ANGULAR POSITION CONTROL OF A DC MOTOR USING PID CONTROLLER

Lab Project: Power Transmission

Submitted To

Sir Alamdar Hussain

Submitted By

Abubakar Ashfaq (FA19-BEE-002/ATK)

Suhaib Safdar (FA19-BEE-013/ATK)

Muhammad Mohsin Khan (FA19-BEE-034/ATK)

Madiha Zahra (FA19-BEE-037/ATK)

Comsats University Islamabad, Attock Campus

(2019-2023)

ACKNOWLEDGEMENT

First of All, we thank to ALLAH Almighty, the Most Merciful, who gave us strength to complete this project. Then to our parents who support and encourage us. We would like to thank Sir Alamdar Hussain for their helpful guidance in this project. Throughout the semester, they provide their very valuable comments based on our output and put us back onto the right track. Without their inspiriting comments, this project would become very impractical for general use. Last but not the least, we would like to thanks to all my friends and seniors who share their experiences with us.

Table of Contents

ABSTRACT

Nowadays, the DC motors are used in various applications such as defense, industries, robotics because of their simplicity, ease of application, reliability and cost effective. Generally, in a DC Motor, speed control can be achieved by varying the terminal voltage, but position control of the shaft cannot be achieved. The position control of a DC motor is crucial in applications for precision control system. The purpose of a motor position controller is to take a signal representing the required angle and to drive a motor at that position. Arduinos can provide easy control of a DC motor.

Typically, Arduinos are used to implement PID controllers. They receive the input pulses from the encoder, then feed them through a control algorithm to output the motor speed. Software based PID gives us more reliable control upon any parameter you want affecting your response, but hardware is typically harder to adjust.

The proposed design describes the designing and development of a Angular Position Control of DC Gear Motor using PID Controller. It is based on HC05 Bluetooth module and L298 Motor driver which is dedicated for motor control applications. The designed drive is a low-cost motor control drive used to control the angular position of DC gear motor and is targeted for industrial and electric appliances e.g., power winches on trucks, windshield wiper motors and power seat or power window motors. Jacks, cranes, lifts, clamping, robotics, conveyance, and mixing are just some of the applications gearmotors are used for in industry.

CHAPTER I

Introduction

1.1 Introduction:

There are many different DC motor types in the market and all with it good and bad attributes. One such bad attribute is the lag of efficiency. In order to overcome this problem a controller is introduced to the system. There are also many types of controllers used in the industry, one such controller is PID controller. PID controller or proportional–integral– derivative controller is a generic control loop feedback mechanism widely used in industrial control systems. A PID controller attempts to correct the error between a measured process variable and a desired set point by calculating and then outputting a corrective action that can adjust the process accordingly. So by integrating the PID controller to the DC motor we will be able to correct the error made by the DC motor and control the position of the motor to the desired speed or point. Electronic analog controllers can be made from a solid-state or tube amplifier, a capacitor and a resistance. Electronic analog PID control loops were often found within more complex electronic systems. However, nowadays, electronic controllers have largely been replaced by digital controllers implemented with microcontrollers or FPGAs. In this project, PID controller is chosen as the controller for the DC motor. This is because PID controller helps get the output, where we want it in a short time, with minimal overshoot and little error. We must be able to position the motor very precisely, thus the steady state error due to motor position should be zero. We will also want the steady state error due to a disturbance to be zero as well. The other performance requirement is that the motor must reach its final position very quickly. In this case, we want it to have a settling time of 40ms.We also want to have an overshoot smaller than 16 %.If we simulate the reference input by a unit step input, then the motor position output should have a step response with the above-mentioned requirements.

In this project, a PID controller design for a DC motor angular position control. DC motor angular position systems are usually controlled by proportional integral- derivative (PID) control algorithms with PID coefficients tuned for optimizing operation. The objective of a PID controller in a position control system is to maintain a position set point at a given value and be able to accept new set-point values dynamically. Modern position control environments require controllers that are able to cope with parameter variations and system uncertainties. To implement a PID controller the proportional gain KP, the integral gain KI and the derivative gain KD must be determined carefully. controlling the DC motor without using the PID controller will give some oscillation in the signal and because the system is nonlinear, controlling by function is the best way to control the nonlinear systems and PID controller is the best choice to achieve this task.

