

**Sudan University of Science & Technology**  
**College of Postgraduate**



**Improving the accuracy of pick and place  
robotic arm using PID controller**

**تحسين دقة ذراع اخذ و وضع صناعي باستخدام  
المتحكم التناسبي التكاملي التفاضلي**

A thesis Submitted for Partial Fulfillment for the  
Requirement of M.Sc. Degree in Mechatronics Engineering

**Presented by:-**

Eslam Haroon Ahmed

**Supervisor:-**

Dr.Alaa Eldein Awooda

July 2018

# الآية

قال تعالى

تَرْفَعُ دَرَجَاتٍ مِّنْ تَشَاءُ وَفَوْقَ كُلِّ ذِي عِلْمٍ عَلِيمٌ

صدق الله العظيم

سورة يوسف الآية 76

## **Dedication**

I dedicate this project to God the almighty my creator, my source of inspiration, wisdom, knowledge and understanding, he has been the source of strength throughout this project. Also I dedicate this work to My Family and my friends Thank all, my love for you all can never be quantified.

# **ACKNOWLEDGEMENT**

I would like to express my special thanks of gratitude to my supervisor

**Dr Alaa eldein Awooda**

As well as to the engineers **.Eng: Ahmed abdalbarry, Eng: Ashraf mohammed** and **Eng: Mohammed** who helped me to do this project as it should be I am really thankful to them.

## **Abstract**

The industry is moving from current state of automation to Robotization, to increase productivity and to deliver uniform quality. One type of robot commonly used in industry is a robotic manipulator or simply a robotic arm. It is an open or closed kinematic chain of rigid links interconnected by movable joints. In some configurations, links can be considered to correspond to human anatomy as waist, upper arm and forearm with joint at shoulder and elbow. At end of arm a wrist joint connects an end effector which may be a tool and its fixture or a gripper or any other device to work. This research aim to implement the controller of the robotic arm and improve the accuracy of it in order to achieve high response and performance the movement of the arm is controlled through potentiometer and to move forward and backward, left, right, motions. Five motors are connected to microcontroller arduino for the body movement. The PID is used here to make the process of positioning of the arm more accurate.

By using hardware components ( robotic arm, MPU6050 sensor L298 ,l292 driver arduino )and software components( arduino code PID controller proteus simulation ) the angle of the arm is read by the sensor and the PID controller has adapted the position of the arm and then the accuracy of the robotic arm has improved The robotic arm now is able to reach to the stability in its position as fast as possible.

## المستخلص

تنتقل الصناعة من حالة التشغيل الآلي الحالية إلى الاتمه ، لزيادة الإنتاجية وتقديم جودة موحدة.

أحد أنواع الروبوتات الشائعة الاستخدام في الصناعة هو المعالج الآلي أو ببساطة ذراع آلية. وهي عبارة عن سلسلة حركية مفتوحة أو مغلقة من الروابط الجامدة المترابطة مع المفاصل المنقولة في بعض التكوينات ، يمكن اعتبار الارتباطات متوافقة مع تشريح جسم الإنسان مثل الخصر ، الذراع العلوي والساعد مع المفصل عند الكتف والمرفق في نهاية الذراع ، يربط مفصل المعصم طرفًا مستجيبًا قد يكون أداة تثبيت أو أي جهاز آخر للعمل.

يهدف هذا البحث إلى تنفيذ تحكم الذراع الروبوتية وتحسين دقة ذلك من أجل تحقيق استجابة عالية وأداء. يتم التحكم في حركة الذراع من خلال المقادير المتغيرة للتحرك للأمام وللخلف

إلى اليسار و إلى اليمين ترتبط خمس محركات بميكرو كونترولر اردوينو لحركة الجسم يتم استخدام خوارزمية المتحكم التناسبي التكاملية التفاضلية هنا لجعل عملية تحديد موضع الذراع دقيقة

بإستخدام المكونات الملموسة و غير الملموسة تمت قراءة كل من الزاوية الخاصة بالذراع بإستخدام الحساس الخاص بقراءة الزوايا و أيضا تمت موازنه موضع الروبوت باستخدام المتحكم التناسبي التكاملية التفاضلية

## TABLE OF CONTENTS

	Tittle	Page
	الآية	I
	Dedication	II
	Acknowledgement	III
	Abstract	IV
	مستخلص	V
	Table of Contents	VI
	List of Tables	IX
	List of Figures	X
	List of Abbreviations	XII
<b>CHAPTER ONE</b>		
<b>INTRODUCTION</b>		
	1.1introduction	1
	1.2 Problem statement	3
	1.3 proposed solution	3
	1.4 Aims and Objectives	4
	1.5 Methodology	5
	1.6 Thesis layout	5
<b>CHAPTER TWO</b>		
<b>LITERATURE REVIEW</b>		
	2.1 Over View	6
	2.2 previous work	6
	2.3 robotic arm	8
	2.3.1 technical description	9
	2.3.2 robot programming and interface	11
	2.3.3 components of the robotic arm	12
	2.4 PID controller	12
	2.4.1Proportional controller (P)	12
	2.4.2 Integral controller (I)	13
	2.4.3 Derivative controller(D)	14
	2.4.4 Proportional Integral controller (PI)	15
	2.4.5 proportional Derivative controller (PD)	15
	2.4.6 PID Algorithms	16
	2.4.7 Interactive PID algorithms	17

2.4.8 Non PID Interactive algorithms	17
2.4.9 parallel PID algorithms	18
2.5 PID Tuning	19
2.5.1 Manual tuning method	20
2.5.2 Ziegler - Nichols method	20
2.5.3 Cohen - tuning method	21
2.6 Arduino controller	22
2.7 Motion Processing Unit (MPU6050)	23
2.7.1 Software of the MPU6050	26
2.8 Driver L298	27
2.8.1 Specification of the driver L298	27
2.9 DC motor	28
2.10 Main Types of DC Motors	29
2.10.1 Series DC Motors	29
2.10.2 Shunt DC Motors	29
2.10.3 Compound DC Motors	29
2.10.4 Permanent Magnet DC Motors	30

### **CHAPTER THREE**

#### **RESEARCH METHDOLOGY**

3. Research methodology	31
3.1 over view	31
3.2 research activities	32
3.3 system overview	32
3.4 circuit diagram of the overall system	35
3.5 circuit components	36
3.5.1 Microcontroller Arduino UNO	36
3.5.2 Motion processing unit (MPU6050)	36
3.5.3 Potentiometer	37
3.5.3.1 potentiometer in arduino	37
3.5.4 DC motor in arduino	37

### **CHAPTER FOUR**

#### **System Implementation and Testing**

4. System Implementation and Testing	39
4.1 System Implementation	39

### **CHAPTER FIVE**

#### **CONCLUSION AND RECOMMENDATIONS**

5.1 Conclusion	47
----------------	----



5.2 Recommendations	48
References	49

---

## Appendix

### LIST OF TABLES

Table No	Title	Page
2.1	characteristics of PID controller	16
2.2	Ziegler- Nichols method	20
2.3	cohen-coon method	22

## LIST OF FIGURES

Figure No	Title	Page
2.1	PID Controller	15
2.2	Interactive algorithm	17
2.3	Non interactive algorithm	18
2.4	Parallel algorithm	19
2.5	Cohen-coon PID tuning method	21
2.6	Pin description of arduino	23
2.7	Pin description of MPU6050	26
2.8	Pin description of Driver L293	28
2.9	DC motor	30
3.1	Research activities	31
3.2	Block diagram of the overall system	33
3.3	flow chart of the overall system	34
3.4	Simulation of the overall system	35
3.5	Connection of MPU6050 to arduino	36
3.6	Connection potentiometer to arduino	37
3.7	Connection of DC motor to arduino	38
4.1	the overall system's components connected together	39
4.2	when the robot is not working	40
4.3	the COM4 shows effect of the PID and the readings of the sensor when the robot is not working	40
4.4	when the arm start to work	41
4.5	the COM4 shows effect of the PID and the readings of the sensor when the robot start to work	41
4.6	when the arm moves down	42
4.7	COM4 shows effect of the PID and the readings of the sensor when the robot moves down	42
4.8	when the arm moves up	43
4.9	COM4 shows effect of the PID and the readings of the sensor when the robot moves up	43

4.10	when the arm moves left	44
4.11	COM4 shows effect of the PID and the readings of the sensor when the robot moves left	44
4.12	when the arm moves right	45
4.13	COM4 shows effect of the PID and the readings of the sensor when the robot moves right	45
4.14	when the gripper is on	46
4.15	the COM4 shows effect of the PID and the readings of the sensor when the gripper is on	46

## LIST OF ABBREVIATIONS

Abbreviation	Description
ADC	Analog to Digital Converter
CMOS	Complementary Metal Oxide Semiconductor
DC	Direct Current
DMP	Digital Motion Processing
IDE	Integrated Development Environment
LED	Light Emitting Diode
MEMs	Micro-Electro-Mechanical systems
MPU	Motion Processing Unit

PD	Proportional Derivative
PI	Proportional Integral
PID	Proportional Integral Derivative
PV	Process Variable
SP	Set Point

# **Chapter One**

## **Introduction**

# Chapter One

## 1. Introduction

### 1.1 Overview

Robot technology or so-called robotics is one of the most advanced artificial intelligence technologies in terms of applications that offer complete solutions to problems

It has currently become the most important devices on which it depends; it has great importance in life and in the scientific, industrial, medical and other fields

Robot in general is a machine especially one programmable by a computer capable of carrying out a complex series of actions automatically. Robots can be guided by an external control device or the control may be embedded within. Robots may be constructed to take on human form but most robots are machines designed to perform a task with no regard to how they look.

The robot usually simulates one skill in human. The simplest form of robot is generally composed of three parts; the first is the mechanical arm, and transmission mechanism and the actuator. [1] Intelligent robots have an additional sensor-related part that makes it aware of the surrounding environment and modifies its processes according to the variables that occur in the environment

[1]

The advances of the robots are that a clear increasing in productivity, a single robot in a unit of production can increase the productivity of the system if it is better

used. Reducing the expenses the use of smart machines such as robot reduces long-term investment spending, but this of course does not apply to limited tasks or small projects.

As well as overcoming the lack of labor force, replacing many manual labors in many professions that were almost extinct or difficult to perform. [1] Provide flexibility in the industrial business as the robot can be reprogrammed to work in different works.

pick and place mechanical arm is a human controlled based system that detects the object, picks that object from source location and places at desired location. For detection of object, human detect presence of object and move machine accordingly. With many end-of-arm-tooling options available, pick and place robots can be customized to fit specific production requirements. [1]

The use of pick and place for placing products and transfer of the products between different stations in the packaging lines is very common in all industries. High speed pick-and-place robots for placing small items like candy and cookies in packages are often combined with a visual observation system for identifying products.

## **1.2 Problem Statement**

Manual movement of the robotic using human was suffering from non accurate, position, movement, time consuming and cost the industries are moving from current state of automation to Robotization, to increase productivity and to deliver uniform quality. The pick and place mechanical arm is a human controlled based system that detects the object, picks that object from source location and places at desired location For detection of object, human detect presence of object and move machine accordingly.

