

IMPLEMENTATION OF PID ON PIC24F SERIES MICROCONTROLLER FOR SPEED CONTROL OF A DC MOTOR USING MPLAB AND PROTEUS

Sohaib Aslam¹, Sundas Hannan¹, Umar Sajjad², Waheed Zafar²

¹ Superior University, 15-KM Raiwind Road, Lahore, Pakistan, e-mail: sohaib_engg2000@yahoo.com, engg.sundas@live.com

² Superior University, Lahore, Pakistan, e-mail: engr.umarsajjad@gmail.com, waheed.zafar@ymail.com

Received: 2016.03.26
Accepted: 2016.07.05
Published: 2016.09.01

ABSTRACT

Speed control of DC motor is very critical in most of the industrial systems where accuracy and protection are of essence. This paper presents the simulations of Proportional Integral Derivative Controller (PID) on a 16-bit PIC 24F series microcontroller for speed control of a DC motor in the presence of load torque. The PID gains have been tuned by Linear Quadratic Regulator (LQR) technique and then it is implemented on microcontroller using MPLAB and finally simulated for speed control of DC motor in Proteus Virtual System Modeling (VSM) software. Proteus has built in feature to add load torque to DC motor so simulation results have been presented in three cases speed of DC motor is controlled without load torque, with 25% load torque and with 50% load torque. In all three cases PID effectively controls the speed of DC motor with minimum steady state error.

Keywords: proportional integral derivative controller, pulse width modulation, linear quadratic regulator, peripheral interface controller, MPLAB, proteus, duty cycle.

INTRODUCTION

DC motor has been playing an important role as a drive configuration in many applications for a wide range of torques and speeds. The simplicity, accuracy to control and high performance of DC motors result in continuous increase in demand of DC motor speed control based applications. It has number of domestic and industrial applications few of them are mixer, treadmill, hair dryer, zero machine, elevator and traction [1]. The purpose of speed control of a system is to regulate the speed under dynamic environment. DC motor is a non-linear device so its speed gets varied with change in load torque or because of added electromagnetic interference. Requirement of accurate DC motor control in numerous applications result in development of several control techniques over a last few decades [2]. PID is one of the most famous and yet the simplest control technique to implement and it gives robust and efficient performance for most real world control

problems [3]. PID smartly controls the transient and steady state response of a closed loop system but the effectiveness of PID depends on how well the PID controller gains K_p , K_i and K_d have been tuned. For many years Ziegler-Nichols frequency response method has been used to tune gains but it requires long time and huge effort. In spite of an ordinary architecture and robustness of PID control technique, tuning of gains is still a complex problem, so to improve the performance of the traditional PID controller several tuning methodologies have been documented in literature [4]. Evolution of optimal control techniques have developed popular LQRs, so by employing the Lyapunov's method the optimal LQR problem minimizes to algebraic riccati equation which is further used to find optimized gains [5]. In recent years microcontroller based drive systems are getting immense attraction for implementing cheaper and robust controllers due to its integrated peripherals like Analog to Digital Converters (ADC) and Pulse Width Modulation (PWM) gen-

erator etc., less need of components, improved reliability, fast speed operation, more adaptability to latest control technique and increased flexibility of design [6]. PIC24FJ128GA010 is a 16-bit with low power consumption microcontroller. It has high performance capabilities in comparison to traditional 8-bit microcontrollers and it is not explored to the best potential for its hands on experience in educational and industrial sectors [7]. In this research work tuned PID has been implemented on PIC24F series microcontroller for speed control of a DC motor in the presence of load torque.

TRANSFER FUNCTION MODEL OF DC MOTOR

There are four types of DC motor. They are Shunt, Series, Permanent Magnet and Compound DC motor. In this research work Permanent Magnet DC motor (PMDC) is used. DC motor model is shown in Figure 1. V_m is the input voltage, L_m is the armature inductance, R_m is the armature resistance, V_b is the back emf generated by the motor. Moment of inertia, speed of motor, motor torque and load torque are J_m , ω_m , T_m and T_L respectively

[8]. The load torque is in opposite direction to motor torque as shown by the opposite arrows direction in Figure 1. Effective tuning of PID gains using LQR is based on the accuracy with which the system is modelled. The transfer function of DC motor is obtained by using the internal parameters of DC motor from Table 1 [9].

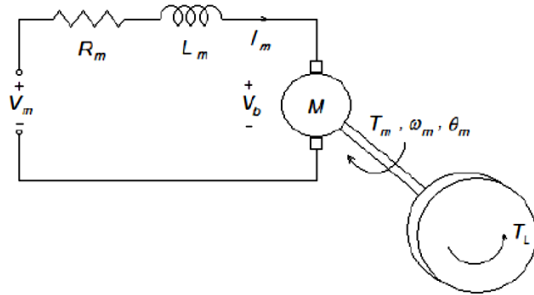


Fig. 1. DC motor Model

Table 1. Parameters of DC motor

No.	Parameter	Value
1	Moment of inertia	$J_m = 0.00025 \text{ Nm/rad/s}^2$
2	Damping factor	$B_m = 0.0001 \text{ N}_m/\text{rad/s}$
3	Armature resistance	$R_m = 0.5 \Omega$
4	Armature inductance	$L_m = 1.5 \text{ mH}$
5	Torque constant	$K_m = 0.05 \text{ Nm/A}$

