

## Simulation of Direct Current (D.C) Electrical Motor Speed Control

B.I.Bakare<sup>1</sup> and Abdullah B. Tahir<sup>2</sup>

<sup>1</sup>(Department of Electrical Engineering, Rivers State University, Port Harcourt, Nigeria.)

<sup>2</sup>(Department of Mechatronics, College of Engineering, Federal University of Agriculture, Abeokuta, Nigeria.)

Corresponding Author: B.I.Bakare

**ABSTRACT :** Mathematical equations describing the voltage and corresponding rotational angular speed of the DC motor were applied and simulated using MATLAB. The desired Direct Current (DC) motor speed control is realized by the proper design and implementation of the Proportional (P) controller. Results from the analysis of DC motor speed control showed that when 1 volt was applied to the system, the motor could only achieve a maximum speed of 0.1 rad/sec, which was ten times smaller than the desired speed. For a unit-step input, the DC motor angular speed had dynamic response with settling time, steady-state error and percentage overshoot of 2.070 sec, 0.853 and 0.000% respectively. The implementation of the proportional gain  $K_p$  from proportional controller reduced the steady-state error. However, increasing  $K_p$  often resulted in increased overshoot with too large steady-state error, therefore, it appeared that not all of the design requirements could be met with a simple proportional controller.

**KEYWORDS** –Compensator, Controller, Overshoot, Settling time and Steady-State error.

Date of Submission: 28-06-2018

Date of acceptance: 13-07-2018

### I. INTRODUCTION

DC motors have excellent control characteristics. Even today, the DC motor remains industry standards as far as the control aspects are concerned [1]. In the lower power levels and also in regenerative systems the DC machines still have major relevance. More efficient motors can significantly reduce energy consumption and help mitigate environmental problems such as acid rain and global warming. Electrical drives involving various types of DC motors turn the wheel of industry. The main reason for their popularity is the ability to control their torque and flux easily and independently [2]. Therefore, DC motors are comprehensively used in various industrial applications such as electrical equipment, computer peripherals, robotic manipulators, actuators, steel rolling mills, electrical vehicles, and home appliances. Its applications spread from low horse power to the multimega watt due to its wide power, torque, speed ranges, high efficiency, fast response, and simple and continuous control characteristics [3].

The focus of control objectives is on the transient behavior of the system such as to produce a zero steady state error, a fast transient response to a step command, a short settling time and low overshoot. It is also desirable to make the system less sensitive to disturbance [4]. A control system is a mechanism which makes certain physical variables of a system, called a plant, behave in a prescribed manner, despite the presence of uncertainties and disturbances. [5] discussed that one of the principal reasons for using feedback in control systems is to reduce the effects of undesirable disturbances of various origins on the system performance. The most common disturbances of this type are unwanted inputs and variations of the static and dynamic characteristics of the system.

This paper presents the simulations of the equations describing the voltage and corresponding rotational angular speed of a DC motor using MATLAB and the realization of the desired Direct Current (DC) motor speed control by proper design and implementation of Proportional (P) controller

## II. ELECTRICAL COMPENSATORS

The alteration or adjustments of a control system in order to provide a suitable performance is called compensation. A compensator is an additional component or circuit that is inserted into control system to compensate for a deficient performance. Compensation in control systems is necessary if a given system does not conform to set design criteria. System designers can improve transient response characteristics, such as rise time, settling time, and overshoot, as well as steady state error with help of compensators such as lead, lag, fuzzy logic, neural networks, proportional, proportional-integral, proportional – derivative and proportional-integral-derivative controllers. Compensators are used to alter the response of a control system in order to accommodate set design criteria.

### 2.1 Proportional (P) Controller

Proportional controller is a controller that acts in a way that the control action is in proportion to the current error value. A pure proportional controller will not always settle at its target value, but may retain a steady-state error. Specifically, drift in the absence of control, biases a pure proportional controller and leads to DC offset error and droop. DC offset error cannot be tolerated in many systems, but where it can, the modest Proportional Controller can suffice [6].

