BALANCING A SPHERICAL INVERTED PENDULUM ON OMNIDIRECTIONAL WHEELS MOBILE ROBOT

by

Rattapan Pitaksongkram

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Mechatronics

Examination Committee:	Prof. Manukid Parnichkun (Chairperson)
	Dr. Mongkol Ekpanyapong
	Dr. Tanujjal Bora
Nationality:	Thai
Previous Degree:	Bachelor of Engineering
	Assumption University
	Bangkok, Thailand
Scholarship Donor:	Royal Thai Government Fellowship

Asian Institute of Technology
School of Engineering and Technology
Thailand

December 2022

AUTHOR'S DECLARATION

I, Rattapan Pitaksongkram, declare that the research work carried out for this thesis was in accordance with the regulations of the Asian Institute of Technology. The work presented in it is my own and has been generated by me as the result of my own original research, and if external sources were used, such sources have been cited. It is original and has not been submitted to any other institution to obtain another degree or qualification. This is a true copy of the thesis, including final revisions.

Date: 30 July 2022

Name: Rattapan Pitaksongkram

Signature: 

ACKNOWLEDGMENTS

Firstly, my sincere appreciation goes to Prof. Manukid Parnichkun for being this thesis supervisor. I have benefited from his wealth of knowledge and continued guidance. Without his encouragement and support, this thesis could not be done.

My gratitude should also go to Dr. Mongkol Ekpanyapong and Dr. Tanujjal Bora, the examination committees, for their insightful on how to improve my thesis.

I would like to extend my thanks to Mr. Wiput Tuvayanond, Mr. Prasitthichai Naronglerdrit, Mr. Matee Vadrukchid, and Mr. Jirapod Jintasornrom, students of Mechatronics Engineering, for their valuable advice for my thesis.

I would want to thank Mr. Thanit Pattana and Mr. Hoang Hung Manh, the technical personnel at the mechatronics department laboratory, for their assistance. They assisted me in successfully completing my robot prototype.

I would also want to express my gratitude to the Royal Thai Government for funding my whole master's degree at AIT.

Last but not least, this endeavor would not have been done without the encouragement and support of my parents and friends during my study.

ABSTRACT

In this study, balancing the spherical inverted pendulum is the main purpose. The control system for the three omnidirectional wheels mobile robot prototype is proposed to balance this type of pendulum. The mechanical design, electronics design, and control system are required to activate this robot. It can move in the x-y axis direction without turning clockwise or counterclockwise to keep the pendulum upright. Changing the pendulum tilt angle from the upright position is the disturbance that the system must handle. This system uses three DC motors for each wheel movement, three encoders as a sensor for each wheel, and the IMU sensor to sense the spherical inverted pendulum tilt angle. The distance and velocity of each wheel and the pendulum rod's tilt angle and angular velocity from the sensors are used as the input for the STM32 controller to calculate the voltage for all three motors to move the platform of the mobile robot for balancing the spherical inverted pendulum.

The optimum control algorithm is designed as the control system. This control system was invented based on a 2-DOF joint inverted pendulum with two external forces. These forces are applied along the x-y axis directions for the platform movement.

The Actual Model cannot be used to balance the spherical inverted pendulum because of the spherical inverted pendulum sensor problem. Therefore, this model is changed to be a one-axis inverted pendulum. Furthermore, the Simulation Model is designed to accomplish the main purpose of this study. Finally, these two models can be used to balance the inverted pendulum.

Keywords: Spherical Inverted Pendulum, Omnidirectional Wheels Robot, LQR, Dynamic Model

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

Page

LIST OF ABBREVIATIONS

DC = Direct Current
CAD = Computer-Aided Design
DOF = Degrees Of Freedom
DSP = Digital Signal Processor
EMF = Electromotive Force
FLC = Fuzzy Logic Control
IAE = Integral of the Absolute magnitude of the Error
IMU = Inertial Measurement Unit
ISE = Integral of the Square of the Error
ITAE = Integral of Time multiplied by the Absolute Error
LQR = Linear Quadratic Regulator
N = Negative
NASA = National Aeronautics and Space Administration
NL = Negative Large
NS = Negative Small
P = Positive
PC = Personal Computer
PD = Proportional Derivative
PID = Proportional Integral Derivative
PL = Positive Large
PS = Positive Small
PWM = Pulse Width Modulation
SMC = Sliding Mode Control
USB = Universal Serial Bus
Z = Zero

CHAPTER 1
INTRODUCTION

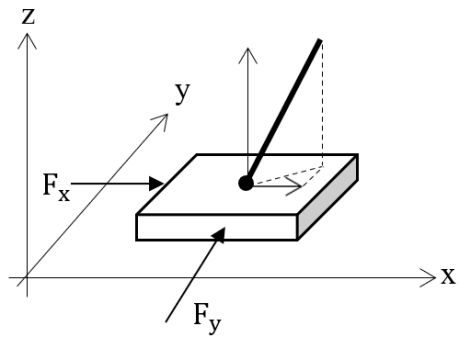
1.1 Background of the Study

These days, the inverted pendulum model is used in many real-world systems. For example, the stability of a rocket is an interesting area for NASA to use an inverted pendulum for better control. The humanoid robots can walk stably using the inverted pendulum model. A personal vehicle like Segway also uses the inverted pendulum model for balancing.

The inverted pendulum control system with a different mechanism is developed for control education. A spherical inverted is a x-y dimensional inverted pendulum. The spherical inverted pendulum is a long cylindrical rod mounted to the spehrical joint with a base that can move horizontally. The planar control force is used to move the base, and the base can move freely.

Figure 1.1

Spherical Inverted Pendulum



There are two equilibrium states of the inverted pendulum. Firstly, the inverted pendulum is asymptotically stable when its position is lower. Another state is at the upper position, which is called virtual stability. Therefore, the system requires a controller to sustain stability to balance the inverted pendulum.

Various types of wheels are used nowadays. Omnidirectional wheels can move translationally with any rotational movement along any desired path; this wheel is popularly used with mobile robots in industry and research.

For the controller design, it is also a challenge for researchers. There are many types of controllers that are used to make the inverted pendulum balanced. Many researchers use Proportional Integral Derivative, Linear Quadratic Regulator, Sliding Mode Control, and Fuzzy Logic Control controller.

This thesis focuses on balancing the spherical inverted pendulum with omnidirectional wheels platform robot. The controller is conducted to control the x-y planar inverted pendulum.

1.2 Statement of the Problem

The problem of a spherical inverted pendulum is unstable, nonminimum phase, and underactuated system. Therefore, this thesis will design the controller to challenge this problem by making the inverted pendulum balanced.

Because the spherical inverted pendulum is the two-dimensional inverted pendulum, it is an x-y planar inverted pendulum. Therefore, the base of this spherical inverted pendulum must be moved horizontally. So, the omnidirectional wheels mobile robot is chosen to be the base.

1.3 Objectives

This research aims to balance the spherical inverted pendulum. The list of objectives is as follows:

1. To design a spherical inverted pendulum on three omnidirectional wheels mobile robot.
2. To conduct a system control algorithm to balance the spherical inverted pendulum.

1.4 Limitations and Scopes

This thesis mainly focuses on designing a balanced spherical inverted pendulum with three omnidirectional wheels mobile platform applying the control algorithm to create the controller. Other specifications are stated as follows.

1. The dimension of the robot is 35×40×30 cm.
2. The diameter of the cylindrical pendulum is 1 cm and the length is 112 cm.
3. The weight of the robot is 18.9 kg.
4. The weight of the pendulum is 18.4 g.
5. The terrain is a flat surface.
6. The mobile robot must not rotate around itself.

CHAPTER 2
LITERATURE REVIEW

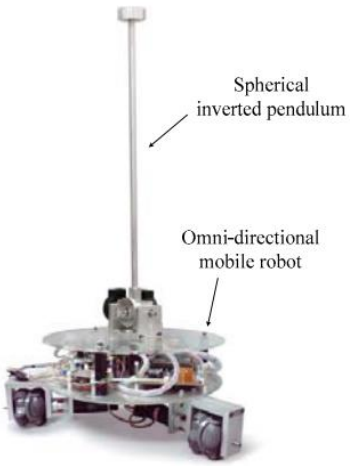
This chapter analyzes several related works to consider the controller for balancing the spherical inverted pendulum with omnidirectional wheels platform. At the beginning of this chapter, two related works which have a similar plant to this thesis plant are included to see the trend of the controller. Then, another part of this chapter discusses the other plants' controllers.

2.1 Spherical Inverted Pendulum Balancing with Omnidirectional Mobile Robot

The spherical pendulum is an x y axes inverted pendulum. It is a cylindrical rod mounted on a x-y axis movable base by spherical joint. The omnidirectional mobile robot is a platform that can move translationally with any rotational movement along any desired path combined. Therefore, the references [1] and [2] chose an omnidirectional mobile robot to be the base of the spherical inverted pendulum.

Figure 2.1

Omnidirectional Wheels Mobile Robot



In [1], the omnidirectional wheels platform is driven by three dc motors. Two optical encoders collect data the pendulum angular displacement with a resolution of 2000 pulses/revolution. The researchers use an 500 pulses/revolution resolution optical

encoder to sense the wheel angular displacement for feedback control. This research compared the experiment's results between LQR and SMC controllers as follows.

Figure 2.2

Experimental Results Comparative Relation of the Pendulum's Angular Position Responses

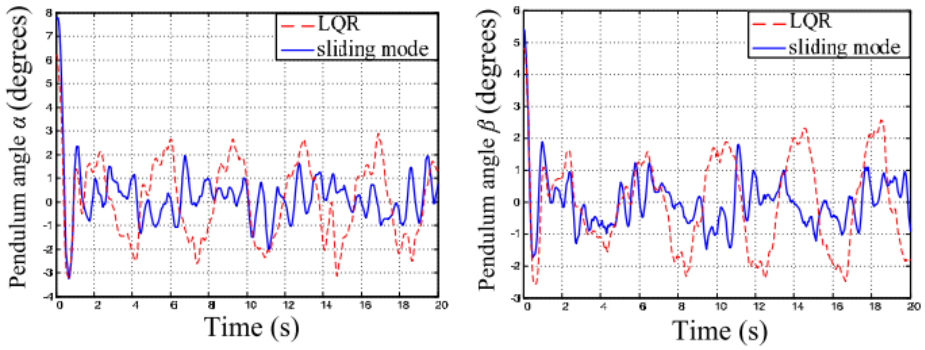
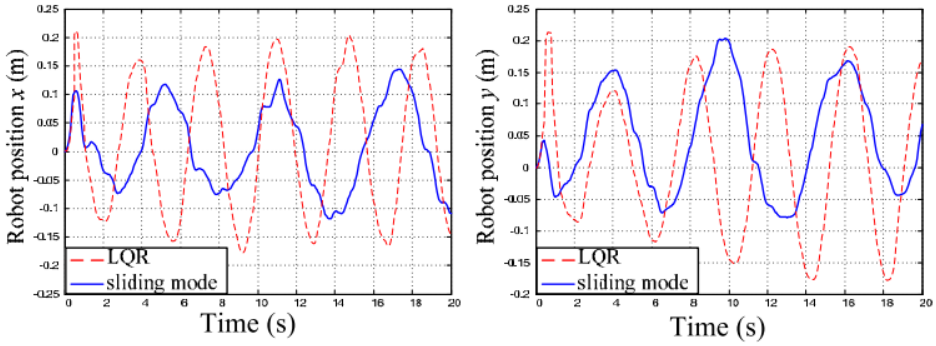


Figure 2.3

Experimental Results Comparative Relation of the Robot's Position Responses



Figures 2.2 and 2.3 show that these two designed control laws can used to balance the spherical inverted pendulum. For SMC, in the upright posture, the pendulum oscillates slightly within a range of ±2° in the pendulum's angle of angular displacement about the y axis (α) and a range of ±2° in the pendulum's angle of angular displacement about the x axis (β). For LQR, within a certain range of ±3° in the angle of α and a range of ±2.5° in the angle of β, the pendulum gently oscillates in the upright position. According to Figure 2.3, the system with LQR control has a great position deviation of the robot.

Component that is unmodeled dynamics such as friction can causes the system-state oscillations with tiny amplitudes

The reference [2] hardware is the same hardware as the reference [1]. The researchers use the LQR technique to design a stabilizing and output regulation controller. Figures 2.4 and 2.5 illustrate the outcomes of the tests.

Figure 2.4

Experimental Results of the Robot's Position Responses with relation to the X and Y axes

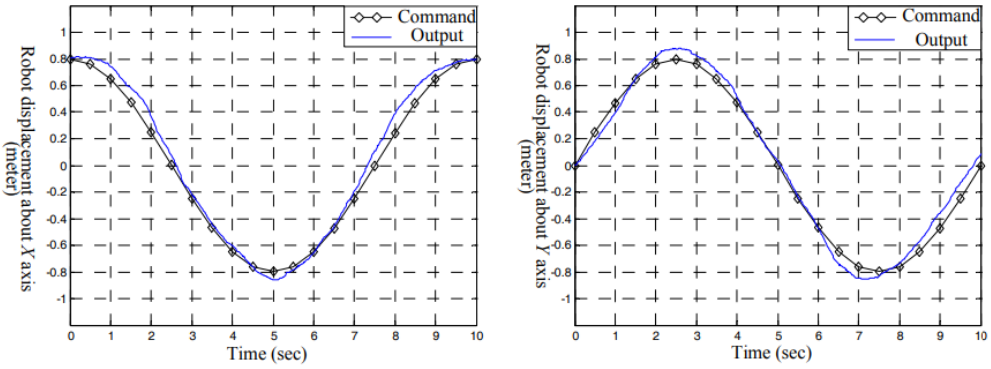
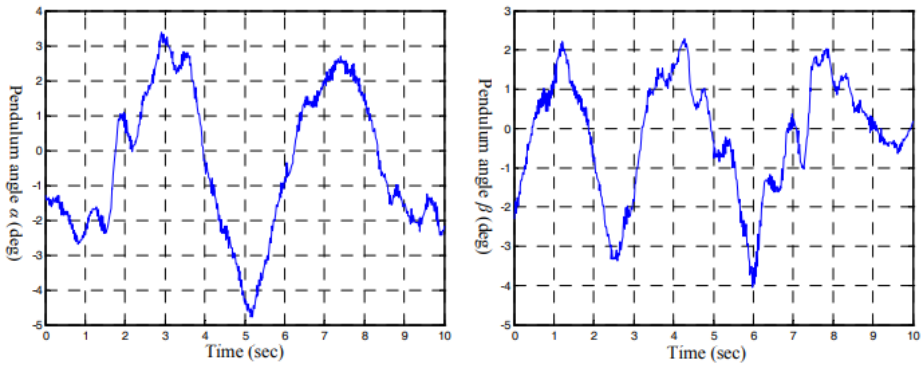


Figure 2.5

Pendulum's Angular Position Response Results



The results of experiment illustrate that the control law which has been designed can balance the spherical inverted pendulum system. The friction and slip which are not be used in dynamic model can cause by the system states small-amplitude oscillations.

2.2 Controller Design

Many researchers use various controllers to find the suitable the inverted pendulum robot's controller.

In [3], the researchers compared the performance of PID and LQR of the inverted pendulum that mounted on two wheels robot. From experimental tests, the researchers discovered that the created controllers can sustain the pendulum from a lower point and follow references until the amplitude reached 20 degrees. For all zero degrees, step, ramp, and parable signals input and output, the 20 ms sampling time loop is implemented. The comparative assessment of controllers is shown in Table 2.1.

Table 2.1

The Comparative Assessment of PID and LQR

	ISE	IAE	ITAE	Signal Reference
PID	10.86	3.35	22.64	Step
LQR	6.61	3.18	26.13	Step
PID	7.12	5.71	46.15	Ramp
LQR	5.56	4.48	32.38	Ramp
PID	7.12	5.71	46.15	Parable
LQR	3.03	3.26	27.85	Parable

Although PID controller can used for tuning and implementing easier, the results show that the LQR performs better control effort, steady state, and the rejection of the disturbance.

Another reference [4] presented the FLC systematic design using the model of Takagi-Sugeno for an inverted pendulum that mounted a cart problem. The control system is optimized to get the system's computational time lower by deducting the number of rule bases. It also includes the implementation and comparison of the PID and FLC. The

researchers use MATLAB Simulink to create the model of the inverted pendulum. The force that acts to the cart is disturbance, which is the input for the plant. The angle and angular velocity are used as feedback signal. It is used as the input of the system for computing the cart's and pendulum's acceleration and angular acceleration, respectively.

Figures 2.6 and 2.7 depict the membership functions.

