

DC Motor Speed Control using PID Controllers

Nikunj A. Bhagat (08307908) $\langle nbhagat@ee.iitb.ac.in \rangle$,
Mahesh Bhaganagare (CEP) $\langle mab@ee.iitb.ac.in \rangle$,
Instructor: Prof.P.C.Pandey

December 1, 2009

Abstract

An Implementation of PID controllers for the speed control of a DC motor is given in this report. The motor is modeled as a first order system and its response is studied. The speed control using PI and PID control modes is explained and an implementation of the controller using OP-AMPs is given. The response of the controller to load variations is looked at.

1 Introduction

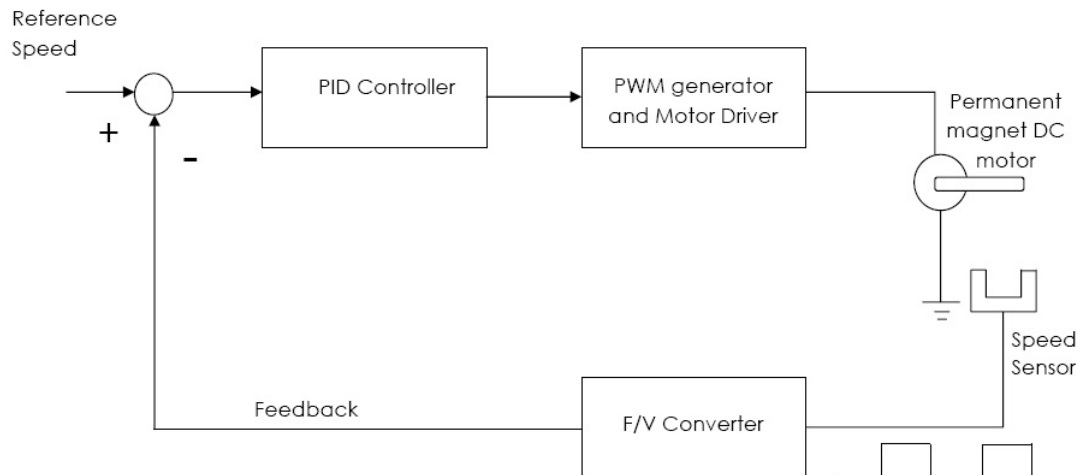


Figure 1: DC Motor Speed Control System

The speed control of a motor is frequently required in industrial applications, robotics, home appliances, etc. In this report, we have implemented a DC motor speed control system. The idea of a speed control system is to maintain the speed of the motor at the desired value under various condition. In practice, the DC motor is a nonlinear device and its speed varies because of change in load demand, disturbances, etc. We have implemented the PID controller algorithm which is a popular controller in industries.

The block diagram of a DC motor speed control system is shown in figure(1). The motor speed is sensed by an optical switch and converted to feedback voltage. It is compared with

the reference signal (i.e. desired speed) by the error detector. The PID controller acts on the error signal and generates appropriate control voltage. The PWM generator block then varies the duty cycle of the voltage supplied to the motor to control its speed.

In the report, section 1 describes the open loop response of DC motor, and calculation of its first order transfer function. Section 2 gives details of speed measurement using optical switch. The Frequency to Voltage Converter in discussed is section 3. Section 4 describes PID theory and its design. Section 5 gives details of PWM generator and motor driver, followed by the complete schematic diagram of the control system in Section 6. Section 7 and 8 shows the test results and discussion. The report is concluded in section 9.

2 DC Motor Specifications

The speed control system was implemented for a Permanent Magnet DC Motor(PMDC). The PMDC consists of rotor or armature and a stator, which is a permanent magnet. There are two ways of speed control for a DC motor,

- Field Control
In this method, the field current or current through stator is varied to control the speed of the motor.
- Armature Control
In this method, the armature voltage is varied to control the speed of the motor.

For the PMDC, a constant field is generated by a permanent magnet and hence we decided to implement Armature control. To control the armature voltage we are generating a Pulse-Width Modulated (PWM) waveform to control the average voltage applied to the motor. The details of the PWM implementation are given in section(5).

DC motor Specifications:

1. 12V Permanent Magnet DC Motor
2. Rated current: 200mA at no load, 290mA at full load
3. Torque: 50gm-cm
4. Maximum speed: 2500 rpm

For the implementation of the project, a DC motor setup available in Control and Computing Lab, EE Dept, IIT Bombay, was used. This setup is manufactured by TECHNO INSTRUMENTS and is currently being used for laboratory experiments. The setup includes a PMDC fitted with slotted aluminum disk and a slotted optical switch H21A1 for speed measurement. This setup has a provision to load the motor using magnetic braking. The effect of loading the motor and the performance of the controller were studied.

2.1 Transfer Function of DC Motor

To study the open loop response of the DC motor, it was modeled as a first order system. The Transfer function of armature controlled DC motor is given in [1],

$$\frac{\omega(s)}{V(s)} = \frac{K_m}{sT + 1} \quad (1)$$

ω : Speed of the motor (rpm)
 V : Armature voltage (V)
 K_m : motor gain constant (rpm/volt)
 T : motor time constant (sec)

2.2 Measurement of Transfer function constants

To evaluate the transfer function of the DC motor, experiments were conducted to study its open loop performance. The values of K_m and T are calculated as explained below.

2.2.1 Measurement of K_m

1. The motor speed is measured for various values of armature voltage as shown in Table1.
2. A graph of armature voltage v/s motor speed is plotted, figure(2). The slope of the graph is the value of K_m .
3. The measured valued of $K_m = 529.9$ rpm/V.

