George Mason University

Implementation of PD Control Using Simulink

Report from laboratory experiment 4 conducted on 16 February 2016
As part of ECE 429 Control Systems Lab
Course Instructor: Dr. Daniel M. Lofaro

Michael Kepler
G00804828

23 February 2016

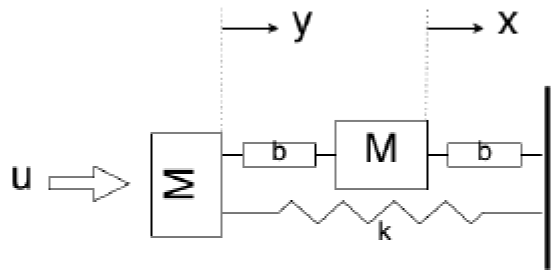
The Volgenau School of Engineering

Objective

The objective of this lab is to introduce Simulink, create models of physical systems, and to affect the behavior of those systems using proportional-derivative (PD) control.

Theoretical Background

The following diagram illustrates the physical closed-loop system to be modeled:



With the following representations:

U: Input (force)
Y: Output (displacement of the first mass)
b: Damping factor of dampers
k: spring constant

The open-loop system omits the spring, and is modeled by the following equations:

$$m\ddot{y} = -b(\dot{y} - \dot{x}) + u \quad (1)$$

$$m\ddot{x} = -b\dot{x} + b(\dot{y} - \dot{x}) \quad (2)$$

The following is the derivation of the transfer function associated with the open-loop system:

Applying the Laplace transforms to equations (1) and (2) respectively:

$$ms^2Y(s) = -bsY(s) + bsX(s) + U(s) \quad (3)$$

$$ms^2X(s) = -2bsX(s) + bsY(s) \quad (4)$$

Solving eq. (4) for $X(s)$:

$$X(s) = \left(\frac{b}{ms + 2b}\right)Y(s) \quad (5)$$

Inserting eq. (5) into eq. (3):

$$ms^2Y(s) = -bsY(s) + bs\left[\left(\frac{b}{ms + 2b}\right)Y(s)\right] + U(s) \quad (6)$$

Combining like-terms:

$$U(s) = \left((ms^2 + bs) - \left(\frac{b}{ms + 2b}\right)\right) Y(s) \tag{7}$$

Multiplying eq. (7) by $\left(\frac{ms+2b}{ms+2b}\right)$ and simplifying:

$$U(s) = \left(\frac{m^2s^3 + 3bms^2 + b^2s}{ms + 2b}\right) Y(s) \tag{8}$$

Solving for $\frac{Y(s)}{U(s)}$:

$$\frac{Y(s)}{U(s)} = \frac{ms + 2b}{m^2s^3 + 3bms^2 + b^2s} \tag{9}$$

Factoring b out of the numerator and the denominator, the transfer function $G_p(s)$ is obtained:

$$\boldsymbol{G_p(s) = \frac{Y(s)}{U(s)} = \frac{\frac{m}{b}s + 2}{\frac{m^2}{b}s^3 + 3ms^2 + bs}} \tag{10}$$

Modifying eq. (1) to represent the closed loop system:

$$m\ddot{y} = -b(\dot{y} - \dot{x}) - ky + u \tag{11}$$

The following is the derivation of the transfer function associated with the open-loop system:

Applying the Laplace transform to equation (11):

$$ms^2Y(s) = -bsY(s) + bsX(s) - kY(s) + U(s) \tag{12}$$

Inserting eq. (5) into eq. (3):

$$ms^2Y(s) = -bsY(s) + bs\left[\left(\frac{b}{ms + 2b}\right)Y(s)\right] - kY(s) + U(s) \tag{13}$$

Combining like-terms:

$$U(s) = \left((ms^2 + bs) - \left(\frac{b}{ms + 2b}\right) + k\right)Y(s) \tag{14}$$

Multiplying eq. (14) by $\left(\frac{ms+2b}{ms+2b}\right)$ and simplifying:

$$U(s) = \left(\frac{m^2s^3 + 3bms^2 + (b^2 + km)s + 2kb}{ms + 2b}\right)Y(s) \tag{15}$$

Solving for $\frac{Y(s)}{U(s)}$:

$$\frac{Y(s)}{U(s)} = \frac{ms + 2b}{m^2s^3 + 3bms^2 + (b^2 + km)s + 2kb} \tag{16}$$

