DEVELOPMENT OF A SEGWAY-TYPE TURNABLE-SEAT WHEELCHAIR PROTOTYPE

by

Jirapod Jintasornrom

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. Pisut Koomsap
Dr. Mongkol Ekpanyapong

Nationality: Thai
Previous Degree: Bachelor of Engineering in Mechatronics Engineering
Assumption University
Thailand

Scholarship Donor: Royal Thai Government Scholarships

Asian Institute of Technology
School of Engineering and Technology
Thailand
May 2021

AUTHOR'S DECLARATION

I, Jirapod Jintasornrom 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 are 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: 10 May 2021

Name: Jirapod Jintasornrom

Signature:

ACKNOWLEDGMENTS

Firstly, I sincerely thank Prof. Manukid Parnichkun for being my advisor for this research. The knowledge and advice he has given me are inexpressible. With his guidance, this research has given me so much valuable experience and expertise.

I want to truthfully thank Dr.Pisut Koomsub and Dr. Mongkol Ekpanyapong for their valuable time as my examination committee. Their suggestions give me a different perspective of the project and make my research better.

I would like to show gratitude to Mr. Thanit Pattana and Mr. Hoang Hung Manh for helping me complete my research successfully.

Finally, I would like to thank my parent for giving me an opportunity to study at AIT.

ABSTRACT

In this research, the control system design for a segway-type turnable prototype wheelchair has been proposed. This vehicle has a turnable wheelchair and two wheels on each side. The user can sit on the chair and have a hand controller for position control. The wheelchair can move in four directions which are forward, backward, turn left, and turn right. Also, the wheelchair is balanced by itself while moving in any direction. The chair rotation of the wheelchair is seen as a disturbance and should be handled by the system. The system uses electrical energy to drive a DC motor, and due to high current consumption, a Li-ion battery is used as a power supply.

The control system is designed according to the optimal control algorithm. The controller develops with the same concept as a two-wheel self-balancing robot. This controller has been designed to balance while the top part of the robot rotates. Additionally, the controller can control the system in both translational and rotational motion.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figures	**Page**

CHAPTER 1
INTRODUCTION

1.1 Background of the Study

The wheelchair is one of the most ordinary pieces of equipment that give mobility and comfort to those people who have difficulty walking. This device grants opportunities for users to study, work and join society again. Moreover, appropriately use a wheelchair benefits the physical health and quality of life for the elderly by lowering usual problems such as osteoarthritis, pressure sore, or accident. So, for many older people or people with difficulty walking, a wheelchair is like another part of their body.

Nowadays, vehicle technology improves drastically with the innovation of mechanical and electrical systems. These technologies then use to improve the quality of the wheelchair. Many wheelchairs have developed to satisfy the need for different situations. However, there are only two mains category which is manual wheelchair and power wheelchair.

A manual wheelchair is a wheelchair that commonly uses in society. This wheelchair has not required any maintenance because it does not have any electrical system inside. A manual wheelchair can move by a user rolling its wheel or other push for them.

Figure 1.1

Manual Type Wheelchair

Even though a manual wheelchair is excellent for those who can move them independently, the drawback is that users can quickly be exhausted, especially when traveling in long-distance. Furthermore, using a manual wheelchair alone in the slope area can be very difficult due to gravity. However, with improvement on an electrical motor and energy, these problems are solved.

A power wheelchair is a wheelchair that drives by electrical energy. It has an electrical battery to keep energy and rotate the wheel by an electrical motor. This wheelchair did not require any external power from the user, which can benefit those who independently operate the wheelchair. Besides, users can control both the speed and direction of the wheelchair, which gives them the ability to move more freely.

Figure 1.2

Power Type Wheelchair

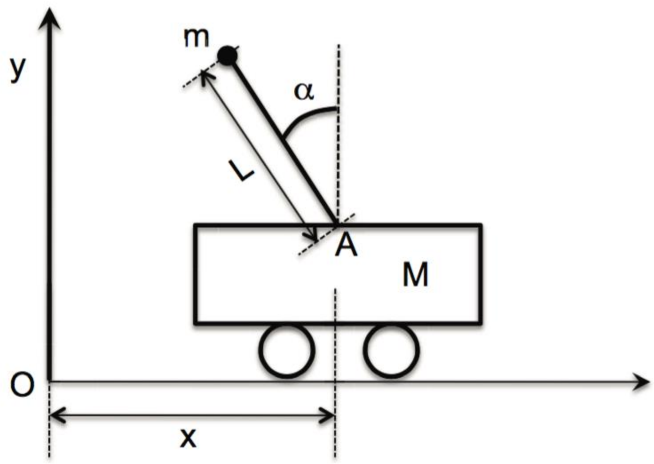
The power wheelchair gives the user the ability to have a comfortable life and a chance to join society confidently again. Nonetheless, a power wheelchair still has plenty of limitations. For example, a stair is one of the most challenging problems that many researchers aim to solve. The problem with stair climbing is that the wheelchair is falling because it is losing balance. However, with the control theory, these obstacles can be resolved.

Control theory is a part of Applied Mathematics dealing with feedback control to determine a system's behavior to accomplish the desired goal. During the modern ages, control systems have become a main role in expanding and improving modern technology and civilization. Moreover, it has been implemented in many devices to achieve the highest efficiency of the system.

A balancing system is an excellent example of a modern control system. It is a method to stabilize the unstable system with the use of control theory. One of the pervasive unstable systems is the inverted pendulum. Since it is unstable without an input force, the pendulum will naturally fall over if there is no system to balance it. In addition, the dynamics of the inverted pendulum are non-linear. So, a control system aims to stabilize the inverted pendulum by giving energy to the body where the pendulum is connected.

Figure 1.3

Inverted Pendulum



The concept of the inverted pendulum has been used as the foundation to further develop the balancing robot. One of them is a two-wheel balancing vehicle. It is a device that uses for human transportation. With the help of a self-balancing system, it can move with just two wheels. This vehicle is commonly known as Segway.

The Segway is made to stay balanced with two wheels, designed to copy human walking motion. It uses the concept of change in the center of gravity to move the vehicle to move to the desire position.

The proposed Segway-type turnable wheelchair use two-wheel for translation and rotation. Due to the nature of the wheelchair, the user mass is acting as a rigid body. So, the design of the control algorithm is similar to a two-wheel robot. The designed control algorithm needs the ability to stabilize when it has chair rotation. This rotation is acting as a disturbance from the deviation of the center of mass.

Figure 1.4

Segway-type Turnable Wheelchair

1.2 Statement of the Problem

The segway-type turnable wheelchair is a challenging task for the balancing system. Unlike Segway, the shifted on the center of mass cannot be used to control the direction of the robot. The wheelchair requires stability to the point that it can move a user to the desire location with its balancing system.

Many two-wheel wheelchairs have a problem balancing while moving because of the high weight, it requires power to balance the system. The use of PID may not have given the optimal result for balancing the system. Moreover, turning left and right of the system need to move the wheel, which not suitable for some situation. This research will mainly use optimal control to design the control system. The designed system must be able to move to the desired position. Additionally, the robot can rotate in the desire direction without moving the wheel.

1.3 Objectives of the Research

The main objective is to develop a segway-type turnable-seat wheelchair prototype. In order to achieve the main objective, the following tasks are required.

1. Designing and constructing a two-wheel turnable wheelchair prototype.
2. Design control algorithm for the system.
3. Wheelchair can turn without moving the wheel.
4. Can balance with any rotating angle of upper part.

1.4 Limitations and Scope

1. The maximum speed of the vehicle will be limited to 5.4 km/hr.
2. Wheelchairs are balanced and move without the user load.
3. Wheelchair can turn without moving the wheel.

CHAPTER 2
LITERATURE REVIEW

Stability and accuracy are essential for almost every system in the world. Technology grows to the point where the missile can accurately hit the target across the planet. A robot can move with one wheel touch the ground. All of this can happen only with the help of physic and mathematic. However, the most challenging part is how to control this knowledge to achieve the desired goal. Physic and mathematic can describe a variable in number but to control those variables is the job of the control system.

Control systems mainly use for controlling variables to the desired value. It uses to manages, controls, addresses, or regulates the behavior of the device. It is covering from simple motor speed control up to the position of the satellite. The control system is not limited to just an electrical system. Any system that can describe by physic and mathematic can use control theory to optimize the system.

Balancing vehicles is a prevalent concept for control theory because of its instability nature. Segway is the perfect example. It is a vehicle that can move on a two-wheel. With the help of a self-balancing system, the user can steadily stand on the platform. These vehicles have many benefits above the usual four-wheel such as turning radius and size.

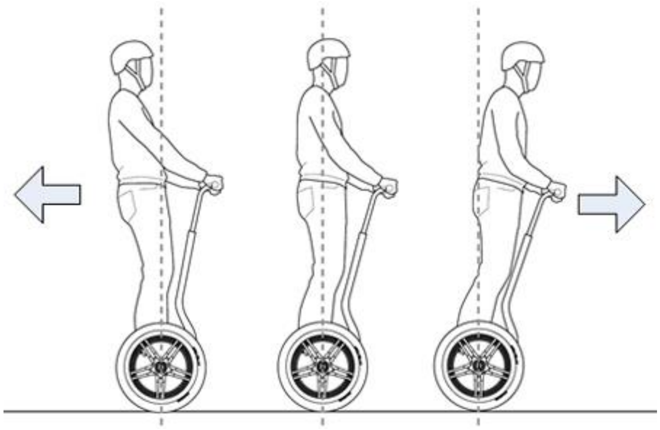
Figure 2.1

SEGWAY PT

The key to handling of Segway is its ability to maintain balance independently. The function of the system for just like how humans walk. When our brain detects that our body leans forward and loses its stability, our feet will automatically move forward to stop us from falling. The same applies when you lean backward. The Segway acts the same as a human reaction. Instead of our feet, the Segway uses a wheel to moves forward while our body is leaning forward or moving backward when lean backward. It uses the sensor to detect leaning angle and then uses a microprocessor to calculate the rotation of the motor.

Figure 2.2

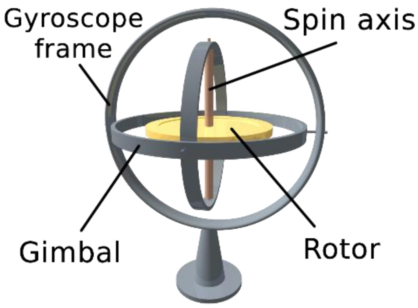
Segway Handling System



The sensor that is used to detect leaning angles is gyroscopes. Gyroscopes are the device used for measuring orientation and angular velocity. There are mainly two types of the gyroscope which are mechanical and optical. The mechanical gyroscope used the spinning rotor’s angular momentum, which caused it to maintain its attitude even when the gimbal assembly was turned. Optical gyroscopes, on the other hand, do not have any moving parts. They use the concept of laser and fiber optics to determine the rotational movement.

Figure 2.3

Mechanical Gyroscope



The modern version of gyroscopes is called a gyro sensor. The principle is the same, but the size has been reduced. The gyro sensor mechanical used oscillation to determine the tilt angle and angular velocity. Accelerometers are electromechanical devices that sense both static and dynamic forces of acceleration. It is a tool that measures acceleration in meters per second square. This sensor comes in handy when sensing vibration in system or orientation applications.

The combination of accelerometers and gyroscopes is an inertial measurement unit or IMU. It can measure various factors, including velocity, heading, acceleration, specific force, or angular rate. Most of IMU has 6 degrees of freedom. This means it has three accelerometers and three gyroscopes. However, the measurement is not precise enough to measure the accurate position and orientation of an object. The use of accelerometer and gyroscope data is to obtain an angular position or tilt angle of the object.

The accelerometer measures all forces acting on the object. It will also observe a lot more than just the gravitational force. All the small forces working on the object will disturb the measurement completely. The accelerometer data is reliable only in the long term, and a low pass filter is required. The gyroscope, on the other hand, has a different problem. The measurement of gyro sensor integration over time tends to drift and not return to zero when the system went back to its original position. So, the gyroscope data is reliable only in the short term, as it starts to drift in the long term.

