Experiment-2

Objective: Modeling of PID controller-based d.c. Motor in Simulink.

Equipment/Software: Matlab-Simulink.

Theory:

This experiment will illustrate the modeling of separately excited d.c. motor without a controller in cases 1 and 2, and a PID controller will be designed to control the speed of the motor in case 3 using Simulink.
Two methods are mainly used to control the speed of d.c. motor, which are briefly discussed below.

A. Voltage control method

   By varying input voltage, the speed of separately excited d.c. motor can be controlled as expressed in the relation given below. ϕ (Flux/pole) is kept constant in this method.

$$\omega = K_e(V_a - I_a R); \quad back\ emf = V_a - I_a R \ and\ K_e = \frac{K}{\phi}$$

B. Field current control

   By decreasing/increasing the flux, the speed can be increased/decreased and vice versa. Input voltage is kept constant while flux is varied.

$$\omega = \frac{V_b}{\phi}$$

The Voltage control method will be used to control d.c. motor in this experiment.

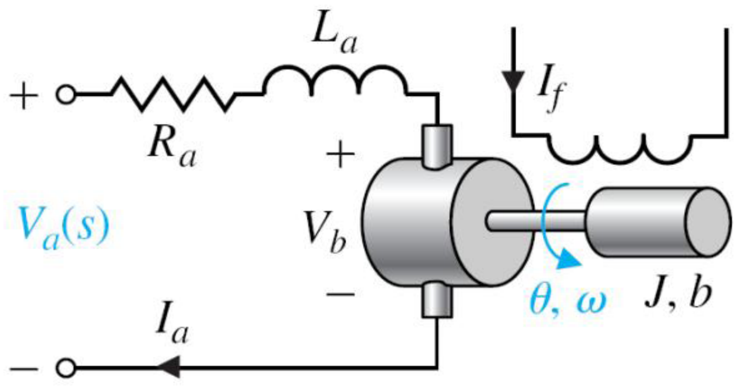

Figure 1. Circuit diagram of DC motor armature-controlled rotational actuator.

CASE 1: Modeling of DC motor in Simulink

1. Mathematical model of DC motor

Electromagnetic torque: $T = K_t i$ (1)

Back Emf $V_b = K_e \frac{d\theta}{dt}$ (2)

Motor shaft angular position, $\int\int \frac{d^2\theta}{dt^2} = \theta$ (3)

Armature current $\int \frac{di}{dt} = i$ (4)

Equation of Motion

$L\frac{di}{dt} = -Ri + V - V_b \Rightarrow \frac{di}{dt} = \frac{1}{L}(-Ri + V - K_e \frac{d\theta}{dt})$ (5)

$J\frac{d^2\theta}{dt^2} = T - B\frac{d\theta}{dt} \Rightarrow \frac{d^2\theta}{dt^2} = \frac{1}{J}(T - B\frac{d\theta}{dt})$ (6)

J=Moment of inertia, B=Friction constant, Ke=Back-emf proportion constant, Kt=Torque constant, V =Input voltage.

2. Steps to design the mathematical model of the DC motor using Simulink blocks

- Choose the number of integrators equal to independent variables of the plant called ‘state variables’.
- Select and place blocks required for design from the Simulink library, i.e., add, subtract, gain, integrator, step, scope, etc.
- Run the model and see the step response.

