Arduino-PID Based PMDC Motor Angular Position Control System

By: Ajaykumar K H [1], Akash R B [1], Koushik S [1], Sachin Pawar [1]
Under the guidance of: Mr. T B Dayananda, Associate professor [2]
[1]-Student,6th semester, Dept. of EEE, Dr. Ambedkar Institute of technology, Bengaluru-560056
[2]- Dept. of EEE, Dr. Ambedkar Institute of technology, Bengaluru-560056

Abstract: This paper helps you to get the precise angular position of shaft of PMDC geared motor using Arduino and PID control. Arduino UNO is used here to control the motor, L293D motor driver is used to drive the motor in required direction. HC-05 Bluetooth module is used to give the input through smart phone. To reduce the error, we use PID controller.

Index terms: PMDC geared Motor, L293D, Optical Encoder, and PID Controller.

Introduction

As technology is growing, industrial automation also updating with time. Many applications use motor which need both speed as well as position control. Small applications use PMDC motor because of its higher starting torque, quick starting, reversing, and they are easier and cheaper to control. By itself DC motor position can't be controlled like stepper or servo but adding an encoder can easy to control the DC motor.

In this paper, we use 5V geared PMDC motor coupled with incremental optical rotary encoder. Based on rating, the encoder sends pulses corresponding to position of motor shaft. L293D motor driver IC with PWM (Pulse width modulation) helps motor to turn in both directions depending upon the reference position. PID controller is used to calculate control signal and for smooth operation.

Block Diagram

Block diagram given below(fig.1) is shows the flow of procedure. We use the closed loop control system as shown, here the motor and motor and motor driver are acts as Plant, rotary encoder (sensor) is acts as Feedback, and Arduino is acts as Controller for the system.

Fig 1: Block diagram

Hardware Specifications

In this section, you will get to know about all the details about the hardware used in this position control system.

1) Arduino UNO:

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable.

Fig 2: Arduino UNO

It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo.

2) <u>L293D Motor Drive</u>:

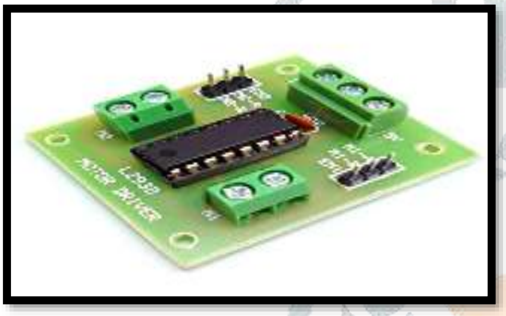

Fig 3: Motor driver module

L293D Motor driver module is a medium power motor driver perfect for driving DC motors and Stepper motors. It uses the popular L293 motor driver IC. It can Drive 2 DC motors with direction and speed control.

It can drive motors up to 12V with total DC current of up to 600mA. L293D is a 16 pin IC, with 8 pins each side, allowing us to control the motor. L293D consist of two H-bridge circuit. H-bridge is the simplest circuit for changing polarity across the load connect to it.

There are 2 OUTPUT pins, 2 INPUT pins, and 1 ENABLE pin for driving each motor.

3) <u>DC Geared Motor</u>:

The geared DC motor shown in Fig-4 is powered with voltage of 5V with current consumption approx. 180 mA. The motor has a gear (reduction ratio) 48:1, rotation speed approx.80 RPM and a torque of 0.5 kg-cm.

Fig 4: dc geared motor

These motors are light weight, high torque and low RPM. They can used in high torque application like hill climbing, load lifting etc., plus you can also mount on either side of the motor with its double-sided output shaft.

4) <u>Optical Rotary Encoder</u>:

Fig 5: Optical rotary encoder

We use this rotary encoder to know the actual position of the shaft of motor by coupling motor shaft with the encoder.

This is a 400 PPR 2 phase resolution optical encoder with quadrature outputs for increment counting. It will give 1600 transitions per rotation between outputs A and B, which means 800 transition per phase.

It is need to connect pull-up resisters connected to phase A and phase B because adding pull-up resister will help to avoid the effect of interface in output and get the precise logical output value from encoder at the same time, and also protects from damage due to direct V_{cc} supply short circuit.

5) <u>Bluetooth module</u>:

We use HC-05 Bluetooth Module to establish communication between controller and user. We use a Bluetooth application to communicate between Bluetooth device and module.

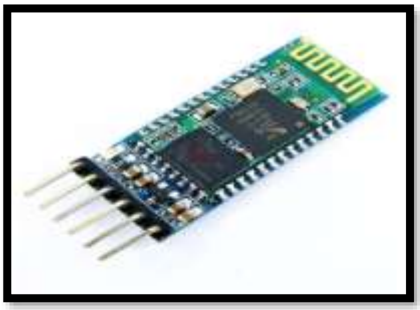

Fig 6: HC-05 Bluetooth module

HC-05 Bluetooth Module is an easy-to-use Bluetooth SPP (Serial Port Protocol) module, designated for transparent wireless serial connection setup. Its communication is via serial communication which makes an easy way to interface with controller or PC or a smartphone. It provides switching mode between master and slave mode which means it able to use neither receiving nor transmitting data.

PID CONTROL

PID stands for Proportional Derivative and Integral calculation.

PID is the part of programming, on which a complex calculation carried out by controller and manipulate the output as per the feedback from the process.

There are 4 key values of PID

1. PV (process value) or feedback
2. SV (set value)
3. Error
4. Output

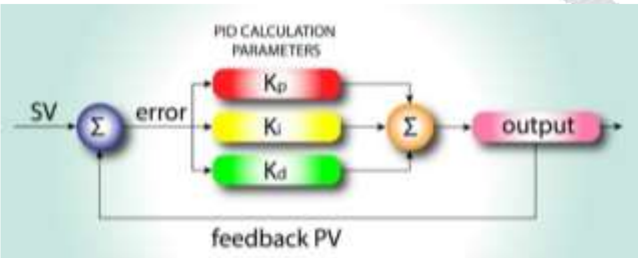

Fig 7: PID diagram

Basic PID working is as follows, PID try to keep PV as near as possible to SV.