CHAPTER II

Circuit Components

2 Circuit Components:

Following components are required to make the

SL.NO.	NAME OF THE COMPONENT	SPECIFICATIONS	QUANTITY
1.	ARDUINO	NANO	1
2.	DC GEAR MOTOR WITH ENCODER	TG-47C-SG-100-E221	1
3.	BLUETOOTH MODULE	HC - 05	1
4.	MOTOR DRIVER IC	L289	1

2.1 Arduino Nano:

The Arduino Nano is Arduino's classic breadboard friendly designed board with the smallest dimensions. The Arduino Nano comes with pin headers that allow for an easy attachment onto a breadboard and features a Mini-B USB connector. The classic Nano is the oldest member of the Arduino Nano family boards.

2.1.1. Specifications:

1. Operating voltage: 5 volts
2. Input voltage: 6 to 20 volts
3. Digital I/O pins: 14 (6 optional PWM outputs)
4. Analog input pins: 8
5. DC per I/O pin: 40 mA
6. DC for 3.3 V pin: 50 mA
7. Flash memory: 32 KB, of which 0.5 KB is used by bootloader
8. SRAM: 2 KB
9. EEPROM: 1 KB
10. Clock speed: 16 MHz
11. Length: 45 mm
12. Width: 18 mm
13. Mass: 7 g

14. USB: Mini-USB Type-B [5]
15. ICSP Header: Yes
16. DC Power Jack: No

Figure 1: Arduino Nano

2.1.2 Pin Configuration:

There is total 14 digital Pins and 8 Analog pins on your Nano board. The digital pins can be used to interface sensors by using them as input pins or drive loads by using them as output pins. A simple function like **pinMode()** and **digitalWrite()** can be used to control their operation. The operating voltage is 0V and 5V for digital pins. The analog pins can measure analog voltage from 0V to 5V using any of the 8 Analog pins using a simple function like **analogRead().**

These pins apart from serving their purpose, can also be used for special purposes, which are discussed below:

- Serial Pins 0 (Rx) and 1 (Tx): Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.
- External Interrupt Pins 2 and 3: These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM Pins 3, 5, 6, 9 and 11: These pins provide an 8-bit PWM output by using **analogWrite()** function.
- SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK): These pins are used for SPI communication.

- In-built LED Pin 13: This pin is connected with a built-in LED. When pin 13 is HIGH – LED is on and when pin 13 is LOW, it is off.
- I2C A4 (SDA) and A5 (SCA): Used for IIC communication using Wire library.
- AREF: Used to provide reference voltage for analog inputs with analogReference() function.
- Reset Pin: Making this pin LOW, resets the microcontroller.

2.2 L298n Motor Driver IC:

L298N module is a high voltage, high current dual full-bridge motor driver module for controlling DC motor and stepper motor. It can control both the speed and rotation direction of two DC motors. This module consists of an L298 dual-channel H-Bridge motor driver IC. This module uses two techniques for the control speed and rotation direction of the DC motors. These are PWM – For controlling the speed and H-Bridge – For controlling rotation direction. These modules can control two DC motor or one stepper motor at the same time. This is a 15 pin IC. According to the L298 datasheet, its operating voltage is +5 to +46V, and the maximum current allowed to draw through each output 3A. This IC has two enable inputs, these are provided to enable or disable the device independently of the input signals. A black color heat sink is attached to the L298 IC of the module. A heat sink is a passive heat exchanger that transfers the heat generated by an electronic or a mechanical device to a fluid medium, often air or a liquid coolant.

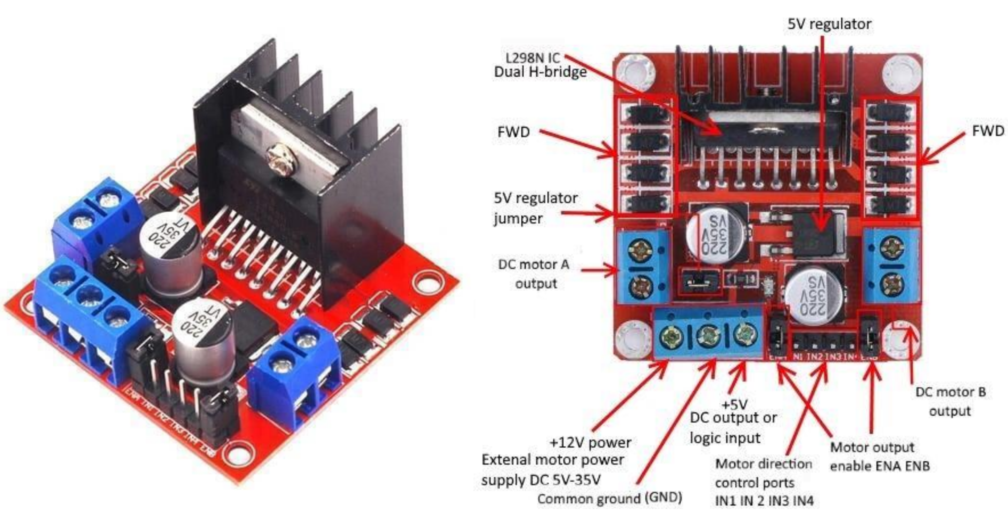
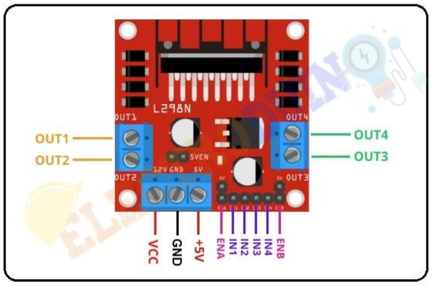