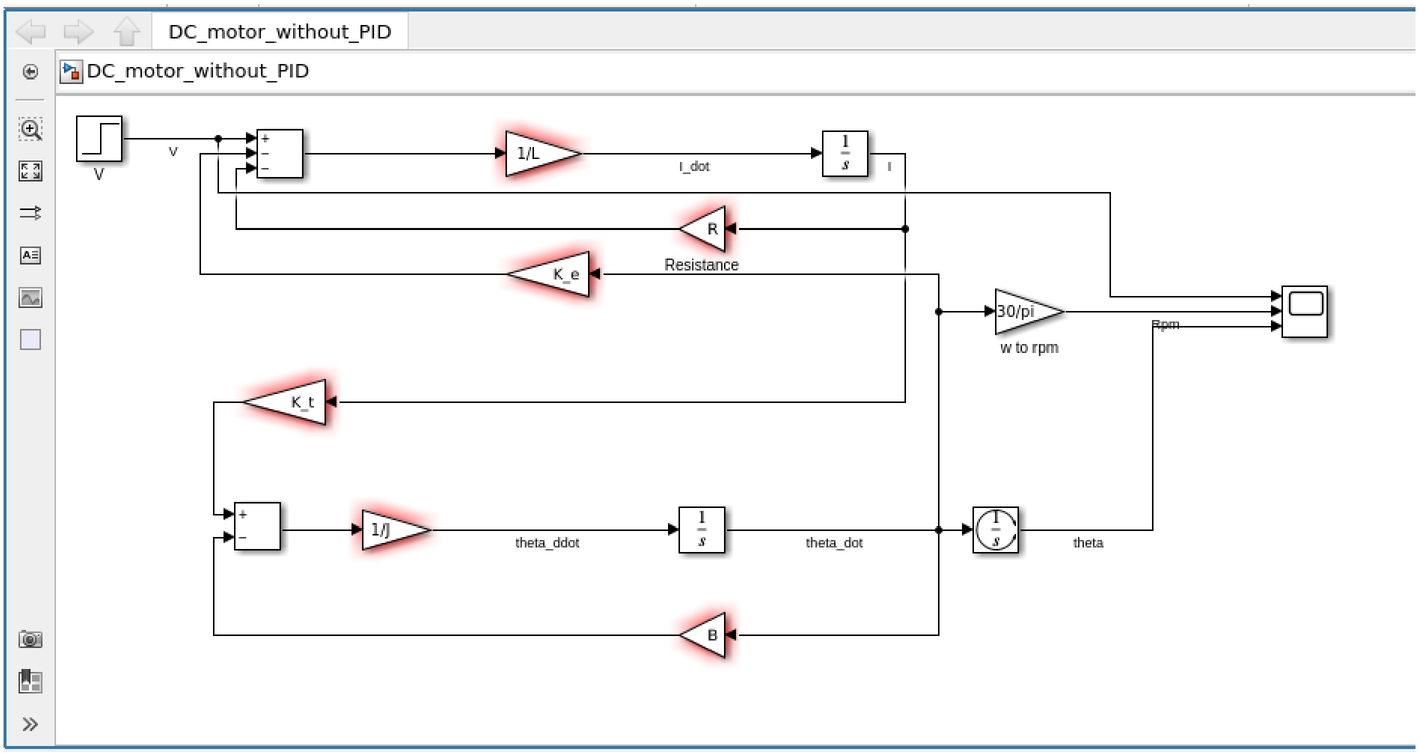

Figure 2. Final block diagram of modeling of DC motor without PID controller

CASE 2: Modeling of DC motor in Simulink using transfer function

Using Laplace transform, the model represented by equations (1) to (6) can be modeled as shown below.