### 2.2 Proportional Integral (PI) Controller

Proportional integral controller combines proportional and integral terms such that the integral term accelerates the movement of the process towards setpoint and eliminates the residual steady-state error that occurs with a pure proportional controller. However, since the integral term responds to accumulated errors from the past, it can cause the present value to overshoot the setpoint value.

### 2.3 Proportional Derivative (PD) Controller

[7] discussed traditionally rheostatic armature control method as widely used for the speed control of low power dc motors. However the controllability, cheapness, higher efficiency, and higher current carrying capabilities of static power converters brought a major change in the performance of electrical drives. The desired torque-speed characteristics could be achieved by the use of conventional proportional-integral-derivative (PID) controllers. As PID controllers require exact mathematical modeling, the performance of the system is questionable if there is parameter variation. The derivative term makes the rate of change of the error signal multiplied with a gain  $K_d$ . The P controller is augmented with a 'D' term to allow the higher proportional gain. The 'D' gain advances the phase of the loop by virtue of the 90 degree phase lead of a derivative. Using the 'D' gain will usually allow the system responsiveness to increase. The 'PD' controller is fast, powerful but more susceptible to stability problems, sampling irregularities, noise, and high frequency oscillations.

### 2.4 Proportional Integral Derivative (PID) Controller

One form of controller widely used in industrial process is called a three terms, or PID controller. "PID" is an acronym for "proportional, integral, and derivative." A PID controller is a controller that includes elements with those three functions. In the literature on PID controllers, acronyms are also used at the element level: the proportional element is referred to as the "P element," the integral element as the "I element," and the derivative element as the "D element." The PID controller can be understood as a controller that takes the present, the past, and the future of the error into consideration. For many years, the PID has been used for a control purposes. Due to its simplicity, robustness, a wide range of applicability and near optimal performance is some of the reasons that have made PID control so popular in the academic and industry applications [8].

Proportional- Integral- Derivative (PID) controllers along with their tuning have been widely used for speed and position control of DC motor [9]. A PID controller has three different actions and each action has specific types of goals that make the system robust performance possible. In spite of developed modern control techniques like fuzzy logic controllers or neural networks controllers, PID controllers constitute an important part of industrial control systems, attempt to correct the error between the reference and the process output, calculates error signal and rapidly adjusts the input to keep error minimal..

## III. MODELING OF A DIRECT CURRENT (DC) ELECTRICAL MOTOR

DC motor has good speed control response, wide speed control range and it is used in high control speed control systems. The electric circuit of the armature and the free body diagram of the rotor are shown in Figure. 1. The model below consists of a battery and DC motor connected to a load. The DC motor is connected to the battery with voltage source  $E_a$  through sockets which are modeled below. The electric motor used consists

of three components such as resistance  $R_a$ , inductance  $L_a$ , voltage source  $E_a$ .  $R_a$  represents the resistance of the electric coil in the motor,  $L_a$  represents the inductance of the coil while  $E_b$  is a voltage called back e.m.f (Electromotive force). The idea here is that as one spins up the load in clockwise direction, the coils on the rotor rotate in the magnetic field and as a result, as the angular velocity of the load increases then the voltage generated by back the e.m.f also increases to give DC motor top speed.

