© Universiti Tun Hussein Onn Malaysia Publisher's Office

JAMEA

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/jamea

e-ISSN : 2716-6201

Journal of Advanced Mechanical Engineering Applications

Simulation and Control of Door Lock System Using PID Controller

Wan Muhammad Haikal Wan Huzainizam[1,2], Badrul Aisham Md Zain[1,2*]

[1]Advanced Dynamic and Automation Control Research Group (ADCARe),
Faculty of Mechanical Engineering & Manufacturing,
Universiti Tun Hussein Onn Malaysia (UTHM), 86400 Parit Raja, Batu Pahat, Johor, MALAYSIA

[2]Faculty of Mechanical Engineering and Manufacturing,
Universiti Tun Hussein Onn Malaysia (UTHM), 86400 Parit Raja, Batu Pahat, Johor, MALAYSIA

*Corresponding Author

DOI: https://doi.org/10.30880/jamea.2021.02.02.005
Received 03 July 2021; Accepted 15 November 2021; Available online 12 December 2021

Abstract: A Proportional–Integral–Derivative (PID) controller (or three-term controller) is a feedback-based control loop mechanism that is extensively employed in industrial control systems and a range of other applications that need constantly modulated control. Technologies using PID controller can be considered efficient and secure compared to other control loop mechanism. The aim of this project is to produce a more efficient way of locking and unlocking the door instead of using traditional key method by implementing the usage of RFID. Next, to produce a mathematical modelling of the PID controller system and use MATLAB software to validate the data. Lastly, to create a stable control system by introducing PID into the system and operate in a best manner. The analysis of stabilization of the system involving the time response analysis, root locus, bode plot and Nyquist plot via MatLab software. The usage of PID controller also involve on tuning the controller through try and error tuning has result to an overshoot percentage value of 8.3% comparing to the result without controller at 25%. To conclude, the present of PID controller into the system has set a major effect on the efficiency and keep the actual output from the door locking process as close to the target or setpoint output as possible.

Keywords: Door lock system, PID controller, Radio Frequency Identification Device (RFID)

1. Introduction

1.1 Background of the Project

This project involved an Arduino-based project using Proportional–Integral–Derivative (PID) controller with the implementation of RFID to lock and unlock the door. Technologies using PID controller can be considered efficient and more stable compared to other types of controllers [1, 2] available in the market-based on the result from previous research and application plus comparing to the result we obtained for this research. RFID devices consist of tags or transponders for radio frequency and readers of radio frequency tags or transceivers [3, 4]. Usually, transponders themselves consist of integrated circuits linked to an antenna. The use of microchips based on silicon allows a wide range of functions ability to be integrated into the transponder [3]. Typical features range from large read or write memories to integrated sensors to function ability for encryption and access control [4]. The benefit of using PID controller is that the hardware is inexpensive. The open standards based PID system can provide a widely known and understandable application procedure from various open-source guides.

*Corresponding author: shamumi11@yahoo.co.uk, aisham@uthm.edu.my

In these modern days, the traditional door lock is not as secure as it was in the past. Over the years, crime cases such as theft, robbery and burglary have been on the rise [5]. This proves that the traditional key lock is becoming less reliable in providing people with a sense of security. The house is an important means for human beings because it can provide all its occupants with a feeling of safety from various criminal acts and can also be used as a place to store their valuables. Crime, especially burglary and theft, is becoming more prevalent as the ability of thieves increases in their actions. The lack of security and the high cost of extra security make vandalism and theft of valuables even higher [5].

Unlike conventional keys, electronic keys move the key automatically after input from the identifier. The use of the PID controller for the door lock system will therefore provide more protection and security compared to the traditional key lock. One of the equipments to perform that can be used is RFID (Radio-Frequency Recognizable Evidence) [4], RFID, Radio Frequency Identification is a basic and inexpensive technology that enables wireless data transmission. This technology has not been used very often in the industry due to lack of standardization among the manufacturing companies earlier [6].

RFID technologies are efficient and secure in comparison to other mediums. The involvement of the PID controller in the RFID system is not a single product but a global system, which comprises three basic elements: the RFID tag (transponder), the reader (transceiver) and the back-end application system (or database) that require the support of the computer network [6]. It is very important to authenticate the user before entering a secure space and RFID provide this solution [7]. The system enables users to access and check out under fast, secure, and convenient conditions [7]. The system provides a door locking system that opens when the user places their tag in contact with the reader and the user data that matches the information already stored in the database [5].

The PID controller system controls the opening and closing of the doors [8]. Throughout this study, PID technology was implemented to provide a solution for secure access to space while keeping the user's record. Here, passive of type of RFID is involved [9]. Passive types of RFID are battery-less and have the power to operate from the reader. The main advantages of passive RFID are its cost-effectiveness and small size [9]. Due to the first advantages, inventory tracking technology is widely used. Current antenna technology makes it possible to be smaller in size. The PID controller door lock is the latest system or a more secure way to lock a door. As for now, most of the door lock still uses the traditional lock. As known, a traditional lock requires the key to be inserted and to manually turn the key to unlock the door.

2. Conventional Automated Door Lock System

Today, there is several technologies used to enhance security systems especially on the door [5], or any related entrance to ease the process of entry including strengthening the safety of the user [7]. Electronic door using PID, and RFID based on Arduino UNO is no longer new technology for human being as Arduino have been widely used in various application for door security system, toll payment system, garment security tags and many more. This device is an automated door lock attachment without using an existing deadbolt-locked door.

Fig. 1 show the conventional automated door system that used controller and mechanical concept for house security [4].

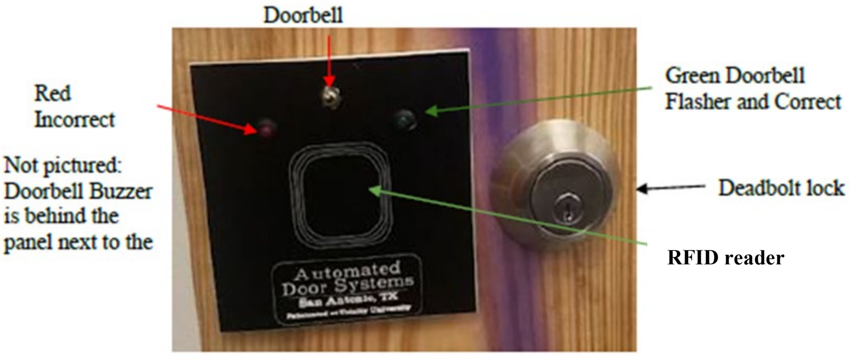

Doorbell
Red Incorrect
Green Doorbell Flasher and Correct
Not pictured: Doorbell Buzzer is behind the panel next to the
Deadbolt lock
RFID reader
Automated Door Systems San Antonio, TX

Fig. 1 - Automated door system

The component used for the system in Fig. 2.1 also includes a reader that was connected to the controller to transfer the data in form of potential different (according to every component transfer function) to operate by locking and unlocking the door. These processes are the same as any system that implementing an automatic process which is by using a specific type of controller.

Fig. 2 and 3 show the component and basic setup of an automated door lock system by attaching all the components properly in its place next, managing all the wiring systems into a box to including the controller module [4].

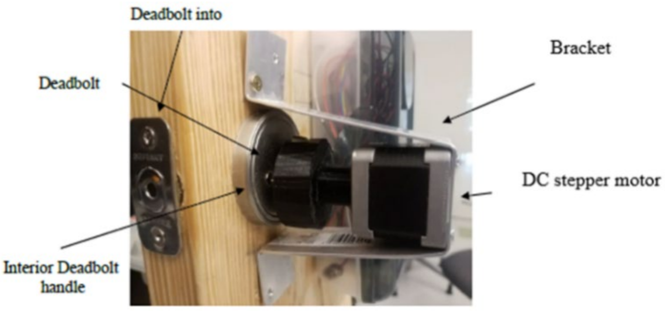


Fig. 2 - Component of automated door lock system

Fig. 3 - Wiring and controller management

2.1 Component and Function

The system consists of six main components which work in sync. Each of the components has a vital function to ensure that the system runs as intended. The components that are used in the system are:

- Arduino Uno: Arduino is an open-source platform that can be used to construct or/and program electronics components [5]. Its function is to receive and send information to most devices. This also can be done through the internet to command a device. Arduino Uno is the circuit board that is programmable through a software called Simplified C++ [5].

- Stepper motor with driver: A stepper motor is a brushless, synchronous motor which divides a full rotation into several steps [10]. Step motors only rotate in discrete angles which are programmed to be in a lock and unlock position at a specified angle [10]. The function of a stepper motor in this system is to enable the door to lock and unlock.

- RFID Module: Radio Frequency Identification (RFID) generates high-frequency electromagnetic field which causes electrons to move and subsequently power the chip [6]. RFID reader module is designed to create an electromagnetic field that is used to communicate with the RFID tags [6]. The RFID module function as a reader device that is triggered by an electromagnetic pulse from the reader and the tag will transmit data.

- Wire: Wire is a basic component that is used to transfer electricity between devices. A wire is used in the system to work as a connection from one component to another component to transmit electricity.

- Micro-push Button: Push-button is a simple switch mechanism that controls some aspect of a machine or a process [8]. It can be used to trigger a yes-no function in electronics when programmed correctly. The push-button is used in this system to allow electricity to flow when the button is pushed and the Arduino to carry out tasks that have been programmed.

- Power Supply: A power supply is an electrical device that is used to supply electric power to electrical devices and components. A 5v power supply is used in this system. It is used to supply electrical power to the system's components such as RFID module, Micro push-button, Stepper motor and Arduino Uno.