Figure 2: L298N MOTOR DRIVER IC

2.2.1 Pin Configuration:



Pins	Detail
POWER SUPPLY PINS	
VCC	VCC pin is used to supply power to the motor. Its input voltage is between 5 to 35V.
GND	GND is a ground pin. It needs to be connected to the power supply ground(negative).
+5V	+5V pin supplies power for the switching logic circuitry inside the L298N IC. If the 5V-EN jumper is in place, this pin acts as output and can be used to power up a microcontroller or other circuitry (sensor). If the 5V-EN jumper is removed, you need to connect it to the 5V power supply of the microcontroller.
Control Pins	
IN1	These pins are input pins of **Motor A**. These are used to control the rotating direction of Motor A. When one of them is HIGH and the other is LOW, Motor A will start rotating in a particular direction. If both the inputs are either HIGH or LOW the Motor A will stop.
IN2	
IN3	These pins are input pins of **Motor B**. These are used to control the rotating direction of Motor A. When one of them is HIGH and the other is LOW, Motor A will start rotating in a particular direction. If both the inputs are either HIGH or LOW the Motor A will stop.
IN4	

Speed Control Pins	
ENA	ENA pin is used to control the speed of **Motor A**. If a jumper is present on this pin, so the pin connected to +5 V and the motor will be enabled, then the Motor A rotates maximum speed. if we remove the jumper, we need to connect this pin to a PWM input of the microcontroller. In that way, we can control the speed of Motor A. If we connect this pin to Ground the Motor A will be disabled.
ENB	ENB pin is used to control the speed of **Motor B**. If a jumper is present on this pin, so the pin connected to +5 V and the motor will be enabled, then the Motor B rotates maximum speed. if we remove the jumper, we need to connect this pin to a PWM input of the microcontroller. In that way, we can control the speed of Motor B. If we connect this pin to Ground the Motor B will be disabled.
Output Pins	
OUT1 & **OUT2**	This terminal block will provide the output for **Motor A**.
OUT3 & **OUT4**	This terminal block will provide the output for **Motor B**.

2.2.2 Features and Specification:

The features and specifications of L298 IC include the following.

1. The operating voltage supply is up to 46 v.
2. Total DC is up to 4A.
3. The saturation voltage is less.
4. Protection from over temperature.
5. Operating voltage ranges from +5 to +46v
6. The maximum voltage supply voltage is 50v.
7. Maximum input & enable voltage is +7v.
8. TTL controlled inputs.

9. Storage temperature ranges from -40°c – 150°c.
10. Operating temperature ranges from -23°C to – 130°C
11. The maximum allowed the current flow to draw through every output is 3a.

2.2.3 Applications of L298 Motor Driver IC:

1. L298N motor driver IC is used in different fields like robotics, embedded, etc. We know that microcontrollers work with very little voltage and current, but motors use high voltage and current. So, for this reason, motor driver ICs are used to provide high voltage and current.
2. L298 motor driver is applicable where H- BRIDGE is used.
3. This motor driver is used in high power-based applications.
4. This IC is used where current control & PWM operable IC is required.
5. This IC is used where the control unit provides only TTL outputs.

2.3 Bluetooth HC 05 Module:

The HC-05 is a popular Bluetooth module which can add two-way (full-duplex) wireless functionality to your projects. HC-05 Bluetooth Module is an easy-to-use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Its communication is via serial communication which makes an easy way to interface with controller or PC.