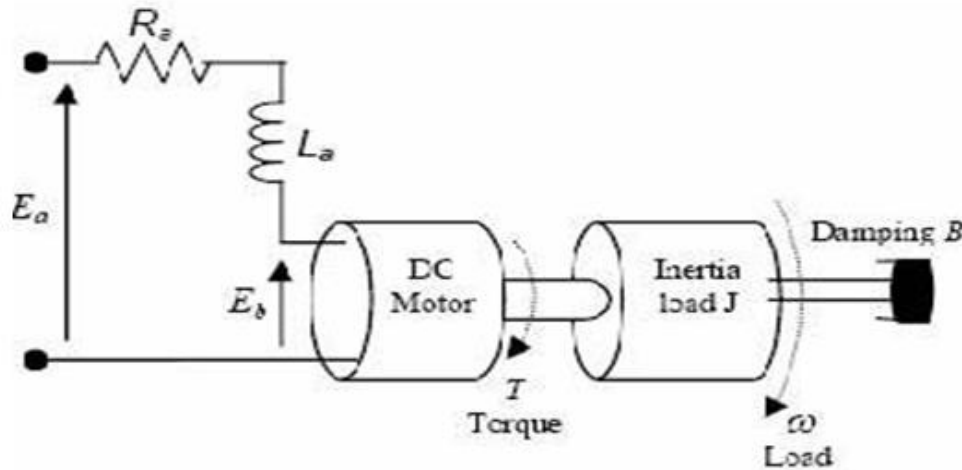


Figure 1: Pictorial representation of Model of a DC Motor

Mathematical description of DC motor:

- $\omega$  = rotational angular velocity
- $R_a$  = armature resistance
- $L_a$  = armature inductance
- $K_t$  = motor-torque constant
- $K_b$  = back emf constant
- $J$  = moment of inertia of the motor
- $I_L$  = moment of inertia of the load
- $\dot{\omega}$  = angular acceleration
- $b$  = viscous-friction coefficient of the motor
- $T_m$  = the motor torque
- $i_a$  = armature current
- $E_a$  = armature voltage
- $E_b$  = the back emf

Mathematical Representation:

The back emf,  $E_b$  is related to the rotational velocity,  $\omega = \frac{\partial \theta}{\partial t}$ , by the following equation:

$$E_b(t) = K_b \omega(t) \quad (1)$$

The motor torque,  $T_m$ , is related to the rate of change angular velocity,  $\dot{\omega}$ , by the moment of inertia of the load,  $I_L$ :

$$T_m(t) = I_L \dot{\omega}(t) \quad (2)$$

The motor torque,  $T_m$ , is related to the armature current,  $i_a$ , by a constant factor  $K_t$ :

$$T_m(t) = K_t i_a(t)$$

From mechanical part of DC motor:

$$J \frac{d\omega}{dt} + b\omega(t) = T_m(t) \quad (3)$$

The voltage loop of DC motor:

$$E_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + E_b(t) \quad (4)$$

Where

$R_a i_a(t)$  = voltage across  $R_a$

$L_a \frac{di_a(t)}{dt}$  = voltage across  $L_a$

The equations above are solved to analyze the behavior of the DC by determining  $i_a(t)$  from the diagram above where derived equations such as:

$$E_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + E_b(t) \tag{5}$$

$$E_b(t) = K_b \omega(t) \tag{6}$$

$$T_m(t) = K_t i_a(t) \tag{7}$$

$$T_m(t) = I_L \dot{\omega}(t) \tag{8}$$

Equating equations (7 and 8):

$$K_t i_a(t) = I_L \dot{\omega}(t) \tag{9}$$

Substituting equation (6) into (5):

$$E_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + K_b \omega(t) \tag{10}$$

Re-arranging equations (9 and 10), and replacing  $\dot{\omega}(t)$  with  $\frac{d\omega(t)}{dt}$ :

$$\frac{d\omega(t)}{dt} = \frac{K_t}{I_L i_a(t)} \tag{11}$$

$$\frac{di_a(t)}{dt} = \frac{1}{L_a} [E_a - R_a i_a(t) - K_b \omega(t)] \tag{12}$$

Taking the integral  $\int$  of equations (11 and 12):

$$\int \frac{d\omega(t)}{dt} = \int \frac{K_t}{I_L i_a(t)}$$

$$\int \frac{di_a(t)}{dt} = \int \frac{1}{L_a} [E_a - R_a i_a(t) - K_b \omega(t)]$$

$$\omega(t) = \int \frac{K_t}{I_L i_a(t)} \tag{13}$$

$$i_a(t) = \int \frac{1}{L_a} [E_a - R_a i_a(t) - K_b \omega(t)] \tag{14}$$

With initial conditions  $t_0$  equations (13 and 14) become:

$$\omega(t) = \int \frac{K_t}{I_L i_a(t)} + i_a(t_0) \tag{15}$$

$$i_a(t) = \int \frac{1}{L_a} [E_a - R_a i_a(t) - K_b \omega(t)] - R_a i_a(t_0) - K_b \omega(t_0) \tag{16}$$