Apply KVL to the loop in above figure to get electrical equation of the system:

$$V_m = R_m i_m + L_m \frac{di(t)}{dt} + k_m \omega_m \tag{1}$$

Motor is an electromechanical system so it is also represented by an equation of motion as shown in Eq. (2):

$$J_m \frac{d\omega_m}{dt} = T_m - T_L - B_m \omega_m \tag{2}$$

For simplicity here $T_L = 0$ is assumed, while $T_m = K_m I_m$. K_m = motor torque constant, B_m is damping friction. After applying Laplace transform to Eq. (1) and (2) and simultaneously solve them to get the transfer function model as shown in Eq. (3):

$$\frac{\omega_m}{V_m} = \frac{k_m}{[(L_m J_m) s^2 + (R_m J_m + B_m L_m) s + (k_m^2 + R_m B_m)]} \tag{3}$$

Substituting DC motor parameters in Eq.(3), the transfer function is:

$$\frac{19.607}{[(0.00014) s^2 + (0.0049) s + 1]} \tag{4}$$

IMPLEMENTATION OF PID

Automatic control system is playing an important role in today's modern society and due to ease of use and reliability PID is one of the most

famous control techniques and has been used in more than 90% of the installed automatic control systems all around the world [9]. It is combination of three term i-e Proportional, Integral and Derivative controller and it is mostly used to regulate the

time-domain response of vast range of dynamic systems [10]. The Proportional term performs control action proportional to the error, Integral term minimizes the steady state error by low frequency reparation through integrator and Derivative term improves the natural response of process by high frequency reparation through differentiator [11] PID calculates the error value which is the difference between the desired and the actual value of the system and attempts to reduce the error by adjusting the process control inputs [12].The architecture of PID controller is shown in Figure 2.

Where, K_p , K_i and K_d are gains of Proportional, Integral and Derivative elements of PID controller. The three controllers are connected in parallel and their output signal drives the plant. The output of plant or process is measured and compared with the set-point via feedback to find the error and then PID uses this error to again calculate the control signal and this process continues until the desired steady state is achieved. The mathematical representation of PID controller in S-domain is shown in Eq (5).

$$U(s) = \left[K_p + sK_d + \frac{K_i}{s} \right] E(s) \quad (5)$$

The effectiveness of PID controller depends on how well the gains have been tuned. There are number of tuning techniques developed in the re-

cent past to make PID more robust and more effective in dynamic environments. In this research work Linear Quadratic Regulator (LQR) is used to get optimal gains for effectively controlling the speed of DC motor. The tuning technique and the implementation flow chart is discussed below.

PID tuning using LQR

Conventional optimal control theory has been revolutionized in recent years to develop the famous LQR which reduces the excursion in state trajectories of a process while requiring least effort of controller [13]. This classic behavior has encouraged control system engineers to use LQR for tuning of PID gains [14]. LQR based PID tuning is shown in Figure 3 [15].

In the below figure for LQR tuning reference input $r(t) = 0$ as external reference input does not affect the controller tuning so as $r(t)$ is zero then $y(t) = -e(t)$ which is a valid linear regulator problem. The transfer function Eq. (5) transformed by Eq. (4) below:

$$\frac{Y(s)}{U(s)} = \frac{19.607}{0.00014s^2 + 0.0049s + 1} = \frac{-E(s)}{U(s)} \quad (6)$$

The LQR quadratic cost function is shown in Eq. (7):

$$J = \int_0^{\infty} [X^T(t)QX(t) + u^T(t)Ru(t)]dt \quad (7)$$

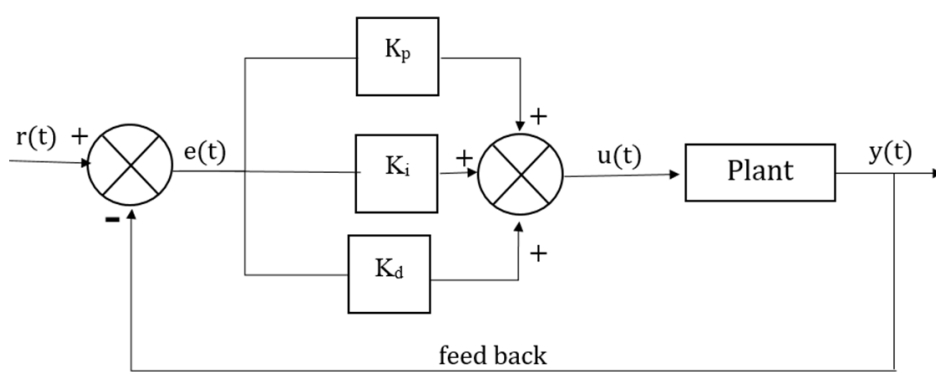


Fig. 2. Architecture of PID

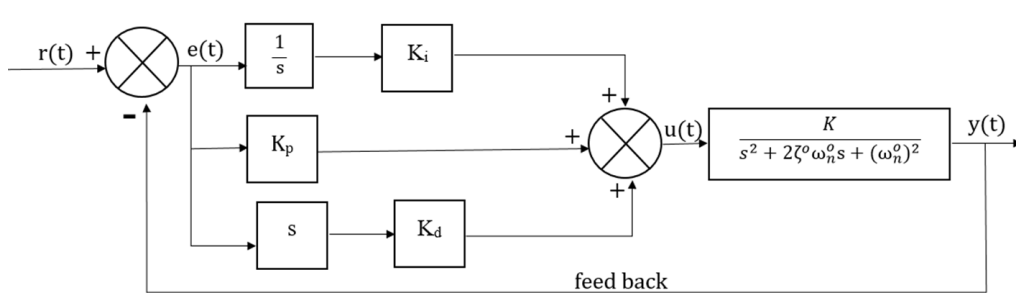


Fig. 3. LQR based PID tuning for Second Order System

Where Q and R are weight matrices, u(t) is the control signal and X(t) is the state vector. To get optimal control above cost function is to be minimized which results in the Eq. (8) given below [15]:

$$u(t) = -R^{-1}B^T P X(t) = -F X(t) \tag{8}$$

Where P is a symmetric positive definite solution of continuous algebraic Ricatti equation shown in Eq. (9):