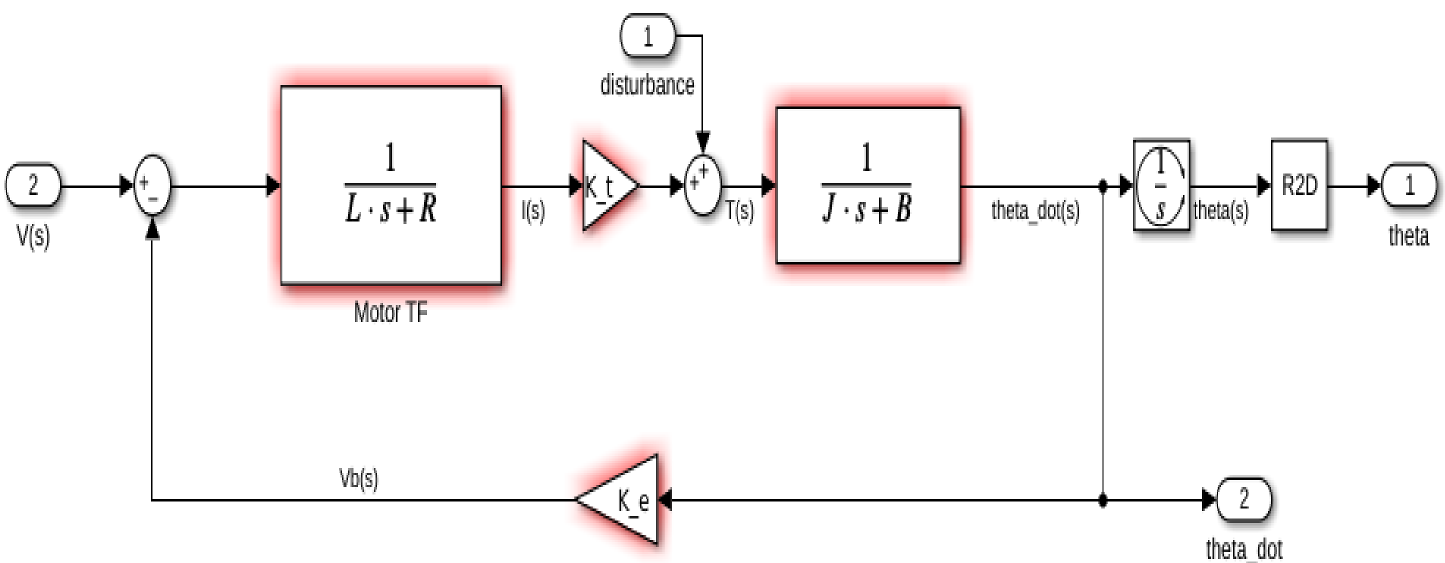

Figure 3. Plant

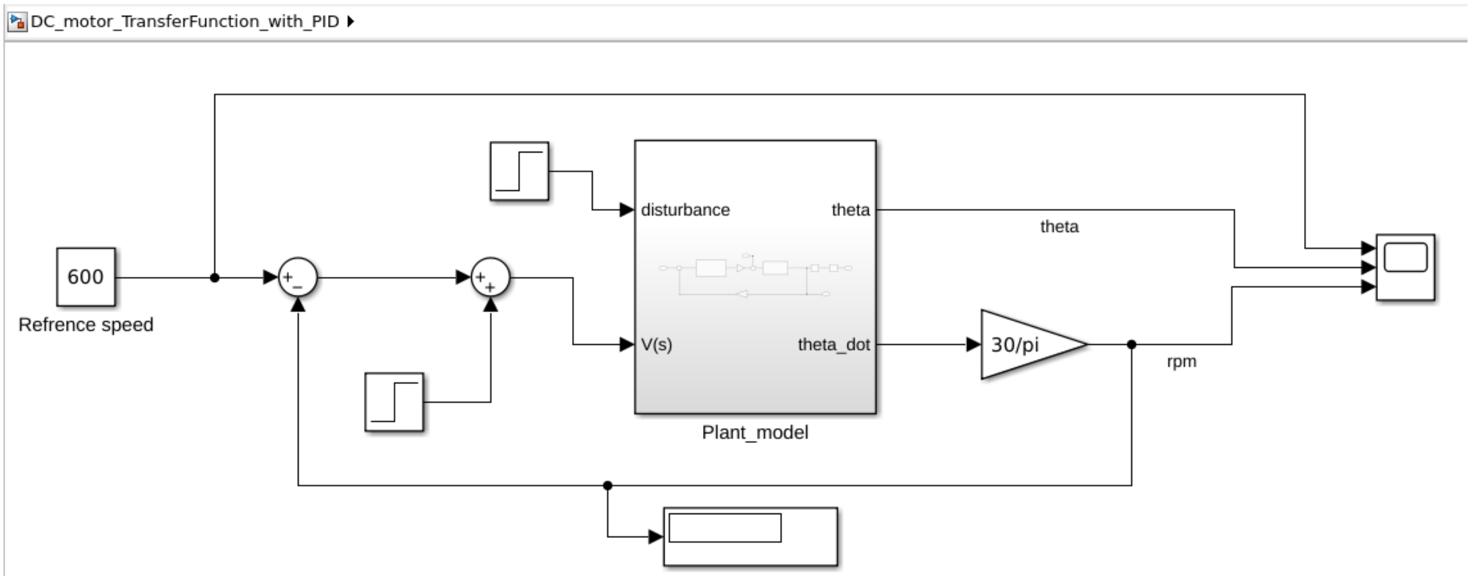

Figure 4. Plant without controller

In response figure 5:-
red color shows: Feedback speed/Measure speed
Yellow color shows: Desired speed
Blue color shows: Shaft position