Setting the initial condition to zero i.e.  $t_0 = 0$ , equations (15 and 16) return to equations (13 and 14), these equations are run in Simulinks digital representation of mathematical model of the Dc motor as shown to in Figure. 2, where  $\omega(t)$  and  $i_a(t)$  are output of the DC motor

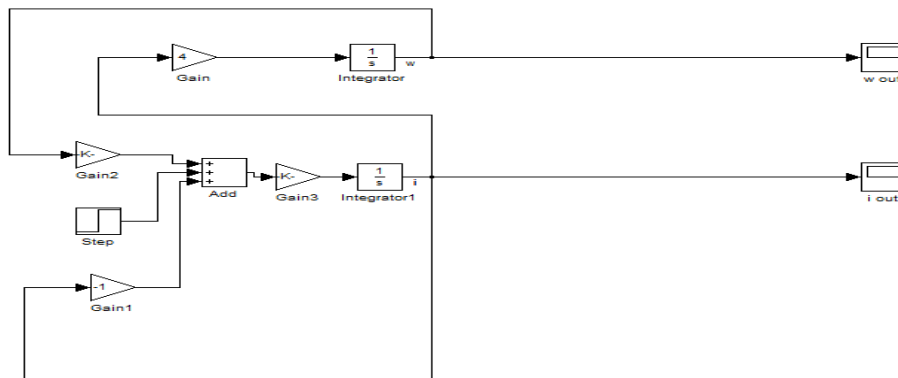


Figure 2: Model of DC motor in Simulinks

Integral actions are employed in the simulinks design of Dc motor instead of derivative representation of amature current as a result of the sensitivity of derivative gain to noise

### 3.1Laplace Transform Analysis of DC Motor

Transfer function of DC motor with load can be represented with P(s) such that

$$P(s) = \frac{Out(s)}{In(s)}$$

Out(s)= transfer function of the output of the system;

In(s)= transfer function of the input to the system.

And  $E_a(s) = In(s)$  and  $\omega(s) = Out(s)$ , where the rotational speed is considered the output and the amature voltage is considered the input.

$$P(s) = \frac{\omega(s)}{E_a(s)} \quad (17)$$

Using equations (5) and (11):

$$E_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + E_b(t)$$

$$\frac{d\omega(t)}{dt} = \frac{K_t}{I_L} i_a(t)$$

Using unilateral Laplace transform:

$$E_a(s) = R_a I_a(s) + sL_a I_a(s) + K_b \omega(s) \quad (18)$$

$$s\omega(s) = \frac{K_t}{I_L} I_a(s) \quad (19)$$

Solving for  $I_a(s)$ , equation (19) becomes:

$$I_a(s) = \frac{s\omega(s)I_L}{K_t} \quad (20)$$

Substituting equation (20) back to equation (18):

$$E_a(s) = R_a \frac{s\omega(s)I_L}{K_t} + sL_a \frac{s\omega(s)I_L}{K_t} + K_b \omega(s)$$

$$E_a(s) = \omega(s) \left[ s^2 L_a \frac{I_L}{K_t} + sR_a \frac{I_L}{K_t} + K_b \right]$$

$$P(s) = \frac{\omega(s)}{E_a(s)} = \frac{1}{\left[ s^2 L_a \frac{I_L}{K_t} + sR_a \frac{I_L}{K_t} + K_b \right]} \quad (21)$$

The response of DC motor to input signals such as impulse and step inputs can be verified using a model designed in Simulinks.

