DEVELOPMENT AND CONTROLLER DESIGN OF LOW-COST AIR LEVITATION SYSTEM

ANG ZE GUAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DEVELOPMENT AND CONTROLLER DESIGN OF LOW-COST AIR LEVITATION SYSTEM

ANG ZE GUAN

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronic Engineering with Honours

Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka

2022

DECLARATION

I declare that "Development And Controller Design Of Low-Cost Air Levitation System" is the result of my own work except for quotes as cited in the references.

Signature : ....

Author : ANG ZE GUAN

Date : 3 JANUARY 2022

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

Signature : ....

Supervisor Name : DR SHARATUL IZAH BINTI SAMSUDIN

Date : 4 JANUARY 2022

DEDICATION

This thesis is dedicated to my dear family members.

ABSTRACT

The air levitation system is a non-linear and unstable system. The goal of this work is to develop a low-cost air levitation system. The PID controller is designed and applied to the physically developed system to increase the stability of the system. The platforms that are used to develop and simulate the system consist of the Arduino IDE platform, MATLAB platform, and Simulink platform. The developed system is connected to the Simulink platform for real-time simulation. The transfer functions of the systems are estimated by using system identification technique in MATLAB platform. The PID controller is then designed while the step response is analyzed. The values of the percentage overshoot, rise time, steady-state error, and settling time are investigated. The stability of the air levitation system is increased and it was found that the percentage overshoot, settling time, and steady-state error can be greatly reduced with the application of the PID controller to the system.

ABSTRAK

Sistem pengangkatan udara adalah sistem yang tidak linear dan tidak stabil. Tujuan kerja ini adalah untuk membina sistem pengangkatan udara dengan kos yang rendah. Pengawal PID direkabentuk dan digunakan pada sistem yang telah dibinakan dengan tujuan untuk meningkatkan kestabilan sistem tersebut. Platform yang digunakan untuk membangunkan dan mensimulasikan sistem adalah platform Arduino IDE, platform MATLAB, dan platform Simulink. Sistem yang dibangunkan akan disambungkan ke platform Simulink untuk simulasi masa nyata. Rangkap pindah sistem dianggarkan dengan menggunakan teknik pengenalan sistem dalam platform MATLAB. Pengawal PID kemudian direkabentuk sementara sambutan sistem dianalisis. Parameter peratus lajakan, masa naik, masa penetapan dan ralat keadaan mantap disiasat. Kestabilan sistem pengangkatan udara dapat ditingkatkan dan peratus lajakan, masa penetapan dan ralat keadaan mantap berjaya dikurangkan.

ACKNOWLEDGEMENTS

First of all, I would like to thank all of my family members, who always provide financial support and spiritual support during the project development process and my study. Throughout the final year project, I have received a lot of help from my supervisor, friends, and panels. Without their help and support, this study would never be accomplished.

Furthermore, I would like to express my sincere gratitude to my supervisor, Dr. Sharatul Izah Binti Samudin, who has given me valuable guidance in developing this final year project especially in using the MATLAB platform and thesis writing. In addition, my sincere appreciation goes to my panels too, Dr. Khairun Nisa Binti Khamil and PM. Dr. Soo Yew Guan, who gave me a lot of advice and ideas for my final year project.

Moreover, I would also like to thank all the lecturers of the Faculty of Electronics and Computer Engineering, who helped me to develop the fundamental and essential academic competence within my four years of study.

TABLE OF CONTENTS

CHAPTER 3 METHODOLOGY

CHAPTER 4 RESULTS AND DISCUSSION

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

D Derivative.

DC Direct Current.

FAL Floating Air Levitation.

FOPDT First Order Process With Time Delay.

GUI Graphical User Interface.

I Integral.

IAE Integral Absolute Error.

IDE Integrated Development Environment.

IMC Internal Model Control.

ITAE Integral Time Multiple Absolute Errors.

KC Controller Constant.

KD Derivative term.

KI Integral term.

KP Proportional term.

KU Ultimate Gain.

LED Light Emitting Diode.

P Proportional.

PC Personal Computer.

PCB Printed Circuit Board.

PD Proportional Derivative.

PI Proportional Integral.

PID Proportional Integral Derivative.

PIR Passive Infrared Sensor.

PLC Programmable Logic Controller.

PU Ultimate Period Of Oscillation.

PWM Pulse Width Modulation.

QDR Quater Decay Ratio.

RIP Rotary Inverted Pendulum.

SMC Sliding Mode Controller.

TD Derivative Time Constant.

TI Integrator Time Constant.

US United States.

VRL Virtual and Remote Labs.

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Project Background

Control system engineering plays an important role in human life and industry. A Control system can be defined as a system that can perform in the desired manner. The control system can reduce human interference in the daily work of households or industries by reducing human error [1]. The control system can be classified into two types which are open-loop system and closed-loop system. An open-loop system is a system that does not has a feedback path to the controller, while the closed-loop system does have the feedback path to the controller so that the final value is compared with the reference value assigned by the user and the error is reduced.

The Proportional Integral Derivative (PID) controller is widely used in the industry for controlling the process due to its simplicity. It is because, in industry, all of the machines must perform the exact output that is expected by the user else it will cause some troubles to the company or user. The PID is referred to the three terms which are the proportional term, integral term, and derivative term. The block diagram of the PID controller is shown in Figure 1.1 [2]. PID controller is used as a pneumatic, hydraulic or mechanical controller or had a simple interface for manual tuning of the controller. PID is simply an equation that the controller uses to evaluate the controlled variables [3].

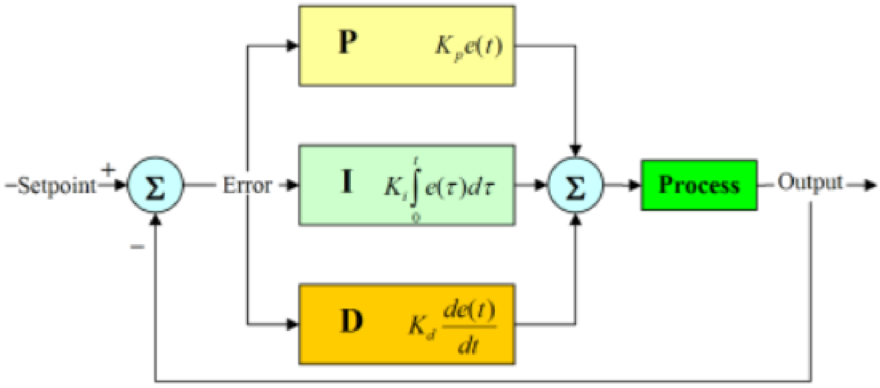


Figure 1.1: PID controller.

The Proportional term (KP) is the term that decreases the rise time of the system but the number of orders of the system will remain. The output of the system will be directly proportional to the input applied. The increase of the proportional gain will cause the error of the system to decrease. The Integral term (KI) or integral gain is used to get rid of the steady-state error of the system and the order of the system is increased by 1. The Ki will decrease the rise time and the oscillation of the system will be increased. While the Derivative term (KD) or the derivative gain is used to reduce the oscillations that are caused by the increase of the Kp and Ki [4]. Several tuning methods can be used for PID tuning which consists of manual tuning method, Ziegler-Nicholas method, software tools method, and Cohen-coon method [3].

The Air Levitation System is a system that able to lift an object without any mechanical support in a stable or desired position, by providing an upward force that is equal to the gravitational force exerted on the object [5]. Figure 1.2 [6] shows the prototype of the air levitation system.

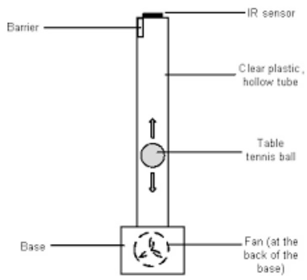


Figure 1.2: Air Levitation System.

1.2 Problem Statement

The stability of a system is a great concern in control engineering. An unstable system will cause the system unable to return to the state of balance and unable to accomplish the task [7]. Nowadays, the control engineers need to have both a wide experience implementing solutions in real problems, plants, and processes and a deep understanding of the mathematics and theory that lie behind these solutions. However, reaching a balance between theoretical proofs and physical intuition is a major challenge in control education. Other than that, the commercial control experiment systems that can be used to increase the understanding towards the control system engineering which exists in the market are expensive, tend to be large and heavy, and it is not flexible for many cases [5].

1.3 Aim and Objectives

This study aims to develop a low-cost air levitation system with PID control and to investigate the effect of the application of PID control on the system.

The objectives for research study are:

- To develop a low-cost air levitation system.
- To design the PID controller and applied to the air levitation system.
- To analyse the response of the developed system with PID control.

1.4 Scope of Project

In this study, the Arduino Uno is used as the microcontroller of this project with the purpose to control all of the activities of the output and input devices. The ultrasonic sensor HC-SR04 is used as the input device to detect the distance between the object and the sensor. The transparent acrylic plexiglass lucite tube with a diameter of 50mm is used as the medium where the object will be lifted. The ping pong ball is used as the object that will be lifted due to the smaller size and weight. The weight and the size of the ping pong ball are 2.7g and 40mm respectively. The valid distance that can be detected by the ultrasonic sensor is from 10 cm to 27 cm due to the length of the acrylic transparent tube being 30 cm. The 12V DC blower fan is used as the output device that produces the airflow to lift the object upward and the IRF520 MOSFET module driver is used as a switch to control the blower fan. The Simulink platform is used to design the PID controller and the system without the PID. The designed PID controller is then applied to the developed air levitation system. Through the Simulink, the value of the proportional term, integral term, and derivative term can be adjusted using the knob block and the desired height of the ping pong ball can be input by the user. The system identification toolbox is used to estimate the transfer function for the system with the PID controller and the system without the PID controller. The response of the air levitation system with the PID controller is analyzed.

1.5 Thesis Outline

The reminder of this document is structured as follows.

In Chapter 1, this chapter was introduced with the background of the control system, air levitation system, and PID control technique. The aim and objectives, problem statement, the scope of projects, and thesis outline will be covered in this chapter.

In Chapter 2, this chapter provides the reader with more detailed insight into the air levitation system with PID control and the knowledge related to the project. The previous works related to the study are discussed.

In Chapter 3, this chapter discusses the PID tuning method and the processes involved in completing this project. The flow chart of the operation of the air levitation system is shown and explained. All of the components including the hardware, platform, and controller that are used in this project also will be discussed in this chapter.

In Chapter 4, this chapter will explain the working principle for both of the developed systems. The result or the response of the system obtained from the MATLAB platform will be analyzed and discussed. The parameters such as rise time, percentage overshoot, steady-state value, steady-state error, and the settling time are determined from the obtained response. The project sustainability is discussed in this chapter.

In Chapter 5, this chapter will make the conclusion based on the research study that was made, and the future improvement that can be done on this study are suggested.

1.6 Summary

In this chapter, the background of this project is introduced and the problem statement is briefed. The purpose and the objectives of the project are discussed. The hardware, platform, and controller that going to be used in developing this project are explained.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In Chapter 2, the knowledge and previous study that related to the proposed project from the articles and journals are reviewed. The Proportional, Integral and Derivative control techniques will be discussed.

2.2 Types of System

There are two types of systems in the world which are linear systems and non-linear systems. However, in the real world, most of the systems behave as non-linear systems.

2.2.1 Linear System

A system is known as a linear system where the output of the system is proportional to the input that is applied. In other words, a system is known as linear when it is fulfill the superposition principle and homogeneity.

2.2.2 Non Linear System

A system is known as the non-linear system where the output of the system is not proportional to the input that is applied to the system [8]. There are some the examples of non-linear systems in real life such as air levitation system and magnetic levitation system. In this project, the unstable and non-linear air levitation system is chosen as the system to study the effect of the PID controller on the stability of the system. An air levitation system is a system that can lift an object at the desired position by providing the levitating force that is equal to the gravitational force that occurred on the object as shown in Figure 2.1 [9].

Figure 2.1: Concept of Air Levitation System.

2.2.2.1 General Equation for Air Levitation System

According to [10–12], based on Newton's second law, the equation of air levitation system has been investigated and written as:

$$m\Delta\ddot{z} = F = \frac{1}{2} \cdot C_d \cdot \rho \cdot A \cdot (V_w - \dot{z})^2 - m \cdot g \tag{2.1}$$

- m = mass of the levitated object;
- z = the position of object in vertical view;
- ρ = density of air;
- A = area of object exposed to air flow;
- V_w = velocity of air inside tube;
- g = gravity;
- C_d = drag coefficient.

According to research paper [5],the right hand side of the equation 2.1 represents the the levitating force and the gravitational force that are exerted to the object. The value of C_d is assumed as constant when the velocity is low and the constant variable, $\alpha = \frac{1}{2} \cdot C_d \cdot \rho \cdot A$ is inserted to equation 2.1 and express as:

$$\ddot{z} = \frac{\alpha}{m} \cdot (V_w - \dot{z})^2 - g \tag{2.2}$$

When the object is at a stable position without moving, the value of $\ddot{z}$ and $\dot{z}$ is equal to 0 and the speed of air to make the object at the stable position is known as V_{eq} and the equation with the function of g and $\frac{\alpha}{m}$ can be expressed as:

$$g = \frac{\alpha}{m} \cdot (V_{eq})^2 \tag{2.3}$$

$$\frac{\alpha}{m} = \frac{g}{\cdot (V_{eq})^2} \tag{2.4}$$

The equation 2.4 is then inserted into equation 2.2, and the final equation obtained will be:

$$\ddot{z} = g \cdot ((\frac{V_w - \dot{z}}{V_{eq}})^2 - 1) \tag{2.5}$$

The $x = \frac{V_m - \dot{z}}{V_{eq}}$ is substituted into the equation 2.4. The Taylor's approximation approach ($f(x) \approx f(1) + f'(1) \cdot (x-1)$) is used to linearized the system by setting $x = 1$ and the equation is expressed as:

$$\ddot{z} = 2 \cdot g \cdot (x-1) = \frac{2 \cdot g}{V_{eq}} \cdot (V_w - \dot{z} - V_{eq}) \tag{2.6}$$

The transfer function of the system can be defined as equation 2.7 , by assuming the system is well defined by the linearized plant.

$$\frac{\Delta z(s)}{\Delta v(s)} = \frac{1}{s} \frac{a}{(s+a)} \tag{2.7}$$

where

- $\alpha = \frac{2g}{V_{eq}}$
- Δz = Changing of the position of the object.
- Δv = speed of wind near the stable position.

While the transfer function with input voltage speed of wind of the fan can be represented as:

$$\frac{\Delta v(s)}{\Delta u(s)} = \frac{k_v}{\tau s + 1} \tag{2.8}$$

where

- k_v is the gain that relates the input voltage to the wind speed at steady state;
- τ is the fan time constant.

The equation 2.7 and equation 2.8 are multiplied to obtain the transfer function of the air levitation process that can be written as:

$$z(s) = \frac{1}{s}\frac{ak_v}{(s+a)(\tau s+1)} \tag{2.9}$$

2.3 PID Controller

The Proportional Integral Derivative (PID) controllers have been widely used in industrial process control in regulating the pressure, speed, temperature, and other process variables [13, 14]. It is because the PID controller has a simple design, provides a high stability response, and is easier to be applied [15]. There are 97% of the controllers in the industry or factory are applied PID control [16, 17]. The PID controller consists of three controllers which are the proportional controller, integral controller, and the derivative controller that produce a control signal to the controller to produce the desired output. The analog electronic components are used to execute the PID control before the invention of the microprocessors [13].

In 1911, the PID controller was invented by Elmer Sperry, then the Taylor Instrumental Company developed a tuneable pneumatic controller in the year 1933. The steady error produced by the proportional controller was removed by the control engineer by the feedback of the error to the controller was made the error approximate to zero. This feedback that consists of the error was known as the Proportional Integral (PI) controller. The overshooting problem for the response was reduced through a derivative action at which the first pneumatic PID controller was introduced in 1940 [13].

A closed-loop system is a system that has a feedback path to the controller, if there is an error, a control signal will be produced to the controller to compensate for the error and obtain the desired output. The PID controller provides better stability than the ON/OFF type controller at which the controller will be on or off under some assigned situation. As an example, if the output value is lower than the desired value, the controller will be on, and when the output value is higher than the desired value, the controller will be off. This situation will produce multiple overshoots for the output response and unstable, therefore the PID controller is widely used instead of the ON/OFF controller due to the ability for maintaining the output with zero error [13].

According to [18] and [19], the equation of PID controller can be represent as:

$$u(t) = K_p(e(t) + \frac{1}{T_i}\int e(t)\,dt + T_d\frac{de(t)}{dt}) \tag{2.10}$$

Based on [4], the equation 2.10 also can be written as by substituting the $K_i = \frac{K_p}{T_i}$ and $K_d = K_pT_d$ into the equation:

$$u(t) = K_pe(t) + K_i\int_0^t e(t)\,dt + K_d\frac{de}{dt} \tag{2.11}$$

2.3.1 Proportional Controller

For the Proportional (P) controller as shown in Figure 2.2 [13], the error of the system will be increased if the proportional gain increased. The setpoint will be compared with the output value of the system, the value of the error will be multiplied with the proportional gain to produce the output if the error is not zero. When the proportional gain increases the speed of the response also increases [13].