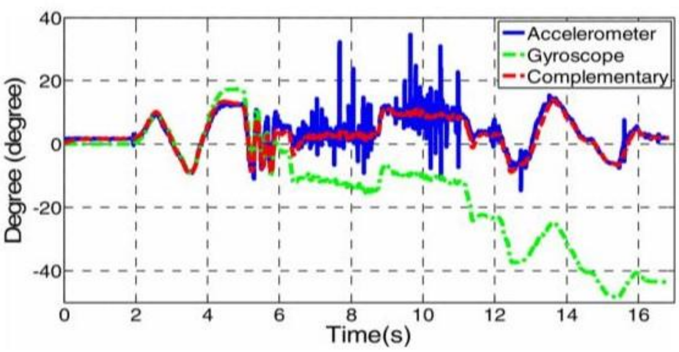
Complementary filter is the tool to bring the best result from IMU. In the short term, the data from the gyroscope is better because it is exact and not sensitive to disturbance. In the long term, the data from the accelerometer are used, as it does not drift. In its most simple form, the filter looks as follows

$$\text{Angle} = 0.98(\text{Angle} + \text{gyroData} \times dt) + 0.02(accData)$$

The gyroscope data is integrated every time-step with the current angle value. After this it is combined with the low-pass data from the accelerometer. The constants (0.98 and 0.02) have to add up to 1 but can be changed to tune the filter properly. The experiment has been taken by using a complementary filter on IMU. The coefficient that has been chosen for the complementary filter are 0.98 for gyroscope and 0.02 for an accelerometer.

Figure 2.4

Complementary Filter

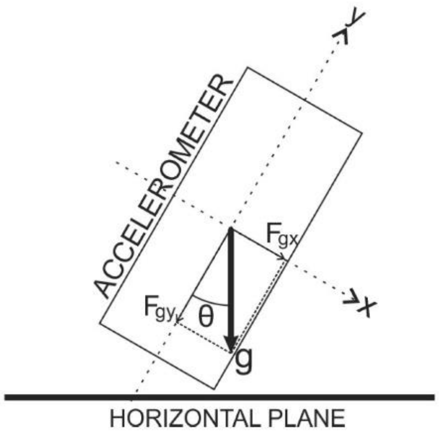


Kalman filter is another example of reducing error from the sensor. From [9], It is an algorithm that produces an estimation of unknown variables given the measurements observed over time. Kalman filters are used to estimate states based on linear dynamical systems in state-space format. This method is a lot more advance than a complementary filter. Comparing the result from using complementary filter and Kalman filter. These filters have been used to estimate the value for Uni-wheel robot. The result show that the complementary filter causes the time delay about 20ms and the estimation has a RMSE (Root Mean Square Error) of 2.8191(degree). While Kalman filter data shows the time delay about 10ms and RMSE is 2.5875(degree).

Kalman filter use for filtering the noise and error from Accelerometer-Gyro pair sensor. The sensor is mainly used to measure the tilt angle of the two-wheeled, self-balancing vehicle.

Figure 2.5

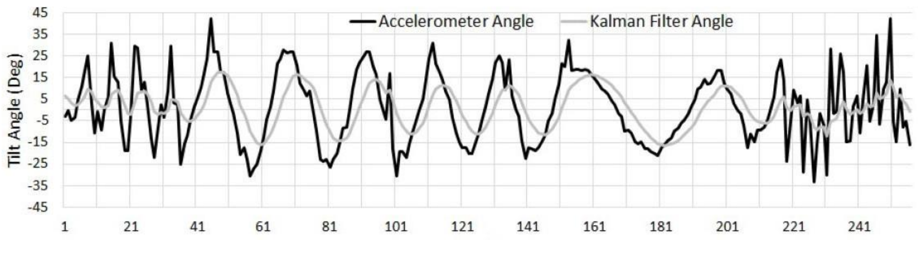
Tilt Angle Measurement



The use of accelerometers and gyros to measure tilt angle has extremely high noise and sensitive to disturbance. In additional, the nature of a self-balancing vehicle is full of vibration and a highly unstable system. So, without the proper measurement with fast response time, the robot will lose its stability. With proper calculation and Kalman filter design, the system can accurately measure the tilt angle of the robot, as shown.

Figure 2.6

Kalman Filter



In normal state the maximum deviation of the tilt angle without filter of the accelerometer are ±2 degrees because of the measurement noise. After the gyro is added as a corrective element with the Kalman filter, the maximum deviation from noise reduce to only ±0.5 degrees.

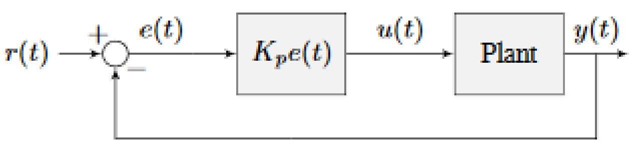
The theory behind the ability to stand with a two-wheel is on its balancing system. This system is designed using the knowledge of control theory. Control theory is the concept used to control the behavior of dynamic systems. It uses a control algorithm to develop the controller to stabilize the unstable system or even bring it toward desired value.

Design the controller for the system requires the mathematical model of that system. Mathematical models come in many forms, including dynamic model, statistical model, and differential equations. Mathematical models can be derived in many ways, and for the most general methods are Lagrangian and Newton. The Lagrangian method uses the knowledge of potential energy and kinetic energy to describe the system's dynamic, while Newton's method uses the concept of force to describe the system's behavior. Most of the time, the quality of the controller depends on how well mathematical models have been designed.

The control algorithm can be designed in many ways, and each of them has its benefit. The most straightforward controller is called Proportional controller or P-controller. This controller is a form used in a close loop system. Although P-controller is easy to implement with good response time, it makes a deviation from the set point. This problem makes pure P-controller unusable in many systems.

Figure 2.7

P-controller Block Diagram

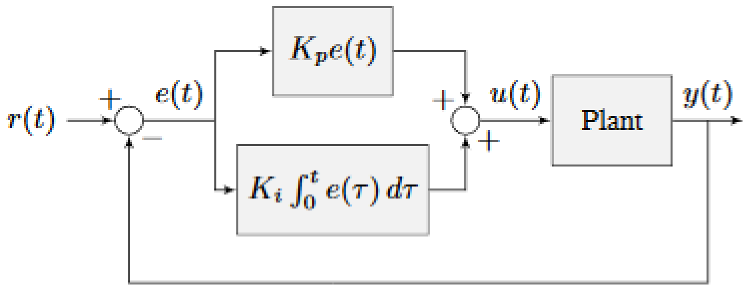


To reduce set point deviation on P-controller, I-controller must be added into the system. I-controller or Integral controller is essentially used for minimizing any offset in operation. The negative error will cause the signal to decrease, while the positive error will increase the output signal. But I-controller alone has a slow response time.

Combining P and I-controller will make the system have a faster response while lowering the deviation.

Figure 2.8

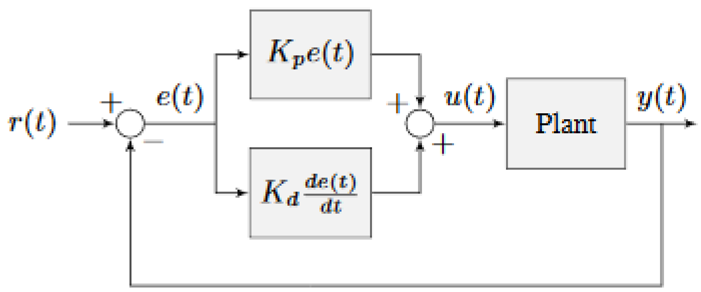
PI-controller Block Diagram



However, even though PI-controller solves set point error and has an acceptable response time, it is still slower than P-controller by almost 50 %. So, it often combines with D-control. D-control or Derivative control is a form of feed forward control. It estimates the process condition by examining the change in error, which improves the system's response time. Unlike proportional and integral controllers, derivative controllers do not bring the system to a steady state. This property makes it unusable without paring with P, I, or PI controllers to properly control the system.

Figure 2.9

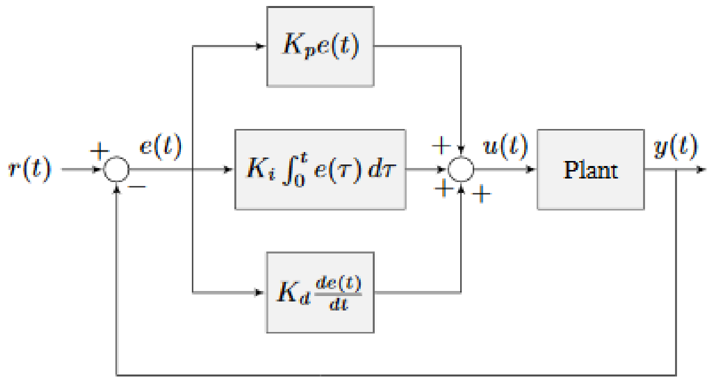
PD-controller Block Diagram



The controller that combines all P, I, and D is called PID controller. This controller is the most commonly used because it combines the benefit of each controller. It has fast response time with low steady-state error. So, the system that requires stability and accuracy PID controller is a good choice.

Figure 2.10

PID-controller Block Diagram



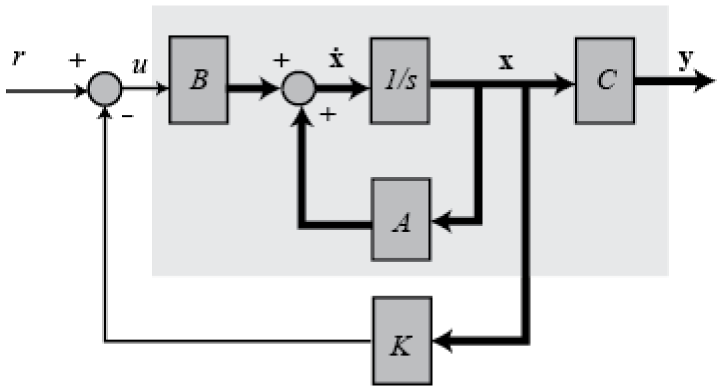
Although PID-controller is good for many control systems, it still not good enough to handle a high-complexity system like multiple inputs and outputs system. For those kinds of systems, a higher advanced controller is required. Optimal control and Robust are two primary control techniques that generally use in these kinds of problem.

Optimal control is the method of determining control and state trajectories of a dynamic system in one period of time to minimizing a performance index. There are many types of optimal control, each of them depending on a performance index, type of domain, constraint, and variables. The requirements before for applying the optimal control theory are mathematical model of the system, performance index specification, system constrain, and statement of free variables.

One of the most general optimal controllers is Linear Quadratic Regulator or LQR controller. This controller use concept of the cost function. The goal is to design the feedback control law that minimizes this cost function. The LQR controller comes with a powerful tool which is Q and R matrix. With Q-matrix, the control engineer can prioritize the control variable, while R-matrix allows choosing the right amount of gain to match the system component.

Figure 2.11

LQR-Controller Block Diagram



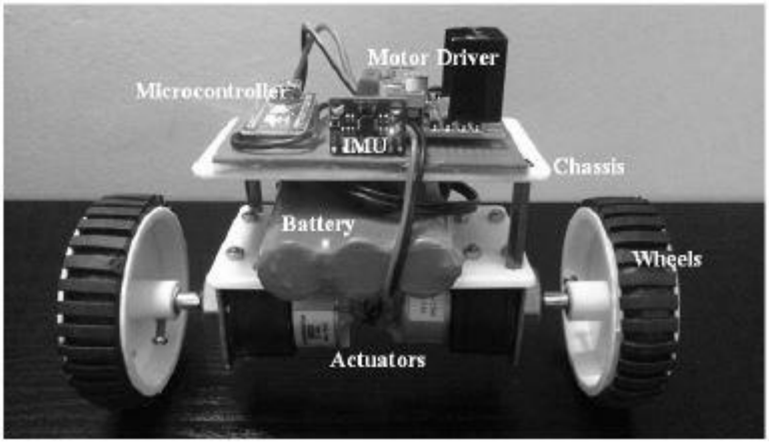
An optimal controller is the best way to find the proper control signal for the system only if our mathematical model is accurate. However, in the actual system, disturbance from the environment has a massive effect on the system. It is uncontrollable and makes the system react differently from what it should be. Although sometimes the designed controller can handle these disturbances, many sensitive systems cannot take that risk.

Robust control theory is a method to measure the change in performance of the system with varying system parameters. The power of robust control is the ability to handle the disturbances of the system. The controller can be designed so that it insensitive to change in the system, which allows them to stabilize even with the disturbances. Robust control is related to the control of unknown plants with unknown dynamics subject to unknown disturbance. This control law has a high advantage from the uncertainty of the system.

Control theory also plays a significant role in balancing robots. The instability of the system is very high. The control method generally uses to balance the robot is a proportional–integral–derivative controller or PID. These controllers have been used because it easy to apply and control the desired result. The research uses a PID controller to control the self-balancing robot.