Computing impulse response of DC motor from transfer function  $P(s)$  using Simulinks in MATLAB, partial fraction and inverse Laplace transform can be deployed such that

$$P(s) = \frac{\omega(s)}{E_a(s)} = \frac{1}{\left[ s^2 \frac{L_a I_L}{K_t} + s \frac{R_a I_L}{K_t} + K_b \right]} \quad (22)$$

Partial fraction expression:

$$P(s) = \frac{R_1}{s-p_1} + \frac{R_2}{s-p_2} \quad (23)$$

Inverse Laplace Transform: (using Laplace table) equation (23) becomes

$$\frac{R_1}{s-p_1} \equiv R_1 e^{p_1 t} u(t) \quad (24)$$

Programming in MATLAB for sensitivity analysis:

**MATLAB script file 1**

```

a = [a2 a1 a0]
b = [1]
a0 = Kb
a1 = s * (Ra * IL / Kt)
a2 = s^2 * (La * IL / Kt)
[r, p, k] = residue(b, a)
>> La = 0.2H
>> Ra = 1Ω
>> IL = 0.005kgm^2
    
```

```
>> Kt = 0.02N*m/A
```

```
>> Kb = 0.22V/rad/s, where r, p and k respectively represent zero, pole and gain of the transfer function that represents the DC model.
```

```
In MATLAB environment, >> [r, p, k] = residue(b, a) results to:
```

```
r = -7.3521 or 7.3521
```

```
p = -3.8601 or -1.1399; such that equations (23) becomes:
```

$$P(s) = \frac{-7.3521}{s - (-3.8601)} + \frac{7.3521}{s - (-1.1399)}$$

Using the inverse Laplace transform,

$$p(t) = -7.3521e^{-3.8601t}u(t) + 7.3521e^{-1.1399t}u(t)$$

Output of DC motor in response to input of unit step function can be computed using equation (21)

$$P(s) = \frac{\omega(s)}{E_a(s)} = \frac{1}{[s^2 \frac{L_a I_L}{K_t} + s \frac{R_a I_L}{K_t} + K_b]}$$

Using  $E_a(t) = 10vu(t)$  at zero initial condition,  $E_a(s) = \frac{10}{s}$  with Inverse Laplace Transform and  $P(s) = \frac{\omega(s)}{E_a(s)}$  in equation (21)  $\omega(s) = P(s)E_a(s)$  such that

$$\omega(s) = \frac{10}{[s^2 \frac{L_a I_L}{K_t} + s \frac{R_a I_L}{K_t} + K_b]} \quad (25)$$

Time-response of the motor with Inverse Laplace can be solved using partial fraction expression such as

$$\omega(s) \equiv \frac{R_1}{S - P_1} + \frac{R_2}{S - P_2} + \frac{R_3}{S - P_3}$$

### MATLAB script file 2

$$R_1 = 19.0463, R_2 = -64.5008, R_3 = 45.4545$$

$$P_1 = -3.8601, P_2 = -1.1399, P_3 = 0$$

$$\omega(s) = \frac{19.0463}{s+3.8601} - \frac{64.5008}{s+1.1399} + \frac{45.4545}{s}$$

With Inverse Laplace:

$$\omega(t) = 19.0463e^{-3.8610t}u(t) - 64.5008e^{-1.1399t}u(t) + 45.4545u(t)$$

>> plot (t,ω) results to a graph representing the output of DC motor in response to input of unit step function.

## 3.2 Simulink control design

Simulink control design helps control systems designers to analyze and tune control systems modeled in Simulinks. Tuning of PID controller could be automatic done in Simulinks, arbitrary controller architectures and linearization of modules to study modules behavior around specific operating system can be achieved. A PID controller may be tuned with aid of PID tuners that automatically linearize models in Simulinks, the PID gains can be calculated and responses can be easily plotted. This design can be tuned interactively and tested in non-linear simulation. Interactive techniques such as root locus and bode plot or automated tuning for more complex systems. Dynamic characteristics and the stability of the control system can be analyzed by linearizing around operating points. Linear models can be examined in time and frequency domains and imported to MATLAB environment for simulation. Simulation based frequency responses can be carried out on models that cannot be easily linearized.

### 3.2.1 Design objective

The objective is to design a PID controller that will improve the effect of parameter variations on DC motor speed control. For the exploration of the effect of parameter variations on DC motor speed control, the PID controller is designed to achieve a fast response to the step input, an improved steady-state error, and a reduction of the effect of disturbance on the DC motor.

