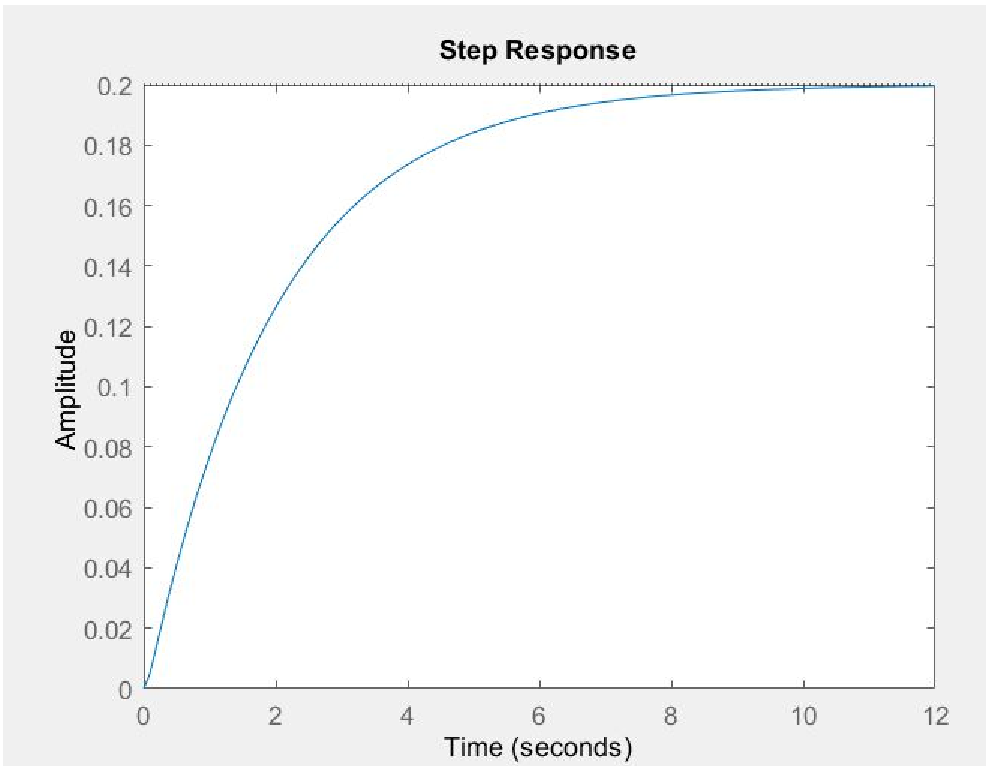
Mass-spring-damper

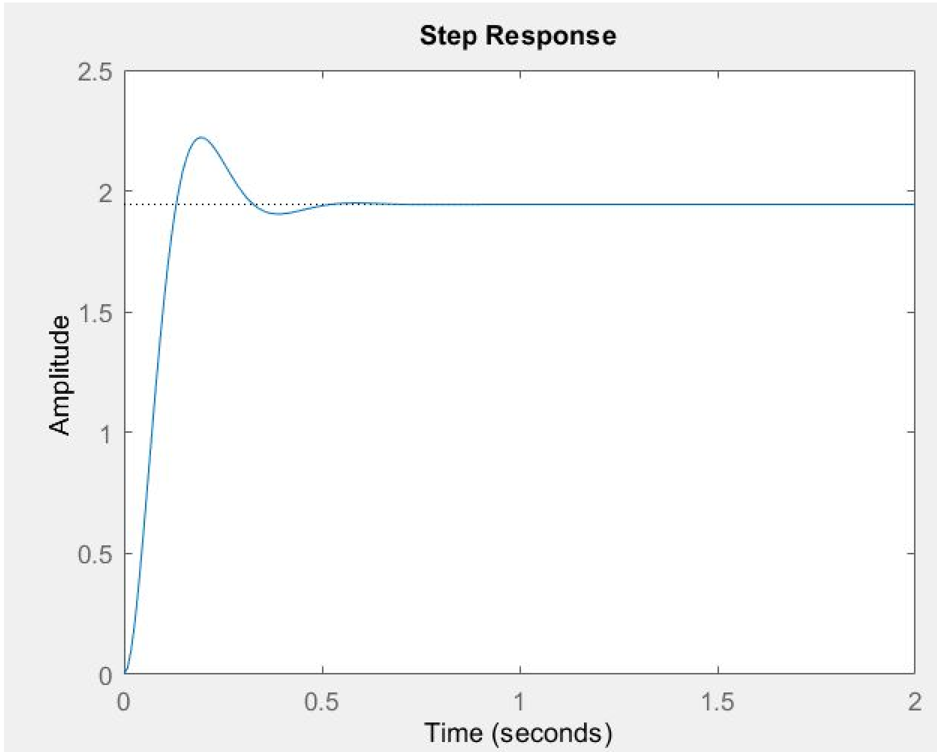
1.

```
>> sys=tf(1,[1 20 10]);
>> step(2*sys)
```