2.3.1 Specification:

- Model: HC-05
- Input Voltage: DC 5V
- Communication Method: Serial Communication
- Master and slave mode can be switched

2.3.2 Pin Configuration:

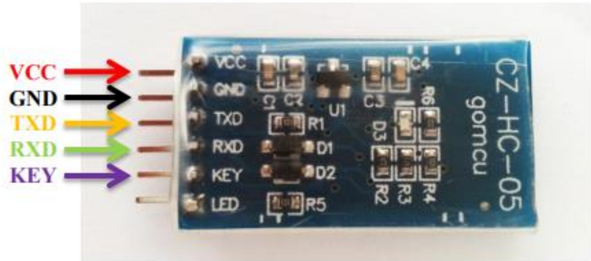


Figure 3: HC - 05 Bluetooth Module

The following is the pin configuration of Bluetooth Module HC - 05.

	Description	**Function**
1	Enable / Key (Mode switch input)	This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default, it is in Data mode
2	Vcc	Powers the module. Connect to +5V Supply voltage
3	Ground	Ground pin of module, connect to system ground.
4	TX – Transmitter (UART_TXD, Bluetooth serial signal sending PIN)	Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data. Connect with the MCU's (Microcontroller and etc) RXD PIN.
5	RX – Receiver (UART_RXD, Bluetooth serial signal receiving PIN)	Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth. Connect with the MCU's (Microcontroller and etc) TXD PIN.
6	State	The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly.
7	LED	Indicates the status of Module ➢ Blink once in 2 sec: Module has entered Command Mode ➢ Repeated Blinking: Waiting for connection in Data Mode ➢ Blink twice in 1 sec: Connection successful in Data Mode
8	Button	Used to control the Key/Enable pin to toggle between Data and command Mode

2.4 DC Gear Motor with Encoder:

The 12V DC gear-motor is a powerful motor to drive the position control system. It comes with the photoelectric encoder output and planetary gear ratio reduced by 80:1 gear. It can provide 120 rpm with the rated voltage of 12VDC. To read the count values from the encoder, the user would check the condition of channel A and B rotation applying by experiment. For the rotor shaft count per revolution values, it is very important to multiply the gear ratio by count values. Another direction is counterclockwise caused by channel B leading to channel A because the B interrupt seeing attached to A whenever it changes state LOW to HIGH or HIGH to LOW. Another direction is counterclockwise caused by channel B leading to channel A because the B interrupt seeing attached to A whenever it changes state LOW to HIGH or HIGH to LOW. The specification of the motor is

2.4.1 Specification:

- No load speed: 126rpm
- Workable Voltage:6-24V
- Rated voltage: 12V
- No load current：46mA
- Load torque: 0.85kg.cm
- Load current: 250mA
- Load output:1.25W
- Stall torque: 4.2kg.cm
- Reducer size: 21mm
- Operating Temperature Range: -40℃~120℃

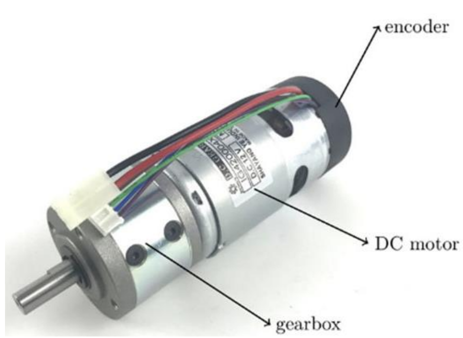
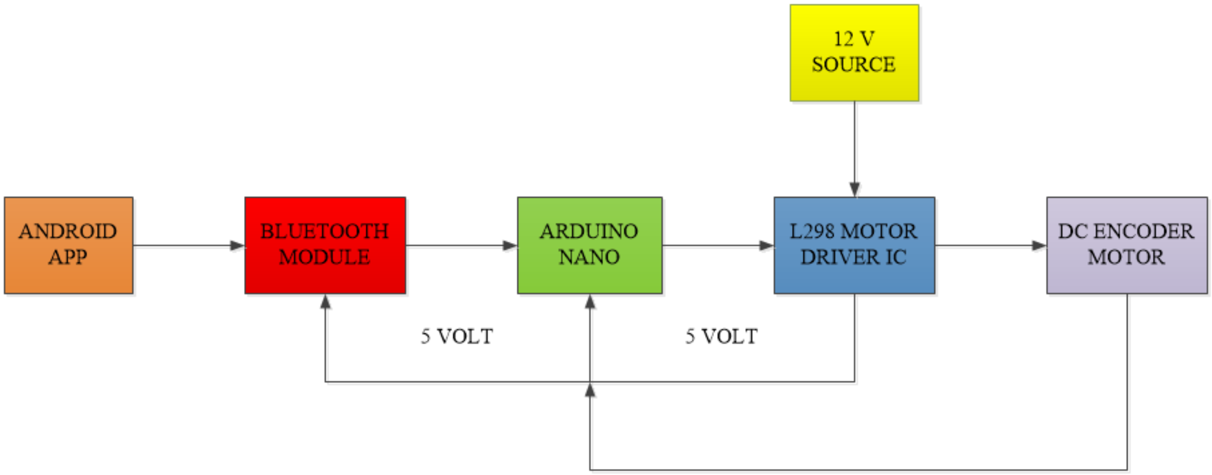


