Controling DC Motor Position, Using PID Controller Made by PIC Microcontroller

Seyed Yahya Moradi
Department of Biomedical Engineering
Faculty of Engineering, University of Isfahan
Isfahan, IRAN
s.yahyamoradi@mehr.ui.ac.ir

Enayatallah Saeedi
Department of Electronic Engineering
Faculty of Electrical Engineering, Sahand University of Technology
Tabriz, IRAN
enayats1992@gmail.com

Abstract — **The Proportional Integral Derivative (PID) controller is the most common form of feedback used in the control systems. It can be used for various Industrial functions. One of the applications used here is to control the position of the DC motor. Controlling the position of a DC motor is very significant in that any small change can lead to instability of the closed loop system. Conventional PID controllers are used to control the DC motor for various industrial processes for many years due to their simplicity in operation. PID controllers require mathematical models to control the plant of different control processing applications. This paper seeks to focus on controlling the position of DC motors using PID controller made by PIC microcontroller. Consequently hardware has been designed and produced which in practice worked well.**

Keywords- PID Controller, PIC Microcontroller, Position Controlling.

INTRODUCTION

DC machine is a highly versatile and flexible machine. It can satisfy the demands of load requiring high starting, accelerating and retarding torques[1]. A DC machine is also easily adaptable for drives with range of speed control and fast reversal. They are widely used in industrial applications. The DC motors are used in rolling mills, in traction and in overhead cranes[2]. They are also used in many control applications as actuators and as speed or position sensors. In such applications, as that of position sensors and robotics, drives performance must precisely follow the desired performance. A number of control schemes such as proportional (P), proportional integral (PI), proportional derivative integral (PID), PWM, adaptive and fuzzy logic controller (FLCs) are used for control of speed and Position of DC motors. The proposed controller system uses the PID control scheme for position control of DC motor[1, 3, 4]. PID controllers due to their simplicity are widely used in industrial applications for speed and position control of DC motor for several years. Due to its simplicity, clear functionality, applicability and ease of use offered, PID controllers are used in more than 95% of the industrial control processing applications[4]. If the PID parameters are tuned properly they provide robust and reliable performance for most systems. In this paper, PID controller is used to control the position of DC Motor. As PID controllers require mathematical models of the system to be controlled, hence mathematical models of DC motor is derived[5, 6].

DC Motor's Model

DC Motor plays a crucial role in research, industry and laboratory experiments because of their low cost[6, 7]. The position of the motor can be controlled by three methods namely terminal voltage control method, armature rheostat control method and flux control method. Here in this paper terminal voltage control method is employed. A control system is an interconnection of components forming a system configuration that will provide a desired system response[2, 8].

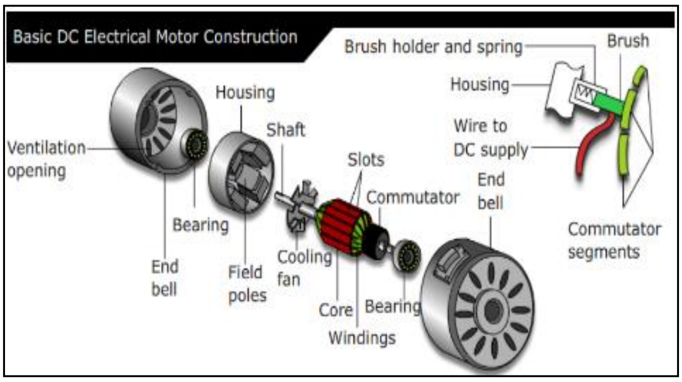


Figure 1: Components of DC Motor

In armature voltage control scheme for DC motors, voltage applied to armature is varied without varying the voltage applied to the field[1, 4]. Equivalent model of DC motor is shown in following figure (Figure 2).