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \tag{9}$$

Eq. (6) is transformed in to the differential form shown in Eq. (11):

$$0.00014 s^2 E(s) + 0.0049 E(s) + E(s) = -19.607 \tag{10}$$

$$0.00014 \ddot{e} + 0.0049 \dot{e} + e = -19.607u \tag{11}$$

Declare three state variables x_1 , x_2 and x_3 as shown below in Eq. (12)

$$x_1 = \int e(t) dt, x_2 = e(t), x_3 = \frac{de(t)}{dt} \tag{12}$$

So the Eq. (11) minimizes to Eq. (13):

$$0.00014 \dot{x}_3 + 0.0049 \dot{x}_2 + x_2 = -19.607u \tag{13}$$

State space equation of Eq. (13) is shown in Eq. (14):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -7142.85 & -35 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -140050 \end{bmatrix} u \tag{14}$$

$$F = R^{-1} B^T P = R^{-1} [0 \quad 0 \quad -140050] \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} \tag{15}$$

$$= -140050 R^{-1} [P_{13} \quad P_{23} \quad P_{33}] \tag{16}$$

As Q is a symmetric positive weighting matrix and R is the constant weighting factor, so generally R is kept fixed while Q is varied to get optimized control signal from LQR.

$$u(t) = -F x(t) = 140050 R^{-1} [P_{13} \quad P_{23} \quad P_{33}] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{17}$$

Characteristics of a closed loop polynomial system is shown in Eq. (18) [17]:

$$s^3 + [2\zeta^0 + R^{-1}k^2 P_{33}]s^2 + [(\omega_n^0)^2 + R^{-1}k^2 P_{23}]s + R^{-1}k^2 P_{13} = 0 \tag{18}$$

Modify the above equation for preferred damping ratio (ζ) and natural frequency (ω_n), the characteristic polynomial for a closed loop system is shown in Eq. (19):

$$s^3 + (2 + m)\zeta^c \omega_n^c s^2 + [(\omega_n^c)^2 + 2m(\zeta^c)^2 (\omega_n^c)^2]s + m\zeta^c (\omega_n^c)^3 = 0 \tag{19}$$

By comparing the above two equations gains of PID can be found by the following relations:

$$k_i = \frac{m\zeta^c (\omega_n^c)^3}{k} \tag{20}$$

$$k_p = \frac{(\omega_n^c)^2 + 2m(\zeta^c)^2 - (\omega_n^0)^2}{k} \tag{21}$$

$$k_d = \frac{(2+m)\zeta^c - 2\zeta^0 \omega_n^0}{k} \tag{22}$$

By putting the values of open loop and closed loop $\zeta^0, \omega_n^0, \zeta^c, \omega_n^c$ and m in above equations PID gains k_p, k_i and k_d can be easily calculated. In this research work $\zeta^0=0.00245$ and $\omega_n^0=1$ are taken from Eq. (4),

while desired closed loop values are $\zeta^c=1$, $\omega_n^c=15$ and $m=9$. Finally $k_p=207$, $k_i=1549$ and $k_d=7$ have been calculated.

PID Implementation Flow Chart

The first step to implement PID for speed control of a DC motor is to attain the transfer function of DC motor and then find the PID gains using LQR. After that introduce the reference point, introduce gains and find the difference between the desired value and the actual value in the form of error to calculate the control signal that is applied to DC motor. Finally a feedback is taken from output to find the error and this procedure continues until the DC motor is regulated at desired speed. The implementation flow chart is shown in Figure 4.

SOFTWARE TOOLS

The software tools used in this research work to develop the simulations of implementation of PID on PIC24F series microcontroller for speed control of a DC motor are MPLAB and Proteus Professional. They are precisely discussed below.

MPLAB IDE

MPLAB Integrated Development Environment (IDE) is software to develop programs for PIC microcontrollers. It is called IDE because it provides development, debugging and software services on a single platform. The Language tools present in MPLAB are Assembler (MPASM), Linker (MPLINK) and a C compiler (MPLAB C30). The newest version present at the time of this writing is MPLAB X IDE but in this research work MPLAB 8.10 version with C 30 compiler is used. The steps required to develop a program for the PIC controller and generate hex file using MPLAB is shown in Figure 5 [7].