Figure 2.12

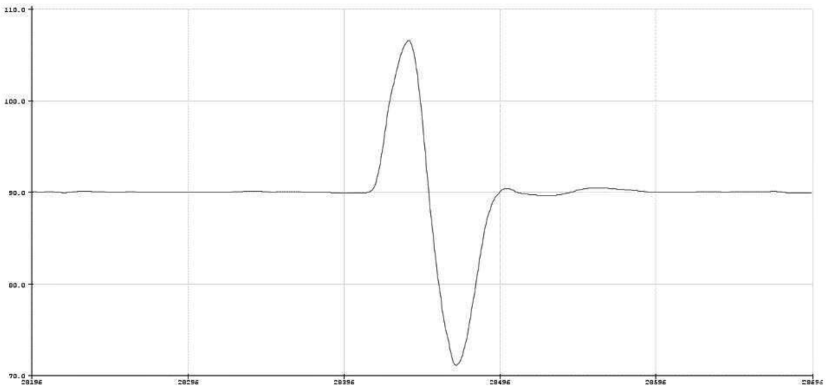
Prototype of the Self-Balancing Robot



The pitch angle of the robot has been analyzed with respect to time. The robot can stay upright with a set point of 90 degrees. When they receive the disturbances from an external source, the angle saturated between 72 and 106 degrees and came back to 90 degrees afterwards.

Figure 2.13

Tilt Angle from PID-controller



The resulted from PID good enough to make some robots stay upright. But if the robot has increase in size and complexity the more advance controller is required.

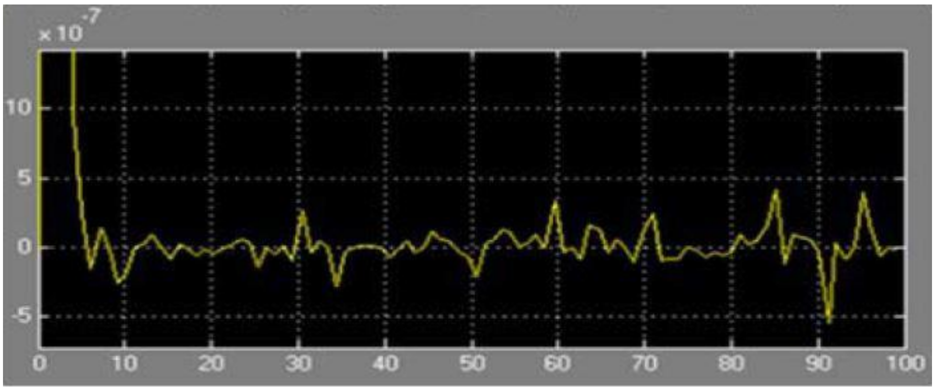
The optimal controller is the modern control method of the modern control law. One of the most popular algorithms is the Linear quadratic regulator or LQR. This controller comes with many tools that bring the system to what the user desired. To design an LQR controller, two constant values need to be selected.

The first one is the Q matrix or state weighting matrix. It is the matrix the user can set the priority of each state variable. This is the best feature for LQR because priority can easily set with a simple positive number. The second one is the R matrix or control weighting matrix. This matrix allows the user to select the value of the control signal that appropriates to the system. If the value of R is high, the system will react slower but require less power. On the contrary, if the value of R is low, the system will have a faster response but require high control signal.

The LQR method to design the control system for Uni-wheel self-balancing robots. It is one of the most complex control systems due to the highly un-stabilize system. Unlike a two-wheel robot, a Uni-wheel can fall in two directions. Such a system requires a very optimal control algorithm.

Figure 2.14

Uni-wheel Tilt angle



The result shown that the LQR can steadily control the balance of the Uni-cycle with RMSE of 2.58 degree.

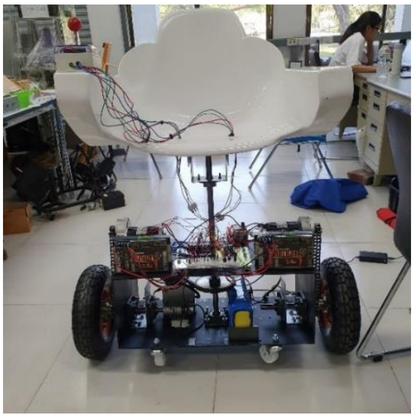
CHAPTER 3
METHODOLOGY

3.1 Mechanical Design

The body of the system have total of three motor which are motor for left wheel, right wheel and motor for rotating the chair. The lower part of the robot is called chassis.

Figure 3.1

Segway-type Turnable Wheelchair



3.1.1 CAD Model and Final Design of the System

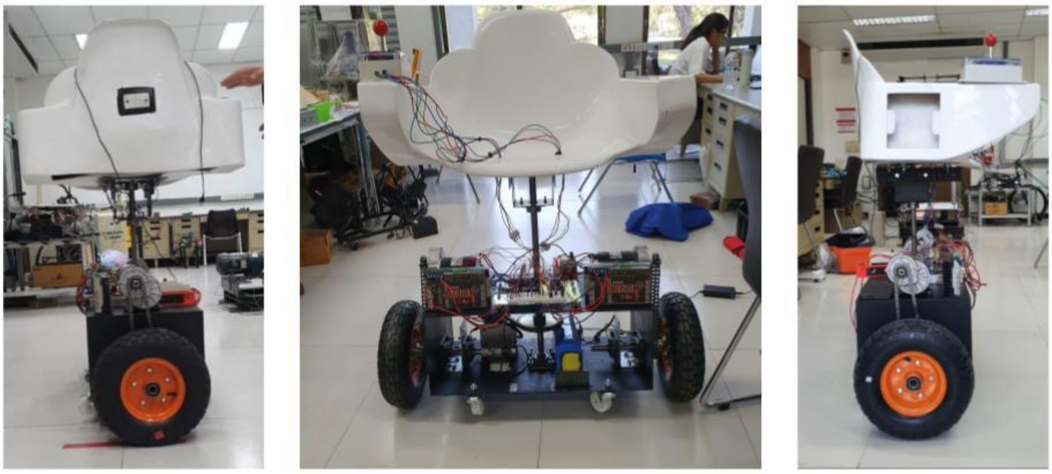
The CAD model is designed using Solid Works 2016 software.

Figure 3.2

CAD Model of Segway-turnable Wheelchair

Figure 3.3

Final Designed of Segway-type Turnable Wheelchair

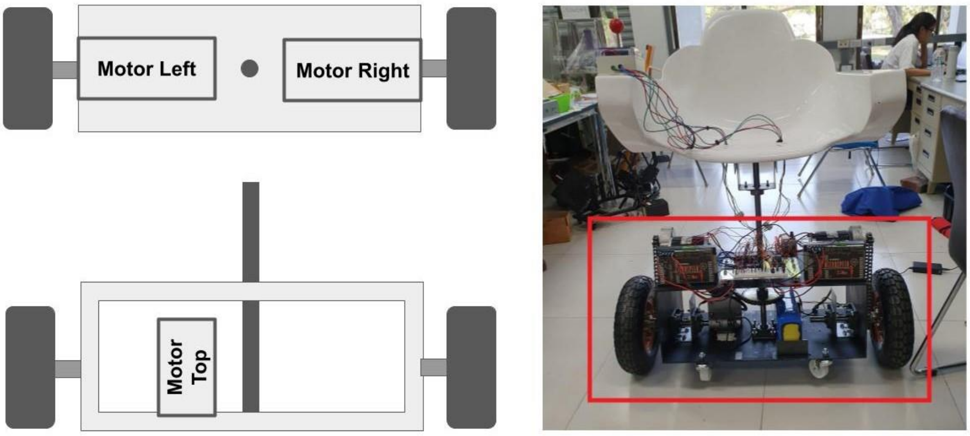


3.1.2 Chassis Composition

The most important part in this system is at chassis. It holds all three motor, two motor that uses to drive the wheel are equip on top of chassis, while the one use to rotate the chair equip inside the chassis.

Figure 3.4

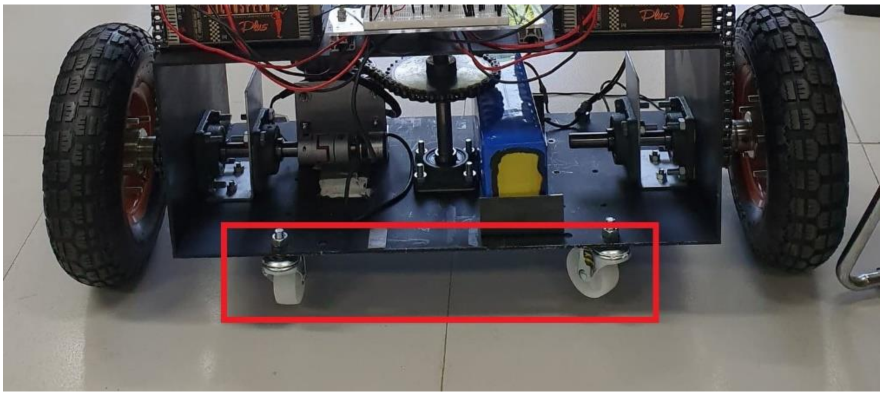
Chassis and Motor Location



The safety wheel is installing under the chassis. This wheel is using to set the maximum leaning angle of robot to 7 degree, this use help prevent robot falling to the ground.

Figure 3.5

Safety Wheel



3.1.3 Chair Installation

The purpose of this project also includes the ability to rotate the upper part(chair) without moving it wheel, so bearing is required. In this system the shaft is fix with the upper part using coupling, while go through bearing that equip in the middle of chassis. The shaft then connects to the motor which locate inside the chassis.

Figure 3.6

Chassis Bearing for Upper Part

3.2 Electrical Design

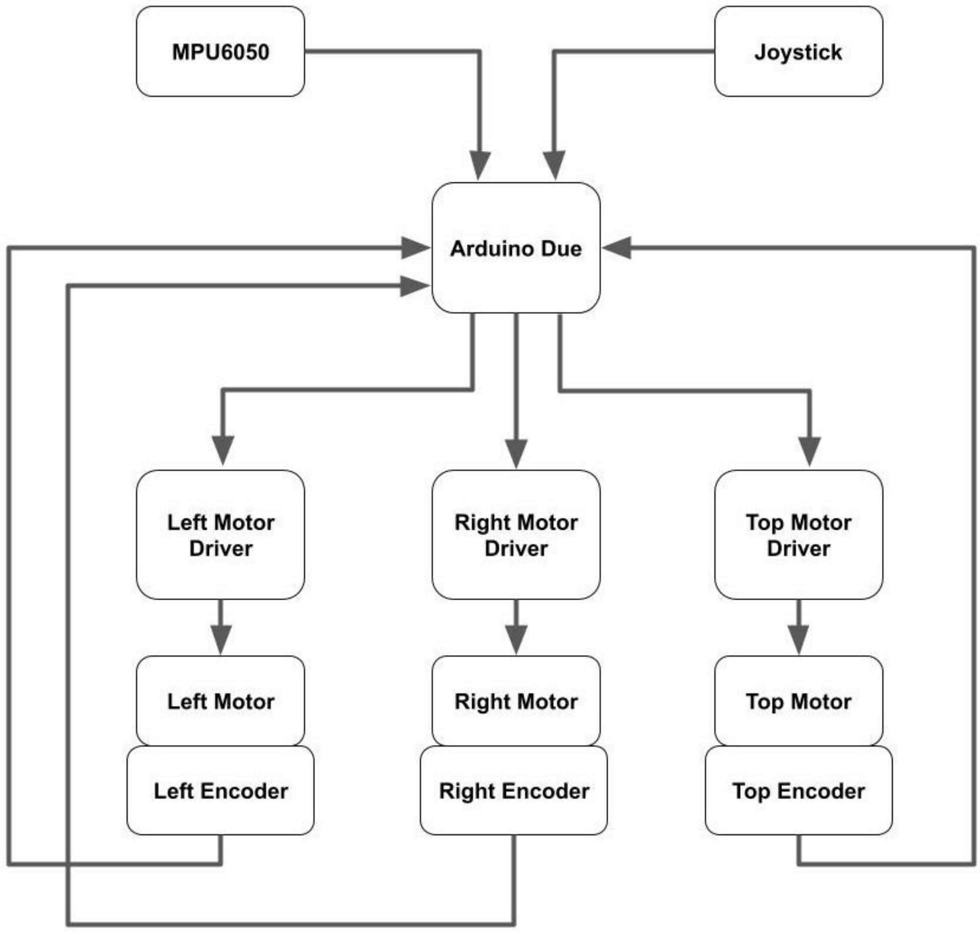
These sections are separate into 3 parts which are