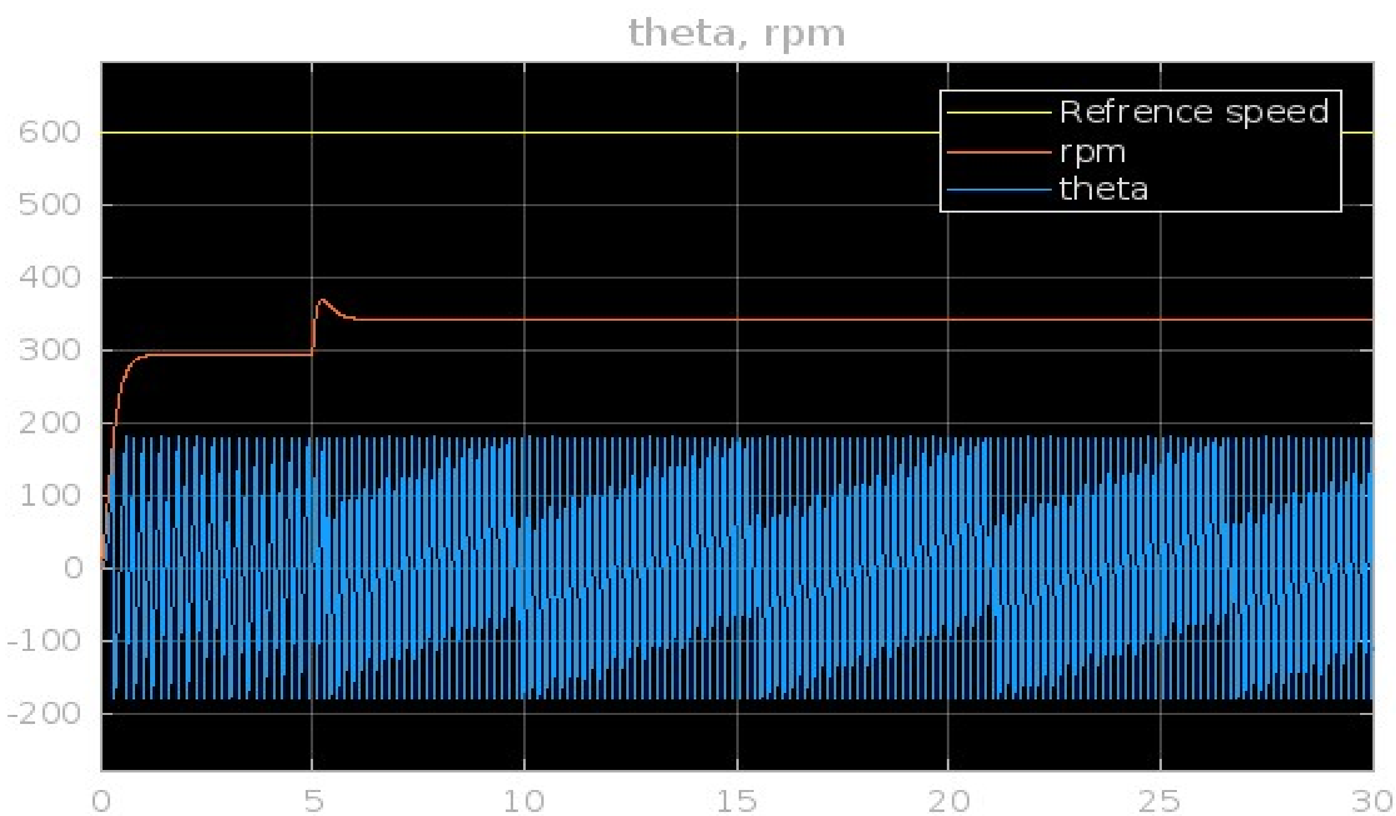

Figure 5. Response of plant

CASE 3: Block diagram of the armature-controlled DC motor with PID controller

A proportional-integral-derivative controller (PID controller) is a control loop feedback mechanism (controller) widely used in industrial control systems.

1) The proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error (e = desired - feedback) by a constant Kp, called the proportional gain constant.

$$e_p = K_p e$$

2) The integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously.

$$e_i = K_i \int e\, dt$$

3) The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain Kd. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, Kd.