Proteus

Proteus is classic Electronic Design Automation (EDA) simulation software. The huge device library and wide range of peripherals is the special feature of this simulator. The virtual instrumentation feature offers a great platform for simulation of microcontroller units. Proteus Virtual System Modeling (VSM) uses mixed mode SPICE circuit simulation, animated components and mi-

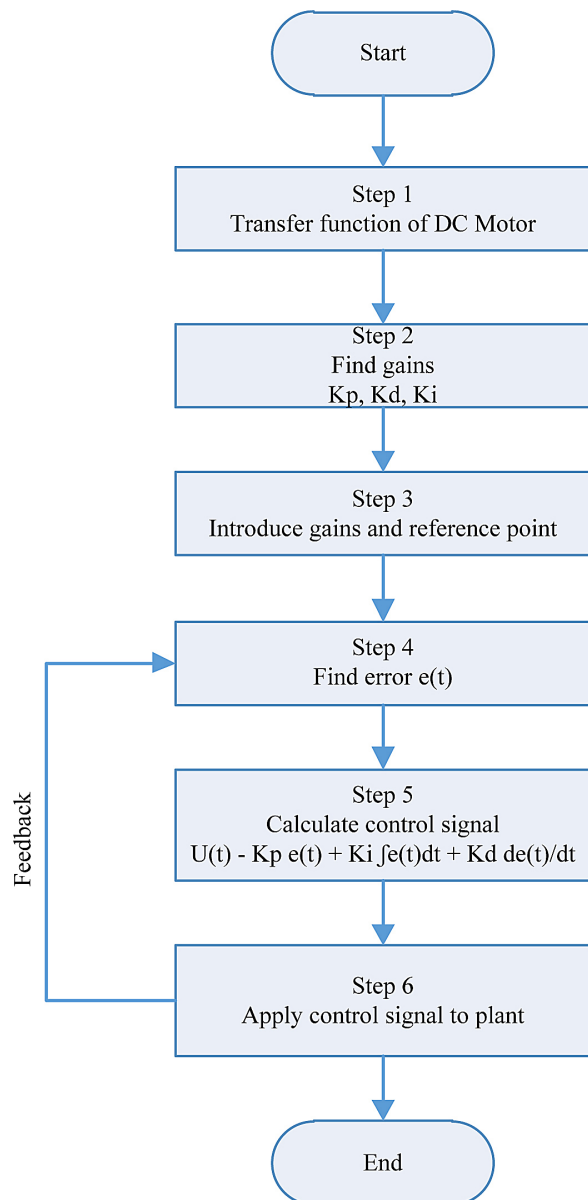


Fig. 4. Flow Chart of PID Implementation

croprocessor models to make possible complete simulations of microcontroller based designs. For the first time ever, it is possible to develop and test such designs before a physical prototype is constructed [16]. The steps required to simulate microcontroller based applications in Proteus are shown in Figure 6.

SIMULATIONS

The simulations have been developed by first implementing the block diagram then following the speed controller algorithm flow chart and finally implementing the circuit diagram in Proteus to realize the speed controller. The block diagram,

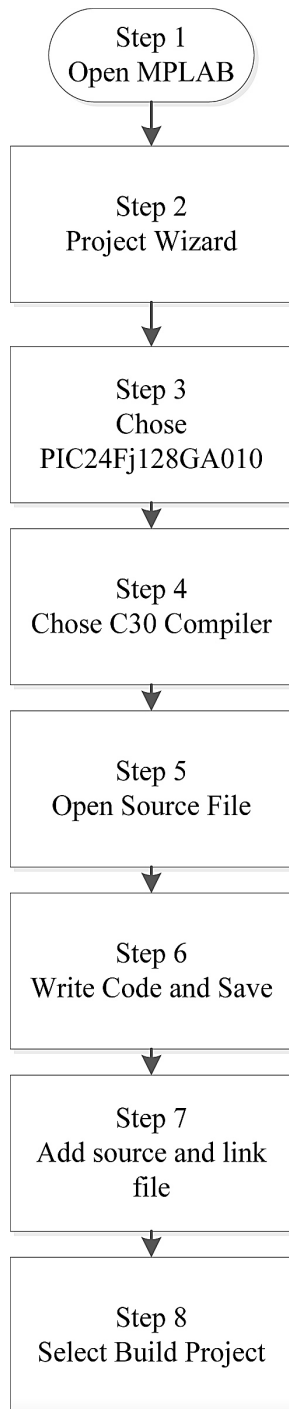


Fig. 5. Steps to develop program for PIC Microcontroller

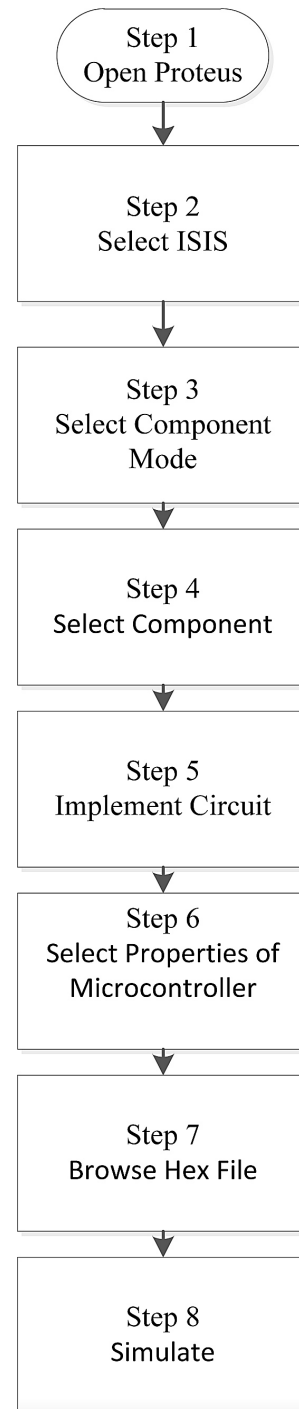


Fig. 6. Steps to Simulate PIC Microcontroller

algorithm flow chart and circuit diagram is elaborated below.