Figure 4: DC Encoder Motor

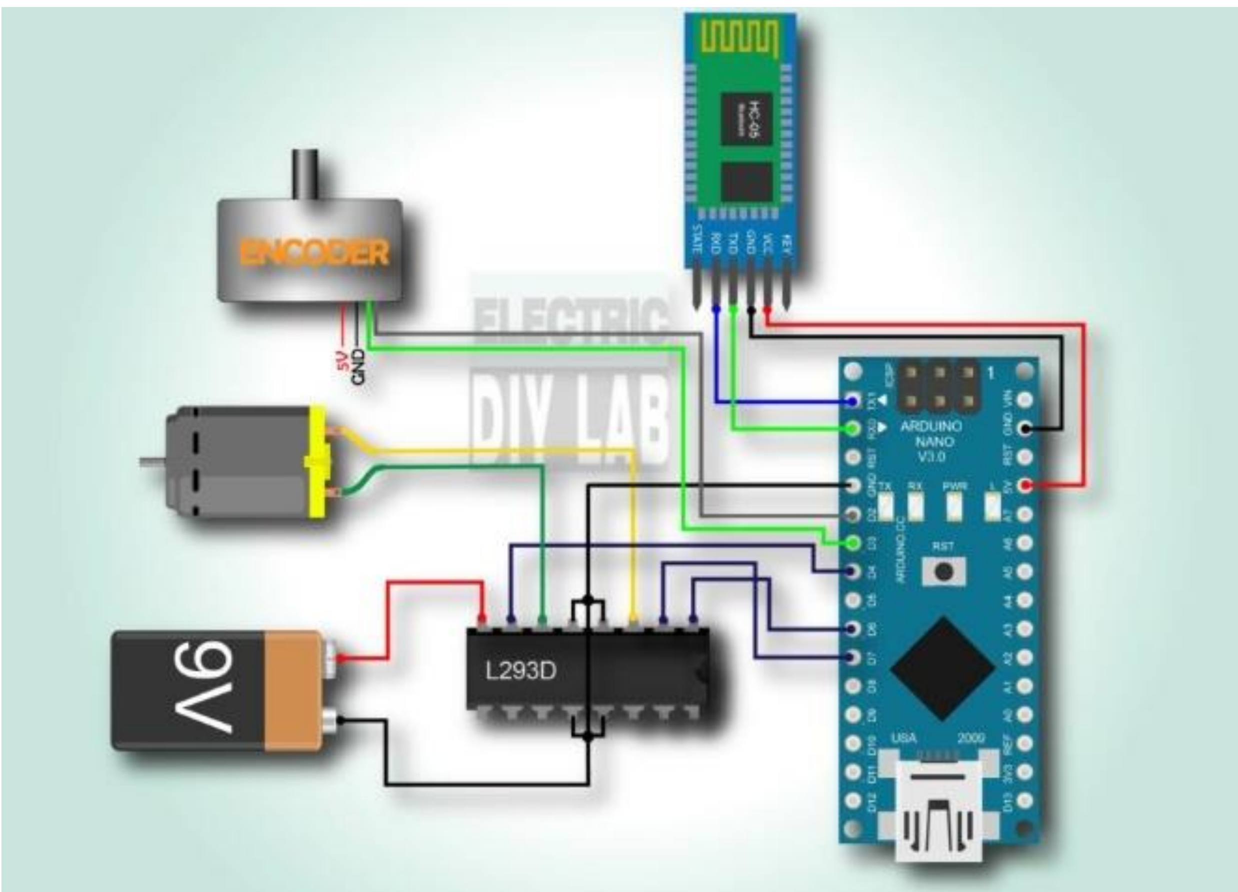
CHAPTER III

Circuit Diagram, its Working and Applications

3.1 Block Diagram:



3.2 Circuit Diagram:



3.3 Working:

The circuit controls the speed of the 12-V DC motor using the Android app on a smartphone. The app sends the commands to start or stop the motor and to change the Angular position(0-360 and 360-0) of the motor via the smartphone's Bluetooth.

