1 [5 pts total] Use the Simulink package in Matlab to find a numerical solution to the differential equation in 8.3.16; use ode45 as your solver. Since you have already solved an analytic solution to 8.3.16 in Homework 1, you can easily verify the accuracy of your Simulink solution. Plot your Simulink solution overlayed with the analytic solution. Also, show the Simulink diagram for full credit.

Recall 8.3.16

\[ 2\ddot{x} + \dot{x} - x = e^{4t}, \quad \text{where } x(0) = 3 \quad \text{and} \quad \dot{x}(0) = 0 \]

The solution (from Homework 1) is

\[
X(s) = \frac{6s^2 - 21s - 11}{(s - 4)(2s - 1)(s + 1)}
\]

\[
x(t) = \frac{1}{35}e^{4t} + \frac{40}{21}e^{0.5t} + \frac{16}{15}e^{-t}
\]

Figure 1: Plot for Simulink solution and analytic solution [2 pts]
2[15 pts total]
(a) 9.12.1 of Goodwine. You will use this Plant model for the rest of this problem. [2 pts]

$$v_{in}(t) = i(t)R + k_e \dot{\theta}(t)$$
$$\tau = k_\tau i(t)$$

Shaft has moment of inertia $J$. 
Since sum of moments equals time rate of change of angular momentum,

\[ J\ddot{\theta} = \tau \]
\[ Js^2\theta(s) = T(s) \]
\[ T(s) = k_\tau I(s) \]
\[ V_{in}(s) = I(s)R + k_c s \theta(s) \]
\[ V_{in}(s) = \frac{1}{k_\tau} JRS^2\theta(s) + k_c s \theta(s) \]
\[ V_{in}(s) = \frac{JR^2 + k_c s k_\tau}{k_\tau} \theta(s) \]
\[ \frac{\theta(s)}{V_{in}(s)} = \frac{k_\tau}{JR^2 + k_c k_\tau} \]
\[ = \frac{2}{4 \cdot 3s^2 + 1 \cdot 2s} = \frac{2}{12s^2 + 2s} = \frac{1}{6s^2 + s} \]

(b) With

\[ k_c = 1; \quad k_\tau = 2; \quad R = 3; \quad J = 4 \]

develop a PID feedback control system in Simulink. Assume that the reference signal is the unit step function; select ode45 as your solver. Show the Simulink diagram for full credit. [5 pts]

Figure 3: Simulink diagram
(c) In an *ad hoc* manner, tune the PID controller such that the step response has an overshoot < 10%, rise time < 1 second, and a settling time < 3 seconds. This semester, we’ll have plenty of time to develop a more scientific methodology for controller design. [5 pts]

![Step Response Plot](image)

**Figure 4: step response**

Matlab code for plot of step response is shown:

```matlab
% plot of step response

k_p = 50;
k_d = 20;
k_i = .05;

sim Problem_2b

figure(1)
plot(tout, r, 'k--', tout, theta, 'b-', 'LineWidth', 1.5)
xlabel ('Time [sec]')
ylabel ('Step Response [rad]')
legend ('Reference', 'Step Response')
title('k_p=50, k_i=0.05, k_d=20')
```

**Figure 5: matlab code**
(d) Write down any observations you had while tuning the PID controller. [3 pts]

decrease kp → decrease overshoot, increase rise time.
decrease kd → increase overshoot, decrease rise time.
increase ki → increase overshoot, longer setting time.