2.2 Types of Loop System

There are two (2) types of loop systems available in the environment of a control circuit that is open-loop system and closed-loop system [11]. The distinction between this loop is the open-loop does not have any feedback return to the input while the close loop system consists of at least one (1) feedback compared with the input to produce an error, e(t) flow into PID and plant [11]. The door lock system is using a close loop where the end position of the lock mechanism will be defined and act as a feedback before further action can be proceeded.

2.3 Schematic Diagram of PID Controller Door Lock System

Fig. 4 shows the schematic diagram for the PID controller door lock system.

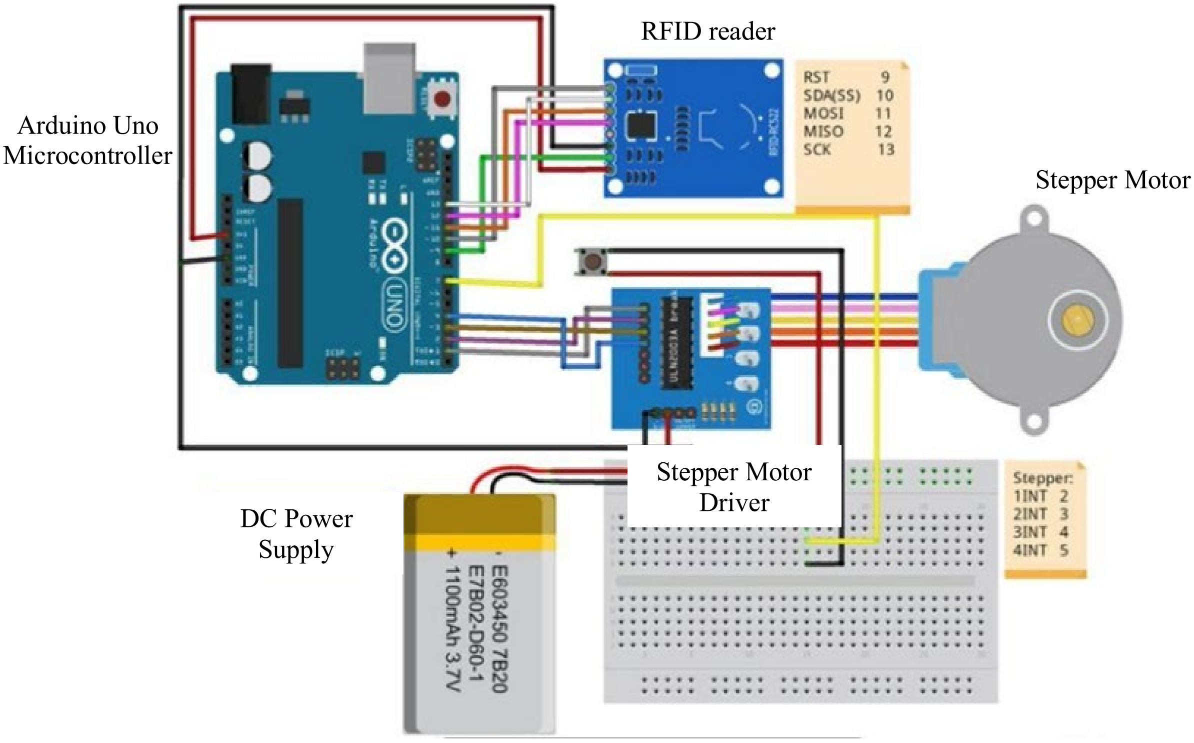


Fig. 4 - Schematic diagram of PID controller door lock system

Based on the schematic diagram above, the system consisted of several components that are Arduino Uno controller, RFID reader, stepper motor, stepper motor driver and lastly a DC power supply to power up the whole system throughout the process of locking and unlocking the door.

From the schematic diagram, it is needed to determine the block diagram for the whole system. The purpose of producing a block diagram is to help identify each component belong to which part of the controller system. For example, the control device will be the Arduino Uno controller while the stepper motor will be in the group of actuators. In Fig. 5, every component is classified according to its function in a block diagram model [11].

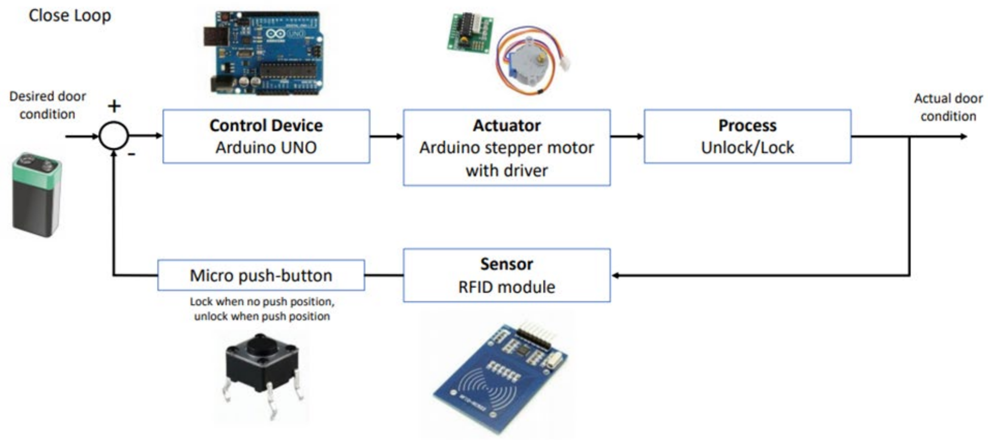


Fig. 5 - Block diagram model of a PID controller door lock system

From the block diagram above, we can identify the input and the desired output and all the components within the system which represent a control device or PID controller. The input for an electrical mechanical system usually is the electrical supply or power supplied by the battery and the output for our system will be the process of locking and unlocking the door. The sensor (RFID module) act as unity feedback through the process.

3. Mathematical Modelling and Transfer Function

The derivation of the mathematical modelling of a PID controller door lock system will consist of all the components stated in the previous chapter. The whole system will be divided into two (2) subsystems which are electrical and mechanical subsystems and all components will be derived based on their mathematics equation.

3.1 Transfer Function of PID Controller Door Lock System

For our system, we can divide it into two (2) subsystems which are electrical and mechanical subsystems. In the electrical subsystem, the components that will represent this subsystem will be the resistor, inductor including the power supply and the controller device. While the stepper motor, gear and push-button will be located in the mechanical subsystem. Fig. 6 shows the division of the subsystem by referring to our schematic diagram.

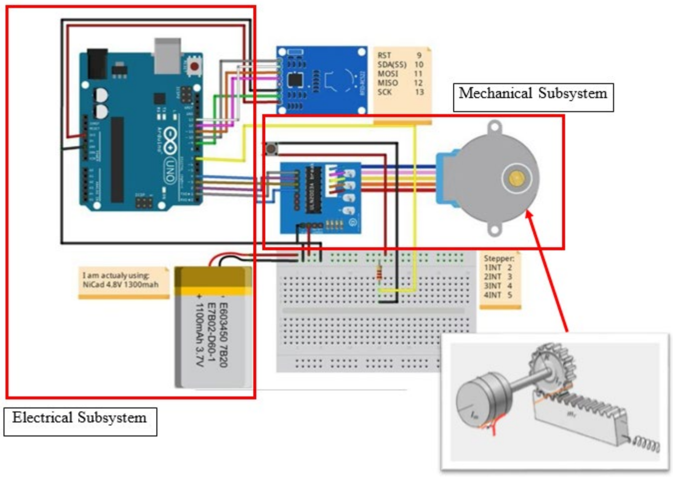


Fig. 6 - Division of subsystem for the PID door lock system

The transfer function of the system is written as:

$$\frac{x(s)}{V(s)} = \frac{(s^2+n^2)(k_t) - k_c\omega n - kr(s)\big((L_a s + R_a)(s^2+n^2)\big)}{mr\, s^2(L_a s + R_a)(s^2+n^2)} \tag{1}$$

3.2 Mathematical Modelling by Subsystem

3.2.1 Electrical Subsystem

For electrical, consist of the voltage input or the power supply, resistor, inductor and the electromagnetic field (emf) voltage. The equation for the electrical subsystem is stated as below;

$$V = L_a \frac{di_a}{dt} + R_a i_a + e_b \tag{2}$$

3.2.2 Mechanical Subsystem

For the mechanical subsystem, we will start from the stepper motor, transfer the torque of the motor to the rack and pinion system and lastly finish and the micro-push button where we assume the operation mechanism same as spring. For the stepper motor, we just take from the existing mathematic model and replace it with the emf constant inside the electrical subsystem. The equation for rack and pinion is as below,

$$T = F_t r \tag{3}$$

T	= Motor torque	(Nm)
F_t	= Tangential force	(N)
r	= Radius of pinion gear	(m)

From rack,

$$F_t = m\ddot{x} + kx \tag{4}$$

m	= Mass of the rack (lock bar)	(kg)

k	= Spring stiffness coefficient	(N/m)
x¨	= Acceleration of rack	(m/s2)
x	= Displacement of rack	(m)

Next, we combine both equations by the relationship of tangential force exist in both equation and the result is as below,

Combined pinion and rack equation,

$$T = mr\ddot{x} + krx \tag{5}$$

Convert into Power Transformation

$$e_b = k_c\omega \sin n\theta \tag{6}$$

Replace into an electrical subsystem

$$V = L_a\frac{di_a}{dt} + R_a i_a + k_c\omega \sin n\theta \tag{7}$$

$$T = K_t i_a \tag{8}$$

Replace into a mechanical subsystem

$$K_t i_a = mr\ddot{x} + krx \tag{9}$$

$$0 = mr\ddot{x} + krx - K_t i_a \tag{10}$$

3.3 Laplace Transform

From the mathematic equation obtain before, it is converted into Laplace transform (s domain) before obtaining the transfer function.