These commands are received by the HC05 module, which passes them on to the Arduino NANO via the Tx and Rx pins. As per the commands sent over, Arduino will run or stop the DC motor and vary its Angular position/rotation from 0 to 360 and vice versa. Arduino generates a PWM signal on its pin to run or stop the motor or to vary the motor Angular position. To stop the motor, the pulse width on the pin D2 and D3 is 0 (0%). And to rotate the motor at 360-degree, pulse width on the pin D2 and D3 is (100%). So, as Arduino changes the pulse width on its pin, the motor Angular position changes from min to max or vice versa.

3.4 Code:

```
#include <SoftwareSerial.h>

SoftwareSerial myserial(4, 5); // RX, TX

#include <PID_v1.h>

#define MotEnable 11 //Motor Enamble pin Runs on PWM signal

#define MotFwd  10  // Motor Forward pin

#define MotRev  9 // Motor Reverse pin

String readStringk; //This while store the user input data

char c;

int UserInput = 0; // This while convert input string into integer

int encoderPin1 = 2; //Encoder Output 'A' must connected with intreput
pin of arduino.

int encoderPin2 = 3; //Encoder Otput 'B' must connected with intreput
pin of arduino.

volatile int lastEncoded = 0; // Here updated value of encoder store.
```

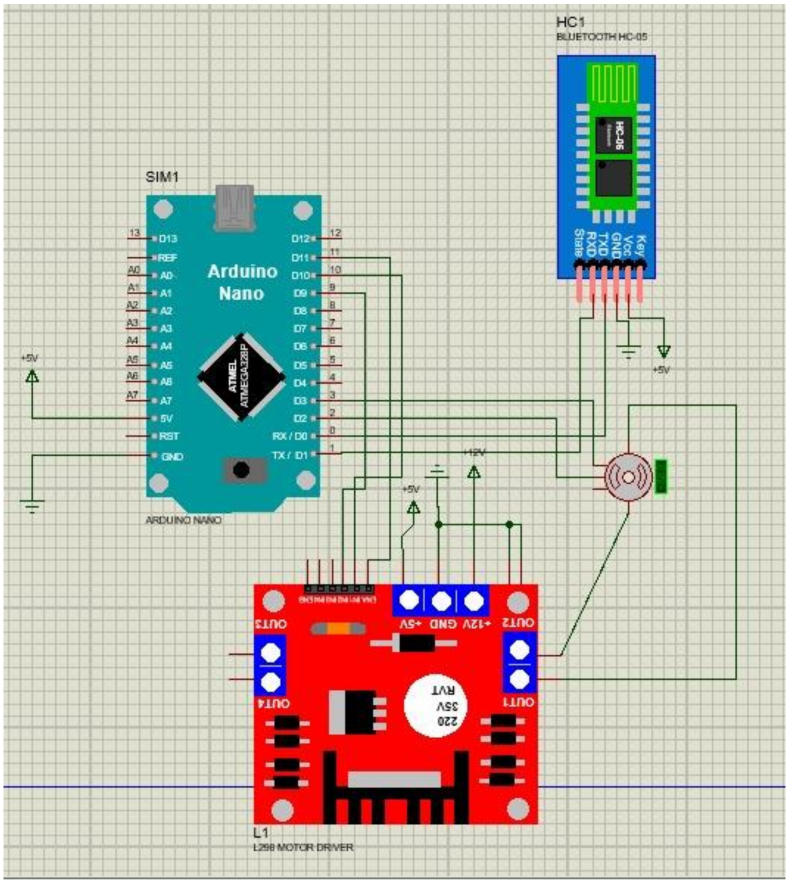
```
volatile long encoderValue = 0; // Raw encoder value

int PPR = 1600;  // Encoder Pulse per revolution.

int angle = 360; // Maximum degree of motion.

int REV = 0;          // Set point REQUIRED ENCODER VALUE

int lastMSB = 0;

int lastLSB = 0;

double kp = 5 , ki = 5 , kd = 0.01;              // modify for optimal
performance

double input = 0, output = 0, setpoint = 0;

PID myPID(&input, &output, &setpoint, kp, ki, kd, DIRECT);

void setup() {

  myserial.begin(9600);///bluetooth

  pinMode(MotEnable, OUTPUT);

  pinMode(MotFwd, OUTPUT);

  pinMode(MotRev, OUTPUT);

  Serial.begin(9600); //initialize serial comunication

   pinMode(encoderPin1, INPUT_PULLUP);

  pinMode(encoderPin2, INPUT_PULLUP);

  digitalWrite(encoderPin1, HIGH); //turn pullup resistor on

  digitalWrite(encoderPin2, HIGH); //turn pullup resistor on

  //call updateEncoder() when any high/low changed seen

  //on interrupt 0 (pin 2), or interrupt 1 (pin 3)

  attachInterrupt(0, updateEncoder, CHANGE);

  attachInterrupt(1, updateEncoder, CHANGE);
```

```
  TCCR1B = TCCR1B & 0b11111000 | 1;  // set 31KHz PWM to prevent motor noise

  myPID.SetMode(AUTOMATIC);   //set PID in Auto mode

