Sudan University of Science and Technology

College of Engineering

Electrical Engineering

DESIGN AND IMPLEMENTATION OF BALL AND PLATE CONTROL SYSTEM USING PID CONTROLLER

تصميم وتنفيذ نظام تحكم في الكرة والشاشة بإستخدام المتحكم التناسبي التكاملي التفاضلي

A Project Submitted In Partial Fulfillment for the Requirements of the Degree of B.Sc. (Honor) In Electrical Engineering (Control)

Prepared By:

1- Mohammed Babiker Abdullah Hussien
2- Mohamed Gamal Eldeen Mohamed Yousif
3- Abdullah Omer Haj Ali Altoum
4- Mohammed Faisal Ali Abdullraheem

Supervised By:
Dr. Awadalla Taifour Ali

October 2017

الآية

قال تعالى :

{لِلَّهِ مَا فِي السَّمَوَاتِ وَمَا فِي الْأَرْضِ وَإِنْ تُبْدُوا مَا فِي أَنْفُسِكُمْ أَوْ تُخْفُوهُ يُحَاسِبْكُمْ بِهِ اللَّهُ فَيَغْفِرُ لِمَنْ يَشَاءُ وَيُعَذِّبُ مَنْ يَشَاءُ وَاللَّهُ عَلَى كُلِّ شَيْءٍ قَدِيرٌ{284} آَمَنَ الرَّسُولُ بِمَا أُنْزِلَ إِلَيْهِ مِنْ رَبِّهِ وَالْمُؤْمِنُونَ كُلٌّ آَمَنَ بِاللَّهِ وَمَلَائِكَتِهِ وَكُتُبِهِ وَرُسُلِهِ لَا نُفَرِّقُ بَيْنَ أَحَدٍ مِنْ رُسُلِهِ وَقَالُوا سَمِعْنَا وَأَطَعْنَا غُفْرَانَكَ رَبَّنَا وَإِلَيْكَ الْمَصِيرُ{285}لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا لَهَا مَا كَسَبَتْ وَعَلَيْهَا مَا اكْتَسَبَتْ رَبَّنَا لَا تُؤَاخِذْنَا إِنْ نَسِينَا أَوْ أَخْطَأْنَا رَبَّنَا وَلَا تَحْمِلْ عَلَيْنَا إِصْرًا كَمَا حَمَلْتَهُ عَلَى الَّذِينَ مِنْ قَبْلِنَا رَبَّنَا وَلَا تُحَمِّلْنَا مَا لَا طَاقَةَ لَنَا بِهِ وَاعْفُ عَنَّا وَاغْفِرْ لَنَا وَارْحَمْنَا أَنْتَ مَوْلَانَا فَانْصُرْنَا عَلَى الْقَوْمِ الْكَافِرِين{286}

[سورة البقرة: الآيات 284- 286]

DEDICATION

This Project is dedicated to our parents for their emotional and financial support, to fathers who taught us that the best kind of knowledge to have it is that which is learned for its own sake. It is also dedicated to our mothers, who taught us that even the largest task can be accomplished if it is done one step at a time, to professor **Awadalla Taifour** who was a real support to us and without him this would not be accomplished and to everyone who support us and wished to us all the best.

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance from Professor **Awadalla Taifour Ali** and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank him.

ABSTRACT

The ball and plate system can usually be found in most university control labs, since it is relatively easy to build, model and control theoretically. This project will resolve in talking the ball-and-beam concept and develop a ball-and-plate balancing system. The system includes a ball, a plate, a touch screen, servo motors and a controller.

The ball and plate system is a ball balancing system which used a closed loop control system. The closed loop control which is used a digital Proportional Integral Derivative (PID) controller. The system controllers keep a free rolling ball in desired position on a flat plate. The mathematical model for this system is inherently nonlinear.

In this project, a theoretical analysis of the ball and plate system is conducted by employing physical laws. A performance is analyzed using experimental conditions. The main part of the system is Arduino, microcontrollers which receive the current ball position from the sensor (touch screen) and compare it with the desire position. PID algorithm was built to process the difference signal between current position and desired position and send it to servo motors to make the action. The systems feasibility and optimal operation were fully considered in the design phase.

MATLAB software program has been used to plot instant system response by interfacing Arduino with computer to determine system characteristics with different values of controller parameters in order to choose parameters values which obtained best performance for the system.

مستخلص

يتوفر نظام اتزان الكرة على اللوح عادة في معظم معامل التحكم الجامعية ، حيث انه نسبيا سهل التصميم و البناء و التحكم نظريا. هذا المشروع سوف يتحدث مفهوم اتزان الكرة على العارضة و تطويره لنظام اتزان الكرة على اللوح. يتضمن النظام :كرة ، لوح ، شاشة لمس، ومحركين ،و متحكم.نظام اتزان الكرة على اللوح هو نظام يستخدم دالة مسار التحكم المغلق .

المتحكم المستخدم هو التناسبي التكاملي التفاضلي. الفكرة الاساسية هي ان المتحكم يقوم بالمحافظة على الكرة في الموضع المحدد على سطح اللوح. النموذج الرياضي لهذا النظام غير خطي بطبيعته، في هذا المشروع يتم تطبيق القوانين الفيزيائية و يتم تحليل الأداء باستخدام الظروف التجريبية.

المكون الرئيسي لهذا النظام هو متحكم (اردوينو) الدقيق الذي يستقبل موضع الكرة من الحساس (شاشة اللمس) و يقارنها مع الموضع المطلوب .تم بناء خوارزمية المتحكم التناسبي التكاملي التفاضلي لمعالجة اشارة الفرق بين الموضع المقاس و الموضع المطلوب وارسالها للمحركين لتغيير موضع الكرة.

تم استخدام برنامج (ماتلاب) لرسم الاسجابة الحالية للنظام عند تغيير ثوابت المتحكم لاختيار قيم الثوابت التي تحقق افضل استجابة.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABELS

LIST OF SYMBOLS

K_p	Proportional gain
K_i	Integral gain
K_d	Derivative gain
T_i	Integral time
T_d	Derivative time
T	Motor torque
I	Armature Current
K_t	Torque constant
E	Electromotive force
θ	Angular velocity of the shaft
K_b	Electromotive force constant
K	Motor torque and back EMF constant
J	The moment of inertia of the rotor
B	The motor viscous friction constant
La	Electric inductance
Ra	Electric resistance
V	Voltage source
Kg	Gears ratio
Θ	Servo gear angle
$\propto$	Angle of the plate
r	Ball position
R	Radius of the ball
M	Mass of the ball
G	Gravitational acceleration
D	Lever arm of set
L	Length of the plate
D	Lever arm offset

LIST OF ABBREVIATIONS

AC	Alternating Current
CPU	Central Processing Unit
DC	Direct Current
EEPROM	Electrically Erasable Programmable Read Only Memory
EMF	Electromotive Force
EPROM	Erasable Programmable Read Only Memory
Hz	Hertz
IOT	Indium Tin Oxide
LED	Light Emitted Diode
MMF	Magnetic Motive Force
PD	Proportional Derivative controller
PI	Proportional Integral controller
PID	Proportional Integral Derivative
PROM	Programmable Read Only Memory
PWM	Pulse Width Modulation
RAM	Random Access Memory
ROM	Read Only Memory
RX	Receiver mode
TX	Transmitter mode

.

CHAPTER ONE

INTRODUCTION

1.1 General Concept

Ball on plate balancing system is one of the common and simplest control systems as it can be used to apply the control concepts studied in the system modeling course such as the settling time, overshooting, rise time and steady state error. It demonstrates the feedback closed loop system and how input affects the behavior of the output. Moreover, it shows the effect of nonlinearities and how it deviates the system from the theoretical model and how can we deal with this nonlinearities by introducing extra parameters like integration factor which forces the steady state error to be zero.

1.2 Problem Statement

The problem is to balance a ball on a flat plate by tilting the plate to move the ball. Given the constraints of cost, performance and functionality, a complete dynamic system should be made to balance the ball on the desired point on the plate.

1.3 Objectives

The main objectives of this study are to:

- Design proportional integral derivative (PID) controller to control ball position and improve system response.
- Implement the real time response of ball and plate system.
- Study the effects of PID controller parameters (proportional gain objectives of (K_p) – integral gain (K_i) – derivative gain (K_d)).

1.4 Methodology

- Study of all previous studies.
- Descriptive analysis of DC servomotor.
- Mathematical modeling of ball and plate system.
- Build Arduino microcontroller program to control ball position with PID algorithm as software included in the controller.
- Design of PID controller using manual tuning.
- Build MATLAB program to plot system response.

1.5 Layout

This study consists of five chapters: Chapter One gives an introduction to the principles of the work, in addition its reasons, motivation and objectives. Chapter Two discuss the theoretical background of control systems, ball and plate system, nonlinear systems, servomotors, touch screen, PID controller and microcontroller systems .Chapter Three presents the system mathematical modeling and control design of position control of ball and plate system. Chapter Four deal with the practical model of the system and shows the experimental results. Finally, Chapter five provides the conclusions and recommendations.

CHAPTER TWO
THEORETICAL BACKGROUND AND LITERATURE REVIEW

2.1 Linear Systems

A system is called linear if the principle of superposition applies. The principle of superposition states that the response produced by the simultaneous application of two different forcing functions is the sum of the two individual responses. Hence, for the linear system, the response to several inputs can be calculated by treating one input at a time and adding the results. It is this principle that allows one to build up complicated solutions to the linear differential equation from simple solutions. In an experimental investigation of a dynamic system if cause and effect are proportional, thus implying that the principle of superposition holds, then the system can be considered linear.

2.2 Nonlinear Systems

A system is nonlinear if the principle of superposition does not apply. Thus, for a nonlinear system the response to two inputs cannot be calculated by treating one input at a time and adding the results. Although many physical relationships are often represented by linear equations, in most cases actual relationships are not quite linear. In fact, a careful study of physical systems reveals that even so-called “linear systems” are really linear only in limited operating ranges. In practice, many electromechanical systems, hydraulic systems, pneumatic systems, and so on, involve nonlinear relationships among the variables. For example, the output of a component may saturate for large input signals. There may be a dead space that affects small signals. (The dead space of a component is a small range of input variations to which the component is insensitive.) Square-law nonlinearity may occur in some components. For instance, dampers used in physical systems may be linear for low-velocity operations but may become nonlinear at high velocities, and the damping force may become proportional to the square of the operating velocity [1].

2.2.1 Linearization of nonlinear systems

In control engineering a normal operation of the system may be around an equilibrium point, and the signals may be considered small signals around the equilibrium. (It should be pointed out that there are many exceptions to such a case.) However, if the system operates around an equilibrium point and if the signals involved are small signals, then it is possible to approximate the nonlinear system by a linear system. Such a linear system is equivalent to the nonlinear system considered within a limited operating range. Such a linearized model (linear, time-invariant model) is very important in control engineering.

The linearization procedure to be presented in the following is based on the expansion of nonlinear function into a Taylor series about the operating point and the retention of only the linear term. Because we neglect higher-order terms of the Taylor series expansion, these neglected terms must be small enough; that is, the variables deviate only slightly from the operating condition. (Otherwise, the result will be inaccurate.) [1].

2.3 Ball & Plate System

A ball and plate system is one of the challenging control bench marking systems integrated into many practices and techniques. This project will resolve in taking the ball-and-plate concept and develop a ball-and-plate balancing system. The system will utilize sensors, actuators, and control law to manipulate the servos in a feedback stabilization using three-degree-offreedom compensation. This is essentially implementing two ball-and-beam experiments in parallel to constructing a ball-and-plate prototype.

The concept of the ball-and-plate system is a simple system that is an unstable open-loop .Without an active feedback control system, the horizontal plate will tilt to either side, and the ball will roll off in the plate. In order to stabilize the ball, a control system is applied to measure the position of the ball and adjusts the plate accordingly. The objective of this project is to keep a ball on a platform within a predetermined boundary.

The sensor's function is to monitor a ball's position. When the ball moves outside the boundary the sensor will send signals to the controller to determine the coordinates needed to stabilize the ball to its designated location. The controller and the servos will simultaneously control the plate's pitch and roll to center the ball within the boundary. The platform consists of a touch panel and a base to house the components.