## **1.3 Proposed Solution**

Choosing the suitable controller for the robotic arm and improving the accuracy of it is a one of the most important things in the robotic arm field and this are going to be taken in to consideration in order to achieve high performance. Adding to that PID controller is going to be used to correct the errors.



### **1.4 Aim and Objectives**

This research aim to implement the controller of the robotic arm and improve the accuracy of it in order to achieve high response and performance to achieve this aims:

1. Suitable controller for the robotic arm will be designed
2. The system will be simulate for improving the performance of the robotic arm
3. Comparison of the proposed controller with a well defined one will be run for performance evaluation

## **1.5 Methodology**

The first step of the system is to sense the robotic arm position using MPU5060 sensor which will sense the angle of the robotic arm. The signal will be send to the arduino UNO microcontroller to apply PID for error correction of the arm position

The robot is going to be programmed by using arduino UNO adding to that the potentiometers are used to control the movement of the motors of the robotic arm and the sensor.

## **1.6 Thesis Outlines**

This thesis is organized in five chapters:

Chapter two gives a solid review for the data related to pick and place robotic arm besides it summarizes related previous studies of researches in the field of pick and place robotic arm and highlights deficiencies and problems they faced.

Chapter three describes discusses and justifies the research approach, methods and techniques use in this

research work. General methodology frameworks followed by specific sub framework of all phases and way forward to achieve research objective are presented and explained.

Chapter four the results will be highlighted and analyzed and an evaluation of performance of developed system is provided.

Chapter five concludes the overall research work and gives recommendations for future work.

# **Chapter Two**

# **Literature Review**



## **Chapter Two**

### **2. Literature Review**

#### **2.1 Overview**

This chapter is about pick and place robotic arm, controllers and previous case studies, these studies which have been done previously by other researchers. It is very essential to refer to the variety of sources in order to gain more knowledge and skills to complete this project. These sources include reference books, thesis, and papers.

#### **2.2 Previous Works**

There are a lot of researches that had been done similar works with this project, mostly by foreign manufacturers, universities and colleges.

Robot-arm pick and place behavior programming system using visual perception this paper presents the programming of a robot-arm system for carrying out flexible pick and place behavior using visual perception. Object manipulation from visual data involves determining the pose of the object with respect to the manipulator. Taking into account that visual positioning is an ill-posed problem due to the perspective projection, this system uses a camera and a sensor distance, and both of them mounted on a robot-arm tool adapter for locating (positioning and orienting) objects. On the other hand, this programming system is modular, composed of different dynamic link libraries to be independent with

the hardware and offers a friendly graphic interface where the user can define pick and place object locations on the image space [2]

A method for Estimating Servomotor Parameters of a Parallel Robot for Rapid Pick-and-Place Operations.

This paper presents a 2-DOF high-speed translational parallel manipulator as an object of study; this paper presents an approach that enables the servomotor parameters of parallel robots for pick-and-place operations to be estimated in an effective manner using the singular value decomposition. These parameters include the moment of inertia, speed, torque, and power of the motor required for producing the specified velocity and acceleration of the end effector. An example is given to determine these parameters of a device for the rechargeable battery quality inspection [3]

Optimal Design of a 4-DOF Parallel Manipulator: From Academia to Industry

This paper presents an optimal design of a parallel manipulator aiming to perform pick-and-place operations at high speed and high acceleration. After reviewing existing architectures of high-speed and high-acceleration parallel manipulators, a new design of a 4-DOF parallel manipulator is presented, with an articulated traveling plate, which is free of internal singularities and is able to achieve high performances. The kinematic and simplified, but realistic, dynamic

models are derived and validated on a manipulator prototype. Experimental tests show that this design is able to perform beyond the high targets, i.e., it reaches a speed of 5.5 m/s and an acceleration of 165 m/s<sup>2</sup>. The experimental prototype was further optimized on the basis of kinematic and dynamic criteria. Once the motors, gear ratio, and several link lengths are determined,

A modified design of the articulated traveling plate is proposed in order to reach a better dynamic equilibrium among the four legs of the manipulator. The obtained design is the basis of a commercial product offering the shortest cycle times among all robots available in today's market [4]

Accelerometer-based control of an industrial robotic arm. This paper presents an accelerometer-based system to control an industrial robot using two low-cost and small 3-axis wireless accelerometers. These accelerometers are attached to the human arms, capturing its behavior (gestures and postures). An Artificial Neural Network (ANN) trained with a back-propagation algorithm was used to recognize arm gestures and postures, which then will be used as input in the control of the robot. The aim is that the robot starts the movement almost at the same time as the user starts to perform a gesture or posture (low response time). The results show that the system allows the control of an industrial robot in an intuitive way.

However, the achieved recognition rate of gestures and postures (92%) should be improved in future, keeping the compromise with the system response time (160 milliseconds). Finally, the results of some tests performed with an industrial robot are presented and discussed. [5]

## 2.3 Robotic Arm

An industrial robot is a [robot](#) system used for [manufacturing](#). Industrial robots are automated, programmable and capable of movement on two or more axes.

Typical applications of robots include [welding](#), painting, assembly, [pick and place](#) for [printed circuit boards](#), [packaging and labeling](#), [palletizing](#), product inspection, and testing; all accomplished with high endurance, speed, and precision. They can help in material handling and provide interfaces [6]

### 2.3.1 Technical Description

Several parameters are used for the robotic arm

- Number of axes- two axes is required to reach any point in a plane; three axes are required to reach any point in space. To fully control the orientation of the end of the arm (i.e. the wrist) three more axes ([yaw, pitch, and roll](#)) are required. Some designs (e.g. the SCARA robot)



trade limitations in motion possibilities for cost, speed, and accuracy.

- [Degrees of freedom](#) - this is usually the same as the number of axes.
- [Working envelope](#) - the region of space a robot can reach.
- [Kinematics](#) - the actual arrangement of rigid members and [joints](#) in the robot, which determines the robot's possible motions. Classes of robot kinematics include articulated, cartesian, [parallel](#) and SCARA.
- Carrying capacity or [payload](#) - how much weight a robot can lift.
- Speed - how fast the robot can position the end of its arm. This may be defined in terms of the angular or linear speed of each axis or as a compound speed i.e. the speed of the end of the arm when all axes are moving.
- Acceleration - how quickly an axis can accelerate. Since this is a limiting factor a robot may not be able to reach its specified maximum speed for movements over a short distance or a complex path requiring frequent changes of direction.
- Accuracy - how closely a robot can reach a commanded position. When the absolute position of the robot is measured and compared to the commanded position the error is a measure of accuracy. Accuracy can be improved with external sensing for example a vision system or Infra-Red. See [robot calibration](#). Accuracy can

vary with speed and position within the working envelope and with payload (see compliance).

- Repeatability – how well the robot will return to a programmed position. This is not the same as accuracy. It may be that when told to go to a certain X-Y-Z position that it gets only to within 1 mm of that position. This would be its accuracy which may be improved by calibration. But if that position is taught into controller memory and each time it is sent there it returns to within 0.1mm of the taught position then the repeatability will be within 0.1mm.
- Motion control – for some applications, such as simple pick-and-place assembly, the robot need merely return repeatedly to a limited number of pre-taught positions. For more sophisticated applications, such as welding and finishing ([spray painting](#)), motion must be continuously controlled to follow a path in space, with controlled orientation and velocity.
- Power source – some robots use [electric motors](#), others use [hydraulic](#) actuators. The former are faster, the latter are stronger and advantageous in applications such as spray painting, where a spark could set off an [explosion](#); however, low internal air-pressurization of the arm can prevent ingress of flammable vapors as well as other contaminants.
- Drive – some robots connect electric motors to the joints via [gears](#); others connect the motor to the joint

directly (direct drive). Using gears results in measurable 'backlash' which is free movement in an axis. Smaller robot arms frequently employ high speed, low torque DC motors, which generally require high gearing ratios; this has the disadvantage of backlash. In such cases the [harmonic drive](#) is often used.

- Compliance - this is a measure of the amount in angle or distance that a robot axis will move when a force is applied to it. Because of compliance when a robot goes to a position carrying its maximum payload it will be at a position slightly lower than when it is carrying no payload. Compliance can also be responsible for overshoot when carrying high payloads in which case acceleration would need to be reduced[7]

### **2.3.2 Robot programming and interfaces**

- The setup or [programming](#) of motions and sequences for an industrial robot is typically taught by linking the robot controller to a [laptop](#), desktop [computer](#) or (internal or Internet) [network](#).
- Software: The computer is installed with corresponding [interface](#) software. The use of a computer greatly simplifies the programming process. Specialized [robot software](#) is run either in the robot controller or in the computer or both depending on the system design.
- Robot simulation tools allow for robotics programs to be conveniently written and debugged off-line with the final version of the program tested on an actual robot.

The ability to preview the behavior of a robotic system in a virtual world allows for a variety of mechanisms, devices, configurations and controllers to be tried and tested before being applied to a "real world" system. Robotics simulators have the ability to provide real-time computing of the simulated motion of an industrial robot using both geometric modeling and kinematics modeling[7]

### **2.3.3 The components of the robotic arm**

The robotic arm which is used in this system consists of five motors as follow:

Motor one (DCM1): is a DC motor to open and close the gripper, bidirectional, Motor two (DCM2) is a DC motor to band the hand, bidirectional, Motor three (DCM3)] is a DC motor to rotate hand 5 RPM Motor four (DCM4)] is a DC motor to move hand 2 RPM Motor five (DCM5): it is a DC motor to rotate the gripper.

### **2.4 PID Controller**

Conventional controllers are typically used control loop feedback in industrial and control system applications. It is simply an equation that the controller used to evaluate the controlled variables which is measures and feedback to the controller. The controller then compares the feedback to the set point and generates an error. It then tries to minimize the error by incrementing or decrementing the control inputs to the process, so that

process variable moves closer to the set point. The value is examined by one or more of three proportional, integral and derivative methodologies. Each controller has their specific functions. To improve the performance, PID controller must be adjusted according to the specific applications. [8]

#### **2.4.1. Proportional (P) Controller**

P controller is mostly used in first order processes with single energy storage to stabilize the unstable process. The main purpose of the P controller is to decrease the steady state error of the system. However, despite the reduction, P control cannot manage to eliminate the steady state error of the system. In addition, it decreases the rise time and after a certain value of reduction on the steady state error, increasing K only leads to overshoot of the system replication.