Factoring b out of the numerator and the denominator, the transfer function $G_p(s)$ is obtained:

$$\boldsymbol{\frac{Y(s)}{U(s)} = \frac{\frac{m}{b}s + 2}{\frac{m^2}{b}s^3 + 3ms^2 + (b + \frac{km}{b})s + 2k}} \tag{17}$$

To verify that the feedback function $H(s)$ is equal to k, the following equation will be used:

$$\frac{Y(s)}{U(s)} = \frac{G(s)}{1 + H(s)G(s)} \tag{18}$$

Replacing $G(s)$ and $H(s)$ with eqn. (10) and k respectively:

$$\frac{Y(s)}{U(s)} = \frac{\left(\frac{\frac{m}{b}s + 2}{\frac{m^2}{b}s^3 + 3ms^2 + bs}\right)}{1 + k\left(\frac{\frac{m}{b}s + 2}{\frac{m^2}{b}s^3 + 3ms^2 + bs}\right)} \tag{19}$$

Multiplying and combining like terms:

$$\frac{Y(s)}{U(s)} = \frac{\left(\dfrac{\frac{\boldsymbol{m}}{\boldsymbol{b}}\boldsymbol{s} + \boldsymbol{2}}{\frac{\boldsymbol{m}^2}{\boldsymbol{b}}\boldsymbol{s}^3 + \boldsymbol{3ms}^2 + \boldsymbol{bs}}\right)}{\left(\dfrac{\frac{\boldsymbol{m}^2}{\boldsymbol{b}}\boldsymbol{s}^3 + \boldsymbol{3ms}^2 + \left(\boldsymbol{b} + \boldsymbol{k}\frac{\boldsymbol{m}}{\boldsymbol{b}}\right)\boldsymbol{s} + \boldsymbol{2k}}{\frac{\boldsymbol{m}^2}{\boldsymbol{b}}\boldsymbol{s}^3 + \boldsymbol{3ms}^2 + \boldsymbol{bs}}\right)} \tag{20}$$

Which simplifies to:

$$\frac{\boldsymbol{Y(s)}}{\boldsymbol{U(s)}} = \frac{\frac{\boldsymbol{m}}{\boldsymbol{b}}\boldsymbol{s} + \boldsymbol{2}}{\frac{\boldsymbol{m}^2}{\boldsymbol{b}}\boldsymbol{s}^3 + \boldsymbol{3ms}^2 + (\boldsymbol{b} + \frac{\boldsymbol{km}}{\boldsymbol{b}})\boldsymbol{s} + \boldsymbol{2k}} \tag{21}$$

Upon comparing eqn. (21) with eqn. (17), it can be seen that they are equal and thus the feedback function, $H(s)$, is equal to k.

MATLAB Implementation

The following SIMULINK models will use the following parameters:

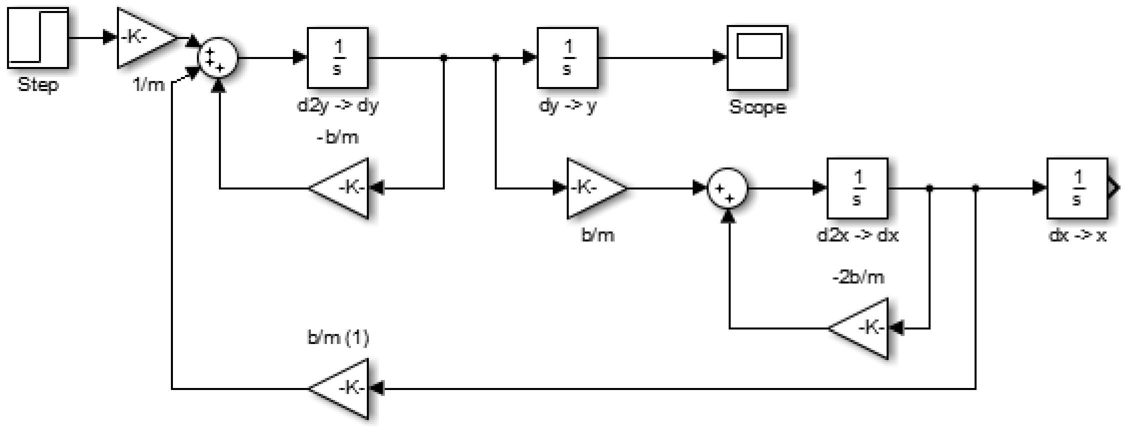
$$m = 1$$
$$b = 0.1$$
$$k = 0.5$$
$$step\ size = 0.1$$