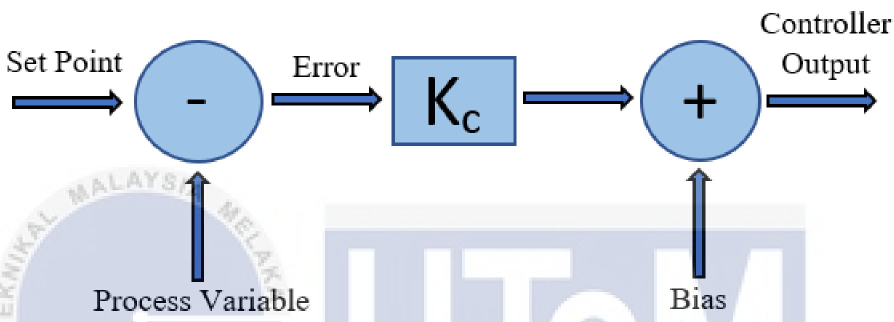


Figure 2.2: P Controller.

2.3.2 Integral Controller

For the Integral (I) controller as shown in Figure 2.3 [13], this controller is developed to overcome the limitation that has by the P controller at which there is always a difference in value for output and the setpoint. The integral controller can get rid of the error by integrating the error over time until the error becomes zero. The speed of the response and the stability of the system will be affected by the I controller. When the integral gain is increased, the speed of the response will be decreased [13].

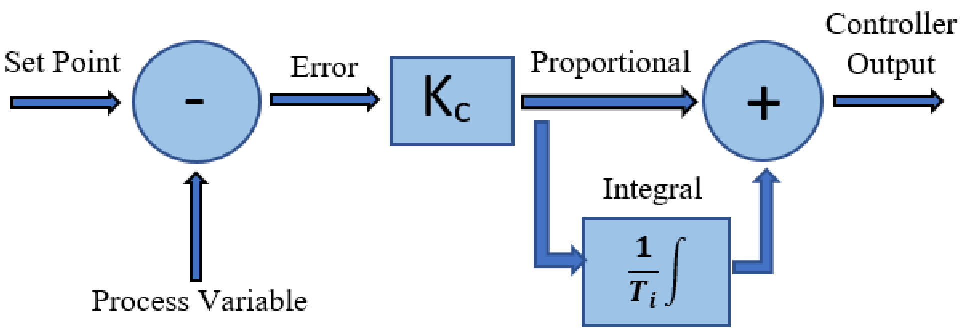


Figure 2.3: PI controller.

2.3.3 Derivative Controller

For the Derivative (D) controller as shown in Figure 2.4 [13], since the integral controller is unable to predict the future error, therefor the D controller is used to overcome this problem at which the output depends on the rate of change of error concerning time and the result is multiplied with the derivative gain. The stability of the system is increased by eliminating the phase delay caused by the I controller. When the derivative gain increases, the speed of the response also increases [13].

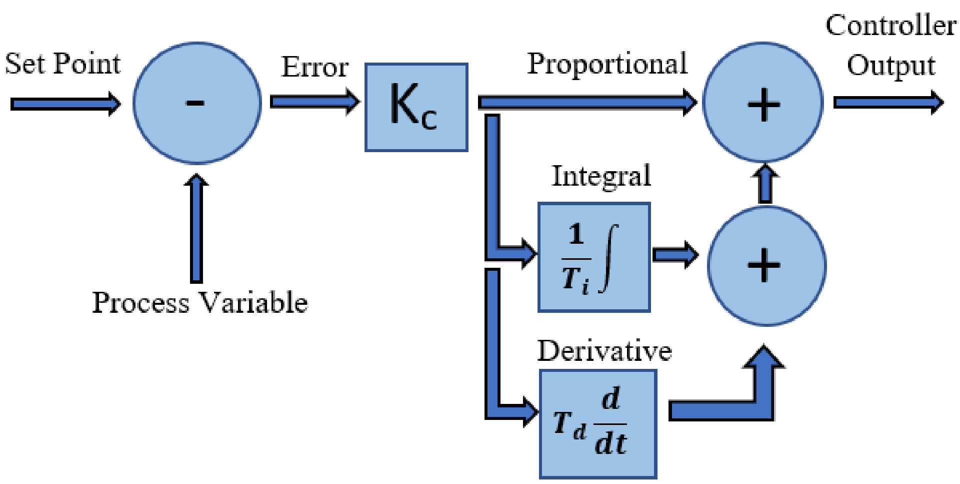


Figure 2.4: PID Controller.

2.3.4 PID Tuning Methods

The PID controller needs to be tuned to make sure the output of the process can be controlled [13]. There are several popular controller tuning methods that can be used to tune the controller and it consists of the physical tuning method, Ziegler-Nichols, and Cohen-Coon method. These methods are used in the mathematical model of the system that is not available [20].

2.3.4.1 Physical Tuning Method

The physical tuning method is a simple and easy method that can be used to tune the parameters of the controllers when the controller is operating to meet the desired response [13]. It is one of the methods that used to obtain the optimal values of Kp, Ki, and Kd of a PID controller.

2.3.4.2 Ziegler-Nichols Method

In 1942, a tuning method name Ziegler-Nichols method has been introduced by Ziegler and Nichols [21]. The values of Ultimate Gain (KU) and the Ultimate Period Of Oscillation (PU) are needed to be determined in order to get the value of Controller Constant (KC). This method can be refined so that a better approximation of the controller can be produced. The Ziegler-Nichols closed-loop tuning method cannot apply in an open-loop system. The value of ultimate gain, Ku is referred to the value of proportional gain of the controller, Kp that is applied when the amplitude of the oscillation produced are constant. While the ultimate period of oscillation, Pu is the period for one complete cycle of the constant amplitude oscillation that occurred [20]. Then, the values of Controller Constant (KC), Integrator Time Constant (TI), Derivative Time Constant (TD) are obtained through the tuning rule table as shown in Table 2.1 [13].

Table 2.1: Tuning Rule Table for Ziegler-Nichols Method

Control Structure	Kc	Ti	Td
P	0.5Ku	-	-
PI	0.45Ku	0.83Pu	-
PID	0.59Ku	0.5Pu	0.125Pu

2.3.4.3 Cohen-Coon Method

The Cohen-Coon method that is based on a First Order Process With Time Delay (FOPDT) process model is introduced by Cohen and Coon in 1953 [22, 23]. This method overcomes the slow, steady-state response produced by the Ziegler-Nichols method when there is a delay that occurs relative to the open-loop time constant and causes the Cohen-Coon method to be efficient. The Cohen-Coon method is also known as an offline method for turning at which the step change can be applied to the input if the response is at the steady state. The output is measured based on the time constant and the initial control parameter can be calculated through the response. There are a set of pre-determined settings to obtain the minimum offset and standard decay ratio of ¼ Quater Decay Ratio (QDR). A ¼ decay ratio means that the oscillations of a response will keep decrease such as the second oscillation will only have the 1/4 amplitude of the first oscillation. The equations for calculating the value of Kc, Ti, and Td are shown in Table 2.2 [20].

Table 2.2: Equation for optimization of Cohen-Coon Prediction

Control Structure	Kc	Ti	Td
P	(P/NL)*(1+(R/3))	-	-
PI	(P/NL)*(0.9+(R/12))	L*(30+3R)/(9+20R)	-
PID	(P/NL)*(1.33+(R/4))	L*(30+3R)/(9+20R)	4L/(11+2R)

where

- P = Percent Change of Input;
- N = Percent Change of Output;
- L = τ_{dead};
- R = $\frac{\tau_{dead}}{\tau}$.

2.4 Parameters for Step Response

There are several parameters such peak time (T_p), delay time(T_d, percentage overshoot($\%OS$), rise time (T_r), settling time (T_s), steady-state value and steady-state error can be obtained from a step response of a system. A step response of a system is defined as its response to a unit step input [24].

The percentage overshoot is referred to as the difference between the maximum value and the final value, and the value is divided by the final value. The equation 2.12 is used to calculate the percentage overshoot of the system.

$$\%OS = \frac{maximum\ value - final\ value}{final\ value} \tag{2.12}$$

The delay time is referred to the amount of time taken for the system to reach half of the steady-state value. The peak time is the time taken for the response to reach the maximum value for the first time. The rise time is the amount of time taken for the response to rise from the 10% to 90% of steady-state value. While the settling time is the time taken for the response to enter the steady-state phase where the value is within 2% or 5% of the steady-state value. The steady-state value is referred to as the final value of the response while the steady-state error is the difference between the steady-state value with the input that is applied to the system [25].

2.5 Related Research

There are a few previous types of research and journals that are related to the project are reviewed to get the idea and the solution that can be proposed to achieve the goals of this project.

2.5.1 Experimental and Implementation of Robust Control Via Floating Air Levitation and Balancing Rotary Inverted Pendulum

According to the research that is conducted by S Howimanporn, S Chookaew, and S Chaiyapora, there are two types of non-linear processes are chosen which are Floating Air Levitation system (FAL) and Rotary Inverted Pendulum system (RIP). The purpose of FAL system is to lift an object at the desired position while the RIP system is to keep the pendulum at a stable upright position. Both of the systems are operated based on the Programmable Logic Controller (PLC) The controllers such as PID controller and Sliding Mode Controller (SMC) are designed and implemented to the PLC system by using the SYSMAC Studio software. This study concentrates on the performance of the designed controllers that are applied to the systems. The parameters such as rise time, percentage overshoot, settling time, peak time, and dead time of the response are compared between both of the systems for each of the controllers [26].

2.5.2 Design of a Low-Cost Air Levitation System for Teaching Control Engineering

According to the research conducted by Jesus Chacon, Jacobo Saenz, Luis de la Torre, Jose Manuel Díaz, and Francisco Esquembre, a low-cost virtual and remotely controlled air levitation system based on open solutions are developed. A remote lab is a physical system that can be accessed or controlled through a platform by connected to the Internet. While the virtual lab is the replacement of the physical system with the mathematical model that can generate the response of the system. The presented system has a simplified design which means the system will be affordable and easy to be replicated. It is due to the commercial experimental platforms are expensive and sometimes less flexible. The components such as the methacrylate tube, a small and light object, 3D printed parts, a single-board computer (Beaglebone Black), a Passive Infrared Sensor (PIR), a Personal Computer (PC) fan, some discrete electronic components, a Printed Circuit Board (PCB) and a webcam are used to build the system. There are a few lab activities that can be performed by the students which consist of PID tuning, disturbance analysis, and system identification. For the PID tuning activity, the proportional, integral and derivative terms can be adjusted to achieve the stated requirement of the system. For the disturbance analysis, the wind flux outside the fan is varied to obtain the disturbance that occurs in the system. For the system identification activity, the information and data needed to be collected from the remote and virtual lab to obtain the information related to the order, poles, and zeros of the system. The goal of this research is to develop a low-cost virtual and remote control air levitation that can be used by the teachers in teaching control engineering subjects [5].

2.5.3 Experimental Study of Nonlinear PID Controllers in an Air Levitation System

According to the research study conducted by J.Chacon, H. Vargas, S. Dormido, and J.Sanchez, this paper study the performance of the different non-linear Proportional Integral, PI controllers using an academic platform based on an Air Levitation system. There are a few of the control strategies that are carried out, tested, and compared in an academic plant. The comparison for the applied control strategies is done based on the performance index such as the Integral Absolute Error (IAE), or Integral Time Multiple Absolute Errors (ITAE). The purpose of this study is to understand the advantages and disadvantages of the application of different control strategies on the academic plant and to demonstrate the significance of the designed air levitation system in control engineering education. The control strategies consist of a PI controller, a PI controller with Clegg's Integrator, an event-based PI controller with feedforward and robust adaptive hybrid PI are applied and tested. The classical PI controller is used as the reference for the comparison purpose with other control strategies. This research has proved that several non-linear modifications can be considered to cope with different control needs which consist of optimization on the number of control actions or go beyond the restrictions that linearity impose on the performance that can be achieved by a PID by adding the component such as integrator to the PI controller [9].

2.5.4 The Air Levitation System

According to the research conducted by Jesus Chacon, Luis de la Torre, and Sebastian Dormido the proposed air levitation system is a plan to be used as the experimentation framework in teaching and learning control engineering [27]. The major concern towards Virtual and Remote Labs (VRL) has been solved and the importance and significance of VRL in teaching have been raised due to its capability to produce the learning outcomes as the hands-on labs [28] [29]. The air levitation system consists of a tube with an object inside that can be lifted to the desired position and a fan is used to produce the airflow to levitate the object. The Infrared, IR sensor is used to determine the position of the object [5]. An Arduino board is used to control the operating activities of the input and output devices, while an extra feature is added to the system by using the Raspberry PI that can use as data storage. The proposed air levitation system is allowed to perform different experiments. The two kinds of proposed experiments are to understand the behaviour of the system and to control the system to meet the design specification. There are two types of control strategies that consist of PI control and sliding mode control are used in this research. The Air Levitation System is a low-cost experimental platform that is used in teaching control engineering courses. The system is flexible which can be used as a hand on lab and implemented as the VRL that can be accessed by different users at the same time [27].

2.5.5 Low-Cost Air Levitation Laboratory Stand Using Simulink And Arduino

According to the research conducted by Ewelina Cholodowicz and Przemyslaw Orlowski, a low-cost air levitation project is proposed that is used as a training toolkit in teaching control subjects. The proposed system is providing an opportunity for the students to control and learn the concept of control engineering. In this study, the portable and the affordable designed system is used as a tool for learning control engineering. The components involved in this system consist of Arduino Mega 2560, axial box fan, ultrasonic sensor, plexiglass tube, ping pong ball, electronic circuits, and MATLAB/Simulink platform. The air levitation is an unstable nonlinear system and it is classified as a single input and single output in control systems. The airflow produced by the fan is referred to as the input signal while the distance between the sensor and the lifted object is the output signal. A PID controller is developed in the MATLAB platform and implemented on the designed system. To control the non-linear system, system identification and optimization techniques are used by the researchers. For the proposed system, the students can tune the parameters of the PID controller in the platform. The implementation of the control algorithms can be done on the proposed system that can increase the understanding of students towards control engineering and the total cost for the system is under 80 United States (US) dollars [30].

2.5.6 Comparison Study of PID Controller Tuning Using Classical/Analytical Methods

Based on the study conducted by B.Mabu Sarif, D. V. Ashok Kumar, and M. Venu Gopala Rao, the Sundaresan and Keishnaswamy method is applied to obtain the model. The open and closed-loop tuning techniques are applied to determine the performance of the PID controller. The Internal Model Control (IMC) method is applied to tune the parameters of the controller for input tracking. The open-loop response of a system is obtained by disconnecting the connection of the feedback controller and the step change of input applied. The closed-loop tuning technique depends on the parameters of ultimate gain, Ku and ultimate period, Pu. The value of Ku and Pu is obtained by setting the value of integral and derivative terms to zero and the proportional term is increased to produce constant amplitude of oscillations. The Pu is the time interval between two peaks of oscillation that have identical values of frequency and amplitude. While the Ku is the value of gain that is used to produce the frequency and amplitude oscillation become constant. The robustness analysis is carried out by the researcher to determine the capability of the system to withstand the changing of the process variable without affecting the stability of the system. In this study, better setpoint tracking but sluggish disturbance rejection can be provided by the IMC-PID. While the Ziegler-Nichols and Cohen-Coon techniques provided better disturbance rejection for both systems which are superheated steam temperature systems of 500MW boiler and mean arterial blood pressure system. There is only one parameter that needs to be tuned for the IMC-PID technique to achieve the desired performance and this technique provide better performance and robustness if compared with other techniques [21].

2.5.7 Comparison of Ziegler-Nichols and Cohen-Coon Tuning Method for Magnetic Levitation Control System

Based on the research conducted by F. Isdaryani, F. Ferrionika, and R. Ferdiansyah, the magnetic levitation system is nonlinear, it can be used to determine the efficiency of the control technique that applied. The nonlinearity of the magnetic levitation system is reduced by placing two effect hall sensors on the top and bottom of the magnetic coil. The sensors are used to measure the value of the magnetic field and the value will be converted into a voltage signal through the signal conditioning circuit before it is applied to the input of the controller. There are two tuning methods which include of Ziegler-Nichols method and Cohen-Coon method are compared to obtain the appropriate structure either it is Proportional Integral (PI) controller, Proportional Integral Derivative (PID) controller, Proportional Derivative (PD) controller, or Proportional (P) controller and the initial parameter of the nonlinear system is reduced. Based on the response of the system obtained from two different approaches, the control structure PID controller for the Ziegler-Nichols method provided good response than the PI controller for Cohen-Coon method at which the rise time, and settling time were lower than that in the PI controller. Both of the responses for control structures of PID and PI for different methods do not produce the overshoot, delay, and steady-state error [31].

2.6 Summary for Literature Review

In short, the concept of developing an air levitation system and the PID control techniques that can increase the stability of a system are studied in this chapter. Based on some of the research papers that have referred, the output devices of the air levitation system that are used to lift the object at the desired position are the PC fan and axial fan. There are some of the microcontrollers that are used by the researchers in developing the physical air levitation system such as Beaglebone Black, and Arduino Mega 2560. The microcontroller is chosen based on the requirement and function of their project. While the sensors that are used by the researchers to measure the distance between the sensor and the object consist of a PIR sensor, an ultrasonic sensor. Some of the research papers have proposed the concept of virtual and remote lab VRL where multiple users are allowed to access the system at the same time. Most of the systems in the real world are naturally unstable and nonlinear. Therefore, the control technique such as PI controller, PD controller, or PID controller is needed to be applied to make sure the system behaved as desired. Various tuning technique methods can be used to tune the parameter of PID which are the proportional term, integral term, and derivative term. These methods consist of the physical tuning method, Ziegler-Nicholas method, and Cohen-Coon method. Based on the research conducted by other researchers, the Ziegler-Nicholas Method provides a better response than the Cohen-Coon method in the magnetic levitation system.