The ball-and-plate system is designed with the purpose to have key functionalities in mind. The first is automatic stabilization of the ball's location. The system should also provide feedback on where the ball is located on the sensor, and be able to have variable predestination of the desired ball location. So the ball's desired position can be changed at any time from user input [2].

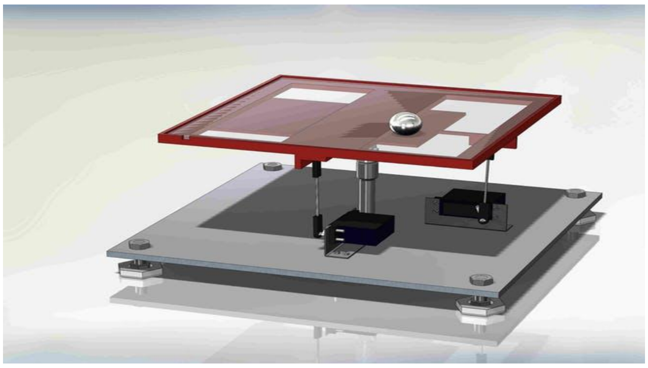

Figure 2.1: Ball & Plate model

2.4 Proportional, Integral & Derivative Controller

A proportional integral derivative (PID) controller is a control loop feedback mechanism (controller) widely used in industrial control systems. A PID controller calculates an error value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process through use of a manipulated variable.

The acronym PID stands for **P**roportional-**I**ntegral-**D**ifferential control. Each of these, the **P**, the **I** and the **D** are terms in a control algorithm, and each has a special purpose. Sometimes certain of the terms are left out because they are not needed in the control design. This is possible to have a PI, PD or just a P control .It is very rare to have an ID control.

The standard PID control configuration as shown is Figure 2.2. It is also sometimes called the "PID parameter form".

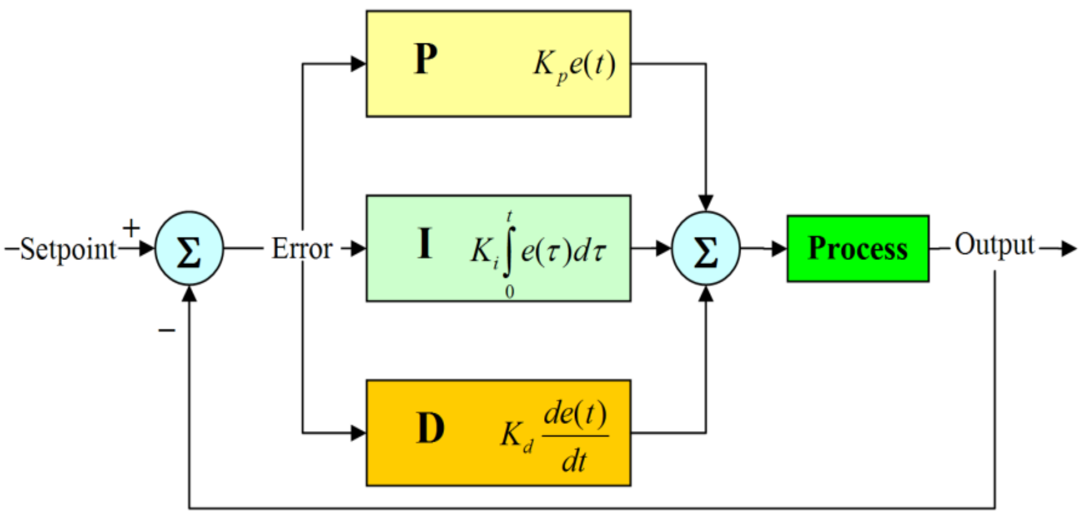

Figure 2.2: PID Controlled System

In this configuration, the control signal u (t) is the sum of three terms. Each of these terms is a function of the tracking error e (t).The term K_p indicates that this term is proportional to the error. The term K_i/s an integral term,and the term $K_d s$ a derivative term. Each of the terms works “independently”1 of the other.

2.4.1 Effects of proportional, integral and derivative action

Proportional control is illustrated in Figure 2.3. The controller is given by function $P(s) = 1/(s+1)^3$ with $T_i = \infty$ and $T_d = 0$. The figure shows that there is always steady state error in proportional control. The error will decrease within creasing gain, but the tendency towards oscillation will also increase.

Figure 2.4 illustrates the effects of adding integral to system with transfer function of $P(s) = 1/(s+1)^3$. It follows that the strength of integral action increases with decreasing integral time T_i. The figure shows that the steady state error disappears when integral action is used. Compare with the discussion of the “magic of integral action”. The tendency for oscillation also increases with decreasing T_i.

Figure 2.5 illustrates the effects of adding derivative action to system with transfer function of $P(s) = 1/(s+1)^3$ and controller gain of K=3, and integral time of T_i=2.The parameters K and Ti are chosen so that the closed loop system is oscillatory. Damping increases with increasing derivative time, but decreases again when derivative time becomes too large.

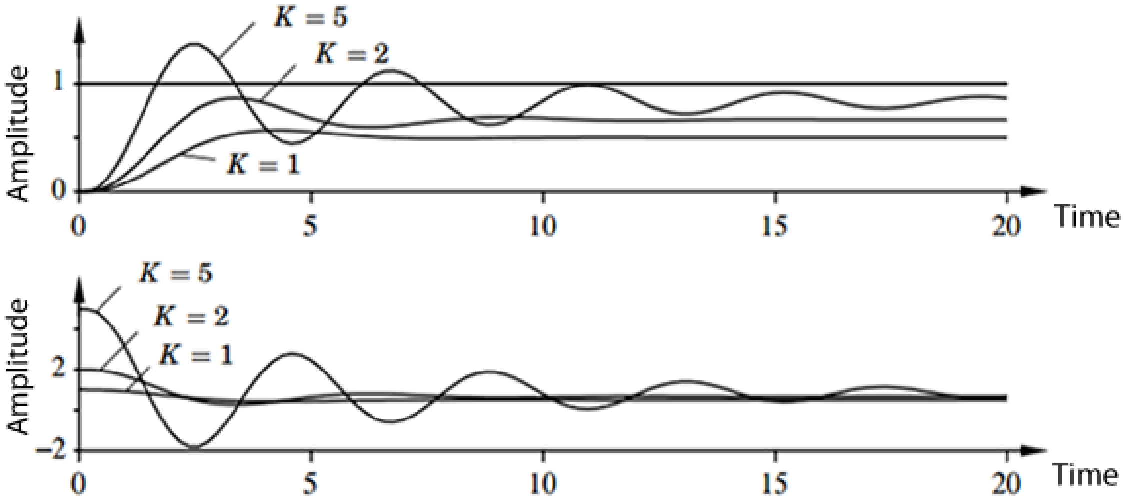

Figure 2.3: Response of the system with proportional control

Recall that derivative action can be interpreted as providing prediction by linear extrapolation over the time T_d. Using this interpretation it is easy to understand that derivative action does not help if the prediction time T_d is too large. In Figure (2.5) the period of oscillation is about 6 s for the system without derivative action. Derivative actions cease to be effective when T_d is larger than a 1s (one sixth of the period). Also notice that the period of oscillation increases when derivative time is increased.

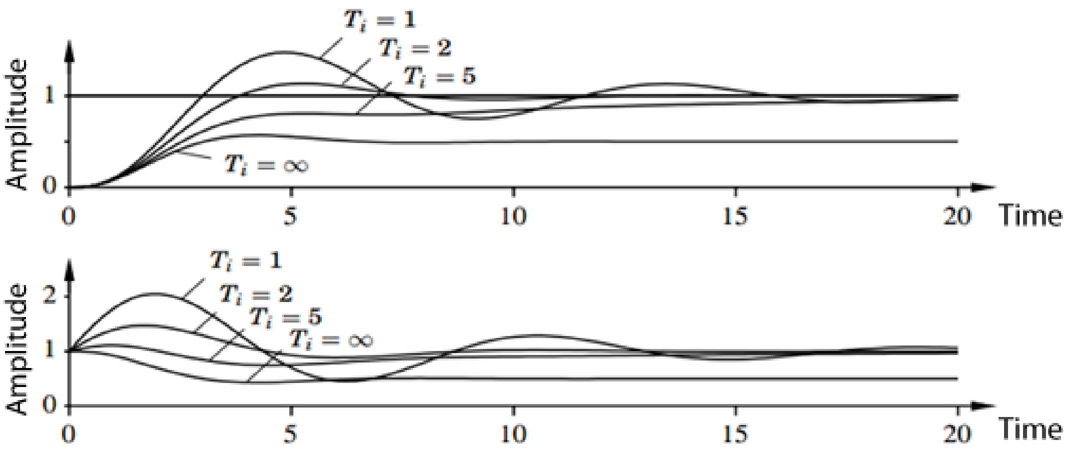

Figure 2.4: Response of the system with proportional and integral control

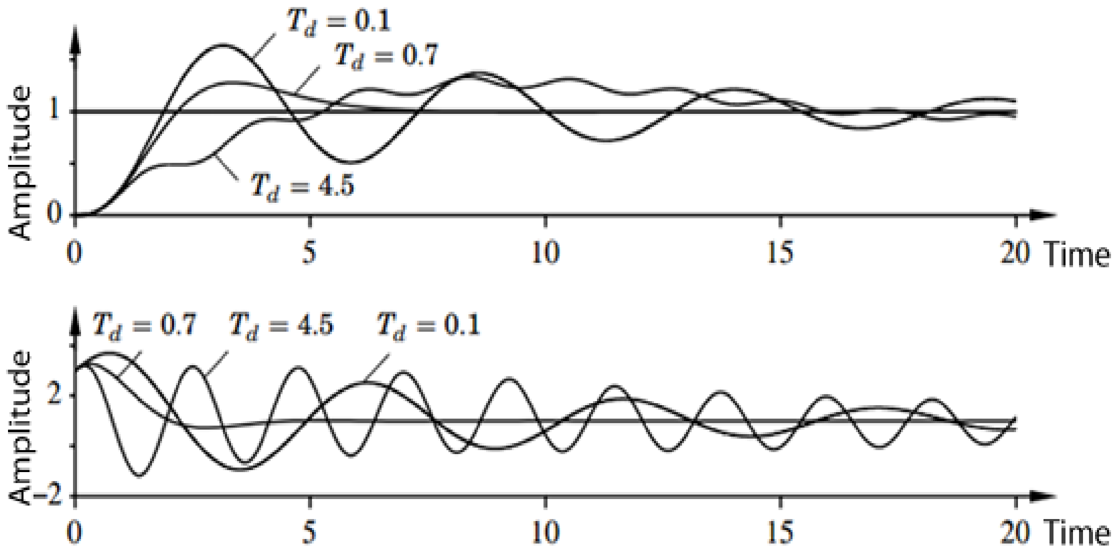

Figure 2.5: Response of the system with proportional, integral and derivative control

Table 2.1: Effects of increasing a parameter independently

Parameter	**Rise time**	**Overshoot**	**Settling time**	**Steady-state**
K_P	Decrease	Increase	Small change	Decrease
K_I	Decrease	Increase	Increase	Eliminate
K_D	No change	Decrease	Decrease	Small effect

2.4.2 Proportional controller

In the **P** controller algorithm, the controller output is proportional to the error signal, which is the difference between the set point and the process variable. In **P** controller the actuating signal for the control action in a control system is proportional to the error signal. The error signal being the difference between the reference input signal and feedback signal obtained from the output.

The proportional response can be adjusted by multiplying the error by a constant K_p, called the proportional gain constant. The proportional term is given by:

$P = K_p (e(t))$ (2.1)

2.4.3 Proportional integral controller

At present, the PI controller is most widely adopted in industrial application due to its simple structure, easy to design and low cost. Despite these advantages, the PI controller fails when the controlled object is highly nonlinear and uncertain. PI controller will eliminate forced oscillations and steady state error resulting in operation of on-off controller and P controller respectively. However, introducing integral mode has a negative effect on speed of the response and overall stability of the system. Thus, PI controller will not increase the speed of response. It can be expected since PI controller does not have means to predict what will happen with the error in near future. This problem can be solved by introducing derivative mode which has ability to predict what will happen with the error in near future and thus to decrease a reaction time of the controller. PI controllers are very often used in industry, especially when speed of the response is not an issue.