Figure 2.6

Inputs Membership Functions

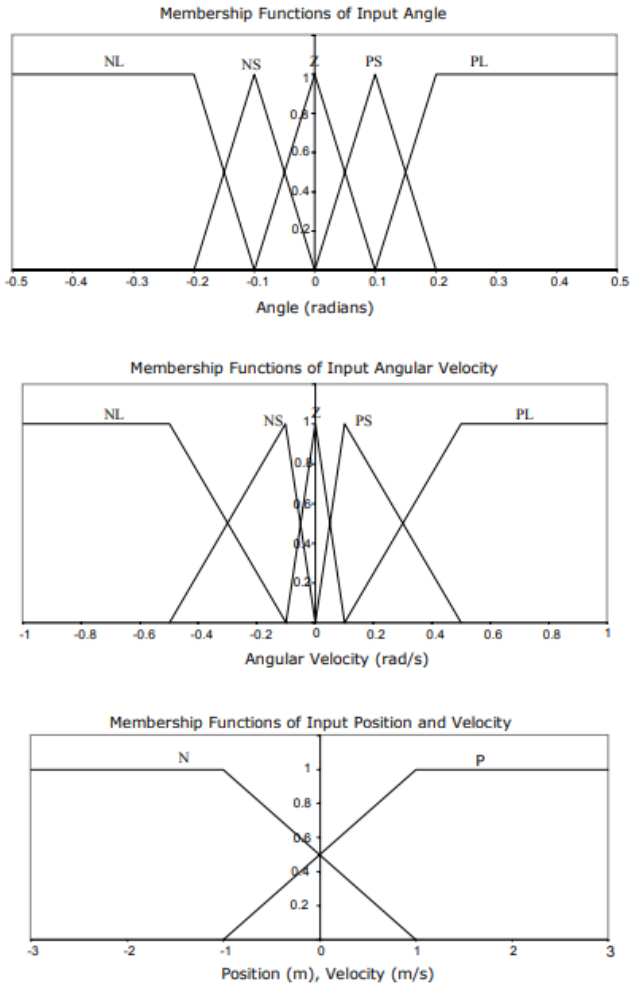


Figure 2.7

Output Membership Function

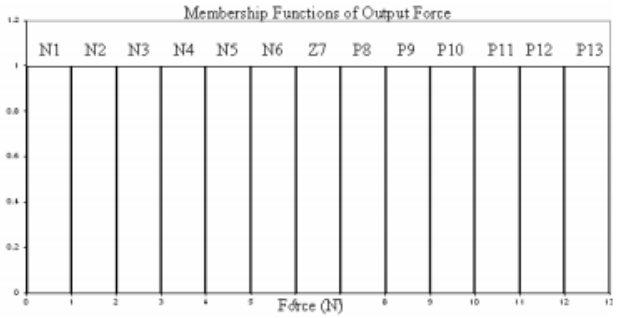


Table 2.2 shows the optimized rule-base, which labels x to be the platform's position, $\dot{x}$ to be the platform's velocity, θ to be the pendulum's falling angle, and $\dot{\theta}$ to be the pendulum angular velocity.

Table 2.2

Optimized Rule-Based

$x : \dot{x}$ / $\theta : \dot{\theta}$	N:N	N:P	P:N	P:P
N:N	N1	N2	N3	N4
N:P	N5	N6	N7	N8
P:N	P9	P10	P11	P12
P:P	P13	P14	P15	P16

The graphs of the results are shown in Figure 2.8 and Figure 2.9.

Figure 2.8

Response of Falling Angle for FLC (Mass Changed)

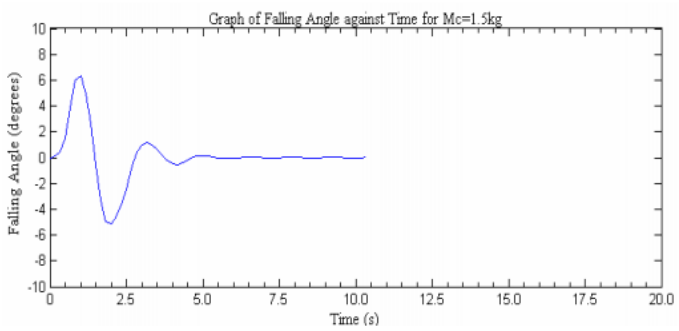
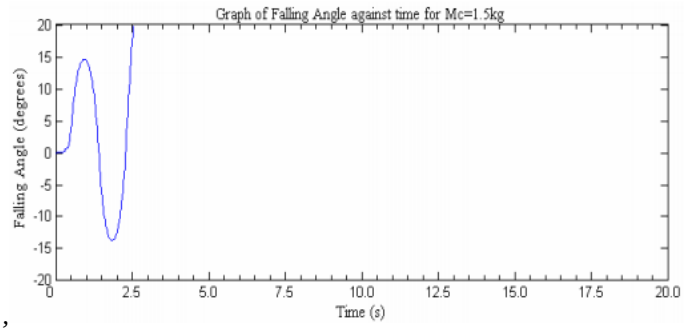


Figure 2.9

Response of Falling Angle for Conventional PID (Mass Changed)



,

The researchers concluded that FLC is simpler to implement because it uses a collection of control rules rather than a complex mathematical modeling technique. Furthermore, the findings demonstrated that the suggested FLC is more robust when compared to the PID controller.

Reference [5] compared two controller performances, SMC and PID controllers. This research determines the horizontal and vertical directions dynamic equations from an inverted pendulum free body diagram that mounts on a cart. Then, to design the suggested SMC and PID, the nonlinear terms in dynamic equations are used. The results are shown by Table 2.3 and Figure 2.10.

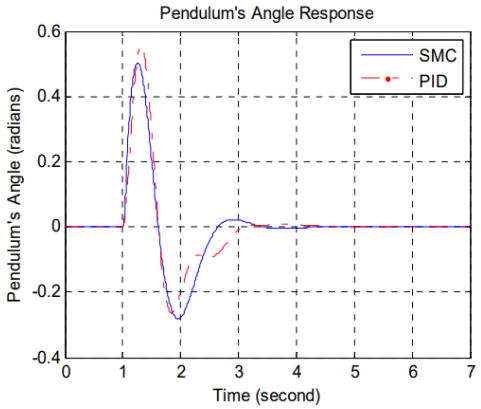
Table 2.3

Performance Characteristics Comparative Relation of Angular Position of Position Inverted Pendulum between SMC and PID

Time Response Specification	SMC	PID
Rise Time (s)	0.14	0.16
Settling Time (s)	3.03	2.93
Steady State Error	0.00	0.00
Maximum Overshoot (rad)	0.50	0.55

Figure 2.10

Nonlinear Inverted Pendulum's Angular Position Response



The results show that, controlling the system of nonlinear inverted pendulum, the SMC controller performs better than the PID controller.

Reference [6] is interesting. A PID controller, an optimal LQR, and an SMC Controller are developed. The researchers compared PID controller, an optimal LQR, and an SMC results shown in Table 2.4 and concluded that SMC presented the best integral square error for both without disturbance and with disturbance cases. Figure 2.11 represents the experimental outcomes of this research.

Figure 2.11

Experimental Results of the Body Pitch Angle Response without External Force Applied

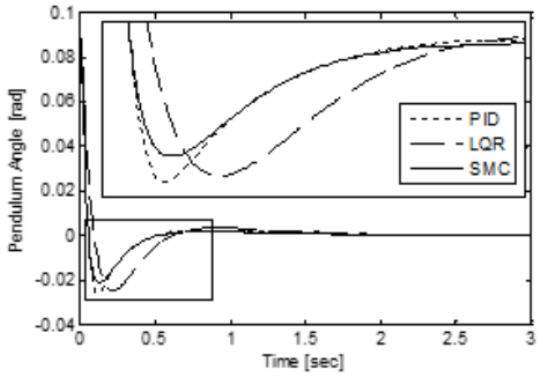


Table 2.4

ISE Comparative Relation of PID, LQR, and SMC without Disturbance

	PID	LQR	SMC
		ISE*10^{-3}	
Body Pitch Angle	0.271	0.393	0.247
Average Angular Wheel Position (right and left)	526.6	810	660.9

According to Figure 2.11, when using the SMC as a controller to stabilize the body pitch angle, it has a smooth response. In case of stabilizing the body pitch angle, the lowest ISE value comes from SMC, but in case of getting the angular wheel position steady state, PID control algorithm give the lowest ISE value, as shown in Table 2.4. The following experiment is to apply the external force to the robot. The external force is applied to the body pitch angle when the robot is stabilized. It is applied for testing the response to disturbances.

After test the controllers with a disturbance, Figure 2.12 and Table 2.5 show that the SMC has faster response than LQR and has less overshoot than PID.

Figure 2.12

Experimental Results of Body Pitch Angle Response when Applying External Force

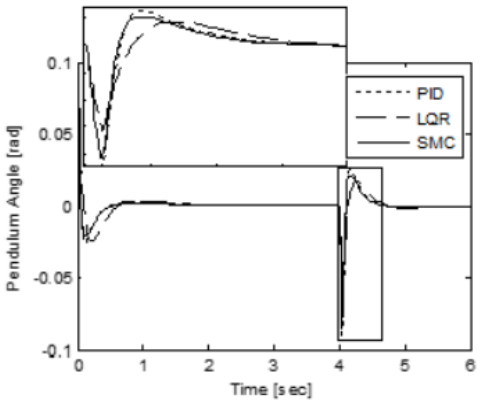


Table 2.5

ISE Comparative Relation of PID, LQR, and SMC with Disturbance

	PID	LQR	SMC
		ISE*10^{-3}	
Body Pitch Angle	0.666	0.622	0.572
Average Angular Wheel Position (right and left)	653.2	947.9	768.2

The two-wheel inverted pendulum in reference [7] compares the performance of three controllers: LQR, Fuzzy Logic, and PID controllers. To use LQR as a controller algorithm, a state space representation of the system with linear mathematical model must be defined. This controller aims to minimize the cost function. The researchers chose the Fuzzy Logic controller because to use this controller is simple. It does not need any mathematical model and uses a simple mathematical calculation to simulate the knowledge of expert. In contrast, the other control systems use a model to define a controller for the plant. The following figure and table show the result.

Figure 2.13 will compare the tilt position of the two-wheel inverted pendulum robot between PID, Fuzzy, and LQR Controller. Moreover, Table 2.6 will show the comparative assessment of all these three controllers.

Figure 2.13

Tilt Position of the Two-Wheel Inverted Pendulum Robot

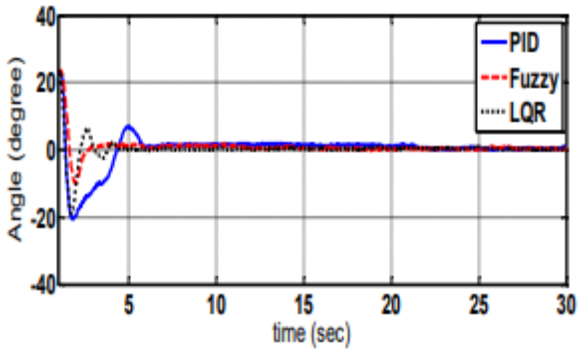


Table 2.6

Comparative Assessment of FLC, LQR, and PID

Controllers	Rise Time (s)	Settling Time (s)	% Overshoot	Current (A) [Max]	Horizontal Distance Absolute (m)
FLC	1.25	2.5	37	8.7	0.05
LQR	0.8	4.2	80	26.13	0.2
PID	0.8	6.5	90	27.85	0.23

From the results, when using FLC as a controller the system has faster response and less overshoot but requires more current than the other two controller.

To conclude, Table 2.7 to Table 2.11 show the performance comparative relation of the controllers used in [3] to [7].

Table 2.7

The Performance Comparative Relation of PID and LQR for Controlling the Pendulum Angle Position (Step Signal) of [3]

	PID	LQR
Rise Time (s)	0.2	0.2
Settling Time (s)	4.3	1.25
Maximum Overshoot (deg)	5.8	5

Table 2.8

The Performance Comparative Relation of PID and FLC for Controlling the Pendulum Angle Position of [4]

	PID	FLC
Rise Time (s)	0.25	0.75
Settling Time (s)	6.5	3.5
Maximum Overshoot (deg)	6.9	5.7

Table 2.9

The Performance Comparative Relation of PID and SMC for Controlling the Pendulum Angle Position of [5]

	PID	SMC
Rise Time (s)	0.14	0.16
Settling Time (s)	3.03	2.93
Maximum Overshoot (rad)	0.05	0.55

Table 2.10

The Performance Comparative Relation of PID, LQR, and SMC for Controlling the Pendulum Angle Position of [6]

	PID	LQR	SMC
Rise Time (s)	0.195	0.260	0.195
Settling Time (s)	0.695	0.750	0.663
Maximum Overshoot (rad)	0.027	0.024	0.025

Table 2.11

The Performance Comparative Relation of PID, LQR, and FLC for Controlling the Pendulum Angle Position of [7]

	PID	LQR	FLC
Rise Time (s)	0.8	0.8	1.25
Settling Time (s)	6.5	4.2	2.5
%Overshoot	90	80	37

CHAPTER 3
METHODOLOGY

This chapter expounds on the robot's design, including the mechanical and electrical design. For mechanical design, two models are included in this thesis: The Actual Model and the Simulation Model.

3.1 Mechanical Design

The body of the system consists of the pendulum rod, the joint of the rod, and the omnidirectional wheels platform.

3.1.1 CAD Model and Final Design of the System

Solid Works 2017 software is used to design the CAD model.