2.2.2 Measurement of T

1. The motor speed is varied from 0 to 2500 rpm. The output of the Frequency -to- Voltage converter, called the feedback voltage V_{fb} is measured.
2. A graph of motor speed v/s V_{fb} is plotted as shown in figure(3). The slope of the graph is the gain of the feedback system, K_t in V/rpm.
3. The measured value of $K_t = 1.84 * 10^{-3}$ V/rpm.
4. To measure T , a square wave $V_m = 4.4V_{pp}$ and frequency $f = 500$ Hz is given as reference input. The measured value of $V_{fb} = 2.62$ V. Using equation(), the calculated value of $T = 1.06$ msec.

$$T = \frac{1}{2f} \frac{1}{\ln(1 - \frac{V_{fb}}{V_m} \frac{1}{K_m K_t})} \quad (2)$$

The calculated transfer function of the DC motor is,

$$G(s) = \frac{529.9}{1 + 1.06 * 10^{-3}s} \quad (3)$$

3 Speed measurement

To measure the motor speed an optical switch is used. It is a LED and photo transistor pair, which generates pulses corresponding to motor speed. A slotted disk having 12 slots is fixed on the motor shaft and it cuts the path between the LED and photo transistor when the motor rotates. For one rotation of the motor, the optical switch generates 12 pulses. The maximum motor speed is 2500 rpm. Therefore the number of pulses generated per second at the output of optical switch, at maximum motor speed = $\frac{2500}{60} * 12 = 500$ pulses/sec. These pulses which vary from 0 - 5V are then converted to ± 5 V using a OPAMP 741. Figure(4) shows the circuit

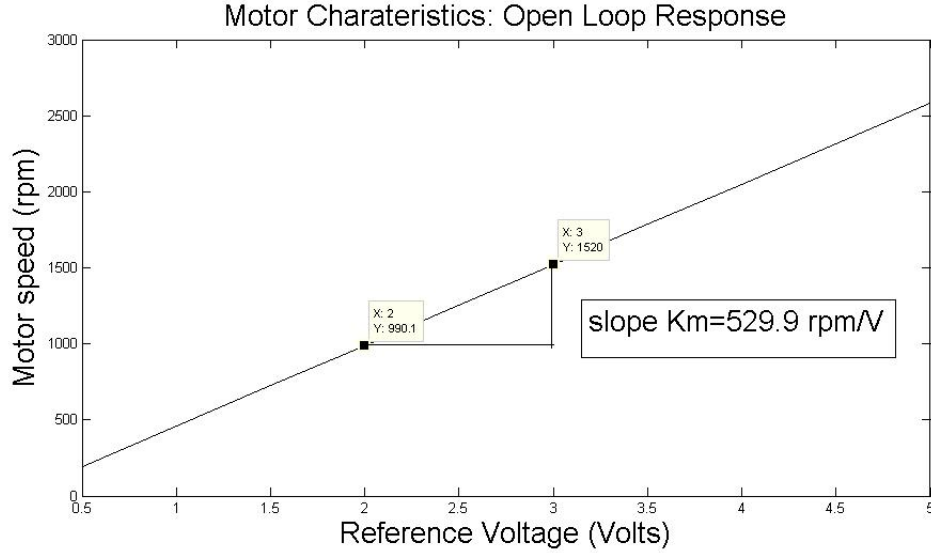


Figure 2: Motor Characteristics

schematic of the optical switch along with PWM generator and motor. We are using H21A1 optical switch manufactured by Fairchild semiconductors.

H21A1 Specifications:

1. $I_f(\text{max}) = 50 \text{ mA}$.
2. $I_c(\text{max}) = 20 \text{ mA}$.
3. $V_D(\text{rev.}) = 6 \text{ V (max)}$

4 Frequency to Voltage (F/V) Converter

The pulses from optical switch are given to a F/V converter. The F/V converter produce a DC voltage (0-5 V) proportional to motor speed. We are using LM2907N F/V converter manufactured by National semiconductor. The circuit diagram of the F/V converter is shown in figure(5).

4.1 Internal Working of LM2907N

The internal schematic of LM2907N is shown in figure(6)[3]. The input stage of LM2907N is a charge pump where the input frequency is converted to a dc voltage. To do this it requires one timing capacitor C_1 , one output resistor R_1 , and one integrating or filter capacitor C_2 . When the input stage changes state (due to a suitable zero crossing or differential voltage on the input) the timing capacitor is either charged or discharged linearly between two voltages whose difference is $\frac{1}{2}V_{cc}$. Then in one half cycle of the input frequency or a time equal to $\frac{1}{2}F_{in}$ the change in charge on the timing capacitor is equal to $\frac{1}{2}V_{cc} * C_1$. Then the average amount of current pumped into or out of the capacitor is,

$$\frac{\Delta Q}{T} = I_{c(avg)} = C_1 * \frac{1}{2}V_{cc} * 2F_{in} = V_{cc} * C_1 * F_{in} \quad (4)$$