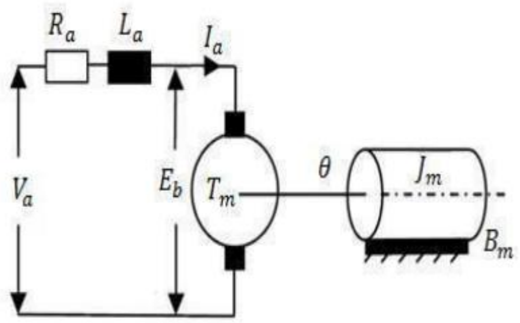


Figure 2: DC Motor Model

$$v_a = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_b(t) \qquad (1)$$

$$e_b(t) = K_b \, . w(t) \qquad (2)$$

$$T_m(t) = K_t \, . i_a(t) \qquad (3)$$

$$T_m(t) = J_m \frac{dw(t)}{dt} + B_m \, . w(t) \qquad (4)$$

Where [1, 9, 10]

v_a= Armature Voltage (V)

R_a= Armature Resistance (Ω)

L_a= Armature Inductance (H)

i_a= Armature Current (A)

e_b = Back EMF (V)

ω= Angular Speed (rad/s)

T_m= Motor Torque (Nm)

θ = Angular Position of Rotor Shaft (rad)

J_m= Inertia of Rotor (Kg-m2)

B_m= Viscous Friction Coefficient (Nms/rad)

K_t = Torque Constant (N-m/A)

K_b= Back EMF Constant (V/rad)

Let us combine the upper equations together:

$$v_a = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + K_b \, . w(t) \qquad (5)$$

$$K_t \, . i_a(t) = J_m \frac{dw(t)}{dt} + B_m \, . w(t) \qquad (6)$$

Laplace transforms and then the transfer function between shaft position and armature voltage at no-load is:

$$\frac{\theta(s)}{v_a(s)} = \frac{K_t}{[L_a . J_m . s^3 + (R_a . J_m + L_a . B_m)s^2 + (R_a . B_m + K_b . K_t)s]}$$

Equation 7: DC Motor Transfer Function

PID controllers

A proportional–integral–derivative controller (PID controller) is a generic control loop feedback mechanism (controller) widely used in industrial control systems. A PID controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs [1, 3]. A PID controller consists of a proportional element, an Integral element and a derivative element, all three connected in parallel. All of them take the error as input. Kp, Ki, Kd are the gains of P, I and D elements respectively. In Figure 1 a schematic of a system with a PID controller is shown [1, 10]. The PID controller compares the measured process value with a reference set point value. The difference or error, “e”, is then processed to calculate a new input process. This input will try to adjust the measured process value back to the desired set point. The alternative to a closed loop control scheme such as the PID controller is an open loop controller. Open loop control (no feedback) is in many cases not satisfactory, and is often impossible due to the system properties. By adding feedback from the system output, performance can be improved [2].

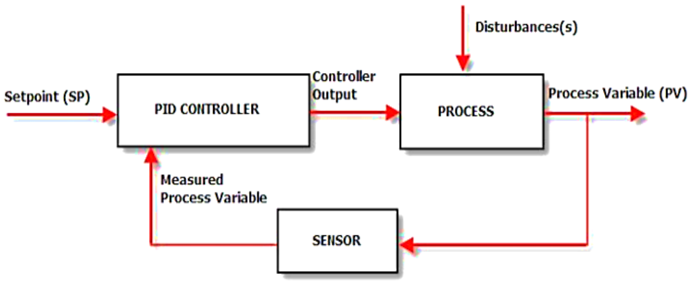


Figure 3: Closed Loop System with PID controller

Unlike a simple proportional control algorithm, the PID controller is able to manipulate the input processes based on the history and the rate of change of the signal. This gives a more accurate and stable control method [1, 5, 11]. The basic idea is that the controller reads the system state by a sensor. Then it subtracts the measurement from a desired reference to generate the error value. The error will be managed in three ways, to: 1. Handle the present, through the proportional term, 2. Recover from the past, using the integral term, 3. Anticipate the future, through the derivative term. In this paper we will use a basic open loop system and a closed loop system with PID system to control the DC motor [8].