1. Component
2. System inputs and outputs
3. System power distribution

3.2.1 System Inputs and Outputs

Figure 3.7

System Inputs and Outputs



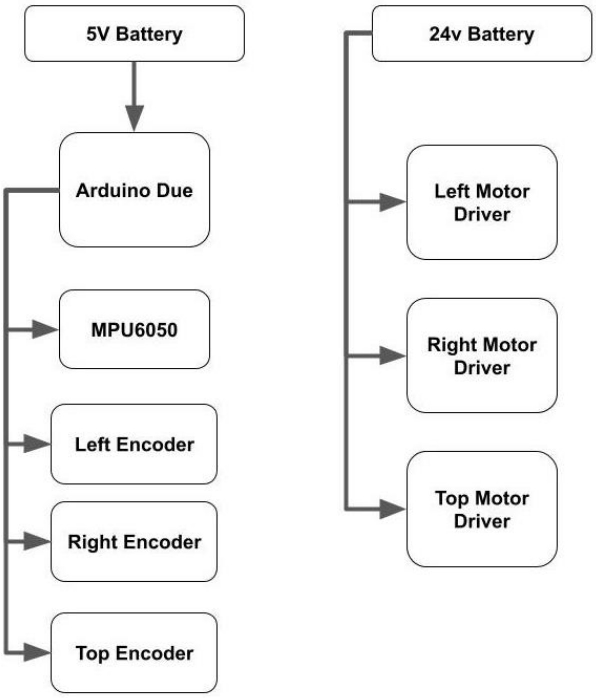
This project has total of 5 input from sensor to Arduino and 3 output from Arduino to driver. Input consists of IMU, joystick and three encoders. IMU connect to the system using I2C (SDA, SCL) pin. The joystick connects to 5 digital pins for stop, move forward, move backward, turn left and turn right. Encoder, each of them connect to 2 digital interrupt pins (A and B).

For output, it consists of 3 motor drivers, each of them requires 3 digital pins. The first pin use for PWM speed control. The second pin use for direction control and last pin use for enabling the motor.

3.2.2 System Power Distribution

Figure 3.8

System Power Distribution



For microcontroller, 5V Battery are connected to it through micro-usb port. The power supplies require for IMU and encoder are supply directly from Arduino 5V power pin.

For motor driver, the voltage require for the system is 24V and the maximum current that battery can give to three motor is at least 50A. With this high current consumption, lead-acid battery can be discharge very quickly and make the system unstable due to battery voltage drop. So, 24V lithium-ion battery is chosen for the task, this because it has the ability to discharge large amount of current without the voltage drop and not shorten it life span.

3.2.3 Component

Microcontroller

For this system, the control signal always requires making it stable. High performance microcontroller with multiple input and output pin is required. So, Arduino Due development board [13] is one of the best choices for the job. This board use very powerful 32bit Cortex M3 ARM core microcontroller. This board come with 54 digital input/output pins which can also as interrupt pin. Moreover, with 84 MHz clock speed it can easily handle the task in this system.

Figure 3.9

Arduino Due Development Board

Hand controller

The two-wheel wheelchair are designed such that the system can move to user desire position. The robot requires multiple input which are stop, forward, backward, turn left and turn right. The controller that uses in the system are arcade joystick. This joystick has over all of 8 input and can be easily understandable for most of the users.

Figure 3.10

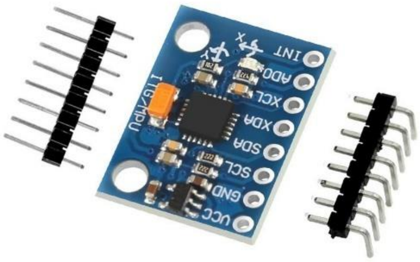
Hand Controller

Inertia measurement unit (IMU)

IMU play very important role for every balancing robot because without the accurate tilt angle measurement system can easily become unstable. In this project MPU6050 are using to measure both angle and angular velocity. MPU6050 [14] are selected because it has both gyroscope and an accelerometer which can be apply complementary filter to reduce noise and offset error of the measurement.

Figure 3.11

Inertia Measurement Unit



Rotary encoder

One of the best ways to detect system location and velocity is by using encoder. This project required three encoders to complete the system. Two encoders are measure the rotation for left and right wheel, and the third are using for measuring chair rotating angle. This project is using 200 P/R rotary encoder which accurate enough for the system.

Figure 3.12

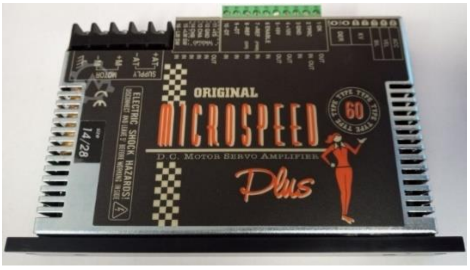
Rotary Encoder

Motor driver

The system require driver that can accurately control both speed and direction of the motor. The nature of balance robot is moving back and forth most of the time to maintain tilt angle and position, this means the motor require very high current consumption while changing it direction. So, the selected driver must powerful enough to easily withstand dose current. In this project, MICROSPEED motor driver [15] are using to drive the system. This driver has safety switched that will cut-off the power if current beyond the limit that have been set and it come with both PWM speed control and direction control.

Figure 3.13

Motor Driver



Actuator

In this project, the overall mass is around 45 kg with 1.5 meters height which mean the selected motor must have enough torque and speed to be able to balance the system. The 500W DC gear motor is chosen to accomplish this task. This motor come with torque of 2450 Nm and speed of 400 rpm.

Figure 3.14

DC Motor

CHAPTER 4
DYNAMIC MODEL AND CONTROL

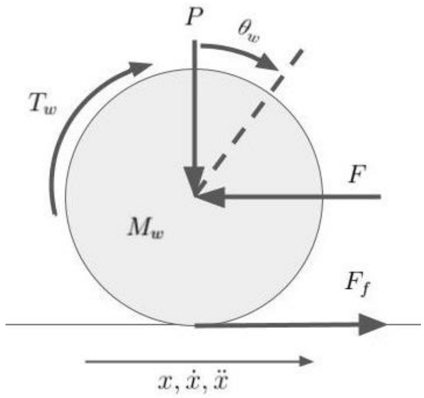
According to [16], there are four dynamic model require in this system.

- Dynamic model of wheel
- Dynamic model of pendulum (Chair)
- Dynamic model of DC motor
- Dynamic model for turning

4.1 Dynamic Model of Wheel

Figure 4.1

Wheel Free Body Diagram



x : Linear displacement

x' : Linear velocity

x'': Linear acceleration

P : Reaction force between wheel and chassis (Vertical)

F : Reaction force between wheel and chassis (Horizontal)

F_f : friction force between wheel and ground

T_ω: Wheel torque

M_ω: Wheel mass

I_ω : Wheel moment of inertia

θ_ω : Wheel angle

θ_ω' : Wheel angular velocity

θ_ω'': Wheel angular acceleration

4.1.1 Linear Motion Calculation

$$\sum F = M_\omega \ddot{x}$$

$$M_\omega \ddot{x} = F_f - F$$

$$F_f = M_\omega \ddot{x} + F \quad (1)$$

4.1.2 Angular Motion Calculation

$$\sum M = I_\omega \ddot{\theta}$$

$$I_\omega \ddot{\theta} = T_\omega - (F_f r_\omega)$$

$$T_\omega = I_\omega \ddot{\theta} + (F_f r_\omega) \quad (2)$$

Substitute equation (1) into (2) and $\ddot{\theta} = \frac{\ddot{x}}{r_\omega}$

$$T_\omega = I_\omega \frac{\ddot{x}}{r_\omega} + M_\omega \ddot{x} r_\omega + F r_\omega \quad (3)$$

Separate equation into two wheel and assume both wheels are identical. Define parameter of $T_{\omega L}$, F_L, $I_{\omega L}$ for the left wheel and $T_{\omega R}$, F_R, $I_{\omega R}$ for the right wheel. So, the torque equation in (3) become

$$T_{\omega L} = I_\omega \frac{\ddot{x}}{r_{\omega L}} + M_\omega \ddot{x} r_\omega + F_L r_\omega \quad (4)$$

$$T_{\omega R} = I_\omega \frac{\ddot{x}}{r_{\omega R}} + M_\omega \ddot{x} r_\omega + F_R r_\omega \quad (5)$$

Rearrange the equation (4) and (5) in term of force F

$$F_L = \frac{T_L}{r_\omega} - \frac{I_\omega}{r_{\omega^2}} \ddot{x} - M_\omega \ddot{x} \quad (6)$$

$$F_R = \frac{T_R}{r_\omega} - \frac{I_\omega}{r_{\omega^2}} \ddot{x} - M_\omega \ddot{x} \quad (7)$$

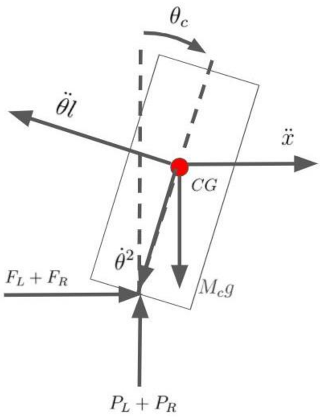
Adding equation (6) and (7)

$$F_L + F_R = \frac{T_L + T_R}{r_\omega} - \frac{2I_\omega}{r_{\omega^2}} \ddot{x} - 2M_\omega \ddot{x} \quad (8)$$

4.2 Dynamic Model of Chassis

Figure 4.2

Chassis Free Body Diagram



M_c: Mass of chassis

g : gravitational force

l : Length from contact point between wheel and chassis to center of mass

I_c : Moment of inertia of chassis

θ_c : Chassis tilt angle

$\dot{\theta}_c$: Chassis angular velocity

$\ddot{\theta}_c$: Chassis angular acceleration

4.2.1 Linear Acceleration x-axis

$$\sum F = M_c \ddot{x}$$

$$M_c \ddot{x} = (F_L + F_R) - M_c \ddot{\theta}_c l \cos\theta_c - M_c \dot{\theta}_c^2 l \sin\theta_c \quad (9)$$

4.2.2 Linear Acceleration Body-axis

$$\sum F = M_c \ddot{x} \cos\theta$$

$$M_c \ddot{x} \cos\theta = M_c g \sin\theta_c - M_c(\ddot{\theta}_c l) + (F_L + F_R)\cos\theta_c - (P_R + P_L)\sin\theta_c \quad (10)$$

4.2.3 *Angular Acceleration*

$$\sum F = I_c \ddot{\theta}$$

$$I_c \ddot{\theta} = -(F_L + F_R)\, l \cos\theta_c + (P_L + P_R)\sin\theta_c \quad (11)$$

For simplicity, the equation (9) (10) (11) will be linearize as angle θ_c very close to zero.

$$\dot{\theta}^2 \approx 0\,,\ \sin\theta_c \approx \theta_c\,, \cos\theta_c \approx 1$$

$$M_c\ddot{x} = (F_L + F_R) - M_c\ddot{\theta}_c l \tag{12}$$

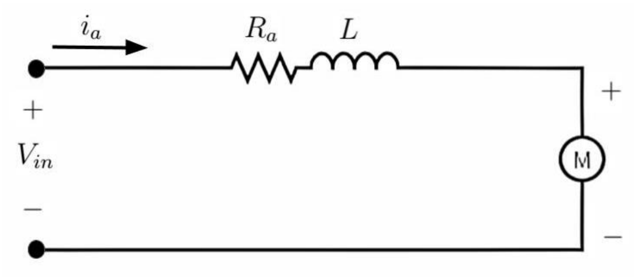
$$M_c\ddot{x} = M_c g\theta_c - M_c(\ddot{\theta}_c l) + (F_R + F_L) - (P_R + P_L)\theta_c \tag{13}$$

$$I_c\ddot{\theta} = -(F_L + F_R)\,l + (P_L + P_R)l\theta_c \tag{14}$$

4.3 Dynamic Model of DC Motor

Figure 4.3

DC motor circuit



i_a : Armature current

R_a: Armature resistance

L : Inductance

V_{in}: Input voltage

V_{inL} , V_{inR} : Input voltage for left and right motor

V_b : Back EMF voltage

K_T: Motor torque constant

K_b: Back EMF constant

T : Torque from motor

$\dot{\theta}_\omega$: Motor angular velocity (Wheel)

$$V_b \ proportional \ to \ \dot{\theta} : V_b = K_b\dot{\theta}_\omega \tag{15}$$

$$T \ proportional \ to \ i_a : T = K_T i_a \tag{16}$$

According to Kirchhoff voltage law (KVL)

$$i_a R_a + L\frac{d}{dt}i_a + V_b = V_a$$

Assume L is very small compare to R_a

$$i_a R_a + V_b = V_a \tag{17}$$

Substitute equation (15) and (16) into equation (17)

$$T = K_T \frac{V_{in}}{R_a} - K_T K_b \frac{\dot{\theta}_\omega}{R_a} \tag{18}$$