Figure 3.1

CAD Model of Actual Model

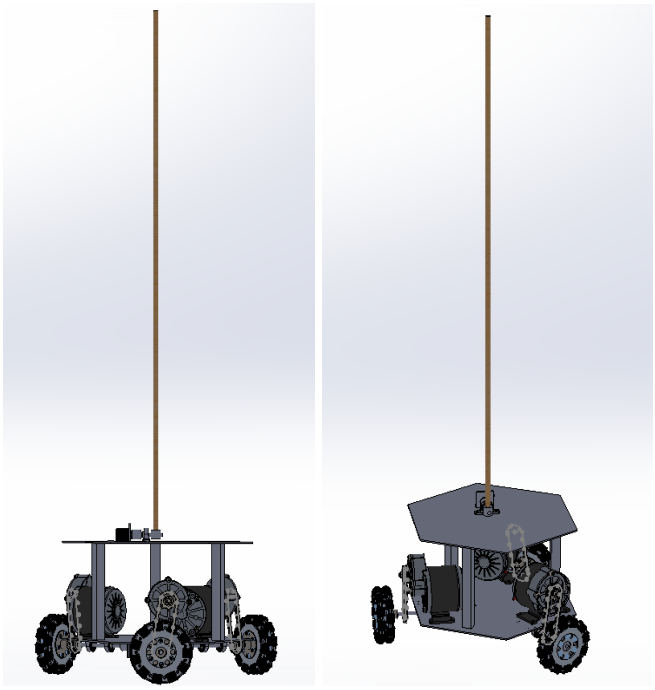


Figure 3.2

Final Design of Actual Model

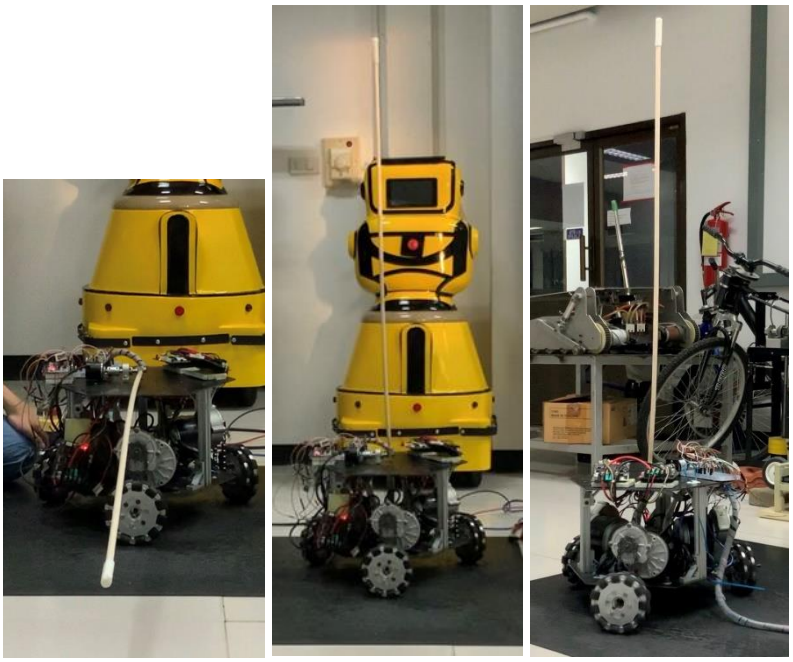


Figure 3.3

CAD Model of Simulation Model

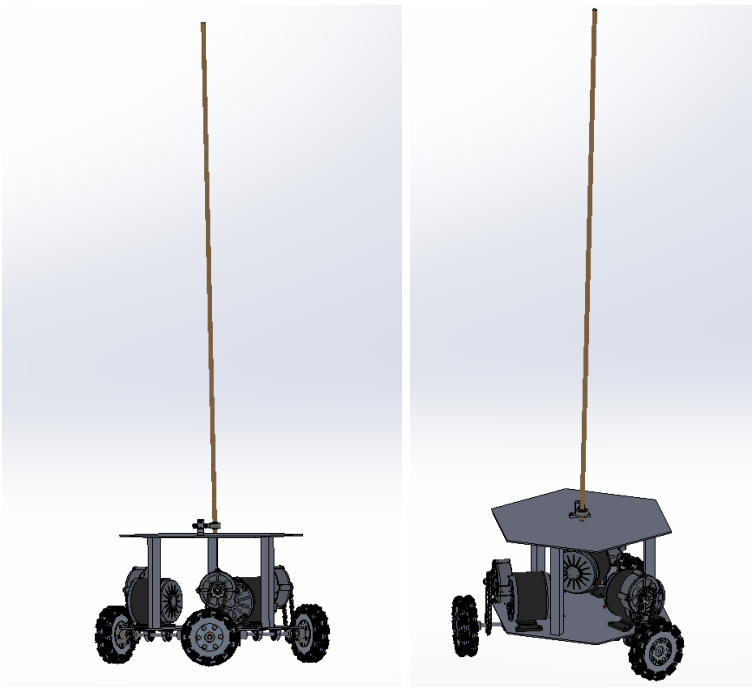
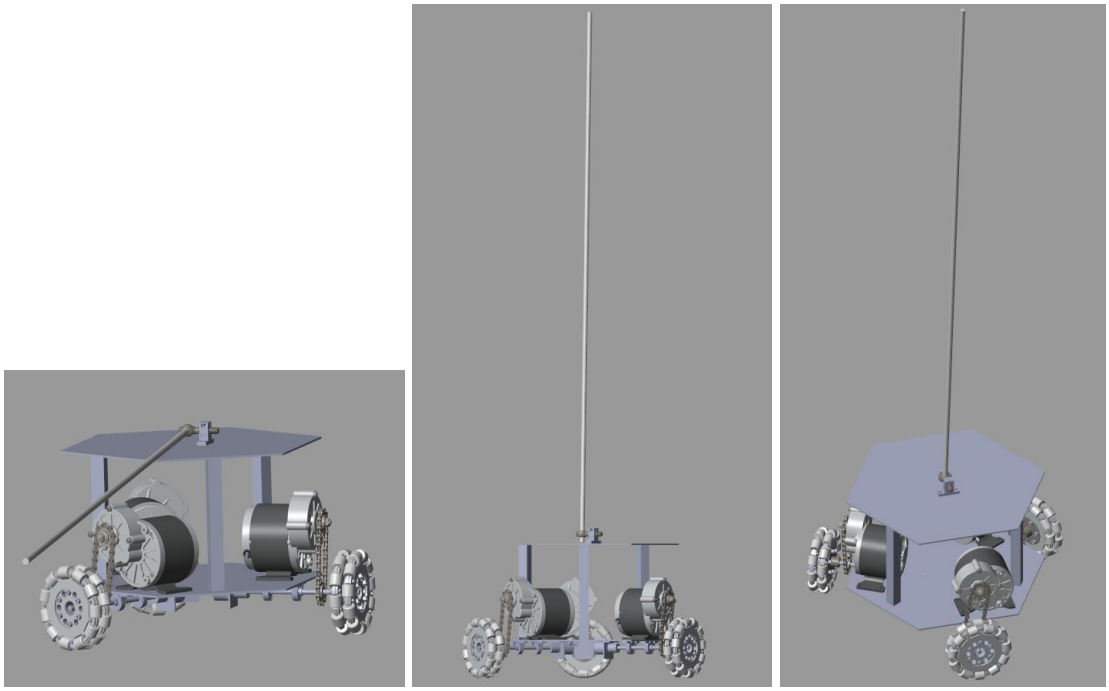


Figure 3.4

Final Design of Simulation Model



3.1.2 Pendulum Rod

The cylindrical Balsa wood is used as the pendulum rod. The diameter is 1 cm and the length is 112 cm. The weight of the rod is 18.4 g.

Figure 3.5

Pendulum rod

3.1.3 Joint of the Pendulum Rod

Two types of joints will be used for this thesis. The first joint is the revolute joint. It is used for the Actual Model. Another is spherical, which is used for the Simulation Model.

Because the actual x-y axis pendulum tilt angle of the proposed model with the spherical joint cannot be detected by the IMU sensor, the revolute joint is used for the Actual Model for the thesis experiment.

Figure 3.6

IMU Sensor

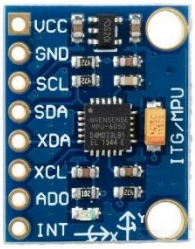
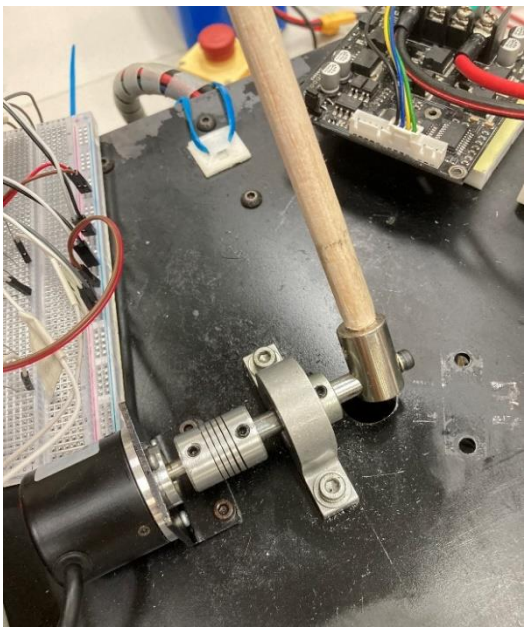


Figure 3.7

Revolute Joint of Actual Model



Because this thesis proposal aims to balance the spherical inverted pendulum, the Simulation Model with the spherical joint is invented to complete the thesis proposal. The Simulation Model is operated by Simscape in MATLAB Simulink, as shown in Figure 3.8.

Figure 3.8

Block Diagram in MATLAB Simulink of Simulation Model

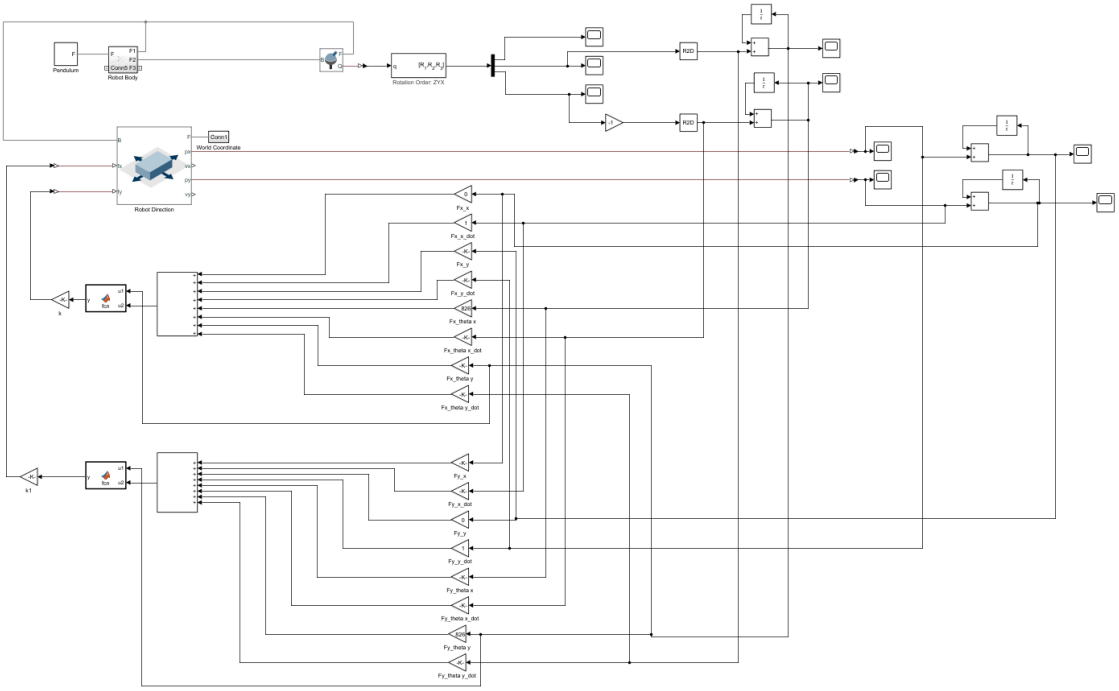
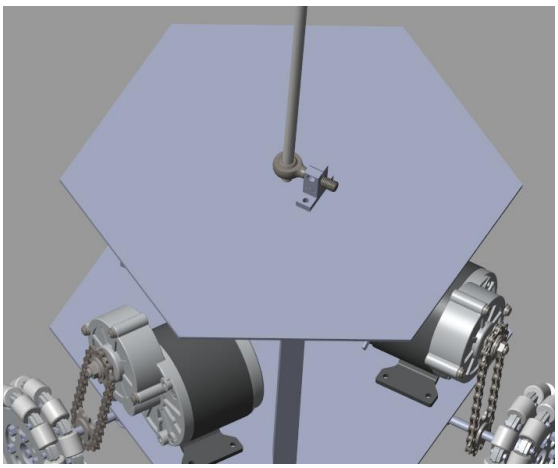


Figure 3.9

Spherical Joint of Simulation Model

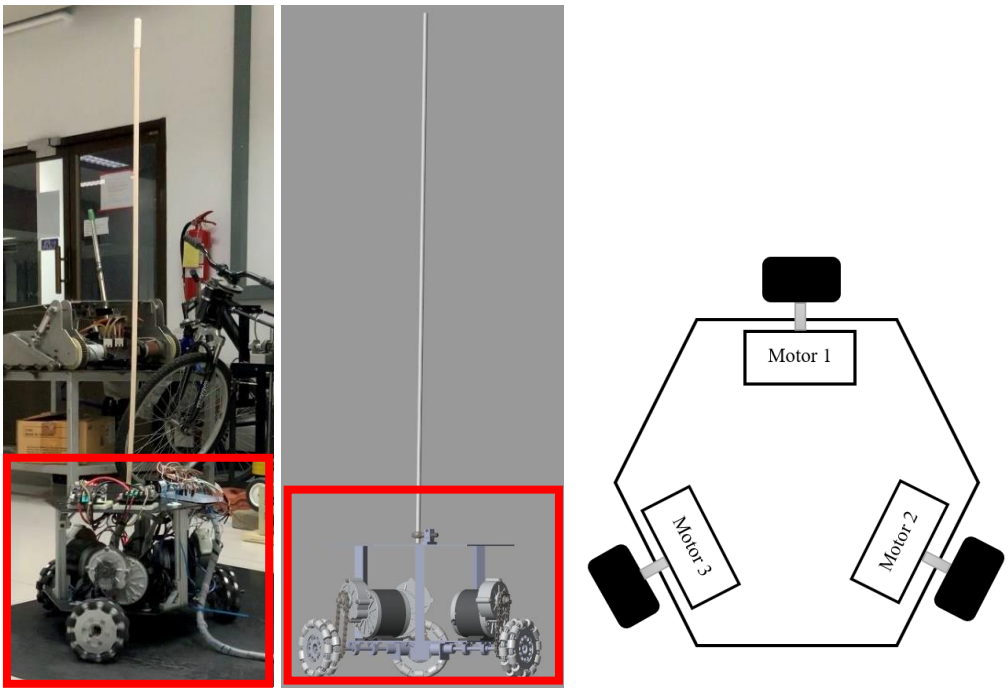


3.1.4 Omnidirectional Wheels Platform

Both Actual and Simulation Models have the same omnidirectional wheels platform. Three omnidirectional wheels are used to actuate the system to balance the rod. Therefore, three motors are used to drive the wheels.

Figure 3.10

Omnidirectional Wheels Platform and Motors Location



3.2 Electrical Design for Actual Model

There will be three parts in this section.

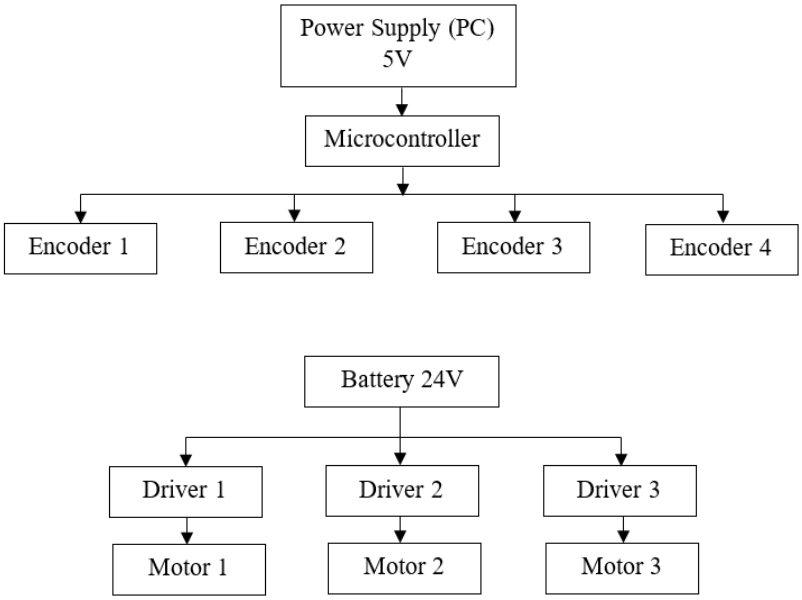
1. System of Power Supply
2. Actuators and Sensors
3. Other Components

3.2.1 System of Power Supply

Figure 3.11 shows the required power supply.

Figure 3.11

Block Diagram of the Power Supply System of the Robot



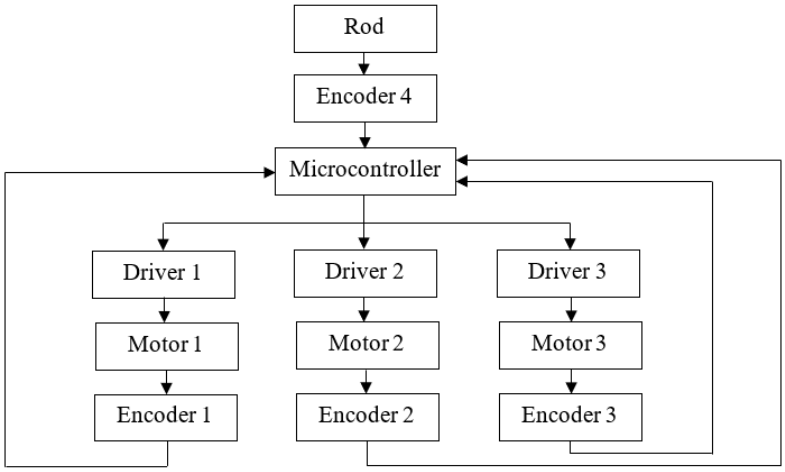
The microcontroller uses a power supply from the computer through a micro-USB port. The power for four encoders is supplied from a microcontroller 5V pin.

3.2.2 Actuators and Sensors

This thesis uses three motors to actuate the model. For the sensors, four encoders are used.

Figure 3.12

Block Diagram of the Electrical System of the Robot



3.2.2.1 **Motors.** The 24V, 350W DC gear motors are actuators. These motors' torque and speed are 10.8Nm and 393RPM, respectively.

Figure 3.13

Motor

3.2.2.2 **Encoders.** Three encoders are used to get the distance and velocity of three omnidirectional wheels. Another encoder is used to get the pendulum rod's tilt angle and angular velocity. Each encoder is a 360 P/R rotary encoder. To get the data of the wheels and pendulum, connect the signal A and B from each encoder to the microcontroller and connect the 5V and ground wires for the power supply.

Figure 3.14

Encoder

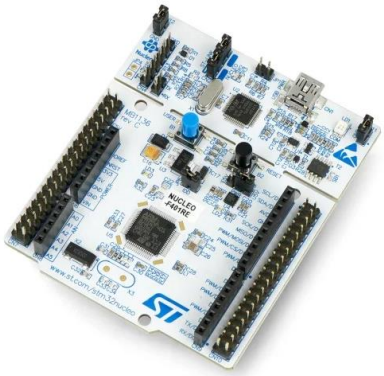
3.2.3 Other Components

3.2.3.1 **Microcontroller.** The STM32 Nucleo-64 F401-RE is chosen. The clock speed of this microcontroller is 84MHz. This microcontroller has a timer to get the data from the sensors. Therefore, the data from sensors are collected every 5 ms. After

collecting data from sensors, the input will be used to calculate the voltage of the output. Then, the output will be used to control the motors. For controlling the motors, each motor must connect to the driver. There are five wires to connect the driver and the motor. The first wire is the PWM wire; this signal will control the PWM of the motor. The second and third wires are the direction wire. When the signal from the second wire is sent, the motor will rotate clockwise. The third wire is for the rotating motor in a counterclockwise direction. The fourth and fifth wires are connected for the motor power supply.

Figure 3.15

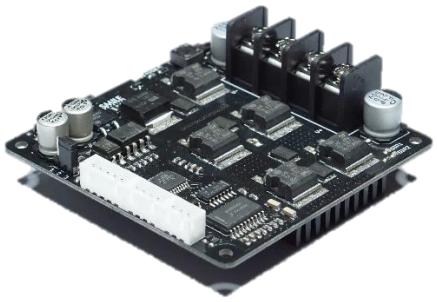
Microcontroller



3.2.3.2 **Drivers.** The EVO24V50 is chosen. Max-driven current is 36A for 24 DC voltage. It is the optoisolator driver. And it has a reverse polarity power supply protection circuit. Therefore, this driver can protect the microcontroller.

Figure 3.16

Driver



CHAPTER 4
DYNAMIC MODEL AND CONTROL

To conduct the control system, the dynamic model is required for balance the pendulum rod. The first part of this chapter, two dynamic models are shown. The next part is the state space of each dynamic model. Another part is Linear Quadratic Regulator calculation.

4.1 Dynamic Models

This section will show all two dynamic models: Simulation Dynamic Model and Actual Dynamic Model. Table 4.1 and Figure 4.1 show the differences between these two models.