$PI = K_p(e(t)) + K_i \int e(t)\, dt$ (2.2)

2.4.4 Proportional derivative controller

The purpose of the derivative action is to improve the close-loop stability.
Because of the process dynamics, it will take some time before a change in the control variable is noticeable in the progress output. Thus, the control system will be late in correction for an error. The action of a controller with proportional and derivative may be interpreted as if the control is made proportional to the predicted process output, where the prediction is made by extrapolating the error by the tangent to the error curve.

$PD = K_p (e(t)) + K_d (d(e(t))/dt)$ (2.3)

Many industrial controllers employ a proportional, integral plus differential PID regulator arrangement that can be tailored to optimize a particular control system. PID controller is most commonly used algorithm for controller design and it is most widely used controller in industry. The controllers used in industry are either PID controller or its improved version. The basic types of PID controller are parallel controller, serial controller, and mixed controller. The PID controller algorithm utilized for is design velocity algorithm, it is also called incremental algorithm. In the industry, PID controllers are the most common control methodology to use in real applications.

PID controller has all the necessary dynamics: fast reaction on change of the controller input (D mode), increase in control signal to lead error towards zero (I mode) and suitable action inside control error area to eliminate oscillations (P mode). Derivative mode improves stability of the system and enables increase in gain K and decrease in integral time constant Ti, which increases speed of the controller response .PID controllers are the most often used controllers in the process industry. The majority of control systems in the world are operated PID controllers. It has been reported that 98% of the control loops in the pulp and paper industries are controlled by single-input single output PI controllers and that in process control applications, more than 95% of the controllers are of the PID type controller. PID controller combines the advantage of proportional, derivative and integral control action.

$$PID\ (t) = K_p\ (e\ (t)) + K_i \int e(t)dt + K_d\ (e(t)/dt) \qquad (2.4)$$

2.4.5 Tuning of proportional integral derivative controller

All general methods for control design can be applied to PID control. A number of special methods that are tailor made for PID control have also been developed, these methods are often called tuning methods. Irrespective of the method used it is essential to always consider the key elements of control, load disturbances, sensor noise, a process uncertainty and reference signals.

The most well-known tuning methods are those developed by Ziegler - Nichols. They have had a major influence on the practice of PID control for more than half a century. The methods are based on characterization of process dynamics by a few parameters and simple equations for the controller parameters. It is surprising that the methods are so widely referenced because they give moderately good tuning only in restricted situations. Plausible explanations may be the simplicity of the methods and the fact that they can be used for simple student exercises in basic control courses.

There are many methods of PID tuning such as:

- Manual tuning.
- Ziegler-Nichols.
- Tyreusluyben.
- Cohen-coon.
- Software tools.

2.5 Servomotors

Also called control motors and have high torque capabilities. Unlike large industrial motors, servomotors are not used for continuous energy conversion but only for precise speed and precise position control at high torques. Of course, their basic principle of operation is the same as that of other electromagnetic motors. However, their construction, design and mode of operation are different. Their power ratings vary from a fraction of a Watt up to a few 100 Watts. Due to their low inertia; servomotors have high speed of response. That is why servomotors are smaller in diameter but longer in length. Servomotors generally operate at very low speeds or sometimes zero speed. Servomotors find wide applications in radar, tracking and guidance systems, process controllers, computers and machine tools. Both DC and AC (2-phase and 3-phases) servomotors are used at present.

Servomotors differ in application capabilities from large industrial motors in the following respects:

- They produce high torque at all speeds including zero speed.
- They are capable of holding static position.
- They do not overheat at standstill or lower speeds.
- Due to low-inertia, they are able to reverse directions quickly.
- They are able to accelerate and decelerate quickly.
- They are able to return to a given position time after time without any drift.

2.5.1 DC servomotors

These motors are either separately excited DC motors or permanent magnet DC motors. The torque/speed characteristics were shown in Figure 2.6 and the schematic diagram of a separately-excited DC motor along with its armature and field MMFs shown in Figure 2.7. The speed of DC servomotors is normally controlled by varying the armature voltage. Their armature is deliberately designed to have large resistance so that torque speed characteristics are linear and have a large negative slope. The negative slope serves the purpose of providing the viscous damping for the servo drive system as shown in Figure 2.8 the armature MMF and excitation field MMF are in quadrature.

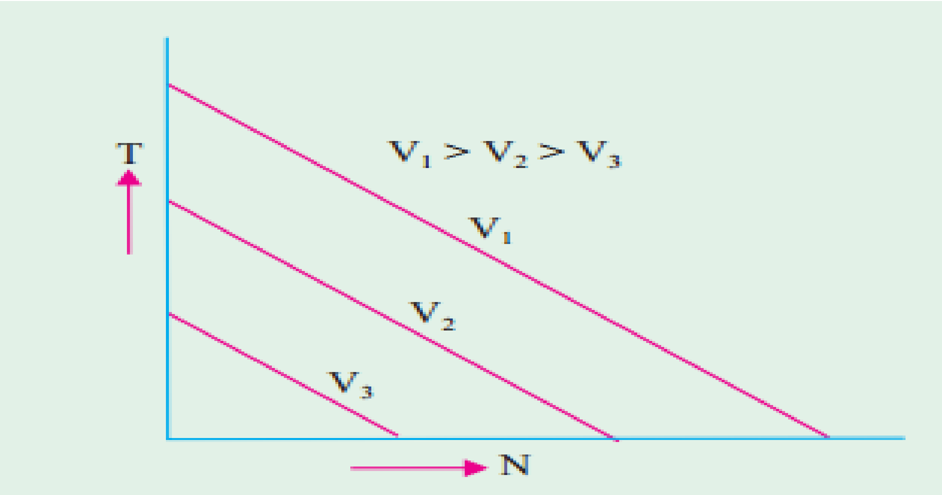

Figure 2.6: Torque/speed characteristics

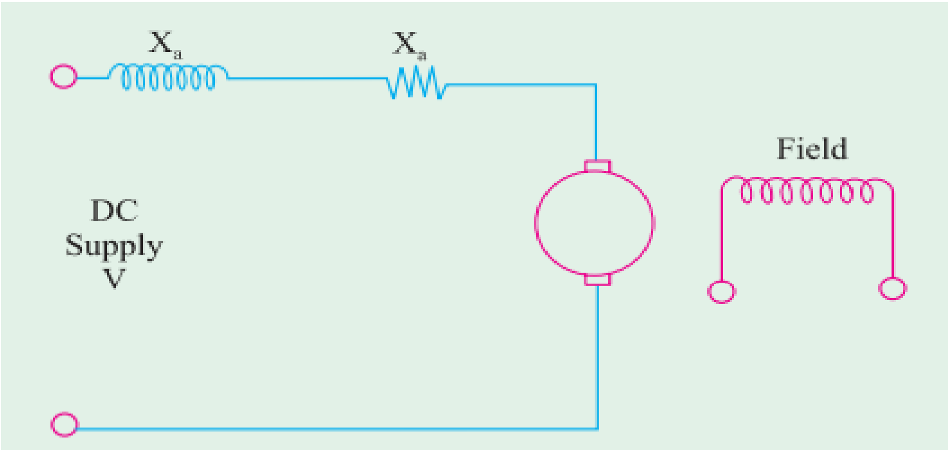

Figure 2.7: The schematic diagram of a separately-excited DC motor

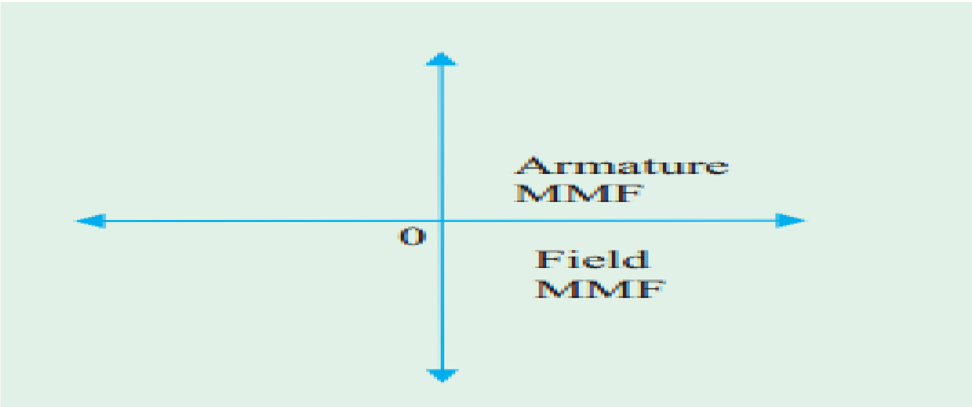

Figure 2.8: The armature MMF and excitation field MMF

2.5.2 AC servomotors

Presently, most of the ac servomotors are of the two phase squirrel cage induction type and are used for low power applications. However, recently three-phase induction motors have been modified for high power servo systems which had so far been using high power DC servomotors. Two-phase AC Servomotors normally run on a frequency of 60 Hz or 400 Hz (for airborne systems). The stator has two distributed windings which are displaced from each other by 90º (electrical). The main winding (also called the reference or fixed phase) is supplied from a constant voltage source, $V_{in}\angle 0^{o}$. The other winding (also called the control phase) is supplied with a variable voltage of the same frequency as the reference phase but is phase-displaced by 90º (electrical). The control phase voltage is controlled by an electronic controller. The schematic of two phase squirrel cage AC servomotor is shown in Figure 2.9. The speed and torque of the rotor are controlled by the phase difference between the main and control windings. Reversing the phase difference from leading to lagging (or vice-versa) reverses the motor direction. Since the rotor bars have high resistance, the torque speed characteristics for various armature voltages are almost linear over a wide speed range particularly near the zero speed. The motor operation can be controlled by varying the voltage of the main phase while keeping that of the reference phase constant. A great deal of research has been to modify a three phase squirrel cage induction motor for use in high power servo systems. Normally, such a motor is a highly non-linear coupled-circuit device. Recently, this machine has been operated successfully as a linear decoupled machine (like a DC machine) by using a control method called vector control or field oriented control. In this method, the currents fed to the machine are controlled in such a way that its torque and flux become decoupled as in a DC machine. This results in a high speed and a high torque response [3].

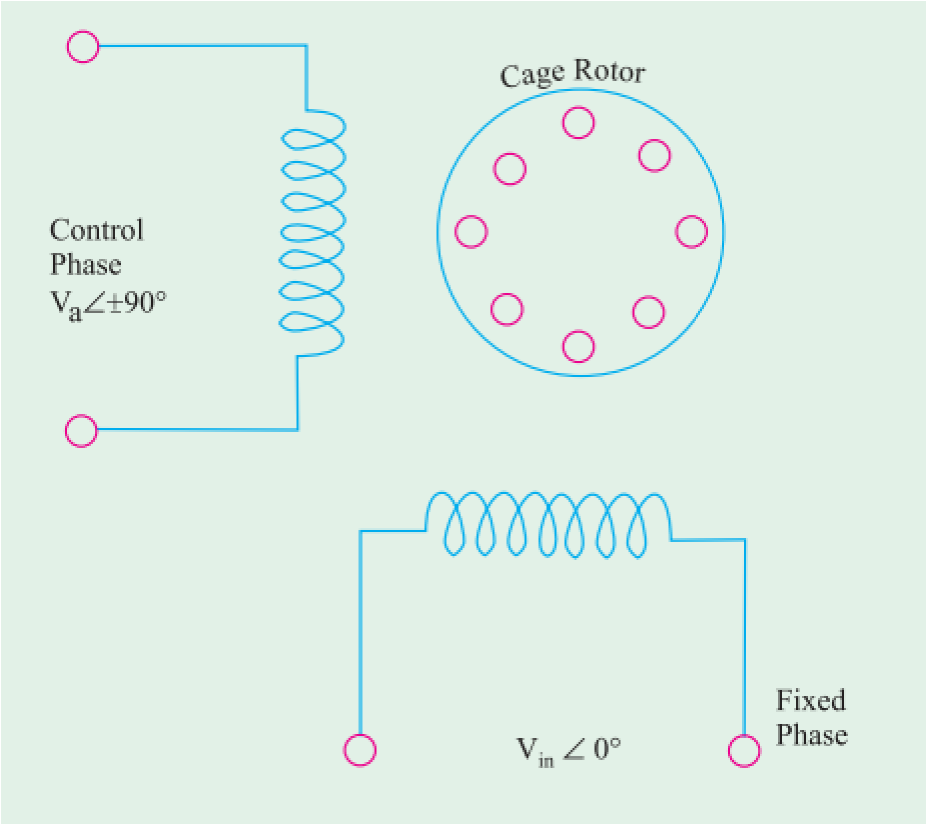

Figure 2.9: The schematic of two phase squirrel cage AC servomotor

2.6 Microcontrollers

Microcontrollers and microprocessors are integrated circuits, but they differ fundamentally from other ICs.