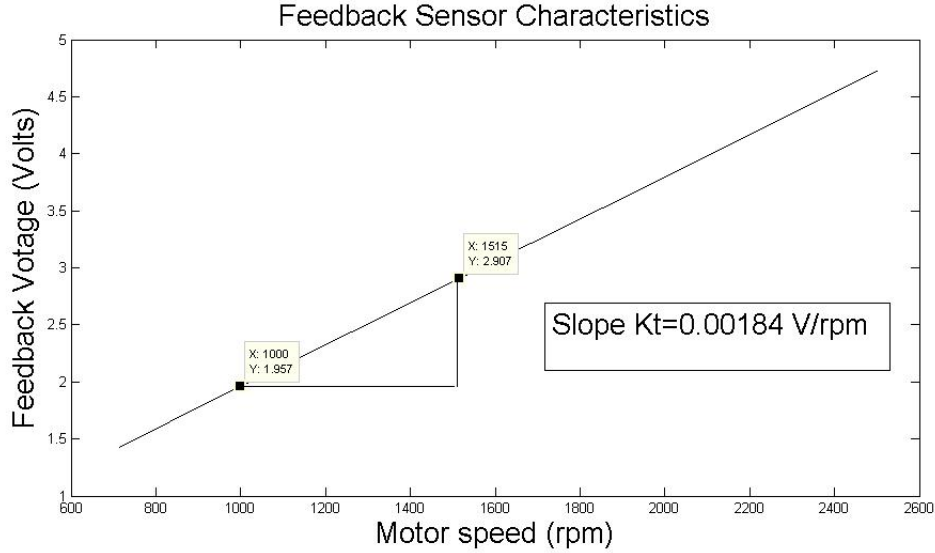


Figure 3: Feedback Sensor Characteristics

The output circuit mirrors this current very accurately into the load resistor R_1 , connected to ground, such that if the pulses of current are integrated with a filter capacitor, then $V_o = I_c * R_1$ and the total conversion equation becomes,

$$V_o = V_{cc} * F_{in} * C_1 * R_1 * K. \quad (5)$$

where K is gain constant, typically $K=1$. The size of C_2 is dependent only on the amount of ripple voltage allowable and the required response time.

4.2 Selecting R_1 and C_1

There are some limitations on the choice of R_1 and C_1 which should be considered for optimum performance. The timing capacitor also provides internal compensation for the charge pump and should be kept larger than 500 pF for very accurate operation. Smaller values can cause an error current on R_1 , especially at low temperatures. The output current at pin 3 is internally fixed and then $\frac{V_o}{R_1}$ must be less than or equal to this value. If R_1 is too large, it can become significant fraction of the output impedance at pin 3 which degrades linearity. Also output ripple voltage must be considered and the size of C_2 affected by R_1 . An expression that describes the ripple content on pin 3 for a single $R_1 C_2$ combination is,

$$V_{ripple} = \frac{V_{cc}}{2} \frac{C_1}{C_2} \left(1 - \frac{V_{cc} * F_{in} * C_1}{I_2} \right) \quad (6)$$

It appears R_1 can be chosen independent of ripple, however response time, or the time it takes V_o to stabilize at a new voltage increases as the size of C_2 increases, so a compromise between ripple, response time and linearity must be chosen carefully.

Lastly, maximum attainable input frequency is determined by V_{cc} , C_1 and I_2 ,

$$f_{max} = \frac{I_2}{C_1 * V_{cc}} \quad (7)$$

The ripples in the F/V converter is more at lower frequency, so we used F/V Converter with 2 pole Butterworth filter to reduce the ripple in the output voltage. Lower pole frequency of

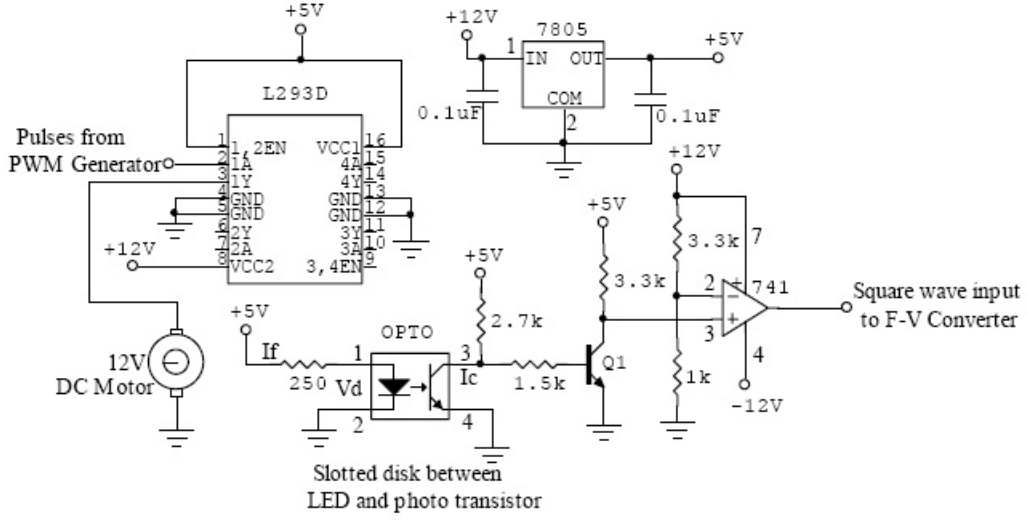


Figure 4: Schematic: Speed Sensor

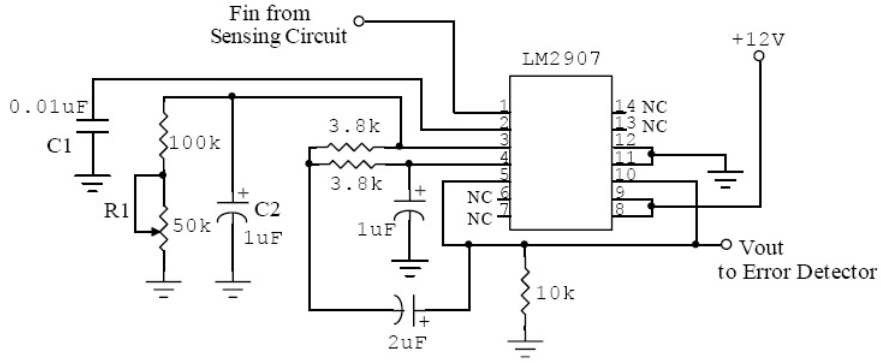


Figure 5: Schematic: F/V converter

30 Hz is decided to get the linearity with reduced ripples in the output voltage. The designed values are shown in figure(5).