2.

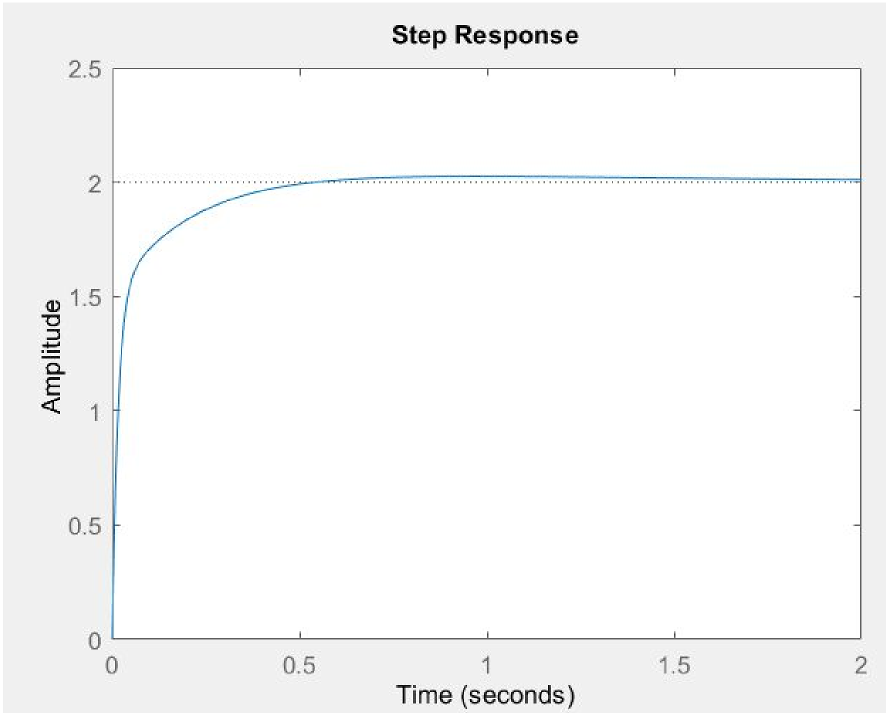
```
>> clear all;
clc;
sys1=tf(1,[1 20 10]);
sys2=pid(350);
sys3=series(sys1,sys2);
sys=feedback(sys3,1)
step(2*sys,2)
```




```
sys =

       350
  ----------------
  s^2 + 20 s + 360

Continuous-time transfer function.
```

3.

```
>> clear all;
clc;
sys1=tf(1,[1 20 10]);
sys2=pid(350,300,50);
sys3=series(sys1,sys2);
sys=feedback(sys3,1)
step(2*sys,2)
```