2.7 Proposed Project

The proposed project is the development of a low-cost air levitation system with a PID controller. The step responses for a low-cost air levitation system with PID controller and without PID controller are generated and analyzed. The Arduino Uno is used as the microcontroller of this project to monitor and coordinate the activities of the input and output devices. The ultrasonic sensor HC-SR04 is used as the sensor to measure the distance between the ping pong ball. The valid distance that can be detected by the ultrasonic sensor is from 4 cm to 400 cm as shown in appendix B [32]. In this project, the ultrasonic sensor is chosen instead of the Sharp IR sensor GP2Y0A21YK due to the ultrasonic sensor does not have the limitation caused by the non-linear response that has by the Sharp IR sensor GP2Y0A21YK as shown in Appendix C [33]. The output device that is used to produce the airflow to lift the ping pong ball is the 12V Direct Current (DC) blower fan. The IRF520 MOSFET module driver is used as a switch to control the blower fan that is connected to a 9V battery. It is because the typical output voltage for Arduino Uno is at 5V and 40mA respectively as shown in appendix A [34]. After the development of the air levitation system, the codes are written in Arduino IDE platform to ensure the functionality of the system. After that, the PID controller and the Simulink model of the system are designed in the Simulink platform by using the block functions that have from the Arduino support package. The designed system is then integrated with the Simulink platform and the user is allowed to tune on the parameters Kp, Ki, and Kd from the knob block in the Simulink platform. Then, the physical tuning method is applied to determine the optimal values of the proportional gain, integral gain, and derivative gain. The optimal values are then applied to the system and the input and output data sets of the system with PID controller and without PID controller are saved in the workspace of the MATLAB platform . The transfer functions for both of the systems are estimated by using the System Identification Toolbox and the step responses of the systems are generated. To investigate the effectiveness of the PID controller, the values of the rise time, settling time, steady-state error, and percentage overshoot are measured.

2.8 Summary

In this chapter, the types of systems that consist of linear and non-linear systems are discussed. The general equation of the air levitation system is explained and the concept of the air levitation system is introduced. The related researches that conducted by other researchers are referred and the project is proposed.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, this chapter discusses the PID tuning method and the processes involved in completing this project. The flow chart of the operation of the air levitation system is shown and explained. All of the components including the hardware, platform, and controller that are used in this project also will be discussed in this chapter.

3.2 Research Methodology Flow Chart

From the flow chart shown in Figure 3.1, all of the processes involved in developing this project are included. The information related to the PID controller and the air levitation system were collected through online search engines for academic publications resources such as Google Scholar, Semantic Scholar, ResearchGate, and IEEE explorer so that the understanding of the concept of PID control and the operation of the air levitation system could be improved. All of the components and the platform that going to be used in developing this project were determined and planned. This project could be classified into three stages which consist of the development of the air levitation system, PID controller design, and result analysis.

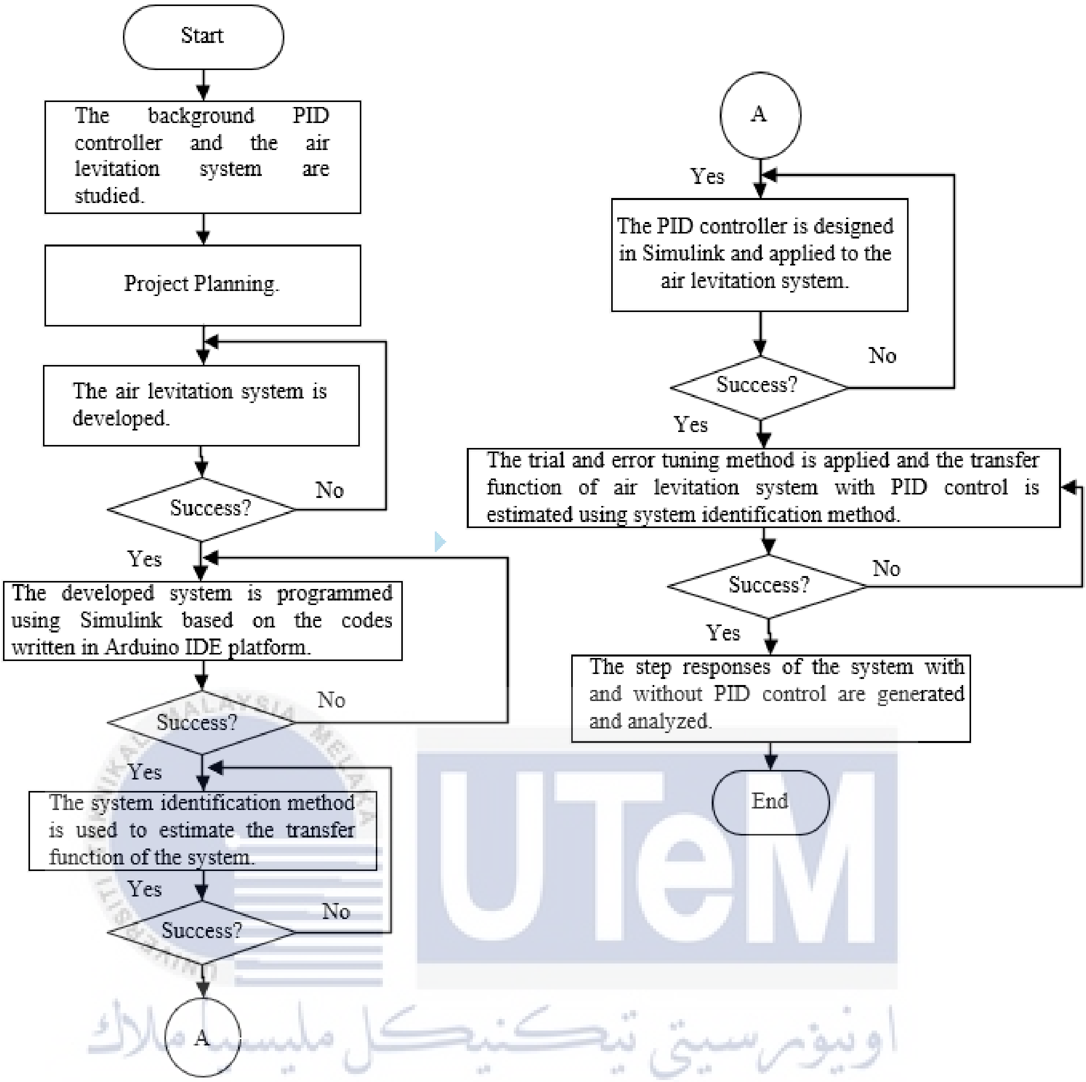


Figure 3.1: Research Methodology Flow Chart.

The development of the air levitation system was the first stage for this project, the components such as the acrylic transparent tube, ping pong ball, ultrasonic sensor, blower fan, IRF520 module driver, and Arduino Uno were used to develop the air levitation system. The C++ programming skill is required to program the system to function as desired by using the Arduino Integrated Development Environment, IDE. The codes were uploaded to the microcontroller or Arduino Uno through the Universal Serial Bus, USB cable type A/B to ensure the functionality of the system. The ping pong ball is lifted by the airflow produced by the blower fan in the tube and the position of the ball is determined by the sensor. The developed system was then programmed using the block functions that have in Simulink based on the codes written in Arduino IDE platform in order to obtain the data sets that consist of input and output values of the system. The data sets were used to estimate the transfer function of the air levitation system without PID control through the system identification toolbox.

For the second stage, the PID controller of the air levitation system was designed in Simulink platform. A closed-loop system that consists of a feedback path is developed to compensate for the error between the setpoint and measured variable. PID control could be achieved by constructing the Simulink model of the PID controller in the Simulink platform and the Arduino support package must be downloaded in the platform so that the real-time simulation of the air levitation system by using Simulink and Arduino board could be realized. The physical tuning method was used in tuning the parameters Kp, Ki, and Kd to get the ideal values of the gains. The gains are then applied to the air levitation system with PID controller. The input and output data sets obtained from the system will be stored at the workspace of MATLAB platform and the transfer function is estimated by using the system identification toolbox.

While for the third stage, the graph with the actual position of the ping pong ball as output and the desired height as input is obtained through the scope feature in Simulink platform. Both of the graphs are compared and the step responses of the systems are generated using MATLAB platform through the transfer functions that are obtained for both of the systems. After the step responses of the systems will be analyzed with the parameters such as rise time, settling time, percentage of overshoot, and steady-state error that are determined from the responses. The collected results and finding for this project will be discussed in Chapter 4 of this report.

3.3 Hardware

For the development of the air levitation system, the components required such as breadboard, transparent acrylic plexiglass lucite tube, ultrasonic sensor, ping pong ball, IRF520 driver module, and 12V blower fan were used. The draft prototype of the air levitation system was sketched by using 3D paint software and shown in Figure 3.2. The blower fan, electronic circuit, and microcontroller were placed in the box to produce a clean and tidy prototype.

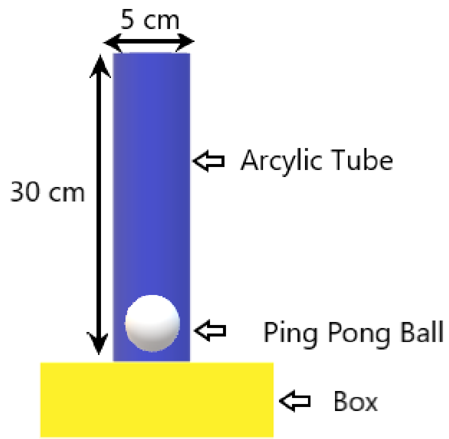


Figure 3.2: Draft Prototype.

3.3.1 Microcontroller

The Arduino Uno was used as the microcontroller for this project. It is because the Arduino board can control all of the activities of the ultrasonic sensor and mosfet module. The ATmega328 was used as the chip for the Arduino Uno board. The Arduino boards consist of digital and analog pins and some of the digital pins have Pulse Width Modulation (PWM) feature. PWM is referred to one type of modulation at which the ON time or the duty cycle of a digital unipolar square wave signal could be adjusted as desired. The application of PWM could be seen in controlling the speed of motors, and the brightness of LED.

The USB cable on the board is used to connect the board to the PC or laptop that allowing a user to program or upload the codes to the microcontroller through the installed platform. The operating voltage of Arduino Uno is 5V which the input voltage that could supply was in the range from 7V to 12 V. The maximum DC rating of the input or output pins of the Arduino Uno is only 40mA that shown in appendix A [34]. Figure 3.3 [35] shows the pinouts of the Arduino Uno board.

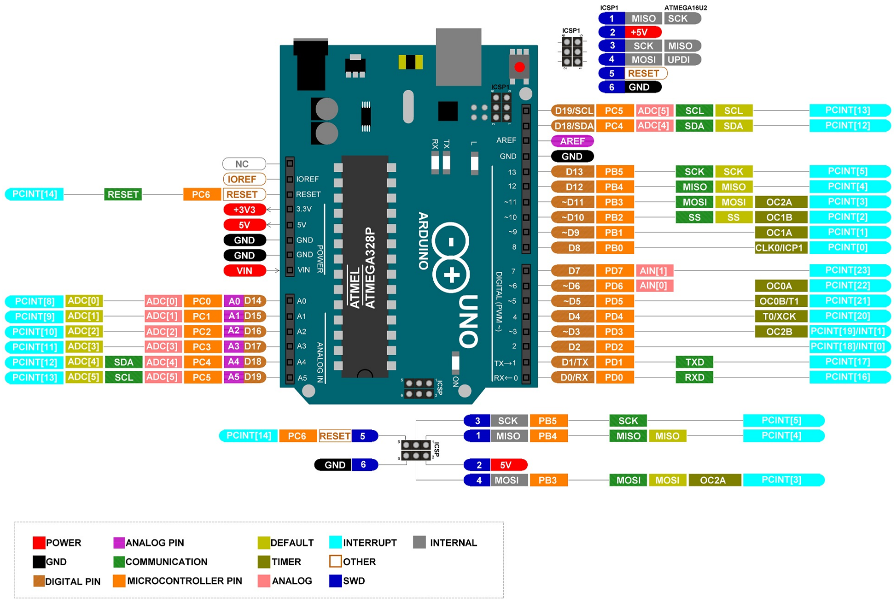


Figure 3.3: Pinout of Arduino Uno.

3.3.2 Ultrasonic Sensor

The ultrasonic sensor HC-SR04 as shown in Figure 3.4 [36] is used to determine the actual position of the ping pong ball in the transparent tube.

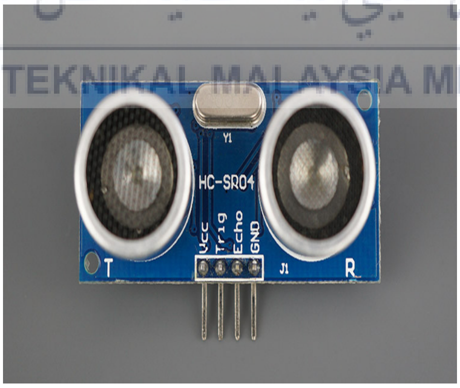


Figure 3.4: Ultrasonic Sensor HC-SR 04.

On an ultrasonic sensor, there were three important components which consist of a transmitter, a receiver, and a crystal oscillator as shown in Figure 3.5 [37]. Moreover, there are four pins on the ultrasonic sensor which are trigger pin, echo pin, voltage pin, and ground pin. The ultrasonic sensor is needed to be connected to a microcontroller so that the sensor can be worked properly.

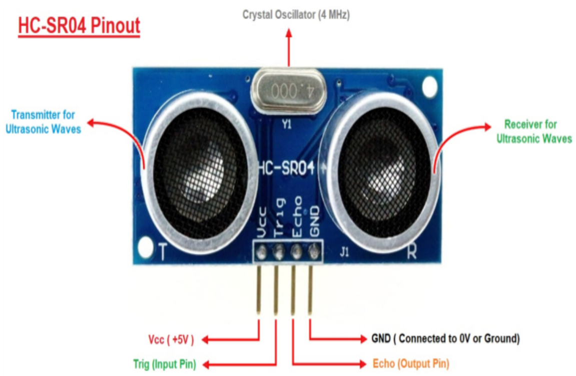


Figure 3.5: Pinouts and Components of Ultrasonic Sensor.

The oscillator is used to create the 4 Mhz signal pulses and the signal will be transmitted by the transmitter of the sensor. While the receiver is used to detect the reflected signal when the transmitted signal has hit on an object in a time of period. The basic operation of the ultrasonic sensor has based on the concept of sonar detecting as shown in Figure 3.6 [36]. The signal will be transmitted by the transmitter through the programming codes written to the microcontroller. When an object is obstructing the transmitted signal, the signal will be reflected and sensed by the receiver.

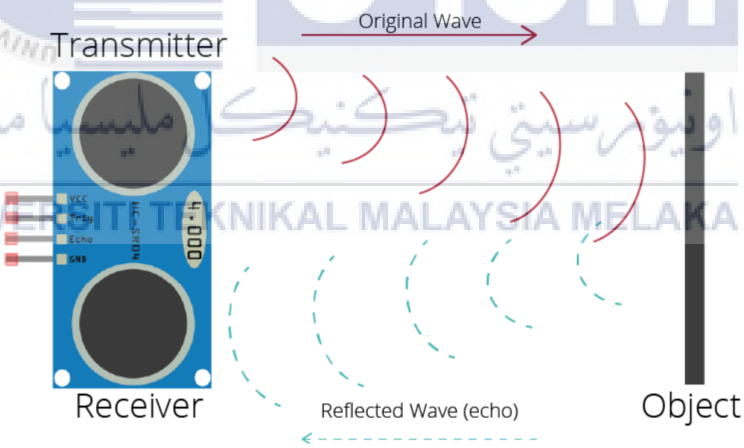


Figure 3.6: Operation of ultrasonic sensor.

According to [36], the equation that is used to calculate the distance can be written as:

$$d = \frac{(v \cdot t)}{2} \tag{3.1}$$

where

- d=distance between sensor and object
- v=speed of sound in the air (20 degree celcius) = 343m/s
- t=time taken for the signal to transmit and receive

Based on Appendix B [32], the ultrasonic sensor has a valid detected range from 2 cm to 400 cm. In this study, the valid distance that could be detected by the ultrasonic sensor was from 10 cm to 27cm. It was due to the acrylic tube that is used in this project being 30 cm in length.

3.3.3 Ping Pong Ball

In this project, the ping pong ball was chosen as the object to be lifted by the blower fan. It is because the ping pong ball has a lower weight and smaller diameter. A ping pong ball weighs 2.7g with a diameter of 40mm. Moreover, the ping pong ball is easier to be obtained. Figure 3.7 shows the ping pong ball that is used in this study.

Figure 3.7: Ping Pong Ball.