Reorganizing eqns. (1) and (2), respectively by solving for the highest derivative:

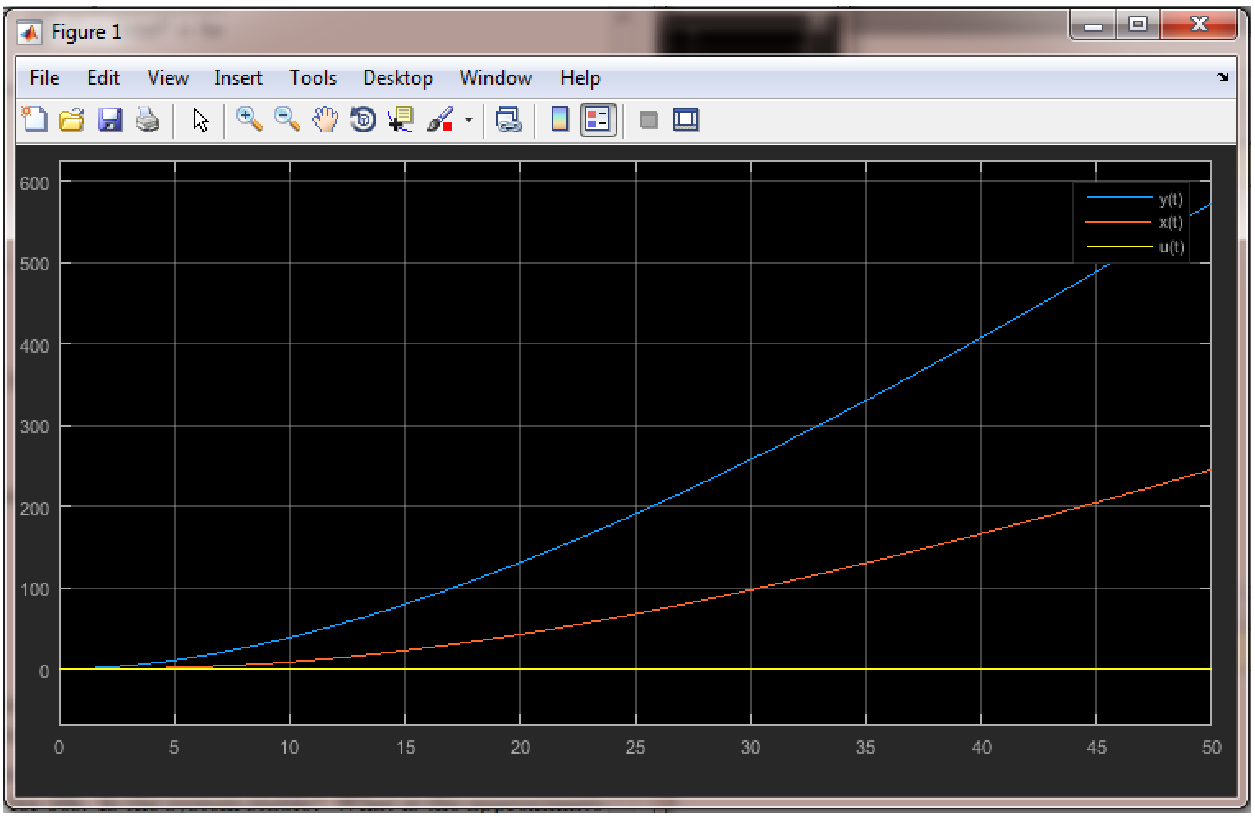
$$\ddot{y} = -\frac{b}{m}\dot{y} + \frac{b}{m}\dot{x} + \frac{1}{m}u \tag{22}$$

$$\ddot{x} = -\frac{2b}{m}\dot{x} + \frac{b}{m}\dot{y} \tag{23}$$

The following is the SIMULINK model based on the two differential equations (as explicitly instructed in the manual) above for the open-loop system:



Running the simulation for 50s with a step input of 1, as stated in the lab manual, yields the following results:

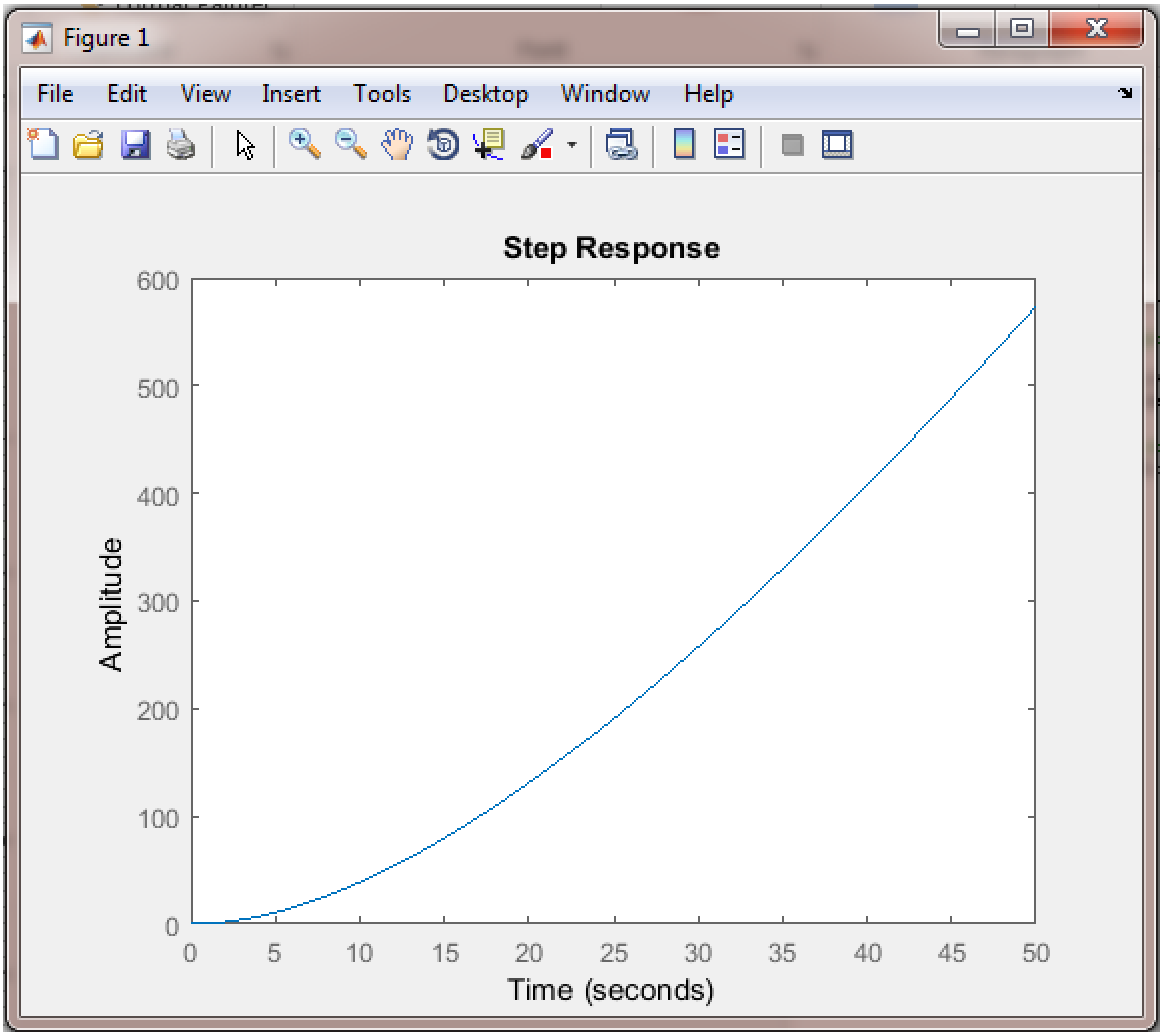


The output can be verified with MATLAB using eqn. (10) and the *step()* function:

```
%Define parameters
k = 0.5
m = 1
b = 0.1
```

```
%Create the system
num = [m/b 2]
den = [ m^2/b 3*m b 0]
sys = tf(num, den)

%Simulate the step response
step(sys,0:0.01:50)
```