$$V = L_a\frac{di_a}{dt} + R_a i_a + k_c\omega \sin n\theta \tag{11}$$

$$0 = mr\ddot{x} + krx - K_t i_a \tag{12}$$

$$(L_a s + R_a)I_a(s) + k_c\omega\left(\frac{n}{s^2+n^2}\right)s = V(s) \tag{13}$$

$$mrs^2x(s) + kr(s) - k_t I_a(s) = 0 \tag{14}$$

Equation 11 into 10

$$(L_a s + R_a)\left[\frac{mr\, s^2x(s) + kr(s)}{k_t}\right] + \frac{k_c\omega n}{s^2+n^2}\,(s) = V(s)$$

$$(L_a s + R_a)[mr\, s^2x(s) + kr(s)](s^2+n^2) + k_c\omega n = V(s)(s^2+n^2)(k_t)$$

$$mr\, s^2x(s) + kr(s) = \frac{V(s)(s^2+n^2)(k_t) - k_c\omega n}{(L_a s + R_a)(s^2+n^2)} \tag{15}$$

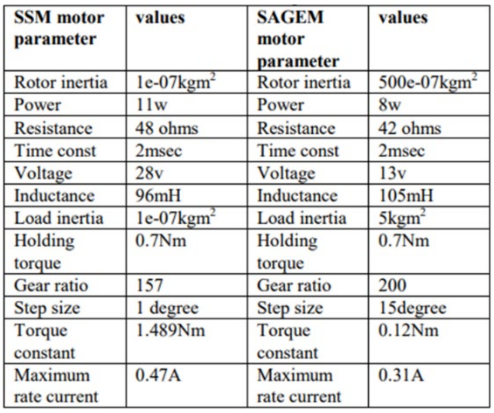
Finally, rearranged the equation by the relationship of output over input which we consider the output for our system is the displacement of micro-push button, x when lock and unlock and the voltage from the power supply will be our input. Relationship between displacement and voltage (output over input)

$$\frac{x(s)}{V(s)} = \frac{(s^2+n^2)(k_t) - k_c\omega n - kr(s)\big((L_a s + R_a)(s^2+n^2)\big)}{mr\, s^2(L_a s + R_a)(s^2+n^2)} \tag{16}$$

3.4 Specifications of Component and its Value

For all the values and properties of the component we use, we refer to the specification table provided by related journals and articles regarding the PID controller door lock system. By taking the standard value of an existing model, we can control the outcome and in the same way avoiding any unstable and failure for our transfer function. The motor specifications [12] is tabulated in Table 1.

Table 1 - Motor specifications [12]



SSM motor parameter	values	SAGEM motor parameter	values
Rotor inertia	1e-07kgm^2	Rotor inertia	500e-07kgm^2
Power	11w	Power	8w
Resistance	48 ohms	Resistance	42 ohms
Time const	2msec	Time const	2msec
Voltage	28v	Voltage	13v
Inductance	96mH	Inductance	105mH
Load inertia	1e-07kgm^2	Load inertia	5kgm^2
Holding torque	0.7Nm	Holding torque	0.7Nm
Gear ratio	157	Gear ratio	200
Step size	1 degree	Step size	15degree
Torque constant	1.489Nm	Torque constant	0.12Nm
Maximum rate current	0.47A	Maximum rate current	0.31A

N $_{pinion}$	20T
N $_{rack}$	11T
Mass (rack and lock bar)	±100gram
Angular velocity of motor	0.69 rad/s
Stiffness coefficient, k	0.81 N/m
Using calculator	
If No. of teeth	20T
then standard diameter for pinion gear	11.6mm
Radius = Diameter/2	5.8mm

Inserting all the value of the unknown, we get,

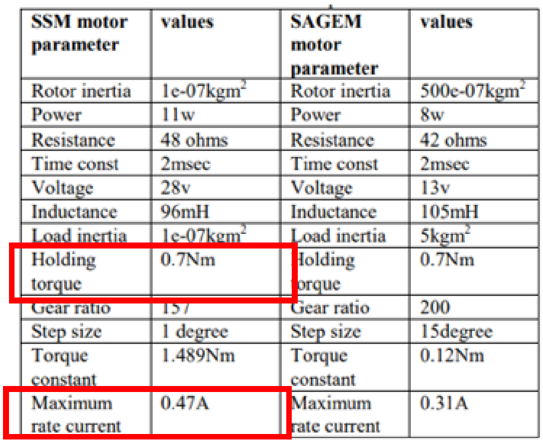
$$\frac{x(s)}{V(s)} = \frac{(s^2+20^2)(k_t) - 1.489(0.69)20 - (0.81)(5.8x10^{-3})(s)\big((96x10^{-3}s+48)(s^2+2}{(0.1)(5.8x10^{-3})\, s^2(96x10^{-3}s+48)(s^2+20^2)} \quad (17$$

To identify the stepper motor torque constant, kt [13]

$$T = K_t i_a \quad (18)$$

T	= Holding torque	(Nm)
K_t	= Stepper motor torque constant	(N)
i	= current	(m)

$$K_t = \frac{T}{i_a}$$



SSM motor parameter	values	SAGEM motor parameter	values
Rotor inertia	1e-07kgm^2	Rotor inertia	500e-07kgm^2
Power	11w	Power	8w
Resistance	48 ohms	Resistance	42 ohms
Time const	2msec	Time const	2msec
Voltage	28v	Voltage	13v
Inductance	96mH	Inductance	105mH
Load inertia	1e-07kgm^2	Load inertia	5kgm^2
Holding torque	0.7Nm	Holding torque	0.7Nm
Gear ratio	157	Gear ratio	200
Step size	1 degree	Step size	15degree
Torque constant	1.489Nm	Torque constant	0.12Nm
Maximum rate current	0.47A	Maximum rate current	0.31A

From the table of specifications above, we obtain the value for holding torque, T and current, i equal to 0.7Nm and 0.47A, respectively. Insert into the formula to obtain Kt.

$$K_t = \frac{0.7Nm}{0.47A}$$

$$K_t = 1.489Nm/A$$

Insert Kt into transfer function,

$$\frac{x(s)}{V(s)} = \frac{(s^2+20^2)(1.489) - 1.489(0.69)20 - (0.81)(5.8x10^{-3})(s)\left((96x10^{-3}s+48)(s^2+20^2)\right)}{(0.1)(5.8x10^{-3})\, s^2(96x10^{-3}s+48)(s^2+20^2)}$$

$$\frac{x(s)}{V(s)} = \frac{-0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518}{0.00005568s^2(s+500)(s^2+400)}$$

$$\frac{x(s)}{V(s)} = \frac{-0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518}{0.00005568s^5 + 0.02784s^4 + 0.022272s^3 + 11.136s^2}$$

Transfer function for RFID Door Lock will be as stated below: -

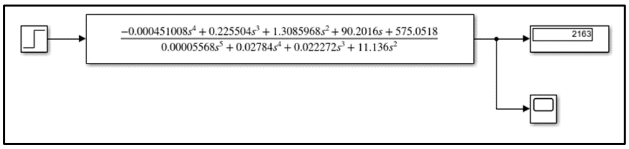
$$\frac{x(s)}{V(s)} = \frac{-0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518}{0.00005568s^5 + 0.02784s^4 + 0.022272s^3 + 11.136s^2} \qquad (19$$

3.5 Time Response Analysis

Time response is the response that consists of transient response and steady-state response. Step input is chosen as the test input for steady-state error analysis. All the analysis is made and validate using MatLab software and the graph produced by using Simulink software.

3.5.1 Open Loop and Close Loop System Analysis by Simulink Software

Figure 7 shows the block diagram of open and close loop produced using Simulink software and step input is the input for our system. The block diagram represents an open loop system where there is no feedback shown in Fig. 7(a) while the block diagram of a close loop is shown in Fig. 7(b) where a unity feedback is present in the block diagram.


−0.000451008s⁴ + 0.225504s³ + 1.3085968s² + 90.2016s + 575.0518
0.00005568s⁵ + 0.02784s⁴ + 0.022272s³ + 11.136s²
2163

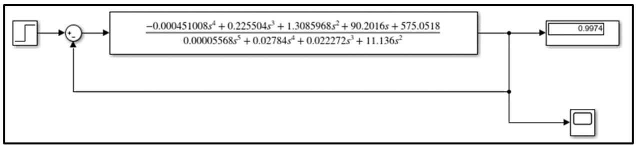
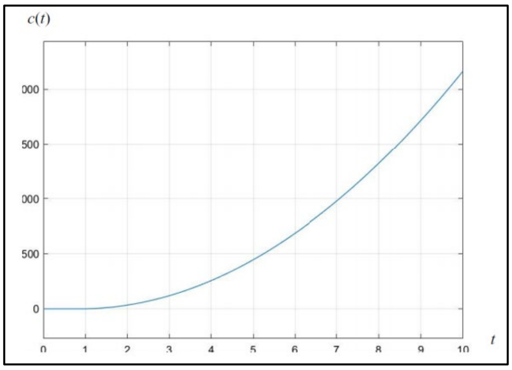

−0.000451008s⁴ + 0.225504s³ + 1.3085968s² + 90.2016s + 575.0518
0.00005568s⁵ + 0.02784s⁴ + 0.022272s³ + 11.136s²
0.9974

Fig. 7 (a) - Open loop system with step input **Fig. 7 (b) - Close loop system with step input**

Next, in Fig. 8(a) and 8(b) below is the comparison of response for the steady-state graph of the open-loop and the close loop system for the PID door lock system.