Block diagram

The block diagram of implementation of PID on microcontroller for speed control of a DC motor in Proteus is shown in Figure 7. Input transducer converts the reference speed to the form compatible to microcontroller then microcon-

troller generates the required PWM by using PID control technique to control the motor speed at desired value. Motor driver regulates the voltage at terminals of DC motor. Output transducer measures the real speed and transforms it to the form compatible to the microcontroller and send the value to microcontroller as a feedback for estimation of PWM. Display unit presents the results of actual speed and the percentage of duty cycle.

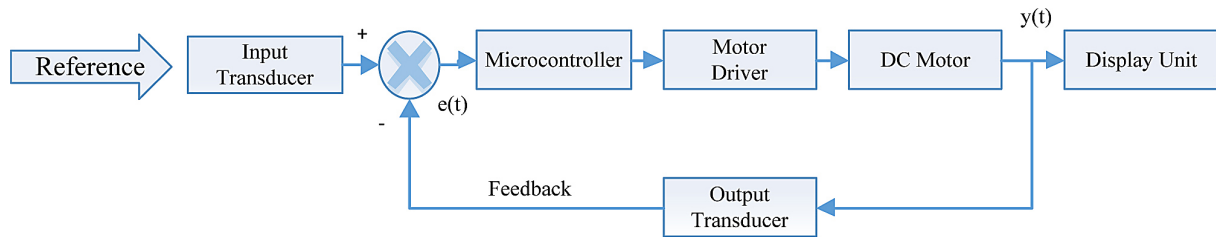


Fig. 7. Block Diagram of Speed Controller Implementation

Input transducer / User interface

Keypad is used as a user interface to provide the desired reference speed for the DC motor and it is also used to select three different cases for speed control of DC motor. In this paper a 4*4 keypad is used. Keypad uses its internal libraries to enable scanning of 4*4 switch array and return the data related with the pressed button. Interfacing of keypad with microcontroller can be done in three ways; depending on the design requirements and pin resources. They are scanning method, logic change interrupt method and external interrupt using IC. In this paper scanning method is used which needs 8 pins for interfacing [7].

Microcontroller

PIC24FJ128GA010 belongs to CMOS family with less power consumption because of power handling modes i-e sleep, idle and alternate clock modes. Its operating voltage range is 2.0 V -3.6V. It is a general purpose 100-pin microcontroller with a modified Reduced Instruction Set computer (RISC) architecture which can operate at up to 32 MHz crystal oscillator with speed of 16 Mega Instructions Per Second (MIPS) [17]. PIC microcontroller uses PID algorithm to generate PWM pulses for regulation of DC motor. PWM generation is elaborated below.

Pulse Width Modulation

PWM is a versatile method of providing varying amounts of electrical power between fully ON and fully OFF. A normal power switch with some power source provides full power only, when switched on. PWM is a relatively new technique, and can be implemented by the latest electronic power switches. In PIC24FJ128GA010 Output Compare Module (OCM) is utilized to generate PWM and it has three pins to produce PWM they are OC1, OC2 and OC3. OCM has number of working modes i-e

Single Compare Match Module, Double Compare Match Mode Generating and Simple PWM. OCM is initialized using OCXCON control register and for generating PWM the first three bits of OCXCON register (OCXCON<2:0>) are set to (110)2. PWM of specific frequency is generated by writing the number of cycles required to produce PWM period to period register of chosen timer. The period register value is calculated by using the following Eq. (23) [17].

$$PWM\ Period = [(PRY) + 1] \cdot TCY \cdot (Prescale\ Value) \quad (23)$$

where: $PWM\ frequency = 1/[PWM\ Period]$ and $TCY = 2/FOSC$.

The duty cycle of PWM is set by putting its value in OCXRS register but first put the initial value of duty cycle to OCXR register then OCXR register becomes only Read-Only duty cycle register after that OCXCON control register is initialized for simple PWM generation [17]. In this research work PWM is generated with frequency of 800 Hz using 16MHz crystal oscillator. The calculated value of clocks for period register using (23) is (2710)16 and the initial value of PWM written to OCXR register is (0F)16. Timer 2 is used for generating interrupt with a frequency of 800Hz. PWM generation is developed in MPLAB by following the sequence of commands shown in Figure 8 [7].

Motor driver

Controller generates 5V PWM and is applied at the input of the driver circuit. Because of PWM driver circuits gives pulses of 12V DC voltage to regulate power at the terminals of DC motor. In this paper Dual H-Bridge IC L293D is used as motor driver. This motor driver is a high current, high voltage four channels driver which is compatible with both DTL and TTL logic. Each channel is capable of producing 600mA output current and can provide up to 1.2A peak output current to drive motor.

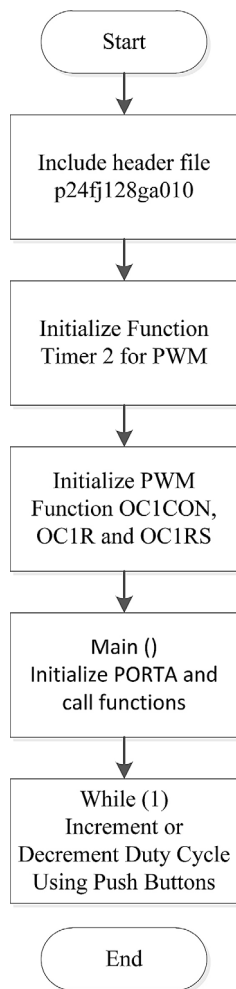


Fig. 8. Algorithm Sequence for PWM generation using OCM

DC motor

The DC motor used in Proteus has 12V and 3600RPM voltage and speed ratings respectively. The DC motor has built in load torque capability. The speed of DC motor is controlled by PID for 0% load torque, 25% load torque and 50% load torque.