5 PID controller

The Proportional(P)+Integral(I)+Derivative(D), i.e. PID control algorithm is most widely used in industrial applications. We have implemented a PID controller to control the motor speed. The implementation of a PID controller is shown in fig(7).

The error between the reference speed and the actual speed is given as input to a PID controller. The PID controller depending on the error changes its output, to control the plant input such that the error is minimized. A detailed information about the theory and tuning of PID controllers is given in [2]. The Transfer function of a PID controller is given as,

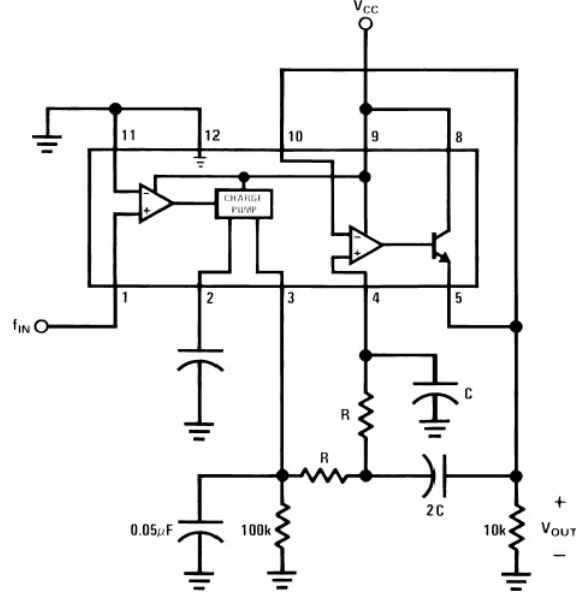


Figure 6: Internal Schematic of LM2907N [3]

$$C(s) = K_P \left(1 + \frac{1}{T_I * s} + T_D * s \right) \quad (8)$$

K_P : Proportional Gain constant

T_I : Integral reset time

T_D : Derivative time or rate time

The proportional control(K_p) is used so that the control signal $u(t)$ responds to the error immediately. But the error is never reduced to zero and an *offset* error is inherently present. To remove the offset error the Integral control action(T_I) is used. To Derivative control(T_D) is used to dampen out oscillations in the plant response. Also, the presence of derivative control reduces the need of K_p being large to achieve stability[2].

In this project, PI and PID controller were implemented and the performance of the system was evaluated. The control algorithms were implemented using OP-AMPs. The details of implementation is given in the following sections.

5.1 Method of PID Tuning

The tuning of PID controller involves designing the controller constants, K_P, T_I and T_D depending on the performance requirements. There are various methods available in literature to design these values. We have used the Ziegler-Nichols Rules for determining the controller constants[2]. These rules have been developed from experiments and by analyzing various industrial processes. We are using the Ultimate Cycle Method as given by Ziegler and Nichols Rules, which uses the results of the experiments performed, with the controller already installed in the loop. The design steps are given below.

5.1.1 Design of PID constants using Ultimate cycle method

1. Complete the closed loop by connecting the controller before the plant.

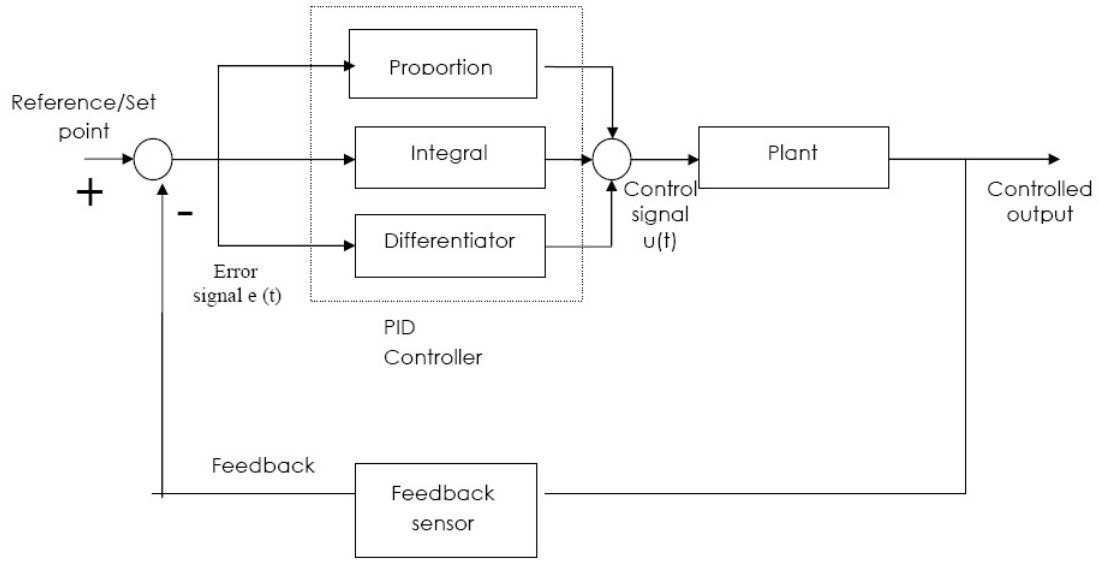


Figure 7: Schematic: PID implementation

2. Disconnect I and D control modes. Initially set K_P at a small value.
3. Increase K_P till the plant begins to exhibit *sustained* oscillations.
4. Measure the time period of oscillations (P_u) and the corresponding ultimate gain(K_{pu}).
5. Use P_u and K_{pu} along with table(??) to find the values of K_P, T_I and T_D

Table 1: Ziegler-Nichols Rules

Control Mode	Ultimate Cycle Method
P control	$K_P = 0.5K_{pu}$
PI control	$K_P = 0.45K_{pu}$ $T_I = 0.83P_u$
PID control	$K_P = 0.6K_{pu}$ $T_I = 0.5K_{pu}$ $T_D = 0.125P_u$

To above described method was used to determine the controller values as,
 $K_P = 5.1$,
 $T_I = 13.28$ usec,
 $T_D = 0.025$ sec.