3.3.4 Blower Fan

For the output device, the 12 V DC blower fan was used instead of the PC fan to produce the airflow that could lift the object at a desired position in the tube. It is because the PC fan is not able to produce sufficient airflow to lift the ping pong ball that the blower fan could do. The operating voltage of blower fan ranges from 9V to 12V. The DC 12V brushless blower fan manufactured by Idea Electronic was used in this study as shown in Figure 3.8.

Figure 3.8: 12V Blower Fan.

3.3.5 IRF520 MOSFET Module Driver

In this project, the blower fan is connected to a 9V battery. To control on or off of the blower fan, an IRF520 MOSFET module driver functions as a switch as shown in Figure 3.9 [38] is required. The IRF520 module driver is chosen in this project. According to the typical transfer characteristic graph in Appendix D [39] , the drain current, I_D that can be produced by IRF520 is sufficient to operate the blower fan when the V_{GS} applied is equal to 5V which is the maximum voltage rating of Arduino Uno board.

There are 3 pins on the module which consist of a signal pin, a voltage pin, and a ground pin. The signal pin is connected to the output pin that has the PWM feature of the Arduino board so that the speed control of the blower fan can be achieved. The voltage pin and ground pin are connected to the 5V and the ground pin of the microcontroller respectively. The power supply of the blower fan and the blower fan are connected to the module driver through the terminal blocks on that module.

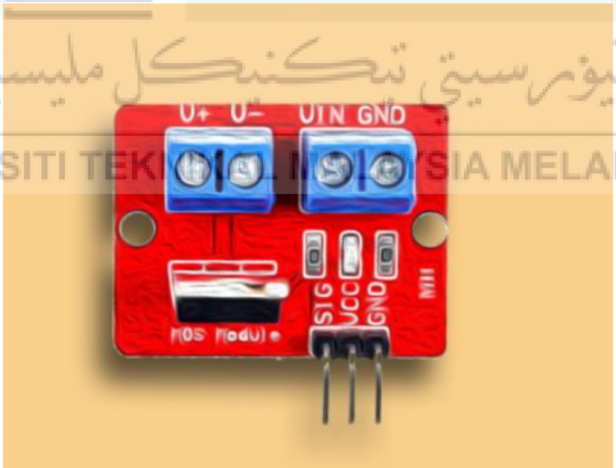


Figure 3.9: IRF520 module driver.

3.3.6 Transparent Acrylic Plexiglass Lucite Tube

The medium that is used where the ping pong ball is lifted is a transparent acrylic plexiglass lucite tube as shown in Figure 3.10 . The tube has an outer diameter of 50mm and an inner diameter of 43.75 mm. While the height of the tube is equal to 30 cm.

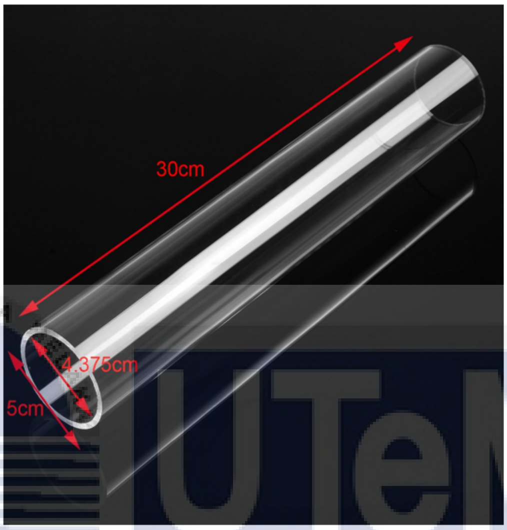


Figure 3.10: Transparent Acrylic Plexiglass Lucite Tube.

3.4 Platform

Three platforms are used in this project which consist of the Arduino IDE platform, MATLAB platform, and Simulink platform.

3.4.1 Arduino IDE Platform

The Arduino Integrated Development Environment (IDE) platform as shown in Figure 3.11 [40] was used to program the operation of the developed system in terms of code and the code was uploaded through the USB type A/B cable to the Arduino Uno to ensure the functionality of the developed system. The Arduino IDE was chosen due to the platform is easier to be used and the existence of the serial monitor tool that could be used to display the character or the sensor value by writing the command "Serial.print():". Other than that, some various examples or libraries could be included and used to program the microcontroller efficiently.

Figure 3.11: Arduino IDE Platform.

3.4.2 MATLAB Platform

The MATLAB platform as shown in Figure 3.12 [41] was used to generate the step response from the obtained transfer function for both of the systems and to perform the analysis for the obtained result. According to [42], MATLAB is a designed programming platform that allows users to analyze and to design system.

Figure 3.12: MATLAB Platform.

While the Simulink platform as shown in Figure 3.13 [43] is used to design the PID controller and to develop the operation of the developed system. A Simulink as shown in is a platform for model-based design that allows system design, simulation of designed system, and code generation based on designed model. [?] There are some predefined functions and blocks such as the PID controller in the Simulink platform which can increase the efficiency in designing a system. Other than that, the knob block in the Simulink platform allows the tuning of the PID parameters such as Kp, Ki and Kd could be done easily by the user in the platform.

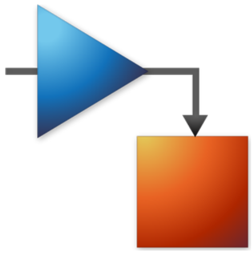


Figure 3.13: Simulink Platform.

3.5 Development of Air Levitation System Using Arduino IDE Platform

The development of the air levitation system could be separated into four parts as shown in Figure 3.14 which consist of circuit design, speed control of the blower fan, the configuration of the ultrasonic sensor, and the combination operation of the blower fan and ultrasonic sensor. All of the dimensions of the components were taking considered in developing the prototype of the system.

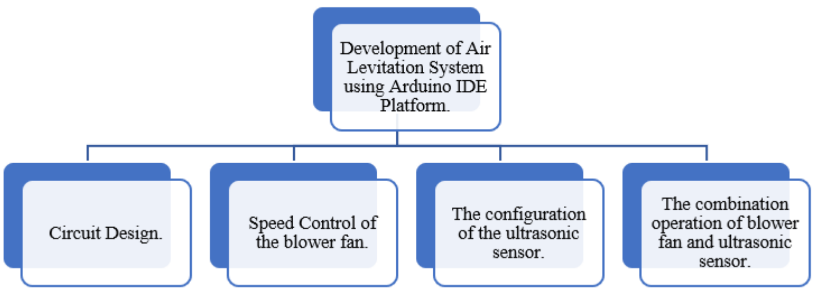


Figure 3.14: Process For Development of Air Levitation System using Arduino IDE Platform.

3.5.1 Circuit Design

Before the system is programmed using Before the system is programmed using the Arduino IDE platform, the physical air levitation system is needed to be developed. Therefore, a circuit that specified the wire connections between the input device, the output device, and the microcontroller is designed as shown in Figure 3.15.

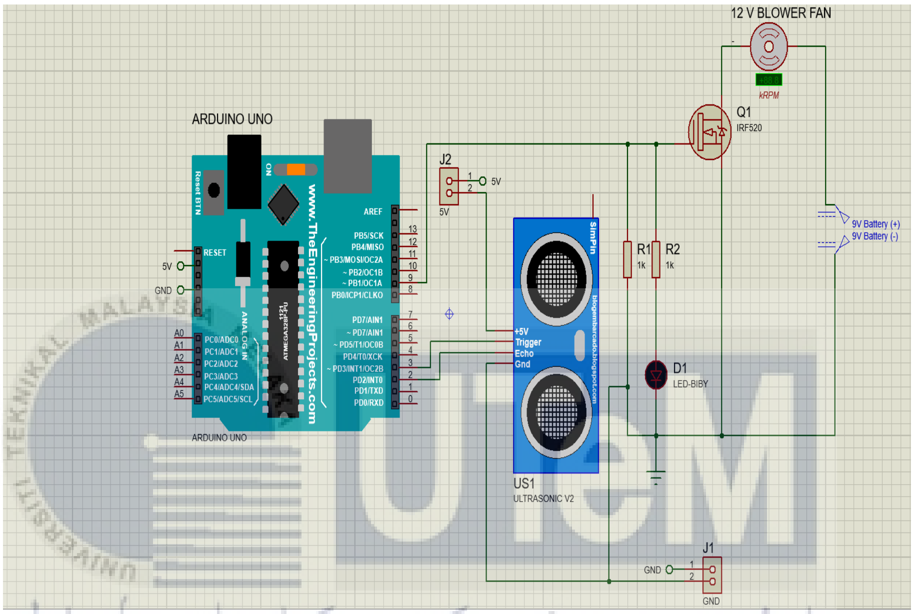


Figure 3.15: Schematic Diagram of Air Levitation System.

From Figure 3.15, the trigger pin and echo pin of the ultrasonic sensor are connected to pin 3 and pin 2 of the Arduino board. While pin 9 has the PWM feature is connected to the gate terminal of the IRF520 or the signal pin of the module. The negative terminal of the blower fan is connected drain terminal of the MOSFET and the positive terminal is connected to the positive terminal of the 9V battery. Then, the designed circuit is then transferred into the hardware circuit as shown in Figure 3.16

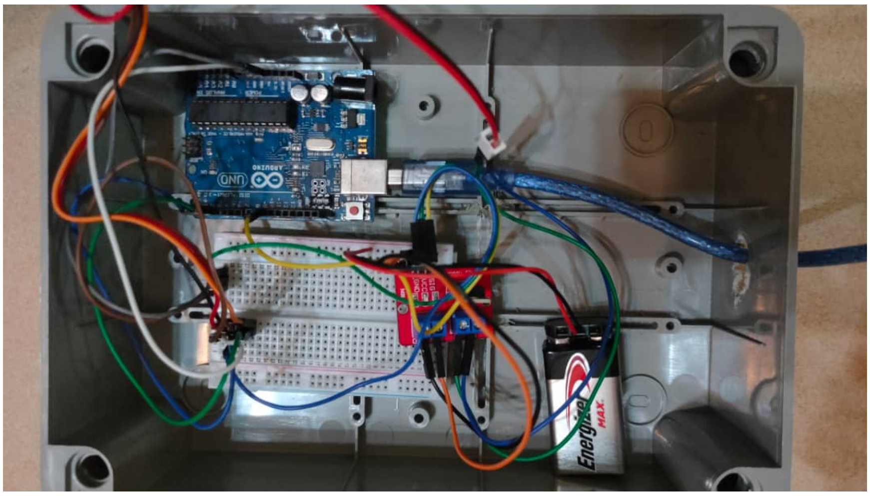


Figure 3.16: Hardware Circuit.

3.5.2 The Speed Control Of The Blower Fan

The speed of the blower fan was controlled by using the PWM feature that has on the input or output pins of the Arduino Uno through the programming. There are 6 pins on the Arduino Uno that have the PWM feature are D3, D5, D6, D9, D10, and D11. In this project, pin D9 was chosen as the output pin to control the speed of the blower. The PWM could be achieved by calling the “analogWrite(pin, value);”, the D9 would be the pin and the value was in the between numbers from 0 to 255. The PWM is controlling the duty cycle of the signal. When the duty cycle is 100 % with the value of 255 is written in the “analogWrite(D9,255);”, the output voltage provided by the output pin is equal to 5V. When the value is equal to 0, the duty cycle of the signal is equal to 0 % which turns off the output pin with the voltage value of 0V. Figure 3.17 shows the codes used to control the speed of the blower.

```
#define PWMPin 9
int i=255;

void setup() {
  // put your setup code here, to run once:
  pinMode(PWMPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  analogWrite(blower_pin,i);
  i--;
}
```

Figure 3.17: Code for Speed Control of Blower Fan.

The maximum value of variable i is set to 255, and the value of i will decrease each time the loop is executed. When the value of i decreases, the speed of the blower will also decrease due to the percentage of the duty cycle of the signal from pin 9 is reduced.

3.5.3 The Configuration of the Ultrasonic Sensor

For the configuration of the ultrasonic sensor, it is important to make sure the echo and trigger pins of the sensor are connected correctly before the codes are applied. The trigger pin initially was set to low for 2 microseconds and set to high for 10 microseconds to transmit eight pulses of signal in 4MHz. The "pulseIn" command is used to measure the time taken for the transmitted signal to be reflected and detected by the receiver and set by setting the echo pin to high. The equation 3.1 is implemented in terms of codes in the Arduino IDE platform. Figure 3.18 shows the codes implemented for the configuration of the ultrasonic sensor.

```
void loop() {
  // put your main code here, to run repeatedly:
  // Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)
  // Displays the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
  delay(100);
  Serial.println(" cm");

}
```

Figure 3.18: Code for the configuration of Ultrasonic Sensor.

The Serial Monitor was used to display the detected distance of the sensor as shown in Figure 3.19, so that the checking and validating process could be done.

```
COM4

Detected Distance:27cm
Detected Distance:27cm
Detected Distance:27cm
Detected Distance:27cm
Detected Distance:27cm
Detected Distance:27cm
Detected Distance:27cm
Detected Distance:27cm
Detected Distance:27cm
Detected Distance:27cm
```

Figure 3.19: Detected Distance that Displayed on Serial Monitor.

3.5.4 Combination Operation of the Blower Fan and Ultrasonic Sensor

After the speed control of the blower fan and the configuration of the ultrasonic sensor are done, both of the operations were combined through the programming codes. Some of the conditions were declared by using the if. . . .else statement, to ensure the microcontroller could perform the desired activities when the condition was fulfilled. The input of the desired position of the ping pong ball could be entered by the user through the serial monitor and this process can be achieved by using the code “input value = Serial.parseInt();”. After the input was entered by the user, the air levitation system would be activated, and the ping pong will be lifted by the blower fan to the desired position. Since the air levitation system is one of the non-linear and unstable systems, the unstable of the system caused the ping pong ball to move upwards and downwards instead of steady. Figure 3.20 shows the codes for reading input from the user and Figure 3.21 shows the developed prototype of the system.


```
void reading_input(){ //reading input from user
  if (Serial.available() > 0) {
  input_value = Serial.parseInt();
  Serial.print("Input:");
  Serial.print(input_value);
  Serial.println("cm");
  delay(200);
   if (input_value > 27)
    {
      Serial.println("Invalid Input ! Please enter again Between range 0 to 27");
    }
  }
}
```

Figure 3.20: Codes For Reading Input from the user.

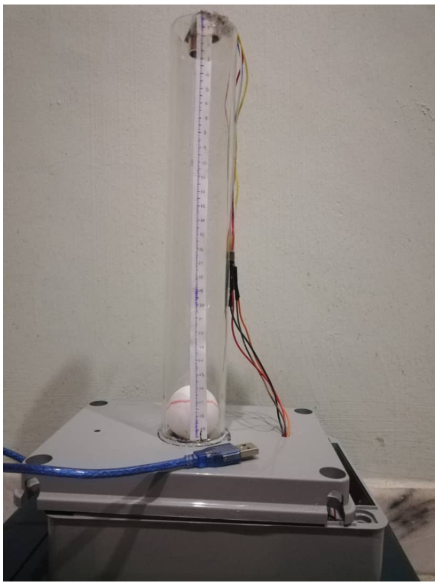


Figure 3.21: Developed Prototype.

3.6 Development of Air Levitation System without PID Controller Using MATLAB/Simulink platform

After the development of the air levitation system using the Arduino IDE platform is done, the model of the air levitation system as shown in Figure 3.22 that have the same operation at the Arduino IDE platform is developed in Simulink platform by using the blocks that have. The reason that the system is transferred from the Arduino IDE platform to the MATLAB platform is to export the input and output data to the workspace of the MATLAB platform. To use the Arduino board with the Simulink platform, the Simulink support package for Arduino Hardware must be installed and a few step-up processes need to be done to configure the microcontroller that going to be used in ths project.

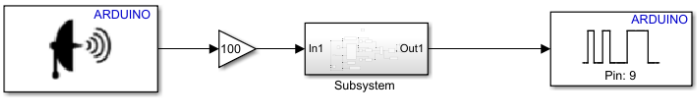


Figure 3.22: Designed Model for Air Levitation System without PID controller.

From Figure 3.22, the ultrasonic sensor block is found from the library browser at the Simulink platform and connected to a gain of 100 to get the value in centimeter. It is because the value of distance that from the block is in meters. The output from the gain block is connected to the input of the subsystem while the output of the subsystem is connected to the PWM block that is connected to pin D9 of Arduino Uno. The subsystem as shown in Figure 3.23 consists of the control method that is designed based on the codes written in the Arduino IDE platform.