$$e_d = K_d \frac{d}{dt}(e)$$

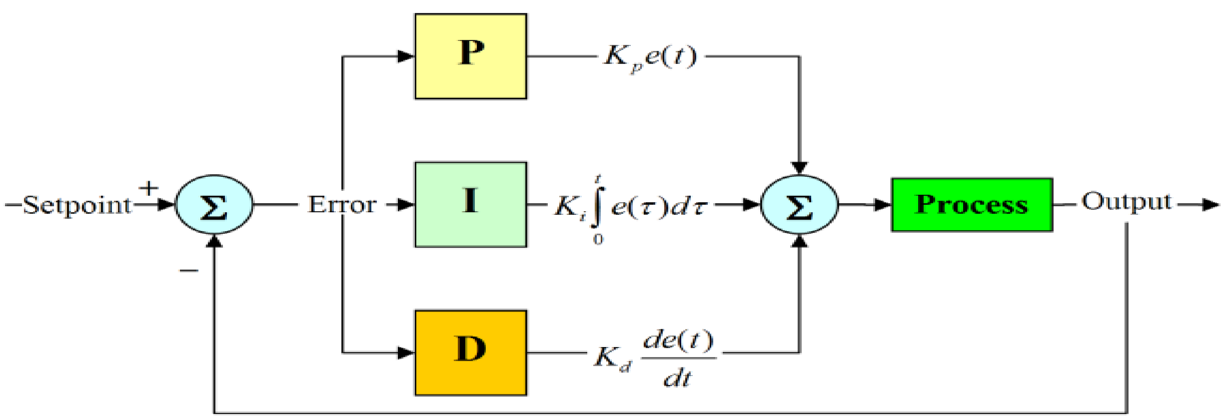

Figure 6. Basic block diagram of PID controller

A PID controller is placed with the plant as shown in figure (7), and it is auto-tuned with PID gain values as shown in figure (8). Response time, percentage overshoot, settling time, rise time, etc., can be calculated according to our desired response.

A step with magnitude 1 is added with errors of voltage and torque after 1 second and 5 seconds, respectively, shown in figure (7,9).