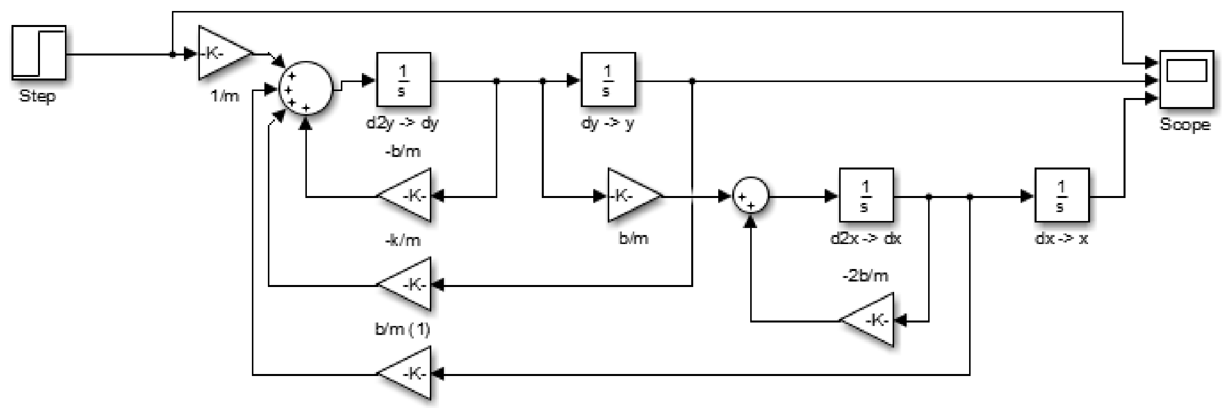
Given that the response tends to infinity as t goes infinity, the system is **not stable**.

To construct a model of the closed-loop system based on the differential equations (as explicitly instructed in the manual), begin by reorganizing eqn. (11), by solving for the highest derivative:

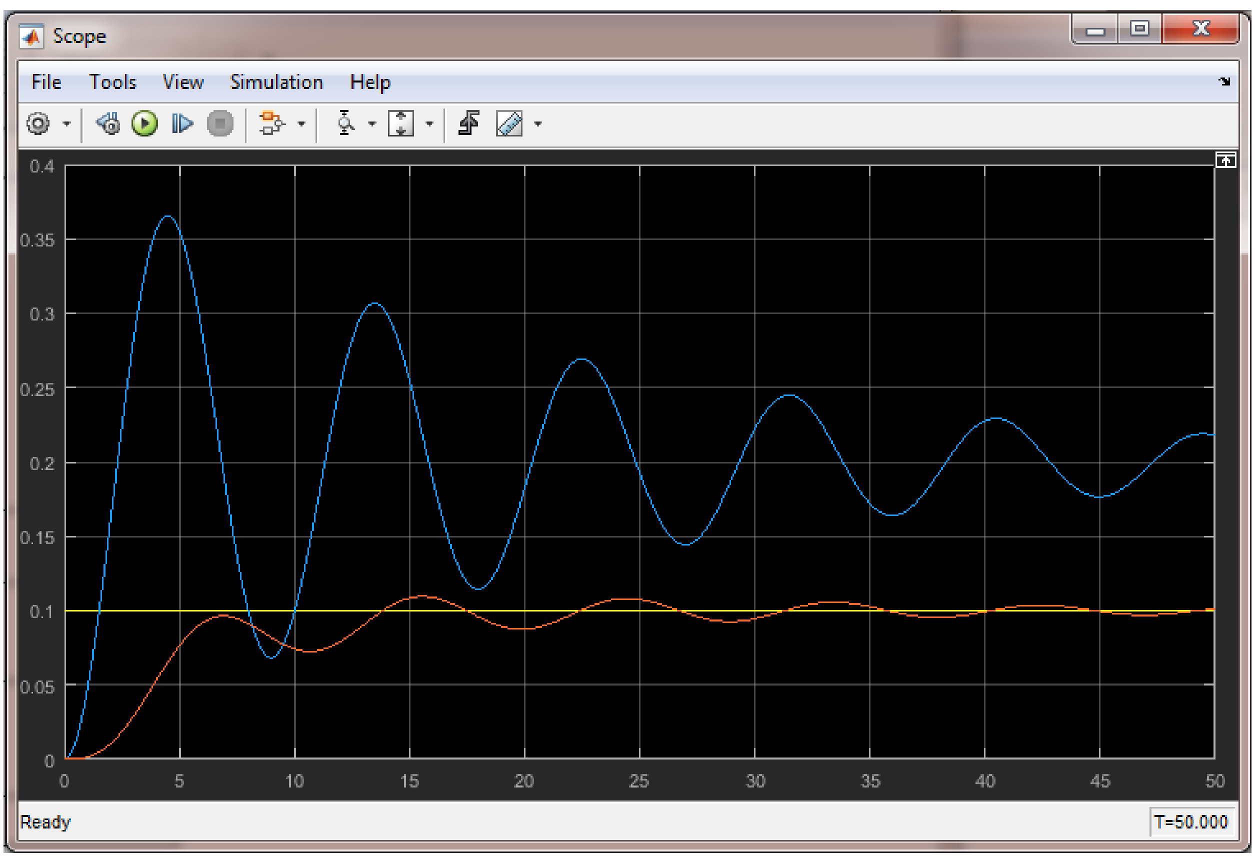
$$\ddot{y} = -\frac{b}{m}\dot{y} + \frac{b}{m}\dot{x} - \frac{k}{m}y + \frac{1}{m}u \tag{24}$$