They are a class in themselves, that the designers have not made them to do a particular job. As such when you buy them from the market, you can not specify what function it will do. In order to get some useful function, these ICs have to be configured. Thus a microprocessor or microcontroller can be configured to check the status of a button, and then turn a motor ON or OFF. While the same ICs can be configured later, to read the status of an infra-red sensor, decode the signal and turn another device ON or OFF. If these two types of circuitries were to be made using conventional digital ICs, it would have required a large number of components. Moreover any change in the specification, like change of Infra-Red codes would result in total change in design! Using a configurable IC is a great idea. Not only the same IC can be configured to do different tasks, but a change in specifications can easily be implemented by just changing the device configuration. This greatly facilitated the engineers and hobbyists to rapidly develop new electronic devices, and continuously improve previous ones. Not only the hardware requirements decreased, but also design time, and time to market were decreased.

2.6.1 Arduino

Arduinos an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing .Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IOT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of user's worldwide [4].

Figure 2.10: Arduino microcontroller board

2.6.2 The Arduino board

It is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating inter-active objects or environments in simple terms, the Arduino is a tiny computer system that can be programmed with your instructions to interact with various forms of input and output. The current Arduino board model, the Uno, is quite small in size compared to the average human hand, as shown in Figure 2.11.

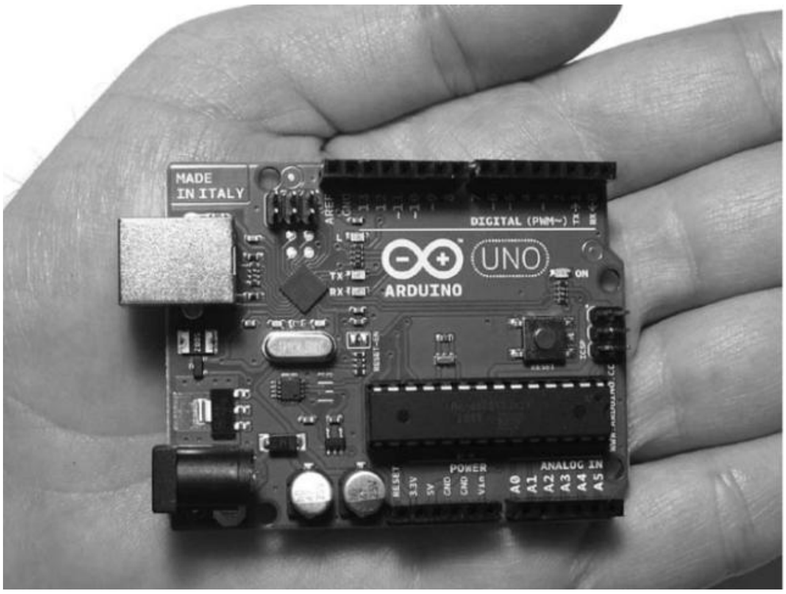

Figure 2.11: An Arduino Uno is quite small.

Although it might not look like much to the new observer, the Arduino system allows creating devices that can interact with the world. By using an almost unlimited range of input and output devices, sensors, indicators, displays, motors, and more, the exact interactions required to create a functional device can be programmed. For example, artists have created installations with patterns of blinking lights that respond to the movements of passers-by, high school students have built autonomous robots that can detect an open flame and extinguish it, and geographers have designed systems that monitor temperature and humidity and transmit this data back to their offices via text message. In fact, there are infinite numbers of examples with a quick search on the Internet. By taking a quick tour of the Uno Starting at the left side of the board there are two connectors, as shown in Figure 2.12.

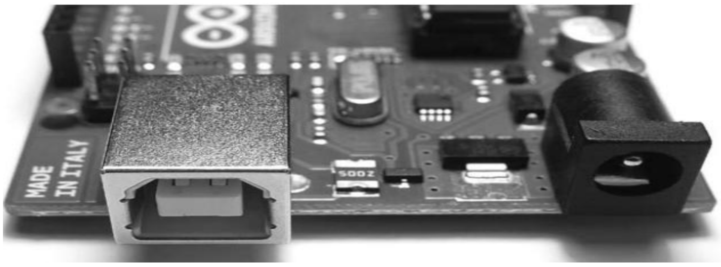

Figure 2.12: The USB and power connectors

On the far left is the Universal Serial Bus (USB) connector. This connects the board to your computer for three reasons; to supply power to the board, to upload the instructions to the Arduino, and to send data to and receive it from a computer. On the right is the power connector, this connector can power the Arduino with a standard mains power adapter. At the lower middle is the heart of the board: the microcontroller, as Shown in Figure 2.13.

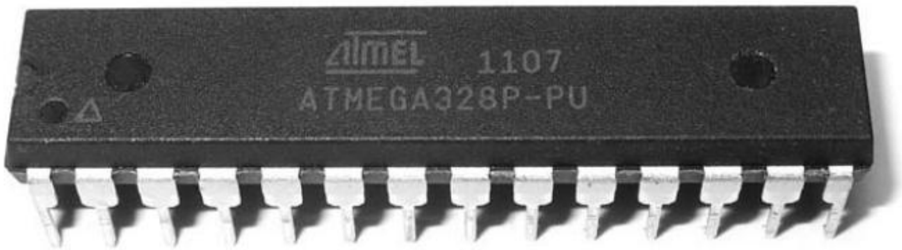

Figure 2.13: The microcontroller

The microcontrollers represent the “brains” of the Arduino. It is a tiny computer that contains a processor to execute instructions, includes various types of memory to hold data and instructions from our sketches, and provides various avenues of sending and receiving data. Just below the microcontroller are two rows of small sockets, as shown in Figure 2.14.

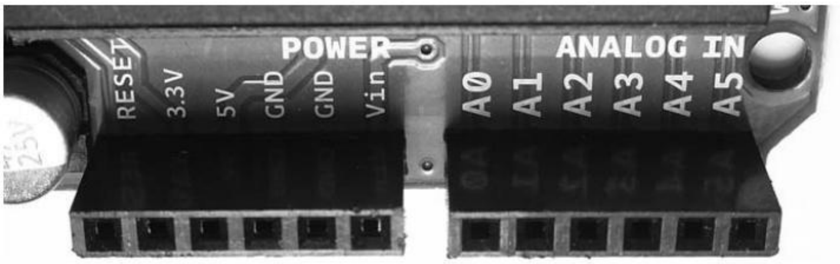

Figure 2.14: The power and analog sockets

The first row offers power connections and the ability to use an external RESET button. The second row offers six analog inputs that are used to measure electrical signals that vary in voltage. Furthermore, pins A4 and A5 can also be used for sending data to and receiving it from other devices. Along the top of the board are two more rows of sockets, as shown in Figure 2.15.

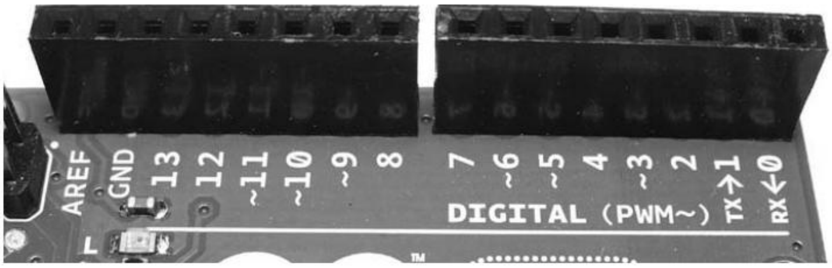

Figure 2.15: The digital input/output pins

Sockets (or pins) numbered 0 to 13 are digital input/output (I/O) pins. They can either detect whether or not an electrical signal is present or generate a signal on command. Pins 0 and 1 are also known as the serial port, which is used to send and receive data to other devices, such as a computer via the USB connector circuitry. The pins labeled with a tilde (~) can also generate a varying electrical signal, which can be useful for such things as creating lighting effects or controlling electric motors. Next are some very useful devices called light-emitting diodes (LEDs); these very tiny devices light up when a current passes through them. The Arduino board has four LEDs: one on the far right labeled ON, which indicates when the board has power, and three in another group, as shown in Figure 2.16. The LEDs labeled TX and RX light up when data is being transmitted or received between the Arduino and attached devices via the serial port and USB.

The L-LED connected to the digital I/O pin number 13. The little black square part to the left of the LEDs is a tiny microcontroller that controls the USB interface that allows Arduino to send data to and receive it from a computer. And, finally, the RESET button is shown in Figure 2.17.

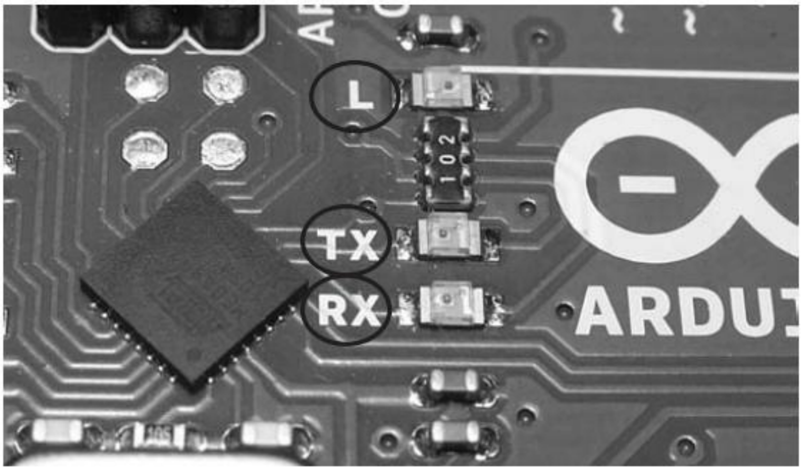

Figure 2.16: The onboard LEDs

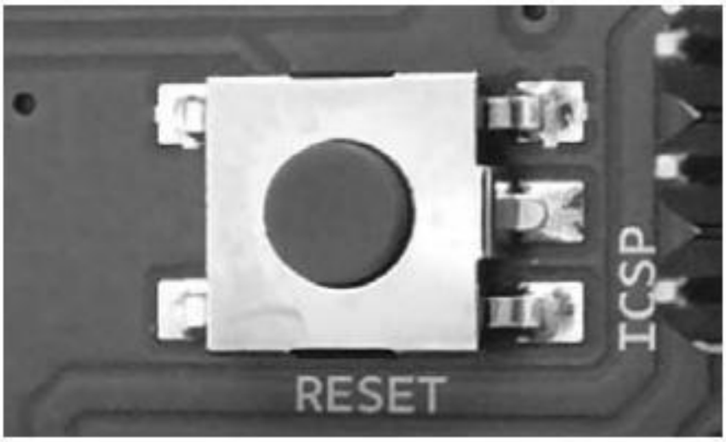

Figure 2.17: The RESET button

2.7 Open and Closed Loop Control Systems

Open loop control systems are in general relatively simple in design and inexpensive. Closed loop control system measures the value of the parameters being controlled at the output of the system and compares this to the desired value.

2.7.1 Open-Loop control systems

Those systems in which the output has no effect on the control action are called open-loop control systems. In other words, in an open loop control system the output is neither measured nor fed back for comparison with the input. One practical example is a washing machine. Soaking, washing, and rinsing in the washer operate on a time basis. The machine does not measure the output signal, that is, the cleanliness of the clothes.

For example of open-loop systems are mechanical systems consisting of a mass, spring, and damper with a constant force positioning the mass the greater the force, the greater the displacement. Again, the system position will change with a disturbance, such as an additional force, and the system will not detect or correct for the disturbance.