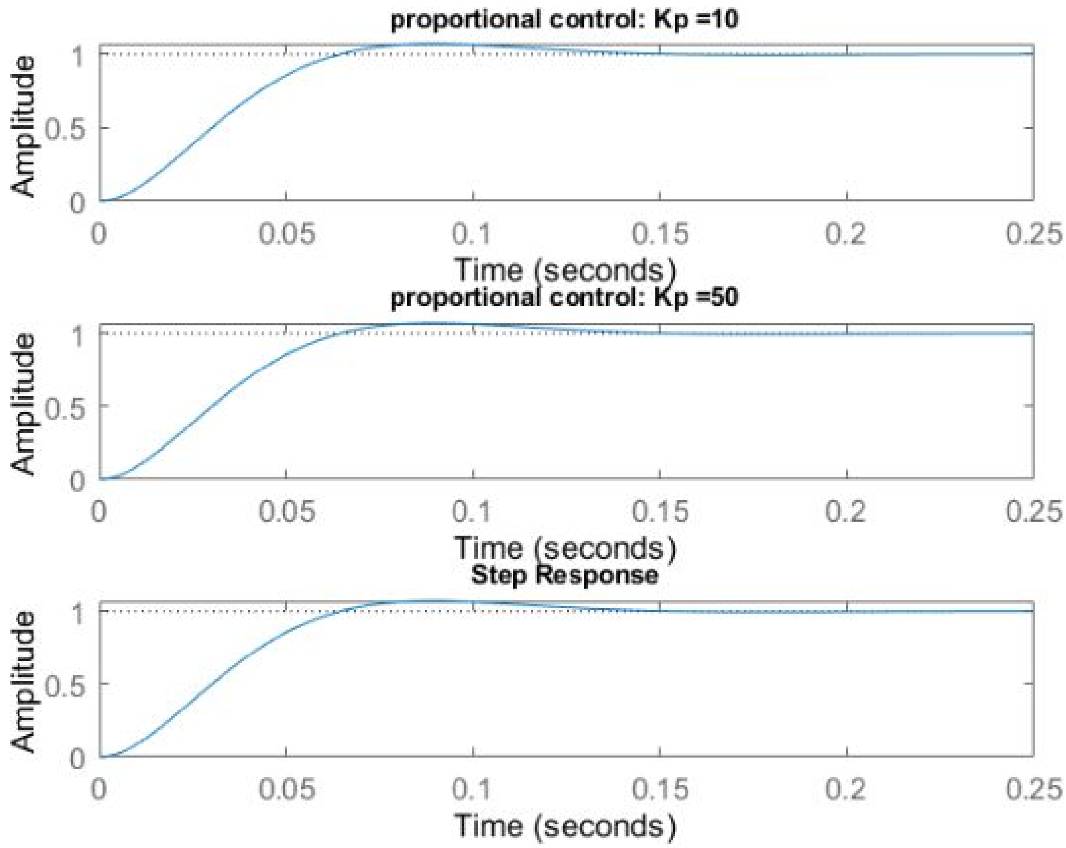



```
sys =

     50 s^2 + 350 s + 300
  --------------------------
  s^3 + 70 s^2 + 360 s + 300

Continuous-time transfer function.
```

Motor Position:

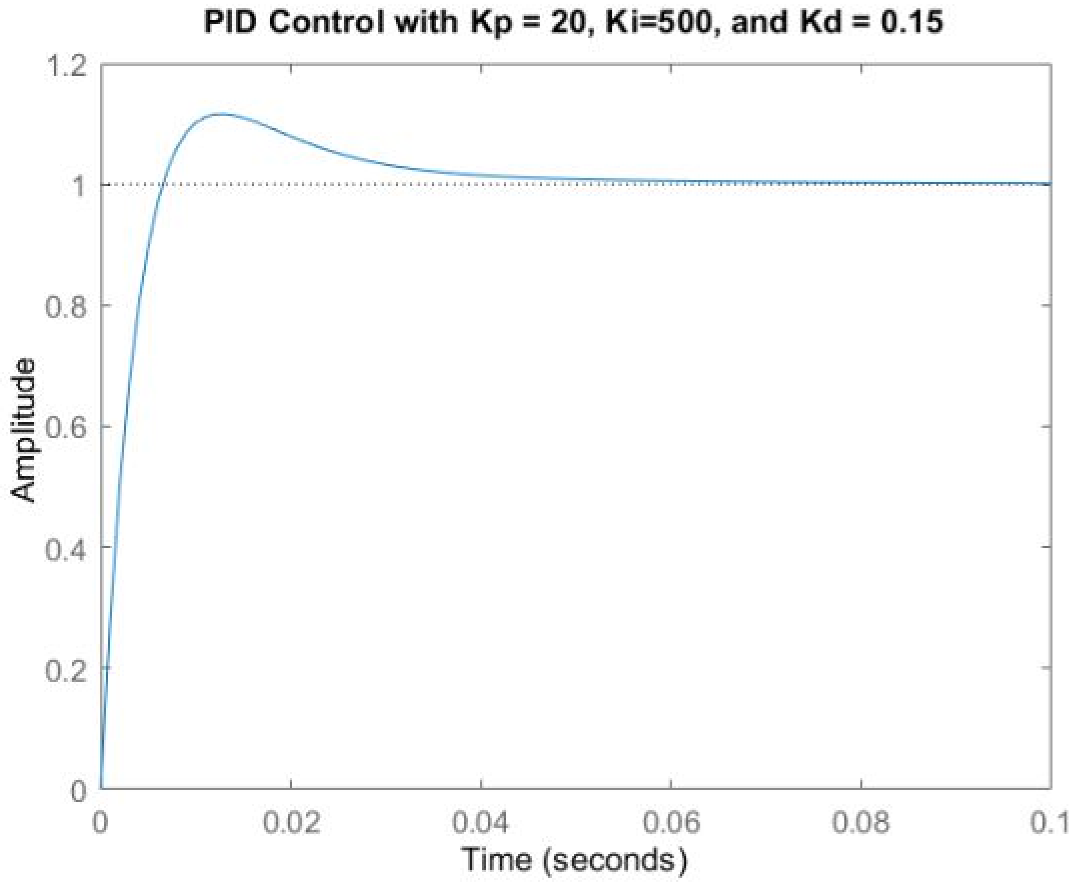
1.

```
j = 3.23*10^-6;
b = 3.51*10^-6;
l = 2.75*10^-6;
r = 4;
K = .0275;

kp = 1;
num = [K*kp];
den = [l*j (l*b+r*j) (b*r+K^2), K*kp];
sys1 = tf(num, den);

kp = 10;
num = [K*kp];
den = [l*j (l*b+r*j) (b*r+K^2), K*kp];
sys10 = tf(num, den);

kp = 50;
num = [K*kp];
den = [l*j (l*b+r*j) (b*r+K^2), K*kp];
sys50 = tf(num, den);

t = 0:.001:.25;
figure(1)
subplot(3,1,1)
title("proportional control: Kp =1")
step(sys1,t);
title("proportional control: Kp =10")
subplot(3,1,2)
step(sys1,t)
title("proportional control: Kp =50")
subplot(3,1,3)
step(sys1,t)
```



2.