Conclusion

Using eqns. (23) and (24) to help construct the closed-loop system:



Using the traditional step input of 1 results in the following output:



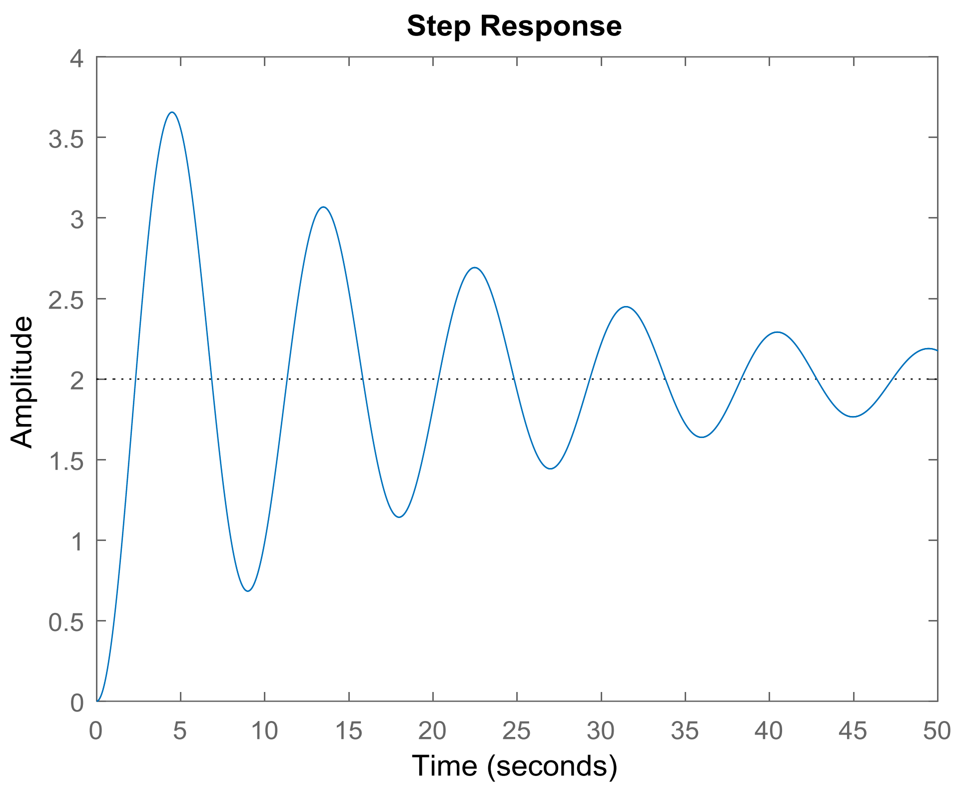
The output can be verified with MATLAB using eqn. (17) and the *step()* function:

```
%Define parameters
```

```
k = 0.5
m = 1
b = 0.1

%Create the closed-loop system
num = [m/b 2]
den = [ (m^2/b) 3*m (b+k*m/b) 2*k]
sys = tf(num, den)

%Simulate the step response
stepinfo(sys)
step(sys,0:0.01:50)
```



The step response has the following characteristics when given a step input of 1:

```
ans =

        RiseTime: 1.5520
    SettlingTime: 81.3035
     SettlingMin: 0.6834
     SettlingMax: 3.6566
       Overshoot: 82.8299
      Undershoot: 0
            Peak: 3.6566
        PeakTime: 4.4861
```

The system has a peak overshoot of 82.8299% of its final value of 2.

As t tends towards infinity the response approaches a value of 2, making the closed-loop system stable.