2.7.2 Closed-Loop (feedback) control systems

Feedback control systems are often referred to as closed-loop control systems. In practice, the terms feedback control and closed-loop control are used interchangeably. In a closed-loop control system the actuating error signal, which is the difference between the input signal and the feedback signal (which may be the output signal itself or a function of the output signal and its derivatives and/or integrals), is fed to the controller so as to reduce the error and bring the output of the system to a desired value. The term closed-loop control always implies the use of feedback control action in order to reduce system error.

2.7.3 Closed-loop versus open-loop control systems

An advantage of the closed loop control system is the fact that the use of feedback makes the system response relatively insensitive to external disturbances and internal variations in system parameters. It is thus possible to use relatively inaccurate and inexpensive components to obtain the accurate control of a given plant, whereas doing so is impossible in the open-loop case. From the point of view of stability, the open-loop control system is easier to build because system stability is not a major problem. On the other hand, stability is a major problem in the closed-loop control system, which may tend to overcorrect errors and thereby can cause oscillations of constant or changing amplitude. It should be emphasized that for systems in which the inputs are known ahead of time and in which there are no disturbances it is advisable to use open-loop control. Closed-loop control systems have advantages only when unpredictable disturbances and/or unpredictable variations in system components are present. Note that the output power rating partially determines the cost, weight, and size of a control system. The number of components used in a closed-loop control system is more than that for a corresponding open-loop control system. Thus, the closed-loop control system is generally higher in cost and power. To decrease the required power of a system, open loop control may be used where applicable. A proper combination of open-loop and closed-loop controls is usually less expensive and will give satisfactory overall system performance.

2.7.4 Advantages of control systems

Control systems giving the ability to move large equipment with precision that would otherwise be impossible. For example pointing huge antennas toward the farthest reaches of the universe to pick up faint radio signals; controlling these antennas by hand would be impossible. Because of control systems, elevators carry us quickly to our destination, automatically stopping at the right floor. Humans alone could not provide the power required for the load and the speed; motors provide the power, and control systems regulate the position and speed.

Control systems build for primary reasons; Power amplification, remote control, convenience of input form and compensation for disturbances [5].

2.8 Touch Screen

A sensor is a device that converts a physical phenomenon into an electrical signal. As such, sensors represent part of the interface between the physical world and the world of electrical devices, such as computers.

2.8.1 Resistive touch screen

A resistive touch screen panel is coated with a thin metallic electrically conductive and resistive layer that causes a change in the electrical current which is registered as a touch event and sent to the controller for processing. Resistive 4- and 5-wire touch sensors are the most popular and most common touch-screen technologies with about 75% market share, mainly due to their low costs and simple interface electronics. The high volume of these screens requires a low-cost reliable interface, often with a low-power element. This can be provided through a range of analog features combined with low-power modes for portable, battery-powered applications.

Usually a resistive touch-screen consists of at least three layers Figure 2.16. A flexible membrane made from poly ethylene terephthalate (PET) film is suspended over a rigid substrate made from glass or acrylic, and both surfaces are coated with a transparent conductive film, usually Indium tin oxide (ITO). The conductive ITO layers are kept apart by an insulting spacer along the edges, and by spacer dots on the inner surface of the two ITO layers. In this way, there will be no electrical connection unless pressure is applied to the top sheet (PET film).

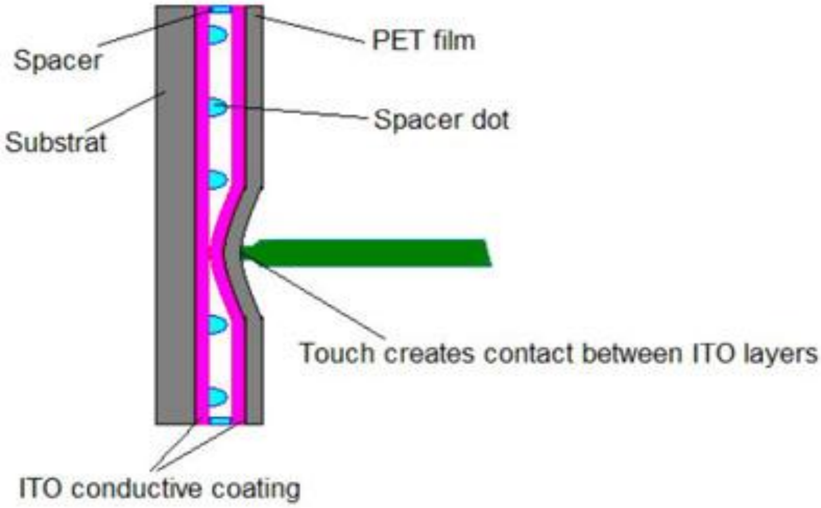

Figure 2.18: The resistive touch-screen sensor structure.

4-wire touch-screens use a single pair of electrodes called bus bars on each ITO layer. The four wires reference X+ (left), X- (right), Y+ (top) and Y- (bottom) locations, which is key to interfacing the sensors.

In 5-wire touch-screens the four wires connect to the electrodes, these are referred to as UL (Upper Left), UR (Upper Right), LL (Lower Left), and LR (Lower Right). The fifth wire is used for sensing the electrode voltage is referred to as the "sense" wire, and is embedded in the top sheet.

5-wire touch-screens is less critical than for 4-wire touch-screens. The pressure points are measured by applying a voltage to the electrode pair in the resistive surface so that a uniform voltage gradient appears across the surface. A second ITO layer is necessary to do a high-resistance voltage measurement.

CHAPTER THREE

SYSTEM MODELING AND DESIGN

3.1 Ball and Plate System Model

The system consist of two separated systems, the first one is the DC servo motor which is an electromechanical system that receives electrical signal from controller and gives output as a rotational displacement (angle).The second is ball and plate model which is a mechanical system that receives rotational displacement (angle) from motor and converts it into a linear displacement.

3.1.1 DC Servo motor model

A common actuator in control systems is the DC motor. It directly provides rotary motion and coupled with wheels or drums and cables can provide translational motion. The electric equivalent circuit of the armature and the free-body diagram of the rotor are shown in Figure 3.1.

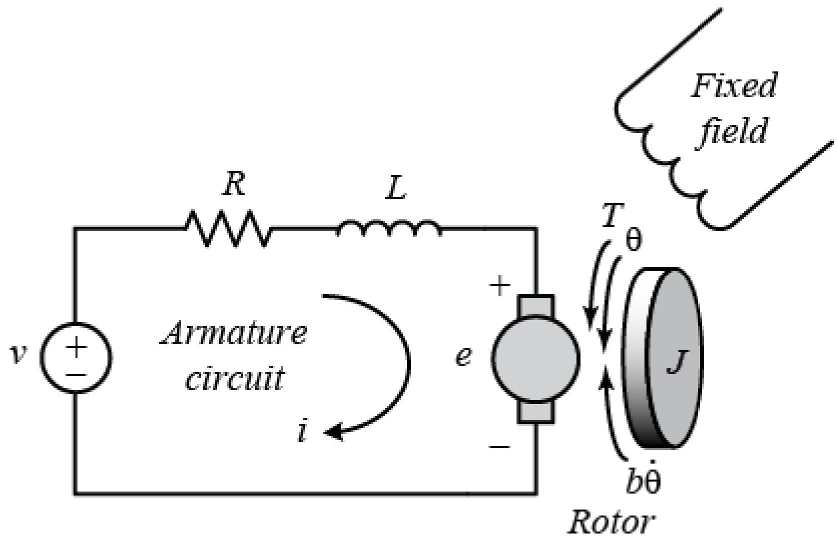

Figure 3.1: Equivalent circuit of DC motor

In general, the torque generated by a DC motor is proportional to the armature current and the strength of the magnetic field. Assume that the magnetic field is constant and, therefore, that the motor torque (T) is proportional to only the armature current (i) by a constant factor (K_t) called torque constant as shown in the Equation (3.1). This is referred to as an armature-controlled motor.

$$T=K_t\, I \tag{3.1}$$

The back electromotive force emf (e) is proportional to the angular velocity of the shaft (θ) by a constant factor (K_b) called electromotive force constant as shown in the Equation (3.2).

$$e = K_b\theta \tag{3.2}$$

In system international units, the motor torque and back emf constants are equal, that is, $K_t=K_b$ therefore, (K) will be used to represent both the motor torque constant and the back emf constant.

By using Figure (3.1) to derive the following governing equations based on Newton's 2nd law in Equation (3.3) and Kirchhoff's voltage law in Equation (3.4).

$$J\theta + b\theta=Ki \tag{3.3}$$

$$La\frac{di}{dt}+Ra*i=V-K_b\theta \tag{3.4}$$

Where (J) is the moment of inertia of the rotor, (b) is the motor viscous friction constant, (La) is the electric inductance, (Ra) is the electric resistance, and (V) is the voltage source.

Applying the Laplace transform, the modeling equations can be expressed in terms of the Laplace variables as shown in Equation (3.5) and Equation (3.6).

$$s(Js+b)\theta(s) = KI(s) \tag{3.5}$$

$$(Las + Ra)I(s) = V(s) - Ks\theta(s) \tag{3.6}$$

Following open-loop transfer function had arrived by eliminating I(s) in Equation (3.7), where the rotational speed is considered the output and the armature voltage is considered the input.

$$G(s) = \frac{\theta(s)}{V(s)} = \frac{K}{(Js+b)(Las+Ra)+K^2}\left[\frac{rad/sec}{V}\right] \tag{3.7}$$

However, during this model continue looking at the position as the output. The position has been obtained by integrating the speed; therefore, it needed to divide the transfer function in Equation (3.8).

$$\frac{\theta(s)}{V(s)} = \frac{K}{s((Js+b)(Las+Ra)+K^2)}\left[\frac{rad}{V}\right] \tag{3.8}$$

The position θ(s) in Equation (3.8) is the rotational displacement which produced on the motor's shaft but there are gears between shaft and load with ratio known as gears ratio (Kg), so the transfer function becomes as shown in Equation (3.9) [6].

$$\frac{\theta(s)}{V(s)} = \frac{K*Kg}{s((Js+b)(Las+Ra)+K^2)} \tag{3.9}$$

3.1.2 Ball and plate system model

The objective of this project is to keep a ball on a platform within a predetermined boundary. The sensor's function is to monitor a ball's position. When the ball moves outside the boundary, the sensor will send signals to the controller to determine the coordinates needed to stabilize the ball to its designated location. The controller and the servos will simultaneously control the plate's pitch and roll to center the ball within the boundary.