(2.1)

Where:

$p_0$ : Controller output with zero error,  $p_{out}$ : Output of the proportional controller,  $K_p$ : Proportional gain,  $e(t)$ : Instantaneous process error at time t,  $e(t) = SP - PV$ , SP: Set point, PV: Process variable

#### **2.4.2 Integral (I) Controller**

An integral term increases action in relation not only to the error but also the time for which it has persisted. So, if applied force is not enough to bring the error to zero,

this force will be increased as time passes. A pure "I" controller could bring the error to zero, however, it would be both slow reacting at the start (because action would be small at the beginning, needing time to get significant), brutal (the action increases as long as the error is positive, even if the error has started to approach zero), and slow to end (when the error switches sides, this for some time will only reduce the strength of the action from "I", not make it switch sides as well), prompting overshoot and oscillations Moreover, it could even move the system out of zero error: remembering that the system had been in error, it could prompt an action when not needed. An alternative formulation of integral action is to change the electric current in small persistent steps that are proportional to the current error. Over time the steps accumulate and add up dependent on past errors; this is the discrete-time equivalent to integration

$$I = K_i \int_0^t e(\tau) d\tau$$

(2.2)

Where:

I= integral term of output,  $K_i$ =integral gain

### 2.4.3 Derivative (D) Controller

A derivative term does not consider the error (meaning it cannot bring it to zero: a pure D controller cannot bring the system to its set point), but the rate of change of error, trying to bring this rate to zero. It aims at flattening the error trajectory into a horizontal line, damping the force applied, and so reduces overshoot (error on the other side because too great applied force). Applying too much impetus when the error is small and is reducing will lead to overshoot. After overshooting, if the controller were to apply a large correction in the opposite direction and repeatedly overshoot the desired position, the output would oscillate around the set point in a constant, growing, decaying sinusoid. If the amplitude of the oscillations increases with time, the system is unstable. If they decrease, the system is stable. If the oscillations remain at a constant magnitude, the system is marginally stable

$$D = K_d \frac{de}{dt} \quad (2.3)$$

Where:

D= derivative term of output,  $K_d$ =derivative gain,  $e(t)$  instantaneous process error at time t.

#### 2.4.4 Proportional Integral (PI) Controller

PI controller is mainly used to eliminate the steady state error resulting from P controller. Integral is equal to error multiplied by the time error has persisted. In this manner, integral increases the response of the system to a given error over time until it is corrected. Mostly PI controllers are used in industries because; the noise producing derivative action is neglected. [8]

#### 2.4.5 Proportional Derivative (PD) controller

PD control combines proportional and derivative control in parallel. Proportional action provides an instantaneous response to the control error which is useful for improving the response of a stable system. Derivative action is useful for fast system response to a rapid rate of change than to a small rate of change since it has an ability to predict the future error of the system response. Also D action directly amplifies process noise.

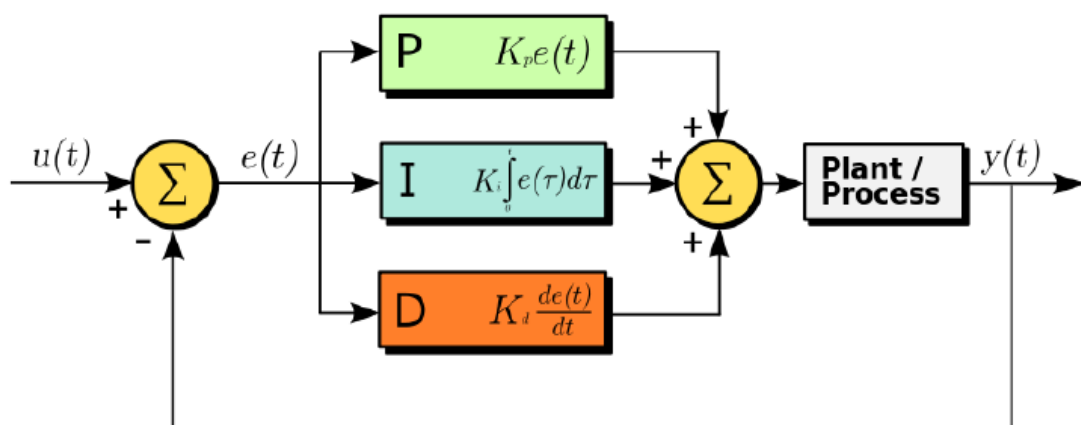


Figure 2.1 PID Controller



A proportional-integral-derivative (PID) control is by far the most common way of using feedback in natural and man-made systems. PID controllers are commonly used in industry and a large factory may have thousands of them, in instruments and laboratory equipment. In engineering applications the controllers appear in many different forms: as a standalone controller, as part of hierarchical, distributed control systems, or built into embedded components. Even though it has a relatively simple algorithm/structure, there are many subtle variations in how it is applied in industry. A PID controller will correct the error between the output and the desired input or set point by calculating and give an output of correction that will adjust the process accordingly. A PID controller has the general form

Where,  $K_p$  is the controller gain, Proportional gain,  $K_i$  is the integral time and  $K_d$  is the derivative, time,  $e$ : Error ,  $t$ : Time or instantaneous time (the present),  $\tau$ : Variable of integration; takes on values from time 0 to the present  $t$ . [8]

Table 2.1: characteristics of PID controller

CL response	Rise Time	Overshoot	Settling Time	S-S Error
$K_p$	Decrease	Increase	Small change	Decrease
$K_i$	Decrease	Increase	Increase	Eliminate

Kd	Small Change	Decreases	Decreases	Small Change
----	-----------------	-----------	-----------	-----------------

### 2.4.6 PID algorithms

The PID controller algorithm involves three separate constant parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D. Simply put, these values can be interpreted in terms of time: P depends on the present error. I on the accumulation of past errors D is a prediction of future errors, based on current rate of change. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve, a damper, or the power supplied to a heating element

Controller manufacturers arrange the Proportional, Integral and Derivative modes into three different controller algorithms or controller structures. These are called Interactive, Non interactive and parallel algorithms. Some controller manufacturers allow you to choose between different controller algorithms as a configuration option in the controller software. [9]

### 2.4.7 Interactive PID Algorithm

The oldest controller algorithm is called the series, classical, real or interactive algorithm. The original pneumatic and electronic controller had this algorithm and it is still found in many controllers today. The

Ziegler-Nichols PID tuning rules were developed for this controller algorithm [9]

$$CO = K_c \left[ E + \frac{1}{T_i} \int E \cdot dt \right] \times \left[ 1 + T_d \frac{d \cdot}{dt} \right]$$

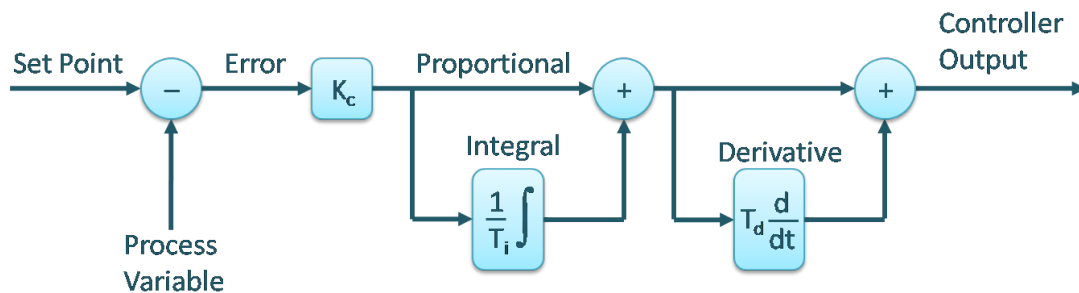


Figure (2.2): Interactive Algorithm

### 2.4.8 NonInteractive PID Algorithm

The non interactive algorithm is also called the ideal standard or ISA algorithm. The Cohen-coon was designed for this algorithm .if no derivative is used (i.e.  $T_d=0$ ), the interactive and non interactive controller algorithms are identical [9]

$$CO = K_c \left[ E + \frac{1}{T_i} \int E \cdot dt + T_d \frac{dE}{dt} \right]$$

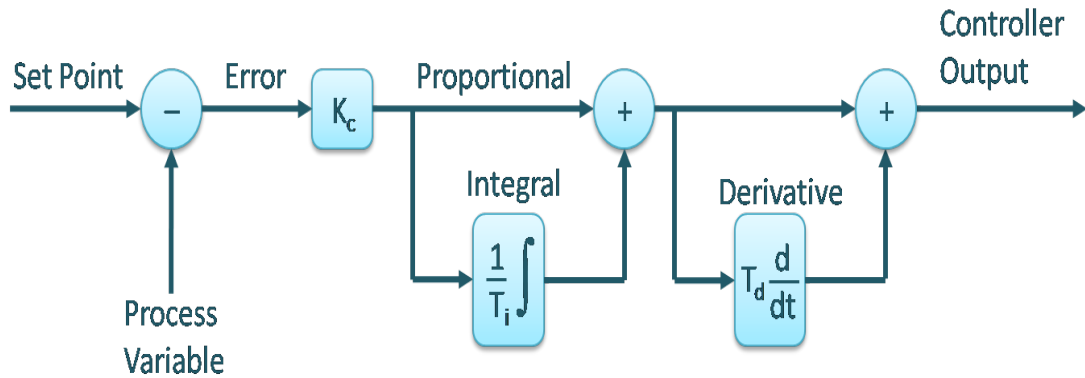


Figure (2.3) non interactive algorithm

### 2.4.9 Parallel Algorithm

Some academic textbooks discuss the parallel form of PID controller, but it is also used in some DCSs and PLCs. This algorithm is simple to understand, but not intuitive to tune. The reason is that it has no controller gain (affecting all three control modes), it has a proportional gain instead (affecting only the proportional mode). Adjusting the proportional gain should be supplemented by adjusting the integral and derivative settings at the same time. Try to not use this controller algorithm if possible (in some DCSs it is an option, so select the alternative).

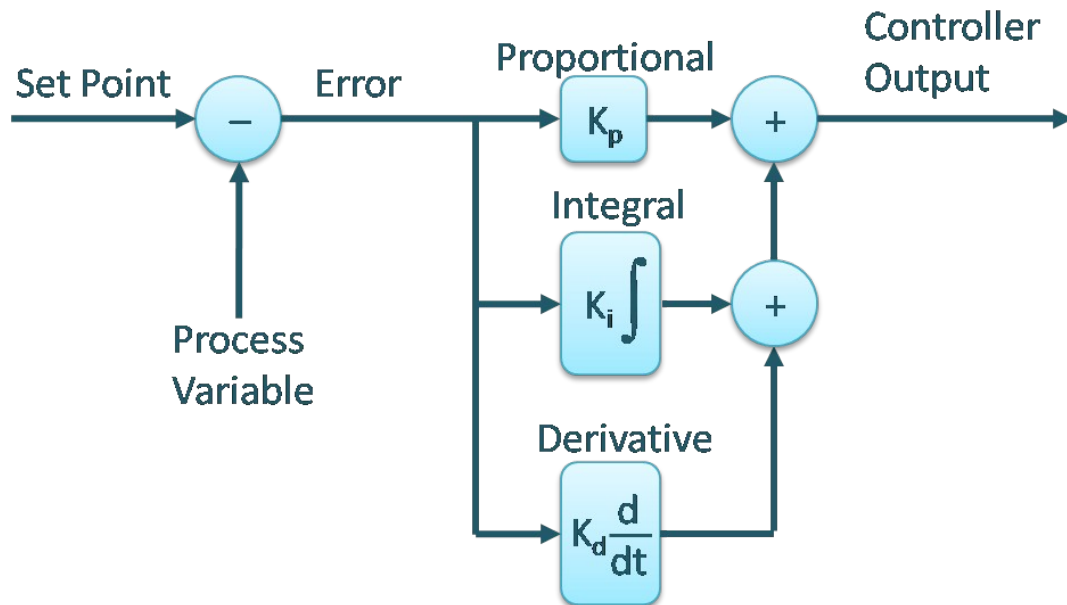


Figure (2.4) parallel algorithm

$$(2. \text{CO} = K_p \times E + K_i \int E \cdot dt + K_d \frac{dE}{dt}$$

## 2.5 PID tuning

Tuning a control loop is arranging the control parameters to their optimum values in order to obtain desired control response. At this point, stability is the main necessity, but beyond that, different systems leads to different behaviors and requirements and these might not be compatible with each other. In principle, P-I-D tuning seems completely easy, consisting of only 3 parameters, however, in practice; it is a difficult problem because the complex criteria at the P-I-D limit should be satisfied. P-I-D tuning is mostly a heuristic concept but existence of many objectives to be met

such as short transient, high stability makes this process harder. [10]