Output transducer

Encoder is used to measure the real speed of motor and converts it into a form compatible to microcontroller. This information is given back to the controller for the estimation of control signal to generate accurate PWM to drive the motor at desired speed. The DC motor used in Proteus has built in encoder to measure speed.

Display unit

LCD is used to show real time results and found in number of applications. In this research work a

16*2 display LCD module is used. A 16*2 LCD has 16 characters in each line and there are two such lines. LCD operations are based on two registers namely command and data where command register controls the operation and data register stores the data to be displayed [7]. In this paper LCD will display real speed, percentage of duty cycle and the number of loops in which speed is regulated.

Microcontroller based PID speed controller algorithm

To develop a complete algorithm for microcontroller based PID speed controller in MPLAB first include header files then declare the pin connections of both Keypad and LCD after that initialize keypad, LCD and PWM functions and also define an Interrupt Service Routine (ISR). In ISR feedback is taken from motor and converted to $e(t)$ to be used by PID to calculate the control signal. In main loop all the functions are called and in while loop PID is implemented and PWM is generated and it is further given to the motor driver to regulate the speed and during while loop if interrupt occurs ISR is executed and this procedure continues for ever until system is stopped. The algorithm flow chart is shown in Figure 9.

Circuit diagram

The circuit diagram of microcontroller based PID speed controller of DC motor is shown in Figure 10. In the above figure LCD and Keypad are connected to PORT B and PORT A respectively. The PWM is generated at Pin No. 72 and it is connected to the input of motor driver. The output terminals of motor driver are connected to DC motor. Motor encoder output in form of pulses is given at Pin No. 73 as a feedback to generate required PWM using PID algorithm. In this way microcontroller based PID speed controller regulates the speed of DC motor.

RESULTS

The simulation results of speed control of DC motor are presented in three cases:

- Case 1: Speed control of DC motor without load torque.
- Case 2: Speed control of DC motor with 25% load torque.
- Case 3: Speed control of DC motor with 50% load torque.

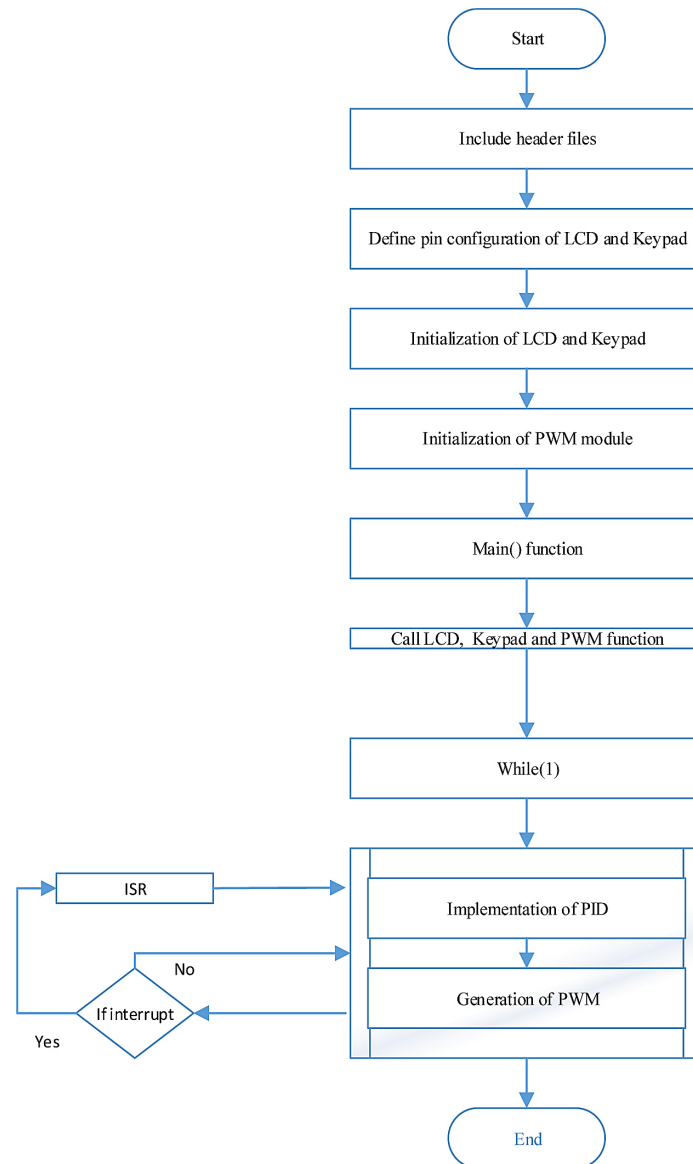


Fig. 9. Algorithm sequence for implementing speed controller

The PID gains in all three cases are same. They are $k_p=207$, $k_i=1549$ and $k_d=7$.

Case 1

The simulation results of speed control with set-point of 100RPM and with 0% load torque are presented in Figure 11. In the Figure 11 LCD shows the actual speed, percentage of duty cycle and number of loops required reaching that speed and oscilloscope shows the respective PWM for that speed. From above figure it is clear that the DC motor speed reaches peak overshoot of 104RPM in 8 loops with duty cycle of 33% and then successfully settles after two loops at 99RPM only giving 1% of steady state error with 31% duty cycle.