Separate the equation into left and right motor

$$T_L = K_T \frac{V_{inL}}{R_a} - K_T K_b \frac{\dot{\theta}_\omega}{R_a} \tag{19}$$

$$T_R = K_T \frac{V_{inR}}{R_a} - K_T K_b \frac{\dot{\theta}_\omega}{R_a} \tag{20}$$

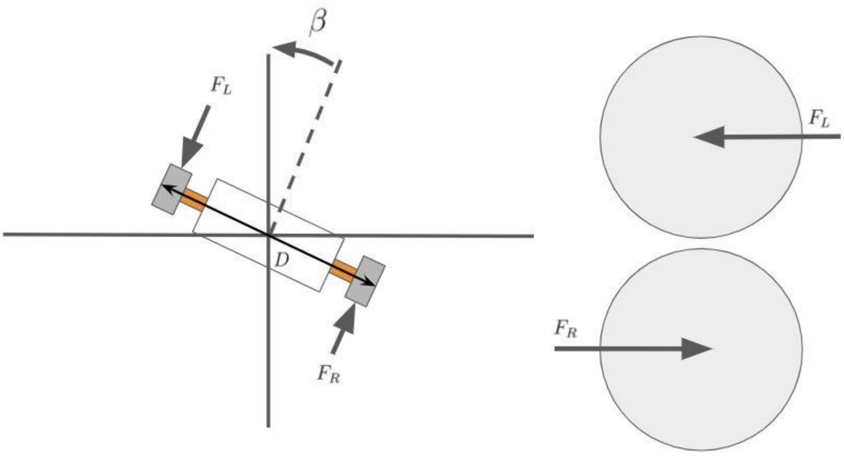
Adding equation (19) and (20) as the motor move in the same speed and direction so input voltage and angular velocity is the same.

$$T_L + T_R = K_T \frac{V_{inL}}{R_a} + K_T \frac{V_{inR}}{R_a} - 2K_T K_b \frac{\dot{\theta}_\omega}{R_a} \tag{21}$$

4.4 Dynamic Model for Turning

Figure 4.4

Body turning free body diagram



D: Diameter from left wheel to right wheel

β: Angle of body axis

Using Newton's second law for rotation

$$\sum T = I_b\ddot{\beta}$$

$$\boldsymbol{I_b\ddot{\beta} = (F_L - F_R)\frac{D}{2}} \tag{22}$$

Substitute equation (6) and (7) into (22)

$$\boldsymbol{I_b\ddot{\beta} = (\frac{T_L}{r_\omega} - \frac{T_R}{r_\omega})\frac{D}{2}} \tag{23}$$

Substitute equation (19) and (20) into (24)

$$\boldsymbol{\ddot{\beta} = \frac{K_T D}{R_a r_\omega 2I_b}V_{inL} - \frac{K_T D}{R_a r_\omega 2I_b}V_{inR}} \tag{24}$$

4.5 Calculating System Dynamic Equation

Substitute equation (8) into equation (12)

$$M_c\ddot{x} = \frac{T_L + T_R}{r_\omega} - \frac{2I_\omega}{r_\omega^2}\ddot{x} - 2M_\omega\ddot{x} - M_c\ddot{\theta}_c l$$

$$(M_c + \frac{2I_\omega}{r_\omega^2} + 2M_\omega)\ddot{x} = \frac{T_L + T_R}{r_\omega} - M_c\ddot{\theta}_c l$$

Define $\gamma = (M_c + \frac{2I_\omega}{r_\omega^2} + 2M_\omega)$ then rearrange the equation in term of $\ddot{x}$

$$\ddot{x} = \frac{T_L + T_R}{\gamma r_\omega} - \frac{M_c\ddot{\theta}_c l}{\gamma} \tag{25}$$

While angular velocity $\dot{\theta}_\omega = \frac{\dot{x}}{r_\omega}$

$$\boldsymbol{\ddot{x} = \frac{K_T}{R_a\gamma r_\omega}V_{inL} + \frac{K_T}{R_a\gamma r_\omega}V_{inR} - \frac{2K_T K_b}{R_a\gamma r_\omega}\dot{x} - \frac{M_c L}{\gamma}\ddot{\theta}_c} \tag{26}$$

Multiply equation (13) by l

$$M_c\ddot{x}l = M_c g\theta_c l - M_c(\ddot{\theta}_c l^2) + (F_R + F_L)l - (P_R + P_L)l\theta_c$$

Then add with equation (14)

$$M_c\ddot{x}l + I_c\ddot{\theta}_c = M_c g\theta_c l - M_c\ddot{\theta}_c l^2$$

Rearrange the equation

$$M_c\ddot{x}l + (I_c + M_c l^2)\ddot{\theta}_c = M_c g\theta_c l$$

$$\ddot{\theta}_c = \frac{M_c g l}{I_c + M_c l^2}\theta_c - \frac{M_c l}{I_c + M_c l^2}\ddot{x}$$

Define $\varepsilon = \frac{M_c l}{I_c+M_c l^2}$

$$\ddot{\theta}_c = \varepsilon g \theta_c - \varepsilon \ddot{x} \tag{27}$$

Substitute equation (26) into equation (27)

$$\ddot{\theta}_c = \varepsilon g \theta_c - \frac{\varepsilon K_T}{R_a \gamma r_\omega} V_{inL} - \frac{\varepsilon K_T}{R_a \gamma r_\omega} V_{inR} + \frac{2K_T K_b \varepsilon}{R_a \gamma r_\omega^2}\dot{x} + \frac{\varepsilon M_c l}{\gamma}\ddot{\theta}_c$$

$$(1 - \frac{\varepsilon M_c l}{\gamma})\ddot{\theta}_c = \varepsilon g \theta_c - \frac{\varepsilon K_T}{R_a \gamma r_\omega} V_{inL} - \frac{\varepsilon K_T}{R_a \gamma r_\omega} V_{inR} + \frac{2K_T K_b \varepsilon}{R_a \gamma r_\omega^2}\dot{x}$$

Define $\mu = 1 - \frac{\varepsilon M_c l}{\gamma}$

$$\ddot{\theta}_c = \frac{\varepsilon g}{\mu}\theta_c - \frac{\varepsilon K_T}{R_a \gamma r_\omega \mu} V_{inL} - \frac{\varepsilon K_T}{R_a \gamma r_\omega \mu} V_{inR} + \frac{2K_T K_b \varepsilon}{R_a \gamma r_\omega^2}\dot{x} \tag{28}$$

Substitute equation (27) into equation (26)

$$\ddot{x} = \frac{K_T}{R_a \gamma r_\omega} V_{inL} + \frac{K_T}{R_a \gamma r_\omega} V_{inR} - \frac{2K_T K_b}{R_a \gamma r_\omega^2}\dot{x} - \frac{M_c l \varepsilon g}{\gamma}\theta_c + \frac{M_c l \varepsilon}{\gamma}\ddot{x}$$

$$(1 - \frac{\varepsilon M_c l}{\gamma})\ddot{x} = \frac{K_T}{R_a \gamma r_\omega} V_{inL} + \frac{K_T}{R_a \gamma r_\omega} V_{inR} - \frac{2K_T K_b}{R_a \gamma r_\omega^2}\dot{x} - \frac{M_c l \varepsilon g}{\gamma}\theta_c$$

$$\ddot{x} = \frac{K_T}{R_a \gamma r_\omega \mu} V_{inL} + \frac{K_T}{R_a \gamma r_\omega \mu} V_{inR} - \frac{2K_T K_b}{R_a \gamma r_\omega^2 \mu}\dot{x} - \frac{M_c l \varepsilon g}{\gamma \mu}\theta_c \tag{29}$$

4.6 System State-space Representation

State-space of dynamic equation can be represented as

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

A: state matrix

B: input matrix

C: output matrix

x: state variable

u: input

The state variable in this system consists of 6 parameters

x1 = x = Displacement

x2 = $\dot{x}$ = Velocity

x3 = θ_c = Tilt angle

x4 = $\dot{\theta}_c$ = Angular velocity

x5 = β = Turning angle

x6 = $\dot{\beta}$ = Turning angular velocity

Convert dynamic equation (24) (28) (29) in form of state-space

$$\ddot{\beta} = \frac{K_T D}{R_a r_\omega 2I_b} V_{inL} - \frac{K_T D}{R_a r_\omega 2I_b} V_{inR}$$

$$\ddot{\theta}_c = \frac{\varepsilon g}{\mu}\theta_c - \frac{\varepsilon K_T}{R_a \gamma r_\omega \mu} V_{inL} - \frac{\varepsilon K_T}{R_a \gamma r_\omega \mu} V_{inR} + \frac{2K_T K_b \varepsilon}{R_a \gamma r_\omega^2 \mu}\dot{x}$$

$$\ddot{x} = \frac{K_T}{R_a \gamma r_\omega \mu} V_{inL} - \frac{K_T}{R_a \gamma r_\omega \mu} V_{inR} + \frac{2K_T K_b \varepsilon}{R_a \gamma r_\omega^2 \mu}\dot{x} - \frac{M_c l \varepsilon g}{\gamma \mu}\theta_c$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\beta} \\ \ddot{\beta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{2K_T K_b}{R_a \gamma r_\omega^2 \mu} & -\frac{M_c l \varepsilon g}{\gamma \mu} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{2K_T K_b \varepsilon}{R_a \gamma r_\omega^2 \mu} & \frac{\varepsilon g}{\mu} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \\ \beta \\ \dot{\beta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{K_T}{R_a \gamma r_\omega \mu} & \frac{K_T}{R_a \gamma r_\omega \mu} \\ 0 & 0 \\ -\frac{\varepsilon K_T}{R_a \gamma r_\omega \mu} & -\frac{\varepsilon K_T}{R_a \gamma r_\omega \mu} \\ 0 & 0 \\ \frac{K_T D}{R_a r_\omega 2I_b} & -\frac{\varepsilon K_T}{R_a \gamma r_\omega \mu} \end{bmatrix} \begin{bmatrix} V_{inL} \\ V_{inR} \end{bmatrix}$$

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

4.7 Linear Quadratic Regulator

The Linear Quadratic Regulator (LQR) controller is one of the best approaches to get the most optimal gain for the system. It has the ability to determine the desired closed-loop poles such that adjust the priority between the acceptable response and the amount of control energy required.

The state-space model

$$\dot{x} = Ax + Bu \tag{30}$$

Linear control law

$$u = -Gx \tag{31}$$

The gain G that minimizes cost function V

$$V = \int_{\tau}^{T} [x'(t)Q(t)x(t) + u'(t)R(t)u(t)]dt$$

Q and R: symmetric matrices

Q: The state weighting matrix

R: The control weighting matrix

The gain G for control signal from (31) can be found from

$$G = R^{-1}B'M \tag{32}$$

Where M can be found by solving algebraic Riccati equation

$$A'M + MA - MBR^{-1}B'M + Q = 0 \tag{33}$$

The optimal gain G for this system will be calculate through LQR function in MATLAB with selected value of Q and R.

This project also modifies the LQR controller when the system has the reference by fixing the error value of that variable. It can be done by select some constant for that error in such a way that the system will not fail. The selected value can be in the range of the error, select the one that not to high that lead to system failure or too low that system does not recognize. This method not just reduce the overshoot on the startup, but also steadily drive system to any desired position.

CHAPTER 5

RESULT AND DISCUSSION

This chapter are focusing on three main part which are getting gain(G) and simulation.

5.1 State-space Representation and LQR

Parameter must be identified in order to complete state space equation.