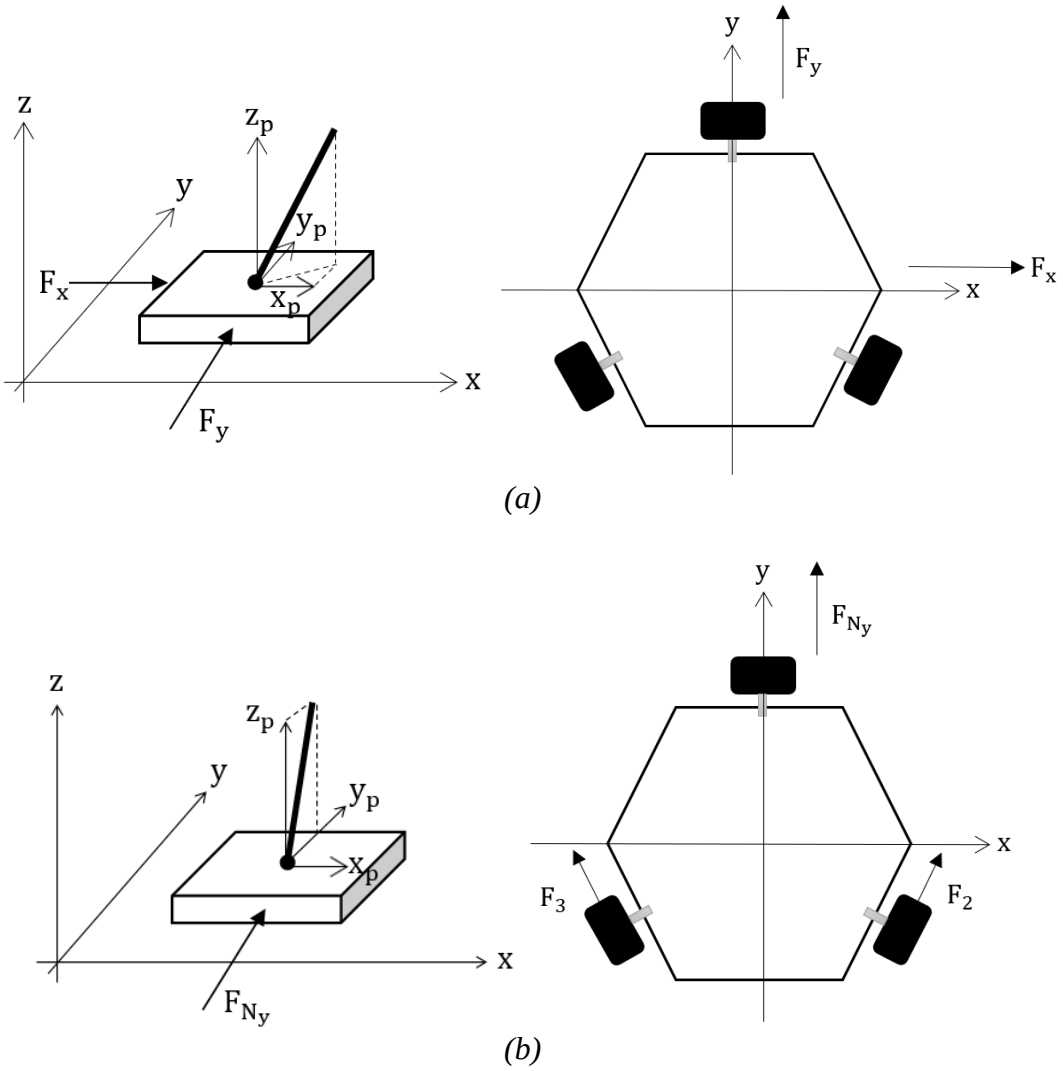
Table 4.1

The Comparative Relation of Simulation Dynamic Model and Actual Dynamic Model

	Simulation Model	Actual Model
Pendulum Rod	Cylindrical	Cylindrical
Weight (kg)	18.4×10^{-2}	18.4×10^{-2}
Diameter × Length (cm)	1 × 112	1 × 112
Omnidirectional Wheel Platform Weight (kg)	18.9	18.9
Joint of Pendulum Rod	Spherical	Revolute
Actuating Force	Force in the x and y axes	Force in one-axis (y-axis)
Plane Movement	x and y planes	One-plane (y-axis)

Figure 4.1

The Difference of the Pendulum Axis and Platform Movement between (a) Simulation Model and (b) Actual Model



In Figure 4.1(a), this model is simulated by Simscape in Simulink MATLAB. In Simscape, the forces can act directly on the platform in the x and y axes. Therefore, the mobile robot moves in the x-y axis from F_x, F_y force. The pendulum of this model is mounted by a spherical joint, which means that the pendulum swings in the x-y axis.

x, y, z : Robot World Coordinate

x_p, y_p, z_p : Pendulum Coordinate

F_x : Actuated Force in x-axis

F_y : Actuated Force in y-axis

In Figure 4.1(b), Because of the IMU sensor limitation, which cannot get the tilt angle of x-y axis pendulum correctly, this model is invented. The pendulum is mounted by a revolute joint. So, the pendulum swings in only one-axis (y-axis). The platform is move in only one-axis (y-axis) by F_2, F_3 net force.

x, y, z : Robot World Coordinate

x_p, y_p, z_p : Pendulum Coordinate

F_2 : Actuated Force of Wheel 2 Motor

F_3 : Actuated Force of Wheel 3 Motor

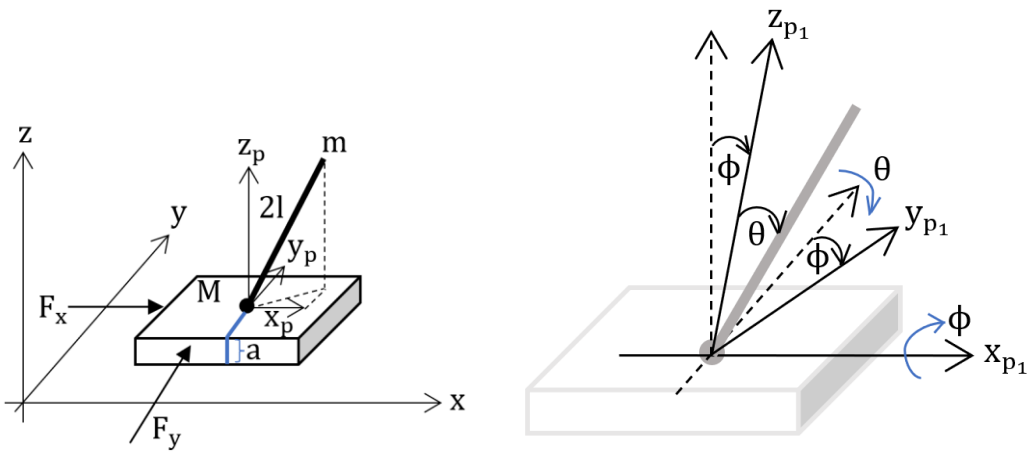
F_{N_y}: Net Force in y-axis of F_2, F_3

4.1.1 Spherical Inverted Pendulum Dynamic Model

The equations of motion of a 2-DOF joint inverted pendulum with two external forces are shown in the reference [8]. Figure 4.2 shows 2-DOF joint inverted pendulum with two external forces.

Figure 4.2

2-DOF Joint Inverted Pendulum with Two External Forces



The pendulum rotates around the x and y axes are denoted as ϕ and θ, respectively. x_{C_1} and x_{C_2} denote the center of mass positions of M and m, respectively.

$$x_{C_1} = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}, x_{C_2} = \begin{bmatrix} x + l\sin\theta \\ y + l\cos\theta\sin\phi \\ a + l\cos\theta\cos\phi \end{bmatrix} \quad (1)$$

The linear velocities are given by

$$\dot{x}_{C_1} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix}, \dot{x}_{C_2} = \begin{bmatrix} \dot{x} + l\cos\theta\dot{\theta} \\ \dot{y} - l\sin\theta\sin\phi\dot{\theta} + l\cos\theta\cos\phi\dot{\phi} \\ -l\sin\theta\cos\phi\dot{\theta} - l\cos\theta\sin\phi\dot{\phi} \end{bmatrix} \tag{2}$$

The kinematic and potential energy of this system are

$$T = \frac{1}{2}M\dot{x}_{C_1}^T\dot{x}_{C_1} + \frac{1}{2}m\dot{x}_{C_2}^T\dot{x}_{C_2}, V(q) = mgl\cos\theta\cos\phi \tag{3}$$

The Euler-Lagrange motion equations for this system with $\mathrm{q} = (\mathrm{x}, \mathrm{y}, \theta, \phi)^{\mathrm{T}}$ are

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{q}} - \frac{\partial\mathcal{L}}{\partial q} = \begin{bmatrix} F_x \\ F_y \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

The motion equations for this system can be written as

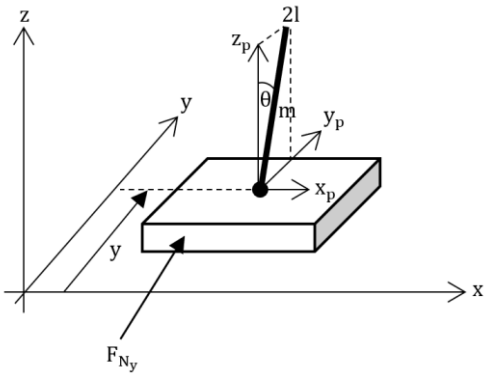
$$\begin{aligned} (M+m)\ddot{x} + ml\cos\theta\ddot{\theta} &= ml\sin\theta\dot{\theta}^2 + F_x \\ (M+m)\ddot{y} - ml\sin\theta\sin\phi\ddot{\theta} &= ml\cos\theta\sin\phi\,\dot{\theta}^2 \\ +ml\cos\theta\cos\phi\ddot{\phi} &+ 2ml\sin\theta\cos\phi\dot{\theta}\dot{\phi} \\ &+ ml\cos\theta\sin\phi\,\dot{\phi}^2 + F_y \\ ml\cos\theta\,\ddot{x} - ml\sin\theta\sin\phi\,\ddot{y} + ml^2\ddot{\theta} &= mgl\sin\theta\cos\phi \\ &- ml^2\sin\theta\cos\theta\dot{\phi}^2 \\ ml\cos\theta\cos\phi\ddot{y} + ml^2\cos^2\theta\ddot{\phi} &= mgl\cos\theta\sin\phi \\ &+ 2ml^2\sin\theta\cos\theta\dot{\theta}\dot{\phi} \end{aligned} \tag{5}$$

4.1.2 Inverted Pendulum with Revolute Joint Dynamic Model

The revolute joint is the joint that pendulum falls in only one-axis. For this thesis, the y-axis is chosen which is shown in Figure 4.3.

Figure 4.3

Inverted Pendulum with Revolute Joint



The tilt angle of the pendulum rod is denoted as θ. The y_m and z_m denote the center of mass positions of m along y-axis and z-axis, respectively.

$$y_m = y + l \sin\theta \ , z_m = l \cos\theta \tag{6}$$

The linear velocities are given by

$$\dot{y}_m = \dot{y} + l \cos\theta\dot{\theta} \ , \dot{z}_m = -l \sin\theta\dot{\theta} \tag{7}$$

The kinematic and potential energy of this system are

$$T = \frac{1}{2}M\dot{y}^2 + \frac{1}{2}m(\dot{y}_m + \dot{z}_m)^2 \ , V(q) = mgl \cos\theta \tag{8}$$

The Euler-Lagrange motion equations for this system with $q = (y, \theta)^T$ are

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \begin{bmatrix} F_y \\ 0 \end{bmatrix} \tag{9}$$

The motion equations for this system can be written as

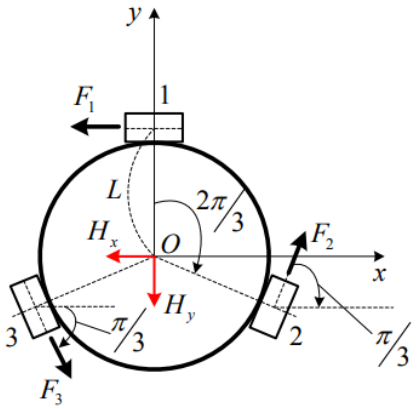
$$\begin{aligned} (M+m)\ddot{y} + ml\cos\theta\ddot{\theta} &= ml\sin\theta\dot{\theta}^2 + F_y \\ ml\cos\theta\,\ddot{y} + ml^2\ddot{\theta} &= mgl\sin\theta \end{aligned} \tag{10}$$

4.1.3 Three Omnidirectional Wheels Platform for Y-Plane Movement Dynamic Model

To find the net force in y-plane for moving the platform in y-axis direction, the forces that are used to move the platform in x-y plane must be defined first. Figure 4.4 shows the omnidirectional wheels mobile platform which is move in the x-y plane.

Figure 4.4

Omnidirectional Wheels Mobile Platform Movement in the X-Y Plane

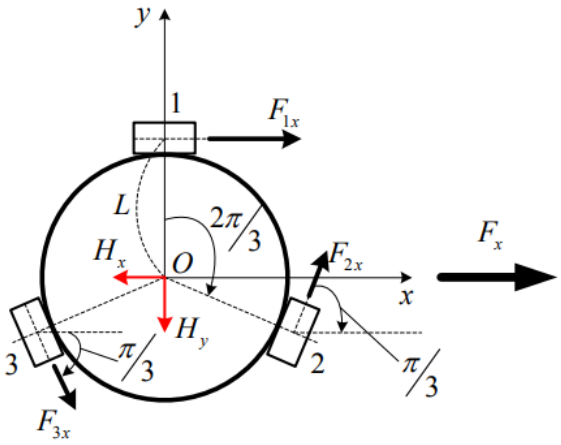


From [9], to find the motion of the omnidirectional mobile platform, some limitations are set. The limitations for platform movement in x-axis are the omnidirectional mobile platform moves in the x-axis direction only without any rotation, only the component on x-axis that generates the total external forces generated on three wheels, and the y component is zero. From the limitation, the followings are obtained:

$$\begin{cases} F_1 L + F_2 L + F_3 L = 0 \\ -F_1 + F_2 \cos\frac{\pi}{3} + F_3 \cos\frac{\pi}{3} = F_x \\ F_2 \sin\frac{\pi}{3} - F_3 \sin\frac{\pi}{3} = 0 \end{cases} \quad (11)$$

Figure 4.5

Omnidirectional Mobile Platform Moving on the X-Axis



From Equation 11, F_{1x}, F_{2x}, and F_{3x} are the solution. The followings are obtained:

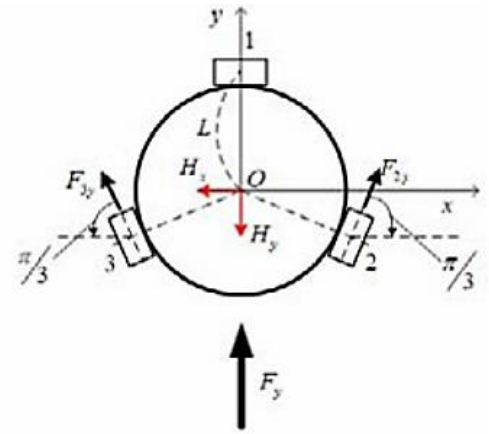
$$\begin{cases} F_{1x} = -\frac{2}{3}F_x \\ F_{2x} = \frac{1}{3}F_x \\ F_{3x} = \frac{1}{3}F_x \end{cases} \tag{12}$$

Similar to the x-axis, the y-axis direction, the followings are obtained:

$$\begin{cases} F_1 L + F_2 L + F_3 L = 0 \\ F_2 \sin\frac{\pi}{3} - F_3 \sin\frac{\pi}{3} = F_y \\ -F_1 + F_2 \cos\frac{\pi}{3} + F_3 \cos\frac{\pi}{3} = 0 \end{cases} \tag{13}$$

Figure 4.6

Omnidirectional Mobile Platform Moving on the Y-Axis



The solutions of Equation 13 are F_{1y}, F_{2y}, and F_{3y}. The following are obtained:

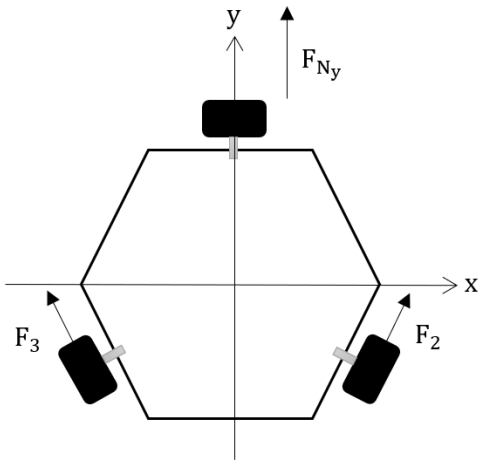
$$\begin{cases} F_{1y} = 0 \\ F_{2y} = \dfrac{1}{\sqrt{3}} F_y \\ F_{3y} = -\dfrac{1}{\sqrt{3}} F_y \end{cases} \tag{14}$$

Because the motion of each direction is independent, therefore the force on each wheel is as follows:

$$\begin{cases} F_1 = F_{1x} + F_{1y} = -\dfrac{2}{3} F_x \\ F_2 = F_{2x} + F_{2y} = \dfrac{1}{3} F_x + \dfrac{1}{\sqrt{3}} F_y \\ F_3 = F_{3x} + F_{3y} = \dfrac{1}{3} F_x - \dfrac{1}{\sqrt{3}} F_y \end{cases} \tag{15}$$

