch screen is a two dimensional sensing device which is constructed of two separated material by spacers. A dc voltage must be applied across resistor network. Therefore, when a screen is touched by a ball, the value of resistance will change at a given point [42]. Fig. 2.5 illustrates the interface circuit between the Arduino microcontroller with a four wires touch screen. To get the y touch screen position, the Arduino microcontroller sets $A0$ to +5V and $A2$ to GND, after that, the microcontroller uses Pin 3 to read the analogue value to represent the y coordinate of the touch point. To get the x touch screen position, the microcontroller sets $A1$ to +5V and $A3$ to GND, then the controller uses Pin 2 to read the analogue value to calculate the x coordinate of the touch point.

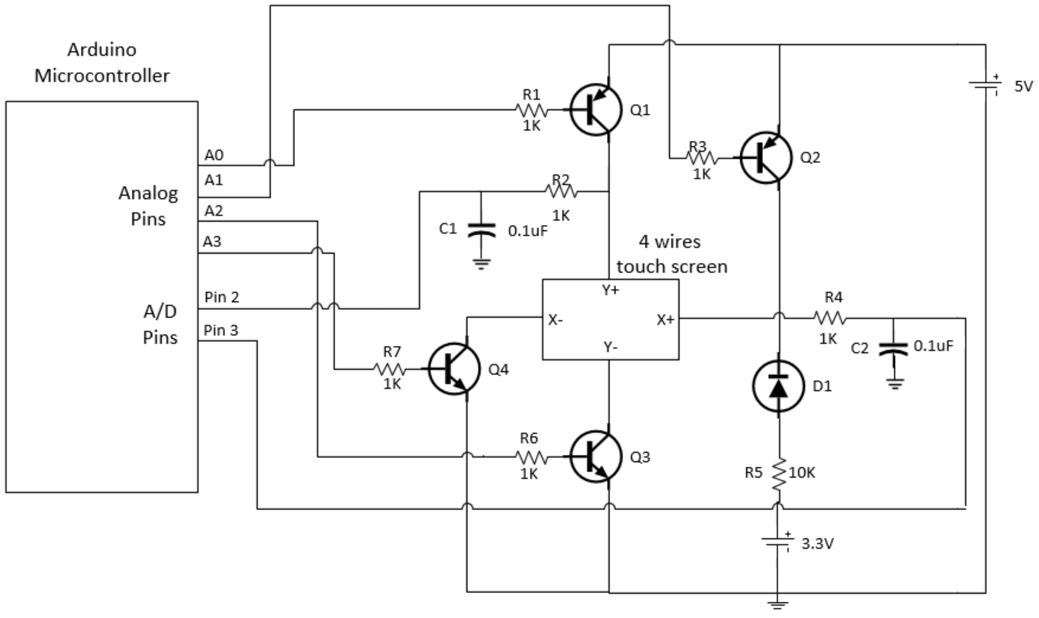


Figure 2.5: Arduino microcontroller and four wires touch screen interface circuit.

2.2.2 Servo Motors

A servomotor is a simple electric motor with a rotary or linear actuator which has the ability to provide precise control of torque, speed, acceleration or position using closed-loop feedback. It also has a relatively customized controller. It often has a dedicated console to control of a servo motor [32]. In this thesis we use FS5106B servo motor which has operating speeds ($0.18sec/60^\circ$ at $4.8V$ and $0.16sec/60^\circ$ at $6.0V$). In addition, the operated torque that can delivered by the servo motor ($5Kg.cm$ at $4.8V$ and $6Kg.cm$ at$6.0V$) [22].

2.2.3 Microcontroller

In this work, Arduino Uno R3 will be used as a microcontroller. Arduino Uno is a microcontroller board based on 8-bit the ATmega 328P. It has 14 digital input/output terminals, where six of them can be used as a pulse width modulation (PWM) outputs. It has six analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, and a reset button [2].

Four of analogue pins and two digital pins were used from Arduino Uno for reading x and y positions from resistive touch screen. In addition, two digital pins were used as a PWM outputs as a control law to move a servo motor at a predicted angle.

Chapter 3

Mathematical Modeling of the BPS

The mathematical model of any mechanical system can be derived using two methods; 1- The classical Newton method which is used for the systems that have multiple degrees of freedom. 2- Modern Euler-Lagrange method which will be used in this thesis. First, the Lagrangian function is defined as follows [71, 23, 39]:

$$L(q_i, \dot{q}_i, t) = T(\dot{q}_i, t) - V(q_i, t) \tag{3.1}$$

where L is the Lagrangian function which represents the difference between the potential energy V and the kinetic energy T, $q_i \in \{x, y, \alpha, \beta\}$ stands for x and y direction coordinate to represent the system states and $\dot{q}_i \in \left\{\dot{x}, \dot{y}, \dot{\alpha}, \dot{\beta}\right\}$ is the system state derivatives.

Then, the general Euler-Lagrange equations is defined as:

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \tag{3.2}$$

where Q_i represents the composite force which acting on the system.

3.1 Mathematical Model of the BPS

For the derivation of the mathematical model of BPS, the following assumptions are made for the free body diagram of BPS shown in Figure 3.1 [6, 35]:-

1. The ball and the plate are in contact all the time and the rolling occurs without slipping.

2. The ball is completely symmetric and homogeneous.

3. All friction forces and rotational moments are neglected.

The free body diagram of BPS as shown in Fig. 3.1.

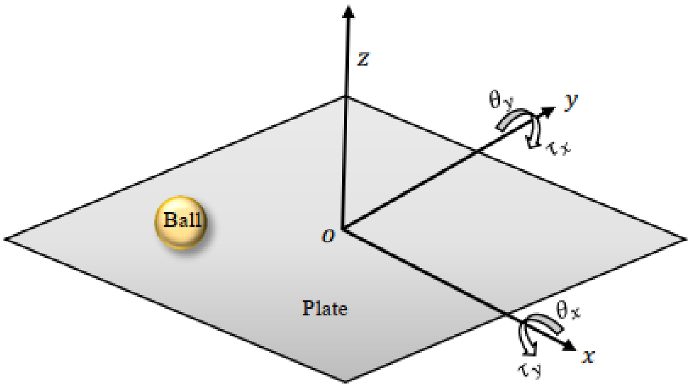


Figure 3.1: A free body diagram of the BPS.

Since the generalized coordinates of the ball position are x and y axis with the origin at the plate center, we must assume that the plate inclination α and β are driven by generalized torques τ_x and τ_y which act on the plate in the corresponding direction. Therefore, the system has four degree of freedom (DOF); two for ball motion along the x and y-axis respectively and two in the inclination of plate in rotation about the x and y-axis [69].

The kinetic energy of the ball is given by rotational energy relative to the centre of the rolling ball as well as translational energy of the center of the ball, that means :

$$T_b = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}I_b(w_x^2 + w_y^2) \tag{3.3}$$

where the relations between translational and rotational velocities are:

$$\begin{aligned} \dot{x} &= rw_x \text{ and} \qquad (3.4) \\ \dot{y} &= rw_y \qquad (3.5) \end{aligned}$$

Therefore, the kinetic energy of the ball is given by

$$T_b = \frac{1}{2}(m + \frac{I_b}{r^2})(\dot{x}^2 + \dot{y}^2) \tag{3.6}$$

We consider that the ball is a point mass which is placed in any (x,y) position and it rotates around its center. Then the kinetic energy of the plate is

$$T_p = \frac{1}{2}(I_p + I_b)(\dot{\alpha^2} + \dot{\beta^2}) + \frac{1}{2}m(x^2\dot{\alpha^2} + 2x\dot{\alpha}y\dot{\beta} + y^2\dot{\beta^2}) \tag{3.7}$$

Then the total kinetic energy of the system T can be yielded by summing T_b and T_p, i.e.;

$$\begin{aligned} T = \frac{1}{2}(m_1 + \frac{I_{b1}}{r_1^2})(\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2}(I_p + I_b)(\dot{\alpha^2} + \dot{\beta^2} \\ + \frac{1}{2}m(x^2\dot{\alpha^2} + 2x\dot{\alpha}y\dot{\beta} + y^2\dot{\beta^2}) \end{aligned} \tag{3.8}$$

The total potential energy of the ball with respect to horizontal plane in the center of the inclined plate is given by:

$$V = mg(x\sin\alpha + y\sin\beta) \tag{3.9}$$

Then, the total energy L of the system by substituting Eqs (3.8) and (3.9) in Eq. (3.1) we yield :

$$\begin{aligned} L = \frac{1}{2}(m + \frac{I_b}{r^2})(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}(I_p + I_b)(\dot{\alpha}^2 + \dot{\beta}^2 \\ + \frac{1}{2}m(x^2\dot{\alpha}^2 + 2x\dot{\alpha}y\dot{\beta} + y^2\dot{\beta}^2) - mg(xsin\alpha + ysin\beta) \end{aligned} \tag{3.10}$$

Applying Lagrange-Euler equation of motion Eq. (3.2) to the total energy of the BPS Eq. (3.10), we obtain the following nonlinear differential equations :

$$\tau_x = (I_p + I_b + mx^2)\ddot{\alpha} + 2mx\dot{x}\dot{\alpha} + mxy\ddot{\beta} + (m\dot{x}y + mx\dot{y})\dot{\beta} + mgxcos\alpha \tag{3.11}$$

$$\tau_y = (I_p + I_b + my^2)\ddot{\beta} + 2my\dot{y}\dot{\beta} + mxy\ddot{\alpha} + (m\dot{x}y + mx\dot{y})\dot{\alpha} + mgycos\beta \tag{3.12}$$

$$(m + \frac{I_b}{r^2})\ddot{x} - m\dot{\alpha}^2x - m\dot{\alpha}y\dot{\beta} + mgsin\alpha = 0 \tag{3.13}$$

$$(m + \frac{I_b}{r^2})\ddot{y} - mx\dot{\alpha}\dot{\beta} - m\dot{\beta}^2y + mgsin\beta = 0 \tag{3.14}$$

The description of variables and parameters of model-deducing process definitions, are listed in Table 3.1.

Table 3.1: The Parameters and Variables of Mathematical Model

Symbol	Description	Unit
m	Ball mass.	Kg
r	Radius of the ball.	m
I_p	Mass moment of inertia of the plate.	$Kg.m^2$
I_b	Mass moment of inertia of the ball.	$Kg.m^2$
x, y	Ball displacement in the x or y directions, respectively.	m
$\dot{x}, \dot{y}$	Velocity of the ball in the x or y axes, respectively.	m/s
α,β	Plate inclination angle in the x or y directions, respectively	rad
$\dot{\alpha}, \dot{\beta}$	Angular velocity of the plate from x or y axes, respectively.	rad/s
τ_x, τ_y	Torque applied to the plate in the x or y axes, respectively.	Kgm^2/s^2
w_x, w_y	Angular Velocity of the ball in the x or y axes, respectively.	rad/s
g	Gravitational acceleration.	m/s^2

The Eqs. (3.11)-(3.12) show the effect of external torque on BPS and how the plate inclination dynamics is influenced by these tourques. Eqs. (3.13)-(3.14) describe the ball motion on the plate, and show that the acceleration of the ball movements depends on the angle and angular velocity of the plate inclination [20].

The interpretation of the some terms in Eqs. (3.13)-(3.14) are listed in Table 3.2 [29, 31].

Table 3.2: Interpretation of the particular terms in Eqs (3.13)-(3.14)

Term	Interpretation
$(I_p+I_b+mx^2)\ddot{\alpha}$ and $(I_p+I_b+my^2)\ddot{\beta}$	Torques as a product of inertia of the ball and plate with angular acceleration about x and y axes, respectively.
$m\dot{\alpha}^2x + m\dot{\alpha}y\dot{\beta}$	The centrifugal torque resulting from plate rotation
$mxy\ddot{\beta} + m\dot{x}y\dot{\beta} + mx\dot{y}\dot{\beta}$,$mxy\ddot{\alpha} + m\dot{x}y\dot{\alpha} + mx\dot{y}\dot{\alpha}$	Influence of gyroscopic moments in x and y axes, respectively.
$2mx\dot{x}\dot{\alpha}$ and $2my\dot{y}\dot{\beta}$	Coriolis accelerations in x and y axes, respectively.
$mgsin\alpha$ and $mgsin\beta$	Moment as a product of ball weigh in x and y axes, respectively.

3.2 Linear Model of the BPS

The BPS system can be rewritten in a matrix form due to Eqs (3.11)-(3.14) by choosing $[x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8]^T = [x\ \dot{x}\ \alpha\ \dot{\alpha}\ y\ \dot{y}\ \beta\ \dot{\beta}]^T$ and $u = [\tau_x\ \tau_y]^T$

as follows:-

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} x_2 \\ A(x_1 x_4^2 + x_4 x_5 x_8 - g\sin x_3) \\ x_4 \\ 0 \\ x_6 \\ A x_5 x_8^2 + x_1 x_4 x_8 - g\sin x_7) \\ x_8 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}$$

where $A = \frac{m}{m + \frac{I_b}{r^2}}$.

The approximate moment of inertia value for the solid ball's is $I_b = \frac{2}{5}mr^2$. Therefore Eqs. (3.13)-(3.14) can be written as:

$$m(\frac{5}{7}\ddot{x} - (x\dot{\alpha}^2 + y\dot{\alpha}\dot{\beta} + g\sin\alpha)) = 0 \tag{3.15}$$

$$m(\frac{5}{7}\ddot{y} - (y\dot{\beta}^2 + x\dot{\alpha}\dot{\beta} + g\sin\beta)) = 0 \tag{3.16}$$

However, above Eq. (3.15)-(3.16) can be simplified by considering the following assumptions:

1. We assume that, the angle of inclination for the plate $-30° \leq (\alpha, \beta) \leq 30°$ with slow rate of change for the plate inclination, which means that $\sin\alpha \simeq \alpha$, $\sin\beta \simeq \beta$, $\dot{\alpha}^2 \simeq 0$, $\dot{\beta}^2 \simeq 0$ and $\dot{\alpha}\dot{\beta} \simeq 0$.

2. We assume α and β angels are considered as an system inputs instead of the torque moments τ_x and τ_y, which means that the Eqs. (3.11)-(3.12)

could be dropped out from the state-space representation.

After these simplifications the equations of the system can be rewritten as follow:

$$\frac{5}{7}\ddot{x} + g\alpha = 0 \tag{3.17}$$

$$\frac{5}{7}\ddot{y} + g\beta = 0 \tag{3.18}$$

By assuiming α and β as inputs to BPS, the transfer function of seperate x and y axes as follow:

$$G(s) = \frac{x(s)}{\alpha(s)} = \frac{g}{\frac{5}{7}s^2} = \frac{7.07}{s^2} \tag{3.19}$$

$$G(s) = \frac{y(s)}{\beta(s)} = \frac{g}{\frac{5}{7}s^2} = \frac{7.07}{s^2} \tag{3.20}$$

Based on basic linearizing assumptions , the simplified BPS state space representation can be obtained as follows:-

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} x_2 \\ 0 \\ x_6 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\frac{5}{7}g & 0 \\ 0 & 0 \\ 0 & -\frac{5}{7}g \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_5 \\ x_6 \end{bmatrix}$$

The BPS dynamics shows that the multi-input-multi-output (MIMO) BPS system which can be regarded as two single-input-single-output (SISO) sub-systems

Chapter 4

Control Approaches of the BPS

Since the BPS is commonly used system in a control lab for testing. Several control theories and strategies can be tested using the BPS in a control lab. The main challenge of the BPS is to build a controller that can stabilize the ball on a desired position on the plate, or to track the ball to a predefined position. Furthermore, such a controller must reject random disturbances which act on the ball or on the plate within physical constrains of the device. However, several challenges in BPS raise due to the following facts :

1. In general, the open loop of BPS is unstable at a desired operating point.
2. Some of the BPS states are not measured directly such as linear velocities which must be accurately estimated.
3. The external disturbances that act on the system are not measurable.

The control scheme for the BPS consists of two main loops. The outer loop is used to regulate a ball position (x, y) on the plate to a desired position. The outputs of this loop are plate inclinations. Then, the inclinations of the plate are sent to the inner loop as reference. A servo motor controller in the inner loop will derive plate slopes to follow the desired reference [12, 68]. To implement this design, the plate must reach the angle that is given by the outer loop before the ball's position is measured again [62]. Moreover, the inner control loop can be only designed by taking the motor physical limitations into account; motor current, motor torque and maximum angles . The block diagram of the control loops of the BPS is shown in Fig. 4.1 . The inner loop consists of the servo motor which has an internal proportional-integral (PI) controller that controls its shaft position with a desired plate angle (α or β), so the torque of the servo motor is controlled indirectly [1].

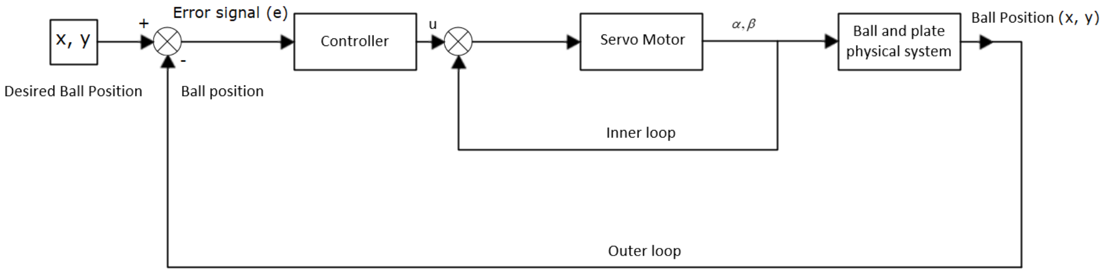


Figure 4.1: Control loops of the BPS.

In this work, we use five control methods to control the ball position of the BPS. In order to compare between the responses and performances of these methods, some of the requirements are fixed, namely the maximum response settling time (T_s), the maximum percentage overshoot ($OS\%$) and the steady state error (e_{ss}). These requirements are fixed to, 2.1 second, 10 % and zero, respectively.

In the upcoming sections, control theories and strategies that are based on the linearized BPS model will be designed and discussed, namely;

- PID controller.

- State feedback controllers (regulator, observer and discrete controller).
- Linear quadratic regulator and tracker (LQR and LQT), that are based on the linear optimal control theory.
- Linear model predictive controller (LMPC).

4.1 PID controller

The general PID control formula in continuous time is as follows [66]:

$$\mathbf{u}(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{dt} \tag{4.1}$$

where K_P, K_I, K_D are the proportional, integral and derivative gains that interprets current, past and future errors, respectively.