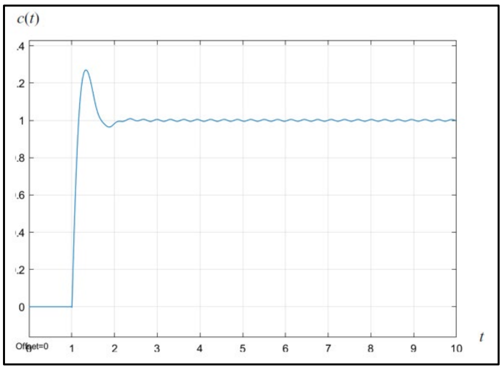


Fig. 8 (a) - Open loop response with step input **Fig. 8 (b) - Open loop response with step input**

3.5.2 Close Loop Response by Calculation

$$\frac{x(s)}{V(s)} = \frac{-0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518}{0.00005568s^5 + 0.02784s^4 + 0.022272s^3 + 11.136s^2}$$

Eliminate unity feedback, H(s) = 1

$$\frac{output}{input} = \frac{G(s)}{1 + G(s)(1)}$$

$$\frac{\frac{-0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518}{0.00005568s^5 + 0.02784s^4 + 0.022272s^3 + 11.136s^2}}{1 + \frac{-0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518}{0.00005568s^5 + 0.02784s^4 + 0.022272s^3 + 11.136s^2}}$$

$$\frac{-0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518}{0.00005568s^5 + 0.02784s^4 + 0.022272s^3 + 11.136s^2 + -0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518}$$

By using manual calculation, we obtain the transfer function for the close loop system by,

$$\frac{x(s)}{V(s)} = \frac{-0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518}{0.00005568s^5 + 0.02739s^4 + 0.2478s^3 + 12.4445s^2 + 90.2016s + 575.0518}$$

3.5.3 Close Loop Graph

From a step input response graph, we can observe the trend of the graph and calculate the value at a specific point such as peak time, rise time, delay time and other step response characteristics. By using MatLab software, we can obtain the value at all the point on the graph by inserting the script related to the step input command as we intended to analyze the response for step input. The coding script that we use is shown below.

```
clc;
clear all;
close all;
num= [-0.000451008 0.225504 1.3085968 90.2016 575.0518];
den= [0.00005568 0.02739 0.2478 12.4445 90.2016 575.0518];
sys= tf (num, den)
step(sys,0:0.01:10);
title ('step response of system');
[r,p,c]=residue(num,den)
```

The *clc;* command is use to clear all the coding available at the workspace before we insert the new script. *num* and *den* represent the numerator and the denominator of our transfer function. The insertion of the numerator and denominator is without inclusive of the s domain into the workspace.

Next, MatLab run the code we insert and generate a step response graph for the system with the particular numerator and denominator.

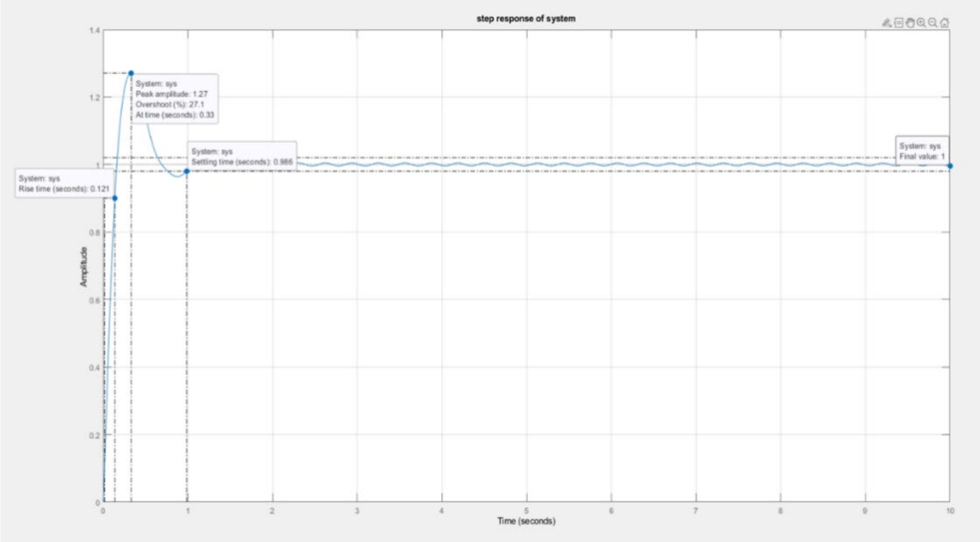
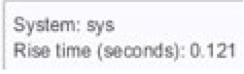


Fig. 9 - Step response of the system

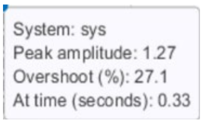
From the step response graph in Fig. 9, we can identify all the values we intended to measure from the coding we insert next generated by MatLab automatically.

From the graph, the value of a few important point is stated below:



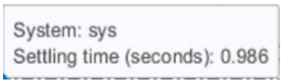
System: sys
Rise time (seconds): 0.121

The rise time only took 0.121 seconds from the initial operation which is when the system is idle or no action is produced on the sensor.



System: sys
Peak amplitude: 1.27
Overshoot (%): 27.1
At time (seconds): 0.33

The peak time takes 0.33 seconds and record at the amplitude of 1.27 with 27.1% overshoot produced calculated from the steady-state condition 1.



System: sys
Settling time (seconds): 0.986

The time taken for the system to meet the steady-state after passing the peak condition is equal to settling time which recorded at 0.986 seconds before beginning to change the transient state to steady-state condition.

3.6 Root Locus

The basic characteristic of the transient response of a closed-loop system is closely related to the location of the closed-loop poles [11]. The closed-loop poles are the roots of the characteristic equation. Finding the roots of the characteristic equation of degree higher than 3 is laborious and will need computer solution. (MATLAB provides a simple solution to this problem.) A simple method for finding the roots of the characteristic equation has been developed by W. R. Evans
and used extensively in control engineering to determine the system's stability. This method, called the root-locus method, is one in which the roots of the characteristic equation are plotted for all values of a system parameter.

Close loop Transfer Function

$$\frac{x(s)}{V(s)} = \frac{-0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518}{0.00005568s^5 + 0.02739s^4 + 0.2478s^3 + 12.4445s^2 + 90.2016s + 575.0518}$$

Characteristic equation is obtained by setting the denominator polynomial equal to zero

$$0.00005568s^5 + 0.02739s^4 + 0.2478s^3 + 12.4445s^2 + 90.2016s + 575.0518 = 0$$

3.6.1 Zeros and Poles

From the basic equation form, we can find the roots of the numerator (zero) and the root of the denominator (pole) by mathematics but we only focus on the value of poles.

The numerator polynomial has 4 zero(s) at

s = -6.32.
s = 507
s = -0.101 + 20j
s = -0.101 - 20j

The denominator polynomial yields n = 5 pole(s) at

s = -484,
s = -4.11 + 6.02j,
s = -4.11 - 6.02j,
s = -0.0196 + 20j.
s = -0.0196 - 20j.

Therefore, there are 5 branches to the locus.

By knowing the number of poles by referring to the denominator of our transfers function, we next can identify the number of angle intercepting the asymptotes located on the real axis of the s-plane. The formula of finding the asymptotes from our transfer function is stated below:

3.6.2 Point of Asymptotes (Centroid)

$$Asymptotes = \frac{sum\ of\ pole - sum\ of\ zero}{n - m}$$

$$Asymptotes = \frac{-492.2592 - 500.478}{1}$$

$$Asymptotes = (-992.73\ , 0)$$

Determine the number of asymptotes of the root loci

No of asymptotes = No of line going to infinity
No of asymptotes = 1

3.6.3 Angle of Asymptotes

Determining the angle of asymptote intercepting the real axis on the s-plane.

$$Angle\ of\ asymptotes = \theta = \frac{(2k+1)}{n-m}(180)$$

Where;

n = number of pole
m = number of zero
k = 0, …, n - m = 1

Angle of asymptotes when k = 0,

$$\theta_1 = \frac{(2(0)+1)}{1}(180°)$$
$$\theta_1 = 180°$$

Angle of asymptotes when k = 1,

$$\theta_2 = \frac{(2(1)+1)}{1}(180°)$$
$$\theta_2 = 540°$$

Angle of asymptotes when k = 2,

$$\theta_3 = \frac{(2(2)+1)}{1}(180°)$$
$$\theta_3 = 900°$$

Angle of asymptotes when k = 3,

$$\theta_4 = \frac{(2(3)+1)}{1}(180°)$$
$$\theta_4 = 1260°$$

Angle of asymptotes when k = 4,

$$\theta_5 = \frac{(2(4)+1)}{1}(180°)$$
$$\theta_5 = 1620°$$

The characteristic equation (i.e., the denominator of the closed loop transfers function) is

$$1 + K.G(s).H(s) = 0\ or\ 1 + K.\frac{N(s)}{D(s)} = 0$$

Where;

N(s) = Numerator polynomial
D(s) = Denominator polynomial

$$\boxed{(1)D(s) + K.N(s) = 0}$$

$$0.00005568s^5 + 0.02739s^4 + 0.2478s^3 + 12.4445s^2 + 90.2016s + 575.0518 + K \rightarrow (-0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518)$$

Next we need to consider how many line will go to the infinity,

$$n - m = 5 - 4 = 1$$

Where;

n = number of pole
m = number of zero

So, one (1) line will go to the infinity.