Case 2

The simulation results of speed control with set-point of 100RPM and with 25% load torque presented in Figure 12. From Figure 12 it is clear that by adding the load torque to DC motor more duty cycle of PWM and more time required to reach the same speed so in case 2 DC motor speed reaches peak overshoot of 104RPM in 11 loops with duty cycle of 39% and then successfully settles after one loop at 99RPM only giving 1% of steady state error with 38% duty cycle.

Case 3

The simulation results of speed control with set-point of 100RPM and with 50% load torque

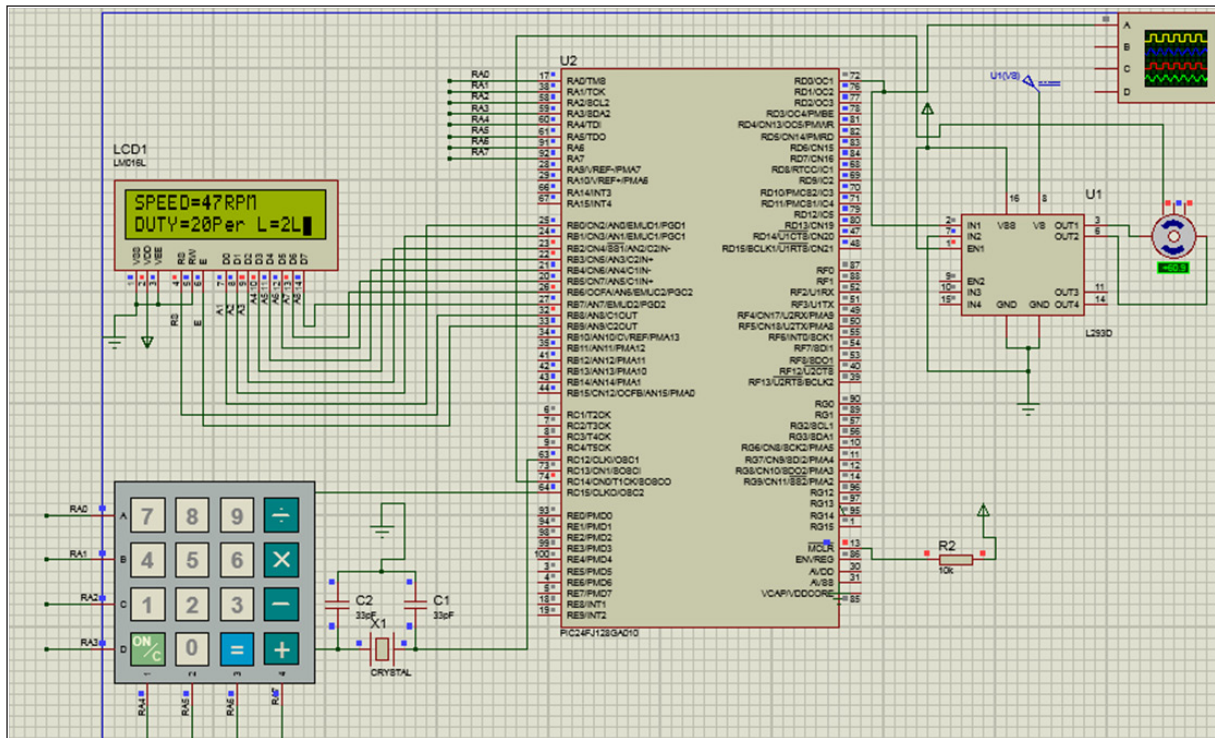


Fig. 10. Circuit diagram in Proteus

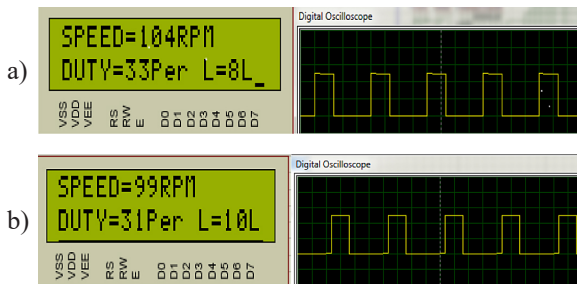


Fig. 11. Speed Control of DC motor without Load Torque: a) Overshoot Results, b) Steady State Results

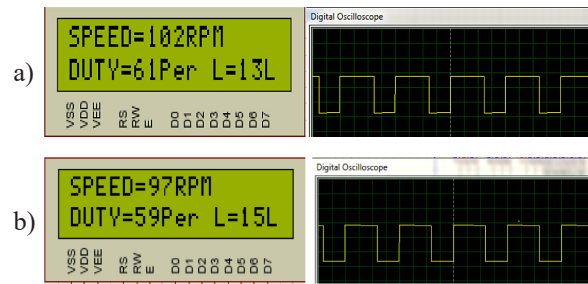


Fig. 13. Speed Control of DC motor with 50% Load Torque: a) Overshoot Results, b) Steady State Results

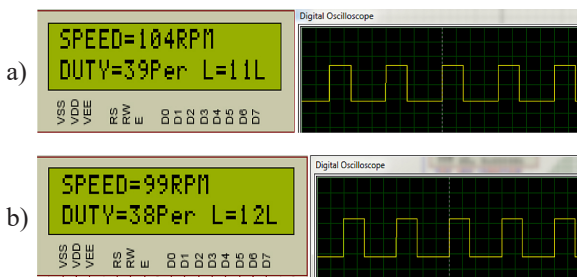


Fig. 12. Speed Control of DC motor with 25% Load Torque: a) Overshoot Results, b) Steady State Results

overshoot of 102 RPM in 13 loops with duty cycle of 61% and then successfully settles after two loops at 97RPM giving 3% of steady state error with 59% duty cycle. The simulation results of all three cases have been summarized in Table 2.