By taking the Laplace transformation for the both side of the Eq. (4.1) [60]. The transfer function of the PID controller that represent the relations between inputs and outputs of the system is as follow:

$$K(s) = K_P + K_I \frac{1}{s} + K_D s \tag{4.2}$$

The transfer function of the PID controller in Eq. (4.2) can be also represented by a proportional plus derivative (PD) compensator that is cascaded by a proportional plus integral (PI) compensator. Thus, the PID controller configuration is of the form:

$$G_{PID}(s) = \frac{K_D(s^2 + \frac{K_I}{K_D}s + \frac{K_P}{K_D})}{s} \tag{4.3}$$

In the designing of PID control algorithm, there are several algorithms that can be used to satisfy the given performance [7]. In order to obtain an easier PID control algorithm, the MIMO system of BPS is simplified to a simpler combination of two SISO systems, then both of x axis and y axes can be controlled individually. For designing a PID controller, a MATLAB SISOtool was used to get a controller gains and make tuning of the PID controller [64], where the MATLAB SISOtool depends on the root locus that examine how the roots of the system change with variation of system parameter.

To use this tool, the open loop transfer function of the BPS in Eq. (3.19) must be defined, then to achieve the desired response, two zeros of the PID controller are used and located at left side of the root locus with one pole at the origin, thus introduce additional -135 degrees. For more detail on this method and tool see [64]. The PID controller gains are listed in Table 4.1.

Table 4.1: PID controller parameters.

K_P	K_I	K_D
1.69	1.12	0.662

To find the discrete time of PID controller that is will be implemented on the microcontroller, we define and differentiate with respect to the time the continuous time PID controller of Eq. (4.1), that means;

$$\frac{du(t)}{dt} = K_p \frac{de(t)}{dt} + K_I e(t) + K_D \frac{d^2e(t)}{dt^2} \tag{4.4}$$

where the values of K_P, K_I and K_D in discrete time will be the same of continous time.

Applying the backward differentiation method to Eq. (4.4) using Euler approximation with sampling time (T) and instant time (k)gives :

$$\frac{u(kT)-u(kT-T)}{T}=K_p\frac{e(kT)-e(kT-T)}{T}+K_Ie(kT)+K_D\frac{\dot{e}(kT)-\dot{e}(kT-T)}{T} \tag{4.5}$$

Solving for $u(KT)$ finally gives the discrete-time PID controller:

$$u(kT)=(\frac{K_p}{T}+K_I+\frac{K_D}{T^2})e(kT)-(\frac{K_p}{T}+\frac{2K_D}{T^2})e(kT-T)+\frac{K_D}{T^2}e(kT-2T)+u(kT-T) \tag{4.6}$$

4.2 State-Space Controllers

The general linear time invariant state space model of a system with states $\mathbf{X}$, inputs u and outputs $\mathbf{Y}$ is written in the following form [15]:

$$\dot{\mathbf{X}}=A\mathbf{X}+B\mathbf{u} \tag{4.7}$$

$$\mathbf{Y}=C\mathbf{X}+D\mathbf{u} \tag{4.8}$$

where $\mathbf{X}\in\mathbb{R}^4$, $u\in\mathbb{R}^2$, $\mathbf{Y}\in\mathbb{R}^2$ and $p\in\mathbb{R}^2$ are the system state, input, output and disturbances vectors, respectively, and $A\in\mathbb{R}^{4\times 2}$, $B\in\mathbb{R}^{4\times 2}$, $C\in\mathbb{R}^{2\times 4}$ and $D\in\mathbb{R}^{2\times 2}$, are the system, input, output and feed-forward matrices, respectively.

The state-space representation of our BPS can be shown in Fig. 4.2.

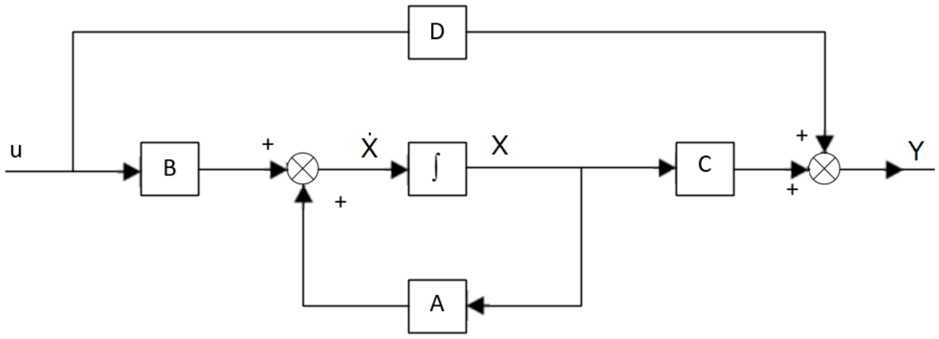


Figure 4.2: Block diagram representation of the state space equations.

To design a controllers based on a state space matrices, the controllability is a key issue that must be studied firstly. The BPS is said controllable if and only if the input can be found that takes every state variables from a desired initial state to a desired final state [59]. A system is said to be completely state controllable if the Kalman controllability matrix $C_m \in \mathbb{R}^{4\times 2}$ is of full rank, that is it contains n linearly independent column or row vectors [16]. Therefore :

$$C_M = [B \quad AB \quad A^2B \quad A^3B] \tag{4.9}$$

When C_M has a full row rank, then the system is controllable, and it is possible to find a gain vector K. To check the controllability of the pair matrices (A, B) of the BPS, the controllability matrix is calculated firstly, and then its rank is found. The controllability test is performed using Matlab [1].

Since C_M has a full row rank 4, it follows that the system is fully controllable, and the gain vector K can be calculated, so that to achieve the

[1] To calculate the controllability matrix and it is rank, we use 'ctrl' and 'rank' commands, respectively.

stability response at the desired operating point, meet the transient specifications, and satisfy robustness specification. Two methods can be used to find the gain matrix vector K, which are represented by the pole placement method [37] and optimal method; however the pole placement method is used to design a state feedback controller, on the other hand, the optimal method is used to design the optimal quadratic controller.

4.2.1 Regulator Design

A state space regulator is designed to bring the ball at the origin of the plate, and to improve its dynamic response specifications by keeping the ball at it is stationary point as shown in Fig. 4.3. This regulator overcomes the effects of disturbances and non-zero initial conditions. Two methods are used to design the state space regulator, namely:

1. State feedback method.
2. Quadratic optimal method.

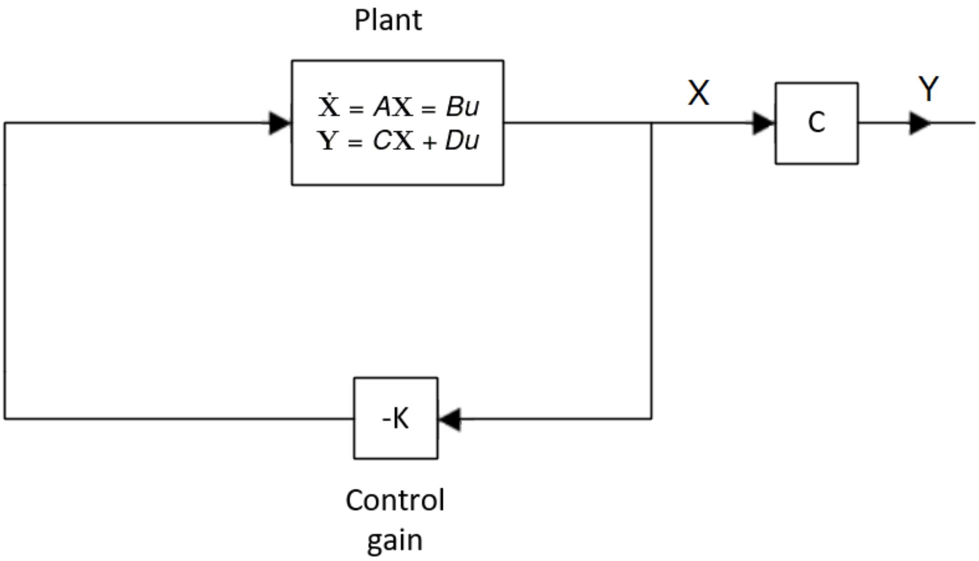


Figure 4.3: Regulator blocks diagram.

State Feedback Regulator

The gain matrix K can be computed by using the state feedback method to place the eigenvalues of the system matrix (closed-loop poles) in the desired location. Now, form the closed-loop system by feeding back each state variable to the input as shown in Fig. 4.3 , forming :

$$\mathbf{u} = -KX \tag{4.10}$$

where $K \in \mathbb{R}^{2\times 4}$

This means that the control signal u is determined by an instantaneous state which is known as state feedback. Substituting equation (4.10) in Eq. (4.7) gives the state equations for the closed-loop system as follow :

$$\dot{\mathbf{X}} = (A - BK)\mathbf{X} \tag{4.11}$$

$$\mathbf{Y} = C\mathbf{X} \tag{4.12}$$

where the eigen values of matrix $(A - BK)$ are called regulator poles.

Assuming the natural frequency (w_n) and the damping ratio (ζ) of the desired closed loop poles which meet the system requirements rang between (3.2 to 3.3) rad/s and(0.6 to 6.5) respectively to avoid the axes redundancy. The closed loop poles (p) location will be at :

$$p = \begin{bmatrix} -1.9048 + 2.5976i & -1.9048 - 2.5976i \\ -2.1450 + 2.5078i & -2.1450 - 2.5078i \end{bmatrix}$$

the gain matrix K are found using Matlab program [2], which achieves the desired closed loop poles.

$$K = \begin{bmatrix} -1.5470 & -0.6062 & -0.0643 & -0.0167 \\ 0.0431 & -0.0046 & -1.4858 & -0.5497 \end{bmatrix}$$

In order to implement the state feedback controller on the Arduino microcontroller, we need to get the discrete equation form of the continuous state space equations that is represented in the Eqs. (4.7)-(4.8). Assuming that, the sampling time of the discretization T_s is very small value ($T_s \ll 1$), the state equations for the discrete time state feedback controller are set as follows:

$$\vec{\mathbf{X}}(k+1) = F\vec{\mathbf{X}}(k) + G\vec{u}(k) \tag{4.13}$$

[2] We use the Matlab function 'place' to find the closed loop gain.

$$\vec{\mathbf{Y}}(k) = C\vec{\mathbf{X}}(k) \tag{4.14}$$

where $k \in N$ is a discrete time instances , $F \in \mathbb{R}^{4\times4} \approx 1 + T_s A$ and $G \in \mathbb{R}^{4\times2} \approx T_s B$ are the state and input matrices, respectively.

Now design a descrite time state feedback controller, the controllability of the pair F, G were checked, this can be performed using Matlab [3]. Which also leads the full rank and thus fully controllable system. However, we can get the discrete time model equation using Matlab software by using some built in commands [4], which is converts continuous time dynamic system to discrete time dynamic system with using zero-order-hold(ZOH) method [3]. By defining the sampling time $Ts = 10\ ms$, the discrete time poles P_d to place the eigenvalues of the system at the desired location will be at :

$$P_d = \begin{bmatrix} 0.7988 + 0.2123i & 0.7988 - 0.2123i \\ 0.7817 + 0.2002i & 0.7817 - 0.2002i \end{bmatrix}$$

This follows that the discrete time giants K_d that achieve the desired closed loop poles as follow:

$$K_d = \begin{bmatrix} -1.5145 & -0.6011 & -0.0615 & -0.0167 \\ 0.0410 & -0.0047 & -1.4574 & -0.5465 \end{bmatrix}$$

The simulink discrete state feedback regulator schematic diagram is shown in Fig. 4.4.

[3] We use the Matlab function 'ctrb' and 'rank' to find the controllability of the system.

[4] We use the Matlab function 'c2d' to find the discrete-time system.

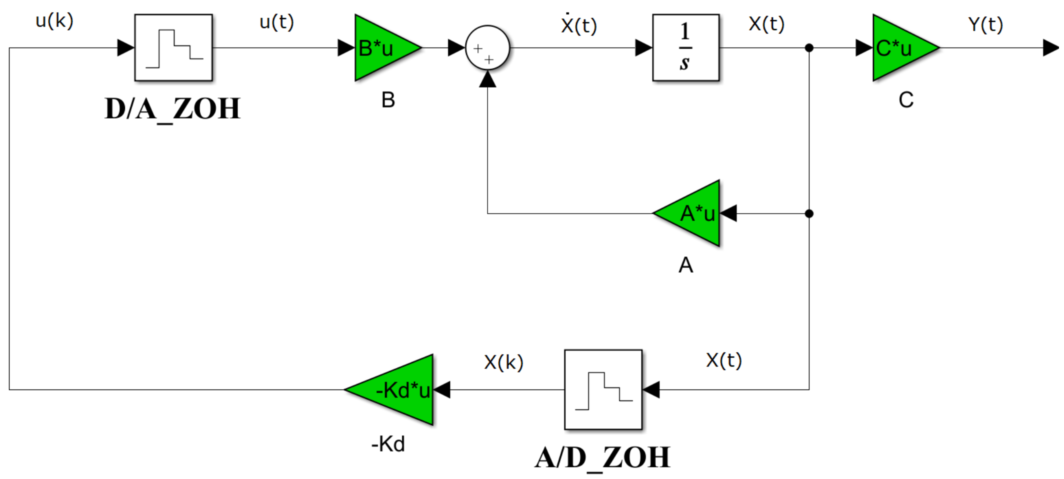


Figure 4.4: The simulink model for discrete time state feedback controller.

Linear Quadratic Regulator (LQR)

The Linear quadratic regulator (LQR) method serves to find the optimal control input $u(t)$, which will drive the system from its initial states $\mathbf{X}(t_0)$ to some final values $\mathbf{X}(t_f)$. It is a powerful technique for designing controllers for complex systems that seeks to find the optimal controller which minimizes a given cost functional. The cost functional is parameterized by two matrices, R which represent the weight of system input, and Q to represent the weight of state vector. LQR obtains the optimal control input by solving the algebraic Riccatti equation based on the state space model [53].

The cost function to calculate the state feedback control gains is given as

$$J = \int_0^\infty (\mathbf{X}^T Q \mathbf{X} + \mathbf{u}^T R \mathbf{u}) dt \tag{4.15}$$

where $Q \in \mathbb{R}^{4\times4}$ and $R \in \mathbb{R}^{2\times2}$ are positive semi-definite matrices which represent the state penalty matrix and the control penalty matrix, respectively [13], $\mathbf{X}$ and $\mathbf{u}$ respectively represents matrices corresponding to system state and input. The matrix Q and R determine the relative importance of the error and the expenditure of this energy [36, 43].

The optimal gain vector K is calculated as

$$K = R^{-1}B^T S \tag{4.16}$$

where S is symmetric positive definite matrix which is determined by algebraic Riccati equation as

$$-SA - A^T S - Q + SBR^{-1}B^T S = 0 \tag{4.17}$$

The designer is free to select the matrices Q and R, but the selection of matrices Q and R is normally based on an iterative procedure using experience and physical understanding of our problem.

The optimal feedback control law of LQR controller is then

$$u = -K\mathbf{X}(t) \tag{4.18}$$

so the state equations for the closed-loop system can be rewritten as :

$$\dot{\mathbf{X}} = (A - BK)\mathbf{X} \tag{4.19}$$

$$\mathbf{Y} = C\mathbf{X} \tag{4.20}$$

In this work, Q and R were chosen as follow :

$$Q = \begin{bmatrix} 90 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 85 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix} \tag{4.21}$$

The Q and R matrices were selected based on achieving the requirements of the controller design in terms of settling time and maximum percentage overshoot, and to be able to apply the control law of linear quadratic regulater with the mechanical and electrical system parts that were selected such as the servo motors. These matrices were evaluated according to the obtained simulation results.

After the Riccati Eq. (4.17) is solved, the optimal gains that minimize the objective functional of Eq. (4.15) is computed as [5]

$$K = \begin{bmatrix} -2.1213 & -0.8096 & 0.0000 & 0.0000 \\ -0.0000 & -0.0000 & -2.0616 & -0.7990 \end{bmatrix}$$

Therefore, the corresponding closed loop poles (p) of the system are located in the left side and represented as follow :

$$p = \begin{bmatrix} -2.8365 - 2.6112i & -2.8365 + 2.6112i \\ -2.7994 + 2.5708i & -2.7994 - 2.5708i \end{bmatrix}$$

The state equations for the discrete time optimal linear quadratic controllers are set in Eqs. (4.13)-(4.14). By defining the sampling time $Ts = 10$ ms, the discrete-time poles (P_d) to place the poles at the desired location

[5] We use the matlab function 'lqr' to find the closed loop gain.

will be in the discrete time as :

$$P_d = \begin{bmatrix} 0.9721 + 0.0250i & 0.9721 - 0.0250i \\ 0.9717 - 0.0254i & 0.97170.0254i \end{bmatrix}$$

The discrete time quadratic optimal controller state gain matrix (K_d) is calculated by using Matlab software using built in commands [6], the discrete value of the gain matrix K_d for the LQR controller as follow :

$$K_d = \begin{bmatrix} -2.0620 & -0.7974 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -2.0046 & -0.7871 \end{bmatrix}$$

4.2.2 Tracker Design

In this section, it is desired to design an optimal controller that able to track trajectories that changed over the time as a desired reference input of the ball, that means to stabilize the ball at any position on the plate.

Linear Quadratic Tracker (LQT)

The linear quadratic tracker (LQT) is formulated to reduce a cost function in terms of the plant's states and inputs. The LQT controller typically consist of two terms feed-forward and state feedback . The feed-forward term is needed to provide the optimal tracking of time varying reference trajectories, while the state feedback guarantees system stability by computing the gain matrix as in previous section. These characteristics of solving the problem over time varying make the LQT controller more appropriate than the LQR when trying to accomplish more precise trajectory following. LQT block diagram is shown in Fig. 4.5.

[6] We use the Matlab function 'dlq"' to find the discrete-time system gains and poles.

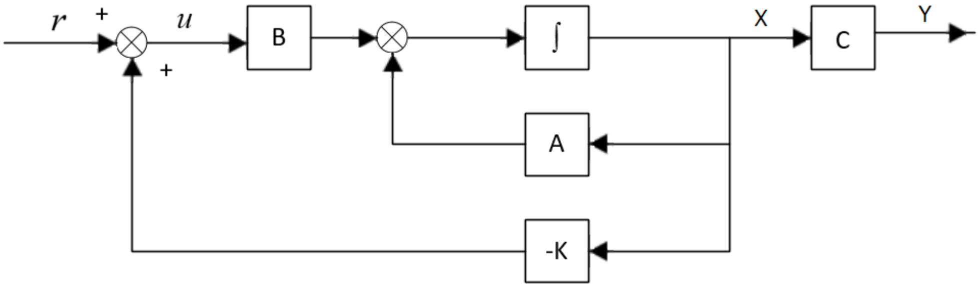


Figure 4.5: LQT schematic diagram.

The LQT controller is applied to our system to find the optimal control input sequence (u) to make the ball on the plate to track a reference trajectory in an optimal manner. This can be obtained by minimizing the finite-horizon performance index (J):

$$J = \frac{1}{2}(C\mathbf{X}-r)^T S(C\mathbf{X}-r)+\frac{1}{2}\int_0^\infty [(C\mathbf{X}-r)^T Q(C\mathbf{X}-r)+u^T Ru]dt \quad (4.22)$$

where $r \in \mathbb{R}^{2\times 2}$ is a reference trajectory.