3.6.4 Routh Stability Criterion

$$\frac{x(s)}{V(s)} = \frac{-0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518}{0.00005568s^5 + 0.02739s^4 + 0.2478s^3 + 12.4445s^2 + 90.2016s + 575.0518}$$

Taking the denominator of the transfer function,

$$0.00005568s^5 + 0.02739s^4 + 0.2478s^3 + 12.4445s^2 + 90.2016s + 575.0518 = 0$$

s^5	0.00005568	0.2478	90.2016
s^4	0.02739	12.4445	575.0518
s^3	$= \frac{(0.02739 \times 0.2478) - (0.00005568 \times 12.4445)}{0.02739}$ $= 0.22250$	$= \frac{(0.02739 \times 90.2016) - (0.00005568 \times 575.0518)}{0.02739}$ $= 89.033$	0
s^2	$= \frac{(0.2225 \times 12.4445) - (0.02739 \times 89.033)}{0.22250}$ $=1.484$	$= \frac{(0.2225 \times 575.05) - (0)}{0.2225}$ $= 575.05$	0
s^1	$= \frac{(1.484 \times 89.033) - (0.2225 \times 575.05)}{1.484}$ $= 2.814$	0	
s^0	$= 575.05$		

$$1.484s^2 + 575.05 = 0$$

$$s^2 = \frac{-575.05}{1.484}$$

$$s = \sqrt{(-575.05)/1.484}$$

$$s = \sqrt{-1}\,\sqrt{387.5}$$

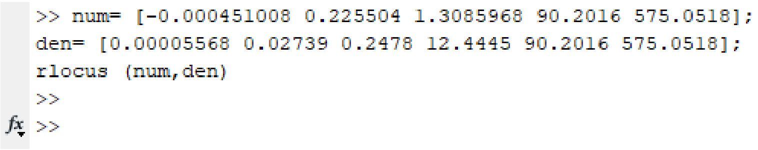
$$s = \pm 19.69i$$

The value of s which is equal to ±19.69i is the interception point of the root locus graph which intercept the imaginary axis on positive and negative side.

3.6.5 Root Locus using MatLab software

Fig. 10 shows the root locus command that has been inserted into MATLAB workspace to generate the root locus graph. In the input for the numerator and the denominator is taken from the transfer function obtained from the previous part. Below is the transfer function that we obtain from our mechanical and electrical mechanism from the PID door lock system.

$$\frac{x(s)}{V(s)} = \frac{-0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518}{0.00005568s^5 + 0.02784s^4 + 0.022272s^3 + 11.136s^2}$$



```
>> num= [-0.000451008 0.225504 1.3085968 90.2016 575.0518];
den= [0.00005568 0.02739 0.2478 12.4445 90.2016 575.0518];
rlocus (num,den)
>>
>>
```

Fig. 10 - Root locus command in MatLab workspace

The ‘num’ command indicates the input for numerator of the transfer function while the ‘den’ command represents the denominator of the transfer function. In the MatLab workspace, to insert the polynomial, we don’t have to include the s domain and needed to leave it in numerical format. Fig. 11 shows the root locus graph generated by the MatLab software.

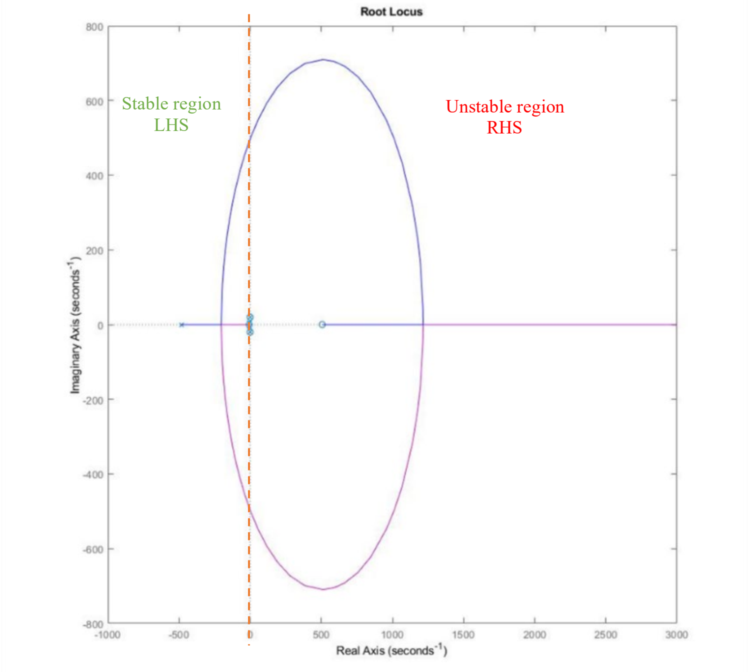


Fig. 11 - Graph of imaginary against real axis for root locus generated by MatLab

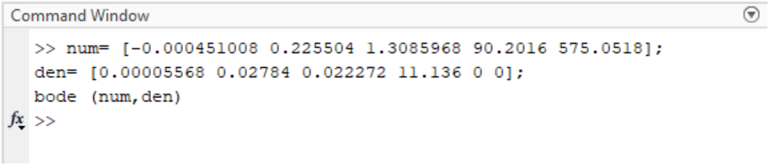
The root locus graph shows that the condition for our system is unstable because the root locus is used to plot on the right-hand side of the s-plane. This condition was recorded before we include the PID controller into our system where it is use to stabilize the whole system by using a try and error tuning in Simulink.

3.7 Bode Plot

We proceed the Bode plot by using MatLab software and entering the coding related to Bode. Fig. 12 shows the command and coding we use to generate Bode diagram. Rewrite the transfer function in proper form.

$$\frac{x(s)}{V(s)}=\frac{-0.000451008s^4+0.225504s^3+1.3085968s^2+90.2016s+575.0518}{0.00005568s^5+0.02784s^4+0.022272s^3+11.136s^2}$$

$$\frac{x(s)}{V(s)}=\frac{-0.00045s^4+0.23s^3+1.31s^2+90.20s+575.05}{0.000056s^5+0.028s^4+0.022s^3+11.14s^2}$$



```
>> num= [-0.000451008 0.225504 1.3085968 90.2016 575.0518];
den= [0.00005568 0.02784 0.022272 11.136 0 0];
bode (num,den)
>>
```

Fig 12 - Coding for Bode diagram in MatLab

Next, after inserting the coding without mistake or error, we click enter and wait for the software to generate magnitude and phase graph regarding to the Bode diagram. Fig. 13 shows the Bode diagram generated by MatLab.

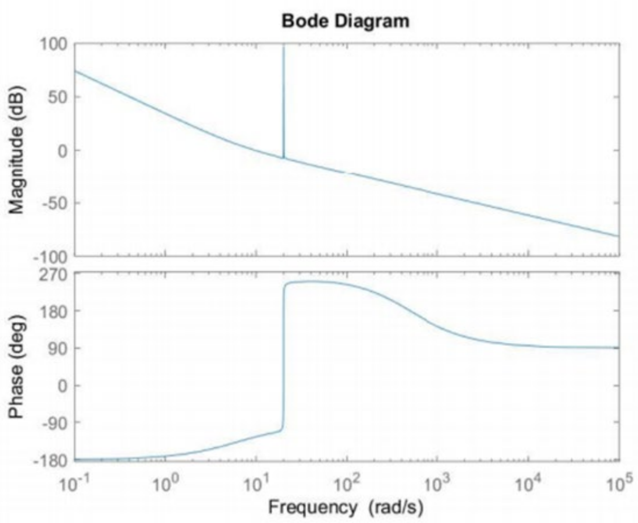


Fig. 13 - Bode diagram of phase and magnitude against frequency for the system [1]

For the graph generated for magnitude, the overall slope value is observed to be -20dB/ decade. The slope or gradient of the magnitude graph get a negative value as we can observe from the visual above the graph is has a downward trend. The overall magnitude graph is gained from the sum of all constituents calculated decade by decade using MatLab software by inserting related commands into the coding script.

For the phase graph, the initial phase is located at -180-degrees from the sum of the first decade. Then started to go up to -90-degree phase as our transfer function containing real poles obtained from the denominator of a proper form transfer function.

Phase margin: The phase margin is the amount of additional phase lag at the gain crossover frequency required to bring the system to the verge of instability. Fig. 13 illustrates the phase margin of a stable system in Bode diagram. For the example of a stable system and unstable system of phase and gain margin, it is shown in Fig. 14.

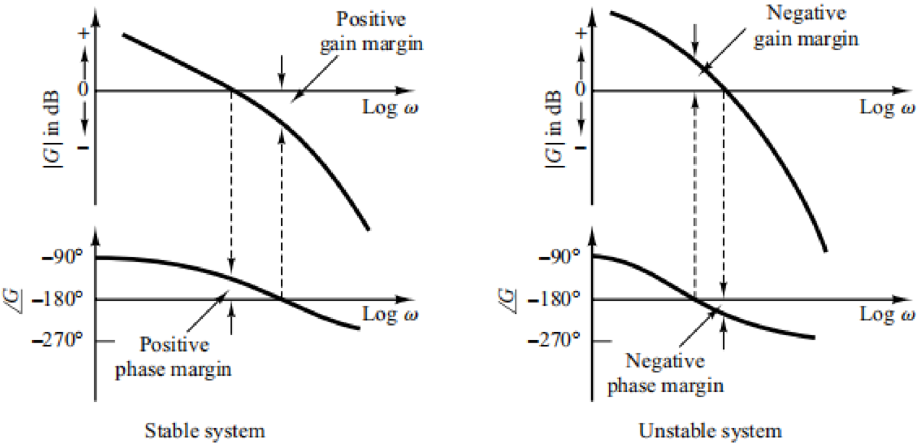


Fig. 14 - Comparison between a stable and unstable system in Bode diagram [1]

By referring to above condition, we can define that from our Bode diagram is in stable system where the phase margin is in positive terms. The positive phase margin is shown in Fig. 15 below.