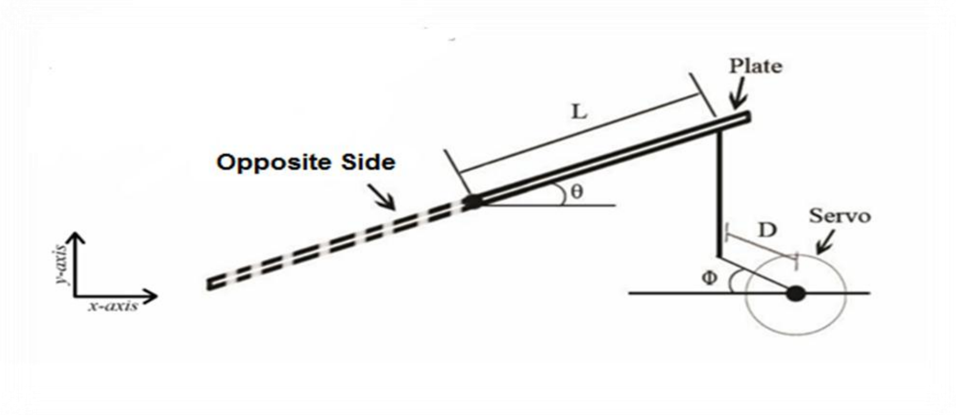

Figure 3.2: Ball and plate model

To help simplify the equations for ease of modeling, friction will be neglected and the assumption that no slippage will occur between the ball and the plate. Using these constraints the force equation centered on the motion of the ball is:

$$F_{Gravity} = mg\sin(\theta) \quad (3.10)$$

$$F_{Normal} = mx\,\theta^{2} \quad (3.11)$$

$$F_{Ball} = \frac{J}{R^2} + mx \quad (3.12)$$

$$F_{Ball} + F_{Gravity} = F_{Normal} \quad (3.13)$$

$\frac{J}{R^2}$ + mx+ mgsin(θ) = mx θ ^{2} (3.14)

$\frac{J}{R^2}$ + mx+ mgsin(θ) -mx θ ^{2} = 0 (3.15)

Now if the assumption of a small angle changes are considered

θ≈0 (3.16)

sin(θ) ≈ 0 (3.17)

cos(θ) ≈ 1 (3.18)

$\frac{J}{R^2}$ + mx+ mgθ = 0 (3.19)

Also, it may be desirable to have motor angel as a variable instead of plate angle. A relation between the two are given by

θ = $\frac{r}{L}$ α (3.20)

Substituting this relationship into the previous equation and solving for

-mgθ = $\frac{J}{R^2}$ + mx (3.21)

-mg$\frac{r}{L}$α = $\frac{J}{R^2}$ + mx (3.22)

X = $\frac{-mg\frac{r}{L}\alpha}{\frac{J}{R^2}+m}$ (3.23)

This is a mathematical representation of system in one direction. Since assuming small angle motions, the two directions, x and y, are decoupled. This means the model above will apply to both directions independently. Therefore the equation is the x and y directions are:

$$X = \frac{-mg\frac{r}{L}\alpha_y}{\frac{J}{R^2}+m} \qquad (3.24)$$

$$Y = \frac{-mg\frac{r}{L}\alpha_x}{\frac{J}{R^2}+m} \qquad (3.25)$$

Where α_y is the motor angle around the x axis and α_x is the motor angle around the y axis. For the system developed, the motor angle and the plate angle are in direct relation because of a 1:1 ratio used in the gearing. Note that the inertia of a ball is given by tow equations for either a solid ball or a hollow ball.

$$J_{Solid} = \frac{2}{5} mR^2 \qquad (3.26)$$

$$J_{Hollow} = \frac{2}{3} mR^2 \qquad (3.27)$$

Substituting this relationship into the previous equations and simplifying

$$X_{Solid} = \frac{-mg\theta_y}{\frac{\frac{2}{5}mR^2}{R^2}+m} \qquad (3.28)$$

$$Y_{Solid} = \frac{-mg\theta_x}{\frac{\frac{2}{5}mR^2}{R^2}+m} \qquad (3.29)$$

$$X_{Solid} = -\frac{5}{7} g \Theta_y \quad (3.30)$$

$$Y_{Solid} = -\frac{5}{7} g \Theta_x \quad (3.31)$$

$$X_{Hollow} = \frac{-mg\Theta_y}{\frac{\frac{2}{3}mR^2}{R^2} + m} \quad (3.32)$$

$$Y_{Hollow} = \frac{-mg\Theta_x}{\frac{\frac{2}{3}mR^2}{R^2} + m} \quad (3.33)$$

$$X_{Hollow} = -\frac{3}{5} g \Theta_y \quad (3.34)$$

$$Y_{Hollow} = -\frac{3}{5} g \Theta_x \quad (3.35)$$

3.1.3 System transfer function

To find the overall transfer function, the ball and plate model and DC servomotors Transfer functions had multiplied as shown in Equation (3.35), so it becomes:

$$G_{x,y}(s) = \frac{K*Kg*g}{\frac{5}{7} s^3[(Js + b)(Las + Ra) + K^2]} \quad (3.36)$$

3.2 PID Controller Design

It is difficult to design a controller with Ziegler-Nicolas for major reason; it was found that the overall system is a fifth order system which mean difficult to design controller for a higher order system. To make the control design easy the whole system is separated into two feedback loops; inner loop and outer loop as shown in Figure 3.3. The purpose of the inner loop is to control the motor gear angle position so that gear angle (θ) tracks the reference signal (ref θ). The outer loop uses the inner feedback loop to control the ball position.

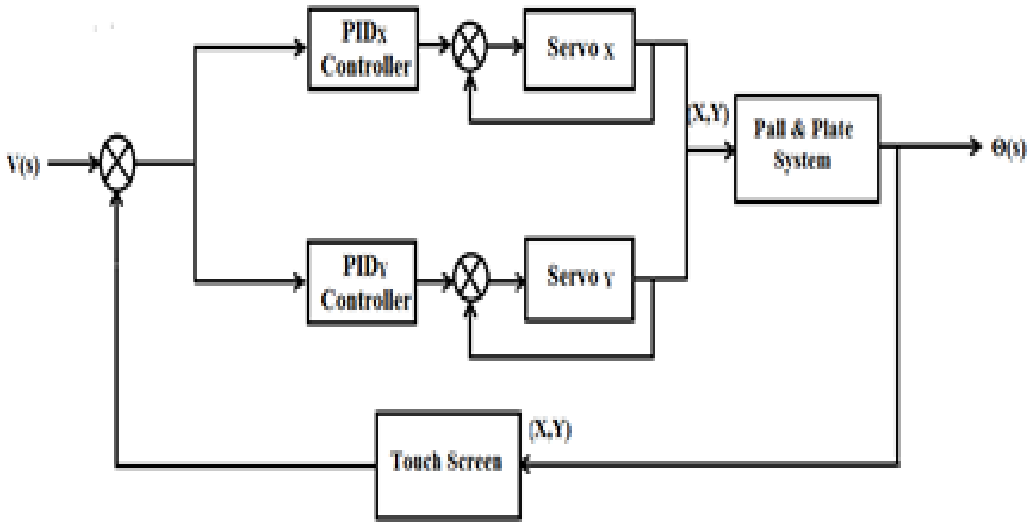

Figure 3.3: Complete system

CHAPTER FOUR
SYSTEM IMPLEMENTATION AND EXPERIMENTAL RESULTS

4.1 System Practical Model

System Practical model is separated into two parts; the mechanical part which consists of platform, and DC servomotor horns and electrical part which consists of two DC servo motor, resistive touch screen and Arduino microcontroller.

4.1.1 System mechanical part

Selection of appropriate material for a mechanical part is an essential element of all engineering projects. The main mechanical parts of the system are the base support, ball and plate as shown in Figure 4.1. Following materials are used in to build the system:

- Base support made of wood has length of (29cm) and width of (29cm) with prop has height of (13.5cm).
- Platform made of wood has diameter of (15inch).
- Two servo motors horns.
- Two lever arms.

The dimension of the physical model is shown in Figure 4.2.

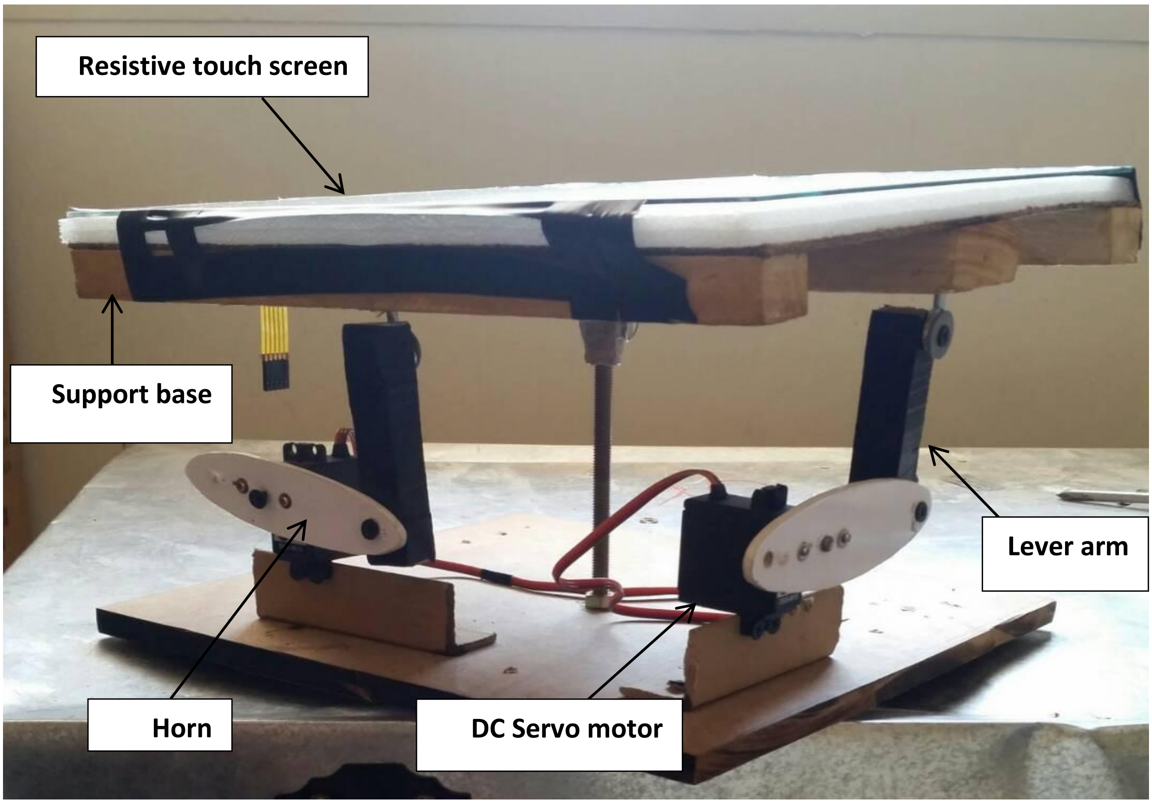

Figure 4.1: Main parts of the system

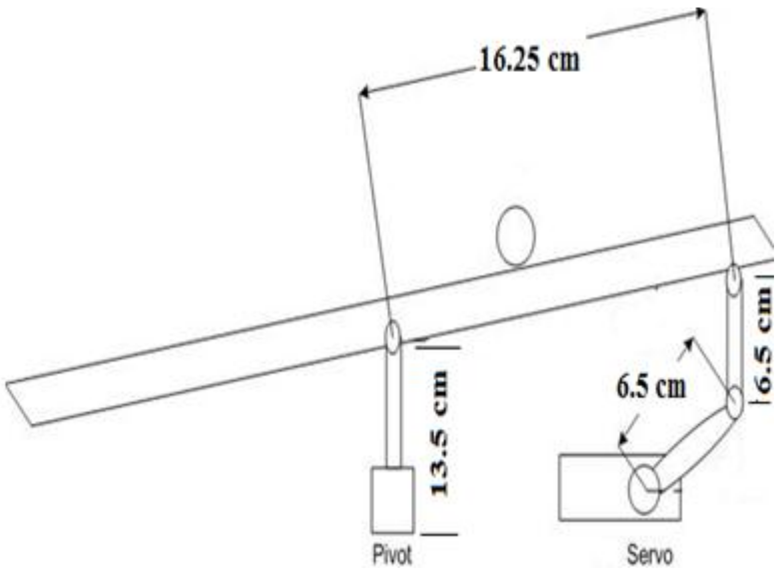

Figure 4.2: Dimensions of the physical model

4.1.2 System electrical part

The Electrical circuit consists of Arduino Uno microcontroller, two Dc servo motors and resistive touch panel.

i. Arduino Uno

It's a microcontroller board based on ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, power jack, an ICSP header and a reset button.

Using a PID library in adruino code according to use a PID as a controller, so using the PID library has many advantages like: saving a lot of time spent in making the PID algorithm, it makes the code easier and able to make it complex like using more than one PID in the same system.