Therefore, the solution of the finite-horizon LQT problems will be:

$$\mathbf{u} = -KX + R^{-1}B^T v \quad (4.23)$$

where v is acquired with solving algebraic equation

$$v = -((A - Bk)^T)^{-1}C^T Qr \quad (4.24)$$

The specifications design of LQT controller are the same of the specification design of LQR controller. According to that, Q and R matrices were chosen with the same value and the state gain matrix K that are used in the design as follow :

$$K = \begin{bmatrix} -2.1213 & -0.8096 & 0.0000 & 0.0000 \\ -0.0000 & -0.0000 & -2.0616 & -0.7990 \end{bmatrix}$$

ince the specification design and weighting matrices (Q, R) of LQR controller is similar to LQT controller, then the discrete time poles (P_d) and the state gain matrix (K_d) of LQR and LQT are the same as follow :

$$P_d = \begin{bmatrix} 0.9721 + 0.0250i & 0.9721 - 0.0250i \\ 0.9717 - 0.0254i & 0.97170.0254i \end{bmatrix}$$

$$K_d = \begin{bmatrix} -2.0620 & -0.7974 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -2.0046 & -0.7871 \end{bmatrix}$$

4.3 Linear Model Predictive Control (LMPC)

Linear model predictive control (LMPC) is a type of a control algorithm that optimize (minimize) an open loop performance objective by compute a manipulated input profile within utilizing process model that subject to model equations and constraints on states and controls over a future time horizon. This means, at instate time k, we measure and estimate system state variables. Then we compute an optimal control by solving an open-loop optimal control problem [48, 63]. The MPC principle is depicted in Fig. 4.6.

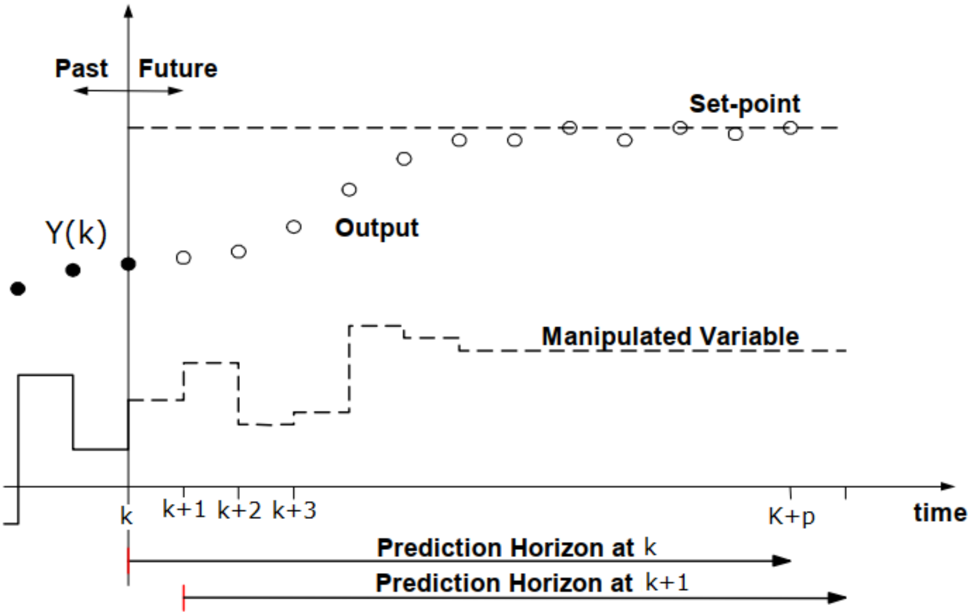


Figure 4.6: MPC Strategy [26].

Consider a discrete-time dynamic system in the state space form:

$$\vec{\mathbf{X}}(k+1) = F\vec{\mathbf{X}}(k) + G\vec{u}(k) \tag{4.25}$$

$$\vec{\mathbf{Y}}(k) = C\vec{\mathbf{X}}(k) + D\vec{u}(k) \tag{4.26}$$

where $k \in N$ is a discrete time instances $(k = 1, 2, 3...)$, $F \in \mathbb{R}^{4\times4} \approx 1 + T_s A$ and $G \in \mathbb{R}^{4\times2} \approx T_s B$ are the state and input matrices, respectively.

The system is subject to the following constraints:

$$u_{min} \le u(k) \le u_{max}, \quad \le |\Delta u(k)|\, \Delta u_{max}, \quad \mathbf{Y}_{min} \le \mathbf{Y}(k) \le \mathbf{Y}_{max}$$

where $u_{min} \le u_{max}$, $\mathbf{Y}_{min} \le \mathbf{Y}_{max}$, and $\Delta u_{max} > 0$ are vectors of upper and lower bounds.

MPC strategy is based on the iterative optimization technique, that means at each sampling time k, MPC estimate the current state, then obtain the optimal input of the system by solving an optimization problem. For the discrete time model, the following cost function can be formulated and used for calculation of the optimal input trajectory over the control horizon:

$$J(k) = \sum_{i=1}^{N_p} Q(i)[\tilde{\mathbf{Y}}(k+i|k) - r(k+i|k)]^2 + \sum_{i=1}^{N_u} R(i)[u(k+i-1|k)]^2 \quad (4.27)$$

where $Q \in \mathbb{R}^{4\times4}$ and $R \in \mathbb{R}^{2\times2}$ are appropriate weighting matrices, $\tilde{\mathbf{Y}}(k+i|k)$ denotes the predicted output, $r(k+i|k)$ denotes the reference trajectory of the desired output signal at sampling instant k, $u(k+i-1|k)$ denotes the vector of control predicted values, N_p is the predictive horizon where $N_p \geq 1$ and N_u is the control horizon where $0 < N_u \leq N_p$.

The predicted state vectors can be obtained by using the previous predicted state vectors [58], which can be computed as follows:

$$\begin{aligned}
\mathbf{X}(k+1|k) &= F(\mathbf{X}(k)) + G(\mathbf{X}(k))(u(k-1) + \Delta u(k|k)), \\
\mathbf{X}(k+2|k) &= F(\mathbf{X}(k+1|k-1)) + G(\mathbf{X}(k+1|k-1))(u(k-1) + \Delta u(k|k) + \Delta u(k+1|k)), \\
&\vdots \\
\mathbf{X}(k+N_p|k) &= F(\mathbf{X}(k+N_p-1|k-1)) + G(\mathbf{X}(k+N_p-1|k-1)) \\
&\quad (u(k-1) + \Delta u(k|k) \cdots \Delta u(k+N_p-1|k)) \quad (4.28)
\end{aligned}$$

Define following vectors:

$$\tilde{\mathbf{Y}}(k) = [\mathbf{Y}(k|k) \;\; \mathbf{Y}(k+1|k) \;\; \cdots \;\; \mathbf{Y}(k+N_p-1|k)]^T,$$

$$\tilde{u}(k) = [u(k|k) \;\; u(k+1|k) \;\; \cdots \;\; u(k+N_p-1|k)]^T,$$

$$\tilde{\mathbf{X}}(k) = [\mathbf{X}(k|k) \;\; \mathbf{X}(k+1|k) \;\; \cdots \;\; \mathbf{X}(k+N_p-1|k)],$$

$$r(k) = [r(k|k) \;\; r(k+1|k) \;\; \cdots \;\; r(k+N_p-1|k)]^T$$

The linear predictor of the system output in matrix form based on the state space description of the system (4.25) is defined as [58]:

$$\tilde{\mathbf{Y}}(k) = W_0\mathbf{X}(k) + E_0u(k) \tag{4.29}$$

where W_0 represent the response of the system and E_0 represent the forced response of the system. Both matrices have form:

$$W_0 = \begin{bmatrix} C \\ CF \\ \vdots \\ CF^{N_p} \end{bmatrix}$$

$$E_0 = \begin{bmatrix} D & 0 & \cdots & & 0 \\ CG & D & & & \\ CFG & CG & D & & \vdots \\ \vdots & & & \ddots & 0 \\ CF^{N_p-1}G & & CFG & CG & D \end{bmatrix}$$

Substitute Eq. (4.29) in Eq. (4.27), we obtain the the cost function of the predictor as follows:

$$J(k) = [W_0X(k|k)+E_0u(k|k)-r(k|k)]^TQ(W_0X(k|k)+E_0u(k|k)-r(k|k))+u(k|k)^TRu(k|k) \tag{4.30}$$

Based on the derivation of the vectors according to [58] with minimizing the cost function of the predictor with the condition $\frac{\partial J(k)}{\partial u(k)} = 0$, the final form of the optimal control law is as :

$$u_{optimal} = \frac{1}{2}(E_0^T Q E_0 + R)^{-1}((Y(k) - r(k))^T Q E_0) \tag{4.31}$$

The optimal control of MPC controller is computed by optimization method based on quadratic programming, which are implemented in the Matlab simulink environment as the (MPC toolbox). The optimal control law subject to the inputs constraints is given by the minimization of the relation:

$$min(\frac{1}{2}u(k)^T(E_0^T Q E_0 + R)u(k) + (Y(k) - r(k))^T Q E_0 u(k)) \tag{4.32}$$

where $(E_0^T Q E_0 + R)$ and $((Y(k) - r(k))^T Q E_0)$ are denotes as the Hessian and gradient matrices.

4.4 Observer Design

The state space controllers assumes that all states of the system are measured and available. However, we only measure the position of the ball $(x,\ y)$ directly. For this reason, an observer must be used to calculate or to estimate the state variables that are not reachable directly from the plant. Here, the observer is a model of the plant, which is done by creating a copy of the system linear dynamic equations, and adding an output term. Furthermore, the use of state space observer reduces the noise data by dealing with system mathematical equations away from derivation and integration.

Fig. 4.7 shows the basic concept of observer design, where the measured outputs of the BPS $\mathbf{Y}$ are compared with the estimated outputs of the observer $\tilde{\mathbf{Y}}$, and then the estimated error output signal is fed back to the observer.

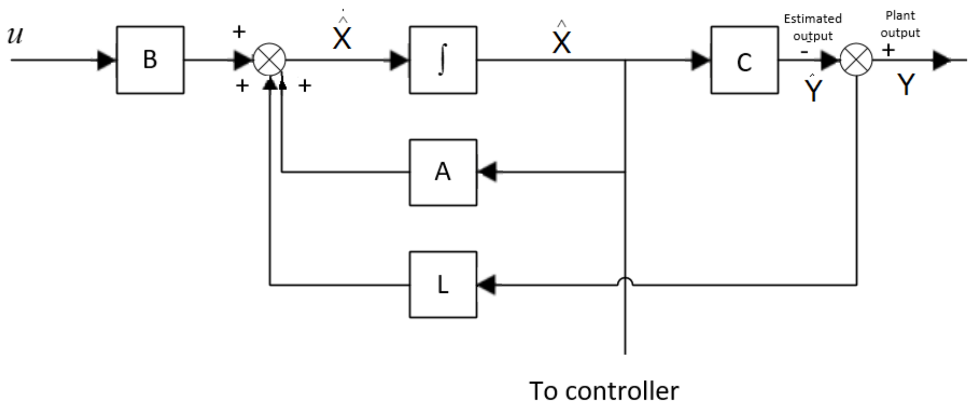


Figure 4.7: Observer design process.

The state equation of the observer is found in Fig. 4.7 as follows:

$$\dot{\tilde{\mathbf{X}}} = A\tilde{\mathbf{X}} + B\mathbf{u} + L(\mathbf{Y} - \tilde{\mathbf{Y}}) \tag{4.33}$$

$$\tilde{\mathbf{Y}} = C\tilde{\mathbf{X}} + D\vec{\mathbf{u}}(k) \tag{4.34}$$

where $\mathbf{X} \in \mathbb{R}^4$ and $\tilde{\mathbf{Y}} \in \mathbb{R}^2$ are the estimated system states and estimated output measurements vectors, respectively, and $L \in R^{4\times2}$ is the observer gain matrix.

The error signal between the measured output Y and the observer output $\tilde{\mathbf{Y}}$ is:

$$\tilde{e} = \mathbf{X} - \tilde{\mathbf{X}} \tag{4.35}$$

Thus from Eq. (4.33) and Eq. (4.7)we get

$$\dot{\mathbf{X}} - \dot{\tilde{\mathbf{X}}} = (A - LC)(\mathbf{X} - \tilde{\mathbf{X}}) \tag{4.36}$$

Now, the error dynamic equation is:

$$\dot{\tilde{e}} = (A - LC)\tilde{e} \tag{4.37}$$

However, the dynamics of the observer must be made much faster than the dynamics of control system, so the poles of the error characteristic equation Eq. (4.37) must be place faraway to the left from those of the controlled system to achieve the desired speed of the observer, which can be achieved by choosing an appropriate gain vector L [51]. By using pole placement method or optimal control method, the gain vector L can be obtained in a similar way to the feedback gain, where observer poles are selected to be four to ten times faster than the poles of system.

However, to check the observability, if the system states can be found by the knowledge and observing the input $u(t)$ and output $\mathbf{Y}(t)$, then the system is said to be observable, otherwise the system is said to be unobservable [8]. The observability of the pair matrix A, C can be checked by the observability matrix O_M, such that:

$$O_M = [C \quad CA \quad CA^2 \quad CA^3]^T \tag{4.38}$$

We also use the built in commands in the Matlab to check the observability [7] which also implies that the BPS is observable.

[7] We use the Matlab function '`obsv`' and '`rank`' commands to find the rank and observ-

The poles of observer should be faster ten times than the poles of controller as mentioned previously, so the poles of observer P_o is:

$$P_o = \begin{bmatrix} -19.7920 + 20.1919 & -19.7920 - 20.1919i & -25.1327 + 18.8496ii & -25.1327 - 18.8496i \end{bmatrix}$$

By using the pole placement and optimal methods respectively within Matlab command [8] , the gains vector L are found to be:

$$L = \begin{bmatrix} 40.0000 & -5.1522 \\ 1056.0000 & -584.0000 \\ -1.8000 & 41.1398 \\ 81.3000 & 1056.0000 \end{bmatrix}$$

$$L = \begin{bmatrix} 24.2499 & 0 \\ 2.5451 & 0 \\ 0 & 23.5726 \\ 0 & 2.5451 \end{bmatrix}$$

4.4.1 Discrete Time Observe Design

In order to implement the observer on the Arduino microcontroller, we need to discretize the continuous time observer to discrete time observer with discrete time instances k. The observer equations in the discrete time are set as follows:

$$\mathbf{X}(k\tilde{+}1) = F\tilde{\mathbf{X}}(k) + G\vec{u}(k) + L_d(\vec{\mathbf{Y}} - \tilde{\mathbf{Y}}(k)) \tag{4.39}$$

ability of the system.

[8] We use the Matlab function 'place' and 'lqr' commands to find the closed loop poles of the system.

$$\tilde{\mathbf{Y}}(k) = C\tilde{\mathbf{X}}(k) + D\vec{u}(k) \tag{4.40}$$

By subtract Eq. (4.39) from Eq. (4.13), we get the final observer error dynamic equation :

$$\vec{e}(k+1) = (F - L_d C)\vec{e}(k) \tag{4.41}$$

The discrete-time observer schematic is shown in Fig. 4.8.

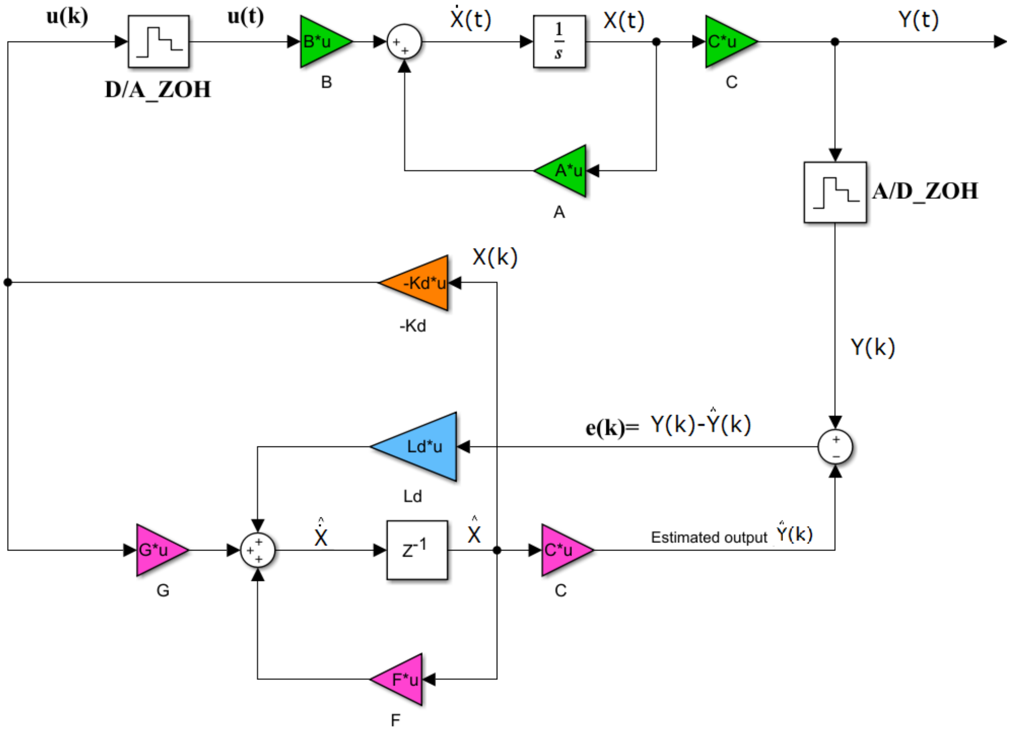


Figure 4.8: Discrete-time observer.

Since the pair (A, B) is observable, so the pair of matrices (F, G) is observable. To get the discrete time observer gain vector (L_d), the pole placement and optimal methods respectively are used through Matlab software:

$$L_d = \begin{bmatrix} 0.4234 & 0.0119 \\ 8.6692 & 0.5949 \\ -0.0116 & 0.4156 \\ -0.5857 & 8.6186 \end{bmatrix}$$

$$L_d = \begin{bmatrix} -0.1487 & -0.0006 \\ 0.6287 & 0.0043 \\ 0.0000 & -0.1482 \\ -0.0016 & 0.6264 \end{bmatrix}$$

In order to implement the discrete observer, we write the discrete state space in explicit equations. Eq. (4.35) can be as;

$$\tilde{X}(k+1) = (F - L_d)\tilde{X}(k) + Gu(k) + L_dY(k) \tag{4.42}$$

Therefor, Eq. (4.42) will be rewritten as follows

$$\begin{bmatrix} \tilde{X_1}(k+1) \\ \tilde{X_2}(k+1) \\ \tilde{X_3}(k+1) \\ X_4(\tilde{k}+1) \end{bmatrix} = \left(\begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{14} \\ F_{21} & F_{22} & F_{23} & F_{24} \\ F_{31} & F_{32} & F_{33} & F_{34} \\ F_{41} & F_{42} & F_{43} & F_{44} \end{bmatrix} - \begin{bmatrix} L_{d11} & L_{d12} \\ L_{d21} & L_{d22} \\ L_{d31} & L_{d32} \\ L_{d41} & L_{d42} \end{bmatrix} \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \end{bmatrix} \right) \begin{bmatrix} \tilde{X_1}(k) \\ \tilde{X_2}(k) \\ \tilde{X_3}(k) \\ \tilde{X_4}(k) \end{bmatrix}$$

$$+\begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}+\begin{bmatrix} L_{d11} & L_{d12} \\ L_{d21} & L_{d22} \\ L_{d31} & L_{d32} \\ L_{d41} & L_{d42} \end{bmatrix}\begin{bmatrix} Y_1(k) \\ Y_2(k) \end{bmatrix} \tag{4.43}$$

Chapter 5

Simulation Results

The next step after controller design process is simulation. The simulation step is very important to check wether the response of the simulation result meets the design specification or not. MATLAB and simulink platforms are used to simulate system response and performance based on the controller gain martix (K) and observer gain vector (L) that are obtained in Chapter 4, with the nonlinear system model that is derived in Chapter 3.