There are many tuning methods, but most common methods are as follows:

- Manual Tuning Method
- Ziegler-Nichols Tuning Method
- Cohen-Coon Tuning Method

### **2.5.1 Manual Tuning Method**

Manual tuning is achieved by arranging the parameters according to the system response. Until the desired system response is obtained  $K_i$ ,  $K_p$  and  $K_d$  are changed by observing system behavior. Although manual tuning method seems simple it requires a lot of time and experience. [10]

### **2.5.2 Ziegler-Nichols Method**

More than six decades ago, P-I controllers were more widely used than P-I-D controllers. Despite the fact that P-I-D controller is faster and has no oscillation, it tends to be unstable in the condition of even small changes in the input set point or any disturbances to the process than P-I controllers. Ziegler-Nichols Method is one of the most effective methods that increase the usage of P-I-D controllers. [10]

Table 2.2:Ziegler-Nichols P-I-D controller tuning method,  
adjusting  $K_p$ ,  $K_i$  and  $K_d$

Table (2.2) Ziegler- Nichols method

Ziegler–Nichols method giving $K'$ values (loop times considered to be constant and equal to $dT$ )			
Control Type	$K_p$	$K_i'$	$K_d'$
$P$	$0.50K_c$	0	0
$PI$	$0.45K_c$	$1.2K_p dT / P_c$	0
$PID$	$0.60K_c$	$2K_p dT / P_c$	$K_p P_c / (8dT)$

**Advantages:**

- It is an easy experiment; only need to change the P controller
- Includes dynamics of whole process, which gives a more accurate picture of how the system is behaving

**Disadvantages:**

- Experiment can be time consuming
- It can venture into unstable regions while testing the P controller, which could cause the system to become out of control
- For some cases it might result in aggressive gain and overshoot

**2.5.3 Cohen-Coon Tuning Method:**

This tuning method has been discovered almost after a decade than the Ziegler-Nichols method. Cohen-Coon tuning requires three parameters which are obtained from the reaction curve as in the Figure

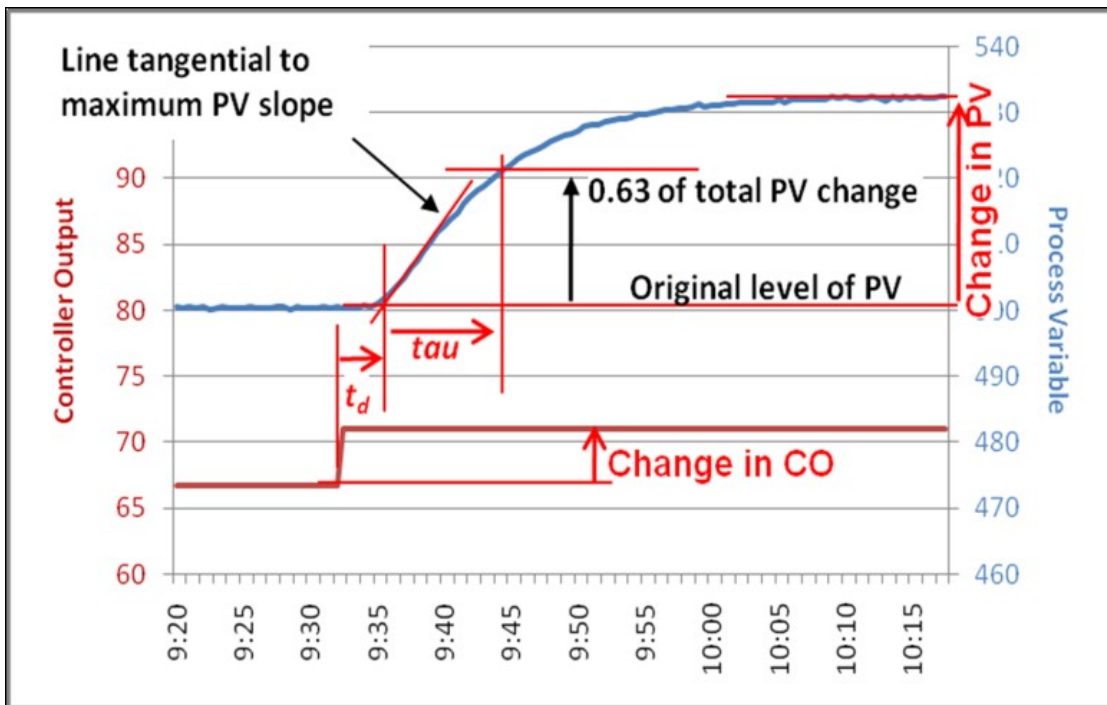


Figure 2.5: Cohen-Coon P-I-D Tuning Method

The controller is manually placed and after the process settled out a few percent of the change is made in the controller output (CO) and waited for the process variable (PV) to settle out at a new value. As observed from the graph, process gain (gp) is calculated as follow:

$$(2.10)$$

The maximum slope at the inflection point on the PV response curve is found and drawn a tangential line.  $t_d$  (dead time) is measured as taking the time difference between the change in CO and the intersection of the tangential line and the original PV level. As a final parameter  $\tau$  (time constant) as the time difference between intersection at the end of the dead time and



the 13 PV reaching 63% of its total change. After converting the time variables into the same units and applying couple of tests until to find similar result, these three variables are used to define new control parameters using the table below. [10]

Table (2.3) cohen-coon method

Table 2.3

	Controller Gain	Integral Time	Derivative Time
P Controller:	$K_c = \frac{1.03}{g_p} \left( \frac{\tau}{t_d} + 0.34 \right)$		
PI Controller:	$K_c = \frac{0.9}{g_p} \left( \frac{\tau}{t_d} + 0.092 \right)$	$T_I = 3.33 t_d \frac{\tau + 0.092 t_d}{\tau + 2.22 t_d}$	
PD Controller:	$K_c = \frac{1.24}{g_p} \left( \frac{\tau}{t_d} + 0.129 \right)$		$T_D = 0.27 t_d \frac{\tau - 0.324 t_d}{\tau + 0.129 t_d}$
PID Controller: (Noninteracting)	$K_c = \frac{1.35}{g_p} \left( \frac{\tau}{t_d} + 0.185 \right)$	$T_I = 2.5 t_d \frac{\tau + 0.185 t_d}{\tau + 0.611 t_d}$	$T_D = 0.37 t_d \frac{\tau}{\tau + 0.185 t_d}$

## 2.6 Ardui

A microcontroller is a very small computer that has digital electronic devices (peripherals) built into it that helps it control things. These peripherals allow it to sense the world around it and drive the actions of external devices.

It is an “embedded computer system” that continuously repeats software (programming) commands. Examples: Arduino Uno, Raspberry Pi, etc

Arduino microcontrollers are programmed using the Arduino IDE (Integrated Development Environment)

Arduino programs, called “sketches”, are written in a programming language similar to C and C++

Every sketch must have a setup() function (executed just once) followed by a loop() function (potentially executed many times); add “comments” to code to

make it easier to read (technically optional, but actually required [11])

Many sensors and other hardware devices come with prewritten software – look on-line for sample code, libraries (of functions), and tutorials

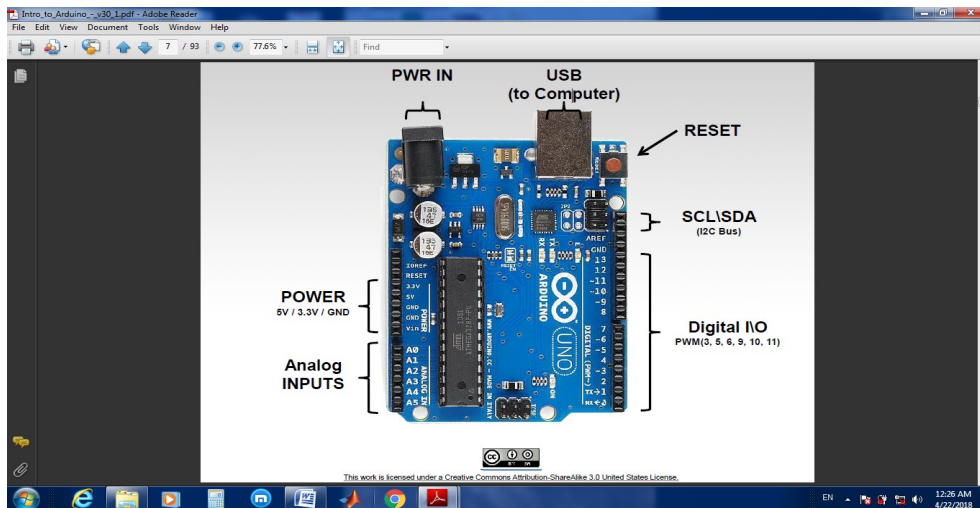


Figure (2.6) pin description of arduino

Microcontroller board. has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, 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

## 2.7 Motion Processing Units (MPUs6050)

Motion Interface is becoming a “must-have” function being adopted by Smartphone and tablet manufacturers due to the enormous value it adds to the end user experience. In smart Phones, it finds use in applications

such as phone control, Enhanced gaming, augmented reality, panoramic photo capture and viewing, and pedestrian and vehicle navigation. With its ability to precisely and accurately track user motions, Motion Tracking technology can convert handsets and tablets into powerful 3D intelligent devices that can be used in applications ranging from health and fitness monitoring to location-based services. Key requirements for Motion Interface enabled devices are small package size, low power consumption, high accuracy and repeatability, high shock tolerance, and application specific performance programmability - all at a low consumer price point. [12]

The MPU-60X0 is the world's first integrated 6-axis Motion Tracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor™ (DMP) all in a small 4x4x0.9mm package. With its dedicated I2C sensor bus, it directly accepts inputs from an external 3-axis compass to provide a complete 9-axis Motion Fusion™ output. The MPU-60X0 Motion Tracking devices, with its 6-axis integration, on-board Motion Fusion™, and run-time calibration firmware, enables manufacturers to eliminate the costly and complex selection, qualification, and system level integration of discrete devices, guaranteeing optimal motion performance for consumers. The MPU-60X0 is also designed to interface with multiple non-inertial digital sensors, such as pressure sensors, on its

auxiliary I2C port. The MPU-60X0 is footprint compatible with the MPU-30X0 families. [12]

The MPU-60X0 features three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs and three 16-bit ADCs for digitizing the accelerometer outputs. For precision tracking of both fast and slow motions, the parts feature a user-programmable gyroscope full-scale range of  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ , and  $\pm 2000^\circ/\text{sec}$  (dps) and a user-programmable accelerometer full-scale range of  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ , and  $\pm 16g$ .