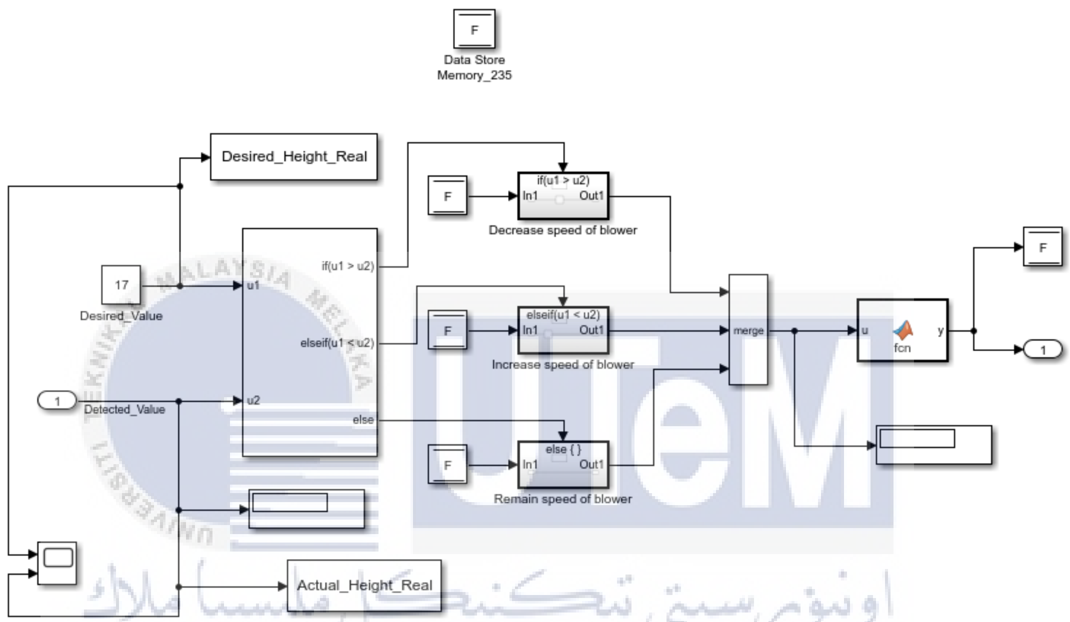


Figure 3.23: Subsystem for system without PID controller.

The control method that is used to control the position of the ping pong ball for the system without PID controller is the conditional statement (if....else) as shown in Figure 3.23. A variable named F is created to store the latest value of the PWM value so that the value can keep changing based on the action for each of the defined conditions. Three conditions have been defined to control the speed of the blower fan based on the position of the ping pong ball and the desired height. For the first condition, when the desired height (input) is lower than the actual height of the ping pong ball (output), the speed of the blower fan will increase (increase PWM value). Besides, if the desired height is higher than the actual height of the object, the speed of the blower fan will decrease (reduce PWM value). The speed of the blower fan is remained if the desired height is equal to the actual position of the ping pong ball.

A constant block named "desired height" allows the user to input the desired height of the object in the tube. Since there are multiple outputs from the conditional actions a merge block is required to pass the output value once at a time when the respective condition is fulfilled. The scope block is used to observe the changes of input and output at the same time. The input and the output data of the system are exported to the workspace of the MATLAB platform by using two To Workspace blocks named desired height and actual height. The input and the output data are used to estimate the transfer function of the system. The mode of the simulation must be changed to external mode and the time is set to infinity in order to run the real-time simulation for the developed system without PID controller.

3.7 PID Controller Design and Implementation

After the development of system without PID controller is done, the PID controller is designed in the Simulink platform as shown in Figure 3.24

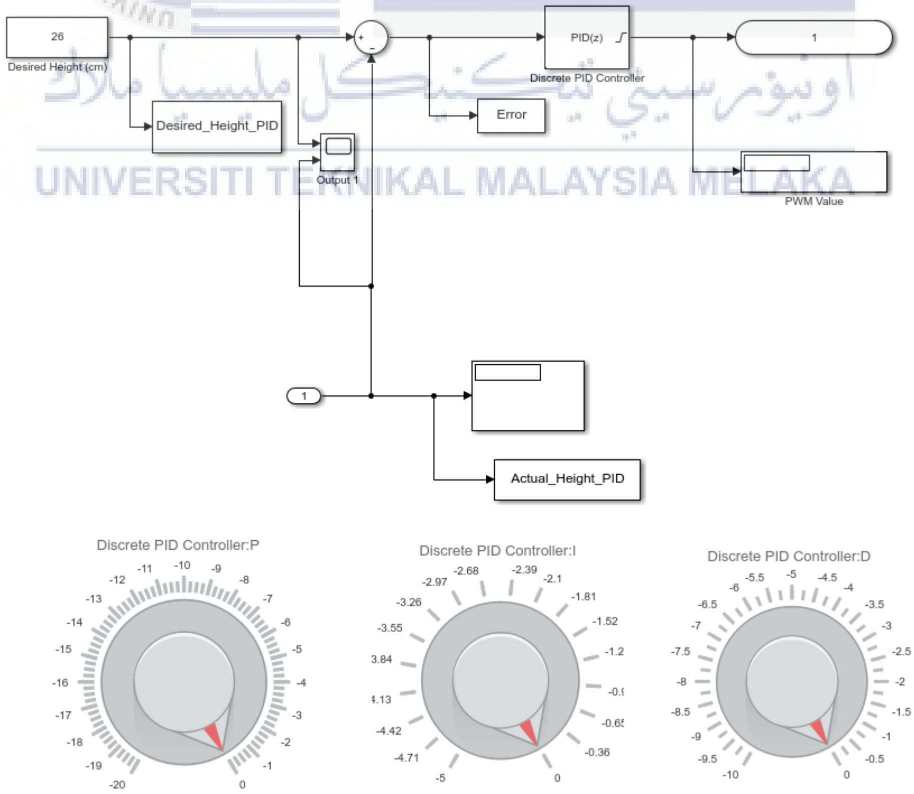


Figure 3.24: PID controller.

From Figure 3.24, a sum block is selected from the library browser in the Simulink platform. The sum block is assigned with the "+-" sign. A closed-loop system is created by connecting the set point or desired height block to the positive sign of the sum block, while the negative sign of the sum block is connected to the output of the ultrasonic sensor. The output of the ultrasonic sensor acts as a feedback path to monitor the actual value of the ping pong ball inside the tube. The output of the sum block is then connected to the PID controller. The created sum block is to compute the difference between the desired height and the actual height or output of the ultrasonic sensor. The difference of the values is then stored in the workspace with the variable name "error" by using the To Workspace block. The computed error will be sent to the PID controller and based on the values of Kp, Ki, and Kd that are applied to the PID controller, the output PWM value will be computed and sent to the pin D9 of the Arduino board.

After the development of PID controller is done, the PID controller is included in a subsystem named PID controller. The output of the gain block is connected to the input of the subsystem and the output of the PID controller subsystem is connected to the PWM pin D9 of the Arduino board as shown in Figure 3.25. The input and the output data of the system are exported to the workspace of the MATLAB platform by using two To Workspace blocks named desired height PID and actual height PID. The input and the output data are used to estimate the transfer function of the system. The mode of the simulation must be changed to external mode and the time is set to infinity in order to run the real-time simulation for the developed system with PID controller.

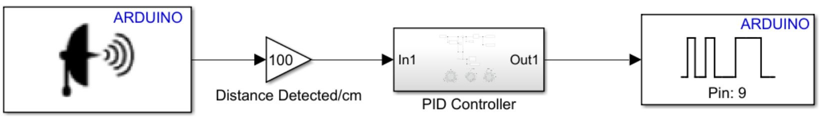


Figure 3.25: Designed Model for Air Levitation System with PID controller.

3.8 Transfer Function of the Systems

There are two approaches to obtain the transfer function of a system which are the system identification approach and the derivation of the general equation of the system. In this project, the system identification approach is used. There are a few steps that need to be taken to estimate the transfer function.

First, the system identification application is opened in the MATLAB platform as shown in Figure 3.26.

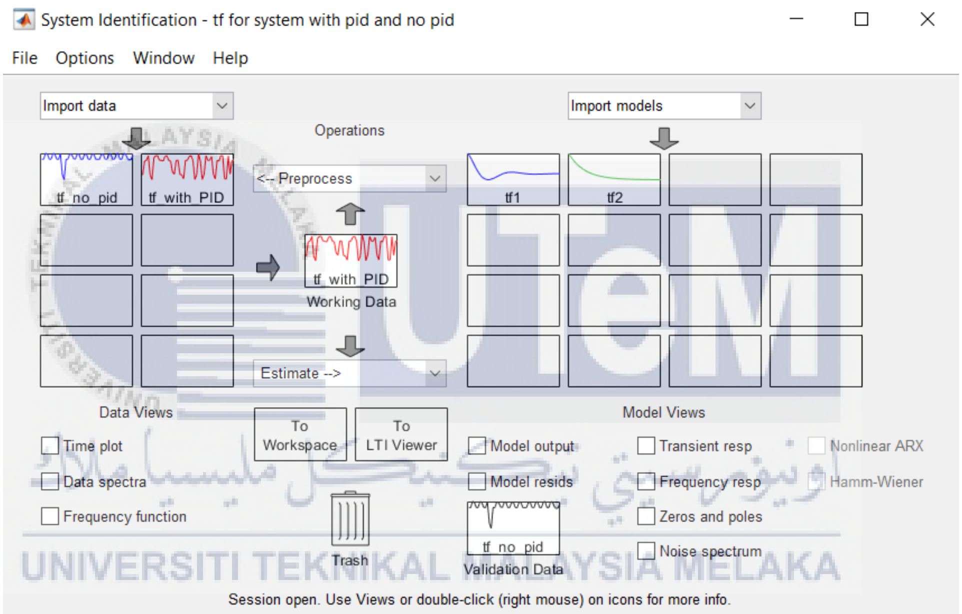


Figure 3.26: System Identification Application.

The import time domain data is selected from the application and the Graphical User Interface (GUI) for importing the data will exist as shown in Figure 3.27. The name of the variable for the input and output data must be filled according to the name of the variables stored in the workspace in the MATLAB platform.

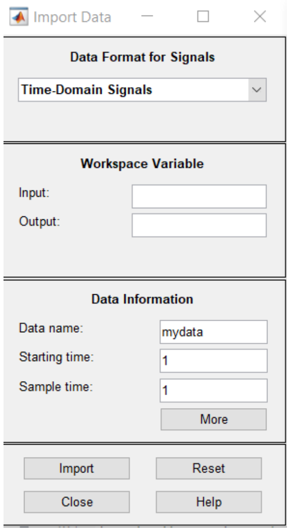


Figure 3.27: GUI for Importing Data.

After the data are successfully imported to the application, the estimate transfer function model is selected from the application and the GUI will appear as shown in Figure 3.28.

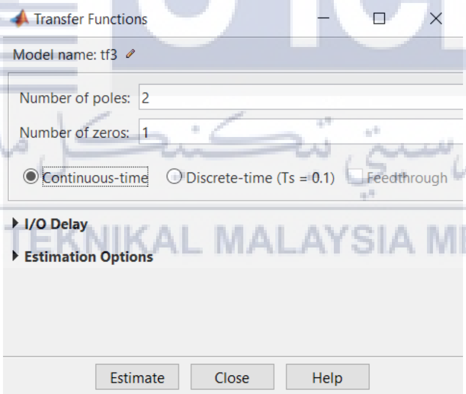


Figure 3.28: GUI for Estimating Transfer Function.

After all of the information is filled in, the estimate button is pressed and the application will start to estimate the transfer function as shown in Figure 3.29

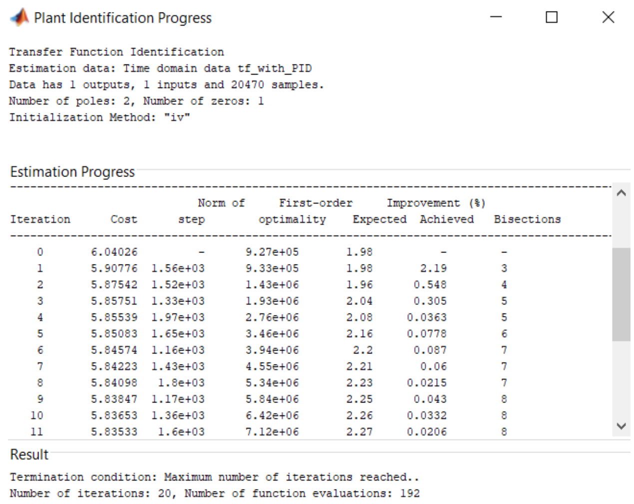


Figure 3.29: Process for Estimating Transfer Function.

Then, the model or the transfer function will be displayed on the right-hand side of the application after the estimating process is done. The obtained transfer function for both of the systems will be shown in Chapter 4.

3.9 Application of Tuning Method

The physical tuning method is chosen as the tuning method instead of the Ziegler Nicholas method due to the limitation of the length of the tube that is used in this project where the constant oscillation of the system cannot be produced. There are a few procedures for the physical tuning method. First of all, the integral and derivative terms are set to zero, and the value of proportional gain, Kp is increased until the output response of the system oscillates. When the Kp is increased, the rise time of the system will decrease without affecting the stability of a system. Next, when the faster response of the system is obtained through adjusting the value of Kp, the integral term, Ki of the controller is set to reduce the steady-state error of the system. When the integral term, Ki is increased the steady-state error of the system will be reduced but the oscillations will be increased. After the Kp and Ki have been set to the desired value with the minimum value of steady-state error. The derivative gain, Kd is used to reduce the oscillations of the system. When the value of Kd is increased ,the overshoot of the system is reduced [44].

3.10 Summary

In this chapter, the processes involved in developing the air levitation system without PID controller and with PID controller are discussed. Next, the hardware and the platform that are used in developing the project are explained. The PID tuning method that is selected for this project is discussed. Other than that, the schematic circuit with the wire connections for each of the hardware components and the final developed prototype are shown in this chapter to provide better visualization and understanding towards this project.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter will explain the working principle for both of the developed systems. The result or the response of the system obtained from the MATLAB platform will be analyzed and discussed.The project sustainability is discussed in this chapter.

4.2 Developed Prototype

Figure 4.1 shows the final developed prototype of the air levitation system. The ultrasonic sensor is attached at the top end of the acrylic transparent tube, and the ping pong ball is put inside the tube. The box that the tube is attached to has the Arduino board, blower fan, 9V battery, and IRF520 MOSFET driver module inside.

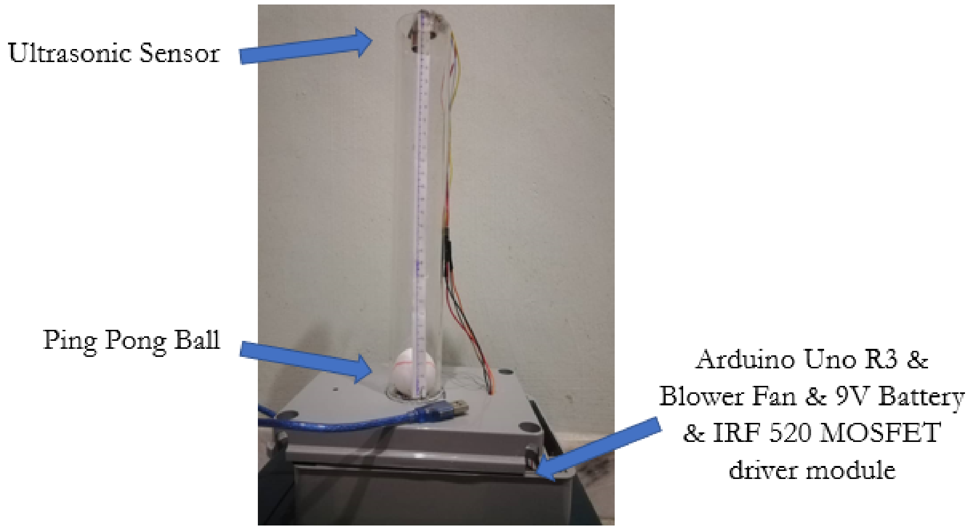


Figure 4.1: Final Developed Prototype.

4.3 Working Principle of the System without PID controller

The operation of the air levitation system without PID controller in the Arduino IDE platform has the same operation as in the Simulink platform. It is because the constructed model in the Simulink platform is based on the codes that are written in the Arduino IDE platform. The reason that the model of the air levitation system is transferred from the Arduino IDE platform to the MATLAB platform is the input and output data are unable to be exported to MATLAB software. In this case, the operation of the system will be explained based on the Arduino IDE platform.

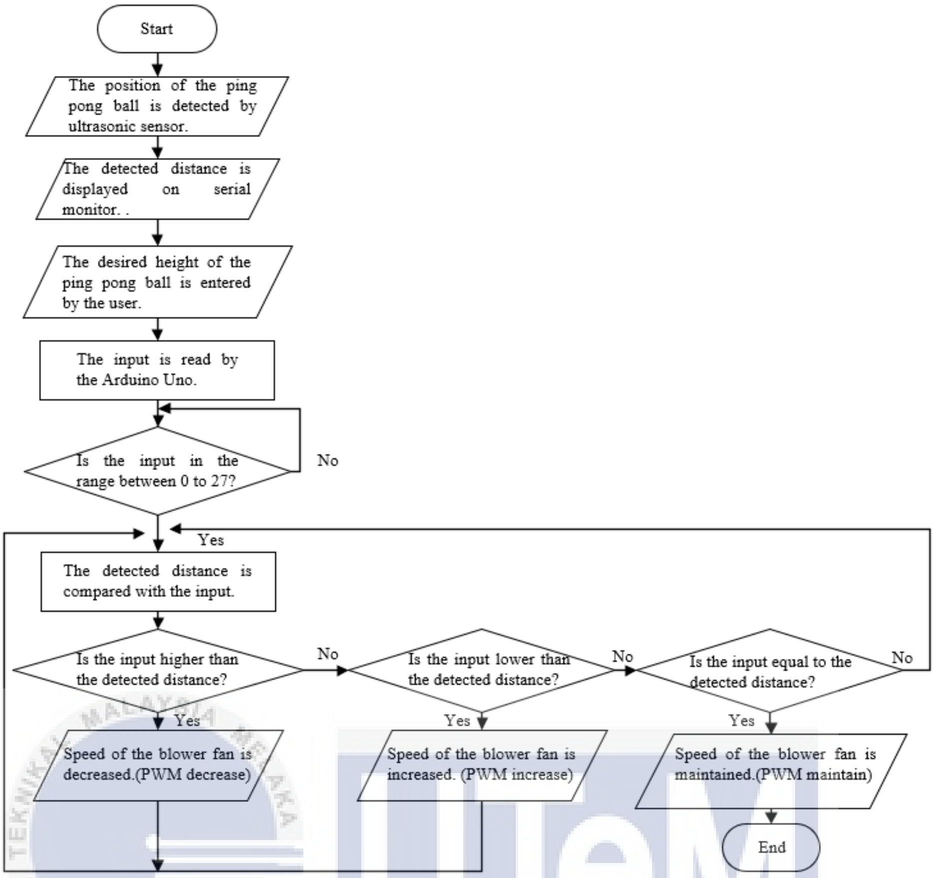


Figure 4.2: Operation of the System without PID controller.