Since the equations of motion of the x and y axis are similar, the results that were applied to the x axis are represented only. Initially, a step resposne of magnitued 0.1 m is simulated, which mean the ball was required to track the position from (0,0) to (0.1,0.1). Then, a sine wave is simulated to represent motion along x axis with radius $= 0.1$ m for all different type of controllers.

5.1 PID Controller

In order to stabilize the ball on a desired position with respect to the requirements, a PID controller are designed for x -axis and another one for y -axis as they are expressed in two separate differential equations. Analysis of the motion along the x - and y -axes is similar and hence only the former is presented. Fig. 5.1 shows the simulation result for step response with disturbance at second 6 for obtained controller for the position of the ball with velocity and angular velocity along x - axis. The controller parameters that are used in the simulation as following; $K_P = 1.69$, $K_I = 1.12$ and $K_D = 0.662$.

Fig. 5.1 shows the response of ball with approximated settling time of 3.13 sec which is does not meet the design requirement, a random disturbance was applied to the controller at the second 6. Table 5.1 includes a comparison between the simulated result and desired response.

Table 5.1: Compared performance specifications for PID controller.

-	Design	Simulated
Percentage overshoot	10	18.3
Settling time (sec.)	2.1	3.13
Steady state error	0	0

In Fig. 5.2 , the simulation result for the tracking of a desired sine wave is presented for the position of the ball with velocity and angular velocity along x - axis with some arbitrary disturbances.

5.2 State Feedback Controller

The second control method that is tested within simulation result is a state feedback controller. The simulation object is to stabilize the ball on the center

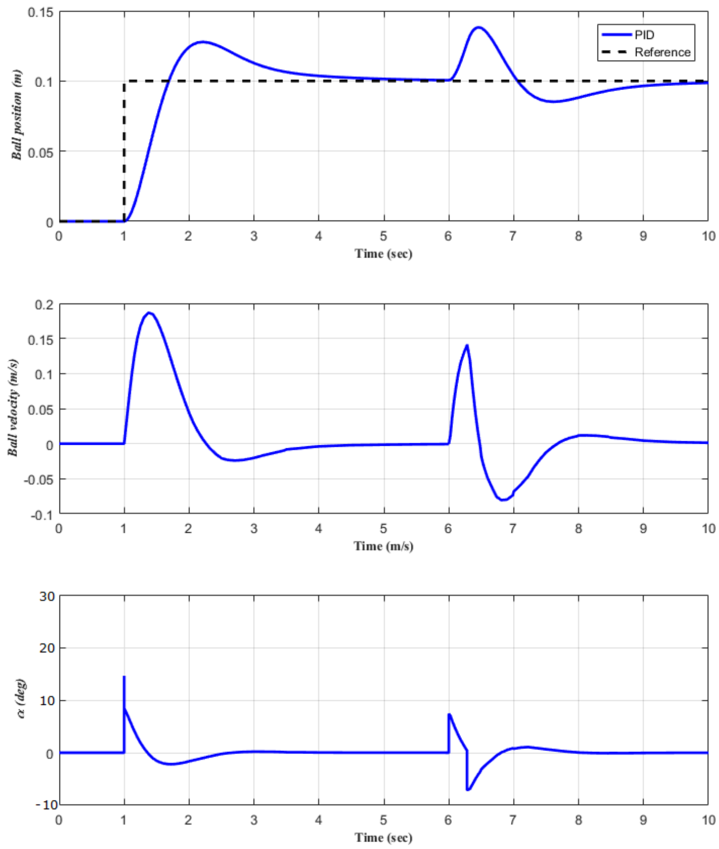


Figure 5.1: Simulated response for ball position, velocity and angle α by using a PID controller for tracking a step reference along x -axis with disturbance.

point at the plate with reject any disturbances that effect on the system.

To satisfy the system requirements , the state feedback gain K:

$$K = \begin{bmatrix} -1.5470 & -0.6062 & -0.0643 & -0.0167 \\ 0.0431 & -0.0046 & -1.4858 & -0.5497 \end{bmatrix}$$

Fig. 5.3 shows the simulation result for stabilizing the ball on a desired position for step response with a random disturbance at the second 6, the position of the ball with the velocity and inclination plate on the x - axis was presented in the result .

Fig. 5.3 shows the response of the ball with settling time of 2.105 sec

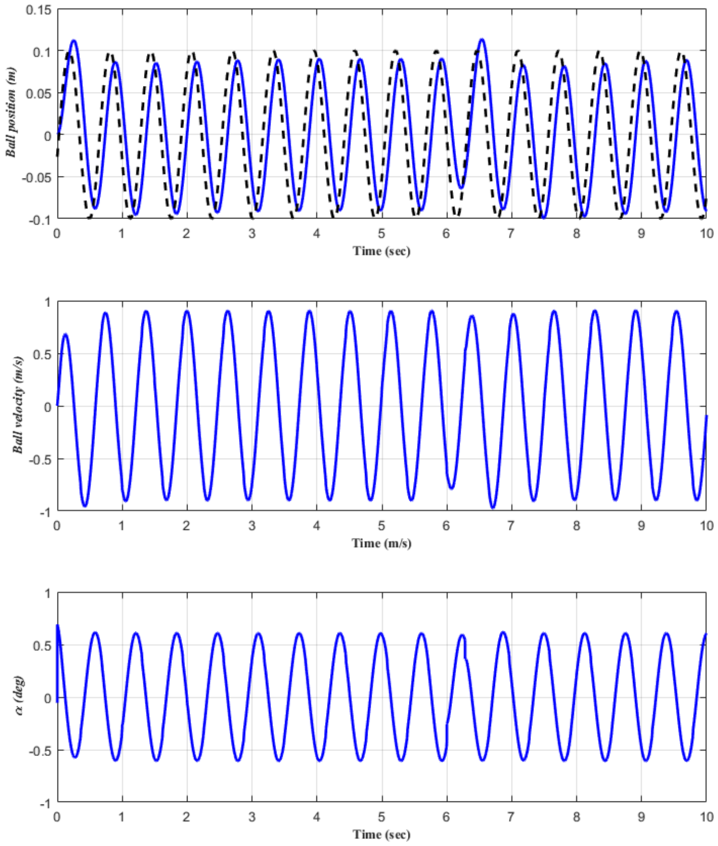


Figure 5.2: Simulated response for ball position, velocity and angle α by using a PID controller for tracking of sine wave along x -axis with disturbance.

which approximately meets the design requirement. Table 5.2 includes a comparison between the simulated result and desired response.

Table 5.2: Compared performance specifications for state feedback controller.

-	Design	Simulated
Percentage overshoot	10	6.36
Settling time (sec.)	2.1	2.105
Steady state error	0	0

In Fig. 5.4, the tracking of desired sine wave are presented with arbitrary disturbance.

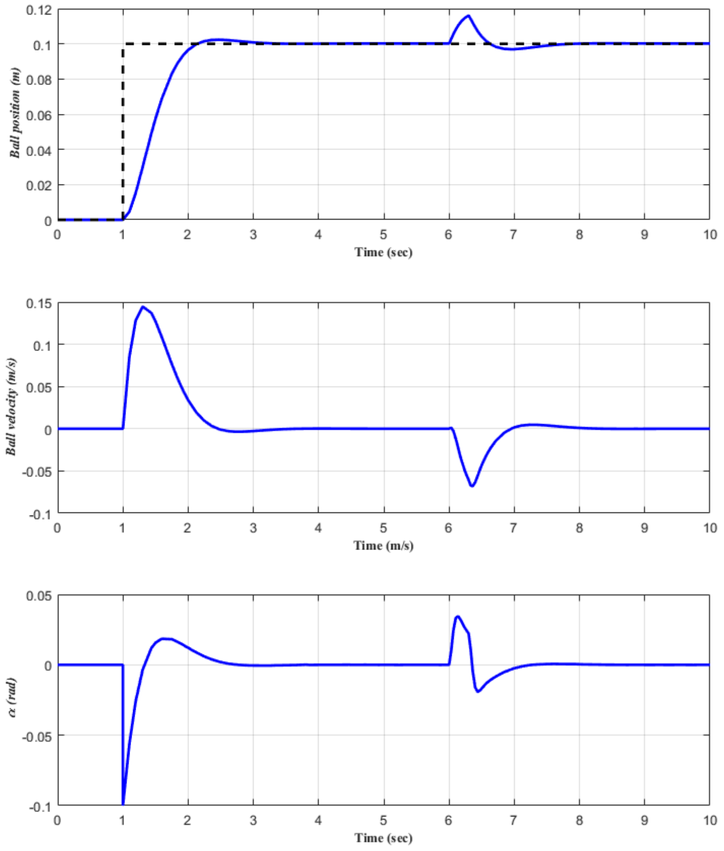


Figure 5.3: Simulated response for ball position, velocity and angle α by using a state feedback controller for a step input along the x -axis with disturbance.

5.3 LQR and LQT Controller

Since the specifications design and weighting matrices Q and R of LQR controller are simillar to LQT controller, then the simulation results of the both controllers are identical. Here, the feedback gain of the both controllers are based upon the minimization of a quadratic cost function. We chose the Q, R matrices as in Eq. (4.21).

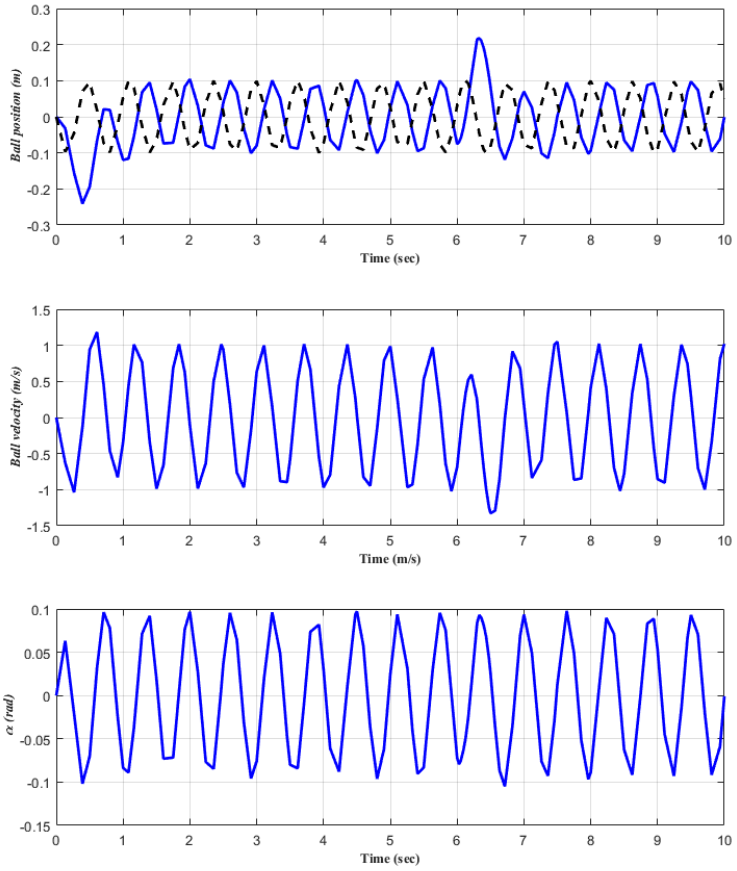


Figure 5.4: Simulated response for ball position, velocity and angle α by using a state feedback controller for tracking of sine wave along x -axis with disturbance.

$$Q = \begin{bmatrix} 90 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 85 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, R = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}$$

The state-feedback gain K is

$$K = \begin{bmatrix} -2.1213 & -0.8096 & 0.0000 & 0.0000 \\ -0.0000 & -0.0000 & -2.0616 & -0.7990 \end{bmatrix}$$

Simulation results of the proposed controllers for x - axis dynamics are presented in Fig. 5.5 for fixed step reference point, and in Fig. 5.6 for time-varying reference trajectory of tracking sine wave. From Fig. 5.5, the settling time is exactly equals to 1.956 second which meets to the design requirements.

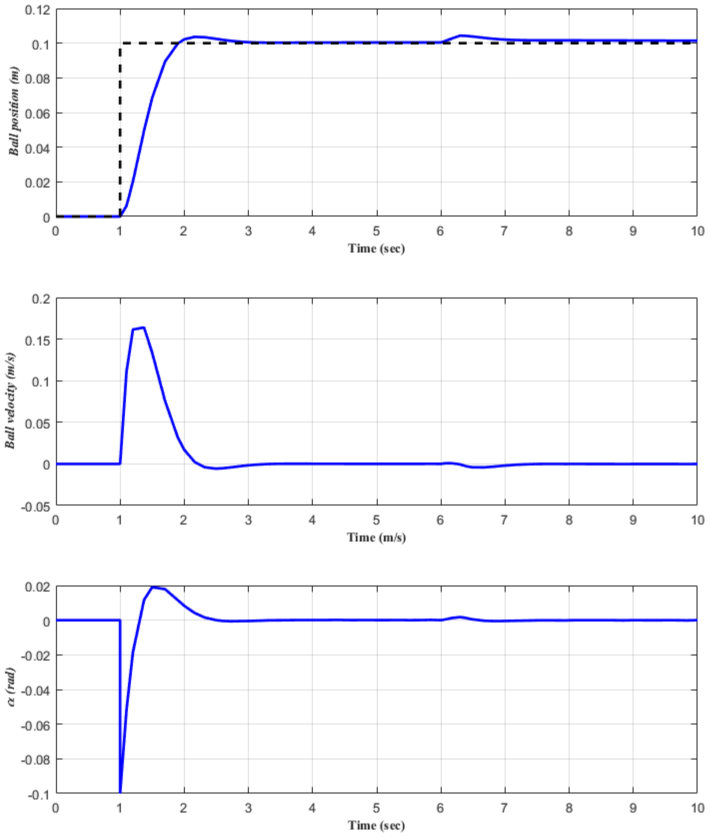


Figure 5.5: Simulated response for ball position, velocity and angle α by using LQR and LQT controller for a step step reference along x -axis with disturbance.

Table 5.3 compares between the simulated result with desired response designed.

Table 5.3: Compared performance specifications for LQR and LQT controller.

-	Design	Simulated
Percentage overshoot	10	3.006
Settling time (sec.)	2.1	1.965
Steady state error	0	0

5.4 LMPC Controller

The last control method that is tested within simulation result is a LMPC. The simulation object is to stabilize the ball on the center point at the plate with reject any disturbances that effect on the system. Here, the simulation parameters are listed in table 5.4

Table 5.4: Simulation parameters for LMPC.

Description	Value	Unit
Simulation time	10	sec.
Sample period	0.01	sec.
Prediction horizon	10	samples
Control horizon	2	samples

Fig. 5.7 shows the response of ball with approximated settling time of 1.83 sec. which is meet the design requirement. Table 5.5 includes a comparison between the simulated result and desired response.

Table 5.5: Compared performance specifications for LMPC.

-	Design	Simulated
Percentage overshoot	20	2.7
Settling time (sec.)	2.1	1.83
Steady state error	0	0

In Fig. 5.8 , the simulation result for the tracking of a desired sine wave is presented for the position of the ball with velocity and angular velocity along x - axis with some arbitrary disturbances.

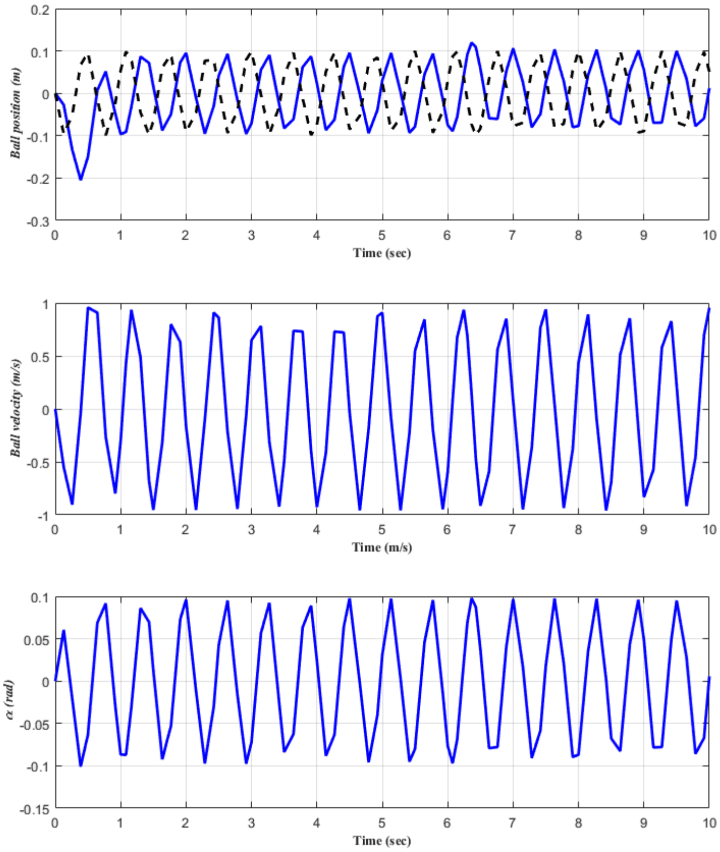


Figure 5.6: Simulated response for ball position, velocity and angle α by using LQR and LQT controller for a sine wave trajectory along x -axis with disturbance.