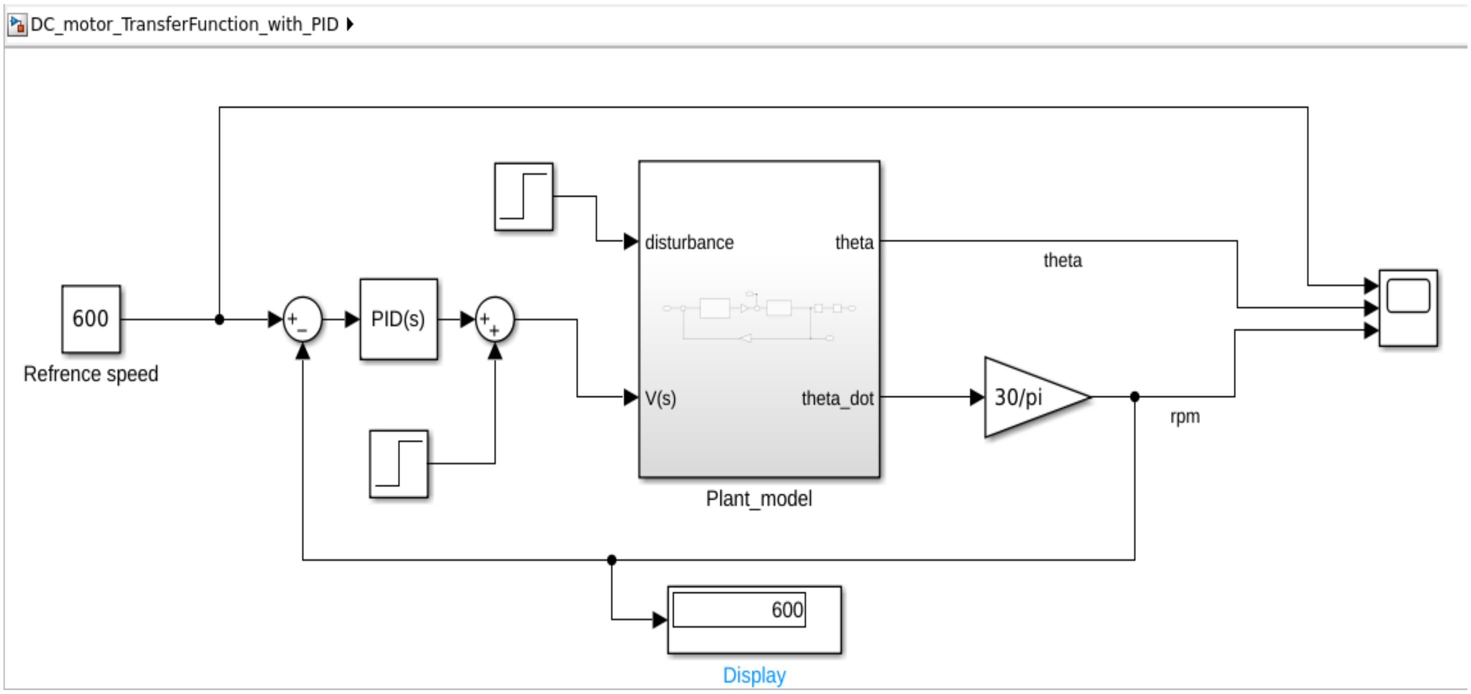

Figure 7. Plant with PID controller

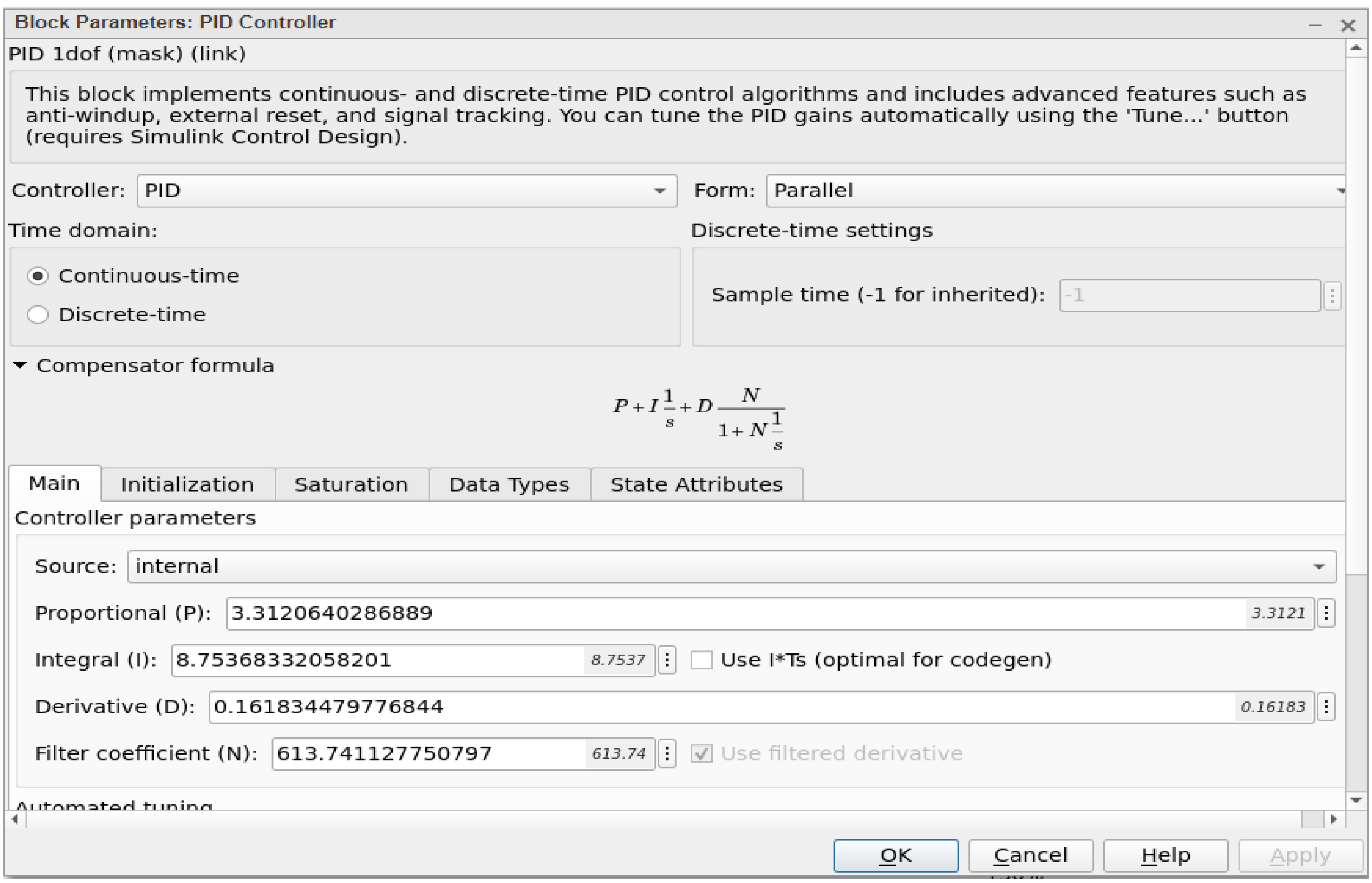

Figure 8. Auto-tuned PID controller values

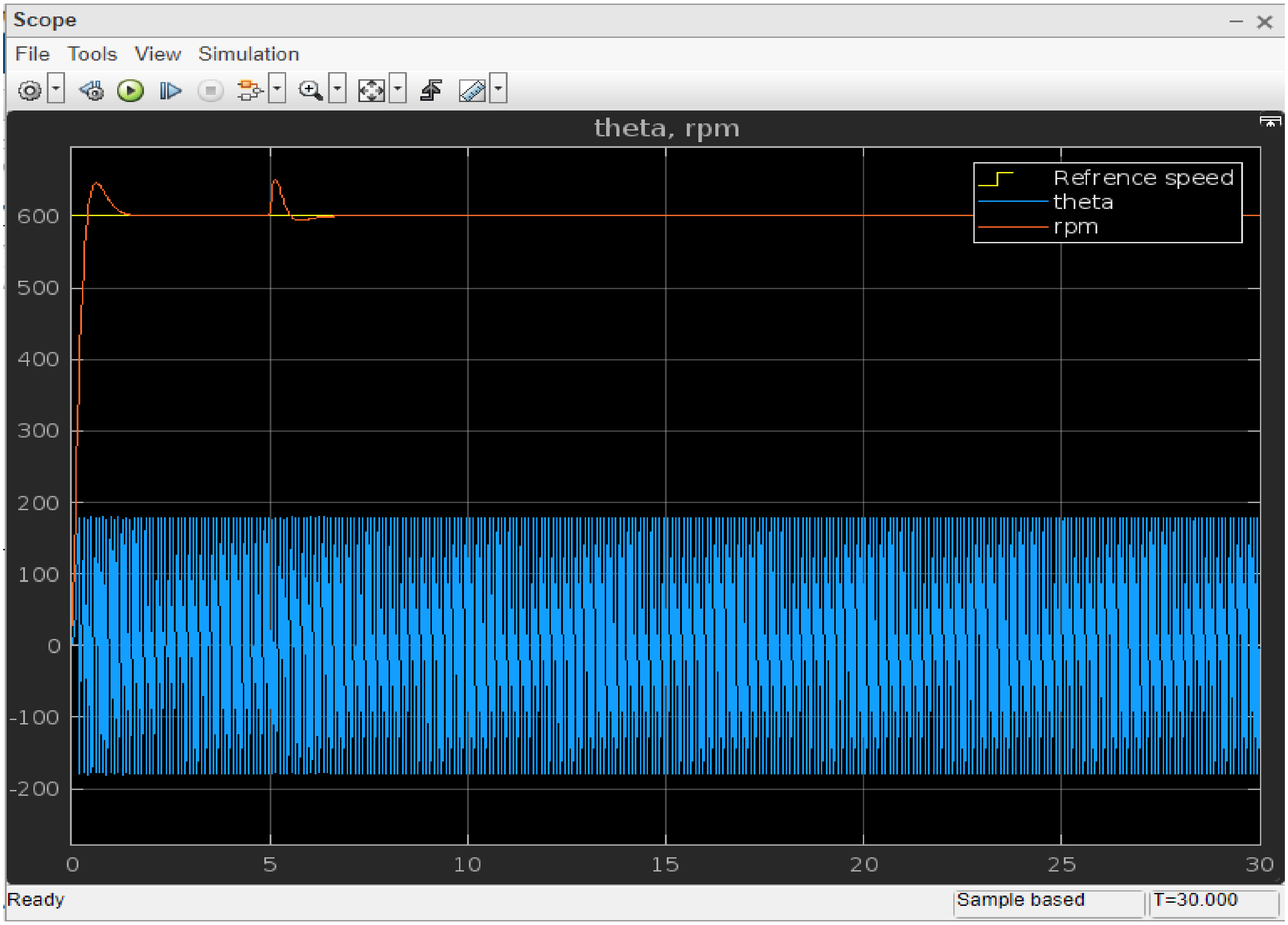

Figure 9. Results

Parameters for Dc Motor

(J)	Moment of inertia of the rotor	0.01 kg.m^2
(B)	Motor viscous friction constant	0.1 N.m.s
(Ke)	Electromotive force constant	0.01 V/rad/sec
(Kt)	Motor torque constant	0.01 N.m/Amp
(R)	Electric resistance	1 Ohm
(L)	Electric inductance	0.5 H