For a unit step input with desired response of 1-rad/sec, the requirements of the design can be chosen such that:

- Steady – state error < 0.01
- Overshoot < 0.05
- Settling time < 2s[10]

$P(s)$  represents the transfer function that denotes the input –output behaviour of DC motor with mechanical and electrical parameters as given in Figure. 1.

From equation (22) and its Inverse Laplace Transform:

$$P(s) = \frac{\omega(s)}{E_a(s)} = \frac{K}{(Js+b)(sL_a+R_a)+K^2} \text{ assuming } K_t = K_b \equiv K$$

$$P(s) = \frac{\omega(s)}{E_a(s)} = \frac{K}{(Js+b)(sL_a+R_a)+K^2} \left( \frac{\text{rad/sec}}{v} \right) \quad (26)$$

The structure of the DC motor has the form shown in the fig. 2.

### MATLAB script file 3

>>  $R_a = 1;$

>>  $L_a = 0.2;$

```
>> K = 0.01;
```

```
>> J = 0.01;
>> b = 0.1;
>> s = tf('s');
```

$$\gg P\_motor = \frac{K}{((J * s + b) * (L_a * s + R_a) + K^2)}$$

```
>>ltiview ('Step', P_motor, 0:0.1:5)
```

```
>>grid
```

```
>>title ('Step Response of DC Motor Speed')
```

DC motor is simulated and the problems (a slow response to the step-input, large steady-state error, rise time, and settling time) with uncompensated DC motor are shown through the output response of the system discussed in IV of this Paper.

### 3.2.2 Proportional control

The Proportional, or, "P", controller as shown in Figure 3 is the most basic controller whose control is directly proportional to error,  $e(t)$ . The proportional term makes the current error signal multiplied with a gain  $K_p$ . The result will be the output signal such as

$$u(t) = K_p e(t) \quad (27a)$$

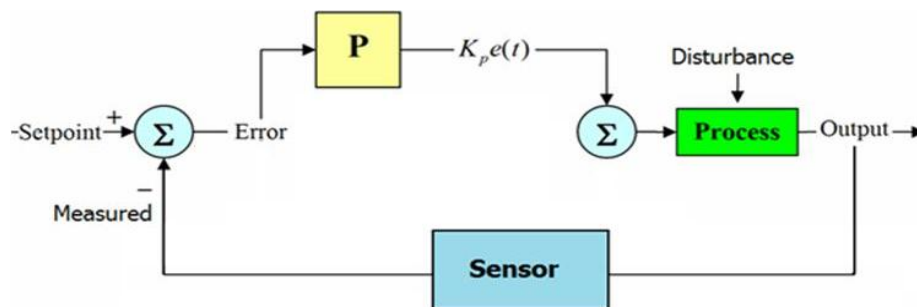


Figure 3: Block diagram of the Proportional Controller.

Improvement in the performance of open – loop transfer function of DC motor simulated above can be achieved using proportional controller (Figure. 3) through a close- loop configuration.

Taking the Inverse Laplace of equation (27a) such that in Simulinks:

$$\gg U(s) = K_p e(s)$$

Employing the proportional controller ( $K_p$ ) with a gain of 100 such that  $U(s) = 100$  makes a compensated DC motor a feedback system.

#### MATLAB script file 4

```
>> Kp = 100;
>> U = pidKp;
>> sys_cl = feedback (U*P_motor, 1);
>> t = 0: 0.01: 5 ;
>> step (sys_cl, t)
>> grid
>>title ('Step Response of DC Motor Speed with Proportional Control')
```

Proportional control gain can be varied in 'Compensator Editor' through sisotool command such that the function of the codes created in M-file above can be expanded by adding the command sisotool (P\_motor) to generate various responses to varying values of  $K_p$ .

```
>> Kp = 100;
>> U = pidKp;
```

```
>>sys_cl = feedback (U*P_motor, 1);
```

```
>> t = 0: 0.01: 5 ;
```



```
>>step (sys_cl, t)
>>grid
>>title ('Step Response of DC Motor Speed with Proportional Control')
>>sisotool(P_motor)
```

IV. RESULTS AND DISCUSSIONS

4.1 System Response

The responses of the simulations of the equations (21) to (25) are indicated in this session. Figure 10 indicates the output of DC motor in response to input of unit step function  $x_i(t)$ ;  $X_i(s)$  when the transfer function  $P(s)$  in equation (3.21) representing the input-output behavior of a DC motor (Figure 4) is modeled in Simulinks / MATLAB.