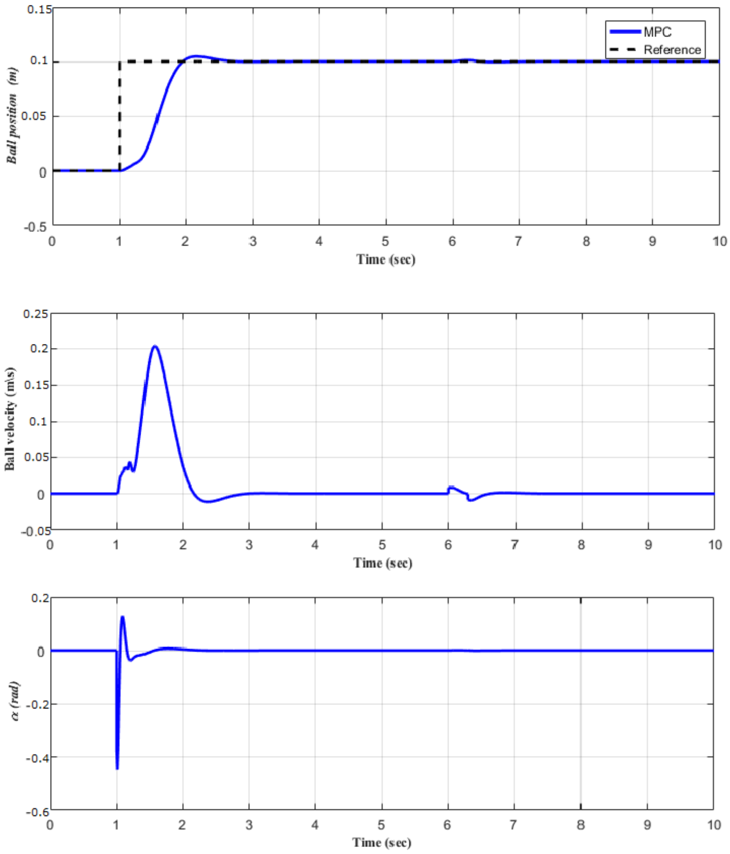


Figure 5.7: Simulated response for ball position, velocity and angle α by using LMPC for a step reference along x -axis with disturbance.

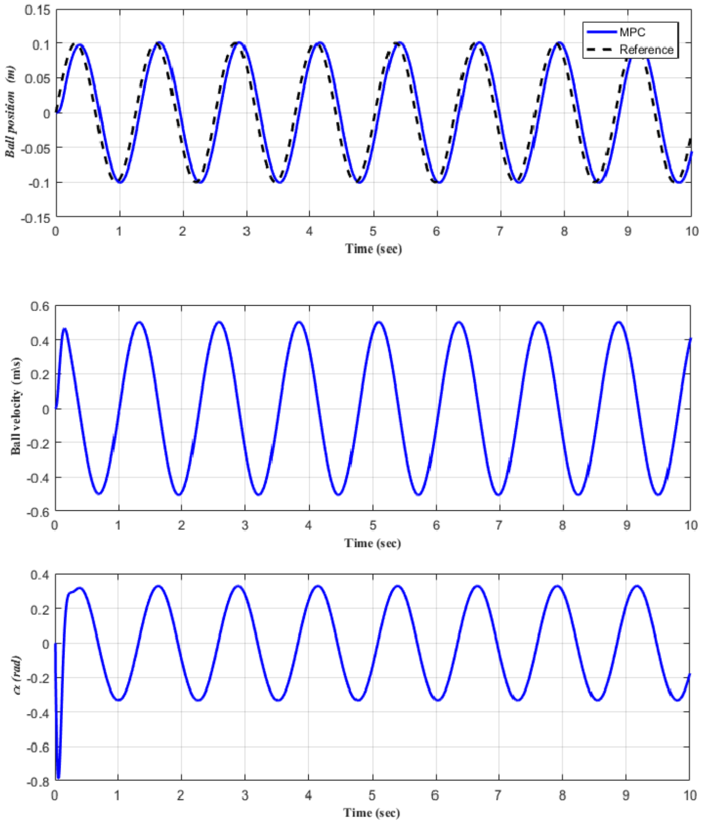


Figure 5.8: Simulated response for ball position, velocity and angle α by using LMPC for a sine wave trajectory along x -axis with disturbance.

Chapter 6

Experimental Results

The state variables of the BPS such as the ball position (x, y), ball velocity $(\dot{x}, \dot{y})$, plate deflection angle (α, β) and deflection angle velocity $(\dot{\alpha}, \dot{\beta})$ can be obtained from the system as a feedback data. These data can be used to analyze the performance of the system and can be plotted to demonstrate and analyze the response of the system to stabilizing the ball and to the external disturbances. The process of transferring data from Arduino microcontroller to another software affects on the stability of the system and increase measurements error because this process requires an additional communication time. In order to reduce the measurements error, the feedback digital data are sent directly from Arduino serially to the Excel program to save the quantified digital data, then the "MATLAB" program is used to draw the outputs [50, 38]. For LMPC strategy, the Matlab program was used directly to save and draw the outputs.

In the BPS, each plate side is supported mainly through the pivot at the center with a deticated servomotor. The plate deflection angles (α and β) are changed from these two servomotor according to feedback state variables in which to move the ball to roll to the center of the plate or to follow a predefined trajectory. Initially, the ball begins rolling on the plate at specified or at any location at the plate, then the controller will move the ball to roll to a stable location or to follow a predefined trajectory. To test the stability and the performance of system, a random disturbance is applied on the ball spontaneously.

There are two types of experiment that are performed to verify the BPS performance. The first experiment is using a random location stabilization to set the ball at the center of the plate, this experiments is used for some of control strategy such as (MPC, PID, LQR and state feedback controller). The second experiment is using curve trajectory tracking to create a circle and square tracking on the plate for some of control approach. The trajectory experiment is used for tracker controller like (MPC, PID and LQT controller). In Arduino microcontroller, there are two sampling rates that are used to perform the control strategy. The first sampling rate is 1 ms using accurate timing loop based on microcontroller internal clock for acquiring the ball position from touch screen. The second sampling rate is 10 ms using the internal interrupt to perform the control action of the control method.

In order to perform the different control strategy, there are some problems that must be solved, namely:

1. The softness surface of the touch screen affects on the ball speed and orientation, and the speed unifirmity may lead to control difficulties.

2. The inclination of the plate is limited ($\pm 30^\circ$), so the contol action is constrained.

3. Different nonlinear factors in the system mechanism, which may affect to the stability of the system.

The BPS system presented in Fig. 6.1 is used to validate the dynamical model and to perform the experimental tests of the different control system designed in this thesis.

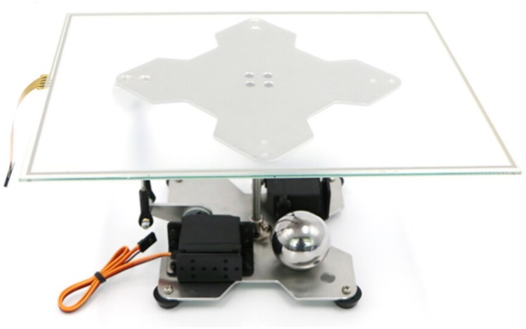


Figure 6.1: BPS.

6.1 Stability Test

For stability test, regardless of the initial conditions of the system, the ball position (x,y) should be enforced to move to at the expected location on the plate.

In this test, the ball will move smoothly to the central point of the plate. In order to verify and to evaluate the performance of our different controllers, the initial x and y ball position have been set as [$x(0)$, $y(0)$]=[0.08, 0.05]m. All other states have been initialized to zero. After stabilizing the ball at the central point of the plate, bounded disturbances are applied to the system.

6.1.1 PID Controller

Using the PID controller, the BPS was tested to control the ball on the central point. The actual value of controller gain that are used on the real system are; $K_P = 1.69$, $K_I = 1.12$ and $K_D = 0.662$.

Figure 6.2 and Figure 6.3 show the response of position and velocity on the x and y - axes for stabilizing the ball at a central position using the PID controller. Initially, it is obvious the system starts to oscillate for a long time to stabilize the ball on the central point. The system needs for 4.8 sec. to settle the ball with an error less than 3 mm and velocity of 0 mm/s on the central point. Then, the ball was pushed by the hand to the third qudrant, but after 6 seconds, the ball was returned to the central point again. The plate deflection angles (α,β) are not settled at 0 ° along the x and y - axes ,which means that the controller is affected by the small disturbances as shown in Figure 6.4. The ball position trajectory on the xy-plane is presented on the Figure 6.5.

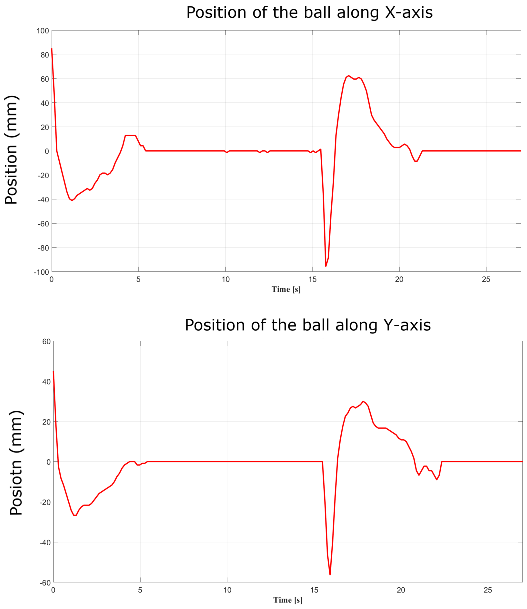


Figure 6.2: x-y position of ball with disturbance for PID controller stability test.

Table 6.1 compares between the actual result of PID controller with the simulated result and the value that are used in the design.

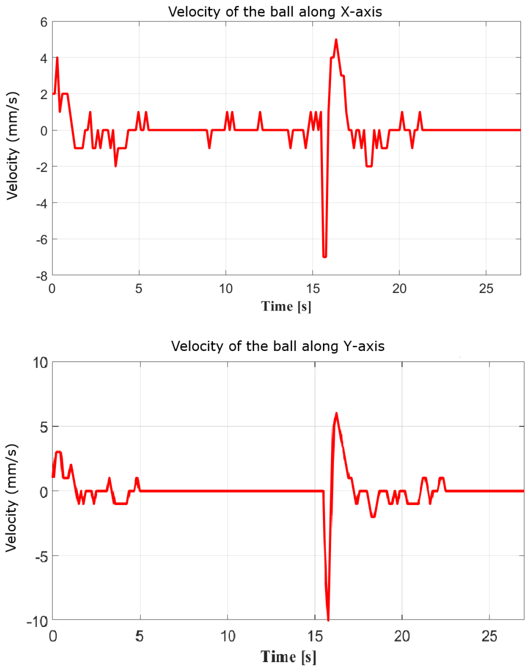


Figure 6.3: Velocity of the ball along x-y-axis for PID controller stability test.

Table 6.1: Performance specifications with simulated and real results for PID controller.

-	Design	Simulated	Actual
Percentage overshoot	10	18.3	26.8
Settling time (sec.)	2.1	3.13	4.8
Steady state error	0	0	3

6.1.2 State Feedback Controller

The ball is moved to the central point of the plate by applying the state feedback controller with controller gains K as

$$K = \begin{bmatrix} -1.5145 & -0.6011 & -0.0615 & -0.0167 \\ 0.0410 & -0.0047 & -1.4574 & -0.5465 \end{bmatrix}$$

By applying this controller, the ball and plate start oscillate for 2.87 sec., the ball is then settled at the plate's center with an error less than 0.6mm

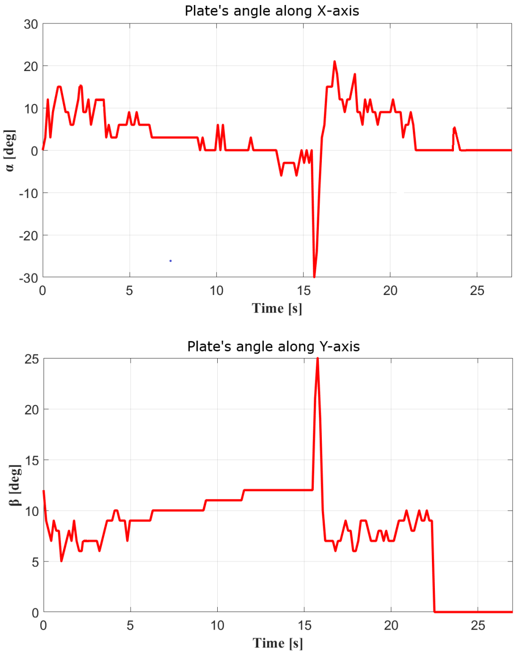


Figure 6.4: Plate deflection angles (α,β) along x-y-axis for PID controller stability test.

and ball velocity of 0 mm/s. The plate's angle are settled at 0°. As an input disturbance, e.g., the ball was pushed by a hand toward the third quadrant at the 14^{th} sec. to check the robustness of this controller. The BPS oscillated again and then settled after 1.6 sec., the ball return to center of the plate again. Figure 6.6 and Figure 6.7 show the response on the x and y axes for stabilizing the ball at the central location with the velocity of the ball on the time domain, while the plate deflection angles (α, β) are demonstrated in the Figure 6.8, it should be noted that the system needs to tilt the plate to the maximum possible degree in order to back the ball to the central position, which means a high exerting torque is needed . The ball position trajectory with disturbance on the x and y axes is shown in the Figure 6.9.

Table 6.2 compares between the actual result of state feedback controller with the simulated result and the value that are used in the design.

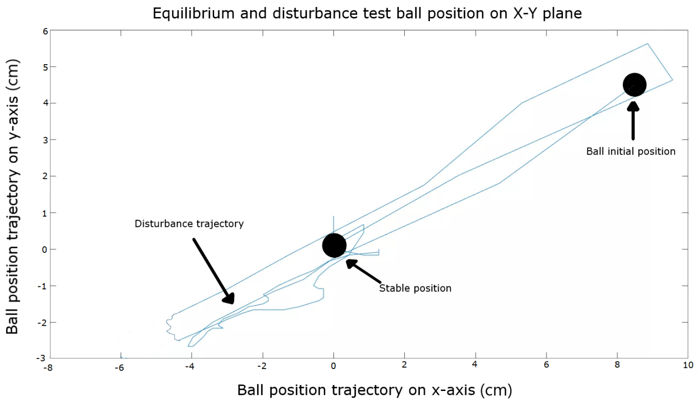


Figure 6.5: System equilibrium with disturbance for ball position for PID controller stability test.

Table 6.2: Performance specifications with simulated and real result for state feedback controller.

-	Design	Simulated	Actual
Percentage overshoot	10	6.36	11.30
Settling time (sec.)	2.1	2.105	2.87
Steady state error	0	0	0.6

6.1.3 LQR Controller

LQR controller is applied here to the real system. This controller tends to stabilize the ball on the central position of the plate by the lowest torque exerted as quickly as possible. The actual controller gains K that are used on the real system upon the calculation of Chapter 4:

$$K = \begin{bmatrix} -2.0620 & -0.7974 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -2.0046 & -0.7871 \end{bmatrix}$$

We test the performance of the static position tracking to the central point using the LQR controller. Figure 6.10 shows the response of the position of the ball on the x and y - axes. Initially, the system starts to oscillate for about 1.92 sec. to stabilize the ball on the central point with an error less than 1.3

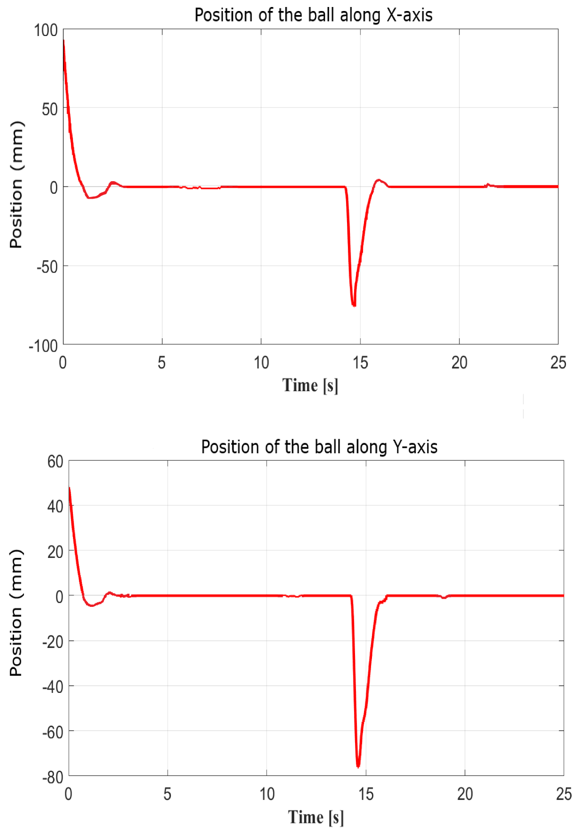


Figure 6.6: x-y position of ball with disturbance for state feedback controller stability test.

mm and ball velocity of 0mm/s and the angle of plate's are settled at 0 °. At the 11^{th} sec., the ball was pushed by the hand to the third quadrant, but after 1.86 sec., the ball returned to the plate's central location again. Figure 6.11 shows the velocity of the ball on the x and y axes. The plate deflection angles α, β are shown in Figure 6.12, we can see that, LQR controller uses less torque to stabilize the system, so the maximum angle is needed since stabilize the system is 20°.

The trajectory of the ball position with disturbance on the x and$y$ axes is shown in Figure 6.13.

Table 6.3 compares between the actual result of LQR controller with the simulated result and the value that are used in the design.

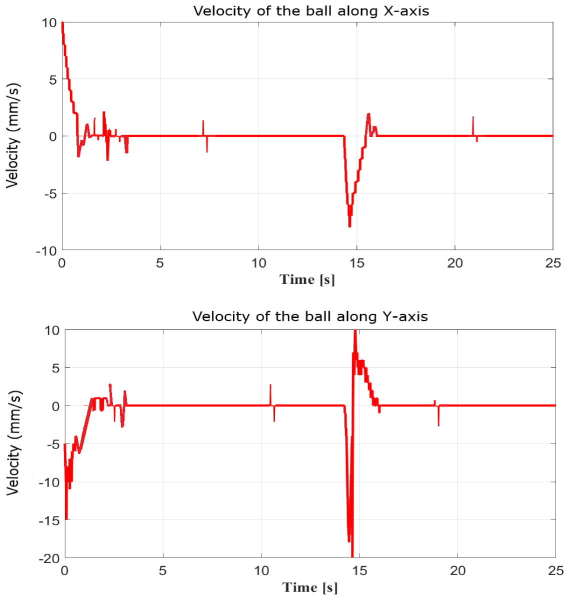


Figure 6.7: Velocity of the ball along x-y-axes for state feedback controller stability test.

Table 6.3: Performance specifications with simulated and real result for LQR controller.

-	Design	Simulated	Actual
Percentage overshoot	10	3.006	11.9
Settling time (sec.)	2.1	1.965	1.84
Steady state error	0	0	1.3

6.1.4 LMPC Controller

LMPC strategy is applied here to the real system. This controller tends to stabilize the ball on the central position of the plate by the lowest torque exerted as quickly as possible. The LMPC parameters that are used on the real system are; sampling time= 0.01 sec., prediction horizon = 10 samples and control horizon = 2 samples. These horizon lengths are set to use the maximum amount of memory allowed by the Matlab software.