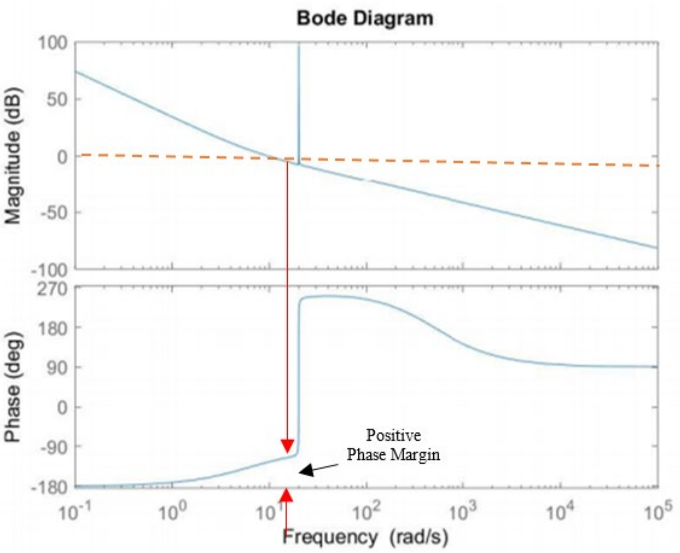


Fig. 15 - Positive phase margin on a stable system

4. PID Controller

Another one (1) component into our block diagram is to increase the rate of time to obtain a steady state and reduce the amount of overshoot as possible. First, we will analyze by observing the effect of proportional control (P) in our system. Then, we will analyze the effect of proportional plus derivative control (PD) and lastly the effect of proportional plus integral plus derivative control (PID). All the effect will be compared to the system response with and without using the PID to determine how much the PID affect the stability and effectiveness of the PID controller.

4.1 Try and Error Rules for Tuning

The process of selecting the controller parameters to meet given performance specifications is known as controller tuning. Try and error are the suggested rules for tuning PID controllers (meaning to set values Kp, Ti and Td) based on experimental step responses or based on the value of Kp that results in marginal stability when only proportional control action is used. The try and error method involves, which are briefly presented in the following, are useful when mathematical models of plants are not known.

The rule of try and error is for determining values of the proportional gain Kp, integral time Ti, and derivative time Td based on the transient response characteristics of a given plant. Such determination of the parameters of PID controllers or tuning of PID controllers can be made by engineers on-site by experiments on the plant.

4.2 Effect of P, PD and PID Controller to The System

Now we will compare the effect of the controller by tuning each controller using Simulink software to obtain the best parameter for the lowest overshoot and with the fastest time to get into steady-state condition.

4.2.1 Proportional Control (P)

4.2.1.1 Block Diagram with P Controller

To check the effect of proportional control, we need to include P(s) function into the close loop block diagram. Before tuning, we will observe the graph condition before inserting any value into the Proportional controller. The block diagram of RFID door lock system inclusive of P controller is shown in Fig. 16:

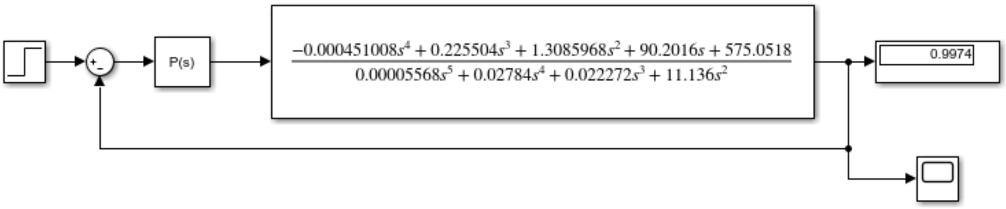
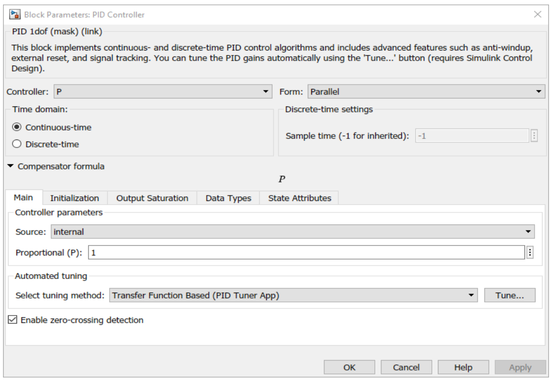

+
-
P(s)
$\frac{-0.000451008s^4 + 0.225504s^3 + 1.3085968s^2 + 90.2016s + 575.0518}{0.00005568s^5 + 0.02784s^4 + 0.022272s^3 + 11.136s^2}$
0.9974

Fig. 16 - Block diagram with P controller

4.2.1.2 Parameter Setting for P Controller

The parameter of the Proportional controller (P) is initially set to the value of one (1) before tuning into any value. It is best to start with a small value of P and gradually increase it. When we select the controller to 'P', the only parameter that we can change is the 'proportional (P)' value. We can also tune Proportional controller by using PID parameter by selecting the controller type to PID and next set the parameter of 'integral (I)' and 'Derivative (D)' to zero.

Fig. 17(a) and 17(b) show two (2) different settings of controller parameters that will produce the same result.


Block Parameters: PID Controller

PID 1dof (mask) (link)

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: P
Form: Parallel

Time domain:
● Continuous-time
○ Discrete-time

Discrete-time settings
Sample time (-1 for inherited): -1

Compensator formula

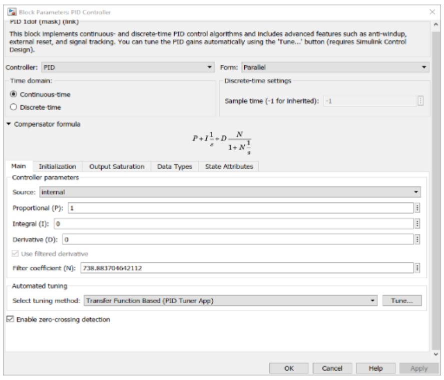
P

Controller parameters
Source: internal
Proportional (P): 1

Automated tuning
Select tuning method: Transfer Function Based (PID Tuner App)

☑ Enable zero-crossing detection

Fig. 17 (a) - Setting by P parameter


Block Parameters: PID Controller

PID 1dof (mask) (link)

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: PID
Form: Parallel

Time domain:
● Continuous-time
○ Discrete-time

Discrete-time settings
Sample time (-1 for inherited): -1

Compensator formula

$$P+I\frac{1}{s}+D\frac{N}{1+N\frac{1}{s}}$$

Controller parameters
Source: internal
Proportional (P): 1
Integral (I): 0
Derivative (D): 0
☑ Use filtered derivative
Filter coefficient (N): 738.883704642112

Automated tuning
Select tuning method: Transfer Function Based (PID Tuner App)

☑ Enable zero-crossing detection

Fig. 17 (b) - Setting by PID parameter

4.2.1.3 Result Comparison With and Without P Controller

Fig. 18(a) below show the step response graph before inserting P controller into the block diagram while Fig. 18(b) is the step response graph after inserting P controller and tune using PID tuner in MatLab software.

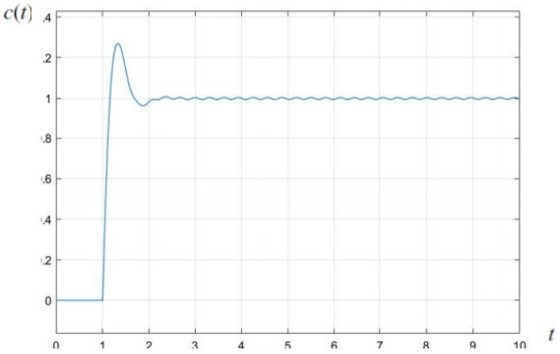


Fig. 18 - (a) Step response graph without controller

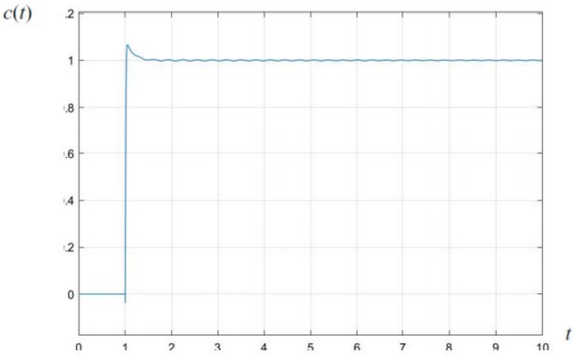
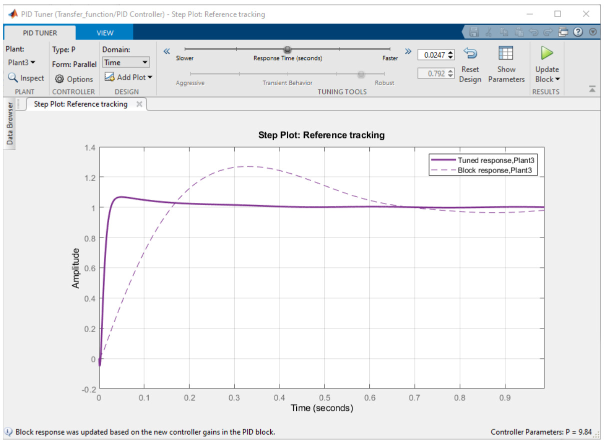


Fig. 18 - (b) Final step response result after tuning

Now we will start to tune by try and error the P controller to a specific parameter to reduce the amount of overshoot and reducing the amount of time to achieve steady-state condition.