An on-chip 1024 Byte FIFO buffer helps lower system power consumption by allowing the system processor to read the sensor data in bursts and then enter a low-power mode as the MPU collects more data. With all the necessary on-chip processing and sensor components required to support many motion-based use cases, the MPU-60X0 uniquely enables low-power Motion Interface applications in portable applications with reduced processing requirements for the system processor. By providing an integrated Motion Fusion output, the DMP in the MPU-60X0 offloads the intensive Motion Processing computation requirements from the system processor, minimizing the need for frequent polling of the motion sensor output.

Communication with all registers of the device is performed using either I2C at 400kHz or SPI at 1MHz (MPU-6000 only). For applications requiring faster

communications, the sensor and interrupt registers may be read using SPI at 20MHz (MPU-6000 only). Additional features include an embedded temperature sensor and an on-chip oscillator with  $\pm 1\%$  variation over the operating temperature range.

By leveraging its patented and volume-proven Nasiri-Fabrication platform, which integrates MEMS wafers with companion CMOS electronics through wafer-level bonding, Invent Sense has driven the MPU-60X0 package size down to a revolutionary footprint of 4x4x0.9mm (QFN), while providing the highest performance, lowest noise, and the lowest cost semiconductor packaging required for handheld consumer electronic devices. The part features a robust 10,000g shock tolerance, and has programmable low-pass filters for the gyroscopes, accelerometers, and the on-chip temperature sensor.

For power supply flexibility, the MPU-60X0 operates from VDD power supply voltage range of 2.375V-3.46V. Additionally, the MPU-6050 provides a VLOGIC reference pin (in addition to its analog supply pin: VDD), which sets the logic levels of its I2C interface. The VLOGIC voltage may be  $1.8V \pm 5\%$  or VDD.

The MPU6050 is the motion-processing unit equipped with both an accelerometer and a gyroscope. It is available in the [kit extension](#).

- The accelerometer is used to detect orientation and acceleration. It can also be applied to detect free-fall.

The accelerometer on the MPU6050 will return three values (in units of g), one along the x, y, and z-axes respectively.

- The gyroscope is used to detect rotation in space, and returns information in units of degrees per second. The gyroscope also returns three values along the x, y, and z-axes.[12]

### 2.7.1 Software of mpu6050

Basic functionality is provided in the MPU6050 class. It allows reading the accelerometer and gyroscope outputs for each axis. [12]

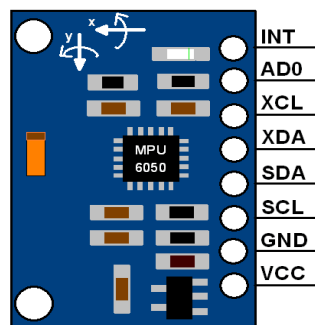


Figure (2.7) MPU6050 pin description of MPU6050

### 2.8 Driver L298

The motor driver is based on the popular [L298 chip](#). It has a peak current of 2A per motor and can be controlled via TTL from any microcontroller with 3.3-5V inputs. The on-board 78M05 5V regulator can be used to supply logic power of ~450mA to the board or external microcontroller with a supply of 7-35V.

Motor connections, power and ground are made easily using the screw terminals while logic connections are provided on a 0.1" header for easy connection using jumper leads.

LED's are provided to show motor direction and power supply. [13]

### 2.8.1 **Specifications of the driver L298**

- Input voltage range  $V_s$ : +5 V-+35 V. If the on-board regulator is used to supply external components then the supply range is +7 V ~ +35 V.
- Peak current:2A per motor. In practice this will depend on ambient conditions and air flow. Maximum power consumption in total is 20W.
- Logic voltage  $V_{ss}$ : +5 V ~ +7 V ( power taking from board: +5 V)
- Power consumption for logical part: 0 ~ 36mA
- Control signal input voltage range:
  - Low:  $-0.3V \leq V_{in} \leq 1.5V$
  - High:  $2.3V \leq V_{in} \leq V_{ss}$
- Enable signal input voltage range:
  - Low:  $-0.3 \leq V_{in} \leq 1.5V$  (control signal is invalid)
  - High:  $2.3V \leq V_{in} \leq V_{ss}$  (control signal)
- Maximum power consumption: 20W (temperature  $T = 75^\circ\text{C}$ )
- Storage temperature:  $-25^\circ\text{C}$  to  $+130^\circ\text{C}$
- Dimensions:  $55\text{m} * 39 * 33\text{mm}$

- Weight:35g

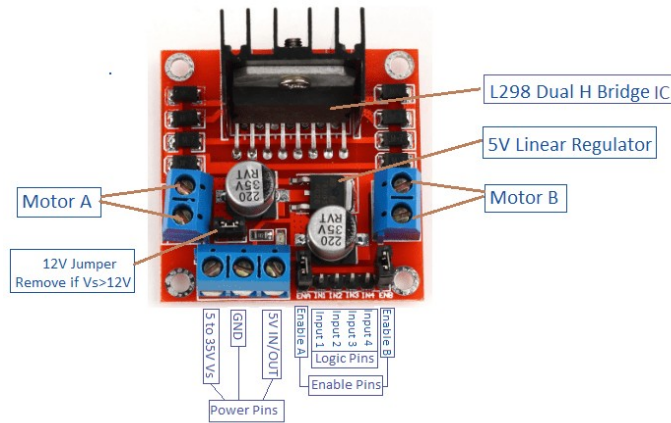


Figure (2.8) driver L298 pin description of L298

## 2.9 DC motor

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic; to periodically change the direction of current flow in part of the motor. [14]

DC motors were the first type widely used, since they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys,



and appliances. The universal motor can operate on direct current but is a lightweight brushed motor used for portable power tools and appliances. Larger DC motors are used in propulsion of electric vehicles, elevator and or in drives for steel rolling mills

A simple DC motor has a stationary set of magnets in the stator and an armature with one or more windings of insulated wire wrapped around a soft iron core that concentrates the magnetic field. The windings usually have multiple turns around the core, and in large motors there can be several parallel current paths. The ends of the wire winding are connected to a commutator. The commutator allows each armature coil to be energized in turn and connects the rotating coils with the external power supply through brushes. (Brushless DC motors have electronics that switch the DC current to each coil on and off and have no brushes.)[14]

## **2.10 Main Types of DC Motors**

### 2.10.1 Permanent Magnet DC Motors

The permanent magnet motor uses a permanent magnet to create field flux. This type of DC motor provides great starting torque and has good speed regulation, but torque is limited so they are typically found on low horsepower applications. [15]

### 2.10.2 Series DC Motors

In a series DC motor, the field is wound with a few turns of a large wire carrying the full armature current. Typically, series DC motors create a large amount of starting torque, but cannot regulate speed and can even be damaged by running with no load. These limitations mean that they are not a good option for variable speed drive applications. [15]

### 2.10.3 Shunt DC Motors

In shunt DC motors the field is connected in parallel (shunt) with the armature windings. These motors offer great speed regulation due to the fact that the shunt field can be excited separately from the armature windings, which also offers simplified reversing controls. [15]

### 2.10.4 Compound DC Motors

Compound DC motors, like shunt DC motors, have a separately excited shunt field. Compound DC motors have good starting torque but may experience control problems in variable speed drive applications.

Between the 4 types of DC motors, the potential applications are numerous. Each type has its own strengths and weaknesses. Understanding which types are most suitable for a particular application is essential. [15]

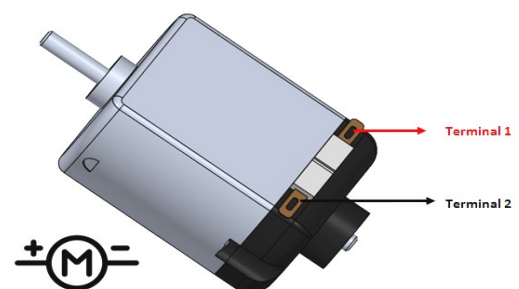


Figure (2.9) the DC motor

# **Chapter Three**

## **Research**

### **Methodology**



## **Chapter Three**

### **3. Research Methodology**

#### **3.1 Overview**

This chapter describes, discusses and justifies the research approach, methods and techniques used in this research work. General methodology frameworks followed by specific sub framework of all phases and way forward to achieve research objective are presented and explained.

The chapter contains research activities, system overview, simulation tools and flowchart.

#### **3.2 Research Activities**

The research activities are presented in figure below

**Literature Review**

**Analysis Requirement**

**Circuit Design**

**Simulation & coding**

**Modeling Implementation**

**Testing**

Figure (3.1) the research activities

Research activities began with a comprehensive literature review that shows how research related to previous studies. Analyses requirement that is followed by studying and analyze the theoretical and practical requirements needed to complete the search. The next step was a phase of circuit design where an online of components of the circuit is drawn and linked. Then, a simulation of the designed circuit is done with a test code. This is followed by implementing the practical hardwiring of circuit components and burning the application code on the memory. Finally comes to the step of testing the practical circuit and recording the results

### **3.3 System Overview**

The system overview began with block diagram of the case study figure 3.2 describes the system components and how they are connected together" which includes the input devices and output devices so the controller processes the input value and gives the output.

The block diagram contains interface tools for user in input and output

**Driver l298**

**Potentiometer**

**Potentiometer**

**Potentiometer**

**Potentiometer**

9

**Potentiometer**

**Motor 5**

The robotic arm movements are controlled through potentiometers (variable resistances).when the direction of the potentiometer is changed to specific angle then the concerned motor will move accordingly. The first potentiometer is connected to a motion processing unit sensor type MPU6050 first then the reading of the sensor will be processed by the controller and then actuate the first motor.

The motors (motor2, motor3, motor4) depend on the first motor (motor1) motion

All the motors (motor1, motor2, motor3, motor4, motor5) are connected to a driver called L298.

Initialize MPU6050

Turn on the desired potentiometer

Read the movement of the arm through the MPU6050 **sensor**

The PID compares the two values process variable (MPU6050 value) and the desired value (pot value) to make the error

The error which comes from the comparison will be corrected through the PID parameters (P, I and D) to adapt the position of the arm

Is the desired part move as it should be?

NO



Yes

End

Figure (3.3) flow chart of the overall system

### 3.4 Circuit diagram of the overall system

The circuit diagram illustrates all components used and how connected.