Table 5.1

Parameter for State-space

Constant Parameter	Value	Units
$\boldsymbol{M_p}$	30	kg
$\boldsymbol{M_\omega}$	5	kg
$\boldsymbol{l}$	0.5	m
$\boldsymbol{r_\omega}$	0.16	m
$\boldsymbol{I_c}$	2.72	$\text{kg}\cdot\text{m}^2$
$\boldsymbol{R_a}$	0.6	Ω
$\boldsymbol{K_b}$	1.63	-
$\boldsymbol{K_T}$	1.63	-
$\boldsymbol{g}$	9.81	m/s^2
$\boldsymbol{D}$	0.65	m

Substitute value into state space equation

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\beta} \\ \ddot{\beta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -17.14 & -9.41 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 25.16 & 28.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \\ \beta \\ \dot{\beta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0.7594 & 0.7594 \\ 0 & 0 \\ -1.15 & -1.15 \\ 0 & 0 \\ 0.4530 & -0.4530 \end{bmatrix} \begin{bmatrix} V_{inL} \\ V_{inR} \end{bmatrix}$$

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

Open loop pole location

The stability of the system can be determined by finding system pole location. The Pole located in the left-half plane are stable while pole locate in right-half plane are unstable. Pole of the system can be determine using MATLAB command.

$$pole = eig(A)$$

where A is state matrix. With the following MATLAB function the open-loop pole location of the system is

[0 -17.95 -3.33 4.13 0 0]

The result shown that one of the pole is on right-half plane, so this system is unstable.

Controllability test matrix

The controllability test matrix can be calculate using MATLAB command **C = ctrb()** Then find its rank using MATLAB command **rank(C)** where **C** is controllability test matrix. With the following MATLAB function, rank is equal to 6, system controllable.

Linear Quadratic Regulator

LQR require matrix Q and R in order to complete the system. In this project the selected Q and R matrix are

$$y = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The gain G for control input can be calculate using MATLAB function **lqr()**

$$G = \begin{bmatrix} 3.16 & 23.68 & -82.44 & -20 & 7.07 & 4.01 \\ 3.16 & 23.68 & -82.44 & -20 & -7.07 & -4.01 \end{bmatrix}$$

Close-loop pole location

With new gain for control input, new state matrix (A_c) can be calculate as

$$A_c = A - BG$$

Pole location of close loop can be check using the same method.

[-18.13 -4.5 -3.04 -0.29 -1.8183+1.76i -1.8183-1.76i]

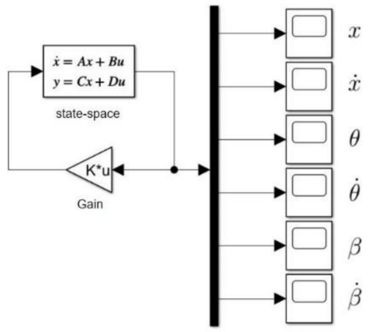
5.2 Simulation

The simulation is perform using **Simulink**, the close-loop diagram for state feedback controller without reference can be represent as figure 5.1.

5.2.1 Simulation System without Reference

Figure 5.1

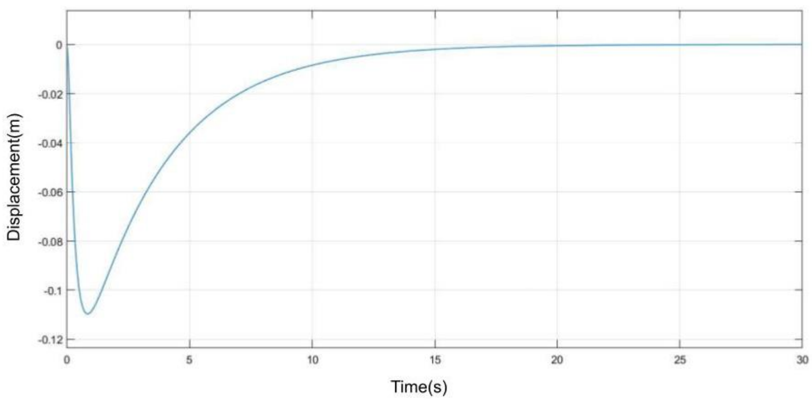
State-space Representation without Reference



The simulation of the system has initial condition angle θ = 0.052 rad, $\dot{\theta}$ = 0.087 rad/s, β = 0.174 rad and $\dot{\beta}$ = 0.174 rad/s

Figure 5.2

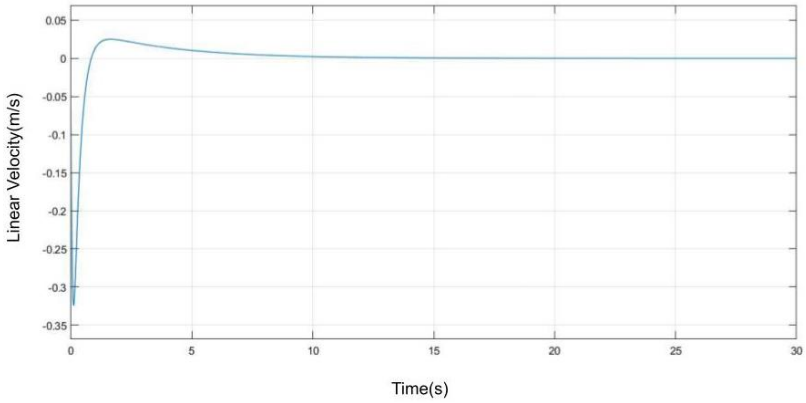
Displacement without Reference



Rise time(s) = 7.02, Settling time(s) = 25, steady-state error = 0

Figure 5.3

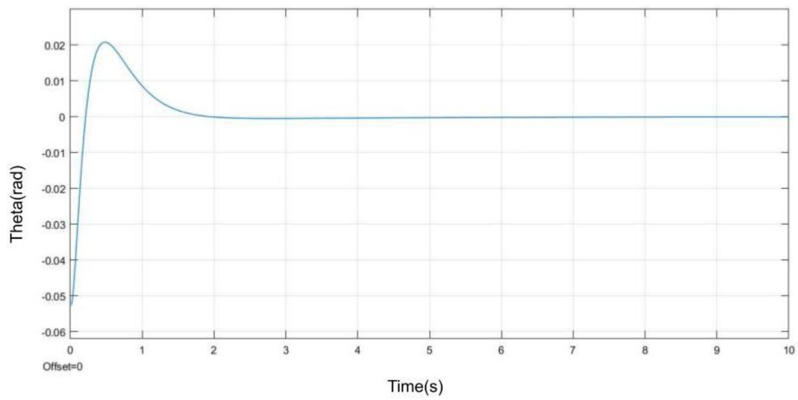
Linear Velocity without Reference



Rise time(s) = 0.46, Settling time(s) = 21, steady-state error = 0

Figure 5.4

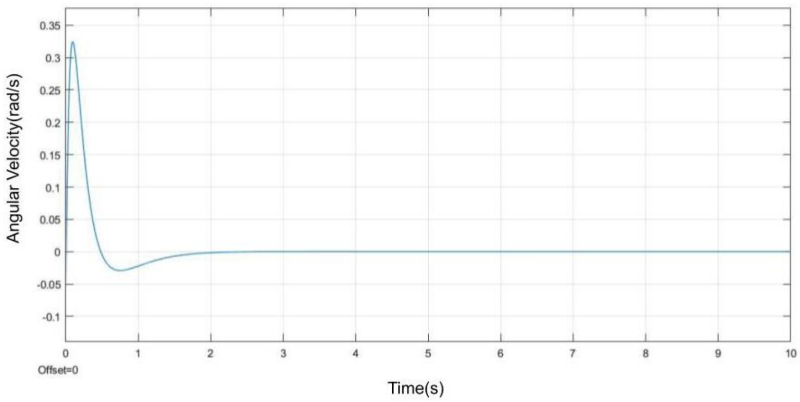
Tile Angle without Reference



Rise time(s) = 0.15, Settling time(s) = 7, steady-state error = 0

Figure 5.5

Angular Velocity without Reference



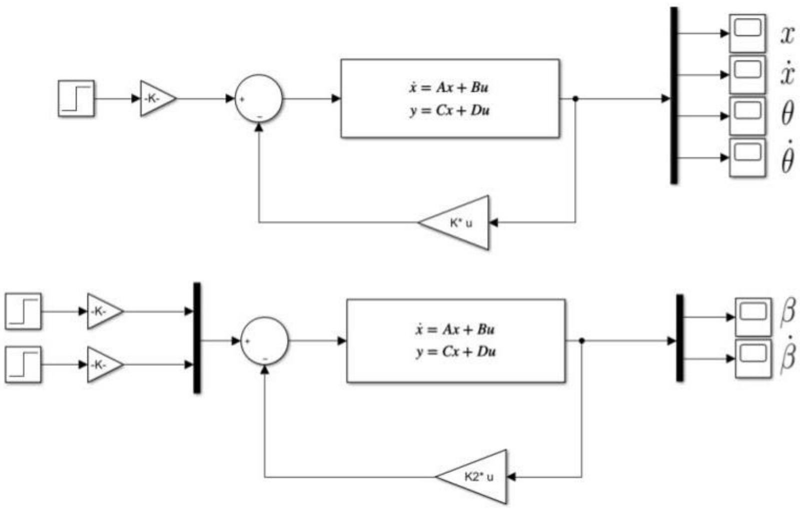
Rise time(s) = 0.013, Settling time(s) = 3, steady-state error = 0

5.2.2 Simulation System with Reference

The LQR controller can be design in such a way that the system become tracking system.

Figure 5.6

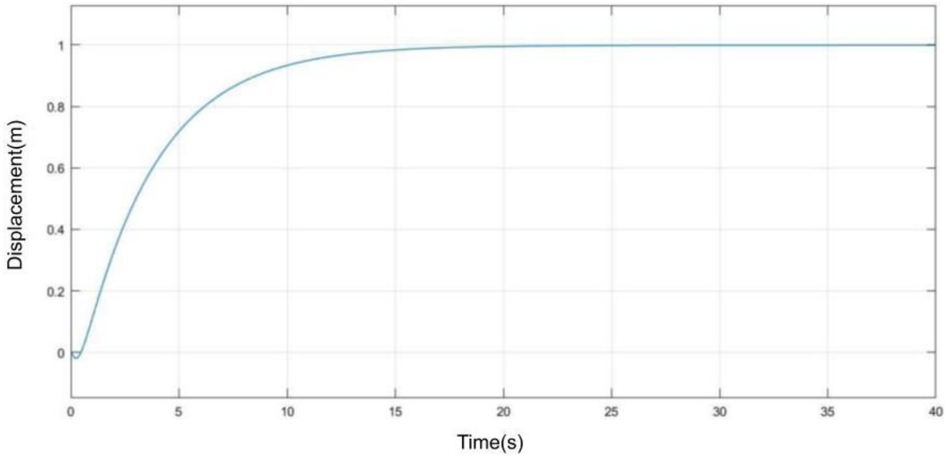
System Separation



The simulation of the system set the desire position x = 1 meter and rotate β = 90 degrees or 1.57 rad while all initial condition is zero.

Figure 5.7

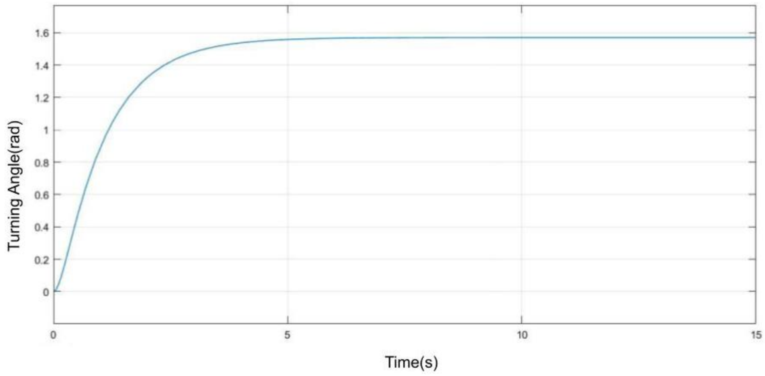
Displacement with Reference



Rise time(s) = 7, Settling time(s) = 23, steady-state error = 0

Figure 5.8

Turning Angle with Reference



Rise time(s) = 0.87, Settling time(s) = 3.5, steady-state error = 0

5.3 Experiment and Result

The experiment is divided into 3 categories.

- Tilt angle balancing
- Linear translation
- Own axis rotation

Each experiment is using LQR algorithm to calculate gain for input signal.

5.3.1 Tilt Angle Balancing

Figure 5.9

Tilt angle balancing

Tilt angle balancing is the most significant system for this robot, without balancing system the robot cannot move nor turn.