The *finalvalue* = 0.0999 indicates that steady state error is much too large and the *settlingtime* = 2.07 appears a little slow as shown in Figure 4.

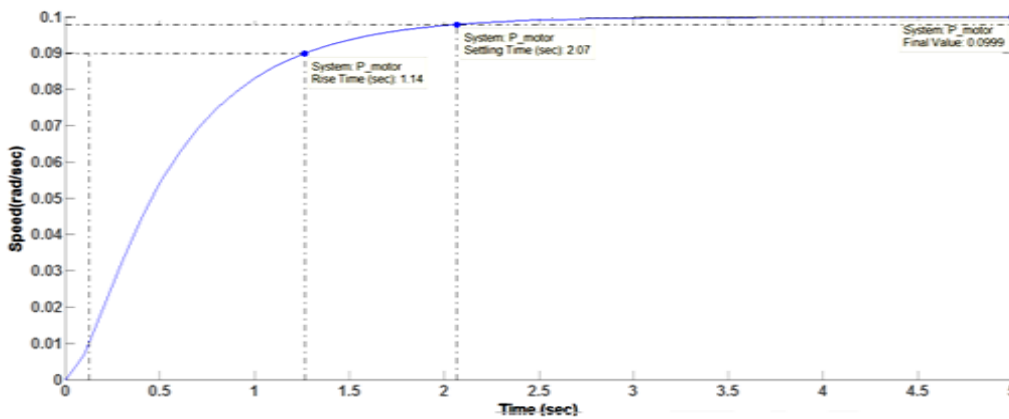


Figure 4: Step Response of DC Motor

From Figure 4 it shows that when 1 volt was applied to the system, the motor could only achieve a maximum speed of 0.1 rad/sec, ten times smaller than the desired speed. Also, it took the motor 2.07 seconds to reach its steady-state speed; this did not satisfy the 2second settling time criterion. Examination of the above shows that the steady-state error was much too large and the settle time might be a little slow.