PID Tuner (Transfer_function/PID Controller) - Step Plot: Reference tracking

Plant: Plant3

Type: P

Form: Parallel

Domain: Time

Response Time (seconds): 0.0247

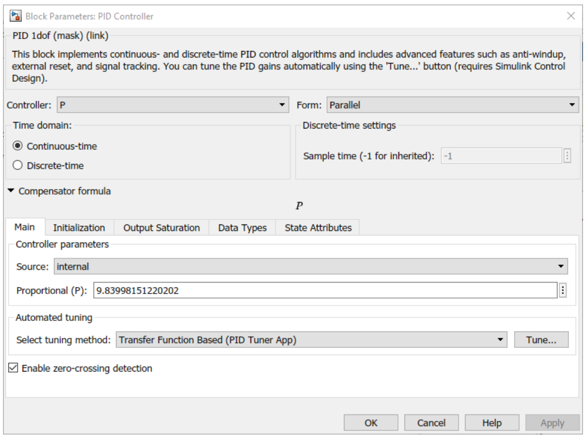
Transient Behavior: 0.792

Block response was updated based on the new controller gains in the PID block.

Controller Parameters: P = 9.84

Fig. 19 - Comparison between tuned and untune trend of the step response

After complete tuning process, we obtain the parameter of P that can be used to adjust the step response graph as shown in Fig. 19. The value of Proportional parameter when using P controller is approximately 9.8399815 to obtain the best result. Fig. 20 shows the parameter we obtain when tuning.


Block Parameters: PID Controller

PID 1dof (mask) (link)

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: P

Form: Parallel

Time domain:

● Continuous-time

○ Discrete-time

Discrete-time settings

Sample time (-1 for inherited): -1

Compensator formula

P

Controller parameters

Source: internal

Proportional (P): 9.83998151220202

Automated tuning

Select tuning method: Transfer Function Based (PID Tuner App)

☑ Enable zero-crossing detection

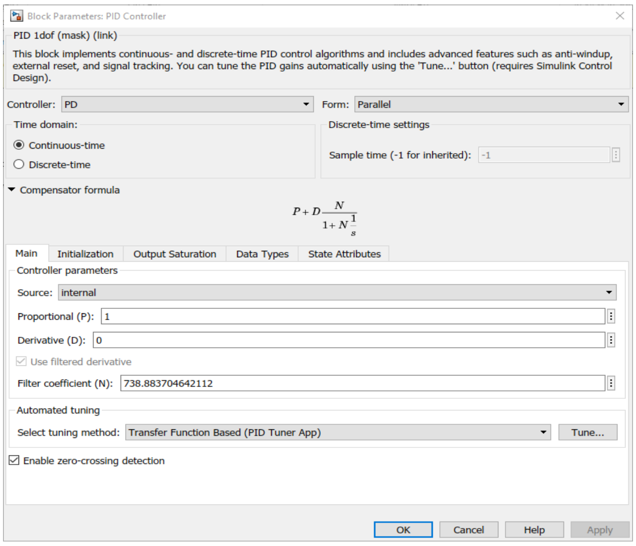
Fig. 20 - Tuned parameter for P controller

4.2.2 Proportional Plus Derivative Control (PD)

4.2.2.1 Parameter Setting for PD Controller

The parameter of the Proportional controller (P) is initially set to the value of one (1) before tuning into any value. It is best to start with a small value of P and gradually increase it. When we select the controller to 'P', the only parameter that we can change is the 'proportional (P)' value. We can also tune Proportional controller by using PID parameter by selecting the controller type to PID and next set the parameter of 'integral (I)' and 'Derivative (D)' to zero.

Fig. 21 shows the initial setting for PD controller parameter where proportional initial state as one and derivative set to zero.



Block Parameters: PID Controller

PID 1dof (mask) (link)

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: PD Form: Parallel

Time domain:

● Continuous-time

○ Discrete-time

Discrete-time settings

Sample time (-1 for inherited): -1

Compensator formula

$$P+D\frac{N}{1+N\frac{1}{s}}$$

Controller parameters

Source: internal

Proportional (P): 1

Derivative (D): 0

☑ Use filtered derivative

Filter coefficient (N): 738.883704642112

Automated tuning

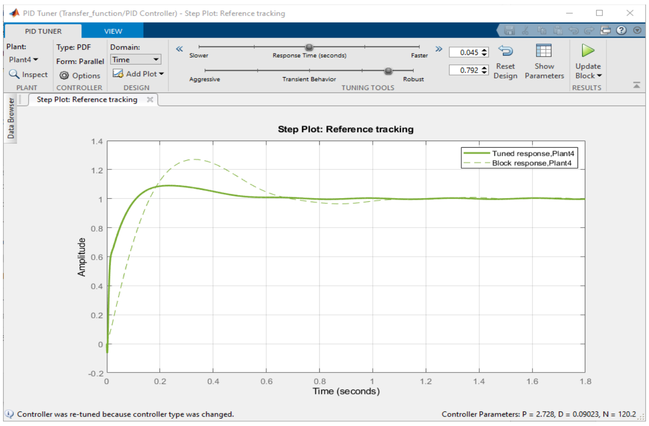
Select tuning method: Transfer Function Based (PID Tuner App)

☑ Enable zero-crossing detection

Fig. 21 - Setting of PD parameter at initial

4.2.2.2 Result Comparison with and without PD Controller

Now we will start to tune the PD controller to a specific parameter in order to reduce the amount of overshoot and reduce the amount of time to achieve steady-state condition as shown in Fig 22.



PID Tuner (Transfer_function/PID Controller) - Step Plot: Reference tracking

Plant: Plant4

Type: PDF

Form: Parallel

Domain: Time

Response Time (seconds): 0.045

Transient Behavior: 0.792

Controller was re-tuned because controller type was changed.

Controller Parameters: P = 2.728, D = 0.09023, N = 120.2

Fig. 22 - Comparison between tuned and untune trend using PD controller

After complete tuning process, we obtain the parameter of PD that can be used to adjust the step response graph. The value of Proportional parameter when using PD controller is approximately 2.728204 while for derivative parameter is at 0.09022886 to obtain the best result. Fig. 23 is the step response diagram after applying PD controller after complete tuning.

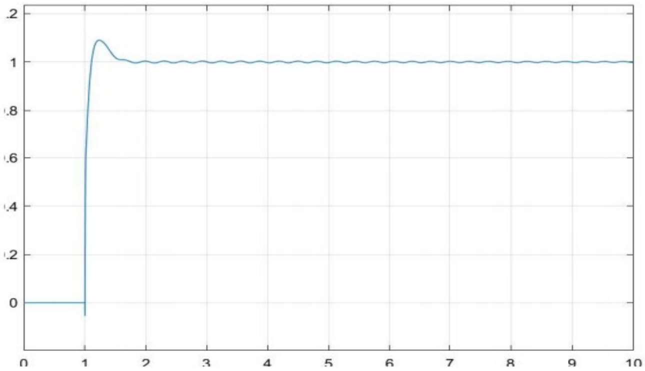


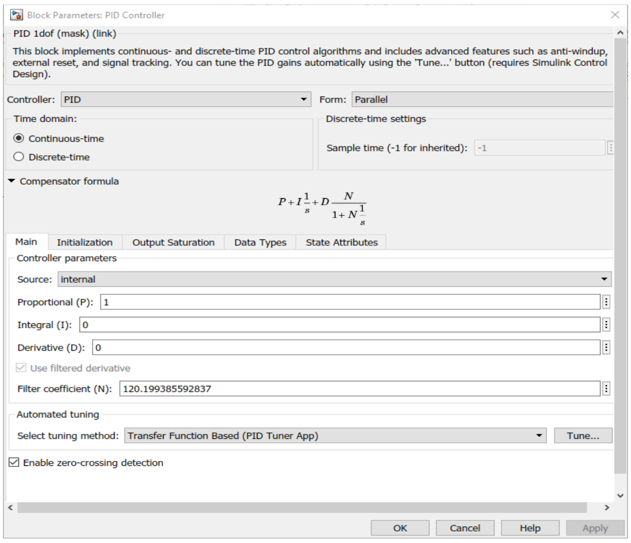
Fig. 23 - Final step response result after tuning using PD controller

4.2.3 Proportional Plus Integral Plus Derivative Control (PID)

4.2.3.2 Parameter Setting for PID Controller

The parameter of the Proportional controller (P) is initially set to the value of one (1) before tuning into any value. It is best to start with a small value of P and gradually increase it. When we select the controller to ‘P’, the only parameter that we can change is the ‘proportional (P)’ value. We can also tune Proportional controller by using PID parameter by selecting the controller type to PID and next set the parameter of ‘integral (I)’ and ‘Derivative (D)’ to zero (0).

Fig. 24 shows the initial setting for PD controller parameter where proportional initial state as one and derivative set to zero (0).



Block Parameters: PID Controller

PID 1dof (mask) (link)

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: PID

Form: Parallel

Time domain:

● Continuous-time

○ Discrete-time

Discrete-time settings

Sample time (-1 for inherited): -1

Compensator formula

$$P+I\frac{1}{s}+D\frac{N}{1+N\frac{1}{s}}$$

Controller parameters

Source: internal

Proportional (P): 1

Integral (I): 0

Derivative (D): 0

☑ Use filtered derivative

Filter coefficient (N): 120.199385592837

Automated tuning

Select tuning method: Transfer Function Based (PID Tuner App)

☑ Enable zero-crossing detection

Fig. 24 - Setting of PID parameter at initial

4.2.3.3 Result Comparison with and without PID Controller

Fig. 25(a) and Fig. 25(b) show the step response graph before inserting PID controller into the block diagram.