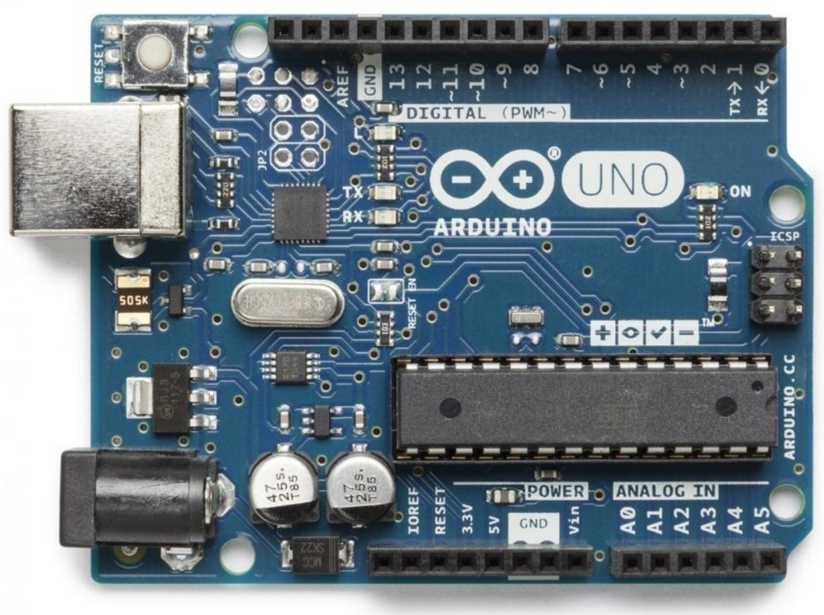

Figure 4.3: Arduino Uno

ii. DC Servomotor (DSS-M15)

DSS-M15 as shown in Figure 4.4 is a heavy-duty metal gear digital servo with 180°wide angle, high torque power, improved stability and durability.

Servo is able to work with 6V and deliver a strong torque power of over 15Kg. This DSS-M15 servo demonstrates a maximum torque of 18Kg without much vibration or heat.

Figure 4.4: DSS-M15 servo motor

iii. Touch screen

Five wire resistive touch screen was used to measure the position of the ball as shown in Figure 4.5. The response time of the touch panel, according to the manufacturer, is 20msec which was acceptable for this experiment. Four of the wires of the touch panel were connected with the four corners whereas the fifth one was used to obtain the analog signal indicating the position of the ball. The contact point is measured using the homogenous resistive surface. An even voltage gradient appears across the resistive surface when a certain potential difference is applied to the electrode pair. Another conductive layer is required to carry out a high-resistance voltage measurement. Hence, the touch screen can be considered as a pressure controlled electric switch. The analog signal from the touch screen ranged from 0V to 5V when a potential difference of 5V is applied in one direction. As the touch screen had only one output pin or the sense pin, both x and y coordinates could not be obtained simultaneously. The configuration of the touch screen used is shown in Table 4.1.

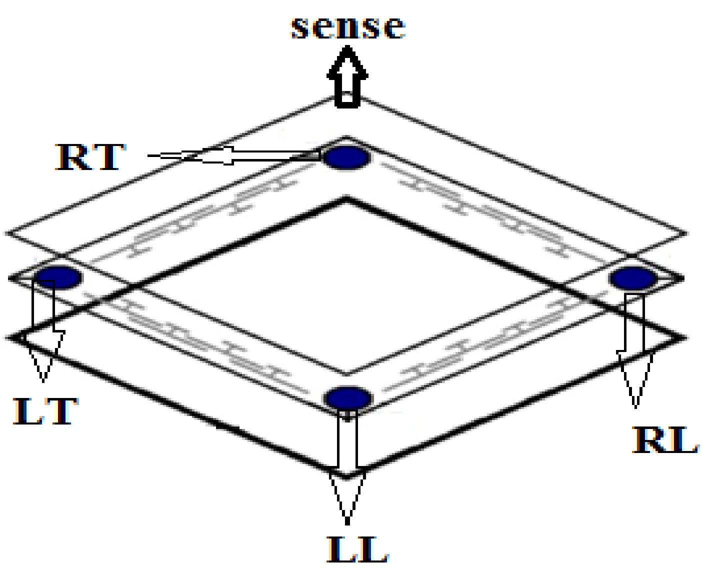

Figure 4.5: Five wire resistive touch screen

Table 4.1: Pin configuration of the touch panel

Mode	LT	RT	LL	RL	Sense
$X_{coordinate}$	Ground	Vcc	Ground	Vcc	Hi Z/ADC
$Y_{coordinate}$	Ground	Ground	Vcc	Vcc	Hi Z/ADC

LT, RT, LL and RL in Table 4.1 represent the wires connected to the left top, right top, left lower and right lower of the touch screen respectively. As can be seen in Table 4.1, the logic signal going to RT and LL had to toggle to measure both coordinates. This was taken care of in the algorithm. The analog signal from the touch panel was converted to digital using the 10-bit analog to digital converter in the Arduino board. This means that the analog values read, which ranged from 0 to 5V, were converted to digital values between 0 and 1023. This gave the resolution of 5V/1024 units or 4.9mV per unit which was satisfactory for this purpose.

iv. Electrical system connections

Five wire Touch screen connected respectively (pin1 , pin2, pin4 , pin5, sense), (8, 2, 9, 3, A0), (analog pin) sends its input signal in analog form to the Arduino to determine the ball position in coordinates, this value represent the input of the PID to be compared with the set point. The servo motors connected to (pin 5, pin 6) which it can be used as a Pulse Width Modulation (PWM), it receives its input (angle) from PID controller. The electrical circuit connection is shown in Figure 4.6.

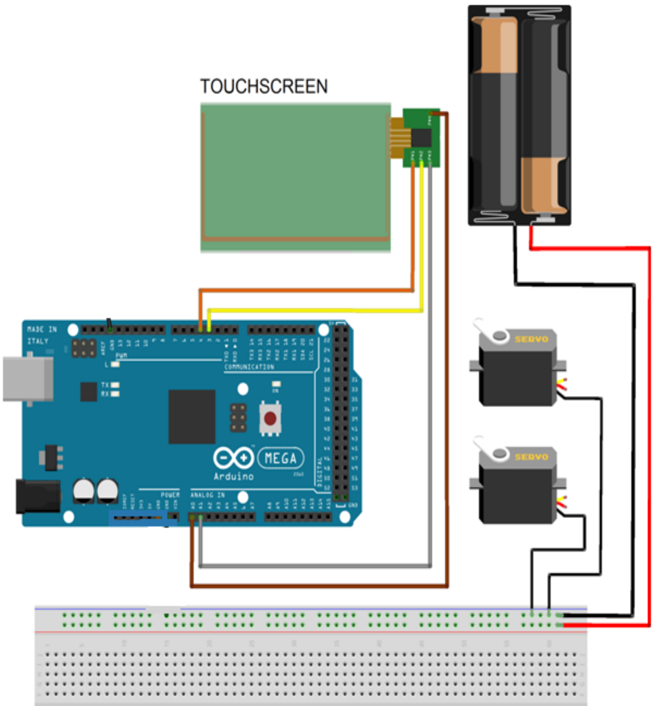

Figure 4.6: The electrical circuit connections

4.2 MATLAB Plotting Results

This section demonstrates the results of plotting using MATLAB .using design of PID controller with manual tuning method. With desired value of ball position on the touch screen, noted that this value can be changed by modifying it in the controller algorithm considering it as the set point (reference) of the control system.

By substituting the values of the system transfer function parameters the root locus plot of the system is shown in Figure 4.7.

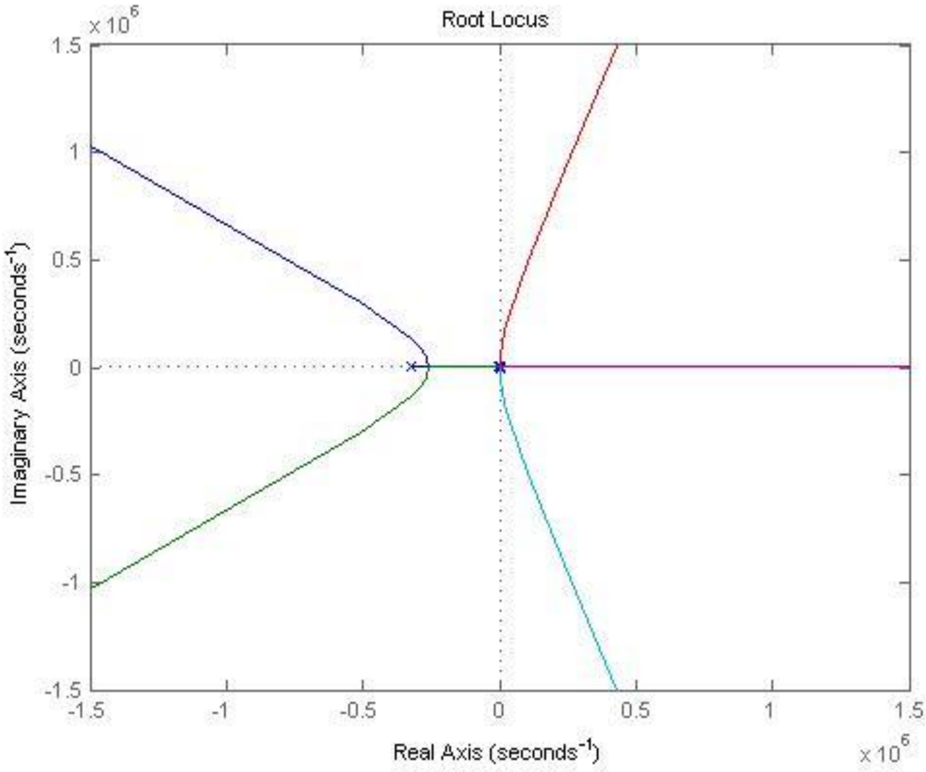

Figure 4.7: The root locus plot

By substituting the values of controller parameters with K_{px}= 2, K_{ix}= 0, K_{dx}= 0, K_{py} = 2, K_{iy} = 0, $K_{dy=}$ 0 the real time plotting of ball position as shown in Figure 4.8.

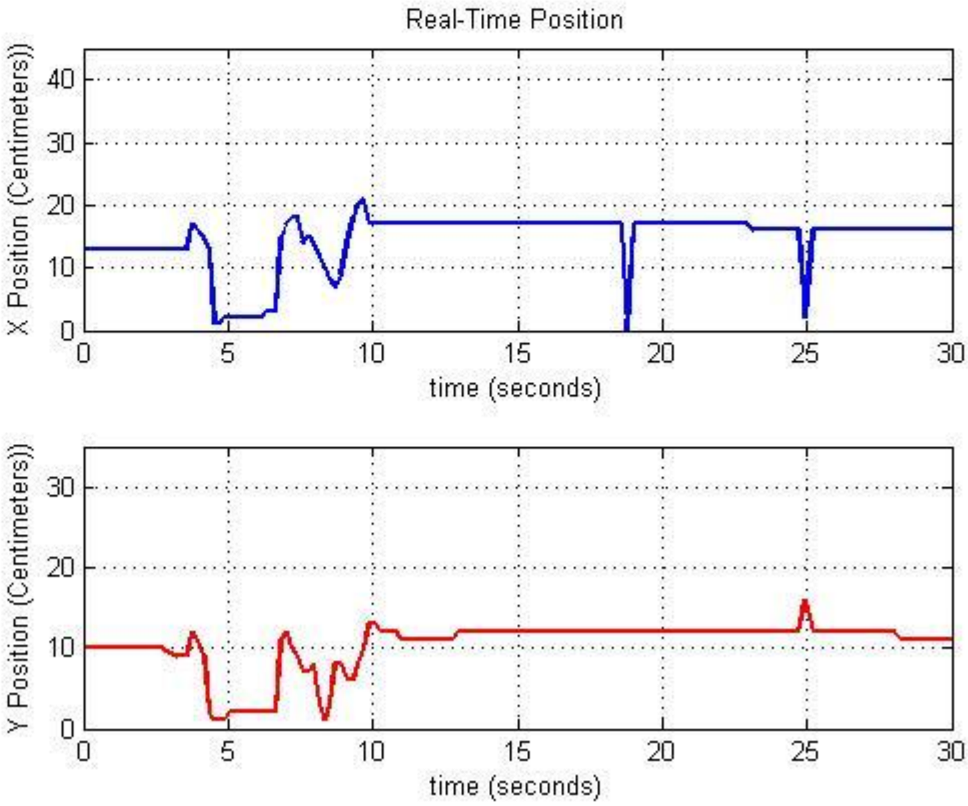

Figure 4.8: System response with K_{px}= 2, K_{ix}= 0,

K_{dx}= 0, K_{py} = 2, K_{iy} = 0, $K_{dy=}$ 0

By substituting the values of controller parameters with K_{px}= 2, K_{ix}=0.1, K_{dx}=0.7, K_{py} = 2.5, K_{iy} = 0.01, K_{dy} = 0.8 the real time plotting of ball position as shown in Figure 4.9 Adding derivative parameter decrease the overshoot .