5.2 PID Implementation and Testing

The P,I and D blocks were implemented using OP-AMP TLO84. The circuit schematic is given in figure(8).

For the Proportional control,

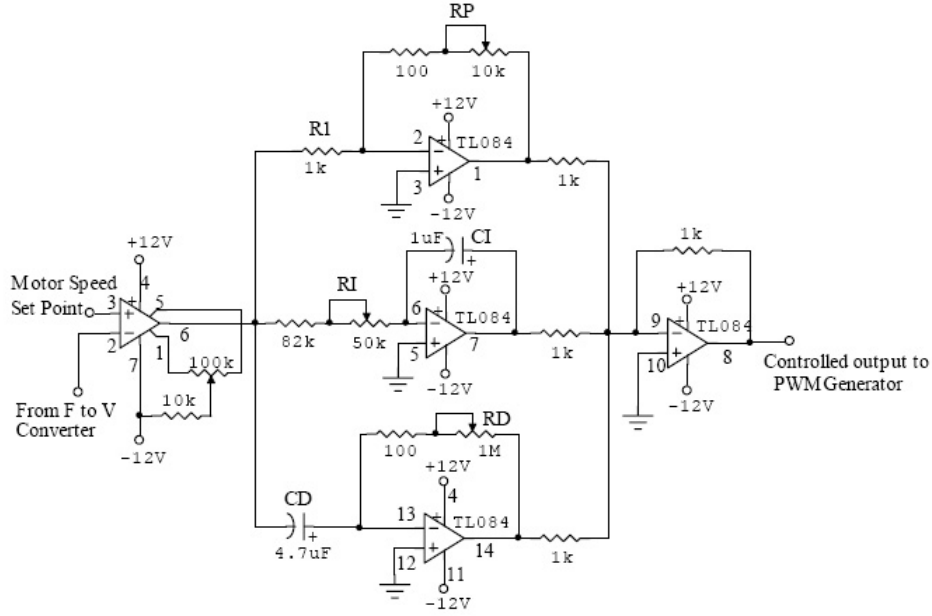


Figure 8: Schematic: PID implementation

$$K_P = 1 + \frac{R_F}{R_1} \quad (9)$$

For the Integral control,

$$T_I = R_I * C_I \quad (10)$$

For the Derivative control,

$$T_D = R_D * C_I \quad (11)$$

As shown in the circuit diagram fig(8), by varying R_F, R_I and R_D the controller values can be adjusted. The PID circuit was tested by giving a $1 V_{PP}$ square wave as input. The values of R_F, R_D and R_I were adjusted till the desired response was obtained.

6 PWM Generator and motor driver

A pulse width modulator is a circuit which generates a square waveform for a DC input voltage. The duty cycle of the square wave can be controlled by changing the magnitude of the input voltage. A PWM generator can be used to control the average voltage supplied to a DC motor, by varying the duty cycle, to control the motor speed. This technique is used in our project, in which the PID controller changes the control voltage, depending on error. The control voltage is applied to the PWM generator which accordingly, changes the duty cycle of the PWM output. The advantage of PWM is that the pulses reach the full supply voltage and will produce more torque in a motor by being able to overcome the internal motor resistances more easily.

Two circuits were tested for the PWM generator. The first circuit, figure(9) shows a PWM generator implemented using a triangular wave generator and a comparator. Following problems were faced in the implementation.

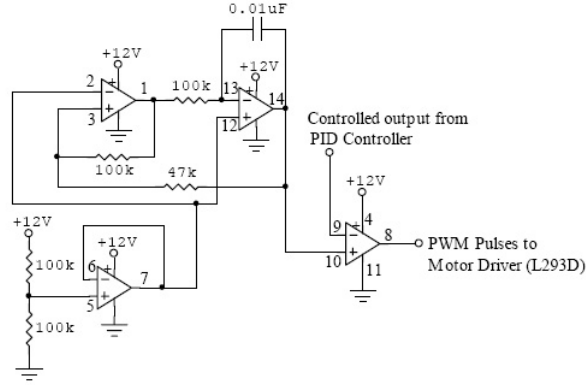


Figure 9: Schematic: PWM generator

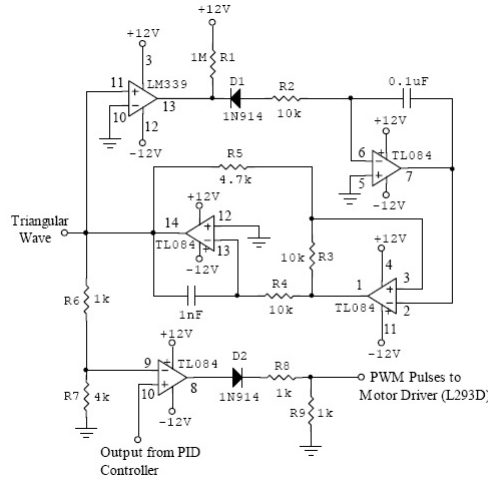


Figure 10: Schematic: Modified PWM generator

6.1 Problem faced and solution

The circuit shown in figure(9) gives significant DC offset error of 1.5 V above zero reference level at the triangular wave output (due to the integrator circuit), which caused a DC offset of 1.5 V at the PWM output. Because of this offset error motor starts responding for reference voltage above 1.5 V.