  myPID.SetSampleTime(1);  // refresh rate of PID controller

  myPID.SetOutputLimits(-130, 130); // this is the MAX PWM value to move motor, here change in value reflect change in speed of motor.

}

void loop() {



  while (myserial.available()>0) { //Check if the serial data is available.

      delay(50);             // a small delay

     c = myserial.read();  // storing input data

    readStringk += c;   // accumulate each of the characters in readString



  }

  if (readStringk.length() >0) { //Verify that the variable contains information

     //printing the input data in integer form

    UserInput = readStringk.toInt();   // here input data is store in integer form

    Serial.println(UserInput);

}

REV = map (UserInput, 0,360, 0, 790); // mapping degree into pulse
```

```
                // printing REV value
setpoint = REV;                        //PID while work to achive this value consider as SET value
  input = encoderValue ;            // data from encoder consider as a Process value


  myPID.Compute();                  // calculate new output
  pwmOut(output);
}
void pwmOut(int out) {
  if (out > 0) {                               // if REV > encoderValue motor move in forward direction.
    analogWrite(MotEnable, out);          // Enabling motor enable pin to reach the desire angle
    forward();                                 // calling motor to move forward
  }
  else {
    analogWrite(MotEnable, abs(out));                    // if REV < encoderValue motor move in forward direction.
    reverse();                                 // calling motor to move reverse
  }
 readStringk=""; // Cleaning User input, ready for new Input
}
void updateEncoder(){
```

```
  int MSB = digitalRead(encoderPin1); //MSB = most significant bit

  int LSB = digitalRead(encoderPin2); //LSB = least significant bit

  int encoded = (MSB << 1) |LSB; //converting the 2 pin value to
single number

  int sum  = (lastEncoded << 2) | encoded; //adding it to the previous
encoded value

  if(sum == 0b1101 || sum == 0b0100 || sum == 0b0010 || sum == 0b1011)
encoderValue ++;

  if(sum == 0b1110 || sum == 0b0111 || sum == 0b0001 || sum == 0b1000)
encoderValue --;

  lastEncoded = encoded; //store this value for next time

}

void forward () {

  digitalWrite(MotFwd, HIGH);

 digitalWrite(MotRev, LOW);

}

void reverse () {

  digitalWrite(MotFwd, LOW);

 digitalWrite(MotRev, HIGH);

}

void finish () {

  digitalWrite(MotFwd, LOW);

 digitalWrite(MotRev, LOW);


}
```