PIC Microcontroller (PIC16F876A)

This powerful (200 nanosecond instruction execution) easy-to-program (only 35 single word instructions) CMOS FLASH-based 8-bit microcontroller packs powerful Microchips, PIC® architecture into an 28-pin package and is upwards compatible with the PIC16C5X, PIC12CXXX and PIC16C7X devices.[12]
Feature of the device:
256 bytes of EEPROM data memory Self programming
ICD (In Circuit Debugging function)
2 Comparators
5 channels of 10-bit Analog-to-Digital (A/D) converter
capture/compare/PWM functions
The synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI™) or the 2-wire Inter-Integrated Circuit (I²C™) bus
Universal Asynchronous Receiver Transmitter (USART)[13]

All of these features make it ideal for more advanced level A/D applications in automotive, industrial, appliances and consumer applications. Figure 4 shows the PIC16F876A.

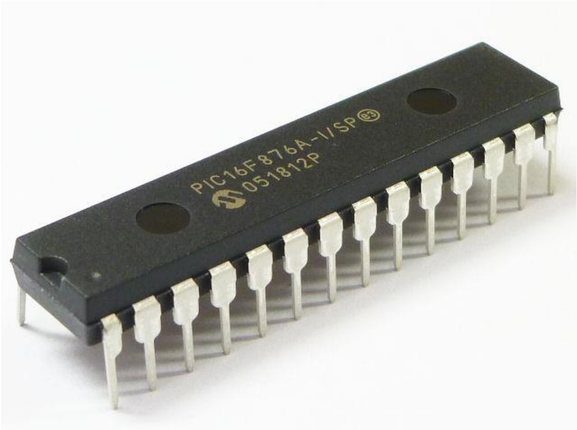


Figure 4: Closed Loop System with PID controller

EASE OF USE

A. HARDWARE

First, we describe the hardware of system and its features.

This project will require following hardware:

a. PIC16F876A
b. PR24 Printed Circuit Board (PCB)
c. LCD (2x16 character)
d. L293D Driver
e. DC Motor SPG20-50K
f. 22mm Precision Wire wound potentiometer
g. Related electronic components

Interface LCD (2x 16 Characters) with PIC16F877A

To use the LCD display, users have to solder 16 pin header pin to the LCD display. LCD used in this project is JHD162A. Figure 5 is a 2*16 character LCD and table 1 is all PIN and connectors.

Table 1: LCD PIN Fuction and Connection

Pin	Name	Pin Function	Connection
1	VSS	Ground	GND
2	VCC	Positive Supply	5V
3	VEE	Brightness	-
4	RS	Select Register	RB7
5	R/W	Select Read/Write	GND
6	E	Start Read/Write	RB6
7	DB0	Data Bus PIN	RB5
8	DB1	Data Bus PIN	RB4
9	DB2	Data Bus PIN	RB3
10	DB3	Data Bus PIN	RB2
11	DB4	Data Bus PIN	RB1
12	DB5	Data Bus PIN	RC5
13	DB6	Data Bus PIN	RC4
14	DB7	Data Bus PIN	RC3
15	LED+	Backlight +	5V
16	LED-	Backlight -	GND

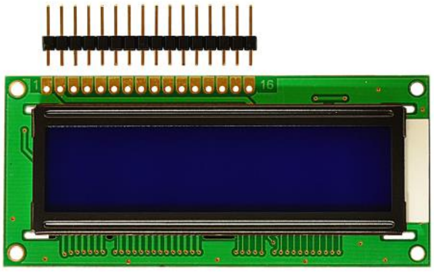


Figure 5: 2*16 character LCD

Figure 6 shows the connection of LCD display. LCD connection pin and function of each pin is shown:

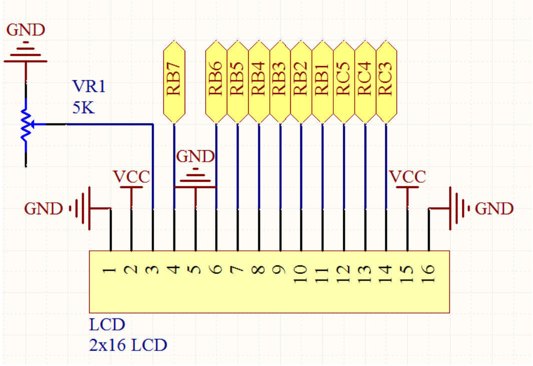


Figure 6: Connection of LCD

Power supply for the circuit

User can choose either AC to DC adaptors (not included in the DIY project set) or 12V batteries (not included in the DIY project set) to power up the circuit. Higher input voltage will produce more heat at LM7805 voltage regulator. Typical voltage is 12V. Anyhow, LM7805 will still generate some heat at 12V. There are two type of power connector for the circuit, DC plug (J1) and 2510-02 (Power Connector). Normally AC to DC adaptor can be plugged to J1 type connector.

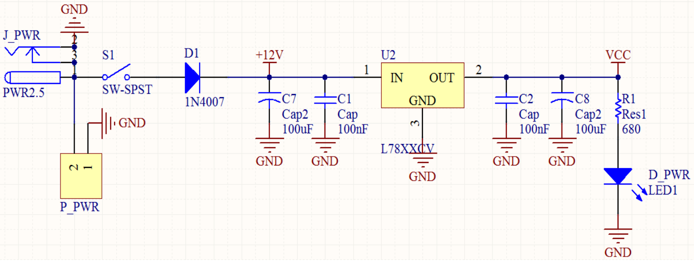


Figure 7: Power supply for the circuit

Refer to Figure 7 the D1 is used to protect the circuit from wrong polarity supply. C1 and C2 are used to stabilize the voltage at the input side of the LM7805 voltage regulator, while the C3 and C4 are used to stabilize the voltage at the output side of the LM7805 voltage supply. LED is to indicate the power status of the circuit. R1 is a resistor to protect LED from over current that will burn the LED.

Push Button as input for PIC microcontroller

One I/O pin (RB0) is needed for one push button as input (or interrupt) of PIC microcontroller. Another push button is used for RESET and connected to MCLR pin. The connections are shown in Figure 8. The pins should be pulling up to 5V using a resistor (with value range 1K-10K) and this configuration will result an active-low input. When the button is being pressed, reading of I/O pin will be in logic 0, while when the button is not pressed, reading of that I/O pin will be logic 1.

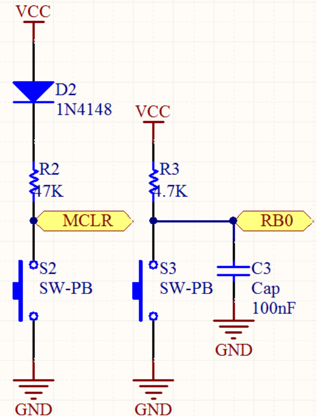


Figure 8: Push Button for the circuit

ICSP for programming PIC microcontroller

MCLR, RB6 and RB7 need to be connected to the USB in Circuit Programmer (UIC00A) to program the PIC microcontroller. The programmer (UIC00A) is not included in DIY project set since it can be used several time for different project set. User can also choose other type of PIC programmer to load the program.

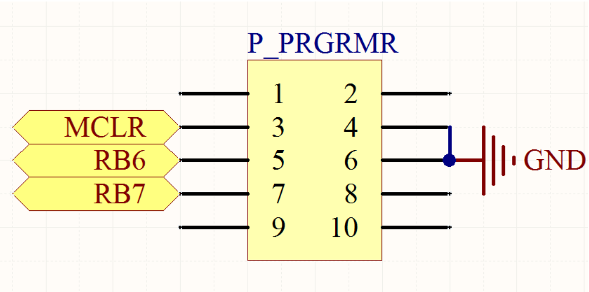


Figure 9: ICSP for programming circuit

Interface L293D with PIC16F876A

L293D is used as motor driver in this project, RC2 is connected to enable pin (EN1) to give the PWM signal to L293D. RC0 and RC1 are connected to IN2 and IN1 respectively to give control on motor direction. OUT1 and OUT2 are connected to DC motor. Figure 13 demonstrates the connection.