By applying this controller for the static position tracking to the central point , the ball and plate start oscillate for about 1.09 sec., the ball is then settled at the plate's center with an error less than 1 mm and ball velocity

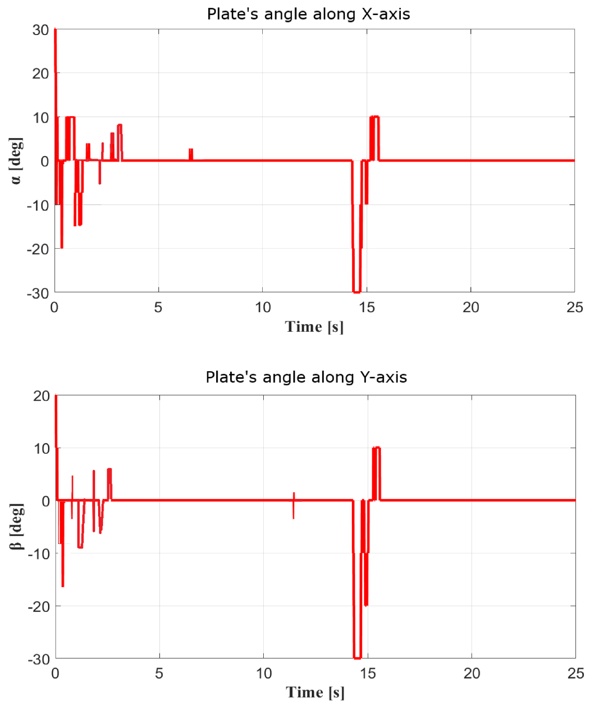


Figure 6.8: Plate deflection angles (α,β) along x-y-axes for state feedback controller stability test.

of 0 mm/s. The plate's angle are settled at 0°. The ball was pushed by the hand as an input disturbance toward the third quadrant at the 14^{th} sec. to check the performance of this controller. The BPS oscillated again and then settled after 2.26 sec., the ball return to center of the plate again. Figure 6.14 and Figure 6.15 show the response on the x and y axes for stabilizing the ball at the central location with the velocity of the ball on the time domain, while the plate deflection angles (α, β) are demonstrated in the Figure 6.16, it should be noted that the system needs to tilt the plate by 18° to back the ball to the central position, which means less exerting torque is needed . The ball position trajectory with disturbance on the x and y axes is shown in the Figure 6.17.

Based on the result that are obtained from LMPC and comparing it with the results of other controllers, we note the extent of LMPC controller

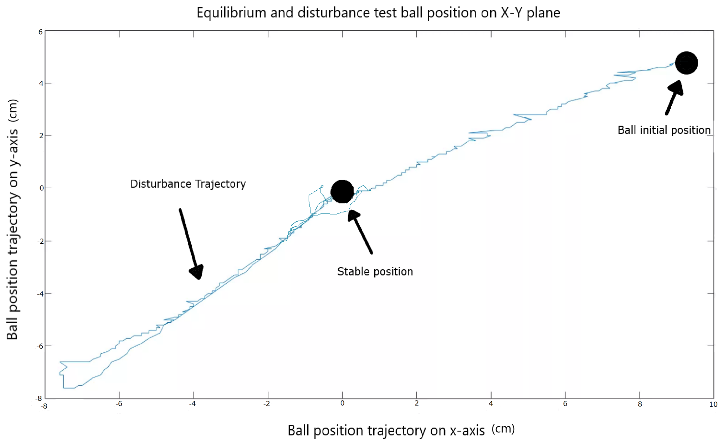


Figure 6.9: System equilibrium with disturbance for ball position for state feedback controller stability test.

superior performance compared to other controllers for the use of real time control with Matlab simulink. Real time contol brings two main adanteges: reduced time for control and it can be easy to change and update the real model according to the real measured data andit allows to obtain information about the real process in a fast way.

Table 6.4 compares between the actual result of LMPC with the simulated result and the value that are used in the design.

Table 6.4: Performance specifications with simulated and real results for LMPC controller.

-	Design	Simulated	Actual
Percentage overshoot	10	2.7	3.1
Settling time (sec.)	2.1	1.83	1.09
Steady state error	0	0	1

A comparison between the five selected methods of control to stabilize the ball in a stable position is presented in Table 6.5. We can say that the LMPC algorithm has the best performance with settling time of 1.09 sec. and an error less than 1 mm, and had the best performance with respect of optimal of used energy of the BPS with 11 ° of plate deflection angle. Although the

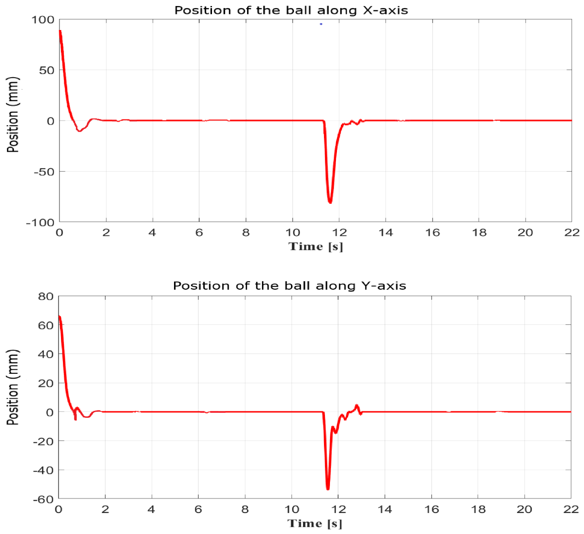


Figure 6.10: x-y position of ball with disturbance for LQR controller stability test.

LMPC algorithm showed the best results for stabilizing the ball in a stable position in terms of the least posible time to recover the ball at the specific positionand the lowest possible angle to tilt the plate, the LMPC algorithm required to use a personal computer with specific speifications like a high performance of central processing unit and random access memory in order to control the BPS, in addition to using an Arduino uno to transfer data to and from the personal computer.

6.2 Trajectory Tracking Test

Two types of the trajectories are selected for these tests: circle and rectangular trajectories. To make the ball follows a circular trajectory in the x-y plane, two arbitrary sinusoidal signals trajectories are applied to the system along x axis ($x_d = M \sin qt$) and along y axis ($y_d = M \cos qt$), where M =0.065m and q=1rad/s. However, in the square trajectory, there are four points which represent corners of the square. To move from one point to an-

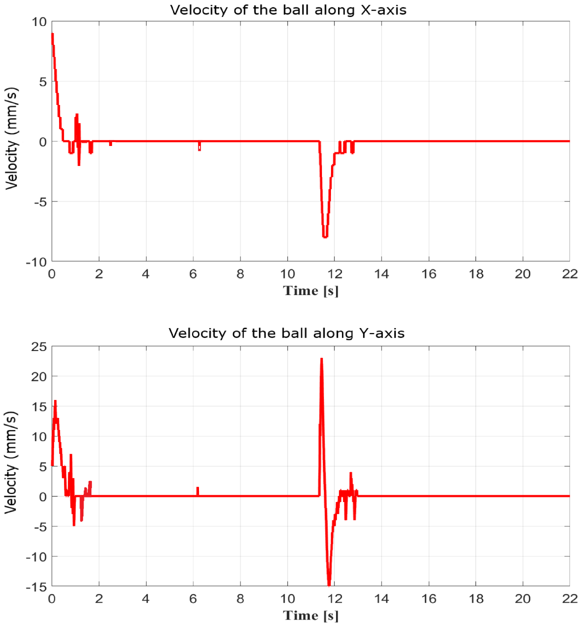


Figure 6.11: Velocity of the ball along x-y-axes for LQR controller stability test.

other point, the control will change only one coordinate of the ball position with the fixed value of the other coordinate

6.2.1 LQT Controler

The linear quadratic tracker (LQT) was tested to track a circular trajectory. The actual controller gains for LQT are:

$$K = \begin{bmatrix} -2.0620 & -0.7974 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -2.0046 & -0.7871 \end{bmatrix}$$

The LQT controller tracking time for a circle trajectory is about 4.13 sec. . Figure 6.18 shows the tracking profile for motion along the x and y axes. The output position of the trajectory tracking along x and y axes gives a circular trajectory with a radius of 0.065 m with an error less than 2 mm as shown in Figure 6.19.

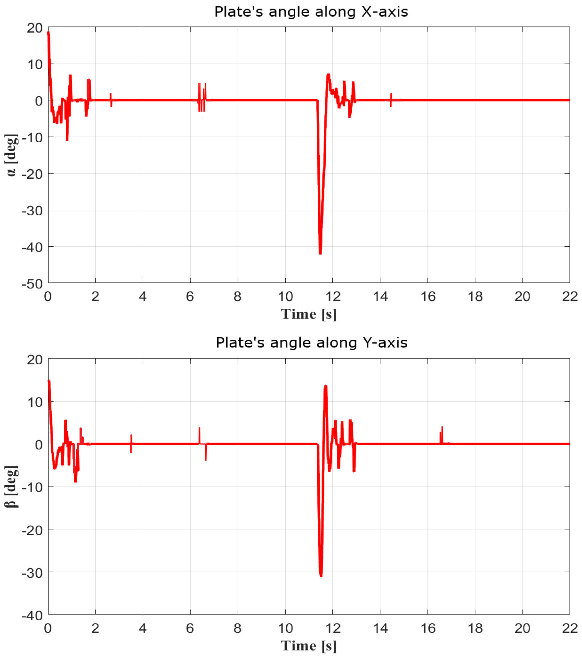


Figure 6.12: Plate deflection angles (α,β) along x-y-axes for LQR controller stability test.

For a square trajectory, the control process through all sides of rectangular takes 7 sec,. Figure 6.20 shows the motion of the square tracking along x and y axes. Figure 6.21 shows the ball tracking trajectory of the xy plane.

6.2.2 PID Controller

The performance of PID controller for trajectory tracking was tested on the real system. For a circle trajectory tracking test, the tracking time is about 5.9 second. The tracking position along x and y axes is shown on the Figure 6.22. The output position of the x-y plane gives a circular trajectory with a radius of 0.065 m with an error 2 mm as shown in the Figure 6.23.

In the square trajectory test, the controller takes about 16.67 sec. to complete one cycle with maximum steady state error of 1.5 mm. Figure 6.24 shows the motion of the square tracking along x and y axes and Figure 6.25

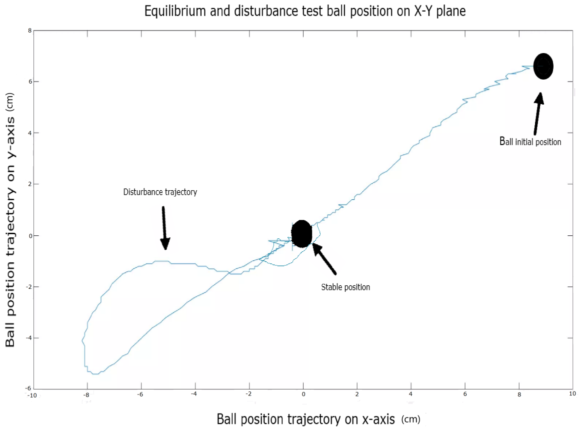


Figure 6.13: System equilibrium and and disturbance test of ball position for LQR controller stability test.

shows the ball square tracking trajectory of the x-y plane.

6.2.3 LMPC

The linear model predictive control (LMPC) was tested to track a circular trajectory.The LMPC algorithm tracking time for a circle trajectory is about 4.25 second. Figure 6.26 shows the tracking profile for motion along the x and y axes. The output position of the trajectory tracking along x and y axes gives a circular trajectory with a radius of 0.065 m with an error less than 1 mm as shown in Figure 6.27.

For a square trajectory, the control process through all sides of square takes 8 seconds. Figure 6.28 shows the motion of the square tracking along x and y axes. Figure 6.29 shows the ball tracking trajectory of the xy plane.

Table 6.6 presents a comparison between the selected methods of control for circle and square trajectorty. It can be said that through the results, the LQT and LMPC controllers showed close results through in terms of the lowest maximum steady state error, but for the the lowest average steady state error in tracking the rolling ball to the desired path, LQT controller

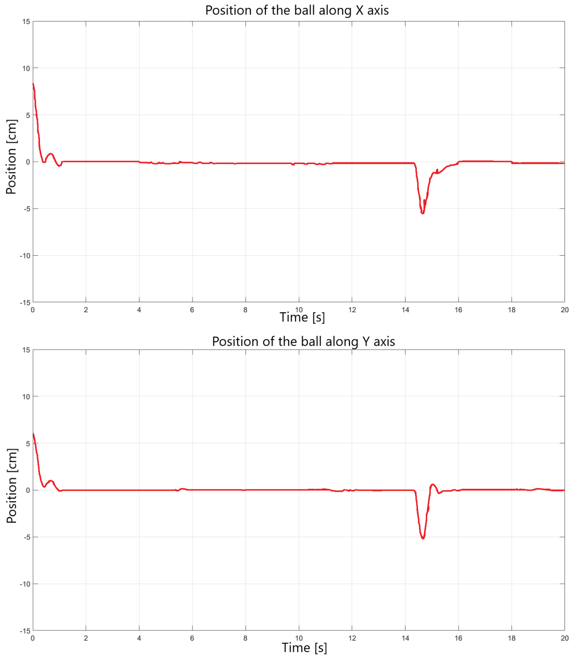


Figure 6.14: x-y position of ball with disturbance for LMPC stability test.

showed the best result for a circle tracking path and LMPC strategy for a square tracking path.

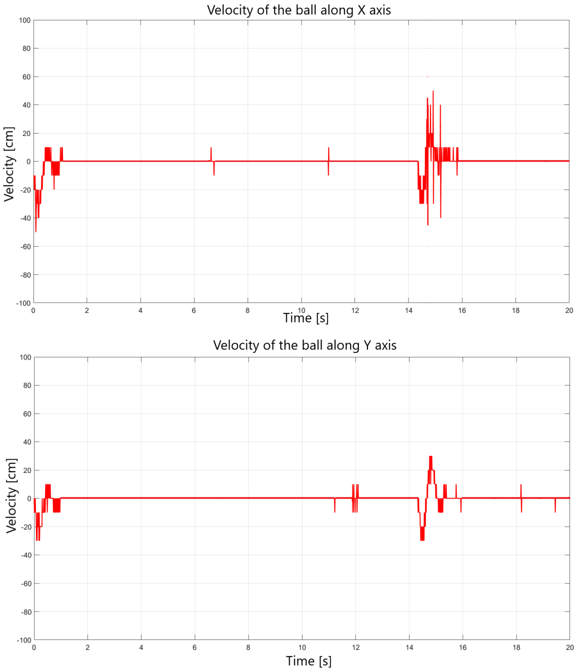


Figure 6.15: Velocity of the ball along x-y-axes for LMPC stability test.

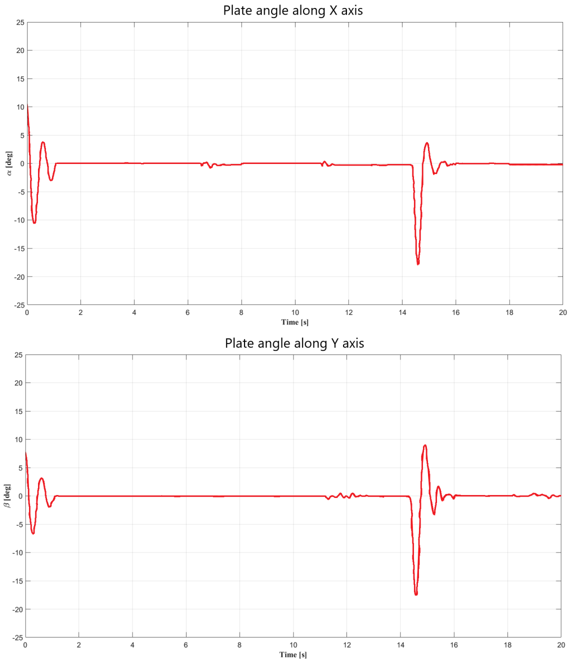


Figure 6.16: Plate deflection angles (α,β) along x-y-axes forLMPC stability test.

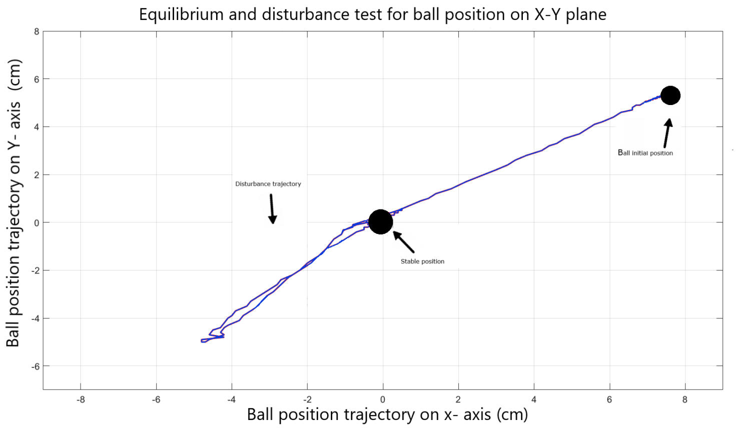


Figure 6.17: System equilibrium with disturbance for ball position for LMPC stability test.

Table 6.5: Static position tracking comparison.

Control method	Settling time (sec.)	Maximum angle (°)	Error steady state (mm)	Hardware
PID	4.8	16	3	Arduino uno microcontroller (20$).
State-feedback	2.87	30	1	Arduino uno microcontroller (20$).
LQR	1.84	18	1.3	Arduino uno microcontroller (20$).
LMPC	1.09	11	1	PC with Arduino uno microcontroller (1000$).

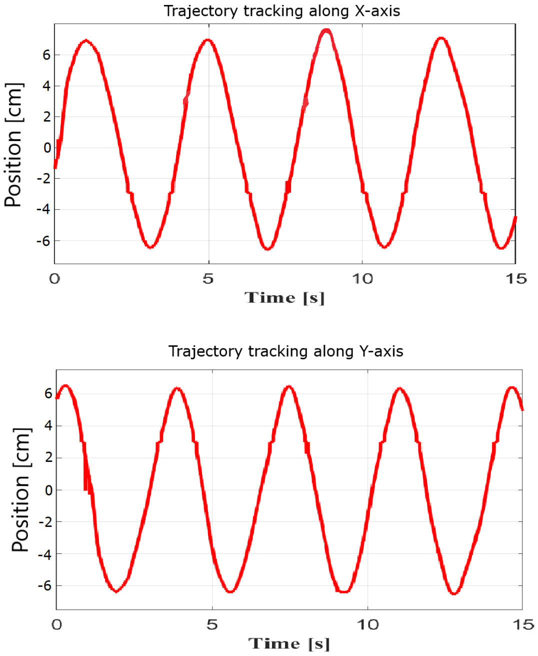


Figure 6.18: x-y position of ball for circle LQT controller tracking test.

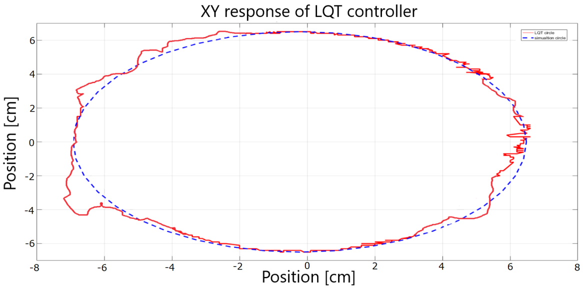


Figure 6.19: Circular trajectory of ball for circle LQT controller tracking test.