From Table 2 it is evident that even by increasing the load torque PID effectively regulates the speed of DC motor at desired speed but both peak time (t_p) and settling time (t_s) also increases. The steady-state error remains constant in first two cases but varies in last case. The duty cycle of PWM has also been increased to regulate the motor at desired speed with increase in load torque. It is also clear that in dynamic environments PID gains have to be adjusted again and again to get optimum transient and steady state response.

presented in Figure 13. From Figure 13 it is clear that increasing the load torque to DC motor results in large increase in duty cycle of PWM and more time required to reach the same desired speed so in case 3 DC motor speed reaches peak

Table 2. Simulation results of all cases

No.	Case	Load torque	Set-point (RPM)	Peak time (tp) (loops)	Setting time (Ts) (loop)	Over shoot	Steady state error	PWM (duty)
1	1	0%	100	8	10	4	1	31
2	2	25%	100	11	12	4	1	38
3	3	50%	100	13	15	2	3	59

CONCLUSION

In this research work PID has been implemented on a 16-bit PIC24FJ128GA010 microcontroller for speed control of a DC motor. The Proteus simulations have been analyzed without load torque, with 25% load torque and with 50% load torque. In all three cases PID effectively controls the speed of DC motor but with increased in tp, ts and duty cycle of PWM which motivates the researchers to work on online optimization of PID gains. Moreover, implementation of PID on PIC24F series microcontroller using MPLAB and Proteus software has given an opportunity to implement and test more control techniques i-e Fuzzy Logic, Linear Quadratic Regulator (LQR) and Model Predictive Control (MPC) for not only Single Input Single Output (SISO) but also for Multiple Input Multiple Output (MIMO) systems as well. Finally this paper also convinces the idea of using the combination of these two tools in educational sector to enhance embedded system skills.

REFERENCES

- Shrivastava S., Rawat J., Agarwal A.: Controlling DC Motor using Microcontroller (PIC16F72) with PWM. *International Journal of Engineering Research*, 1, 2012, 45-47.
- Bhagat N.A., Bhaganara M.: DC Motor Speed Control using PID Controllers, *Electronic System Design Course Project*. University of IIT Bombay, 2009.
- Kamar U., Narvey R.: Speed Control of DC Motor Using Fuzzy PID Controller. *Advance in Electronic and Electric Engineering*, 3, 2013, 1209-1220.
- Saridhar N., Ramrao N., Singh M.K.: PID Controller Auto Tuning using ASBO Technique. *Journal of Control Engineering and Technology*, 4, 2014, 192-204.
- Das S., Pan I., Halder K., Das S., Gupta A.: LQR based improved discrete PID controller design via optimum selection of weighting matrices using fractional order integral performance index. *Applied Mathematical Modeling*, 37, 2013, 4253-4268
- Mukherjee A., Ray S., Das. A.: Development of Microcontroller Based Speed Control Scheme of BLDC Motor Using Proteus VSM Software. , *International Journal of Electronics and Electrical Engineering*, 2014, 2, 1-7.
- Aslam S., Hannan S., Haider A., Tariq M.H.: Exploring PIC 24F Series Microcontroller using MPLAB and Proteus. *Journal of Current Research in Science*, Accepted and Published in June, 2016.
- Hamed B., Al-Mobaid M.: Fuzzy PID controllers using FPGA technique for real time DC motor speed control. *International journal of Intelligent control and automation*, 2, 2011, 233-240.
- Aslam S., Hannan S., Haider A.: Effect of Laguerre function parameters on MPC performance for speed control of a DC motor. *Journal of Control Engineering and Technology*, 6, 2016, 1-13.
- Singh A.P., Narayan U., Verma A.: Speed Control of DC Motor using Pid Controller Based on Matlab. *Innovative Systems Design and Engineering*, 4, 2013, 1-7.
- Hang C.C, Sin K.K.: A comparative performance study of PID auto- tuners. *IEEE Control System*, 11, 1991, 41-47.
- Vikhe P., Punjabi N., Kadu C.: Real Time DC Motor Speed Control using PID Controller in LabVIEW, *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 3, 2014, 12162-12167.
- Anderson B.D.O., Moore J.B.: *Optimal Control: linear quadratic methods*, Prentice-Hall International, Inc., Englewood Cliffs, NJ, 1989
- He J.B., Wang Q., Lee T.H.: PI/PID controller tuning via LQR approach. *Chemical Engineering Science*, 5, 2000, 2429-2439.
- Kumar V.E., Jerome J.: LQR based optimal tuning of PID controller for trajectory tracking of magnetic levitation system, *International conference on design and manufacturing (Procedia Engineering)*, 64, 2013, 254-264.
- Chen J.: Application of Proteus Software in MCU teaching. *Second International Conference on Mechanic Automation and Control Engineering*, 2011, 6359-6362.
- Microchip, PIC24FJ128GA010, datasheet, 2006.