Figure 4.7

Omnidirectional Mobile Platform for Y-Plane Movement



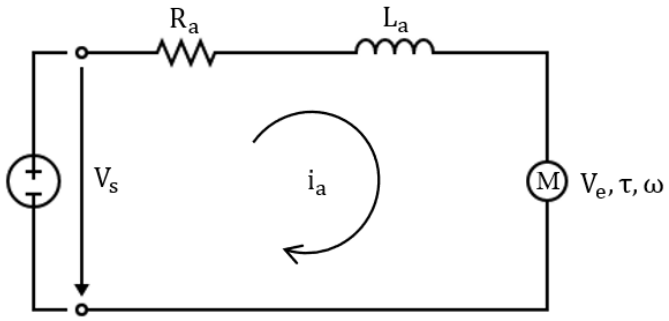
From Equation 15, the force which acts on Wheel 2 and Wheel 3 is equal but different direction. The dynamic models are as follows:

$$F_2 = \frac{1}{\sqrt{3}} F_y$$
$$F_3 = -\frac{1}{\sqrt{3}} F_y \quad (16)$$

4.1.4 Motor Dynamic Model

Figure 4.8

DC Motor Circuit



V_s : Voltage Terminal to Apply Voltage Source

R_a : Armature Resistance of motor

L_a : Armature Inductance of motor

i_a : Armature Current

V_e : Back EMF Voltage

τ : Torque Impose Under the Shaft

ω : Motor Angular Velocity

K_e : Voltage constant of motor

K_t : Torque constant of motor

Because the motor's back EMF voltage is proportional to motor angular velocity and the motor torque is proportional to armature current, the following are obtained.

$$\begin{aligned} V_e \propto \omega, V_e &= K_e \omega \\ \tau \propto i_a, \ \ \tau &= K_t i_a \end{aligned} \tag{17}$$

Apply Kirchhoff's Voltage Law to DC motor circuit

$$V_s - R_a i_a - L_a \frac{di_a}{dt} - V_e = 0 \tag{18}$$

Because when comparing armature inductance with armature resistance of motor, the armature inductance is very low. Therefore, the armature inductance of motor is assumed to be neglected.

$$V_s - R_a i_a - V_e = 0 \tag{19}$$

Substitute i_a and V_e from Equation 17 to Equation 19

$$\begin{aligned} V_s - \frac{\tau}{K_t} R_a - K_e \omega &= 0 \\ V_s - K_e \omega &= \frac{\tau}{K_t} R_a \\ \tau &= (V_s - K_e \omega) \frac{K_t}{R_a} \end{aligned} \tag{20}$$

According to the three omnidirectional wheels platform is moved by force, therefore, from Equation 20, the torque must be derived to get the force that actuate on platform. Because of $\tau = Fr$, which F is force and r is radius of the wheel, the following are obtained

$$\begin{aligned} Fr &= (V_s - K_e\omega)\frac{K_t}{R_a} \\ F &= \frac{V_s K_t}{R_a r} - \frac{K_e K_t \omega}{R_a r} \end{aligned} \tag{21}$$

4.2 State Space Representation

In this section, two state space are represented.

1. Simulation Model State Space
2. Actual Model State Space

Each state space can be represented as.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{22}$$

A : State Matrix

B : Input Matrix

C : Output Matrix

x : State Variable

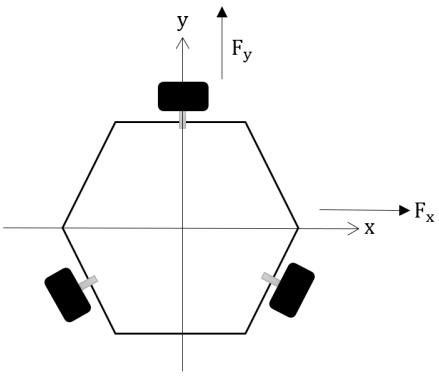
u : Input

4.2.1 Simulation Model State Space

Before getting the state space of this model, the dynamic model must be derived.

Figure 4.9

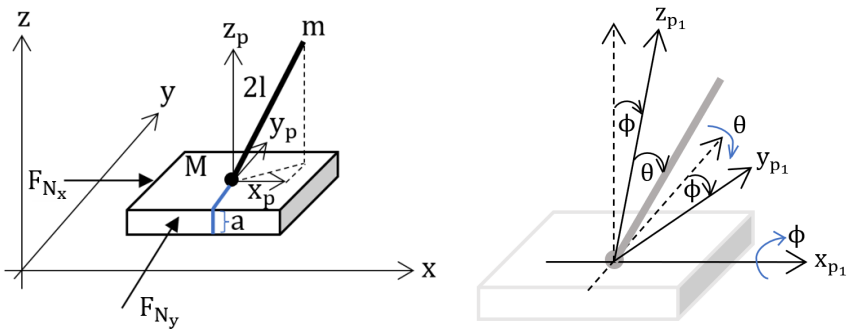
Platform Movement of Simulation Model



From Figure 4.9, the forces that use to actuate the platform are $F_x[t]$ and $F_y[t]$.

Figure 4.10

Spherical Inverted Pendulum with Two External Net Forces



From Equation 5,

$$
\begin{aligned}
(M+m)\ddot{x}[t] + ml\cos[\theta[t]]\ddot{\theta}[t] &= ml\sin[\theta[t]]\dot{\theta}^2[t] + F_x[t] \\
(M+m)\ddot{y}[t] - ml\sin\big[\theta[t]\big]\sin[\phi[t]]\ddot{\theta}[t] &= ml\cos[\theta[t]]\sin[\phi[t]]\,\dot{\theta}^2[t] \\
+ml\cos[\theta[t]]\cos\phi\ddot{\phi}[t] &+ 2ml\sin[\theta[t]]\cos[\phi[t]]\dot{\theta}[t]\dot{\phi}\,[t] \\
&+ ml\cos[\theta[t]]\sin[\phi[t]]\,\dot{\phi}^2[t] \\
&+ F_y[t] \qquad (23) \\
ml\cos\big[\theta[t]\big]\ddot{x}[t] - ml\sin\big[\theta[t]\big]\sin[\phi[t]]\ddot{y}[t] &= \mathrm{mgl}\sin[\theta[t]]\cos[\phi[t]] \\
+ml^2\ddot{\theta}[t] &- \mathrm{ml}^2\sin[\theta[t]]\cos[\theta[t]]\dot{\phi}^2\,[t] \\
ml\cos[\theta[t]]\cos[\phi[t]]\ddot{y}[t] + ml^2\cos^2[\theta[t]]\ddot{\phi}[t] &= mgl\cos[\theta[t]]\sin[\phi[t]] \\
&+ 2ml^2\sin[\theta[t]]\cos\big[\theta[t]\big]\dot{\theta}[t]\dot{\phi}[t]
\end{aligned}
$$

forming the Equation 23 as matrix equation.

$$\begin{pmatrix} M+m & 0 & 0 & ml\cos[\theta[t]] \\ ml\cos[\theta[t]] & -ml\sin[\theta[t]]\sin[\phi[t]] & 0 & ml^2 \\ 0 & ml\cos[\theta[t]]\cos[\phi[t]] & ml^2\cos^2[\theta[t]] & 0 \\ 0 & M+m & ml\cos[\theta[t]]\cos\phi & ml\sin[\theta[t]]\sin[\phi[t]] \end{pmatrix}\begin{pmatrix} \ddot{x}[t] \\ \ddot{y}[t] \\ \ddot{\phi}[t] \\ \ddot{\theta}[t] \end{pmatrix} \quad (24)$$
$$= \begin{pmatrix} ml\sin[\theta[t]]\dot{\theta}^2[t] + F_x[t] \\ mgl\sin[\theta[t]]\cos[\phi[t]] - ml^2\sin[\theta[t]]\cos[\theta[t]]\dot{\phi}^2[t] \\ mgl\cos[\theta[t]]\sin[\phi[t]] + 2ml^2\sin[\theta[t]]\cos[\theta[t]]\dot{\theta}[t]\dot{\phi}[t] \\ ml\cos[\theta[t]]\sin[\phi[t]]\dot{\theta}^2[t] + 2ml\sin[\theta[t]]\cos[\phi[t]]\dot{\theta}[t]\dot{\phi}[t] + ml\cos[\theta[t]]\sin[\phi[t]]\dot{\phi}^2[t] + F_y[t] \end{pmatrix}$$

Assigning matrix P and Q as

$$P = \begin{pmatrix} M+m & 0 & 0 & ml\cos[\theta[t]] \\ ml\cos[\theta[t]] & -ml\sin[\theta[t]]\sin[\phi[t]] & 0 & ml^2 \\ 0 & ml\cos[\theta[t]]\cos[\phi[t]] & ml^2\cos^2[\theta[t]] & 0 \\ 0 & M+m & ml\cos[\theta[t]]\cos\phi & ml\sin[\theta[t]]\sin[\phi[t]] \end{pmatrix}$$

$$Q = \begin{pmatrix} ml\sin[\theta[t]]\dot{\theta}^2[t] + F_x[t] \\ mgl\sin[\theta[t]]\cos[\phi[t]] - ml^2\sin[\theta[t]]\cos[\theta[t]]\dot{\phi}^2[t] \\ mgl\cos[\theta[t]]\sin[\phi[t]] + 2ml^2\sin[\theta[t]]\cos[\theta[t]]\dot{\theta}[t]\dot{\phi}[t] \\ ml\cos[\theta[t]]\sin[\phi[t]]\dot{\theta}^2[t] + 2ml\sin[\theta[t]]\cos[\phi[t]]\dot{\theta}[t]\dot{\phi}[t] + ml\cos[\theta[t]]\sin[\phi[t]]\dot{\phi}^2[t] + F_y[t] \end{pmatrix} \quad (25)$$

Then,

$$P\begin{pmatrix} \ddot{x}[t] \\ \ddot{y}[t] \\ \ddot{\phi}[t] \\ \ddot{\theta}[t] \end{pmatrix} = Q \quad (26)$$

To get $(\ddot{x}[t] \quad \ddot{y}[t] \quad \ddot{\phi}[t] \quad \ddot{\theta}[t])^T$, Equation 26 must be derived as follows.

$$\begin{pmatrix} \ddot{x}[t] \\ \ddot{y}[t] \\ \ddot{\phi}[t] \\ \ddot{\theta}[t] \end{pmatrix} = P^{-1}Q \quad (27)$$

Calculating Equation 27.

$$\ddot{x}[t] = -\frac{\begin{matrix}2((m+4M+mcos[2\theta[t]]-\frac{1}{2}mcos[2(\theta[t]-\phi[t])]-mcos[2\phi[t]]\\ -\frac{1}{2}mcos[2(\theta[t]+\phi[t])])F_x[t]+4msin[\theta[t]](-cos[\theta[t]]sin[\phi[t]]F_y[t]+lM\dot{\theta}^2[t]\\ +Mcos[\theta[t]](-gcos[\phi[t]]+lcos[\theta[t]]\dot{\phi}^2[t])))\end{matrix}}{M(-6m-8M+2mcos[2\theta[t]]+mcos[2(\theta[t]-\phi[t])]+2mcos[2\phi[t]]+mcos[2(\theta[t]+\phi[t])])}$$

$$\ddot{y}[t] = -\frac{\begin{matrix}2(-2msin[2\theta[t]]sin[\phi[t]]F_x[t]+(2m+4M-2mcos[2\theta[t]])F_y[t]\\ +4mMcos[\theta[t]]sin[\phi[t]](-gcos[\theta[t]]cos[\phi[t]]+l\dot{\theta}^2[t]+lcos[\theta[t]]^2\dot{\phi}^2[t]))\end{matrix}}{M(-6m-8M+2mcos[2\theta[t]]+mcos[2(\theta[t]-\phi[t])]+2mcos[2\phi[t]]+mcos[2(\theta[t]+\phi[t])])}$$

$$\ddot{\theta}[t] = -\frac{\begin{matrix}8(\frac{1}{2}cos[\theta[t]](-m-2M+mcos[2\phi[t]])F_x[t]+sin[\theta[t]]((m+M)sin[\phi[t]]F_y[t]\\ -lmMcos[\theta[t]]\,cos^2[\phi[t]]\,\dot{\theta}^2[t]+M(m+M)(gcos[\phi[t]]-lcos[\theta[t]]\dot{\phi}^2[t])))\end{matrix}}{lM(-6m-8M+2mcos[2\theta[t]]+mcos[2(\theta[t]-\phi[t])]+2mcos[2\phi[t]]+mcos[2(\theta[t]+\phi[t])])}$$

$$\ddot{\phi}[t] = \frac{\begin{matrix}sec[\theta[t]](-4msin[2\theta[t]]sin[2\phi[t]]F_x[t]-8(-m-2M+mcos[2\theta[t]])cos[\phi[t]]F_y[t]\\ +M(8lmcos[\theta[t]]sin[2\phi[t]]\dot{\theta}^2[t]+4l(-6m-8M+2mcos[2\theta[t]]\\ +mcos[2(\theta[t]-\phi[t])]+2mcos[2\phi[t]]+mcos[2(\theta[t]+\phi[t])])sin[\theta[t]]\dot{\theta}[t]\dot{\phi}[t]\\ -16sin[\phi[t]](g(m+M)-lmcos[\theta[t]]^3cos[\phi[t]]\dot{\phi}^2[t])))\end{matrix}}{2lM(-6m-8M+2mCos[2\theta[t]]+mCos[2(\theta[t]-\phi[t])]+2mCos[2\phi[t]]+mCos[2(\theta[t]+\phi[t])])}$$

(28)

Converting ẍ[t], ÿ[t], θ̈[t], and $\ddot{\phi}$[t] in form of state space.

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\\ \dot{y}\\ \ddot{y}\\ \dot{\theta}\\ \ddot{\theta}\\ \dot{\phi}\\ \ddot{\phi}\end{bmatrix} = \begin{bmatrix}0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&-\frac{gm}{M}&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&-\frac{gm}{M}&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&\frac{g(m+M)}{lM}&0&0&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&\frac{g(m+M)}{lM}&0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ y\\ \dot{y}\\ \theta\\ \dot{\theta}\\ \phi\\ \dot{\phi}\end{bmatrix} + \begin{bmatrix}0&0\\ \frac{1}{M}&0\\ 0&0\\ 0&\frac{1}{M}\\ 0&0\\ -\frac{1}{lM}&0\\ 0&0\\ 0&-\frac{1}{lM}\end{bmatrix}\begin{bmatrix}F_x\\ F_y\end{bmatrix}$$

$$y = \begin{bmatrix}1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ y\\ \dot{y}\\ \theta\\ \dot{\theta}\\ \phi\\ \dot{\phi}\end{bmatrix}$$

(29)

The state variables in Equation 29 are

$x_1 = x$: X-Axis Direction Displacement

$x_2 = \dot{x}$: X-Axis Direction Velocity

$x_3 = y$: Y-Axis Direction Displacement

$x_4 = \dot{y}$: Y-Axis Direction Velocity

$x_5 = \theta$: X-Axis Direction Tilt angle

$x_6 = \dot{\theta}$: X-Axis Direction Angle velocity

$x_7 = \phi$: Y-Axis Angle velocity

$x_8 = \dot{\phi}$: Y-Axis Angle velocity

4.2.2 Actual Model State Space

Deriving the dynamic model.

Figure 4.11

Platform Movement of Actual Model

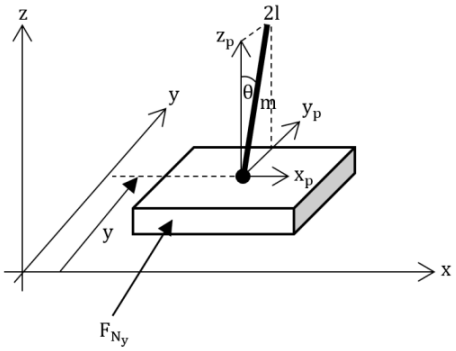
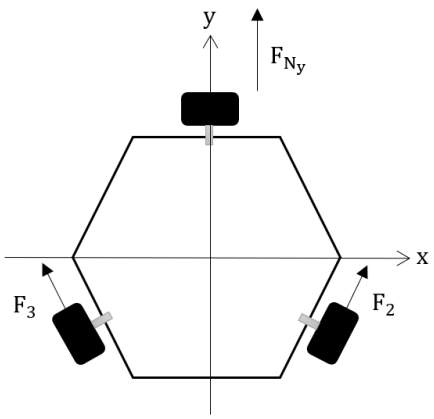


Figure 4.12

One-Axis Inverted Pendulum with External Forces



Converting the Equation 10, Equation 16, and Equation 21 to state space form.

$$\ddot{y}[t] = \frac{K_t(-V_{s_2}[t] + V_{s_3}[t] + K_e(\dot{\omega}_2[t] - \dot{\omega}_3[t])) + lmrR_a(sin[\theta[t]]\dot{\theta}^2[t] - cos[\theta[t]]\ddot{\theta}[t])}{(m+M)rR_a}$$

$$\ddot{\theta}[t] = \frac{gsin[\theta[t]] - cos[\theta[t]]\ddot{y}[t]}{l}$$

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{gm}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{g(-m-M)}{lm} & 0 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\frac{K_t}{MrR_a} & \frac{K_t}{MrR_a} \\ 0 & 0 \\ \frac{K_t}{lMrR_a} & -\frac{K_t}{lMrR_a} \end{bmatrix} \begin{bmatrix} V_{s_2} \\ V_{s_3} \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} \quad (30)$$

The state variables in Equation 30 are

$x_1 = y$: Y-Axis Direction Displacement

$x_2 = \dot{y}$: Y-Axis Direction Velocity

$x_3 = \theta$: Y-Axis Direction Tilt angle

$x_4 = \dot{\theta}$: Y-Axis Direction Angular Velocity

4.3 Linear Quadratic Regulator

In this study, Linear Quadratic Regulator is applied as the controller to balance the inverted pendulum. This controller gives the optimal gain for the system. The optimal gain (K) that is suited to this thesis is calculated by the LQR function in MATLAB with a selected value of Q and R. The state-feedback law is as follows.

$$u = -Kx \quad (31)$$

While K must minimize the following quadratic cost function.

$$J(u) = \int_0^t (x^T Q x + u^T R u) dt \quad (32)$$

The cost function subjects to the dynamic system which are

$$\dot{x} = Ax + Bu \tag{33}$$

K can be derived from

$$K = R^{-1}(B^T S) \tag{34}$$

While the solution (S) comes from the associated Riccati equation which is as follows.

$$A^T S + SA - SBR^{-1}B^T S + Q = 0 \tag{35}$$

The syntax in the MATLAB is as follows.

$$[K, S, E] = lqr(A, B, Q, R) \tag{36}$$

K : Optimal Gain

S : Solution of Riccati Equation

E : Eigenvalues of Closed Loop (matrix(A − B ∗ K) eigenvalues)

A : Dynamic System State Matrix

B : Matrix of Input

Q : Matrix of State Weighting

R : Matrix of Control Weighting

CHAPTER 5
RESULT AND DISCUSSION

The calculation for the gain and the results of the systems when applying gain (K) of both Simulation and Actual Models are shown.