Error = SV – PV

Error is difference between set value and process value and Output is directly proportional to Error hence in other words we can say greater the difference between SV and PV greater the Output, if Error is positive output is positive if error is negative output is negative.AS PV reach closer and closer to SV the Error get smaller and smaller, so in this way we get a clean and precise output.

Proportional controller: It mainly used to measure error to define how far away from the desired position (SV). So, we multiply proportional gain (K_p) to error to reduce it.

If K_p value is low, Overshoot is high, if it is high is good but, if it is too high spinning i.e., continuous rotation of shaft will take place. So medium value is recommended.

$$(K_P * e(t))$$

Derivative controller: Even if K_p is multiplied with error after some disturbance it will cause overshoot, to reduce that we add another constant derivative gain (K_d), it controls the rate of change of error.

To reduce the overshoot, multiply K_d with the rate of change of error and add with previous one.

$$(K_D * \frac{d}{dt}e(t))$$

$$(K_D * \frac{e - e\_prior}{iteration\_time})$$

If K_d value is too low overshoot is not minimized and oscillates, if it is high means took too much time to reach desired position (SV).

Integral controller: Integral controller is used to reduce Steady state error.

It sums up the previous errors to give indication if position is spending more time on that particular position.

So, we use Integral gain (K_i), multiplied with the integral of all the error and add with PD controller.

$$(K_I * \int_0^t e(t)d_t)$$

$$(K_I * (K_I\_prior + e * iteration\_time))$$

If K_i is large will cause overshoot, if it is too low it will take more time to eliminate the Steady state error.

Note: These three_constants K_p, K_d and K_i determines how strongly each term represented in the control loop, we can adjust them to tune response

CONNECTIONS

A double shaft PMDC geared motor is connected with shaft of encoder on one side and on other side a pointer (to indicate the position of shaft) is connected, this pointer points to the angle marked on protractor, rotary encoder is connected with Arduino on interrupt pins and DC motor drive by L293D motor driver IC, a HC-05 Bluetooth module is used to connect our system with android device.

Pin connections are made as given below

	Pin is connected to
Arduino pin	
pin 6	Motor driver enable
pin 4	Motor driver input 1
pin 7	Motor driver input 2
pin 2(interrupt 0)	Encoder output (A phase)
pin 3(interrupt 1)	Encoder output (B phase)
pin 0 (Rx)	Tx of HC-05
Pin 1 (Tx)	Rx of HC-05
5v	➢ HC-05 ➢ Optical rotary encoder
Gnd	➢ HC-05 ➢ Optical rotary encoder ➢ Motor driver ➢ 9V Battery negative terminal
Motor driver pin	
Output 1	Motor terminal 1
Output 2	Motor terminal 2
Power supply	9V Battery positive terminal

Table 1: pin connections of system

WORKING

We have connected controller, motor drive and motor in a closed loop (feedback loop). In feedback loop Motor and Motor drive is considered as Plant, the Optical rotary encoder which is used to measure the position is considered as sensor which sends actual position to calculate the Error, Arduino is acts as Controller.

In Order to control the position of the motor we need provide a target position (SV), for that we use a Bluetooth module and an android device. Then take a difference between target position (SV) and actual position (PV) results in Error

Error, e(t) = SV – PV

After calculation of Error, we use controller to compute a control signal u(t) which is send to plant. The control signal is configured so that it will attempt to reduce Error.

We use PID Algorithm to generate control signal u(t), PID control signal is constructed using a sum of 3 terms i.e., K_p, K_d and K_i terms

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}$$

Here, proportional term is the most important as it is directly responsible for reducing the error, derivative and integral terms are typically used to smooth out the control system response. Integral term accumulates the error over time and derivative computes how quickly the error is changing.

The control signal leads the motor to run mean while encoder sends real time position feedback to Arduino. As per predefined calculation, when encoder pulse matched with requirement it means shaft reaches the desire position Arduino stops the DC motor at that position.

For example, here we have used a optical encoder which give 1600 transition pulse for 360 degree revolution, so if we want to rotate pointer to 90 degree, so the 1600/360 x 90 = 400 pulse from encoder tells us it moves 90 degree.

CONCLUSION

In this paper, we send reference signal (SV) using an android mobile phone through Bluetooth module to Arduino board via serial communication for the PMDC motor control system. We use PID tuning algorithm, we can see no overshoot, no oscillations, and negligible Steady state error. So PID control method is the recommendable method to get the desired position of PMDC motor

ACKNOWLEDGEMENT

I would like to thank to all the teachers from Department of Electrical and Electronic Engineering, Dr. Ambedkar Institute of Technology, (Dr. AIT) for their suggestions and encouragement. I would like to thank my family and friends who have helped me during this research. And also, thanks for Mr. T B Dayananda, Associate professor, Dr. AIT and Dr. G V Jayaramaiah, Head of Department, Department of Electrical and Electronics, Dr. AIT.

REFERENCE

- Myo Maung Maung, Maung Maung Latt and Chaw Myat Nwe "DC Motor Angular Position Control using PID Controller with Friction Compensation", International Journal of Scientific and Research Publications, Volume 8, Issue 11, November 2018 ISSN 2250-3153.
- K. Sailan and K.D. Kuhnert, "DC Motor Anular 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.
- https://www.google.com/search
- https://m.youtube.com
- https://www.wikipedia.org
- https://www.arduino.cc/reference/en/