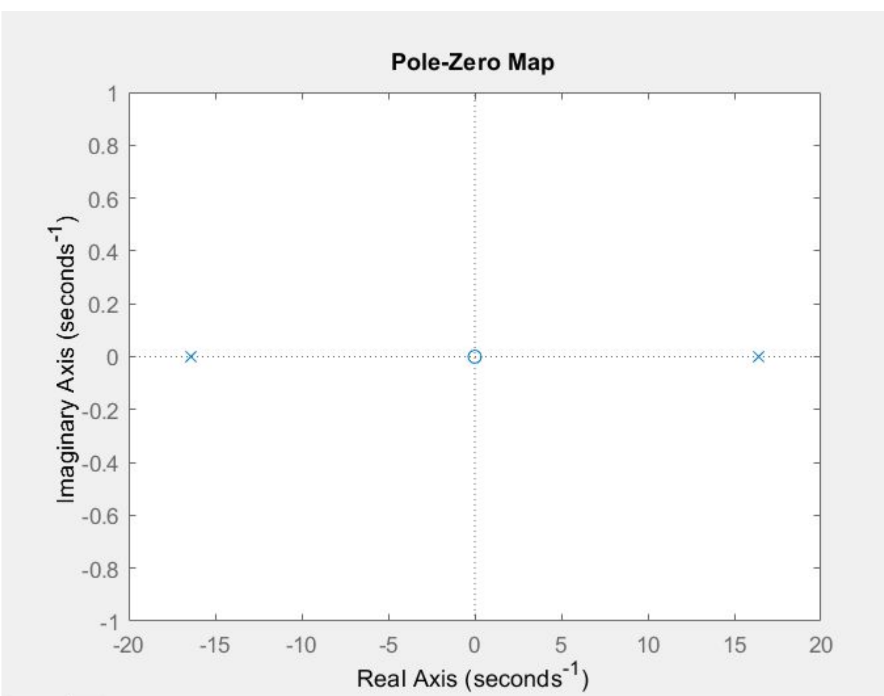
```
kp = 20;
ki = 500;
kd = .15;
num = [K*kd K*kp K*ki];
den = [l*j (l*b+r*j) (b*r+K^2+K*kd) K*kp K*ki];
sysPID = tf(num, den);
t = 0:.001:.1;
figure(2)
step(sysPID, t)
title("PID Control with Kp = 20, Ki=500, and Kd = 0.15")
```


PID Control with Kp = 20, Ki=500, and Kd = 0.15
Amplitude
Time (seconds)
1.2
1
0.8
0.6
0.4
0.2
0
0
0.02
0.04
0.06
0.08
0.1

Part 2:

1.

```
>> num = [31.81 0];
>> dem = [1.155 0 -312]

dem =

    1.1550         0 -312.0000

>> sys = tf(num, den)
Unrecognized function or variable 'den'.

Did you mean:
>> sys = tf(num, dem)

sys =

      31.81 s
  ---------------
  1.155 s^2 - 312

Continuous-time transfer function.

>> pole(sys)

ans =

   16.4356
  -16.4356
```



There is one negative and one positive pole output by our transfer function. Because there aren't two negative poles, this system is not stable.

Lab 5

1. $s^3 + a_1 s^2 + a_2 s + a_3 = s^3 + B_1 s^2 + B_2 s + B_3$

$s^3 + M_x s^2 + T_g s + -G_T G_V = s^3 + M_y s^2 - G_T G_V$

$s^3 + \frac{I_x d_r}{d_g} s^2 + mgd_r s - \frac{k_T}{R}\left(\frac{k_v}{d_r}\right) = s^3 + md_r s^2 - \frac{k_T}{R}\left(\frac{k_v}{d_r}\right)$

$\cancel{s^3} + \frac{.00215(.034)}{.062} s^2 + .955(9.8)(.034)s - \frac{.2025}{5.4}\left(\frac{.3253}{.034}\right) = \cancel{s^3} + .955(.034)s^2 - \frac{.2025}{5.4}\left(\frac{.3253}{.034}\right)$