5.1 Simulation Model

5.1.1 Obtaining the Gain K of Simulation Model by LQR

From the state space representation of Simulation Model in Equation 29, Table 5.1 shows parameters for State Space of Simulation Model.

Table 5.1

Simulation Model State Space Parameters

Constant Parameter	Value	Units
g	9.81	m/s^2
m	18.4×10^{-3}	kg
M	18.9	kg
l	79.3×10^{-2}	m

Substituting values from Table 5.1 into the state space equation

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0095 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.0095 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 12.3702 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12.3702 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0.0529 & 0 \\ 0 & 0 \\ 0 & 0.0529 \\ 0 & 0 \\ -0.0667 & 0 \\ 0 & 0 \\ 0 & -0.0667 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{bmatrix} \quad (37)$$

To calculate the LQR function, Q and R matrices are required. Both matrices are as follows.

$$Q = \begin{bmatrix} 0.00001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.00001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.00001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.00001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 250000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100000000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 250000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100000000 \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (38)$$

Calculating the LQR function to the optimal gain (K) as follows.

$$K = \begin{bmatrix} 0 & -1 & 0 & 0 & -826 & -10002 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -826 & -10002 \end{bmatrix} \quad (39)$$

Therefore, the input value (u) is denoted as $u = -Kx$ which K is from Equation 39 and x is the state variables that mentioned in Section 4.2.2.

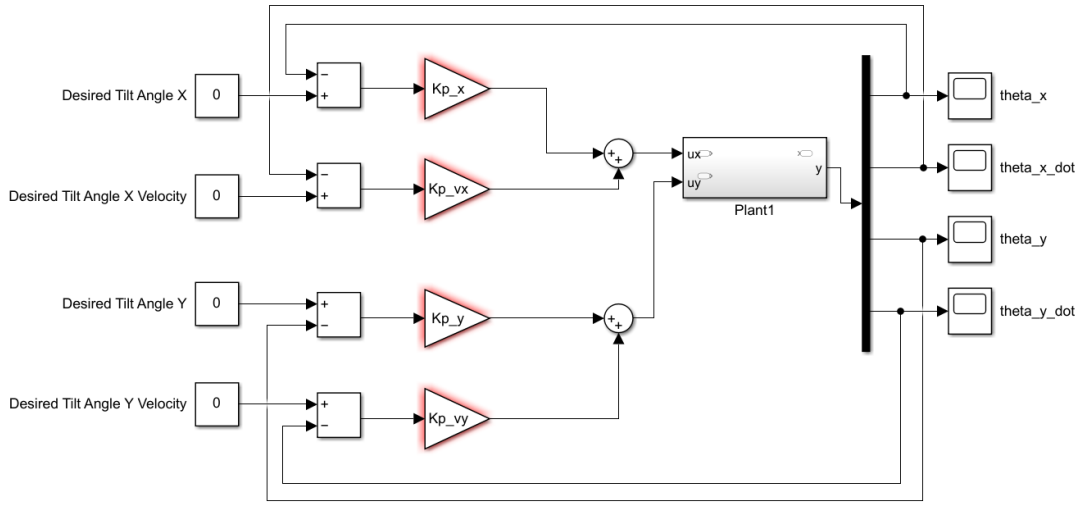
5.1.2 Result of Simulation Model

Two controller algorithms are used to indicate the result of the Simulation Model.

5.1.2.1 **PD-Controller.** The gains of PD-controller for Simulation Model are $[K_p, K_d] = [5, -100]$ while K_p is the gain for tilt angle of the pendulum rod and K_d is the gain for angular velocity. Figure 5.1 shows that these two gains are used in both x and y axes.

Figure 5.1

Block Diagram of PD-Controller System for Simulation Model



The results are as follows.

Figure 5.2

Tilt Angle of the Pendulum Rod in the X Direction of Simulation Model when Applying the PD-Controller

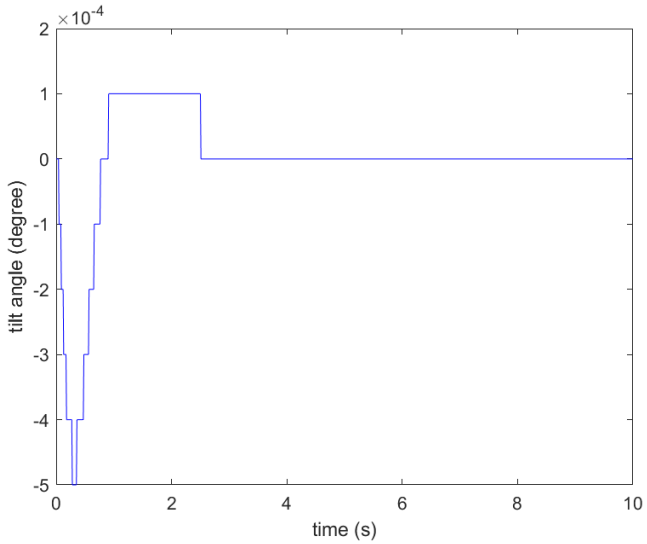


Figure 5.3

Angular Velocity of the Pendulum Rod in the X Direction of Simulation Model when Applying the PD-Controller

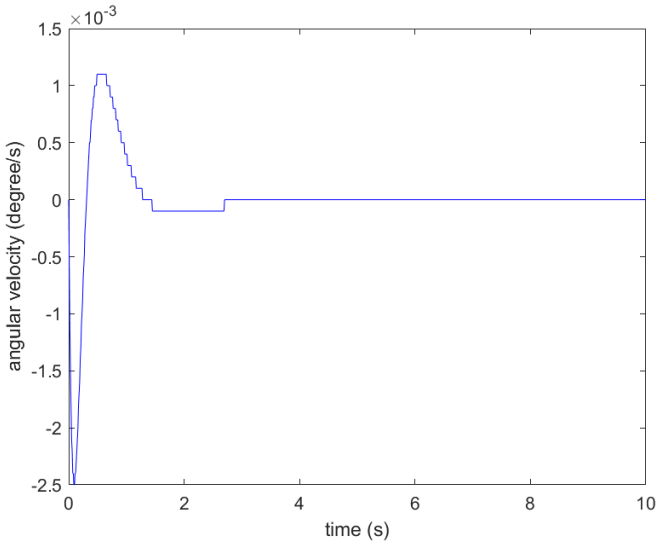


Figure 5.4

Tilt Angle of the Pendulum Rod in the Y Direction of Simulation Model Controller when Applying the PD-Controller

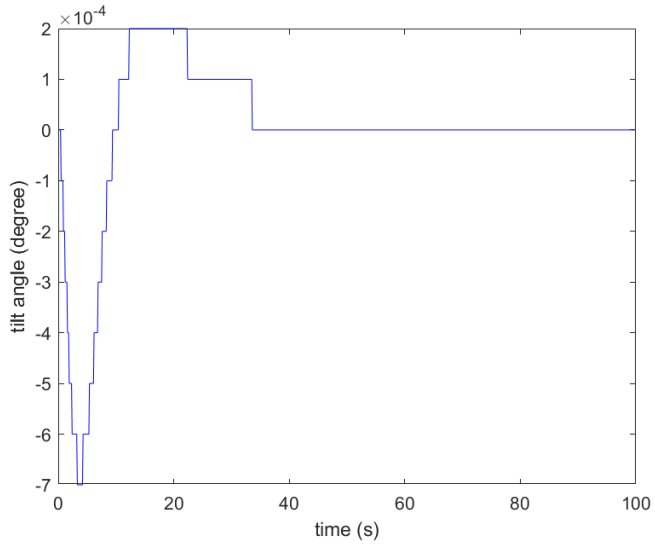


Figure 5.5

Angular Velocity of the Pendulum Rod in the Y Direction of Simulation Model when Applying the PD-Controller

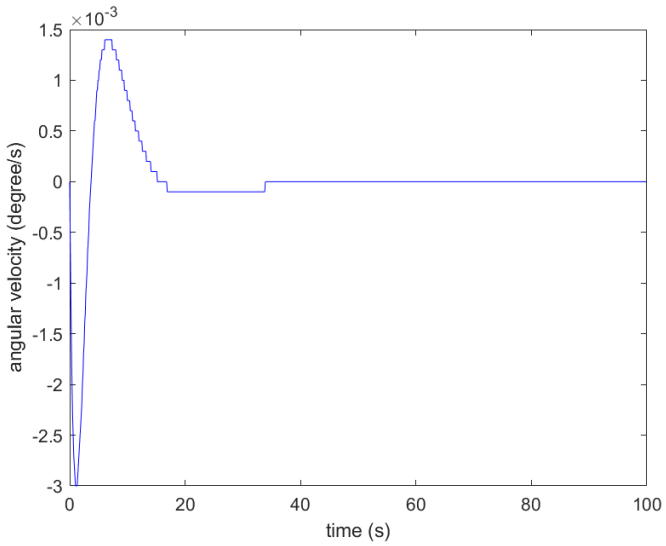


Figure 5.6

Distance of the Mobile Robot in the X Direction of Simulation Model when Applying the PD-Controller

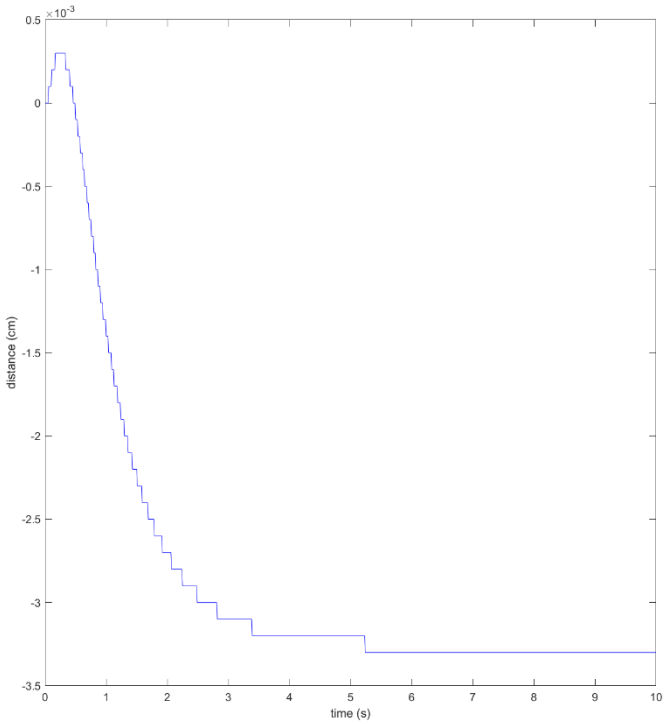


Figure 5.7

Velocity of the Mobile Robot in the X Direction of Simulation Model when Applying the PD-Controller

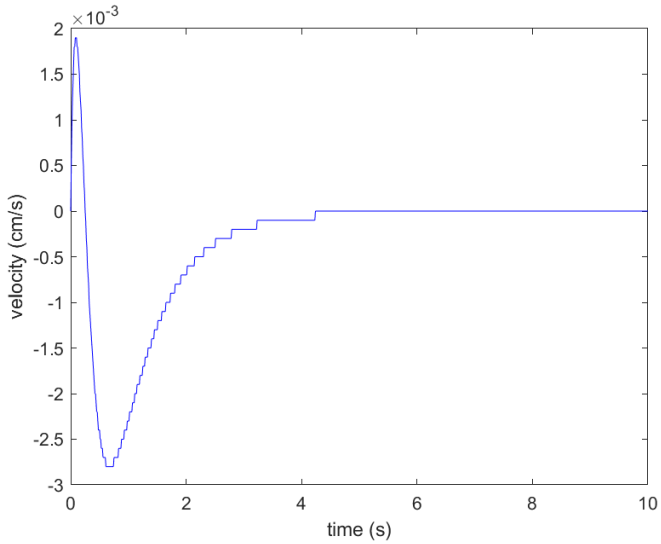


Figure 5.8

Distance of the Mobile Robot in the Y Direction of Simulation Model when Applying the PD-Controller

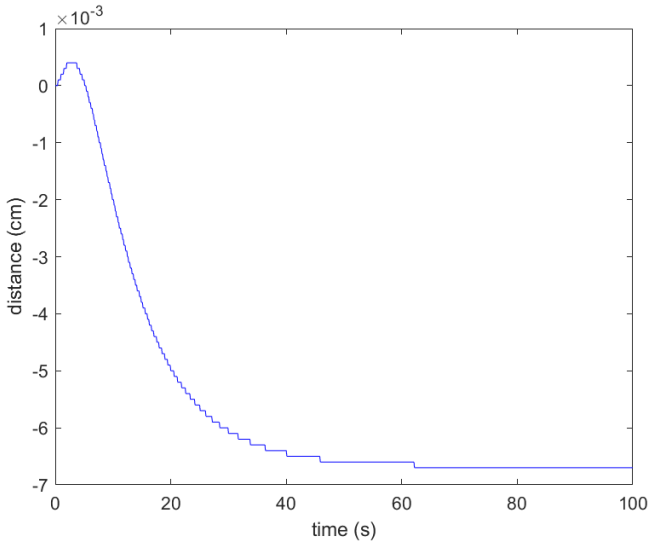
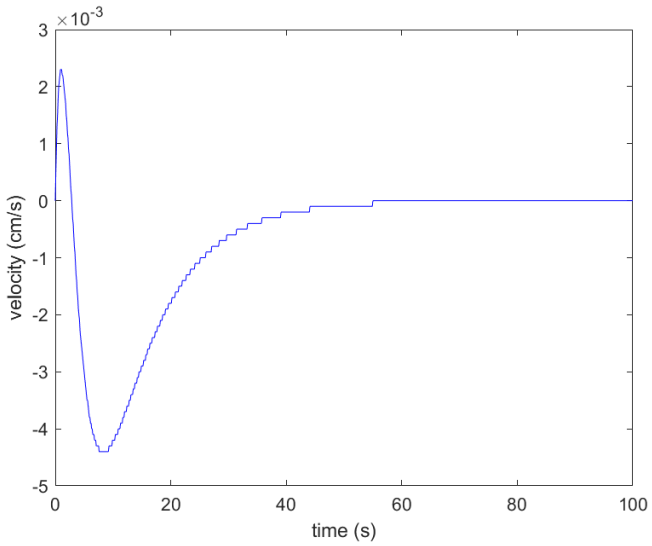


Figure 5.9

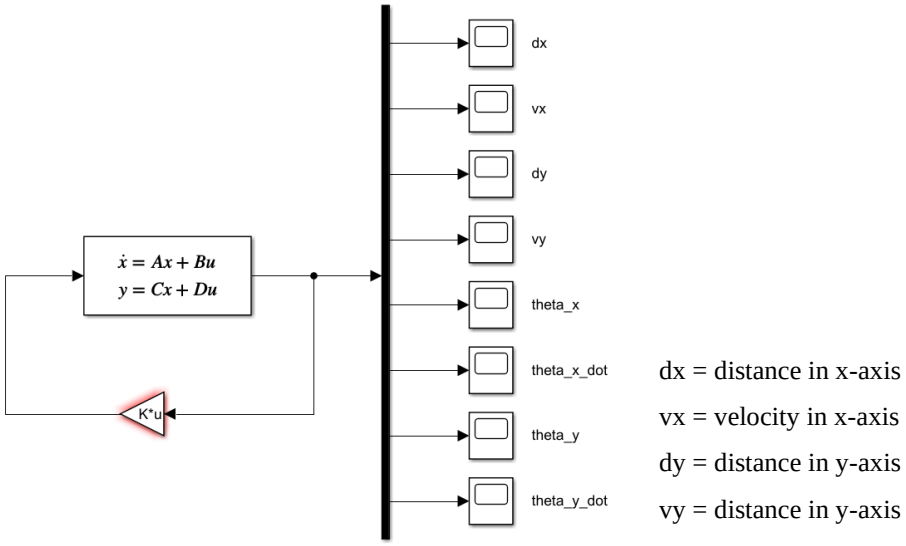
Velocity of the Mobile Robot in the Y Direction of Simulation Model when Applying the PD-Controller



5.1.2.2 **LQR Controller.** Using K from Equation 39 to find the input (u). Applying this input to the Simulation Model in both x and y axis to control the movement of the platform.