3.5 Simulation Results:

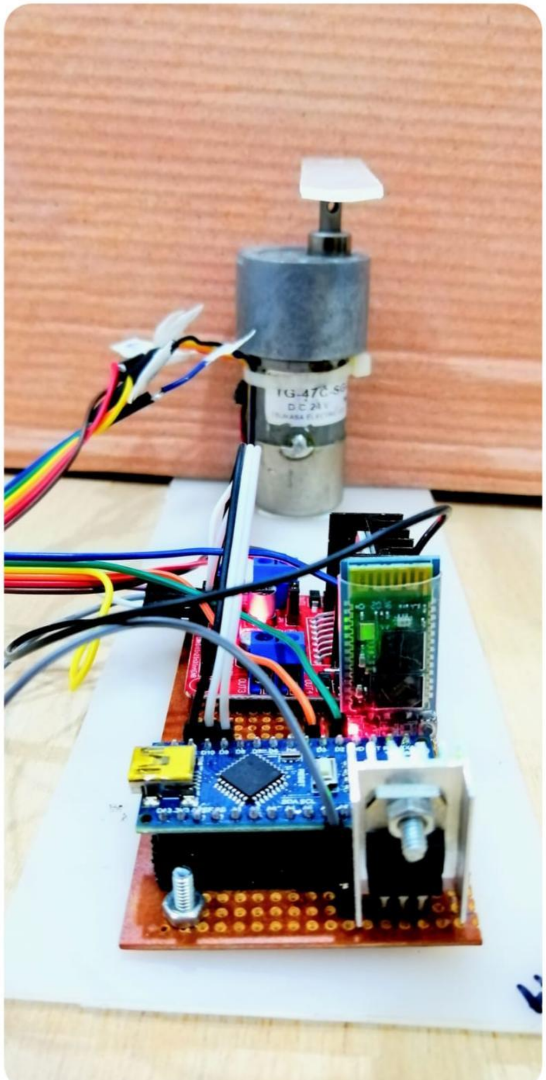


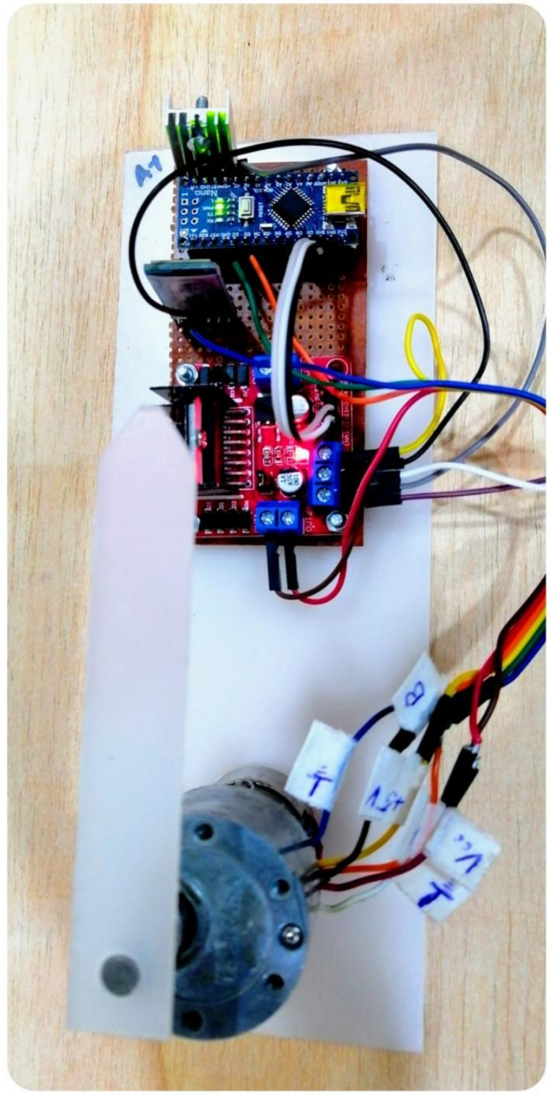
3.6 Applications:

DC motors are used in various applications such as:

1. DC motors are highly versatile and flexible in aspects of speed control.
2. High performance DC motor drives are popular in industrial applications for its enormous, good characteristics such as high starting, accelerating and retard torque, high response performance, rapid braking and easier to be linear control etc.
3. Robotics because of their simplicity, ease of application, reliability and cost effective.
4. It is also used for robotic manipulators, guided vehicles, steel rolling mills, cutting tool, overhead cranes, electrical traction and other application etc.

3.7 Hardware Results:





CHAPTER IV

Advantages, Conclusion & References

4.1 Advantages:

The proposed system has the following advantages:

1. By integrating the PID controller to the DC motor were able to correct the error made by the DC motor and control the angular position of the motor to the desired point.
2. The proposed system has an advantage in both noise reduction and oscillation reduction and the control system runs well and has a good system response.

4.2 Conclusion:

A PID based Angular Position Control of DC Gear Motor has been designed and implemented successfully. In which input signal is send from Android app directly to HC-05 Bluetooth module to the Arduino board via Rx and Tx pins for DC motor control system using L298 Motor driver. PID Gain tuning method is a good choice to reduce the oscillation according to the observed system response. Moreover, PID with friction compensation for control system operates well, especially it has no overshoot, small rise time, no oscillation and steady state error is nearly equal to zero to accomplish the designed criteria of the control system. Therefore, the experimental results are very optimal to get the desired angular position of DC motor.

4.3 References:

1. K. Sailan and K.D. Kuhnert, “DC Motor Angular Position Control using PID Controller for the purpose of controlling the Hydraulic Pump”, International Conference on Control, Engineering and Information, Vol. 1, No. 2, pp. 22,26, 2013.

2. H. Paul and S. Lasrado, “Precise Control of Angular Position of Geared DC Motors for Low Cost Applications”, International Journal of Innovative Research in Science,Engineering and Technology, Vol. 5, Special Issue 9, May 2016.

3. D. Xue, C. Zhao, Y.Q. Chen, Fractional Order PID Control of a DC Motor with Elastic Shaft: A Case Study, Proceedings of the 2006 American Control Conference, USA, 2006.

4. K.J. Astr ˚ om, and T. H ¨ agglund, ¨ Advanced PID control, ISA-The Instrumentation, Systems, and Automation Society, 2006.