The potentiometer RV1 called arm1 PD control is connected to the pin A0 of the controller and the output will actuate the motor called arm 1 PID , the potentiometer RV5 represent the mpu6050 sensor because the MPU6050 sensor in not defined in the proteus simulation so once the value of the RVI and the value of RV5 are equal to each other then that is mean the error is corrected and the motor arm 1 PID will stop in the desired position h, the potentiometer RV2 is connected to the pin A1 of the controller to actuate the motor called base, the potentiometer RV3 IS connected to pin A2 of the controller to actuate the motor called arm2,the potentiometer RV4 is connected to pin A3 of the controller to actuate the motor called arm3,the

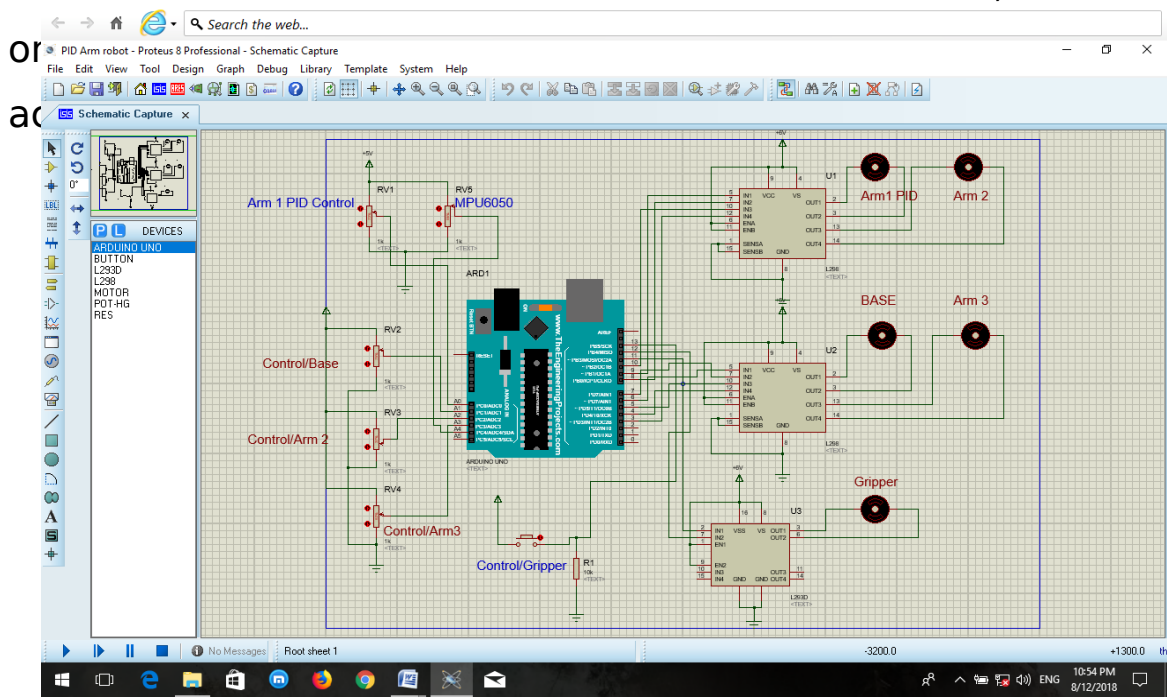


Figure 3.4 the simulation of the overall system

### **3.5 Circuit Components**

The components include sensor actuator drivers control unit and other components.

#### **3.5.1 Microcontroller (Arduino)**

IS Open Source electronic prototyping platform based on flexible easy to use hardware and software. And it provides easy way to interact with a computer physically the arduino Uno is suitable for the research because it is high performance, low power consumption.

#### **3.5.2 Motion Processing Unit (MPU6050)**

Motion Interface is becoming a “must-have” function being adopted by Smartphone and tablet manufacturers due to the enormous value it adds to the end user experience. In smart Phones, it finds use in applications such as gesture commands for applications and phone control, Enhanced gaming, augmented reality, panoramic photo capture and viewing, and pedestrian and vehicle navigation

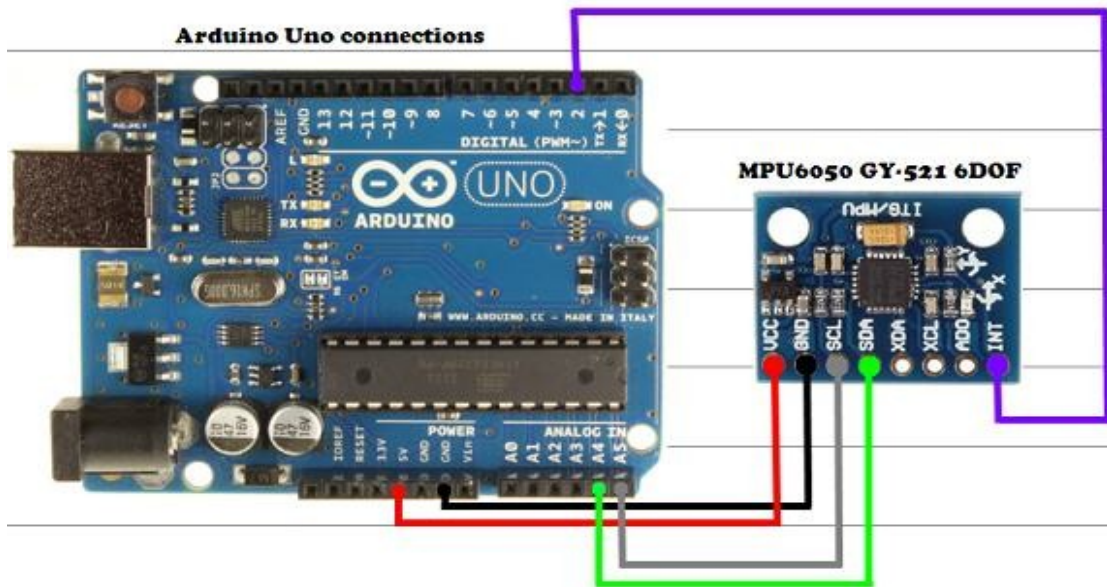


Figure (3.5) the arduino Uno connection with MPU6050

### 3.5.3 Potentiometer

Motion Potentiometers are a common analog sensor used to measure absolute angular rotation or linear motion (string pots) of a mechanism. A potentiometer is a three terminal device that uses a moving contact to from a variable resistor divider. When the outer contacts are connected to 5V and ground and the variable contact is connected to an analog input, the analog input will see an analog voltage that varies as the potentiometer is turned [15]

#### 3.5.3.1 Potentiometer in arduino

Connect the three wires from the potentiometer to the board. The first goes from one of the outer pins of the potentiometer to ground. The second goes from the other outer pin of the potentiometer to 5 volts. The third

goes from the middle pin of the potentiometer to the analog pin

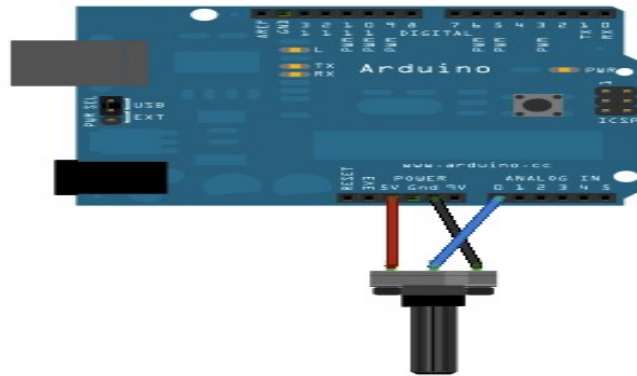


Figure (3.6) Connection of potentiometer to arduino

### **3.5.4 DC motor in the arduino**

PWM, or pulse width modulation is a technique which allows us to adjust the average value of the voltage that's going to the electronic device by turning on and off the power at a fast rate. The average voltage depends on the duty cycle, or the amount of time the signal is ON versus the amount of time the signal is OFF in a single period of time

So depending on the size of the motor, we can simply connect an Arduino PWM output to the base of transistor or the gate of a MOSFET and control the speed of the motor by controlling the PWM output. The low power Arduino PWM signal switches on and off the gate at the MOSFET through which the high power motor is driven



# **System Implementation and Testing**

## Chapter Four

### 4. System Implementation and Testing

#### 4.1 System Implementation

The picture below shows the overall components of the system (the motors of the robotic arm, controller, sensor MPU6050, driver L298 and potentiometers)

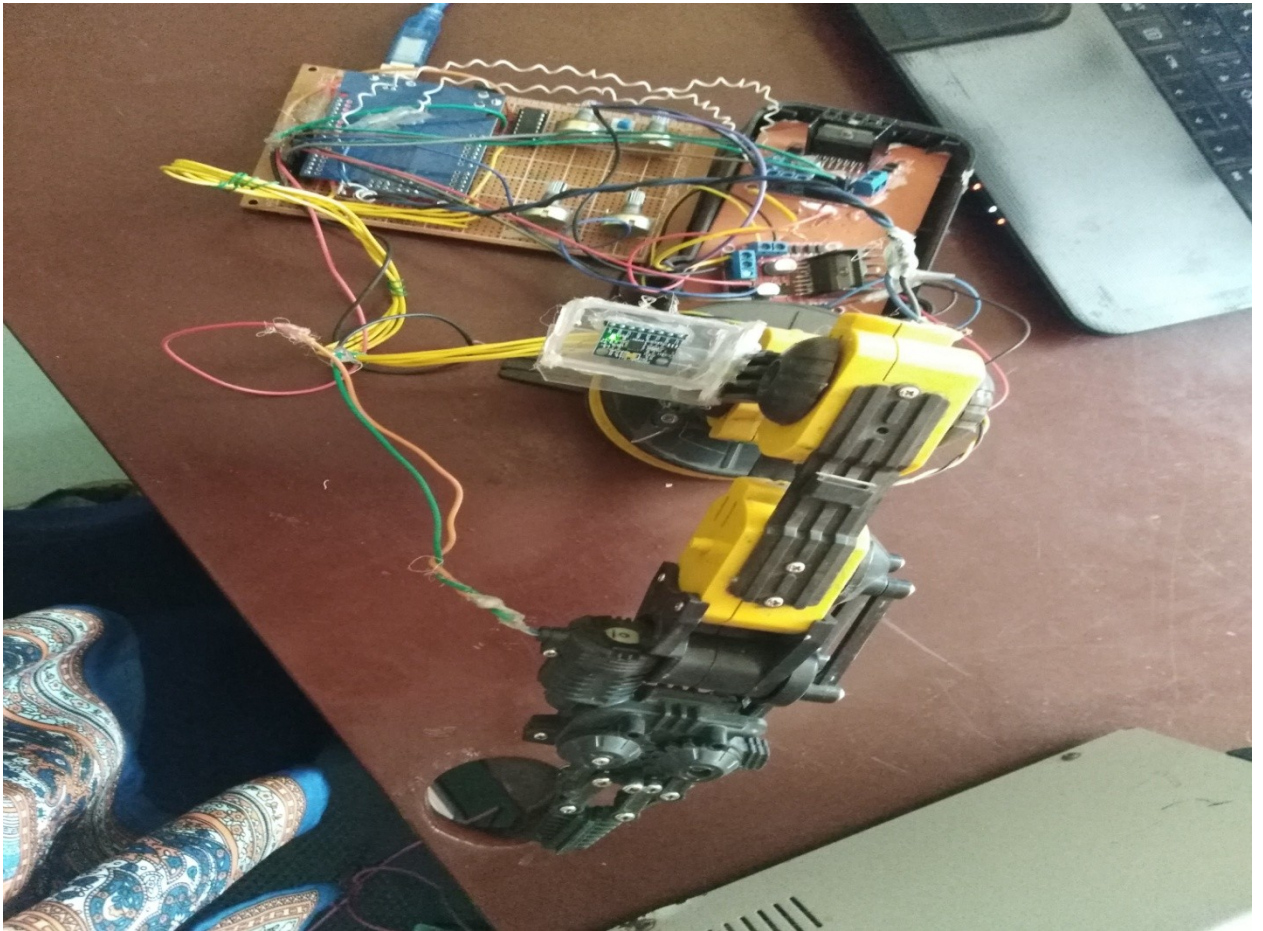






Figure 4.5 the COM4 shows effect of the PID and the readings of the sensor when the robot start to work