From Figure 4.2, the operation of the system was started with measurement of the distance between the ultrasonic sensor and the ping pong ball. The value of the distance is calculated and displayed on the serial monitor. The input of the desired position of the ping pong ball can be entered by the user through the serial monitor in the Arduino IDE platform.

The value entered must be the integer and within the range of 0 between 27. It is due to the valid distance of the ping pong ball that can be lifted is from 0 to 27cm after the dimensions of the sensor and the ball are taking consideration. If the value entered by the user was out of the range assigned, a message would be printed on the window of the serial monitor, notifying the user to enter another valid input.

The entered input value would be compared with the measured distance and the message that shows the system is activated would be displayed if the value was within the range given as shown in Figure 4.3. Three conditions had been defined either the input is higher than, lower than, or equal to the detected distance, the appropriate actions were written for each of the conditions that fulfilled.

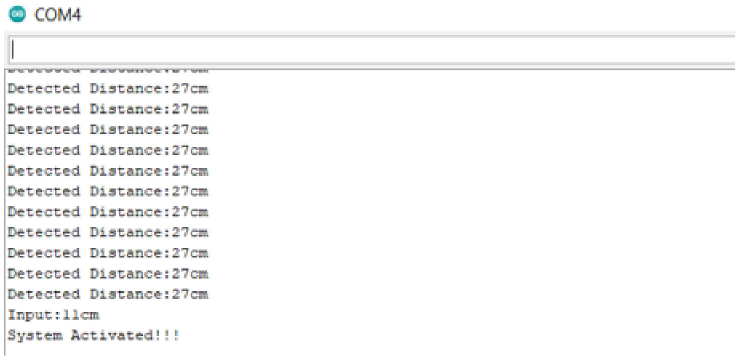


Figure 4.3: System Activated Message Displayed On Serial Monitor.

For the first condition, if the input value (desired height) was higher than the detected distance, the value of the PWM would be decreased which will decrease the speed of the blower by producing smaller airflow to move the lifting object downwards and the value of input and detected distance would be compared again. The decrease of the PWM value of the pin D9 would decrease the duty cycle of the signal and the output voltage was decreased. The loop was continued to check the input value and detected distance.

If the first condition was not fulfilled, the second condition was checked, if the input value was lower than the detected distance, the value of the PWM would be increased that would increase the speed of the blower by producing larger airflow to move the lifting object upwards and the value of input and detected distance would be compared again. The increase of the PWM value of the pin D9 would increase the duty cycle of the signal and the output voltage was increased. The loop was continued to check the input value and detected distance.

The third condition would be executed if the first and second conditions were not fulfilled, the current value of PWM would remain as long as the third condition is fulfilled. It is to maintain the duty cycle of the signal which the same airflow would be produced by the blower to lift the object at a fixed position. The checking process of the input value and the detected distance was continued. However, since the air levitation system is one of the non-linear and unstable systems, therefore without the application of control techniques such as PID control will cause the ping pong ball keeps moving upwards and downloads nonstop. Figure 4.4 shows the increment and decrement of the PWM value when the conditions were fulfilled and when the input was equal to the detected height the PWM value was maintained and displayed on the serial monitor.

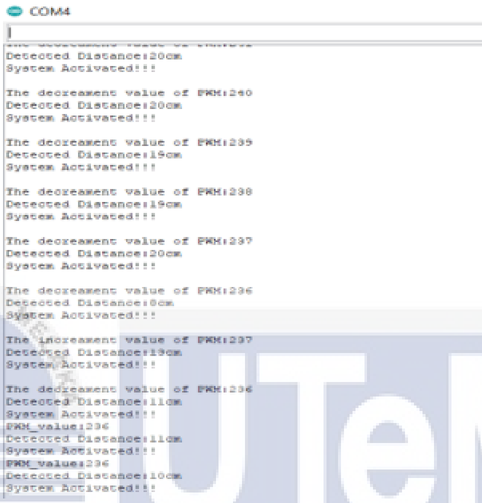


Figure 4.4: Increment and Decrement of PWM Value Displayed on Serial Monitor.

Figure 4.5 and Figure 4.6 show the simulated model that applied to the physical levitation system that has identical operation as the simulated system in the Arduino IDE platform.

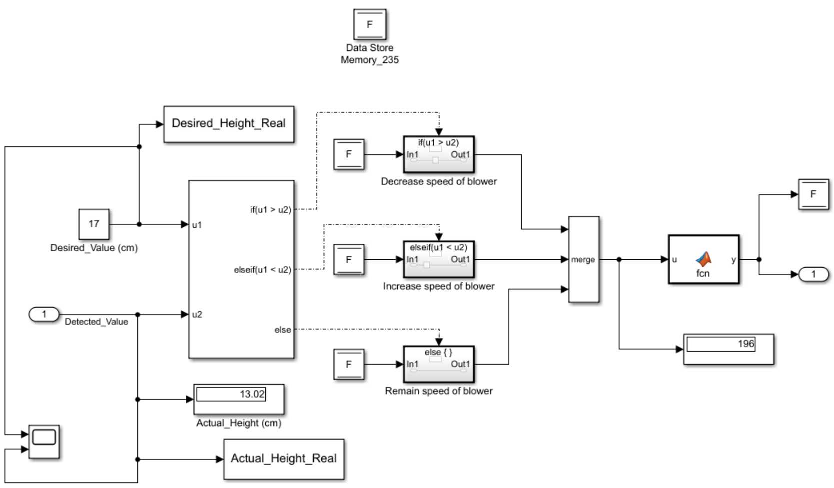


Figure 4.5: Simulated model in MATLAB platform.

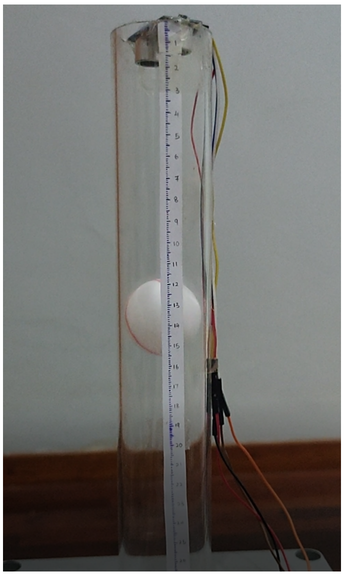


Figure 4.6: Application of the simulated model to the physical system.

4.4 Working Principle of the System with PID controller

For the operation of the air levitation system with PID controller, the output of the PID controller is connected to the PWM pin D9 of the Arduino boards. The physical tuning method that has been discussed in Chapter 3 is used to obtain the optimal value of the Kp, Ki, and Kd. The PID parameters are tuned using the knob block. The optimal values obtain for the Kp, Ki, and Kd after the tuning method is applied are equal to -7.64, -2.71, and -2.3 as shown in Figure 4.7.

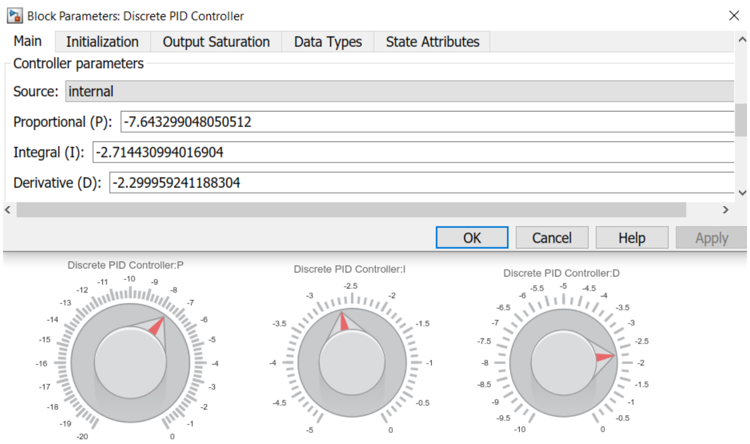


Figure 4.7: Optimal Values for Kp, Ki, and Kd.

After the values are obtained, the values are applied to the PID controller directly. The model is then simulated in the external mode to allow the real-time simulation of the physical system.

The operation of the PID controller is based on the equation 2.11, the $e(t)$ represents the error that is computed from the difference between desired height and the actual height of the ball. When the value of the Kp is increased, it will increase the rise time of the response where the error of the response is increased as well. To reduce the steady-state error of the response, the Ki is required. The value of error will be integrated over time and multiplied to the value of Ki that is applied to the controller to reduce the steady-state error. The presence of the Kp and Ki will cause a large overshoot for the response, and it required Kd to reduce the overshoot of the system by differentiating the error in respect with time to predict the future behavior of the error. The Figure 4.8 and Figure 4.9 show the simulated model that applied to the physical levitation system.

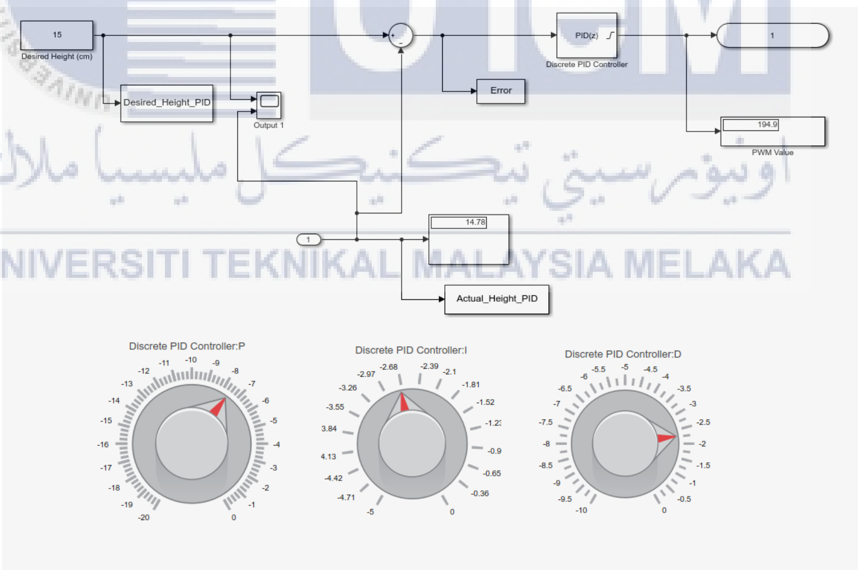


Figure 4.8: Simulated model in Simulink platform.

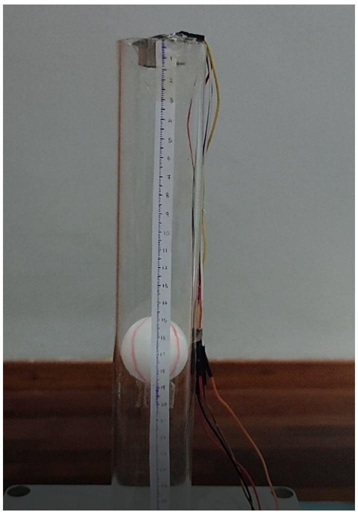


Figure 4.9: Application of the simulated model to the physical system.

4.5 Graph Obtained For Both Of the Systems

The graph that consists of the actual position of the ping pong ball and the desired height that is entered by the user for the air levitation system without the PID controller can be observed through the scope in the Simulink platform as shown in Figure 4.10. The scope must be connected to the output of the desired height and the output from the ultrasonic sensor.

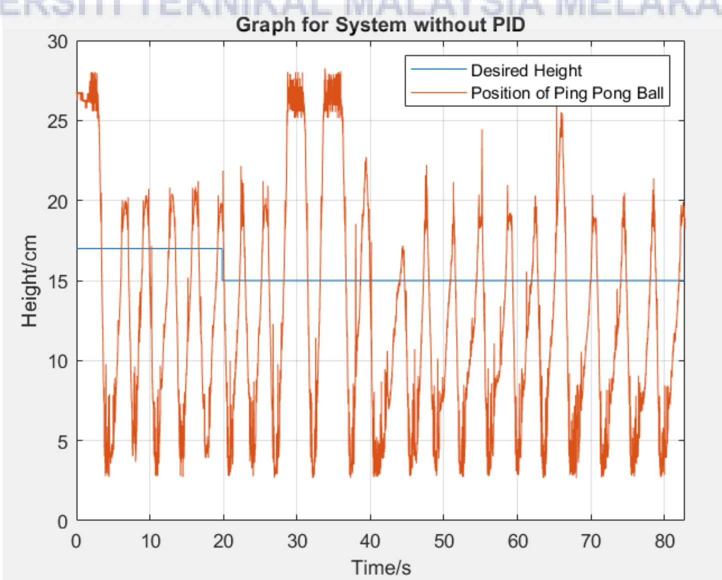


Figure 4.10: Graph for the system without PID controller.

From Figure 4.10, when time is at 0s the desired height is set to 17cm and the position of the ping pong is keeps fluctuating in the range from 5cm to 20cm. Then, the desired height is changed to 15cm when the time is equal to 20s, it is observed that the position of the ping pong ball is kept oscillating. This indicates that the air levitation system without PID controller is unstable and unable to stabilize the ball at the desired height. In real life, this is the unwanted response from a system. While the graph for the air levitation system with PID controller also can be observed from the scope that is connected both of the variables as shown in Figure 4.11.

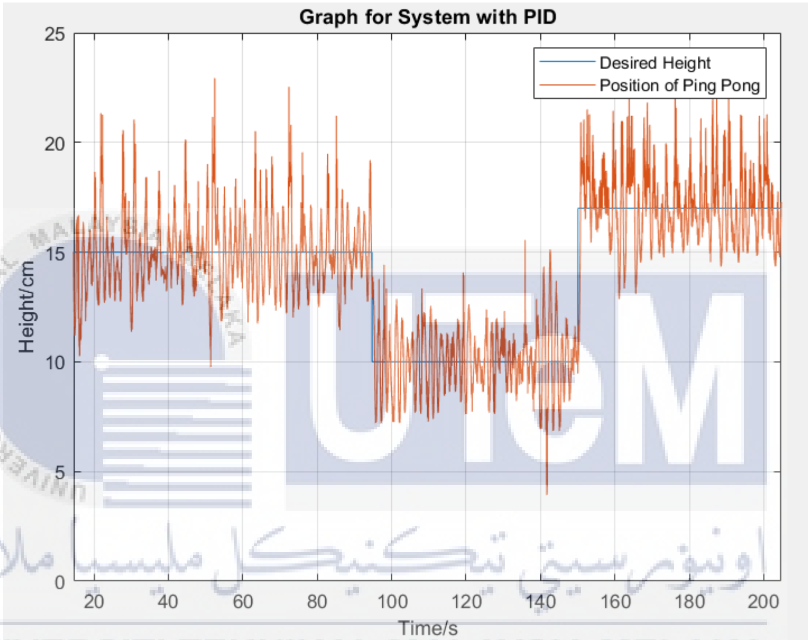


Figure 4.11: Graph for the system with PID controller.

From the graph for the system with PID controller, when the time is equal to 20s the desired height is set to 15cm by the user and the position of the ping pong is oscillating with smaller amplitude if compared with that for the system without PID controller. The desired height is then changed to 10cm when the time is equal to 95s, the position of the ping pong ball is fluctuating from 8cm to 12cm. After that, the desired height is changed to 17cm and it is observed that the position of the ping pong increase to 17cm and have some oscillations. Unlike the system without PID controller, the output of the system with PID controller can track the input that is entered by the user as shown in Figure 4.11.

4.6 Transfer Function Obtained For Both of the Systems

The transfer function for both of the systems is obtained by using the system identification toolbox in the MATLAB platform. To estimate the function using the System Identification Toolbox, the input and output data of the system are required.

For the transfer function of the air levitation system without PID controller that is obtained using system identification approach is shown in Figure 4.13 or equation 4.1.

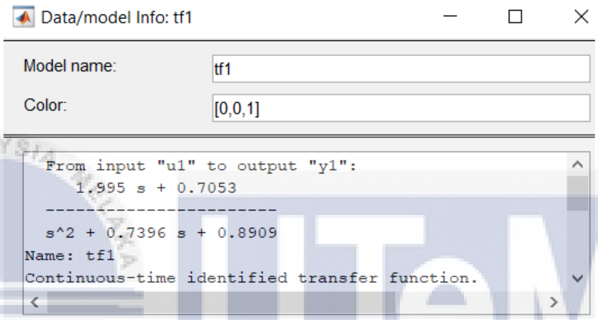


Figure 4.12: Transfer Function of the system without PID Controller.

$$tf1 = \frac{1.995s + 0.7053}{s^2 + 0.7396s + 0.8909} \tag{4.1}$$

For the transfer function of the air levitation system with PID controller that is obtained using system identification approach is shown in Figure 4.13 or equation 4.2.