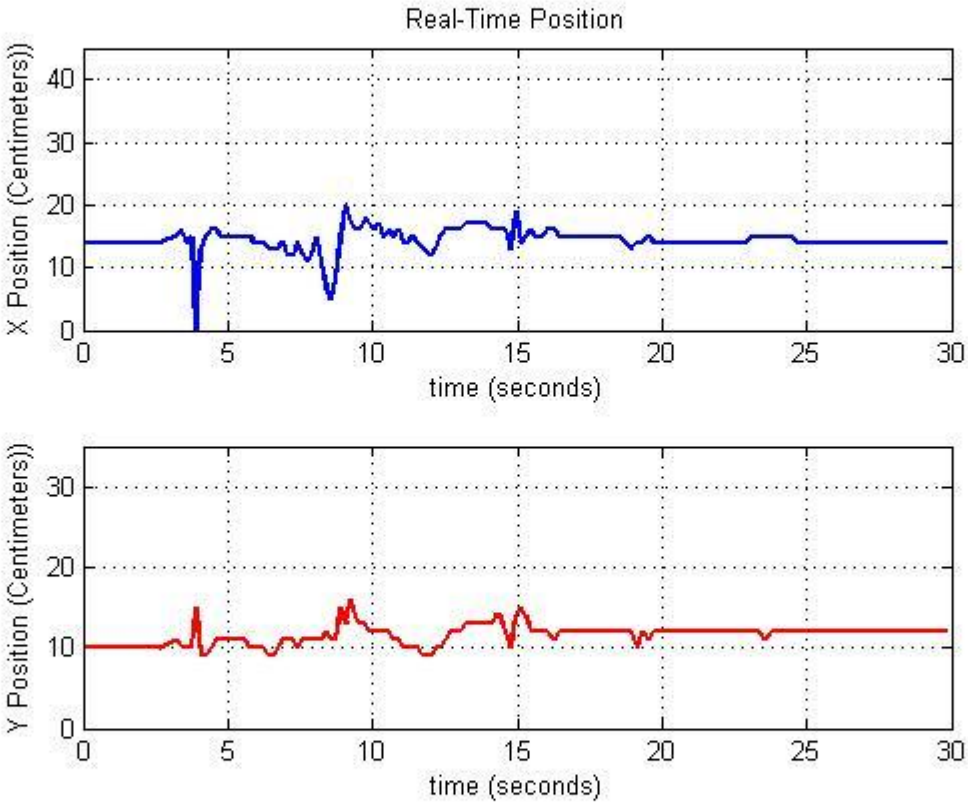

Figure 4.9: System response with K_{px}= 2, K_{ix}=0.1,
K_{dx}=0.7, K_{py} = 2.5, K_{iy} = 0.01, K_{dy} = 0.8

By substituting the values of controller parameters with K_{px}= 2, K_{ix}= 0.1, K_{dx}= 0.7, K_{py}= 2.4, K_{iy}= 0.01, K_{dy}= 0.5 the real time plotting of ball position as shown in Figure 4.10 Gives a better step response by decreasing the settling time and the overshoot.

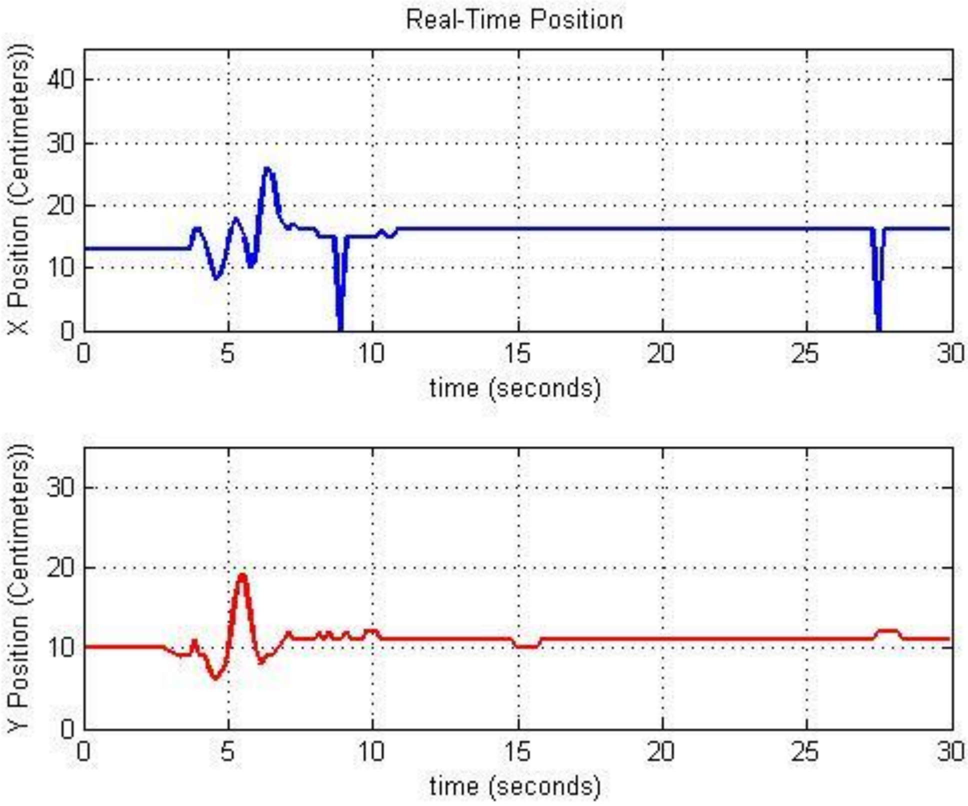

Figure 4.10: System response with K_{px}= 2, K_{ix}= 0.1, K_{dx}= 0.7, K_{py}= 2.4, K_{iy}= 0.01, K_{dy}= 0.5

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

A mathematical model of the ball and plate system was developed by using physical and electrical laws. A simplified mathematical model was derived by system parameters. The controller parameters values (K_p, K_i and K_d) were obtained by using manual tuning method from practical model so as to perform best system response. From experimental results, it is found that the best controller parameters which gave the best response of the system are: K_{px}= 2, K_{ix}= 0.1, K_{dx}= 0.7, K_{py}= 2.4, K_{iy}= 0.01, K_{dy}= 0.5. The accuracy of the system is tested adjusting the position of the ball at three different points and it found that the accuracy doesn't affected by changing the set point.

5.2 Recommendations

- It is recommended that the system can be controlled by more advanced control techniques, such as robust control, fuzzy logic control and neuron fuzzy control.
- Use IR remote control to determine the set point.
- Compare between ball and plate system experimental results and simulation results.

REFERENCES

[1] K. Ogata, Modern control engineering. Englewood Cliffs, N.J.: Prentice-Hall, 1970.

[2] Awtar, S., Bernard, C., Boklund, N., Master, A., Ueda, D., & Craig, K. Mechatronic design of a ball-on-plate balancing system Mechatronics, 12(2), 217-228 , 2002.

[3] B. L. Theraja and A. K. Theraja, A text book of electrical technology. RamNagar, New Delhi: S. Chand & Company Ltd., 2005.

[4] S. Monk, 30 Arduino projects for the evil genius. New York: McGraw-Hill, 2010.

[5] N. S. Nise, Control systems engineering. [Hoboken, NJ]: Wiley, 2004.

[6] A.Naseraldeen A.. A. Taifour Ali*, Ahmed A. M., Almahdi H. A., Osama A. Taha. Design and Implementation of Ball and Beam System Using PID Controller, Science and Education Publishing, 3(1), 1-4, 2017.

APPENDIX A

ARDUINO CODE

```
/////////Ball and Plate///////////////////////////////

/*

BALL AND PLATE PID CONTROL

*/

//////////////////////////////////////////////////////

///Libraries///

#include <PID_v1.h>

#include<Servo.h>

const int pin2 = 4; // Pin 2 at touchs screen measure X

const int pin1 = 8; // Pin 1 at touchs screen 5V

const int pin5 = 3;// Pin 5 at touchs screen is GND

const int pin4 = 9; // Pin 4 at touchs screen measure Y

const int sense = A0; // Pin 3 at touchs screen


// PID values

double SetpointX, InputX, OutputX; //for X

double SetpointY, InputY, OutputY; //for Y


// servos variables

Servo servoX; //X axis

Servo servoY; //Y axis
```

```
/////TIME SAMPLE
int Ts = 120;
//PID const
float KpX = 2;
float KiX = 0;
float KdX = 0.3;

float KpY = 2.5;
float KiY = 0;
float KdY = 0.5;

double Xoutput,Youtput;

//INIT PID
PID myPIDX(&InputX, &OutputX, &SetpointX, KpX, KiX, KdX, DIRECT);
PID myPIDY(&InputY, &OutputY, &SetpointY,KpY,KiY,KdY, REVERSE);

void setup()
{
  //Int Setpoint
  SetpointX=16;
  SetpointY=11;
  servoX.attach(5);
  servoY.attach(6);
```

```
  servoX.write(100);
  servoY.write(83);
pinMode(pin1,OUTPUT);
pinMode(pin2,OUTPUT);
pinMode(pin4,OUTPUT);
pinMode(pin5,OUTPUT);
   Serial.begin(9600);

  myPIDX.SetMode(AUTOMATIC);
  myPIDX.SetOutputLimits(-43, 50);
  myPIDY.SetMode(AUTOMATIC);
  myPIDY.SetOutputLimits(-38, 57);
  // TIME SAMPLE
  myPIDX.SetSampleTime(Ts);
  myPIDY.SetSampleTime(Ts);
  /////
  delay(100);

  ///
 }

void loop()
{
```

```
 SetpointX=16.25;
 SetpointY=12.25;


 InputX= Xcoordinates();  // read and convert X coordinate
 InputY= Ycoordinates(); // read and convert Y coordinate
 myPIDX.Compute();  //action control X compute
 myPIDY.Compute(); //   action control  Y compute
 Xoutput = 97 + OutputX;
 Youtput = 83 + OutputY;
Serial.print(InputX);
Serial.print("\t");
Serial.print(InputY);
 Serial.print("\t");
Serial.print(Xoutput);
Serial.print("\t");
Serial.println(Youtput);
if(InputX > SetpointX + 0.5 && InputX < SetpointX - 0.5 && InputY >
SetpointY + 0.5 && InputY < SetpointY - 0.5 )
{
 servoX.write(97);//control
 servoY.write(83);//control
}
else
{
```

```
  servoX.write(Xoutput);//control
  servoY.write(Youtput);//control
 }
  }
  double Xcoordinates()
{
double xVal;
digitalWrite(pin1,HIGH);
digitalWrite(pin2,HIGH);
digitalWrite(pin4,LOW);
digitalWrite(pin5,LOW);
 xVal = analogRead(sense);
delay(20);
xVal = analogRead(sense);
xVal = map(xVal,267,767,0,32);
return xVal;
}
double Ycoordinates()
{
double yVal;
digitalWrite(pin1,LOW);
digitalWrite(pin2,HIGH);
digitalWrite(pin4,LOW);
digitalWrite(pin5,HIGH);
```

```
yVal = analogRead(sense);

delay(20);

yVal = analogRead(sense);

yVal = map(yVal,234,774,0,24);

return yVal;

}
```

APPENDIX B

MATLAB CODE

```
clc
clear all
prev = 0;
s = serial('COM4','BaudRate',250000);
fopen(s);
x=0;
t(1)=0;
flushinput(s);
tic;
while(~isempty(s) && toc <= 30)

   x = x+1;
   t(x) = toc;
   y(x) = str2double(fgets(s))
   z(x) = str2double(fgets(s))
   drawnow
   subplot(2,1,1)
   plot(t,y,'linewidth',2)
    axis([0 30 0 45]);
    title('Real-Time Position');
   xlabel('time (seconds)');
   ylabel('X Position (Centimeters))');
    grid on
   hold on
   subplot(2,1,2)
   plot(t,z,'r','linewidth',2)
   xlabel('time (seconds)');
   ylabel('Y Position (Centimeters))');
   axis([0 30 0 35]);
   flushinput(s);
   grid on

end
fclose(s);
delete(s);
clear all
```