Figure 5.10

Block Diagram of LQR Controller System for Simulation Model



The results are as follows.

Figure 5.11

Tilt Angle of the Pendulum Rod in the X Direction of Simulation Model when Applying the LQR Controller

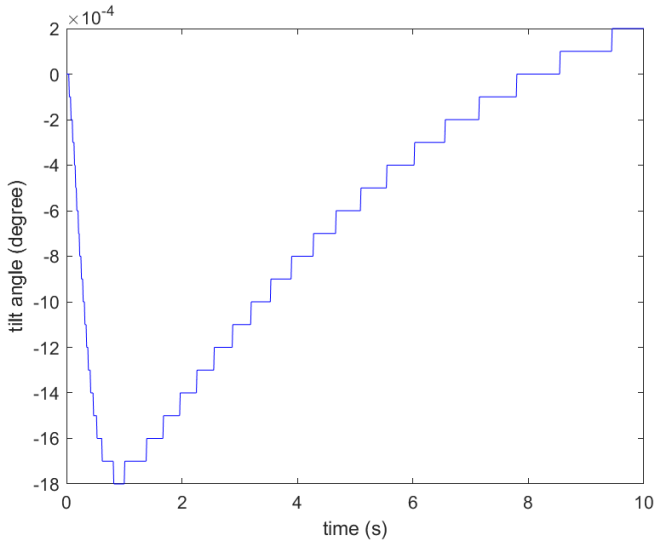


Figure 5.12

Angular Velocity of the Pendulum Rod in the X Direction of Simulation Model when Applying the LQR Controller

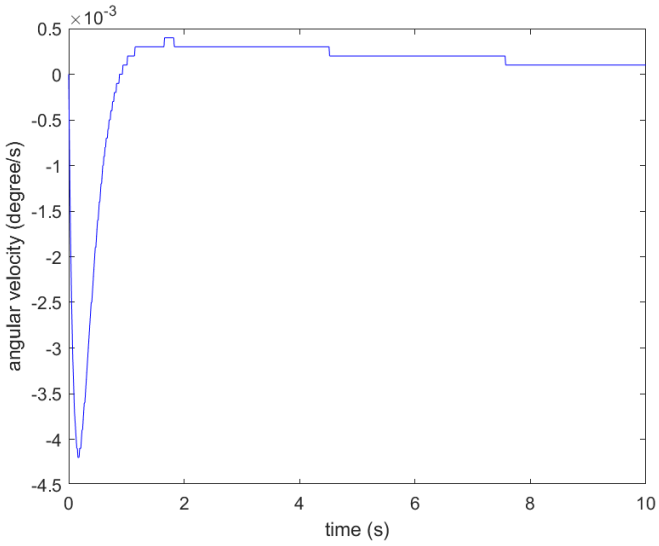


Figure 5.13

Tilt Angle of the Pendulum Rod in the Y Direction of Simulation Model when Applying the LQR Controller

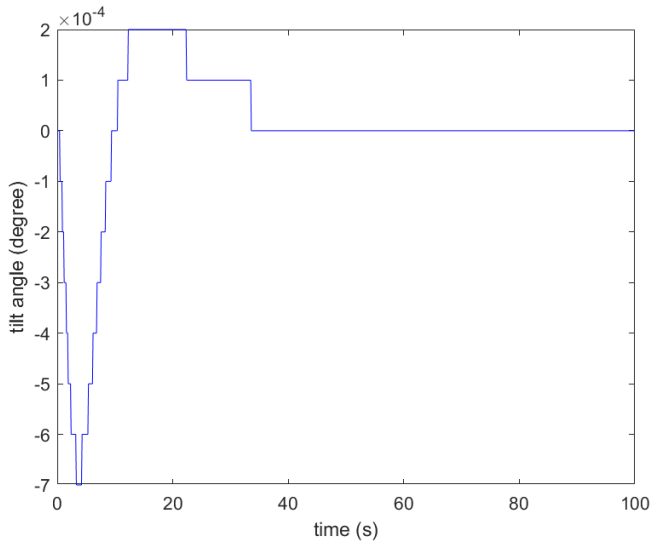


Figure 5.14

Angular Velocity of the Pendulum Rod in the Y Direction of Simulation Model when Applying LQR Controller

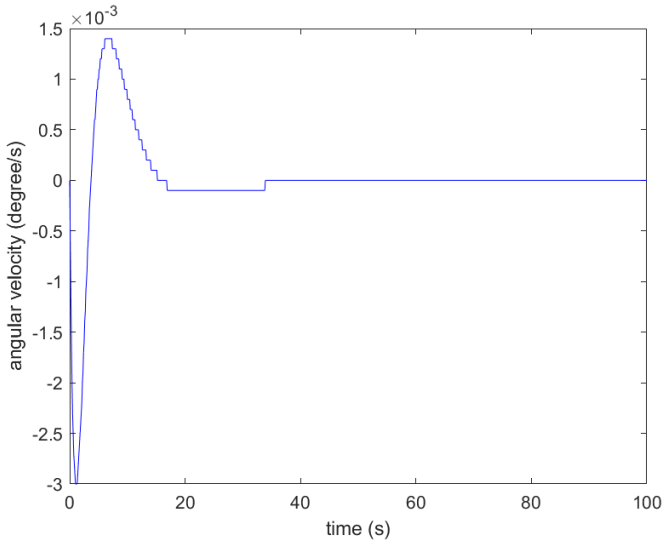


Figure 5.15

Distance of the Mobile Robot in the X Direction of Simulation Model when Applying the LQR Controller

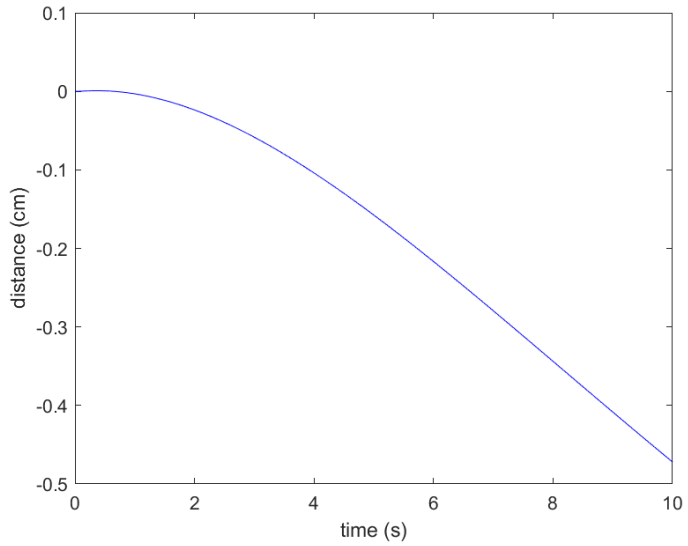


Figure 5.16

Velocity of the Mobile Robot in the X Direction of Simulation Model when Applying the LQR Controller

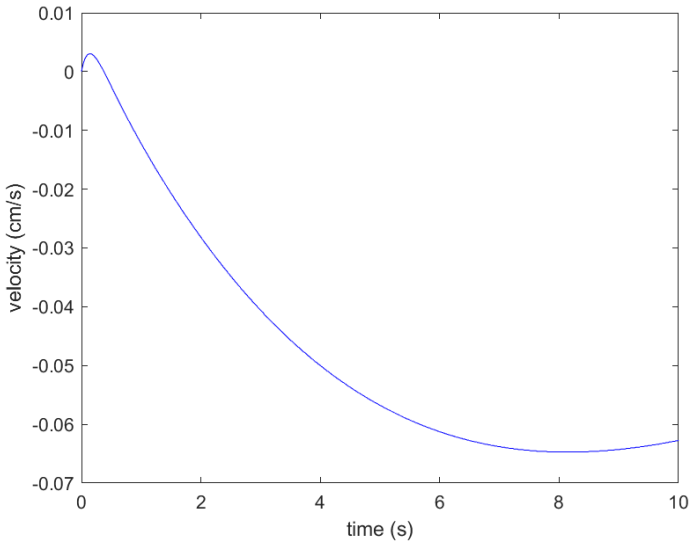


Figure 5.17

Distance of the Mobile Robot in the Y Direction of Simulation Model when Applying the LQR Controller

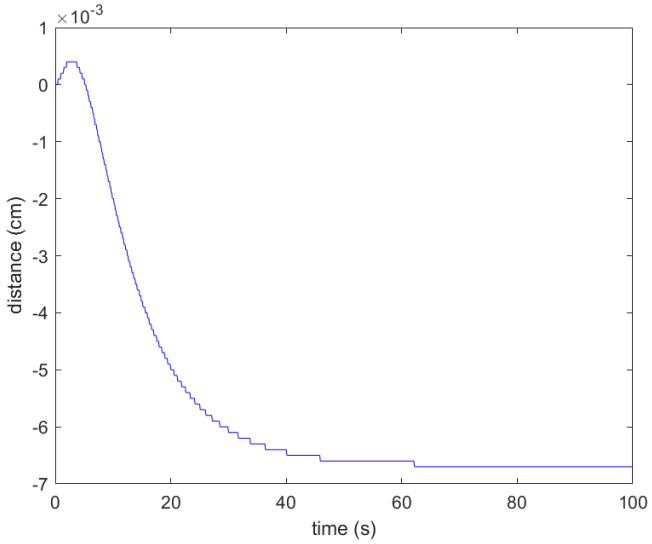
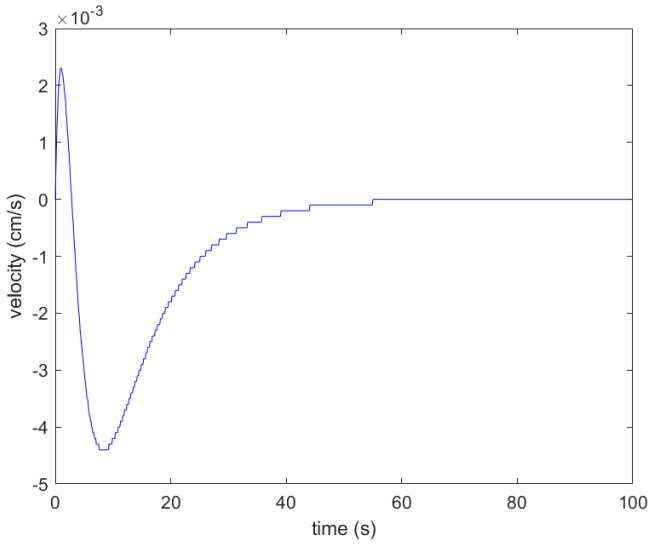


Figure 5.18

Velocity of the Mobile Robot in the Y Direction of Simulation Model when Applying the LQR Controller



5.2 Actual Model

5.2.1 Obtaining the Gain K of Actual Model by LQR

The optimal gain (K) can be calculated from the Actual Model state space representation in Equation 30. The parameters in this state space will be defined in Table 5.2.

Table 5.2

Actual Model State Space Parameters

Constant Parameter	Value	Units
g	9.81	m/s^2
m	1.84×10^{-2}	kg
M	18.9	kg
l	79.3×10^{-2}	m
K_t	5.94×10^{-2}	-
r	6.35×10^{-2}	m
R_a	1.283	Ω

Substituting values from Table 5.2 into the state space equation

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{gm}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{g(-m-M)}{lm} & 0 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\frac{K_t}{MrR_a} & \frac{K_t}{MrR_a} \\ 0 & 0 \\ \frac{K_t}{lMrR_a} & -\frac{K_t}{lMrR_a} \end{bmatrix} \begin{bmatrix} V_{s_2} \\ V_{s_3} \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{40}$$

Q and R matrices are as follows.

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 20000 & 0 \\ 0 & 0 & 0 & 570000000 \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (41)$$

Calculating the LQR function to the optimal gain (K) as follows.

$$K = \begin{bmatrix} 1 & 19 & 2690 & 17337 \\ -1 & -19 & -2690 & -17337 \end{bmatrix} \quad (42)$$

K in Equation 42 is used to calculate the input of the Actual model system.

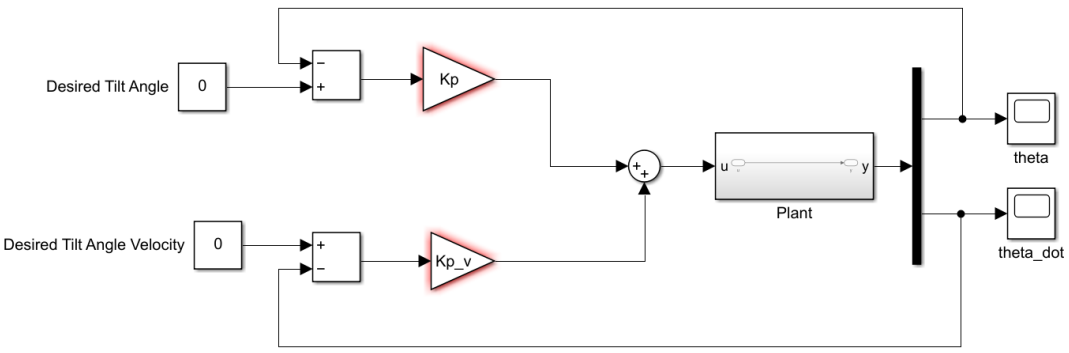
5.2.2 Result of Actual Model

Two controller algorithms are used to indicate the result of the Actual Model.

5.2.2.1 **PD-Controller.** The gains of PD-controller for Actual Model are [K_p, K_d] = [300, -2800] while K_p is the gain for tilt angle of the pendulum rod and is K_d the gain for angular velocity. These two gains are used to find the input for the Actual Model as shown in Figure 5.1.

Figure 5.19

Block Diagram of PD-Controller System for Actual Model



The results are as follows.

Figure 5.20

Tilt Angle of the Pendulum Rod of Actual Model when Applying the PD-Controller

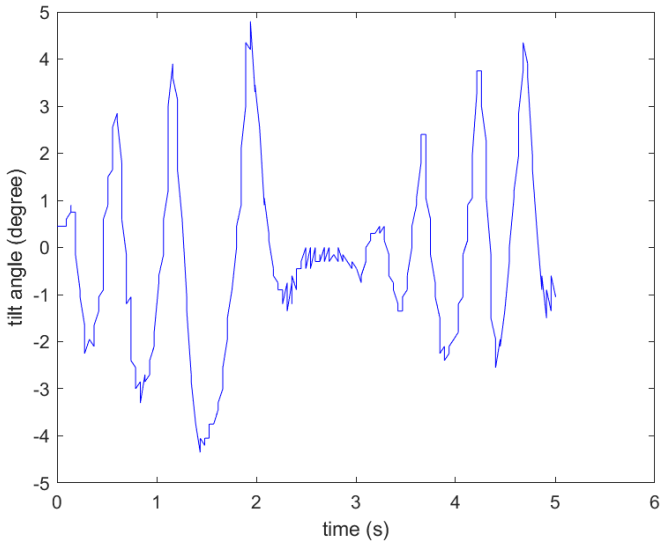


Figure 5.21

Angular Velocity of the Pendulum Rod of Actual Model when Applying PD-Controller

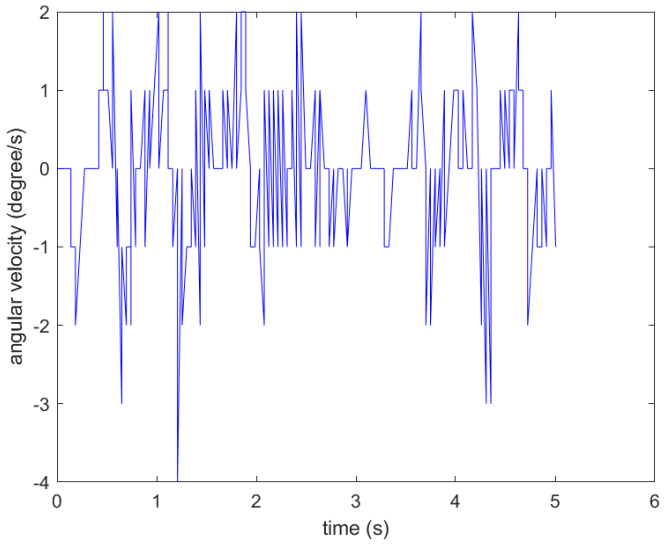
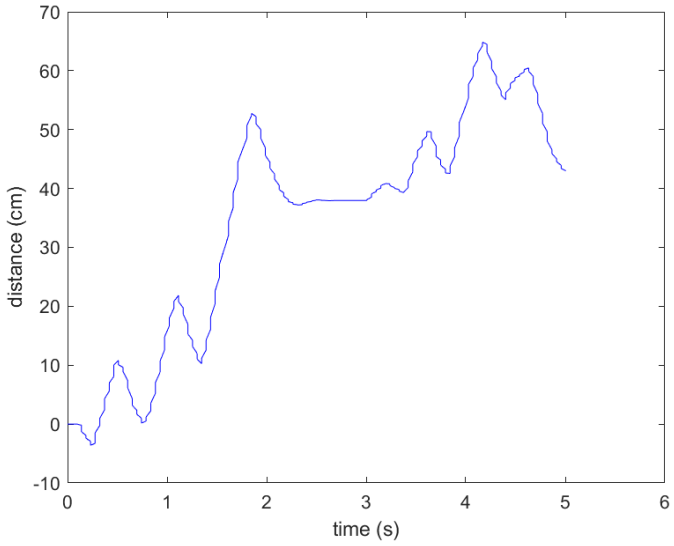
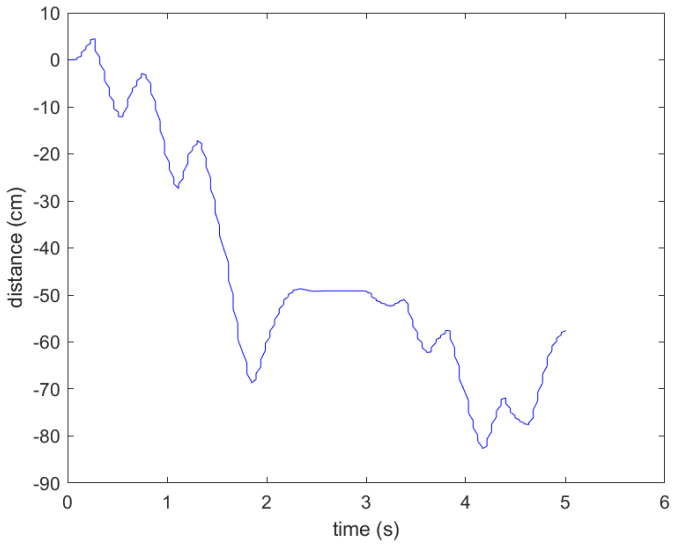