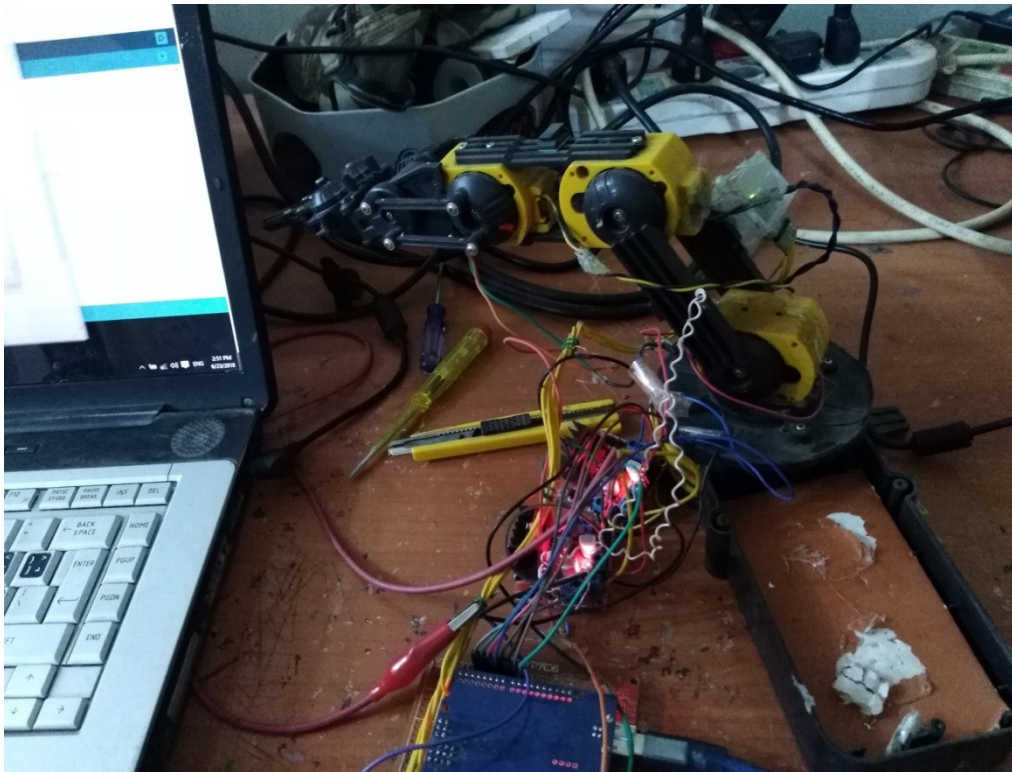
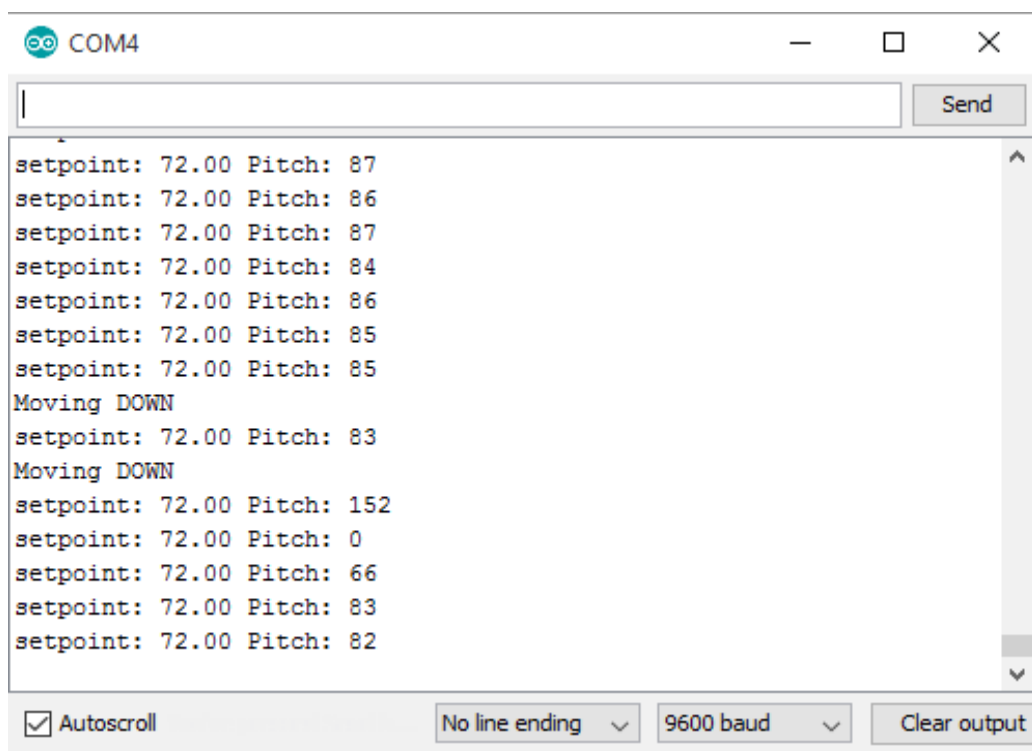
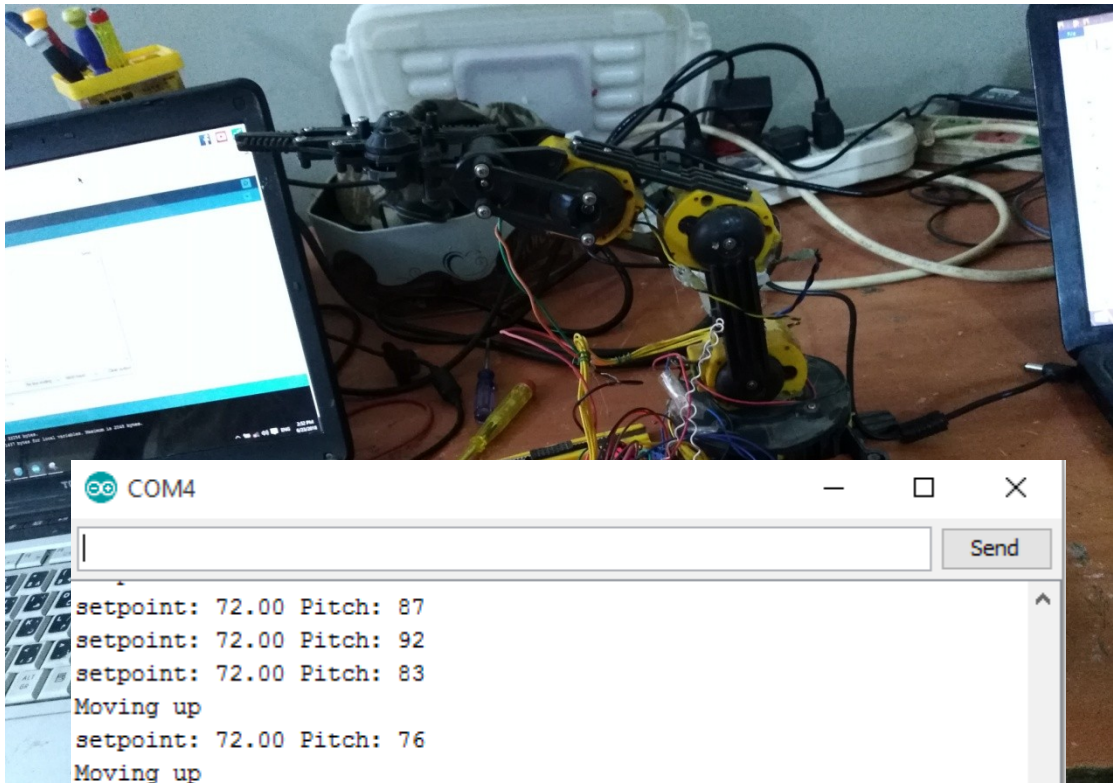