Figure 5.10

Displacement

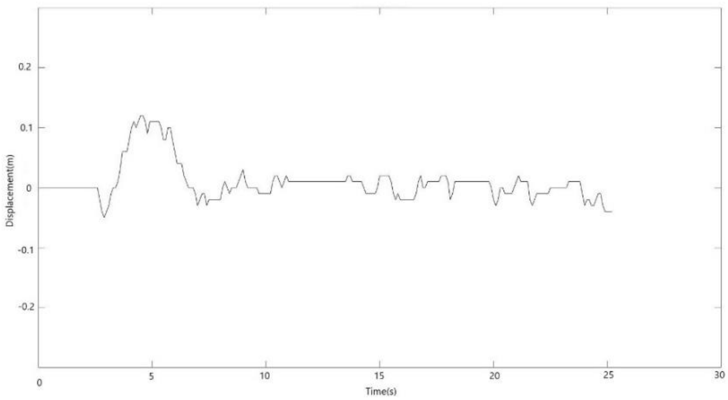


Figure 5.11

Linear Velocity

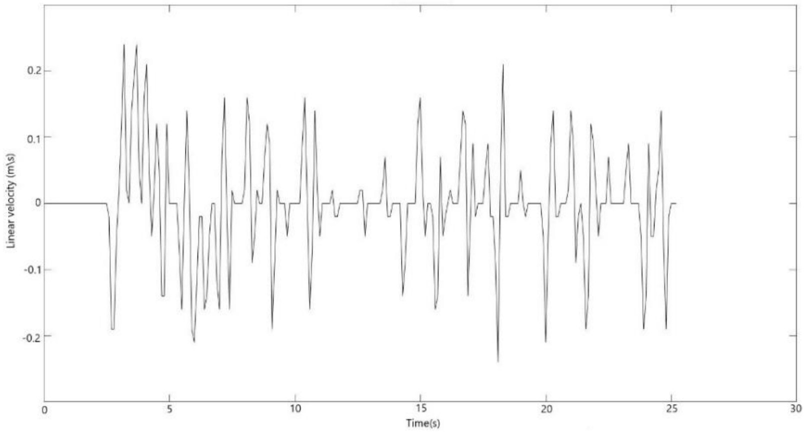


Figure 5.12

Tilt Angle

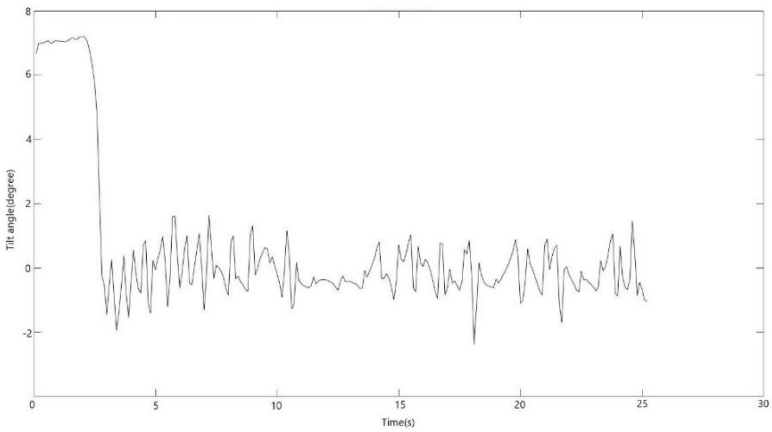
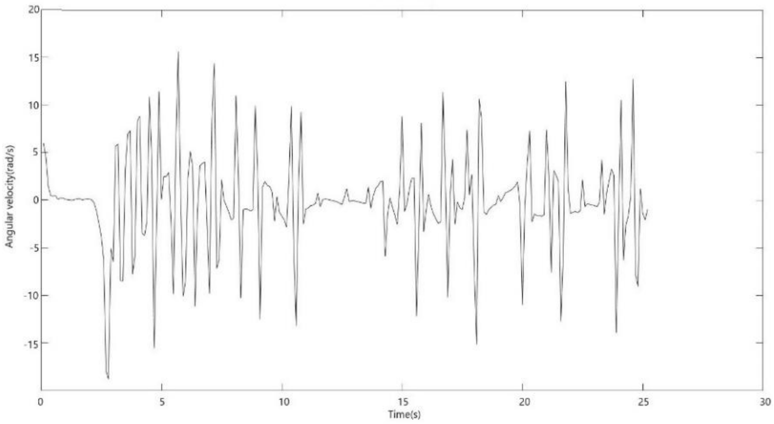


Figure 5.13

Angular Velocity



The result of tilt angle shown extremely fast response. The time takes from initial tilt angle (7 degree) to 0 degree within 280 ms. At the same time, robot move up to maximum of 0.12 m, then it takes 920 ms to recover into it original position.

Table 5.2

Tilt Angle Balancing

Parameter	Displacement	Tilt angle
Settling time (s)	1.56	0.28
RMSE	0.516	0.684

5.3.2 Tilt Angle Balancing with Chair Rotation

In this experiment, upper part of the robot is slowly rotate counterclockwise to create disturbance. This rotation moves the robot center of mass to different angle in the body.

Figure 5.14

Balancing with Top Rotation

Figure 5.15

Turning Angle

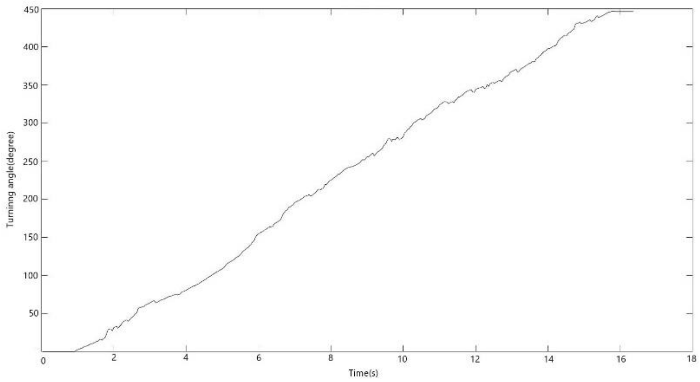
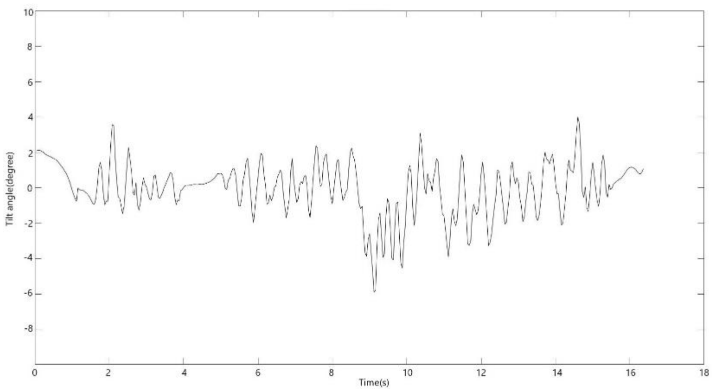


Figure 5.16

Tilt Angle



The result shown that robot can stay balance without falling even with the change in center of mass. The highest lean angle of the robot is at −4.5 degree when the chair rotates up to 278.64 degree at time 9.88 s.

Table 5.3

Tilt Angle Balancing with Chair Rotation

Parameter	**Tilt angle**
RMSE	1.386

5.3.3 Linear Translation

In this experiment, the system is testing when having reference. The goal is to move 1 meter forward while maintain the balance.

Figure 5.17

Robot Translation

Figure 5.18

Displacement

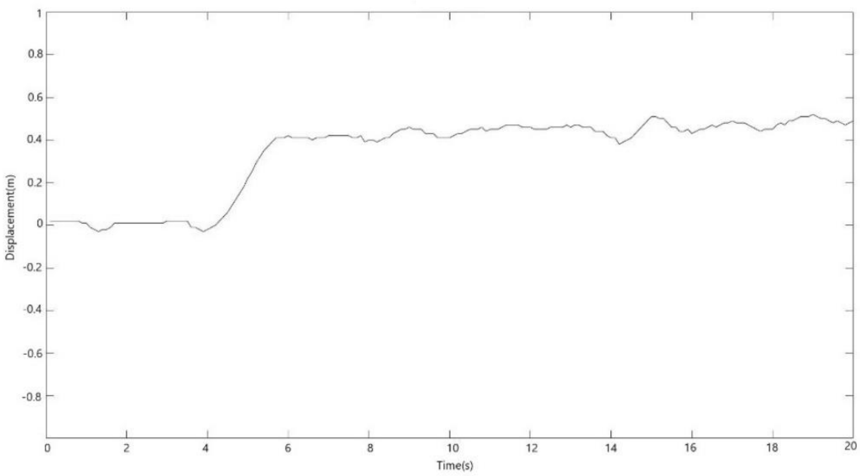


Figure 5.19

Linear Velocity

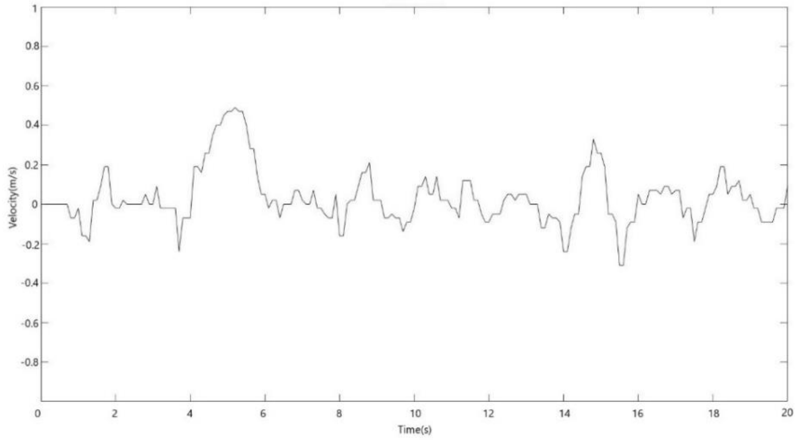


Figure 5.20

Tilt Angle

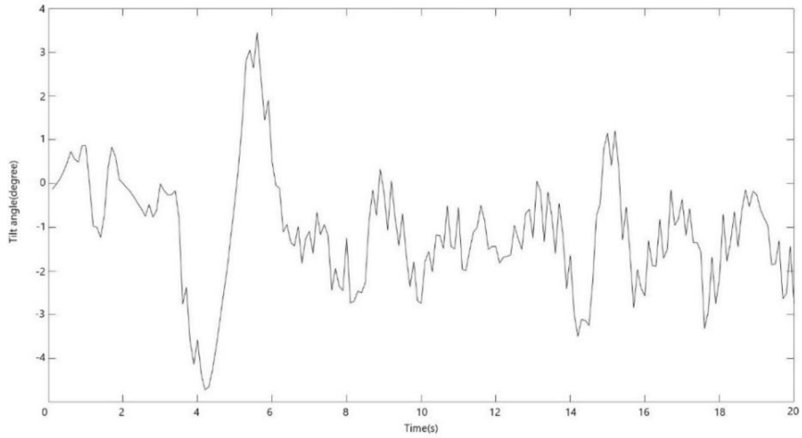
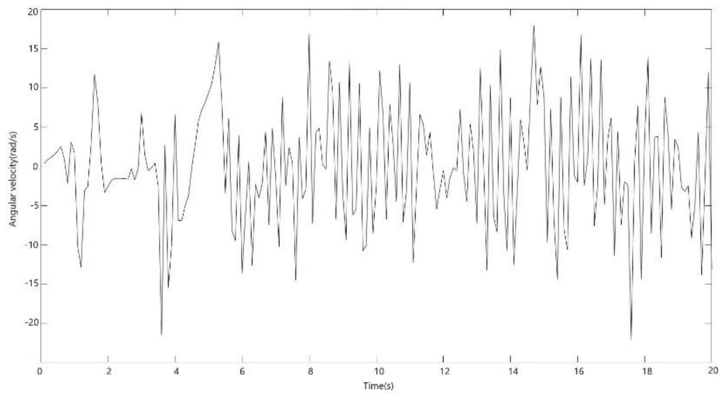


Figure 5.21

Angular Velocity



The most important part of this experiment is on translation. The set point is set to 1 meter, however the robot only moves up to average of 0.46 m. This happens due to low state weighting matrix on displacement. So, when the error gets close to 0, it is losing the power to drive the system forward.