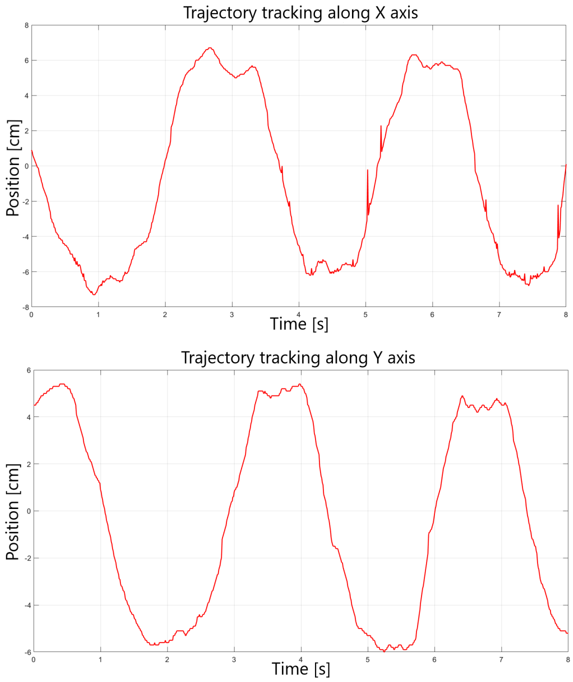


Figure 6.20: x-y position of ball for rectangular LQT controller tracking test.

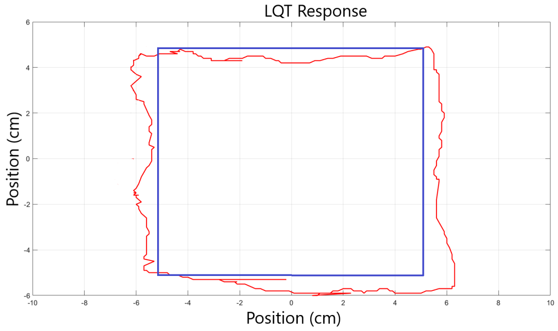


Figure 6.21: Square trajectory of ball for rectangular LQT controller tracking test.

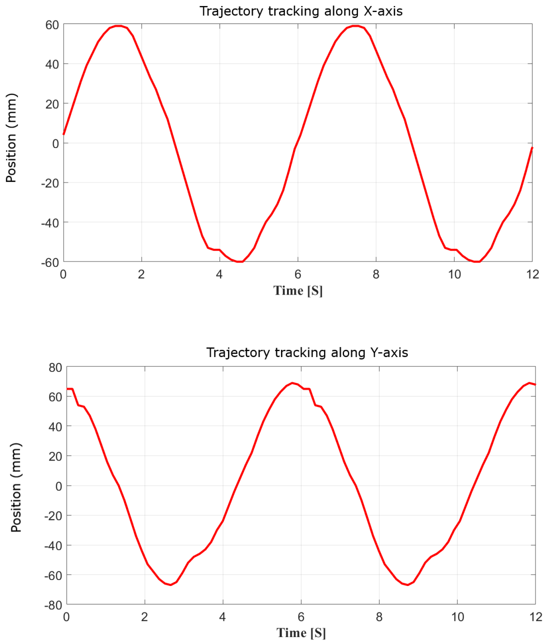


Figure 6.22: x-y position of ball for PID controller circle tracking test.

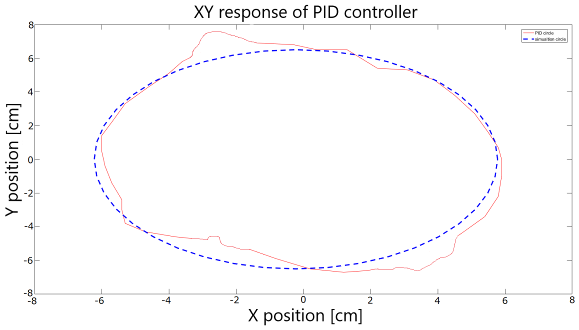


Figure 6.23: Circular trajectory of ball for PID controller circle tracking test.

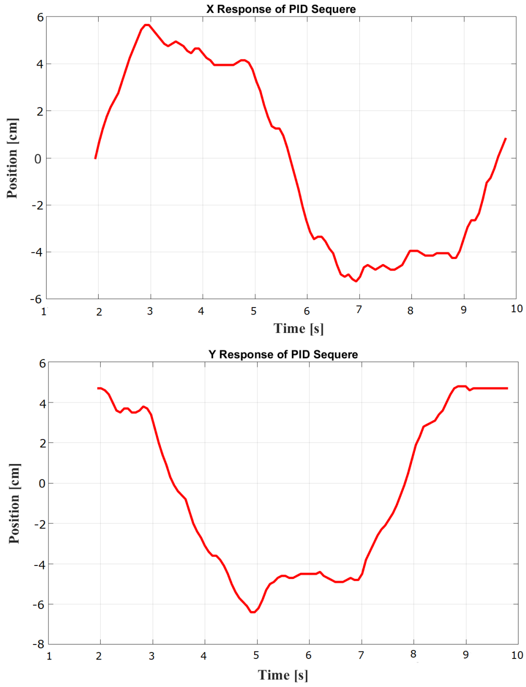


Figure 6.24: x-y position of ball for PID controller square tracking test.

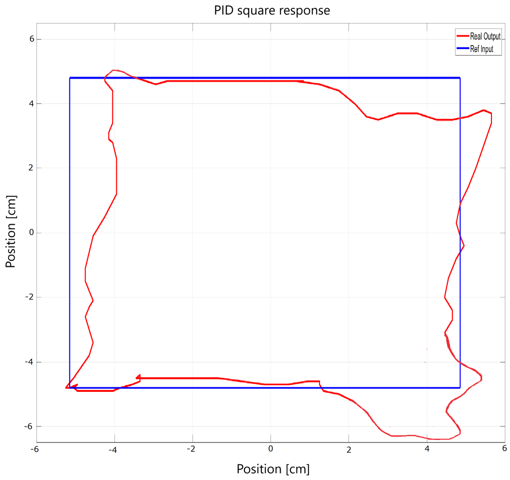


Figure 6.25: Square trajectory of ball for PID controller square tracking test.

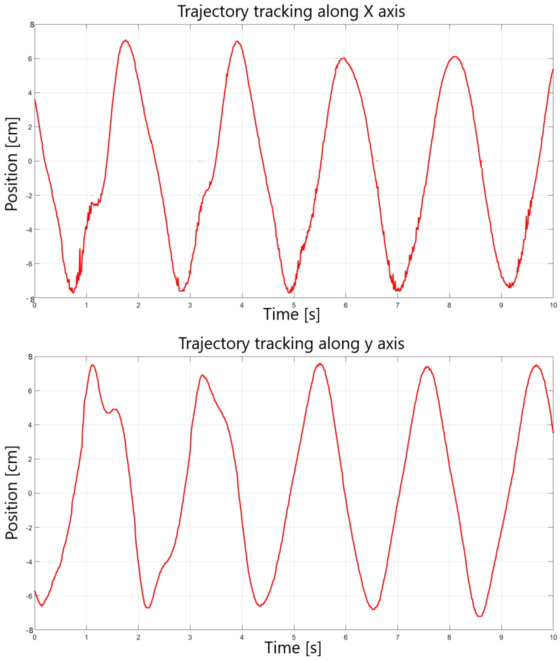


Figure 6.26: x-y position of ball for circle LMPC controller tracking test.

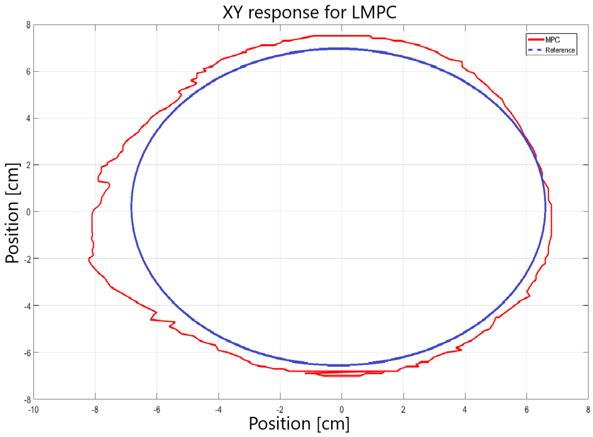


Figure 6.27: Circular trajectory of ball for circle LMPC controller tracking test.

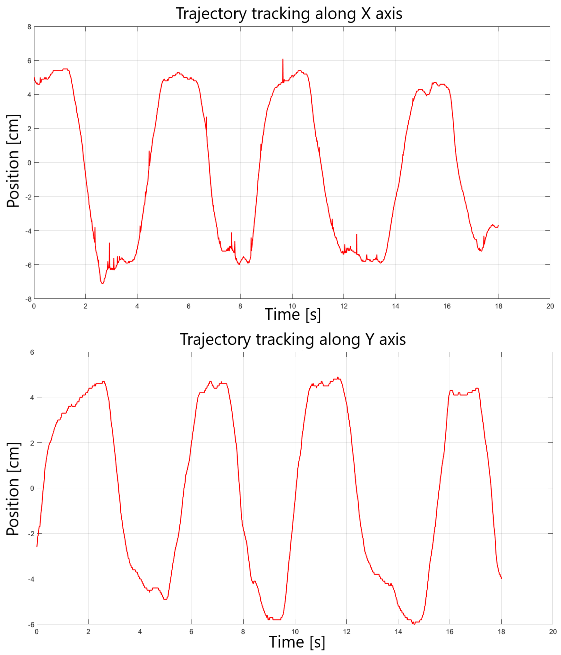


Figure 6.28: x-y position of ball for square LMPC controller tracking test.

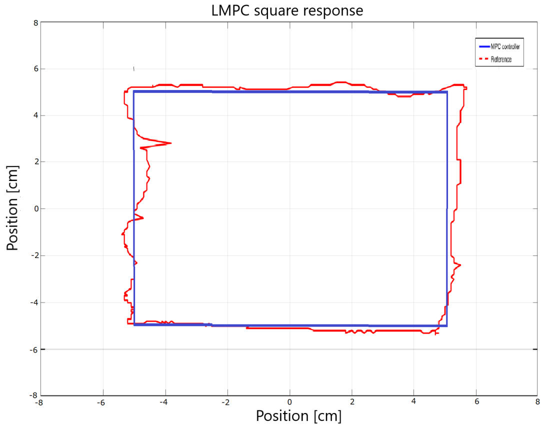


Figure 6.29: Square trajectory of ball for square LMPC controller tracking test.

Table 6.6: circle and Square position tracking comparison.

Control method	Tracking time (sec.)	Maximum error (mm)	Average error (mm)
PID (Circle)	6.69	2.3	1.4
LQT (Circle)	4.13	2	0.6
LMPC (circle)	4.25	1.8	1.2
PID (Square)	16.67	1.5	1.3
LQT (Square)	7	1.5	1.1
LMPC (Square)	8	1.3	0.7

Chapter 7

Conclusion

In this thesis, a BPS is developed, tested and controlled. With this BPS, five different control strategies were applied. It could be concluded that all of the controllers were able to stabilize the BPS with different performances and tracking some reference position such as circle and square for the rolling ball. However it was difficult to have the same results as simulation, because of the nonlinear characteristics of the real BPS, .

Different control schemes including (LMPC, PID and state space controllers) are applied to design a regulator and tracker controllers for the BPS. These controllers are designed based on the linearized mathematical model of the BPS, and then these controllers discretize using digital control theory which are applied to the real BPS. The ball position is provided from the resistive touch screen, but the other states of the system are not provided, so the observer is then designed to get the other states of the system since it is an essential part of the controller algorithm. It is found that such linear controllers work well with the nonlinear BPS. To achieve further improvements on the performance of the BPS, a nonlinear control theories may be used.

It can be observed with a detailed examination of the simulation results of different types of the selected controllers that the response of the stability test and trajectory tracking test had some differernt result from experimental results. The reason for this difference is interconnection terms in the dynamical model that are neglected in controller design procedure. These uncompensated neglected terms are act on the ball adversely and cause this unexpected behaviour for some controllers.

For stability test, although the LMPC controller need for a special personal computer to control of BPS, but the controller showed superior performance over other types of controllers, since it needs less time to stabilize the ball on the origin point and lowest used energy. On other hand, LMPC and LQT cotrollers showed similar result for trajectroty tracking test, where LQT controller has best result for following the circle trajectory and LMPC strategy for the square trajectory in terms of minimum average steady state error.

Several problems and difficulties were encountered during working with this project. One of the main difficulties was existed in the frequency of the Arduino microcontroller, as the frequency of the Arduino is not fully sufficient to perform the required tasks, so the computation time of the tasks becomes slow, which leads to slowing down in sending the commands and sometimes the system becomes unstable.

Further improvements on system performance can be achieved using an accurate solution for speed estimation. Several ideas are being studied to tackle this problem including continuous observers and discrete observers.

Although the controllers has been successfully stabilized the ball on the center of the plate with different performance and some controllers track the predefined trajectory , controller technique must be improved so that a robust controller and a better response can be achieved. More control theories can be applied and tested, like a neural networks, sliding mode controller (SMC) and other fuzzy logic controller.

Appendix A

LQR and LQT Matlab Code

```
% Firas Al-haddad
% Modeling and Control of a Ball and Plate System
% January, 2021
% LQR and LQT Matlab Code

% LQR controller
clear
clc
close all
syms z1 z2 z3 z4 u1 u2 % system state and inputs
z0=[0 0 0 0]*0;%cm % initail condition [X Xdot Y Ydot]

% System parameters
mb=112e-3; % mass in Kg
r=30e-3/2; % radius in m
Ib=(2/5)*mb*r2̂
g=9.81; % gravity
```

```
% Input forces (u1,u2)
f1=-mb*g*sin(u1)/(mb+Ib/2̂);
f2=-mb*g*sin(u2)/(mb+Ib/r2̂);

% System states (z1 ... z4)
zdot1=z2;
zdot2=f1;
zdot3=z4;
zdot4=f2;
z=[z1; z2; z3; z4];
zdot=[zdot1; zdot2; zdot3; zdot4];
u=[u1; u2];

% System matrix A
Az=jacobian(zdot,z);
A=subs(Az,{z1, z2, z3, z4, u1, u2},{0,0, 0, 0,0, 0});
A=double(A);

% Input matrix B
Bu=jacobian(zdot,u);
B=subs(Bu,{z1, z2, z3, z4, u1, u2},{0,0, 0, 0,0, 0});
B=double(B);
C=[1 0 0 0; 0 0 1 0];

Co = ctrb(A,B); % System Controllability
RANK=rank(Co) % System Rank
n=4;% 4 states
```

```
m=2;% 2 inputs
p=2;% 2 outputs

% LQR Method
Q= [90 0 0 0
0 1 0 0
0 0 80 0
0 0 0 1];
R= [10 0; 0 10]*2;
K=lqr(A,B,Q,R) % Gain matrix
Pc= eig (A-B*K) % closed poles
K10= lqr(A,B,Q*2.78,R/46.6);
Pc10= eig (A-B*K10)


%% Observer design
Om=obsv(A,C); % observability matrix Om
Ro=rank (Om) % Check the rank of Om, if rank == n --> the system
is observable
L=lqr(A',C',Q*2.78,R/46.6)'

%% Discrete system
[Kd,P,Pcz] = dlqr(F,G,Q,R);
Poz= exp(10*Ts)*Pcz;
Ld= place (F',C',Poz)'

%% LQT
```

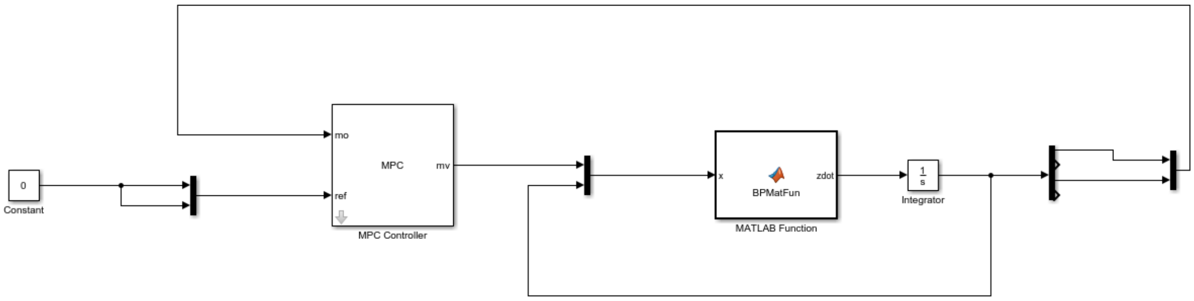
```
C1= [1 0 0 0;0 0 0 0;0 0 1 0;0 0 0 0];

r=[0.1,0,0.1,0];r=r';

V= inv ((A-B*K)')* C1'*Q*r;

S= -inv (R)*B'*V;

S=S';


%% Discrete LQT

[Kd,P,Pcz] = dlqr(F,G,Q,R);

Sd=F-G*Kd;

Sd=-inv(Sd');

Vd=Sd*C1'*Q*r;

%% Discrete LQR

Nd = inv(-C*inv(F-G*K - eye(n))*G)

N = inv(-C*inv(A-B*K - eye(n))*B)
```

Appendix B

MPC Matlab Code and Simulink

```
% Firas Al-haddad
% Modeling and Control of a Ball and Plate System
% January, 2021
% MPCMatlab Code

function zdot=BPMatFun(x)
u1=x(1);
u2=x(2);
z1=x(3);
z2=x(4);
z3=x(5);
z4=x(6);

Ib=1.67e-5;
m=0.15;
```

```
r=0.02;

g=9.81;


f1=-m*g*sin(u1)/(m+Ib/r2̂);

f2=-m*g*sin(u2)/(m+Ib/r2̂);


zdot1=z2;

zdot2=f1;

zdot3=z4;

zdot4=f2;

zdot=[zdot1 zdot2 zdot3 zdot4];

end
```