Calculating the steady state error as time tends towards infinity:

$$E_{ss} \equiv \lim_{t \to \infty} \{y(t) - x(t)\} \tag{25}$$

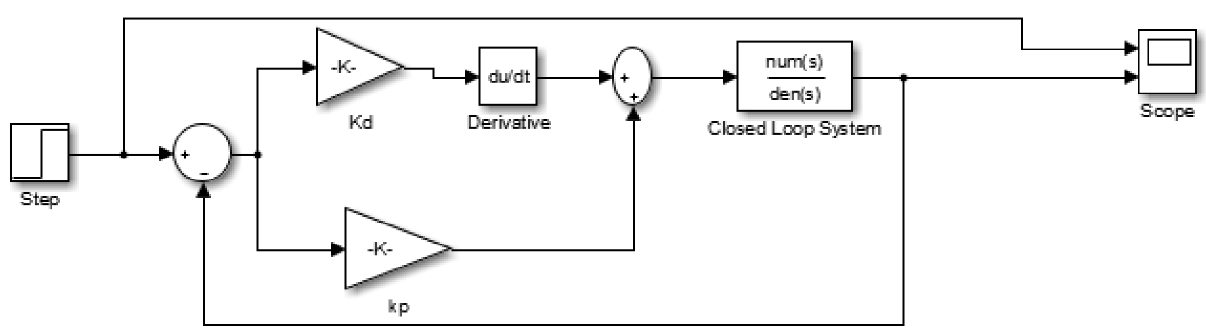
$$E_{ss} = 2 - 1 = 1$$

The last part of the lab manual asks the student to implement PD control with the closed loop model. Since it was not explicitly stated in the lab manual to use the differential equations, I used the transfer function of the closed-loop system, eq. (17):

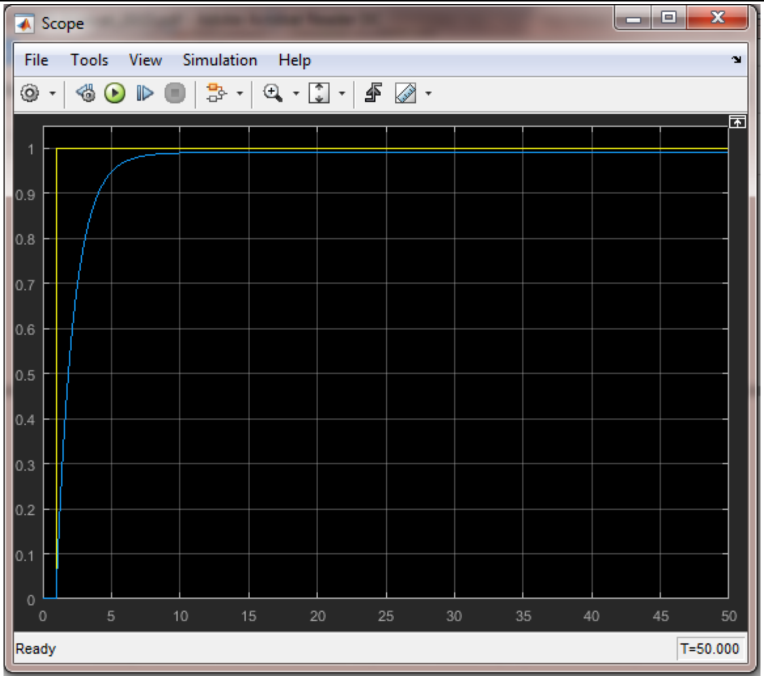
```
%%
%Closed loop system with PD control

%Define parameters
k = 0.5
m = 1
b = 0.1
s = tf('s')

%Create the closed-loop system
num = [m/b 2]
den = [ (m^2/b) 3*m (b+k*m/b) 2*k]
sys = tf(num, den)
```



The gains were to be adjusted to achieve negligible overshoot and a settling time of about 10s:



A proportional gain of 50 and a derivative gain of 65 satisfies the system specifications of negligible overshoot and a settling time of roughly 10 seconds. To determine the appropriate gains, the output as observed and the gains were adjusted by trial and error, beginning with smaller gains and steadily increasing them to satisfy the requirements. Steady-state was not able to be fully eliminated by this controller as illustrated in the difference between the step input and the step response in the graph above.

Conclusion

SIMULINK facilitates system analysis, control, and implementation with its capabilities of creating systems from block components. It is particularly useful in realizing systems of differential equations and transfer functions. Once the system is created and the behavior is observed, creating a PD controller will work to achieve desired specifications like settling time and percent overshoot.