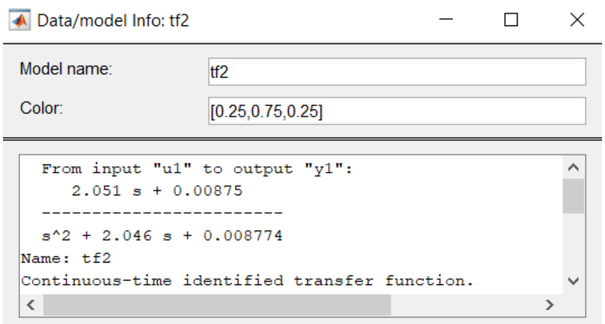


Figure 4.13: Transfer Function of the system with PID Controller.

$$tf2 = \frac{2.051s + 0.00875}{s^2 + 2.046s + 0.008774} \tag{4.2}$$

There are two transfer functions are obtained in this project, the tf1 is the transfer function or the model that represents the air levitation system. While the tf2 is the transfer function that has been included with the model of the developed system and the PID controller which means that the model of the system has been multiplied with the equation of the PID controller which produces the tf2.

4.7 Step Response of the Systems

The step responses for both of the systems are generated by applying a step input to the obtained transfer function. In this project, the "step" command is used to generate the step response of the system in the MATLAB platform. The default value of the step input that is applied to the system by using the "step" command is 1. Figure 4.14 shows the generated step response for the system without PID controller.

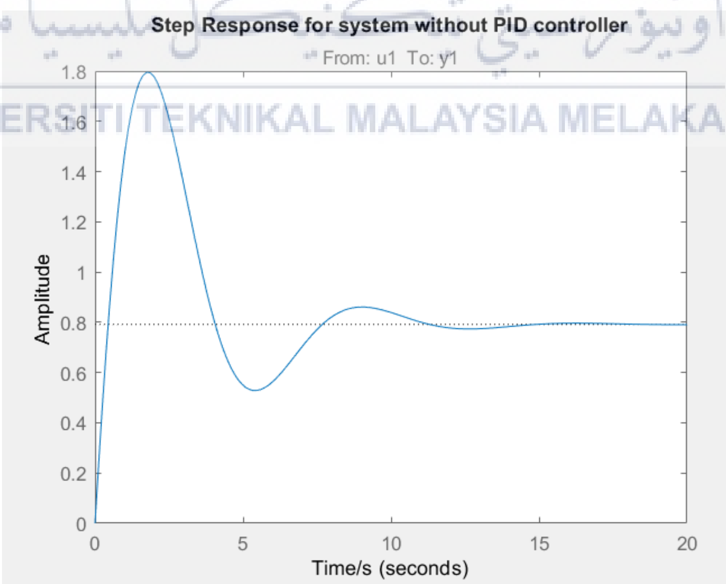


Figure 4.14: Step Response of the System without PID Controller.

From the step response of the system without PID controller, it is clear to observe that the response has a high percentage of overshoot at time = 1.74s and the system oscillates before it reaches the steady-state phase. During the estimation process of the transfer function for the system without PID, the stability is enforced and it results the step response of the system to enter steady-state phase instead of oscillation due to the limitation of the System Identification Toolbox. Figure 4.15 shows the generated step response for the system with PID controller.

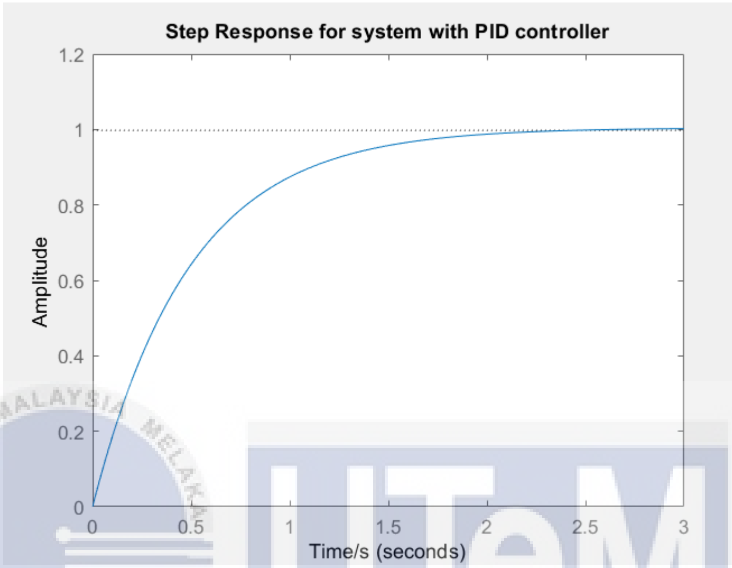


Figure 4.15: Step Response of the System with PID Controller.

While for the step response of the system with PID controller, it is observed that there is no overshoot and oscillation occur for the system. The system reaches the steady-state phase when time = 2 s.

Furthermore, the transfer function of the system with PID controller is applied with multiple-step inputs as shown in Figure 4.16 which start with 15cm, 10cm, and 17cm as shown in Figure 4.17.

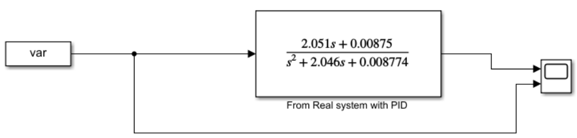


Figure 4.16: Application of Multiple Step Inputs to the Obtained Transfer Function.

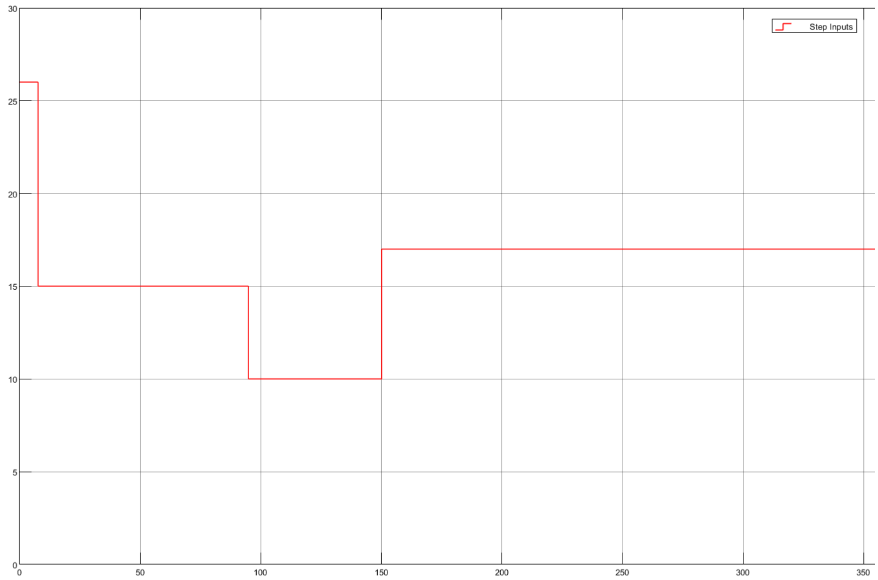


Figure 4.17: Multiple Step Inputs.

The scope in the Simulink platform is used to observe the step response of the system when multiple step inputs are applied to it as shown in Figure 4.18. The yellow line is the step response of the system and the blue line is the steps inputs that are applied. Initially, the inputs are set to 15cm and change to 10cm when time is equal to 94.771s. At time = 150.196s, the input is changed to 17cm. It is obvious to see that the inputs that are applied to the system can be tracked by the developed system with PID controller. This indicates that the controller that is applied to the system is working.

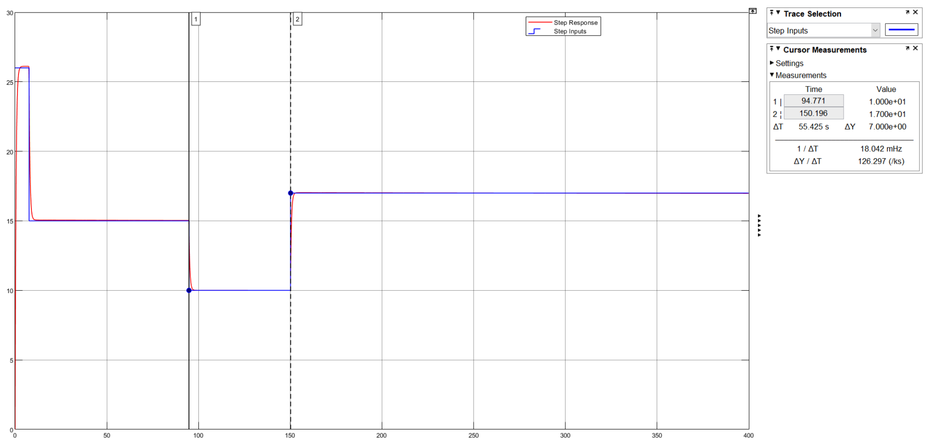


Figure 4.18: Step Response for Multiple Step Inputs.

4.8 Analysis of Step Response

The parameters such as the steady-state error, steady-state value, percentage overshoot, rise time and settling time can be determined from the generated step response for both of the systems as shown in Figure 4.19 and Figure 4.20 in the MATLAB platform.

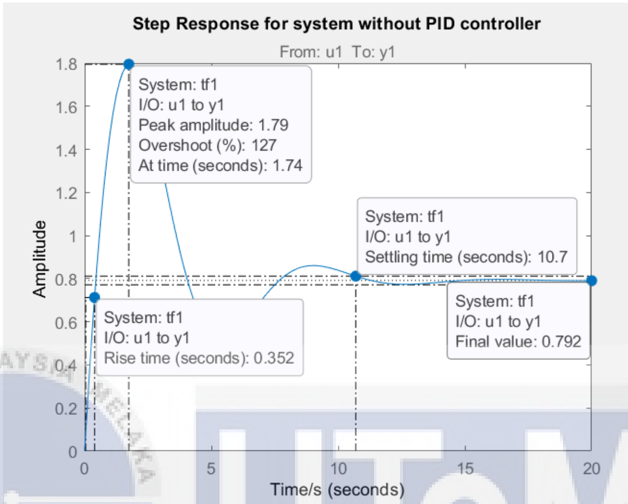


Figure 4.19: Parameters Obtained for the step response of the system without PID Controller.

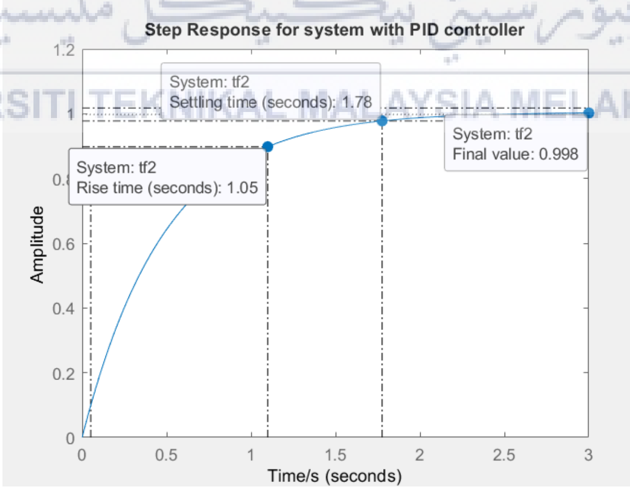


Figure 4.20: Parameters Obtained for the step response of the system with PID Controller.

Based on the Figure 4.19 and Figure 4.20, the step responses for both of the systems are analyzed through the parameters that are obtained as shown in Table 4.1.

Table 4.1: Parameters Obtained of the Step Response from both of the systems

Parameters	System without PID	System with PID
Percentage Overshoot,$\%OS$	127%	0%
Rise Time,T_r	0.352s	1.05s
Settling time,T_s	10.7s	1.78s
Steady-State Value	0.792	0.998
Steady-State Error	1-0.792 = 0.208	1-0.998 = 0.02

From table 4.1, the percentage overshoot for the system without PID is 127% and the percentage overshoot for the system with PID is 0% where there is no overshoot occurs. The high percentage overshoot indicates that the system is unstable. The rise time of the system without PID controller and system with PID controller is equal to 0.352s and 1.05s respectively. The settling time of the system without PID controller is 10.7s which is longer than that for the system with PID controller which is 1.78s. The steady-state value of the system without PID controller is 0.792, while the steady-state value of the system with PID controller is 0.998. The steady-state errors for both of the systems are calculated, the steady-state error of the system without PID controller is equal to 0.208 and the steady-state error of the system with PID controller is equal to 0.02. The percentage errors of the steady-state error parameters are tabulated as shown in Table 4.2.

Table 4.2: Percentage Errors of Steady-State Errors

Types of System	Percentage Error for steady-state error parameter(%)
System without PID	$\frac{1-0.792}{1.000} \times 100\% = 20.8\%$
System with PID	$\frac{1-0.998}{1.000} \times 100\% = 2.0\%$

After comparing both of the parameters of the step responses, it is clear to see that the system with PID controller has a better response in terms of low percentage of overshoot, low settling time, and low steady-state error if compare with the system without the PID controller although the rise time of the system is higher than the system without PID controller. The steady-state error of the system without PID has been improved from 20.8% to 2% if compared with the system with PID controller. The percentage of overshoot also has been improved from 127% to 0%. The stability of the air levitation system is increased with the application of PID controller to the system. The concept of implementing the PID controller to increase the stability of a system also can be applied to other non-linear and unstable systems such as the magnetic levitation system.

4.9 Project Sustainability

The developed air levitation system is easier to be designed and replicated. All of the components that are used to develop the system can be obtained easily. The MATLAB and Simulink platforms that are used to develop this system are sustainable where the platform can be accessed by anyone and the future improvement of the design of the controller can be done in the platform. Next, the other type of controller such as the Sliding Mode Controller can be applied to the air levitation system and the comparison of the effectiveness between the controllers applied can be made. The developed system also can be modified based on the requirement of the user such as making it to become a training toolkit for control engineering education by adding some hands-on lab activities.

4.10 Summary

In this chapter, the result that obtained for this project and the working principle for both of the systems are discussed. The graphs that consist of the desired height and the actual height of the ping pong ball are included with explanation. The step responses for both of the systems are analyzed and compared. The project sustainability is discussed in this chapter.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Introduction

This chapter will make the conclusion based on the research study that was made, and the future improvement that can be done on this study are suggested.

5.2 Conclusion

In conclusion, a low-cost air levitation system is developed by using the Arduino Uno R3, ultrasonic sensor, blower fan, and IRF520 MOSFET driver module. The MATLAB and Simulink platforms are used to design the model of the system and PID controller that will be deployed on the physical hardware. The analysis of the step responses for both of the systems is carried out based on the parameters such as percentage overshoot, rise time, settling time, and steady-state error.

This project demonstrates a control technique which is a PID controller that is applied to a physically developed air levitation system. The comparison is made between the system without the PID controller and the system with PID controller with the purpose to study the effect of the PID controller on the stability of the system. The air levitation system that is chosen for this project is one of the non-linear and unstable systems. The transfer functions for both of the systems are estimated by using the system identification application. The step responses for both of the systems are generated and analyzed. It is proved that the stability of the air levitation system can be increased with the application of PID controller where the percentage overshoot, settling time, and the steady-state error of the system are greatly reduced.

5.3 Future Works

The air levitation system is one of the popular non-linear systems used by other researchers recently for studying the performance of applied controllers due to its simplicity to set up and easier to duplicate. For the future improvement of this project, the non-linear controller such as Sliding Mode Controller (SMC) or fuzzy logic controller can be applied to the developed air levitation system. The performance of the SMC can be used to compare with the PID controller to determine which controller has better robust control to the system. Next, the developed system can be modified to make it becomes a training toolkit in control engineering education by adding more lab activities such as the application of different control techniques (PI or PD) and disturbance analysis. This can provide hands-on activities for the student or control engineer to relate the theoretical knowledge that learned with the practical.

REFERENCES

[1] M. S. Dhaka and D. A. Rehalia, "Importance and Scope of Control Systems," *IJA-ERA-PapersOnLine*, vol. 2, no. 7, pp. 515–518, 2016, International Journal of Advanced Engineering Research and Applications.

[2] "PID Control," November 2021. [Online]. Available: https://drstienecker.com/tech-332/7-pid-control/

[3] A. K. Shri Bhagwan and J. S. Soni, "A Review on: PID Controller," vol. 3, pp. 17–22, February 2016, International Journal on Recent Technologies in Mechanical and Electrical Engineering.

[4] H. Bansal, R. Sharma, and S. Ponpathirkoottam, "PID Controller Tuning Techniques: A Review," vol. 2, pp. 168–176, November 2012.

[5] J. Chacon, J. Sáenz, L. D. L. Torre, J. M. Díaz, and F. Esquembre, "Design of a Low-cost Air Levitation System for Teaching Control Engineering," *Sensors (Basel, Switzerland)*, vol. 17, 2017.

[6] "Developing FPGA-based Embedded Controllers using Matlab/Simulink," November 2021. [Online]. Available: https://www.intechopen.com/chapters/10841

[7] A. Burduk, *Stability Analysis of the Production System Using Simulation Models*, June 2014, pp. 69–83.