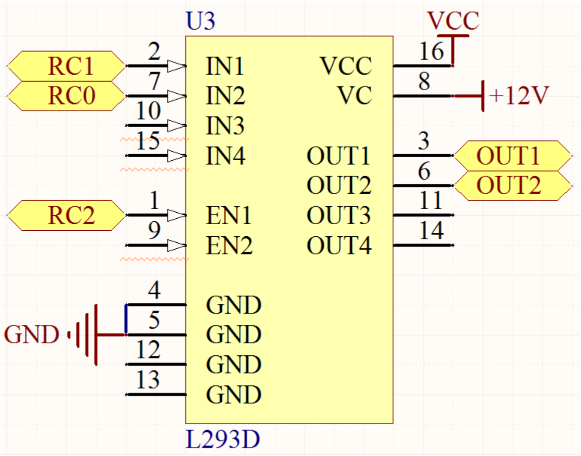


Figure 10: L293D circuit

Interface PID Gains and Angle Adjusters with PIC16F876A

Four analog I/O pins are needed for P, I, D and Angle value adjuster. RA0 is responsible for Angle value, RA2 is responsible for P value, RA3 is responsible for I value, and RA5 is responsible for D value. Figure 11 demonstrates the connection.

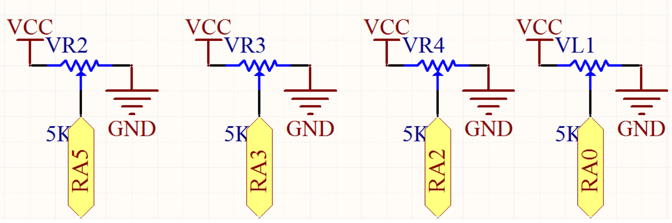


Figure 11: PID Gains and Angle Adjusters circuit

Interface 22mm Precision Wire wound Potentiometer with PIC16F876A

RC6 and RC7 which serve as TX and RX pins in PIC16F876A are used for UART interface. 5V and ground are connected to provide power to the module. User can use this connector for interface with other devices (For example, show the output of this system to the screen of computer). Figure 12 shows the connection.

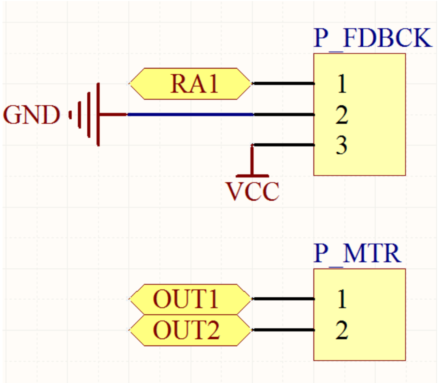


Figure 12: Potentiometer circuit

At the end the designed schematic and PCB are presented:

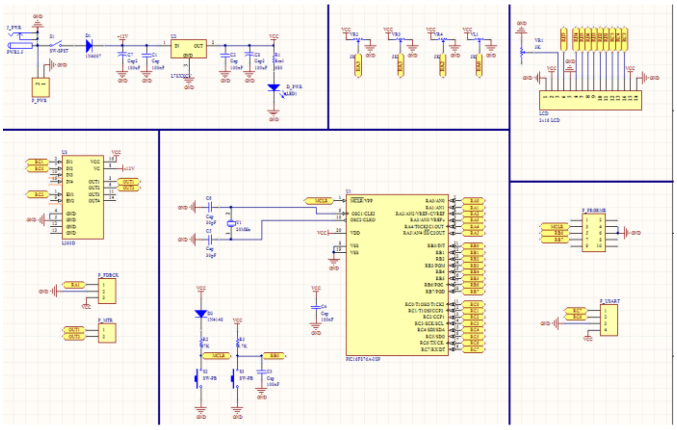


Figure 13: Schematic of the system

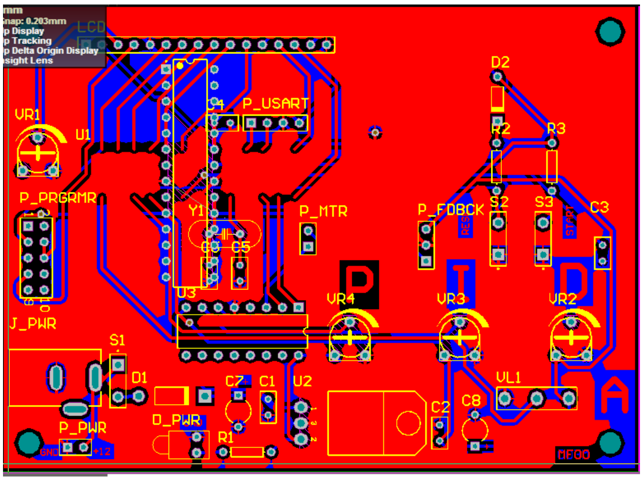


Figure 14: PCB of the system

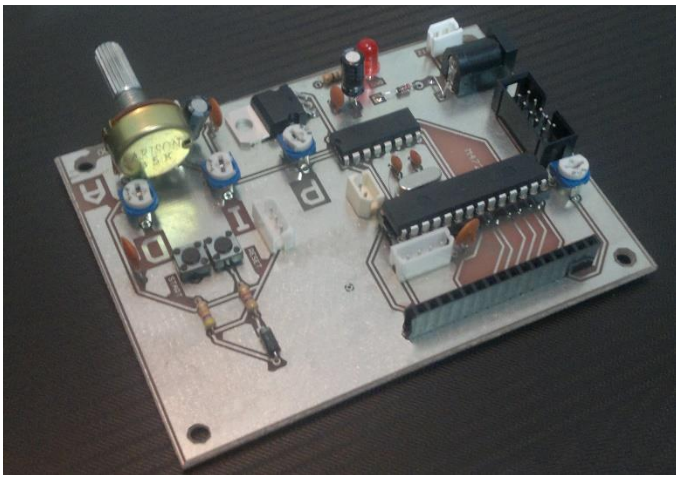
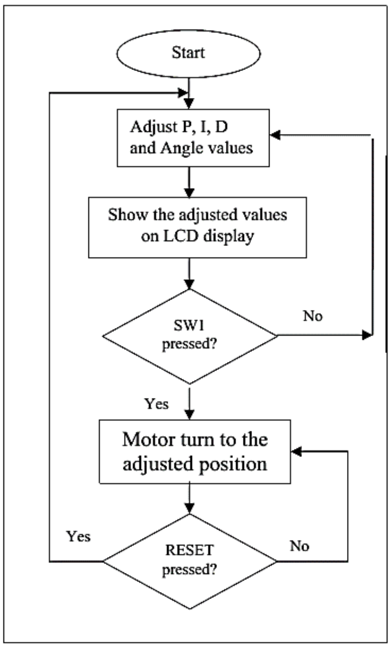


Figure 15: Designed Hardware

B. *CODE AND SOFT WARE*

The most code of the system has been written according the following flowchart:


Start
Adjust P, I, D and Angle values
Show the adjusted values on LCD display
SW1 pressed?
No
Yes
Motor turn to the adjusted position
Yes
RESET pressed?
No

Program (code) written to the system includes:
Include library and Configure PIC
Define
Declare and initialize global variable
Main Function
Functions for motor
Function for LCD
Function for analog input
Interrupt Function