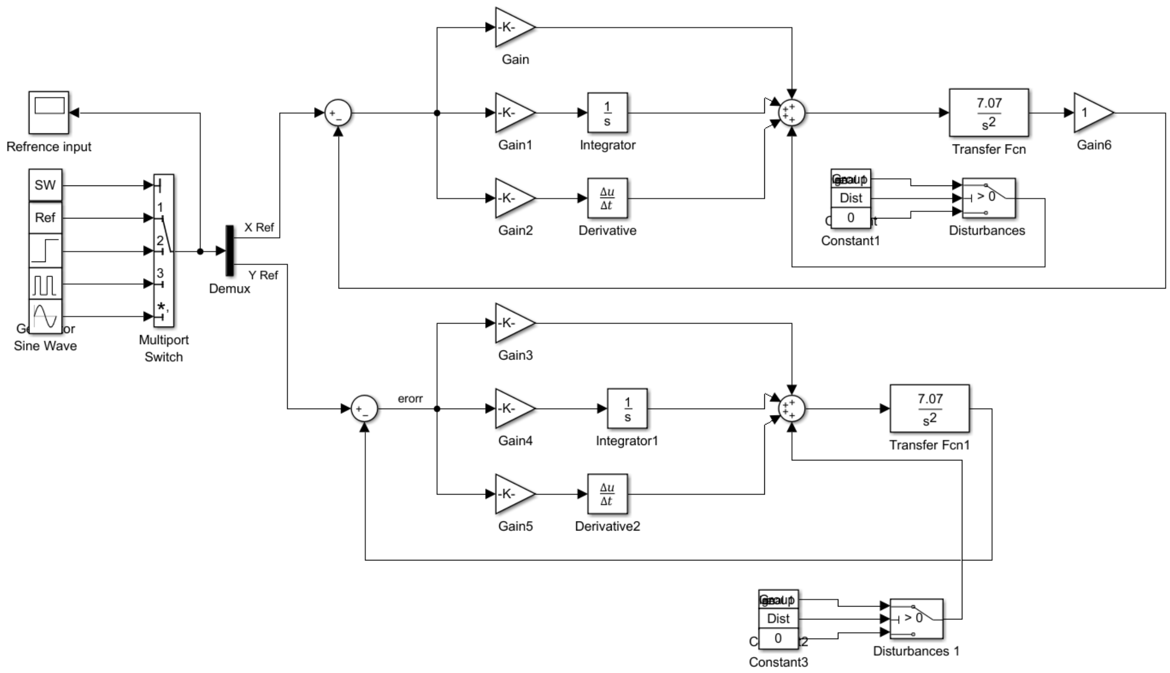
Appendix C

PID Matlab Code and Simulink

```
% Firas Al-haddad
% Modeling and Control of a Ball and Plate System
% January, 2021
% PID Matlab Code

% PID controller
clear
clc
close all
syms z1 z2 z3 z4 u1 u2 % system state and inputs
z0=[0 0 0 0]*0;%cm % initail condition [X Xdot Y Ydot]

% System parameters
mb=112e-3; % mass in Kg
r=30e-3/2; % radius in m
Ib=(2/5)*mb*r2̂
g=9.81; % gravity
```

```
% Input forces (u1,u2)
f1=-mb*g*sin(u1)/(mb+Ib/2̂);
f2=-mb*g*sin(u2)/(mb+Ib/r2̂);

% System states (z1 ...  z4)
zdot1=z2;
zdot2=f1;
zdot3=z4;
zdot4=f2;
z=[z1; z2; z3; z4];
zdot=[zdot1; zdot2; zdot3; zdot4];
u=[u1; u2];

% System matrix A
Az=jacobian(zdot,z);
A=subs(Az,{z1, z2, z3, z4, u1, u2},{0,0, 0, 0,0, 0});
A=double(A);

% Input matrix B
Bu=jacobian(zdot,u);
B=subs(Bu,{z1, z2, z3, z4, u1, u2},{0,0, 0, 0,0, 0});
B=double(B);
C=[1 0 0 0 ; 0 0 1 0];

% From SISOTOOL
Kp = 1.69; % Proportional gain
Ki = 1.12; % Integral gain
```

```
Kd = 0.662; % Derivative gain
```



Appendix D

State Feedback Matlab Code

```
% Firas Al-haddad
% Modeling and Control of a Ball and Plate System
% January, 2021
% State Feedback Matlab Code

% State Feedback controller
clear
clc
close all
syms z1 z2 z3 z4 u1 u2 % system state and inputs
z0=[0 0 0 0]*0;%cm % initail condition [X Xdot Y Ydot]

% System parameters
mb=112e-3; % mass in Kg
r=30e-3/2; % radius in m
Ib=(2/5)*mb*r2̂
g=9.81; % gravity
```

```
% Input forces (u1,u2)
f1=-mb*g*sin(u1)/(mb+Ib/2̂);
f2=-mb*g*sin(u2)/(mb+Ib/r2̂);

% System states (z1 ...  z4)
zdot1=z2;
zdot2=f1;
zdot3=z4;
zdot4=f2;
z=[z1; z2; z3; z4];
zdot=[zdot1; zdot2; zdot3; zdot4];
u=[u1; u2];

% System matrix A
Az=jacobian(zdot,z);
A=subs(Az,{z1, z2, z3, z4, u1, u2},{0,0, 0, 0,0, 0});
A=double(A);

% Input matrix B
Bu=jacobian(zdot,u);
B=subs(Bu,{z1, z2, z3, z4, u1, u2},{0,0, 0, 0,0, 0});
B=double(B);
C=[1 0 0 0; 0 0 1 0];

Co = ctrb(A,B); % System Controllability
RANK=rank(Co) % System Rank
n=4;% 4 states
```

```
m=2;% 2 inputs
p=2;% 2 outputs

   z1=0.59133; z2=0.65; wn1=3.221148; wn2=3.3;
p1=-z1*wn1+i*wn1*sqrt(1-z12̂);
p2=conj (p1);
p3=-z2*wn2+i*wn2*sqrt(1-z22̂);
p4=conj (p3);

   pc= [p1 p2 p3 p4]
K= place (A,B,pc)
Pc= eig (A-B*K)

% Observer design
Om=obsv(A,C); % observability matrix Om
Ro=rank (Om) % Check the rank of Om, if rank == n --> the system
is observable
Po=10*pc(1:4)
L=place(A',C',Po)'

% Discrete system

   Ts= 1e-2;
[F,G]=c2d(A,B,Ts);
Pd=exp(pc*Ts);
Kd=place(F,G,Pd)
Poz=exp(Po*Ts)
```

```
Ld=place(F',C',Poz)'
```

Bibliography

[1] Mohamed Aburaia, Erich Markl, and Kemajl Stuja. New concept for design and control of 4 axis robot using the additive manufacturing technology. *Procedia Engineering*, 100:1364–1369, 2015.

[2] A. AG. Arduino uno rev3. https://store.arduino.cc/arduino-uno-rev3, Accessed: 09.2020.

[3] Amir G Aghdam. Decentralized control design using piecewise constant hold functions. In *2006 American Control Conference*, pages 6–pp. IEEE, 2006.

[4] Hazem I Ali, Haider M Jassim, and Amjad F Hasan. Optimal nonlinear model reference controller design for ball and plate system. *Arabian Journal for Science and Engineering*, 44(8):6757–6768, 2019.

[5] M Amin and R Ji-Chul. Neural network-based pid compensation for nonlinear systems: ball-on-plate example [j]. *International Journal of Dynamics and Control*, 2018.

[6] N Andinet. *Design of Fuzzy Sliding Mode Controller for the Ball and Plate System.* PhD thesis, PhD thesis, aau, 2011.

[7] Karl Johan Åström and Tore Hägglund. *PID controllers: theory, design, and tuning*, volume 2. Instrument society of America Research Triangle Park, NC, 1995.

[8] Karl Johan Åström and Richard M Murray. *Feedback systems: an introduction for scientists and engineers.* Princeton university press, 2010.

[9] Shorya Awtar, C Bernard, N Boklund, A Master, D Ueda, and Kevin Craig. Mechatronic design of a ball-on-plate balancing system. *Mechatronics*, 12(2):217–228, 2002.

[10] Ming Bai, Huiqiu Lu, Jintao Su, and Yantao Tian. Motion control of ball and plate system using supervisory fuzzy controller. In *2006 6th World Congress on Intelligent Control and Automation*, volume 2, pages 8127–8131. IEEE, 2006.

[11] Heeseung Bang and Young Sam Lee. Implementation of a ball and plate control system using sliding mode control. *IEEE Access*, 6:32401–32408, 2018.

[12] R Bars, P Colaneri, L Dugard, F Allgower, A Kleimenov, and CW Scherer. Trends in theory of control system design-status report by the ifac coordinating committee. In *Proceedings of the 17th IFAC World Congress, Séoul*, volume 17, pages 93–114, 2008.

[13] John T Betts. *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.

[14] Manashita Borah, Prasanta Roy, and Binoy Krishna Roy. Enhanced performance in trajectory tracking of a ball and plate system using fractional order controller. *IETE Journal of Research*, 64(1):76–86, 2018.

[15] Hartmut Bossel. *Systems and models: complexity, dynamics, evolution, sustainability*. BoD–Books on Demand, 2007.

[16] Roland Burns. *Advanced control engineering*. Elsevier, 2001.

[17] Chi-Cheng Cheng and Chen-Hsun Tsai. Visual servo control for balancing a ball-plate system. *International Journal of Mechanical Engineering and Robotics Research*, 5(1):28, 2016.

[18] Arindam Das and Prasanta Roy. Improved performance of cascaded fractional-order smc over cascaded smc for position control of a ball and plate system. *IETE Journal of Research*, 63(2):238–247, 2017.

[19] David Debono and Marvin Bugeja. Application of sliding mode control to the ball and plate problem. In *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 1, pages 412–419. IEEE, 2015.

[20] Huida Duan, Yantao Tian, and Guangbin Wang. Trajectory tracking control of ball and plate system based on auto-disturbance rejection controller. In *2009 7th Asian Control Conference*, pages 471–476. IEEE, 2009.

[21] František Dušek, Daniel Honc, and K Rahul Sharma. Modelling of ball and plate system based on first principle model and optimal control. In *2017 21st International Conference on Process Control (PC)*, pages 216–221. IEEE, 2017.

[22] Fabian Enterprises. Fs5106b servo motor datasheet. https://www.fabian.com.mt/viewer/18189/pdf.pdf, 2020.

[23] Xingzhe Fan, Naiyao Zhang, and Shujie Teng. Trajectory planning and tracking of ball and plate system using hierarchical fuzzy control scheme. *Fuzzy Sets and Systems*, 144(2):297–312, 2004.

[24] Paul B Fancher. *Kinematics and dynamics of planar machinery*, volume 1. Englewood Cliffs, N.J., 1979.

[25] E. Fernandez-Car and Enrique Zuazua. Control theory: History, mathematical achievements and perspectives. 01 2003.

[26] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[27] W Gharieb and G Nagib. Fuzzy intervention in pid controller design. In *ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No. 01TH8570)*, volume 3, pages 1639–1643. IEEE, 2001.

[28] Haluk Gözde. Comparative analysis of evolutionary computation based gain scheduling control for ball and plate stabilization system. *Balkan Journal of Electrical and Computer Engineering*, 7(1):44–55, 2019.

[29] R Douglas Gregory. *Classical mechanics*. Cambridge University Press, 2006.

[30] Kyongwon Han, Yantao Tian, Yongsu Kong, Jinsong Li, and Yinghui Zhang. Tracking control of ball and plate system using a improved pso on-line training pid neural network. In *2012 IEEE International Conference on Mechatronics and Automation*, pages 2297–2302. IEEE, 2012.

[31] Russell Charles Hibbeler and Russell C Hibbeler. *Engineering mechanics: statics & dynamics*. Pearson Education India, 2007.

[32] Franz Hover. Lab 4: Motor control - design of electromechanical robotic systems. https://ocw.mit.edu/courses/mechanical-engineering/2-017j-design-of-electromechanical-robotic-systems-fall-2009/labs/MIT2_017JF09_lab4.pdf, 2009. Accessed: 09.2020.

[33] Anna Jadlovska, Štefan Jajčišin, and Richard Lonščák. Modelling and pid control design of nonlinear educational model ball & plate. In *17th International Conference on Process Control*, volume 2, pages 871–874, 2009.

[34] R. E. Kalman. *Control theory*, volume 1. Encyclopedia Britannica. https://www.britannica.com/science/control-theory-mathematics, 2005.

[35] Andrej Knuplez, Amor Chowdhury, and Rajko Svecko. Modeling and control design for the ball and plate system. In *IEEE International Conference on Industrial Technology, 2003*, volume 2, pages 1064–1067. IEEE, 2003.

[36] Huibert Kwakernaak and Raphael Sivan. *Linear optimal control systems*, volume 1. Wiley-interscience New York, 1972.

[37] AJ Laub and M Wette. Algorithms and software for pole assignment and observers. Technical report, California Univ., Santa Barbara (USA). Dept. of Electrical and Computer . . . , 1984.

[38] AJ Laub and M Wette. Algorithms and software for pole assignment and observers. Technical report, California Univ., Santa Barbara (USA). Dept. of Electrical and Computer . . . , 1984.

[39] Kwang-Kyu Lee, Georg Batz, and Dirk Wollherr. Basketball robot: Ball-on-plate with pure haptic information. In *2008 IEEE International Conference on Robotics and Automation*, pages 2410–2415. IEEE, 2008.

[40] Frank Lewis. Lyapunov stability analysis for feedback control design. https://lewisgroup.uta.edu/Lectures/Lyapunov%20stab%20closed%20loop.pdf, 2008.

[41] Dejun Liu, Yantao Tian, and Huida Duan. Ball and plate control system based on sliding mode control with uncertain items observe compensation. In *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, volume 2, pages 216–221. IEEE, 2009.

[42] Mouser Electronics Ltd. Ft series 4-wire analog resistive touch screens with fpc tails. https://eu.mouser.com/pdfdocs/CN-0320FTSerieswithFPCTail.pdf,, 2017. Accessed: 09.2020.

[43] Jia Luo and C Edward Lan. Determination of weighting matrices of a linear quadratic regulator. *Journal of Guidance, Control, and Dynamics*, 18(6):1462–1463, 1995.

[44] Jie Ma, Hao Tao, and Jingwen Huang. Observer integrated backstepping control for a ball and plate system. *International Journal of Dynamics and Control*, pages 1–8, 2020.

[45] Miad Moarref, Mohsen Saadat, and Gholamreza Vossoughi. Mechatronic design and position control of a novel ball and plate system. In *2008 16th Mediterranean Conference on Control and Automation*, pages 1071–1076. IEEE, 2008.

[46] Seyed Alireza Moezi, Ehsan Zakeri, and Mohammad Eghtesad. Optimal adaptive interval type-2 fuzzy fractional-order backstepping sliding mode control method for some classes of nonlinear systems. *ISA transactions*, 93:23–39, 2019.

[47] Nima Mohajerin, Mohammad Menhaj, and Ali Doustmohammadi. A reinforcement learning fuzzy controller for the ball and plate system. pages 1–8, 07 2010.

[48] Kenneth R Muske and James B Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993.

[49] Andinet Negash and Nagendra P Singh. Position control and tracking of ball and plate system using fuzzy sliding mode controller. In *Afro-European Conference for Industrial Advancement*, pages 123–132. Springer, 2015.

[50] Daniel Nichols. Arduino-based data acquisition into excel, labview, and matlab. *The Physics Teacher*, 55(4):226–227, 2017.

[51] Norman S Nise. Control system engineering, john wiley & sons. *Inc, New York*, 2011.

[52] David Núñez, Gustavo Acosta, and Jovani Jiménez. Control of a ball-and-plate system using a state-feedback controller. *Ingeniare. Revista chilena de ingeniería*, 28(1):6–15, 2020.

[53] Katsuhiko Ogata. *Modern control engineering*. Prentice hall, 2010.

[54] M Oravec. *Control of Mechatronical Systems using .NET Applications*. PhD thesis, Master thesis, 2014.

[55] Matej Oravec and Anna Jadlovská. Model predictive control of a ball and plate laboratory model. In *2015 IEEE 13th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 165–170. IEEE, 2015.

[56] Yongyut Pattanapong and Chirdpong Deelertpaiboon. Ball and plate position control based on fuzzy logic with adaptive integral control action. In *2013 IEEE International Conference on Mechatronics and Automation*, pages 1513–1517. IEEE, 2013.

[57] M Ali Rastin, Erfan Talebzadeh, S Ali A Moosavian, and Mojtaba Alaeddin. Trajectory tracking and obstacle avoidance of a ball and plate system using fuzzy theory. In *2013 13th Iranian Conference on Fuzzy Systems (IFSC)*, pages 1–5. IEEE, 2013.

[58] Jirka Roubal and Vladimír Havlena. Range control mpc approach for two-dimensional system. *IFAC Proceedings Volumes*, 38(1):465–470, 2005.

[59] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer Science & Business Media, 2013.

[60] Joel L Schiff. *The Laplace transform: theory and applications*. Springer Science & Business Media, 1999.

[61] Ahmed Shabana. *Dynamics of multibody systems*. Cambridge university press, 2020.

[62] Deejay Stander, Santiago Jiménez-Leudo, and Nicanor Quijano. Low-cost "ball and plate" design and implementation for learning control systems. In *2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC)*, pages 1–6. IEEE, 2017.

[63] Jasem Tamimi. Simulation of three-phase induction motor using nonlinear model predictive control technique. *Cogent Engineering*, 5(1):1516489, 2018.

[64] Chee Pin Tan, Kok Soo Teoh, and Lim Jen Nee Jones. A review of matlab's sisotool; features and contributions to control education. *IFAC Proceedings Volumes*, 41(2):8473–8474, 2008.

[65] S. Valluri and V. Kapila. Stability analysis for linear/nonlinear model predictive control of constrained processes. In *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207)*, volume 3, pages 1679–1683 vol.3, 1998.

[66] John Van de Vegte. *Feedback control systems*. Prentice-Hall, Inc., 1994.

[67] Hongrui Wang, Yantao Tian, Ce Ding, Qing Gu, and Feng Guo. Output regulation of the ball and plate system with a nonlinear velocity observer. In *2008 7th World Congress on Intelligent Control and Automation*, pages 2164–2169. IEEE, 2008.

[68] Hongrui Wang, Yantao Tian, Zhen Sui, Xuefei Zhang, and Ce Ding. Tracking control of ball and plate system with a double feedback loop structure. In *2007 International Conference on Mechatronics and Automation*, pages 1114–1119. IEEE, 2007.

[69] Jun Xiao and Giorgio Buttazzo. Adaptive embedded control for a ball and plate system. In *Proceedings of The Eighth International Conference on Adaptive and Self-Adaptive Systems and Applications IARIA*, pages 40–45, 2016.

[70] Antonio Yarza, Victor Santibanez, and Javier Moreno-Valenzuela. Global asymptotic stability of the classical pid controller by considering saturation effects in industrial robots. *International Journal of Advanced Robotic Systems*, 8(4):36, 2011.

[71] Dongfeng Yuan and Zhenhao Zhang. Modelling and control scheme of the ball–plate trajectory-tracking pneumatic system with a touch screen and a rotary cylinder. *IET control theory & applications*, 4(4):573–589, 2010.

[72] N Yubazaki, Jianqiang Yi, M Otani, N Unemura, and K Hirota. Trajectory tracking control of unconstrained objects based on the sirms dynamically connected fuzzy inference model. In *Proceedings of 6th International Fuzzy Systems Conference*, volume 2, pages 609–614. IEEE, 1997.

[73] A Zeeshan, N Nauman, and M Jawad Khan. Design, control and implementation of a ball on plate balancing system. In *Proceedings of 2012 9th International Bhurban Conference on Applied Sciences & Technology (IBCAST)*, pages 22–26. IEEE, 2012.

[74] P. Zhao and R. Nagamune. Discrete-time state-feedback switching lpv control with separate lyapunov functions for stability and local performance. In *2018 Annual American Control Conference (ACC)*, pages 2023–2028, 2018.