Figure 5.22

Distance of the Mobile Robot of Actual Model when Applying the PD-Controller (a) Wheel 2 (b) Wheel 3



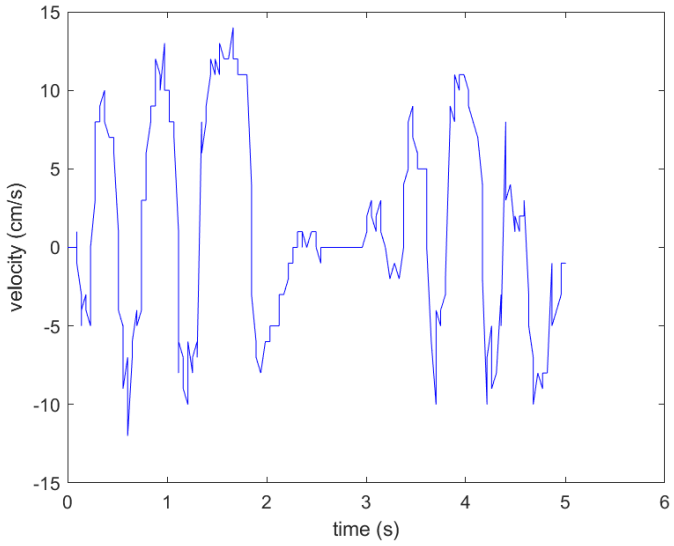
(a)



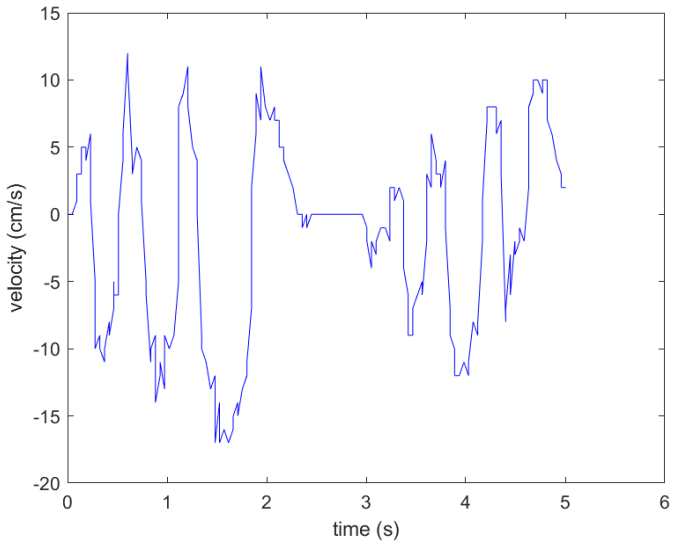
(b)

Figure 5.23

Velocity of the Mobile robot of Actual Model when Applying the PD-Controller (a) Wheel 2 (b) Wheel 3



(a)

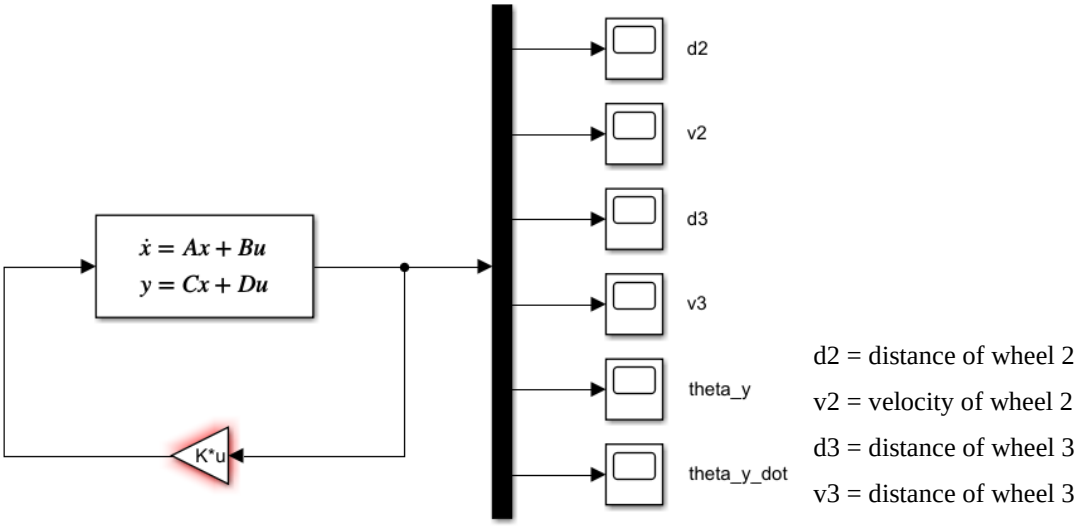


(b)

5.2.2.2 **LQR Controller.** Using K from Equation 42 to find the input (u) Applying this input to the Actual Model to control the movement of the platform.

Figure 5.24

Block Diagram of LQR Controller System for Actual Model



The results are as follows.

Figure 5.25

Tilt Angle of the Pendulum Rod of Actual Model when Applying the LQR Controller

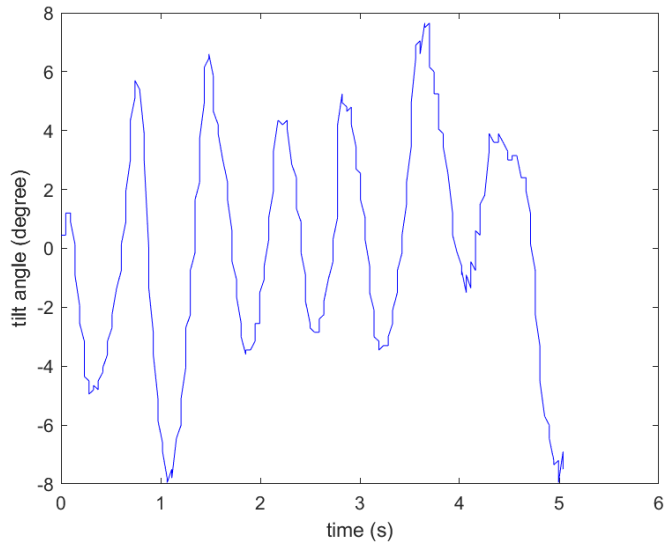


Figure 5.26

Angular Velocity of the Pendulum Rod of Actual Model when Applying the LQR Controller

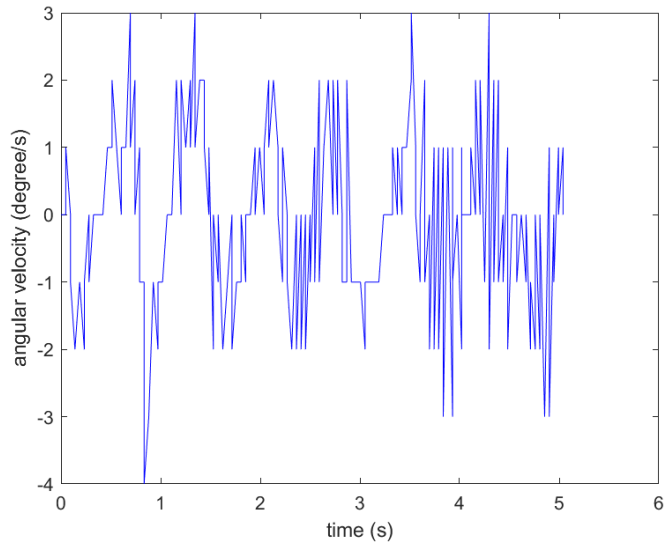
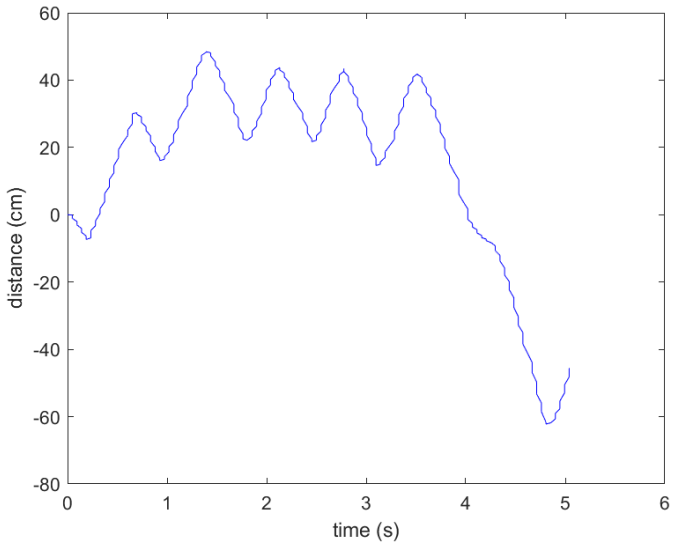
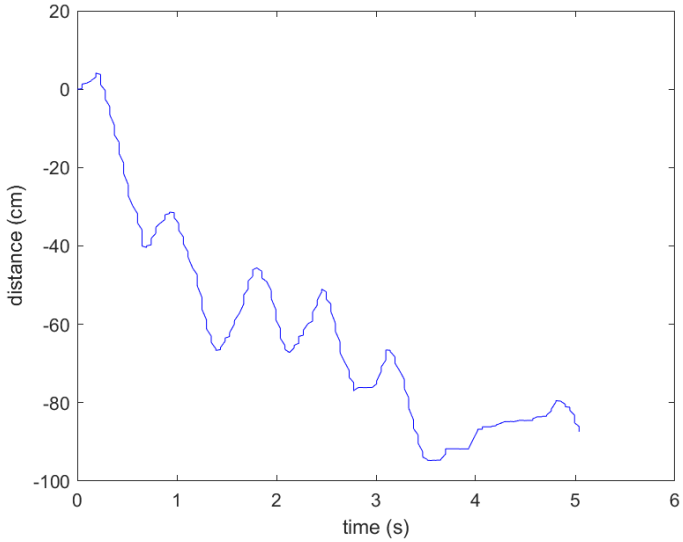


Figure 5.27

Distance of the Mobile Robot of Actual Model when Applying the LQR Controller (a) Wheel 2 (b) Wheel 3



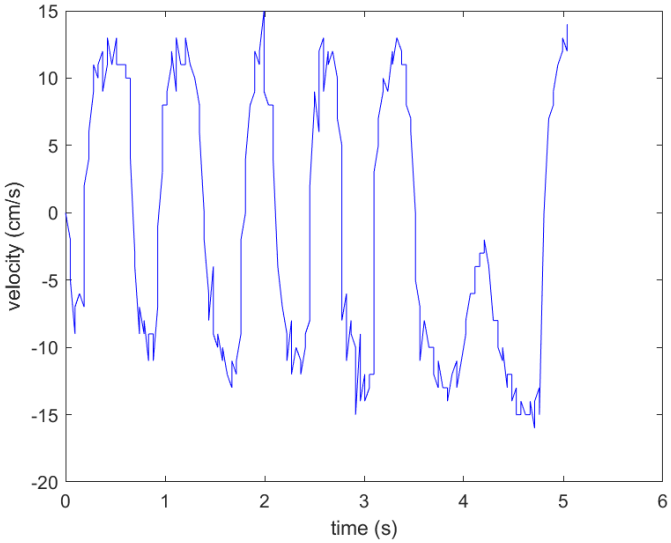
(a)



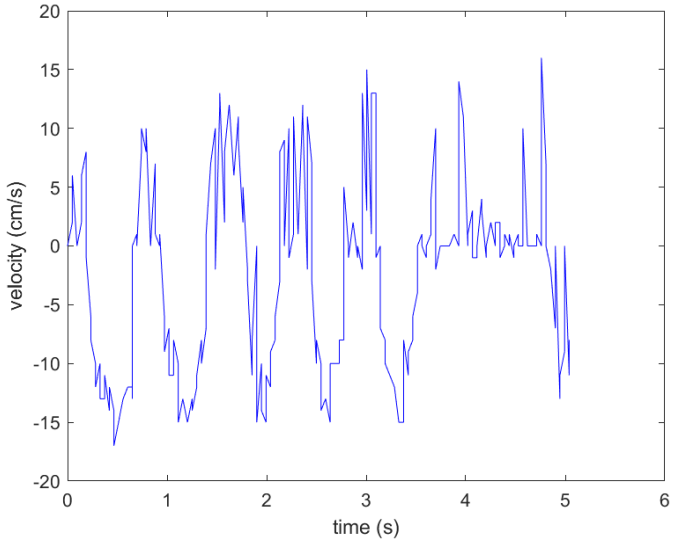
(b)

Figure 5.28

Velocity of the Mobile robot of Actual Model when Applying the LQR Controller (a) Wheel 2 (b) Wheel 3



(a)



(b)

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, a three-omnidirectional-wheel mobile robot was designed and developed to balance an inverted pendulum. The mechanical and electrical designs of the robot were conducted. The system dynamics model was derived. Two control algorithms were designed to control and balance the inverted pendulum; PD and LQR. The inverted pendulum was balanced along both x and y axes in the simulation. However, the inverted pendulum was balanced along only y axis in the experiment. The simulation results showed that the inverted pendulum was balanced successfully. However, the inverted pendulum could not be balanced successfully in the experiment. The major reasons are the noises in the sensor and the gear transmission ratio of the selected motors cannot move and rotate to the desired position and speed.

6.2 Future Works

Because the Actual Model cannot balance the spherical inverted pendulum, finding the proper sensor that can get the data of both tilt angles in the x- and y-axis without noise or with less noise is required. Another drawback of this thesis is that the weight of the platform is too much. The reason is that to balance the inverted pendulum; the robot must move speedy and stop instantaneously. The heavy mass causes great inertia and leads the robot hard to speed up to reach the desired velocity or stop in the desired position. Also, the heavy mass creates the shaking behavior of the robot; this behavior affects the tilt angle of the pendulum rod, which is hard to reach the upright position. Therefore, reducing the platform weight is required for future works.

REFERENCES

S. Kao, W. Chiou and M. Ho, "Balancing of a spherical inverted pendulum with an omni-directional mobile robot," 2013 IEEE International Conference on Control Applications (CCA), Hyderabad, 2013, pp. 760-765, doi: 10.1109/CCA.2013.6662841.

S. Kao and M. Ho, "Tracking control of a spherical inverted pendulum with an omnidirectional mobile robot," 2017 International Conference on Advanced Robotics and Intelligent Systems (ARIS), Taipei, 2017, pp. 52-57, doi: 10.1109/ARIS.2017.8297182.

Villacres J, Viscaino M, Herrera M, Camacho O., “Controllers comparison to stabilize a two-wheeled inverted pendulum: PID, LQR and sliding mode control,” International Journal of Control Systems and Robotics, 2016, pp.29-36.

H. F. Murcia and A. E. González, "Performance comparison between PID and LQR control on a 2-wheel inverted pendulum robot," 2016 IEEE Colombian Conference on Robotics and Automation (CCRA), Bogota, 2016, pp. 1-6, doi: 10.1109/CCRA.2016.7811420.

Nasir AN, Ismail RM, Ahmad MA., “Performance comparison between sliding mode control (SMC) and PD-PID controllers for a nonlinear inverted pendulum system,” World Academy of Science, Engineering and Technology, 2010, 71, pp. 400-5.

M. I. H. Nour, J. Ooi and K. Y. Chan, "Fuzzy logic control vs. conventional PID control of an inverted pendulum robot," 2007 International Conference on Intelligent and Advanced Systems, Kuala Lumpur, 2007, pp. 209-214, doi: 10.1109/ICIAS.2007.4658376.

Bature AA, Buyamin S, Ahmad MN, Muhammad M., “A comparison of controllers for balancing two-wheel inverted pendulum robot,” International Journal of Mechanical & Mechatronics Engineering, 2014, 14(3), pp. 62-68.

Olfati-Saber, Reza. "Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles." PhD diss., Massachusetts Institute of Technology, 2001.

Viet, Tuan Dinh, Phuc Thinh Doan, Hoang Giang, Hak Kyeong Kim, and Sang Bong Kim. "Control of a 2-DOF omnidirectional mobile inverted pendulum." Journal of mechanical science and technology 26, no. 9 (2012): 2921-2928.

Linear-Quadratic Regulator (LQR) design, https://www.mathworks.com/help/control/ref/lqr.html