For more elaboration two examples of Main Function code and Function for LCD code are presented:

```
//main function
 void main (void)
 {
 unsigned long rec_data=0;

 //set I/O input output
 TRISB =0b00000001;
 TRISC =0b00000000;
 TRISA =0b11111111;

 // configure lcd
 send_config (0b00000001);
 send_config (0b00000010);
 send_config (0b00000110);
 send_config (0b00001100);
 send_config (0b00111000);

 //LCD display
 lcd_clr();
 lcd_goto(0);
 send_string("ANGLE=");
 lcd_goto (11);
 send_string("P=");
 lcd_goto(20);
 send_string("I=");
 lcd_goto(26);
 send_string("D=");

 while(1)
 {
           rec_data=(unsigned long) aiRead(0)*3600/225;
              lcd_goto(6);
              send-dec(rec-data,4);
```

```
        kp=ai_read(2);

        lcd_goto (13);

        send_dec(kp,3);


        ki=ai_read(3);

        lcd_goto(22);

        send_dec(ki,3);

        kd=ai_read(4);

        lcd_goto(28);

        send_dec(kd,3);


    while (!SW1);
```

RESULTS

The objective of this experiment is to design a closed-loop control system that regulates the position of the DC motor. The mathematical model of a DC motor is reviewed and its physical parameters are identified. Once the model is verified, it is used to design a proportional-integral-derivative (PID) controller.

In this paper PID controller is designed for position control of DC motor. According to our analysis it is clear that PID is a simple controller based on the mathematical model of the system. It successfully overcomes the drawback of proportional-derivative (PD) controller of steady-state error as steady-state error is zero in PID controllers. However while reducing steady-state error to zero, an overshoot is observed. Overshoot can be reduced by increasing the derivative gain but the rise time also increases as a consequence. Hence, there is a compromise between overshoot and the speed of response i.e. rise time which means we have to sacrifice one for improving another. Overall, PID controller gives best speed response of all the linear controllers of its class.

Typical applications of this hardware:

- Compressors and fans in refrigerators
- Fans in cooker hoods
- Drums and pumps in washing machines
- Robotic Arms
- Surgery Robots
- Fan controller
- Medical Application (Electrocauter, ...)

REFERENCES

1. Xue, D., C. Zhao, and Y. Chen. *Fractional order PID control of a DC-motor with elastic shaft: a case study*. in *Proceedings of American control conference*. 2006.
2. Saranya, M. and D. Pamela, *A real time IMC tuned PID controller for DC motor.* International Journal of Recent Technology and Engineering, 2012. **1**(1): p. 65-70.
3. Kaya, I., *IMC based automatic tuning method for PID controllers in a Smith predictor configuration.* Computers & chemical engineering, 2004. **28**(3): p. 281-290.
4. Inaba, H., et al., *Control apparatus for DC motor*. 1978, Google Patents.
5. Leonhard, W., *Control of a Separately Excited DC Machine.* Control of Electrical Drives, 2001: p. 77-96.
6. Yeung, K. and J. Huang, *Development of a remote-access laboratory: a dc motor control experiment.* Computers in Industry, 2003. **52**(3): p. 305-311.
7. Naveenkumar, R. and D.P. Krishna, *Low Cost Data Acquisition and Control using Arduino Prototyping Platform and LabVIEW.* International Journal of Science and Research, 2013. **2**(2): p. 366-369.
8. Rubaai, A. and R. Kotaru, *Online identification and control of a DC motor using learning adaptation of neural networks.* Industry Applications, IEEE Transactions on, 2000. **36**(3): p. 935-942.
9. Zouari, F., K.B. Saad, and M. Benrejeb, *Adaptive Internal Model Control of a DC Motor Drive System Using Dynamic Neural Network.* Journal of Software Engineering and Applications, 2012. **5**(03): p. 168.
10. Rao, A.P.C., Y. Obulesu, and C.S. Babu, *Robust Internal Model Control Strategy based PID Controller for BLDCM.* International Journal of Engineering Science and Technology, 2010. **2**(11): p. 6801-6811.
11. Petráš, I., *Fractional-order feedback control of a DC motor.* Journal of Electrical Engineering, 2009. **60**(3): p. 117-128.
12. Mazidi, M.A., et al., *Pic microcontroller and embedded systems*. 2005: Prentice-Hall, Inc.
13. Peatman, J.B., *Design with PIC microcontrollers*. 1997: Simon & Schuster Trade.