[8] S. Saat, S. K. Nguang, and A. Nasiri, "Chapter 1 - introduction," in *Analysis and Synthesis of Polynomial Discrete-Time Systems*, S. Saat, S. K. Nguang, and A. Nasiri, Eds. Butterworth-Heinemann, 2017, pp. 1–27. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780081019016000013

[9] J. Chacón, H. Vargas, S. Dormido, and J. Sánchez, "Experimental Study of Nonlinear PID Controller in an Air Levitation System," *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 304–309, 2018, 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318303781

[10] P. Timmerman and K. Weele, "On the rise and fall of a ball with linear or quadratic drag," *American Journal of Physics - AMER J PHYS*, vol. 67, pp. 538–546, June 1999.

[11] J. Escaño, M. Ortega, and F. Rubio, "Position Control of a Pneumatic Levitation System." January 2005.

[12] S. Jernigan, Y. Fahmy, and G. Buckner, "Implementing a Remote Laboratory Experience Into a Joint Engineering Degree Program: Aerodynamic Levitation of a Beach Ball," *Education, IEEE Transactions on*, vol. 52, pp. 205 – 213, June 2009.

[13] "What is a PID Controller : Working Its Applications." [Online]. Available: https://www.elprocus.com/the-working-of-a-pid-controller/

[14] "What is a PID Controller?" 2019. [Online]. Available: https://www.omega.com/en-us/resources/pid-controllers

[15] B. Parvat, S. Lukare, and S. Pandit, "Model-based PID Controller for SISO and MIMO System Control," vol. 10, pp. 9–16, January 2019.

[16] L. Desborough and R. Miller, "Increasing Customer Value of Industrial Control Performance Monitoring -Honeywell's Experience," *AIChE Symposium Series*, vol. 98, pp. 172–192, January 2002.

[17] A. S. Bazanella, L. F. A. Pereira, and A. Parraga, "A New Method for PID Tuning Including Plants Without Ultimate Frequency," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 2, pp. 637–644, 2017.

[18] B. K. Roy and S. Sisodiya, "Speed Control of DC Shunt Motor with Comparative Analysis with PI and Fuzzy Controller Approach," *International Journal of Modern Trends in Engineering and Research*, vol. 3, 2016.

[19] K. Ogata, *Modern Control Engineering*, ser. Instrumentation and controls series. Prentice Hall, 2010. [Online]. Available: https://books.google.com.my/books?id=Wu5GpNAelzkC

[20] *PID Tuning via Classical Methods*, March 2021. [Online]. Available: https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Book%3A_Chemical_Process_Dynamics_and_Controls_(Woolf)/09%3A_Proportional-Integral-Derivative_(PID)_Control/9.03%3A_PID_Tuning_via_Classical_Methods

[21] M. Sarif, D. A. Kumar, and M. V. G. Rao, "Comparison Study of PID Controller Tuning using Classical / Analytical Methods," 2018.

[22] J. M. S. Ribeiro, M. F. Santos, M. J. Carmo, and M. F. Silva, "Comparison of PID Controller Tuning Methods: analytical/classical techniques versus optimization algorithms," in *2017 18th International Carpathian Control Conference (ICCC)*, 2017, pp. 533–538.

[23] H. Cohen and G.A.Coon, "Theoretical Consideration of Related Control," vol. 75, pp. 817–834, 1953.

[24] "The Step Response," March, [Online; accessed 2021-12-26]. [Online]. Available: https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Book%3A_Introduction_to_Control_Systems_(Iqbal)/02%3A_Transfer_Function_Models/2.04%3A_System_Response_to_Inputs

[25] "Time Domain Specifications," November 2021. [Online]. Available: https://www.tutorialspoint.com/control_systems/control_systems_time_domain_specifications.htm

[26] S. Howimanporn, S. Chookaew, and C. Silawatchananai, "Experimental and implementation of robust control via floating air levitation and balancing rotary inverted pendulum," *Journal of Physics: Conference Series*, vol. 1733, p. 012013, January 2021.

[27] J. Chacon, L. de la Torre, and S. Dormido, "The Air Levitation System," *IFAC-PapersOnLine*, vol. 52, no. 9, pp. 33–35, 2019, 12th IFAC Symposium on Advances in Control Education ACE 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319304549

[28] R. Heradio, L. de la Torre Cubillo, D. Galan, F. Cabrerizo, E. Herrera-Viedma, and S. Dormido, "Virtual and Remote Labs in Education: A Bibliometric Analysis," *Computers Education*, vol. 98, pp. 14–38, July 2016.

[29] J. R. Brinson, "Learning Outcome Achievement in Non-Traditional (Virtual and Remote) versus Traditional (hands-on) Laboratories: A Review of the Empirical Research," *Computers Education*, vol. 87, pp. 218–237, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360131515300087

[30] E. Cholodowicz and P. Orlowski, "Low-cost Air Levitation Laboratory Stand using MATLAB/Simulink and Arduino," 2017.

[31] F. Isdaryani, F. Feriyonika, and R. Ferdiansyah, "Comparison of Ziegler-Nichols and Cohen Coon Tuning Method for Magnetic Levitation Control System," *Journal of Physics: Conference Series*, vol. 1450, p. 012033, February 2020.

[32] "Ultrasonic ranging module hc-sr04," November 2021. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

[33] "Sharp gp2y0a21yk data sheet," November 2021. [Online]. Available: https://usermanual.wiki/Document/GP2Y0A21YKDATASHEET.973271779/view

[34] "Arduino datasheet," November 2021. [Online]. Available: https://datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf

[35] "Arduino UNO Pinout, Specifications, Board Layout, Pin Description," November 2021. [Online]. Available: https://www.electronicshub.org/arduino-uno-pinout/

[36] "Complete Guide for Ultrasonic Sensor HC-SR04 with Arduino," November 2021. [Online]. Available: https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/

[37] "Introduction to HC-SR04 (Ultrasonic Sensor)," November 2021. [Online]. Available: https://www.theengineeringprojects.com/2018/10/introduction-to-hc-sr04-ultrasonic-sensor.html

[38] "Interfacing IRF520 MOSFET Driver Module (HCMODU0083) with Arduino," November 2021. [Online]. Available: https://electropeak.com/learn/interfacing-irf520-mosfet-driver-module-switch-button-hcmodu0083-with-arduino/

[39] "Datasheet for irf520 mosfet," November 2021. [Online]. Available: https://www.alldatasheet.com/datasheet-pdf/pdf/250767/VISHAY/IRF520.html

[40] "Arduino IDE," November 2021. [Online]. Available: https://www.arduino.cc/en/software

[41] "Mathworks matlab grader," November 2021. [Online]. Available: https://www.celt.iastate.edu/learning-teaching-technology/mathworks-matlab-grader/

[42] "What is MATLAB?" November 2021. [Online]. Available: https://www.mathworks.com/discovery/what-is-matlab.html

[43] "What is Simulink?" November 2021. [Online]. Available: https://www.mathworks.com/videos/simulink-overview-61216.html

[44] "Pid theory explained," November 2021. [Online]. Available: https://www.ni.com/en-my/innovations/white-papers/06/pid-theory-explained.html

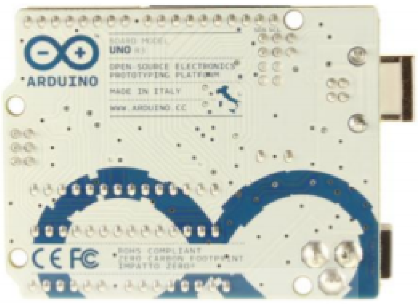
APPENDICES

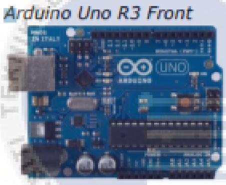
APPENDIX A

Datasheet for Arduino Uno

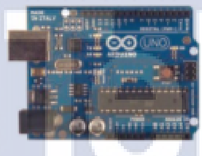
Arduino Uno

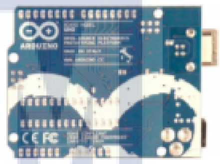

Arduino Uno R3 Front


Arduino Uno R3 Back


Arduino Uno R2 Front


Arduino Uno SMD


Arduino Uno Front


Arduino Uno Back

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.
Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.
Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer)
Schematic: arduino-uno-Rev3-schematic.pdf
Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.
External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.
The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.
The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.

APPENDIX B

Datasheet For Ultrasonic Sensor HC-SR04

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:
(1) Using IO trigger for at least 10us high level signal,
(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

Electric Parameter

Working Voltage	**DC 5 V**
Working Current	**15mA**
Working Frequency	**40Hz**
Max Range	**4m**
Min Range	**2cm**
MeasuringAngle	**15 degree**
Trigger Input Signal	**10uS TTL pulse**
Echo Output Signal	**Input TTL lever signal and the range in proportion**
Dimension	**45*20*15mm**

APPENDIX C

Output Response Characteristic Graph For Sharp GP2Y0A21YK

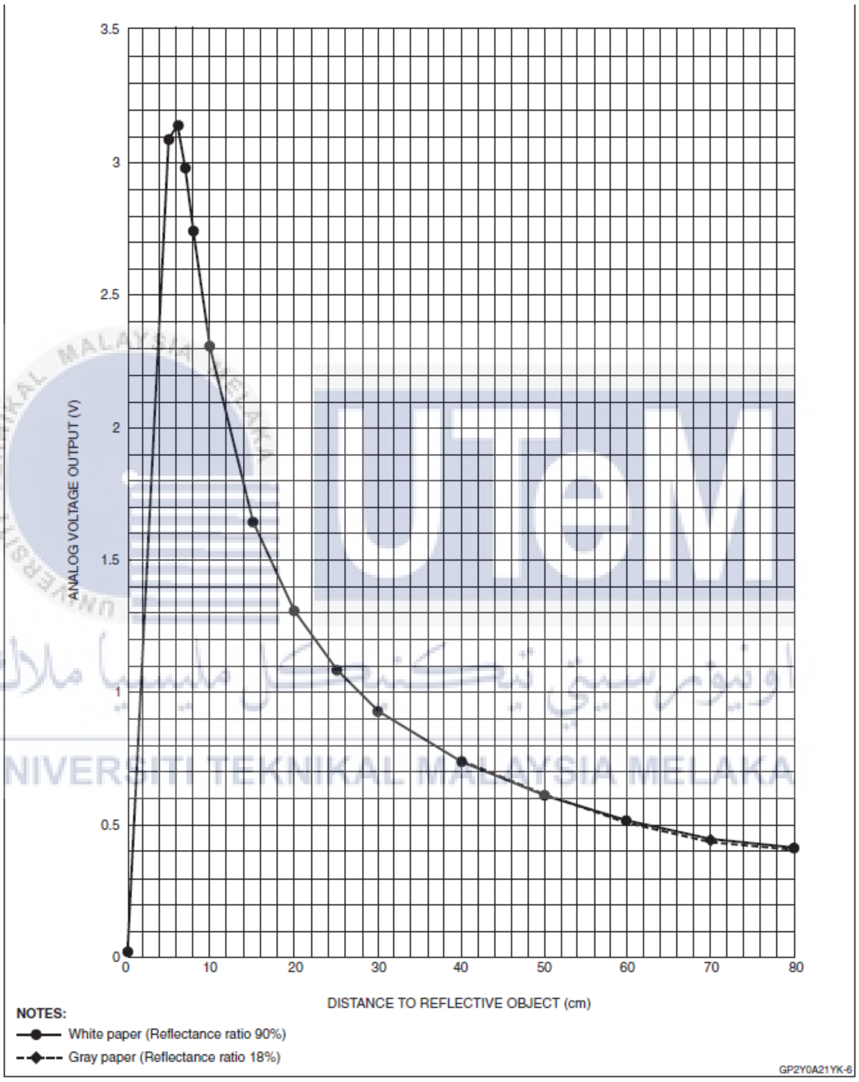


Figure 4. GP2Y0A021YK Example of Output Distance Characteristics

APPENDIX D

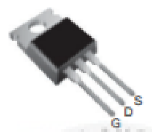
Datasheet for IRF520 MOSFET

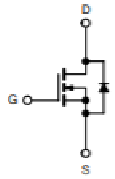
VISHAY.
www.vishay.com

IRF520
Vishay Siliconix

Power MOSFET

TO-220AB





N-Channel MOSFET

PRODUCT SUMMARY		
V_{DS} (V)	100	
$R_{DS(on)}$ (Ω)	V_{GS} = 10 V	0.27
Q_g max. (nC)	16	
Q_{gs} (nC)	4.4	
Q_{gd} (nC)	7.7	
Configuration	Single	

FEATURES
- Dynamic dV/dt rating
- Repetitive avalanche rated
- 175 °C operating temperature
- Fast switching
- Ease of paralleling
- Simple drive requirements
- Material categorization: for definitions of compliance please see www.vishay.com/doc?99912

RoHS*
HALOGEN FREE

Note
* This datasheet provides information about parts that are RoHS-compliant and / or parts that are non RoHS-compliant. For example, parts with lead (Pb) terminations are not RoHS-compliant. Please see the information / tables in this datasheet for details

DESCRIPTION

Third generation power MOSFETs from Vishay provide the designer with the best combination of fast switching, ruggedized device design, low on-resistance and cost-effectiveness.

The TO-220AB package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 W. The low thermal resistance and low package cost of the TO-220AB contribute to its wide acceptance throughout the industry.

ORDERING INFORMATION	
Package	TO-220AB
Lead (Pb)-free	IRF520PbF
Lead (Pb)-free and halogen-free	IRF520PbF-BE3

ABSOLUTE MAXIMUM RATINGS (T_C = 25 °C, unless otherwise noted)

PARAMETER			SYMBOL	LIMIT	UNIT
Drain-source voltage			V_{DS}	100	V
Gate-source voltage			V_{GS}	± 20	
Continuous drain current	V_{GS} at 10 V	T_C = 25 °C	I_D	9.2	A
		T_C = 100 °C		6.5	
Pulsed drain current [a]			I_{DM}	37	
Linear derating factor				0.40	W/°C
Single pulse avalanche energy [b]			E_{AS}	200	mJ
Repetitive avalanche current [a]			I_{AR}	9.2	A
Repetitive avalanche energy [a]			E_{AR}	6.0	mJ
Maximum power dissipation	T_C = 25 °C		P_D	60	W
Peak diode recovery dV/dt [c]			dV/dt	5.5	V/ns
Operating junction and storage temperature range			T_J, T_{stg}	-55 to +175	°C
Soldering recommendations (peak temperature) [d]	For 10 s			300	
Mounting torque	6-32 or M3 screw			10	lbf · in
				1.1	N · m

Notes

TYPICAL CHARACTERISTICS (25 °C, unless otherwise noted)

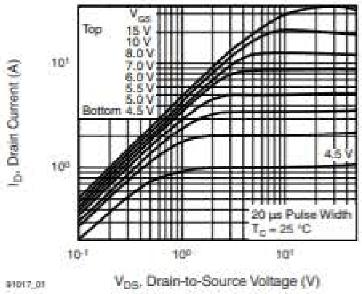


Fig. 1 - Typical Output Characteristics, T_C = 25 °C

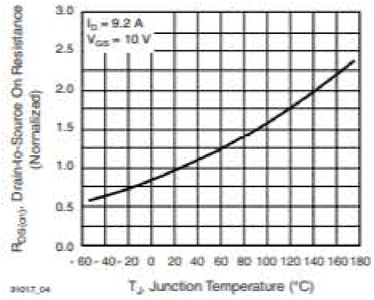


Fig. 4 - Normalized On-Resistance vs. Temperature

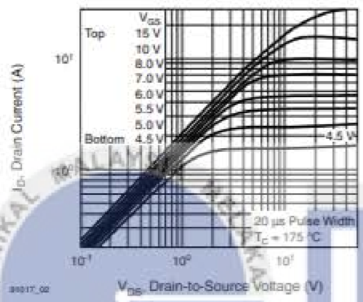


Fig. 2 - Typical Output Characteristics, T_C = 175 °C

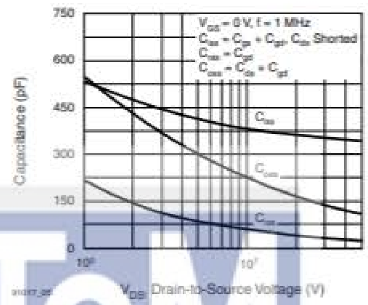


Fig. 5 - Typical Capacitance vs. Drain-to-Source Voltage

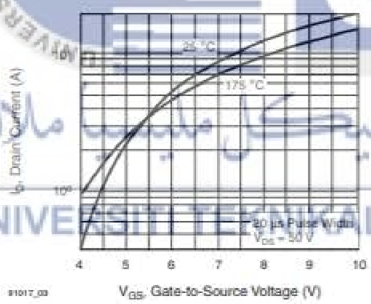


Fig. 3 - Typical Transfer Characteristics

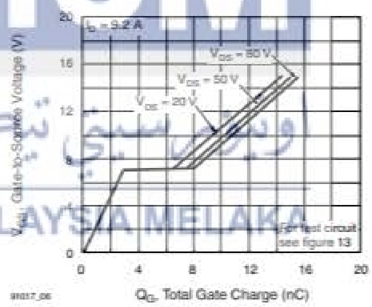


Fig. 6 - Typical Gate Charge vs. Gate-to-Source Voltage