Table 5.4

Linear Translation

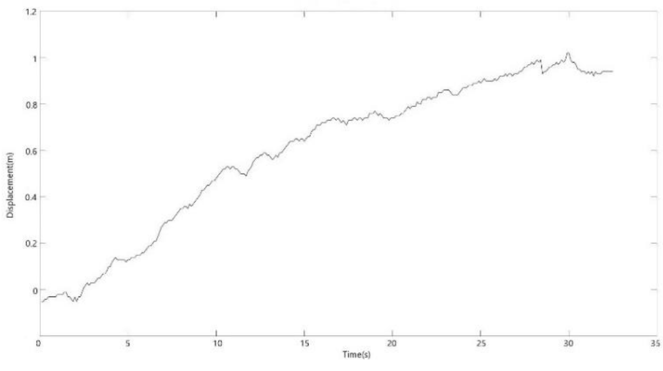
Parameter	Tile angle	Displacement
RMSE	1.51	0.54
Settling time (s)	-	1.8

5.3.4 Linear Translation (LQR with Low Gain Error Saturation)

From previous experiment, the error between desire and current position is very high. So, in order to solve this problem, controller must be modified. This can be done by using LQR with low gain error saturation for displacement such that it has enough power to drive to desire position then bring the variable back to normal after it reach desire location.

Figure 5.22

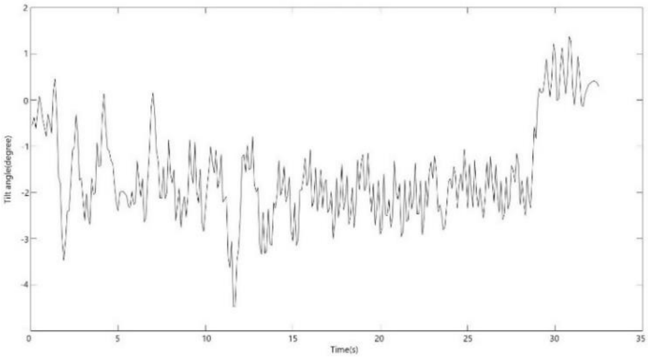
Displacement



The result shown that the robot is moving slowly to the desire position and the curve is smoother compare to previous experiment. The system takes 30 s to bring the system to 1 meter, it reacts slower compare to normal state variable, however it can bring the system to exact desire location.

Figure 5.23

Tilt Angle



The smooth movement also benefit the stability of tilt angle. The tilt angle always be the most important part of the system.

Table 5.5

Linear Translation with Low Gain Error and Saturation

Parameter	**Tilt angle**	**Displacement**
RMSE	0.19	0.07
Settling time (s)	-	30

5.3.5 Self-axis Rotation

This experiment is focus on robot turning by rotate in its axis to change heading to different direction.

Figure 5.24

Turning

Figure 5.25

Turning Angle

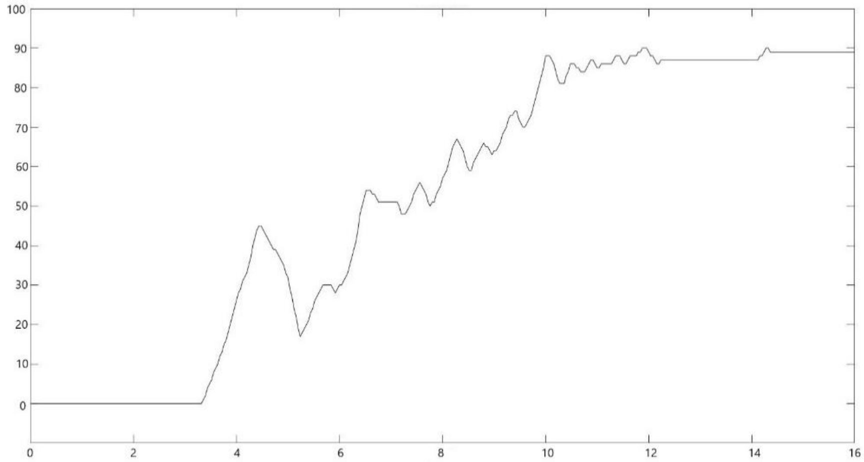
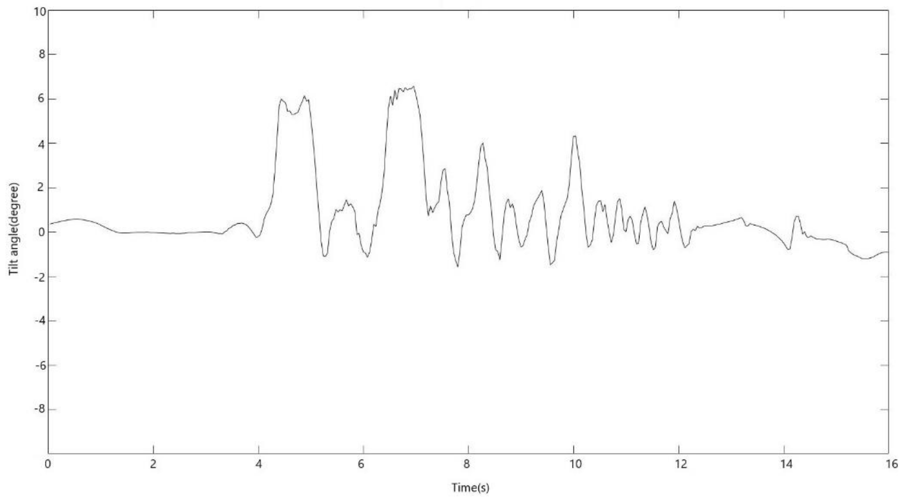


Figure 5.26

Tilt Angle



This result shown that the system successfully rotates to 90 degree without falling. The system can reach the desire position within 8. However, the robot turning very fast at start due to high error, and it make the system shudder which make the robot lean down to almost 7 degree.

Table 5.6

Turning

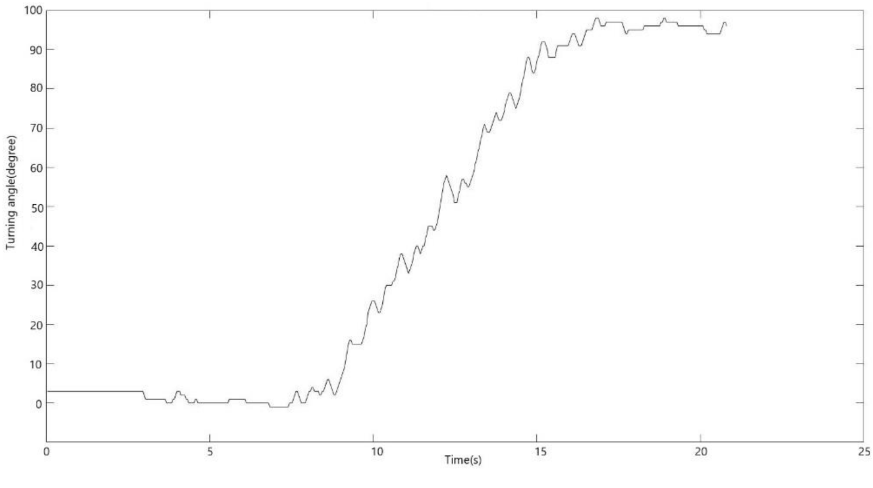
Parameter	Tilt angle	Turning angle
RMSE	2.36	87.43
Settling time (s)	-	10.96

5.3.6 Self-axis Rotation (LQR with Low Gain Error Saturation)

Turning while maintain tilt angle can be difficult. This because the wheel has to do two things at the same time, therefore the robot most of the time have high saturation. However, with the LQR with low gain error saturation approach, it can be used to improve the stability and accuracy of the system. With constant error, it can enhance the smoothness while turning to desire angle.

Figure 5.27

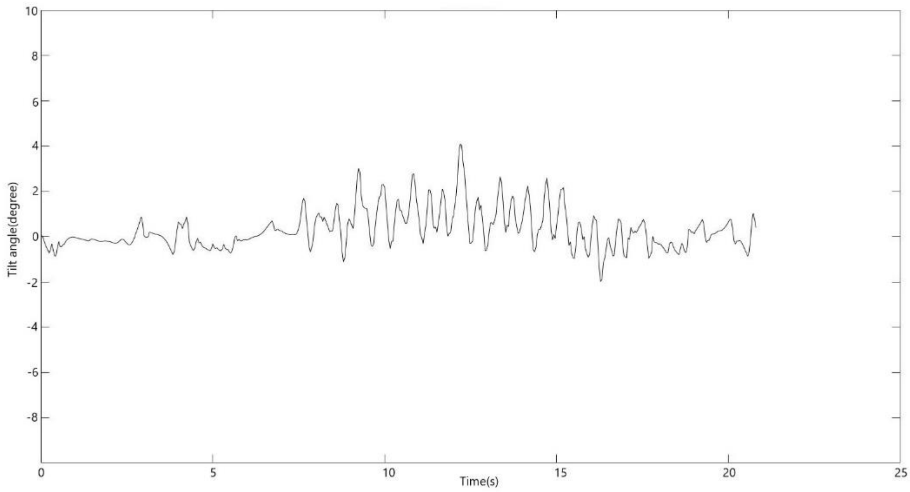
Turning Angle



The result shown that the system has improve in many perspectives. The system takes only 7.8 s to rotate the self to 90 degree with a lot more less saturation compare to normal system.

Figure 5.28

Tilt Angle



The improvement on steadiness while turning greatly increase the stability of tilt angle. The maximum overshoot of the system decreases to just 4 degree.

Table 5.7

Turning with Low Gain Error and Saturation

Parameter	**Tilt angle**	**Turning angle**
RMSE	0.845	2.65
Settling time (s)	-	7.8

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Conclusion

The Segway-turnable wheelchair has been successfully developed and examination. This wheelchair has two wheels with ability to turn the chair to any desired direction. This wheelchair has similar mechanism with two wheels robot, it has to be able to stay balance with it own body mass and disturbance from moving.

The controller has been designed using LQR as main algorithm. However, the performance is not good enough to complete the task, so new version of controller has been developed. The controller called LQR with low gain error saturation. Normally in tracking system, the gain has to multiply with error to get control signal. This creates large amount of starting signal and losing it power at the error get smaller. Nonetheless, with the modified LQR, it consistently provide the input signal which help to improve the performance of the system.

Controller for this system have three main capability consist of balancing, moving and turning. The most crucial system is balancing because this is the core ability, without steady tilt angle the robot cannot complete any other task. For moving and turning, the selection of state weighting matrix pay significant role. This because if the system gives too high priority on moving and turning, the system is not able to keep it up right position. However, if the priority is too low, the system will not have enough power to drive the system to desired reference. So, selecting unbalance state weighing matrix for the system can lead to system failure.

6.2 Future Works

The system can now balance and move with no load condition, but the user load is required in the actual application. To improve the system further, firstly, the system motor driver needed to be upgraded to make it powerful enough to hold the user's weight. Next, the model of the wheelchair has to be revised as the current one still now suitable for holding the user weight. Installing a supporter like a thrust bearing can make the system more sustainable. Moreover, adding the rod to hold the wheelchair while the chair rotates will make this system more stable and safer.

REFERENCES

Peter Woolf et al, Assistant Professor Chemical Engineering, University Michigan, Industrial and Systems Engineering

Victor M. Becerra (2008) Optimal control, Scholarpedia, 3(1):5354

Leo Rollins Carnegie Mellon University, 18-849b Dependable Embedded System (1999)

Chandrasekharan, Robust Control of Linear Dynamical Systems (1996), Academic Press

The Editors of Encyclopaedia Britannica, https://www.britannica.com/technology/gyroscope

Yutthana Pititeeraphab,The effect of average filter for complementary filter and Kalman filter based on measurement angle (2016), *9th Biomedical Engineering International Conference*

Youngjoo Kim and Hyochoong Bang, Introduction to Kalman Filter and Its Applications (2018)

J. Juan Rinc´on Pasaye, Tilt Measurement based on an Accelerometer, a Gyro and a Kalman Filter to Control a Self-Balancing Vehicle (2013), *IEEE International Autumn Meeting on Power Electronics and Computing (ROPEC)*

Chinmay Samak, Design of a Two-Wheel Self-Balancing Robot with the Implementation of a Novel State Feedback for PID Controller using On-Board State Estimation Algorithm, Department of Mechatronics, SRM Institute of Science and Technology.

Miseon Han, Implementation of Unicycle Segway Using Unscented Kalman Filter in LQR control (2013), *10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*

InvenSense Inc, www.invensense.com, PS-MPU-6000A-00 AXOR’s, Microspeed Pluse Manual

Mahmoud Khaled, Balancing a Two Wheeled Robot (2009), *research gate*