ASSIGNMENTS

1. Double Mass-Spring-Damper used to model Suspension System

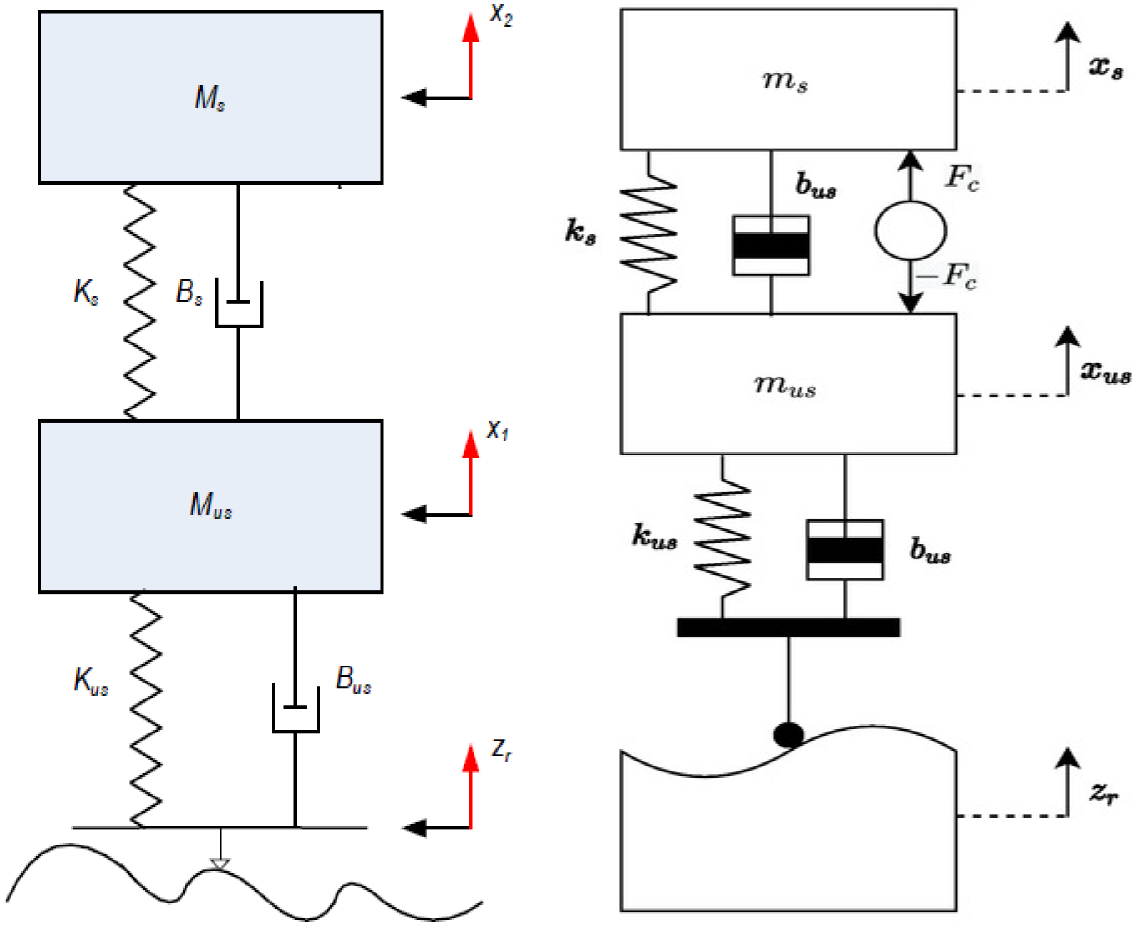

Figure 10. Passive suspension of the car (left) Active Suspension (right)

1. Model the system in simulink for the following responses and realize the system for 2.5 sec,
   - Step Response
   - Impulse Response with a delay of 0.5 sec and impulse time of 0.1 sec
   - Sinosodial response with amplitude of 0.25 and frequency of 1 Hz.

Hint: For the passive suspension model, the Transfer function of car body displacement to the road surface, and tire displacement to the road surface is, $\frac{X_2(s)}{Z_r(s)}$ *and* $\frac{X_1(s)}{Z_r(s)}$ respectively. Use the required parameters given in Table 1.

Parameters for open loop system without PID

Table 1. Parameter set for Question 1

Mass	Spring constants	Damping Constant
(Ms) = 375 kg	Ks =130000 N/m	Bs=9800 N/m/s
(Mus) = 20 kg	Kus=1000000 N/m	Bus=10000 N/m/s

2. Apply and Tune the PID controller for step-response for the parameters given in Table 2 and compare it with uncontrolled step response for Active Suspension, use the amplitude of step input as 0.1.

Parameters for open loop system with PID

Table 2. Parameter set for Question 2

Mass	Spring constants	Damping Constant
(Ms) = 2500 kg	Ks =80000 N/m	Bs=350 N/m/s
(Mus) = 320 kg	Kus=500000 N/m	Bus=15020 N/m/s

Equation of Motion Can be given By

$$M_{us}\ddot{x}_1 + (B_{us} + B_s)\,\dot{x}_1 + (K_{us} + K_s)x_1 = B_{us}\dot{z}_r + K_{us}z_r + B_s\dot{x}_2 + K_s x_2 - F_c$$

$$M_s\ddot{x}_2 + B_s\dot{x}_2 + K_s x_2 = B_s\dot{x}_1 + K_s x_1 + F_c$$

Hint:

For Open-loop response take and for passive suspension consider $F_c = 0$.

For control purpose take $x_1 - x_2 = 0$ As Reference input.