To reduce this offset we modified the circuit to bring back the triangular wave output to the zero reference level. The modified circuit schematic is shown in figure(10). The circuit uses the feedback mechanism with the comparator (LM 339) and integrator circuit to correct the DC offset error and brings back the triangular wave output to zero reference level. Because of this error correction we are able to generate the PWM pulses from 0% to 100% duty cycle after comparing this triangular wave with the controlled output from PID controller.

6.2 Motor driver

These PWM pulses are then fed to motor driver. We are using L293D as a motor driver. L293D is a push-pull four channel driver manufactured by ST Microelectronics.

L293D specifications:

1. Output current capacity per channel = 1 A
2. Internal Clamp diodes, over temperature protection
3. Supply voltage = 36 V (max)
4. Input logic voltage = 7 V (max)

7 Complete Circuit Diagram

8 Testing Results

The DC motor control system was tested with PI and PID controllers. The output waveforms of the PWM generator for two different reference voltage (corresponding to desired speed) and under full load and no load condition are shown in this section. The DC motor is loaded by magnetic breaking to study the controller performance under load conditions.

8.1 Test with PI controller

8.2 Test with PID controller

9 Discussion

The values of K_P , T_I and T_D designed in section 4, required further tuning to smooth system response. The response for PI controller is found to be as desired. When subjected to load variation the controller output would quickly set to reference point. The steady error with PI control is found to be less than 2%.

The response of the system with PID controller is not as desired. Even though the response is fast and with steady state error less than 2%, the system demonstrates oscillations. The PWM generated was not steady and oscillated around the set point. This is seen from the waveforms in section 8. Further tuning of PID control using some different algorithm other than Ziegler-Nichols Rules is required.

10 Conclusion and Future Work

The PID controller for the DC motor speed control was successfully implemented. The effect of variation in load was studied and the response is found satisfactory. It is found that for the control of a first order plant like the DC motor, the PI controller is sufficient to obtain the desired performance.

The future work will aim at designing PID using advance design algorithms. A setup can be made to record the system responses using a micro-controller and plot them on a desktop computer.

References

- [1] I.J.Nagrath and M.Gopal, Control Systems Engineering, (Wiley Eastern Limited).
- [2] William Palm III, Modelling, Analysis and Control of Dynamic Systems, 2nd edition, John Wiley and Sons.
- [3] Datasheets LM2907N, LM339 National Semiconductors.
- [4] Datasheets TL084, TL071, Texas Instruments