Figure 4.6 when the arm moves down





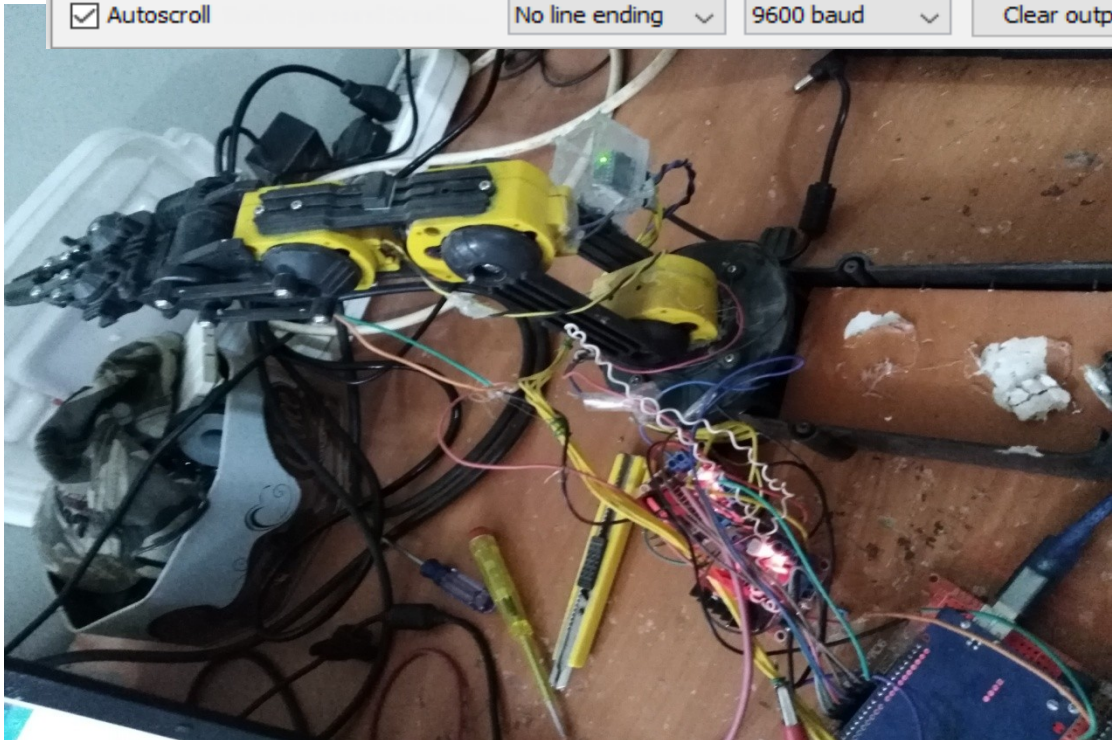
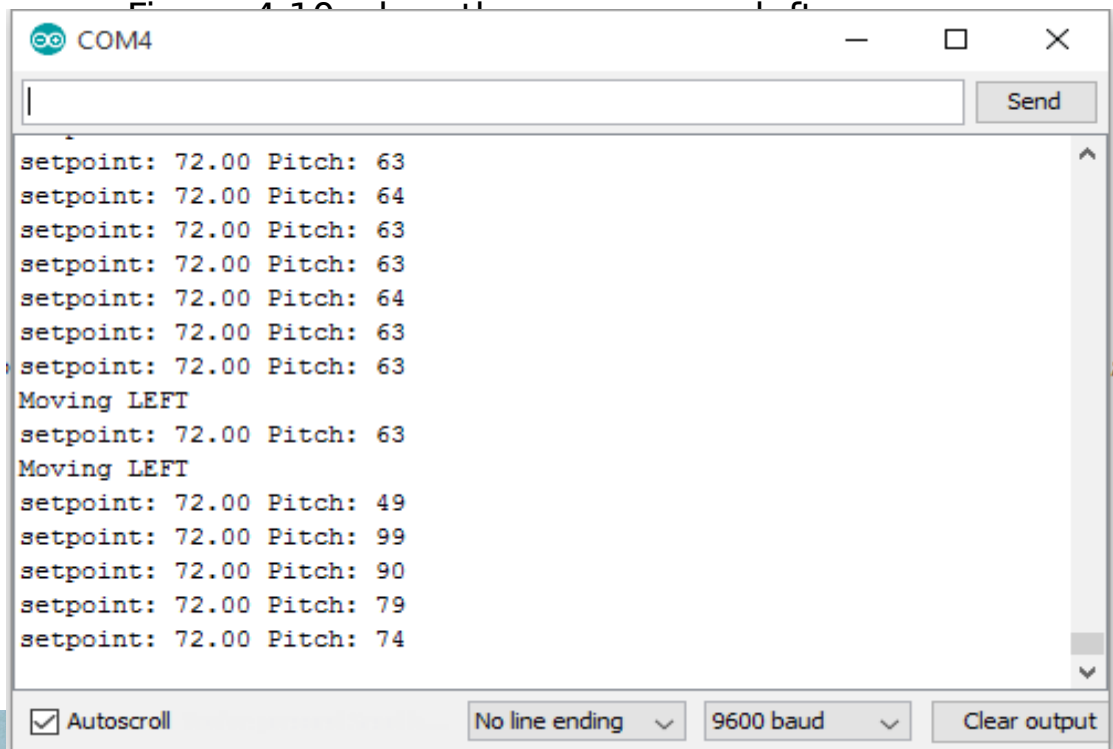


Figure 4.12 when the arm moves right

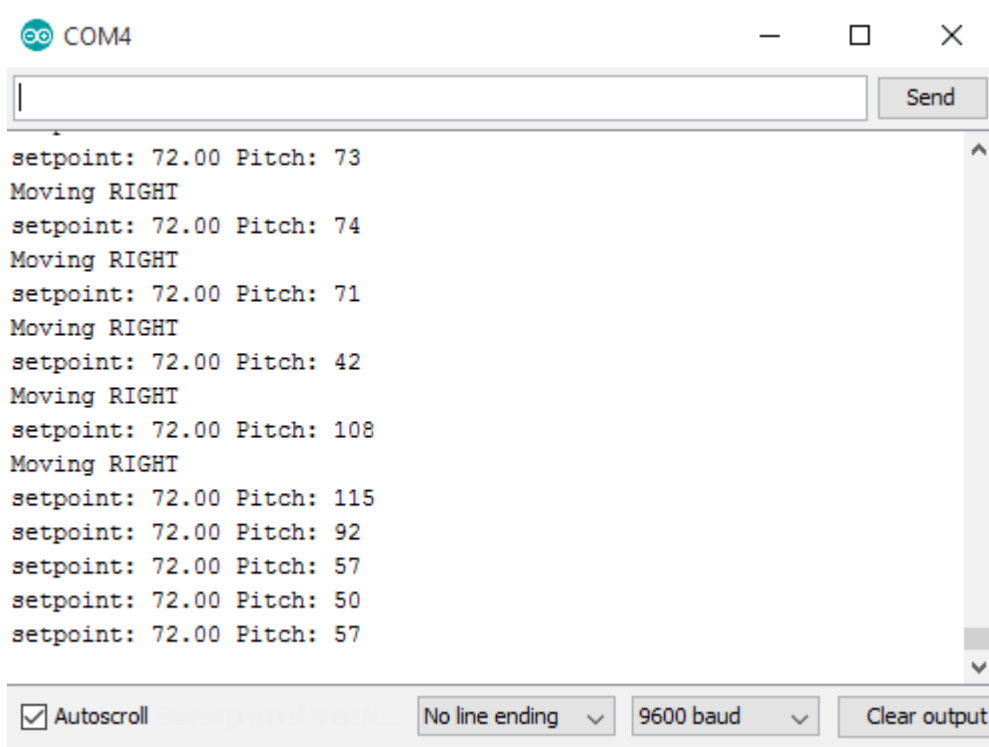


Figure 4.13 the COM4 shows effect of the PID and the readings of the sensor when the arm moves right



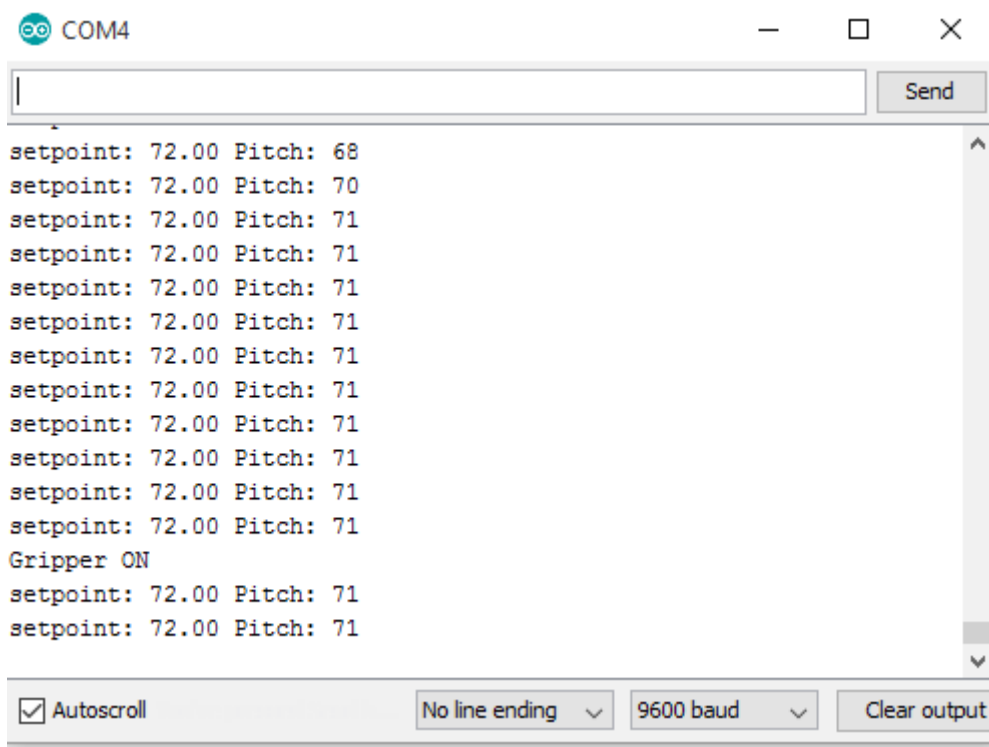


Figure 4.15 the COM4 shows effect of the PID and the readings of the sensor when the gripper is on

# Chapter Five

## Conclusion & Recommendations

## **Chapter Five**

### **Conclusion & Recommendations**

This chapter describes the conclusion of the research and the recommendations that we want the next researchers to do

#### **5.1 conclusions**

The robot is programmed by using arduino UNO adding to that the potentiometers are used to control the movement of the motors of the robotic arm and the sensor MPU 6050 is used also to read the angle of the arm

The both the reading of the angle which come from the sensor and the set point of the PID appear on the screen of the software of the arduino and it changes according to the movement of the arm

By using hardware and software components the angle of the arm is read by the sensor and the PID controller has adapted the position of the arm and then the accuracy of the robotic arm has improved

The robotic arm now is able to reach to the stability in his position as fast as possible

## **5.2 Recommendations**

There are many unsolved problems and fundamental challenges for robotics .At a very high level.

Manipulation and physical interaction with real world:

- a concerted modeling and control efforts is needed together with the development of good hardware to make arms and hands that can perform anything but the simplest of pick and place operation that are prevalent in industry .The pick and place robot is having the very vast area of application.
- Applying fuzzy logic controller to increase the accuracy of the robotic arm
- Increasing the number of the motors to increase the angle of freedom
- Connecting the MPU 6050 sensor very close because the length of the wire affect on the reading of the sensor
- Using more developed connection between MPU 6050 sensor and the system to avoid the noise.

## References

- [1] Robot Arm Dynamics and Control A. K. Bejczy rovot 1974
- [2] Robot-arm pick and place behavior programming system using visual perception - A.J. Sbnchez Univ. Politecnica de Valencia, Spain- September 2000
- [3] A Method for Estimating Servomotor Parameters of a Parallel Robot for Rapid Pick-and-Place Operations - Tian Huang, Jiangping Mei, Zhanxian Li, Xueman hao and Derek G. Chetwynd October 2004
- [4] Optimal Design of a 4-DOF Parallel Manipulator: From Academia to Industry - François Pierrot, Vincent Nabat and Olivier Company - In February 2009
- [5] Accelerometer-based control of an industrial robotic arm - Pedro Neto , J. Norberto Pires and A. Paulo Moreira 27 Sept.-2 Oct. 2009
- [6] RK Mittal and IJ Nagarath "Robotics and Control" BITS Pilani, 2003
- [7] "History of Industrial Robots" (PDF). Archived from the original (PDF) on 2012-12-24. Retrieved 2012-10-27.



- [8] Bennett, Stuart (1993). A history of control engineering, 1930-1955. IET.P. p. 48. ISBN 978-0-86341-299-8.
- [9] JUMO. "JUMO Campus - Control Engineering Basic principles and tips for practitioners". campus.jumo.de. Retrieved 2017-12-05.
- [10] Jinghua Zhong (Spring 2006). "PID Controller Tuning: A Short Tutorial" (PDF). Retrieved 2011-04-04.
- [11] Arduino FAQ - With David Cuartielles. Malmö University. April 5, 2013. Retrieved
- [12] [www.invensen.com](http://www.invensen.com) MPU-6000 and MPU-6050 product specification-Data sheet
- [13] <http://www.st.com> //Driver L298 specification-data sheet

Herman, Stephen. Industrial Motor Control