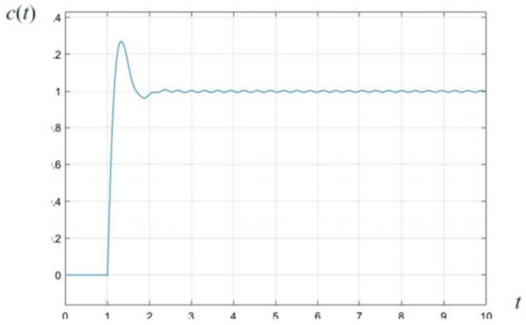


Fig. 25 (a) - Step response graph without PID controller

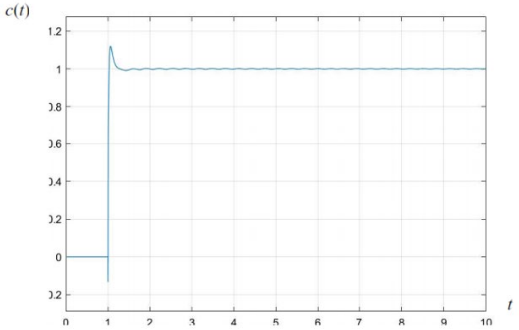
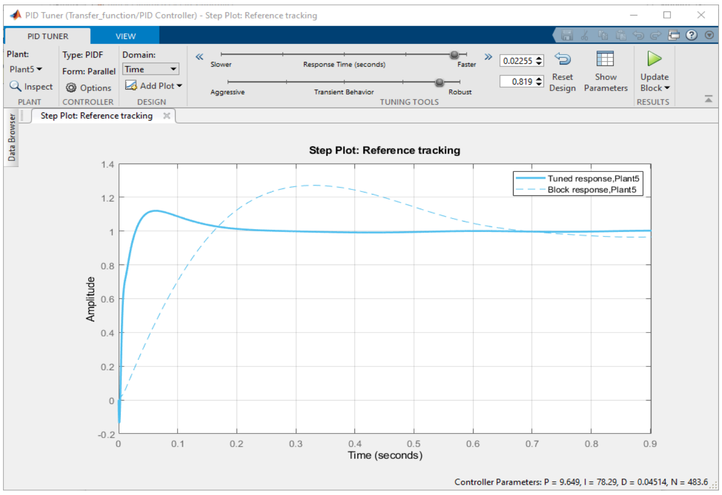


Fig. 25 (a) - Final step response result after tuning

Now we will start to tune the PID controller to a specific parameter to reduce the amount of overshoot and reduce the amount of time to achieve steady-state condition.


PID Tuner (Transfer_function/PID Controller) - Step Plot: Reference tracking

Plant: Plant5

Type: PIDF

Form: Parallel

Domain: Time

Response Time (seconds): 0.02255

Transient Behavior: 0.819

Controller Parameters: P = 9.649, I = 78.29, D = 0.04514, N = 483.6

Fig. 4.11 - Comparison between tuned and untune trend

After the complete tuning process, we obtain the parameter of PID that can be used to adjust the step response graph. The value of Proportional parameter when using PID controller is approximately 9.6488838 while for integral parameter is at 78.2859329 and lastly the derivative parameter is 0.045137596 to obtain the best result.

The comparison of the step response graph without using any controller, with PD controller and lastly using PID controller is tabulated in Table 2. The result from the application of PID controller resulted in the lowest overshoot percentage which gave a most efficient outcome compared to another controller variable.

Table 2 - Comparison of result under 3 controllers conditions

	T_s (s)	T_p (s)	Overshoot %
No controller	2.00	1.50	25.00
PD controller	0.60	0.02	9.167
PID controller	0.20	0.05	8.333

5. Conclusion and Recommendation

Firstly, to conclude this project, all the objectives for this project has been achieved. The objective to produce a more efficient way of locking and unlocking the door are successful after being able to derive the mathematical modelling of every component, next identify the output and input of the PID controller door lock system. From the derivation of mathematical modelling of every subsystem in the door lock system, it is known that the fundamental law governing mechanical systems is Newton's second law. The application of this law to various mechanical systems and derive transfer function models. Next, result of deriving the electrical subsystem, it is known the basic law governing electrical circuits are Kirchoff's laws. In this report, transfer-function models is obtained for various type of electrical component such as resistor, inductance and power supply.

After obtaining complete mathematical modelling for the PID controller door lock system,, it has covered the second objective for this project is to produce a transfer function that covers all of the components inside the system. After that, proceeding to a various method for analysis of the system performance. In practice, the input signal to a control system is not known ahead of time but is random in nature, and the instantaneous input cannot be expressed analytically. Only in some special cases is the input signal known in advance and expressible analytically or by curves, such as in the case of the automatic control of cutting tools. Many design criteria are based on the response to such test signals or on the response of systems to changes in initial conditions (without any test signals). The use of test signals can be justified because of a correlation existing between the response characteristics of a system to a typical test input signal and the capability of the system to cope with actual input signals.

Lastly, for the third objective for this project, the result has successfully applied the PID controller into the system so that the output can achieved more stable and meet the desired value. The usage of a PID controller into our system is to reduce the overshoot value in the same way, provide quicker time to achieve steady-state conditions. The advantage of PID controller is its feasibility and ease to be implemented. The PID gains can be designed based upon the system parameters if they can be achieved or estimated precisely. Moreover, the PID gain can be designed just based on the system tracking error and treats the system to be "blackbox" if the system parameters are unknown. However, PID controller generally has to balance all three-gains impact to the whole system and may compromise the transient response, such as settling time, overshoots, oscillations. If the system parameters cannot be precisely estimated or achieved, the designed PID gains may not resist the uncertainties and disturbances, and thus present low robustness. Even though the PID gains can be well-designed, the PID controller still has low robust ability compared with the robust controller when the system encounters multiple challenges from the operating environment of the system, such as temperature, weather, power surge, and so on.

The recommendation on improving the result is the derivation for the transfer function needs to be more detailed regarding each subsystem to get a more stable system when plotting manually and automatically using MatLab software. The ability to the simplified complex conjugate equations is crucial to ease the process of transferring value into proper form and next able to generate Bode and Nyquist plot manually. Last but not least, more knowledge on handling MatLab software was needed to acquire more detailed graphs and information when inserting a specific coding.

The recommendation for this project is to improve the learning outcome and provide a more quality project is that the syllabus cover for the control engineering course need to be more complex and feeds with a more real-life engineering-based problem. Instead of using several presentation slides for the learning process, focusing on using a fully engineering book as reference such as Modern Control Engineering written by Katsuhika Ogata through completing the research as obviously all the content inside this book covers a total syllabus for this study. The learning process will be more standardized and easier to control the outcome for the study. A mix of multiple lecture notes from many sources can lead to confusion and provide difficulty when we want to do research related to a specific field.

References

[1] Ogata, K., 2010. Modern Control Engineering. Prentice Hall

[2] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, “Feedback Control of Dynamic Systems Sixth Edition,” 2010.

[3] A.Bansal and V. Sharma, “Design and Analysis of Robust H-infinity Controller,” vol. 3, no. 2, 2013

[4] K Verma, G. and Tripathi, P., 2010. 'A Digital Security System with Door Lock System Using RFID Technology', International Journal of Computer Applications, vol. 5, no. 11, pp. 6-8

[5] Hlaing, N. and Lwin, S., 2019. 'Electronic Door Lock using RFID and Password Based on Arduino', International Journal of Trend in Scientific Research and Development (IJTSRD), vol. 3, no. 3, pp. 799-802

[6] Weinstein, R., 2005. 'RFID: A Technical Overview and Its Application to the Enterprise', New Technology, vol. 5, no. 5, pp. 28-33

[7] E. Sarma,, S., A. Weis, S. and W. Engels, D., 2002. 'RFID Systems and Security and Privacy Implications', Auto-ID Center Massachusetts Institute of Technology, vol. 1, no. 2, pp. 1-2

[8] Burns, B., Dahlinger, D., Guenther, B. and Johnson, T., 2018. 'RFID Controlled Door Lock', Mechatronics Final Project, Engineering Science Department, vol. 5, no. 9, pp. 1-5

[9] EPC-RFID. 2021. What is RFID? - EPC-RFIDEPC-RFID. [ONLINE] Available at: https://www.epc-rfid.info/rfid. [Accessed 4 January 2021]

[10] Thirumaleshwa, N. and S. Meenatchi Sundaram, H., 2016. 'Modeling Stepper Motor Control System with Micro-Stepping Excitation Mode', National Conference on Advances in Mechanical Engineering Science (NCAMES-2016), vol. 1, no. 1, pp. 136-137

[11] Circuit Globe. 2021. Difference Between Open Loop & Closed Loop System (with Comparison Chart) - Circuit Globe. [ONLINE] Available at: https://circuitglobe.com/difference-between-open-loop-and-closed-loop-system.html. [Accessed 16 January 2021]

[12] Chris Vavra. 2021. Control Engineering | Stepper motor torque basics. [ONLINE] Available at: https://www.controleng.com/articles/stepper-motor-torque-basics/#:~:text=Holding%20torque%20is%20a%20measurement,applied%20to%20the%20stator%20windings.&text=In%20most%20applications%2C%20electronic%20drivers%20control%20stepper%20motors.. [Accessed 16 January 2021]

[13] Electrical Engineering Stack Exchange. 2021. How to calculate and measure motor constant for stepper motor? - Electrical Engineering Stack Exchange. [ONLINE] Available at: https://electronics.stackexchange.com/questions/383242/how-to-calculate-and-measure-motor-constant-for-stepper-motor. [Accessed 18 January 2021