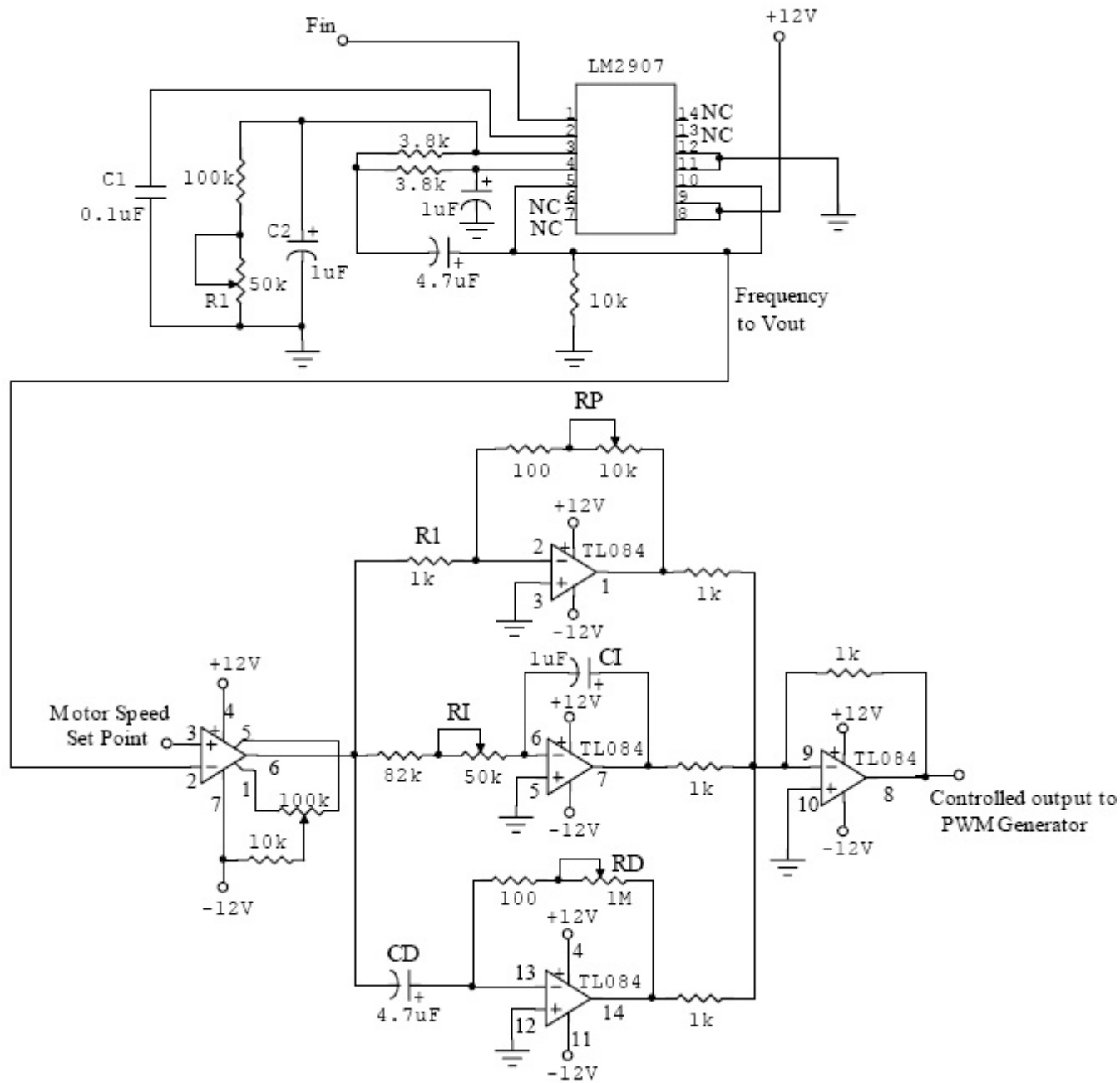


Figure 12: Schematic: Complete circuit, Part2

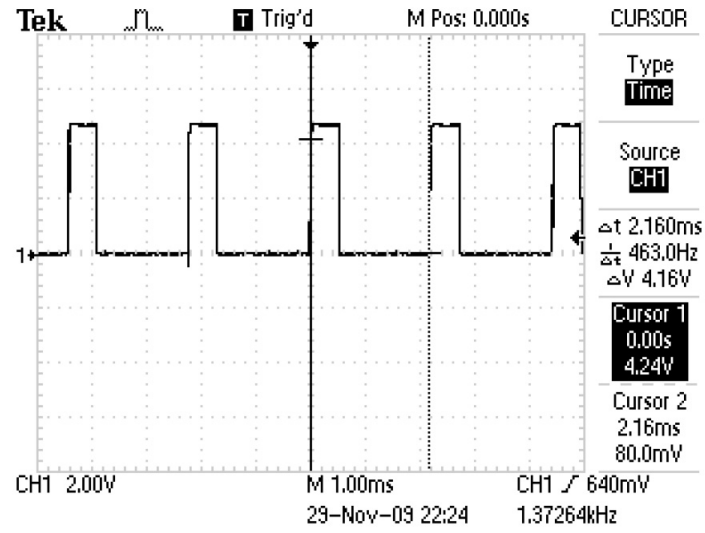


Figure 13: $V_{ref} = 1.5V$, no load condition.

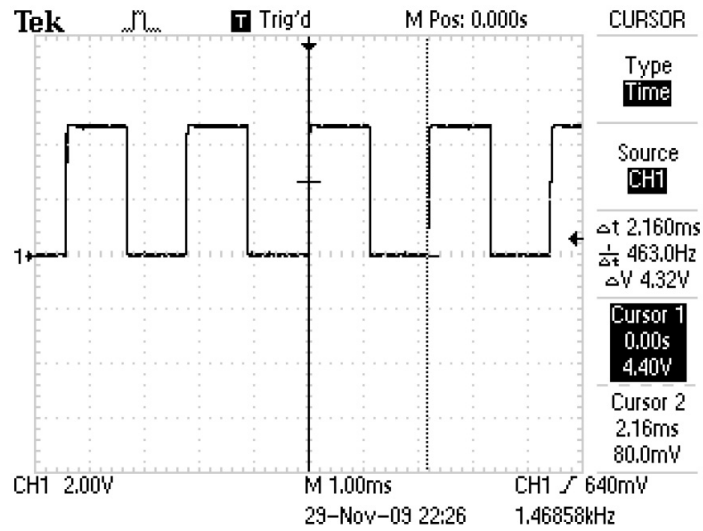


Figure 14: $V_{ref} = 1.5V$, full load condition.

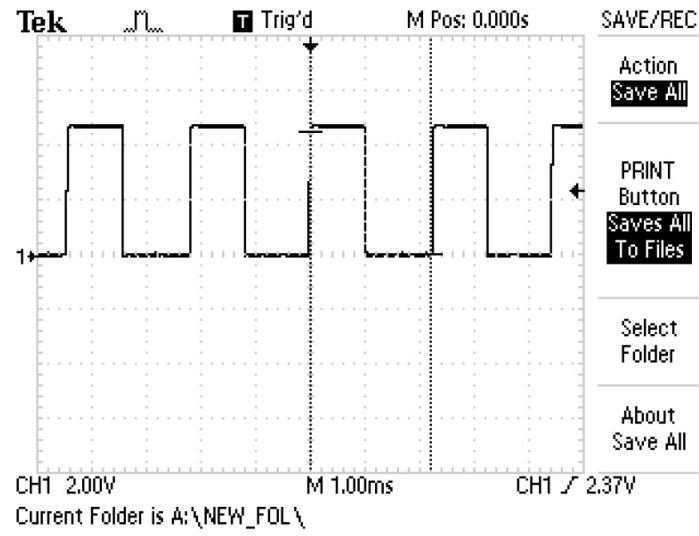


Figure 15: $V_{ref} = 2.5V$, no load condition.

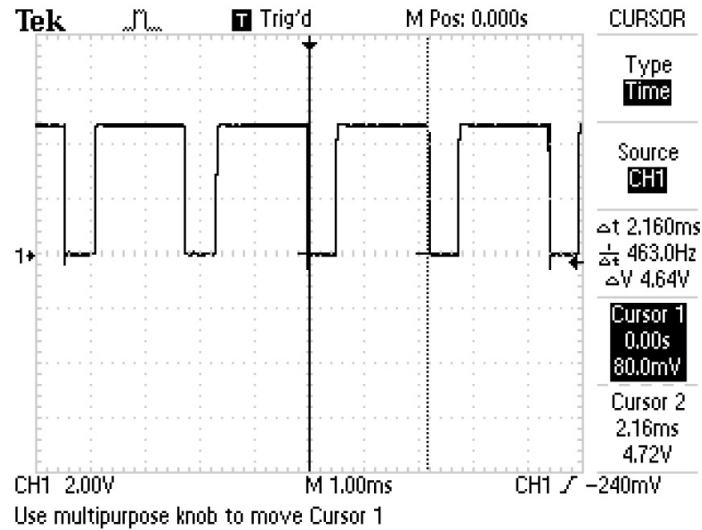


Figure 16: $V_{ref} = 2.5V$, full load condition.