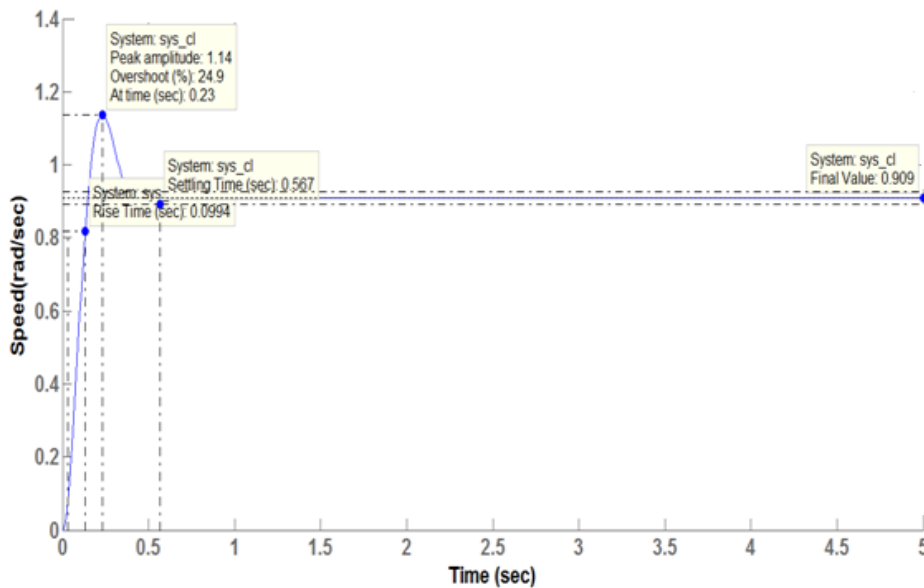


Figure 5: Step Response of DC Motor Speed with Proportional Controller



Figure 5 shows the step response of DC motor speed with Proportional Controller. Proportional control gain is varied through sisotool command such that the function of the codes created in M-file for open-loop step response of DC motor is expanded by adding the command `sisotool` to generate various responses to varying values of  $K_p$ . Improvement in the performance of open – loop transfer function of DC motor whose output is shown in Figure 5, is achieved using proportional controller with step response of *peakamplitude* (as the amount of time the system takes to reach the systems maximum value) = 1.14, *risetime* = 0.0994 and *settlingtime* = 0.587 as against *risetime* = 1.14 and *settlingtime* = 2.07 for step response of DC motor where no compensator is used. From the Figure 5 the implementation of the proportional gain  $K_p$  from proportional controller reduced the steady-state error. However, increasing  $K_p$  often resulted in increased overshoot with too large steady-state error, therefore, it appeared that not all of the design requirements could be met with a simple proportional controller.

## V. CONCLUSIONS

The Mathematical equations describing the voltage and corresponding rotational angular speed of the DC motor were applied and simulated using MATLAB. The desired Direct Current (DC) motor speed control was realized by the proper design and implementation of the Proportional (P) controller. Results from the analysis of the DC motor speed control showed that when 1 volt was applied to the system, the motor could only achieve a maximum speed of 0.1 rad/sec, which was ten times smaller than the desired speed. For a unit-step input, the DC motor angular speed had dynamic response with settling time, steady-state error and percentage overshoot of 2.070 sec, 0.853 and 0.000% respectively. The implementation of the proportional gain  $K_p$  from proportional controller reduced the steady-state error. However, increasing  $K_p$  often resulted in increased overshoot with too large steady-state error, therefore, it appeared that not all of the design requirements could be met with a simple proportional controller.

## REFERENCES

- [1]. Moleykutty, G. (2008). Speed control of separately excited DC motor. Journal of Applied Sciences. American Journal of Applied Sciences 5 (3): 227-233.
- [2]. Weiyao, L., and Qi. (2009). Speed control of DC motor using composite nonlinear feedback control. IEEE International Conference on Control and Automation Christchurch, New Zealand, 2160-2164.
- [3]. Chapman, J. (2000). Electric machinery fundamentals, 3rd edition, WCB/McGraw-Hill, New York.
- [4]. Nizam, M.K., and Sahazati, M. (2008). Simulink implementation of digital cascade control DC motor model- a didactic approach. 2nd IEEE International Conference on Power and Energy (PECon 08), Johor Baha,ru, Malaysia, 1 (1): 1043-1048.
- [5]. Thorgeir, P. 2014. Parameter uncertainties in control system design. Massachusetts Inst. of Tech.,Cambridge. Measurement Systems Lab
- [6]. LJubiša, D., Novak, N., Vojislav, F., and Dragan, P. (2013). Design of PI controllers for hydraulic control systems. Mathematical problems in Engineering, <http://dx.doi.org/10.1155/2013/451312>: 1-10.
- [7]. George, M., (2008). Direct current electric motors (Control) Switching circuits (Control) Torque (Measurement). American Journal of Applied Sciences, 5 (3): 1-5.
- [8]. Cominos, P., and Munro, N. (2002). PID controllers: recent tuning methods and design to specification. IEEE Proceedings Control Theory and Applications, 149(1): 46-53.
- [9]. Farhad, E. A., Birendra, E. K., Ram, S. C., and Gopal, K. C. (2013). A Comparative Analysis of Controllers Controlling Uncertainty in the Form of 2nd Order Load, Affecting the Robust Position Control of DC Motor. International Journal of Soft Computing and Engineering (IJSCE), 3(1): 30-33.
- [10]. James, O., and Arazi, N. (2008). Modeling of speed control in a DC motor using Proportional Integral and Derivative controller. Int. Journal of Control, 14(1): 1137-1146