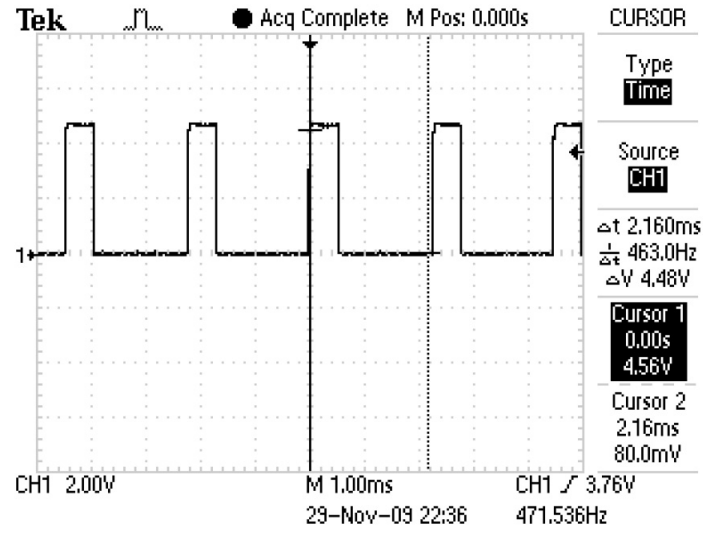


Figure 17: $V_{ref} = 1.5V$, no load condition.

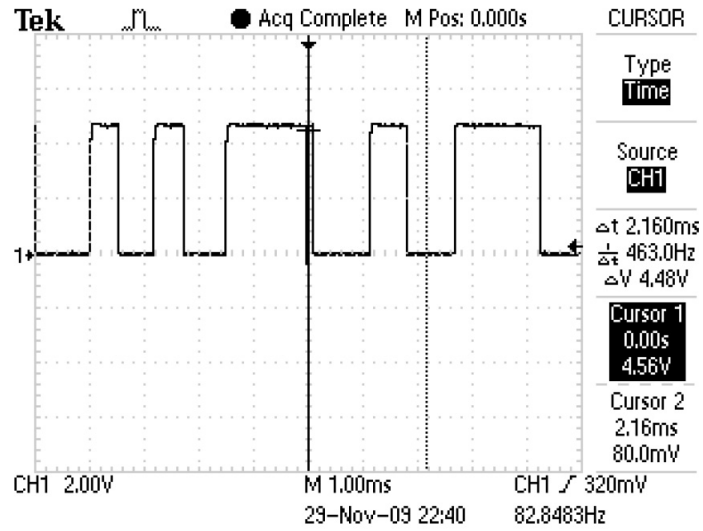


Figure 18: $V_{ref} = 1.5V$, full load condition.

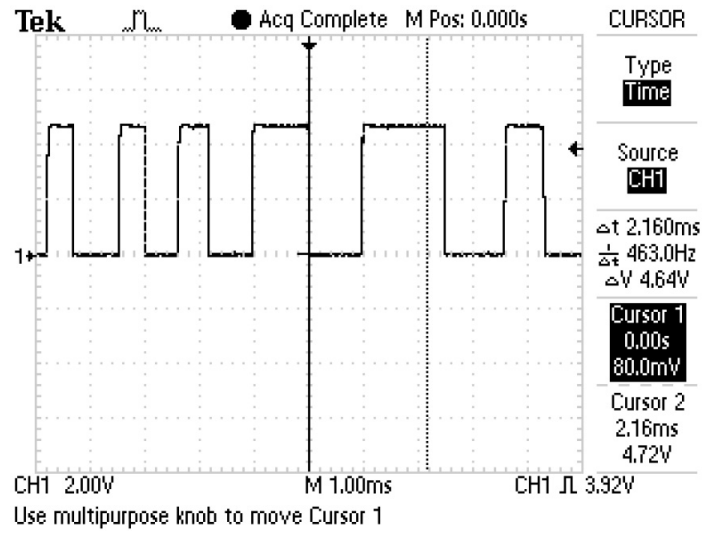


Figure 19: $V_{ref} = 2.5V$, no load condition.

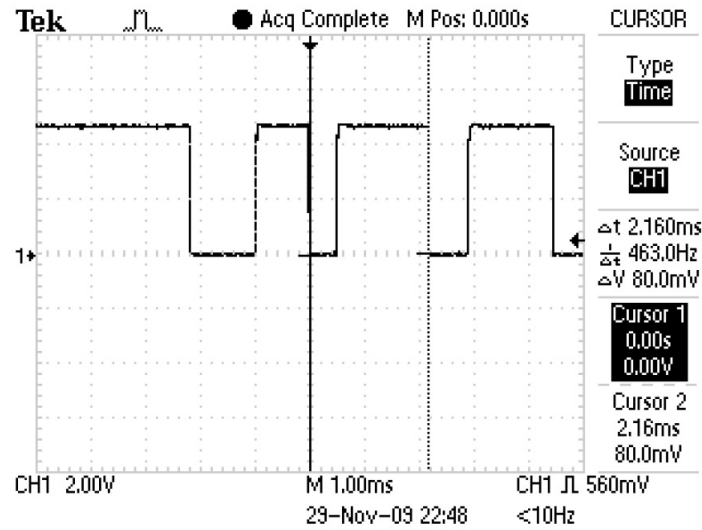


Figure 20: $V_{ref} = 2.5V$, full load condition.