$.001179s^2 + .3182s - \cancel{.3588} = .03247s^2 - \cancel{.3588}$

$-.031291 s^2 + .3182s = 0$

$-.031291 s(s - 10.1691) = 0$ $\quad s = 10.1691, 0$

$s^3 + a_1 s^2 + a_2 s + a_3$ $\quad s^3 + .03247s^2 - .3588$

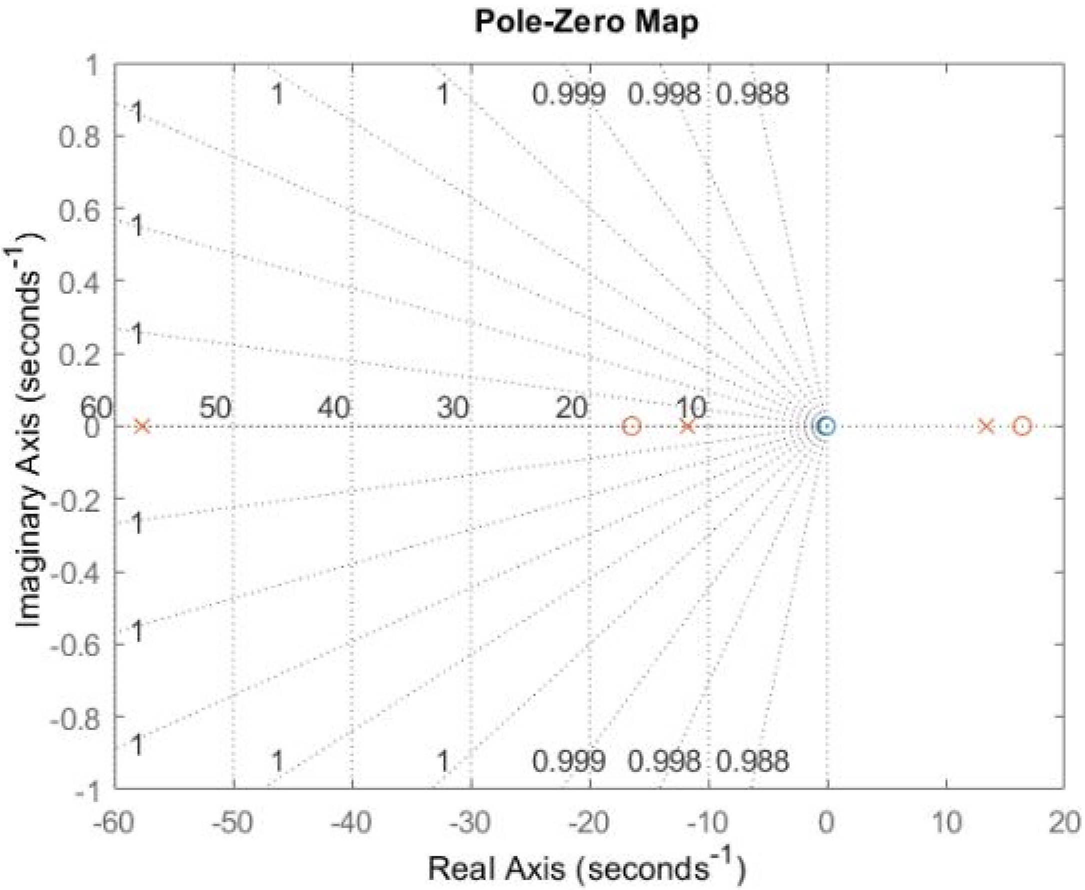
$s^3 + .001179s^2 + .3182s - .3588$ $\quad s = .699927$

$s = .56365$

values for $K = [.56365, .699927, 10.1691]$

```
A = [0,1,0; 270.17,0,-608.63;0,0,-22.10]
B = [0;31.81;1.15]
C = [1,0,0;0,0,1]
D = [0; 0]

k = [.56365 .699927 10.1691]

A = A-B*k

eig(A)

[num,den] = ss2tf(A,B,C,D)
sys1 = tf(num(1,:), den);
sys2 = tf(num(2,:), den);
pzmap(sys1, sys2)
grid on
```

```
A = 3×3
           0    1.0000         0
    270.1700         0 -608.6300
           0         0  -22.1000

B = 3×1
           0
     31.8100
      1.1500

C = 2×3
       1     0     0
       0     0     1

D = 2×1
       0
       0

k = 1×3
      0.5636    0.6999   10.1691

A = 3×3
           0    1.0000         0
    252.2403  -22.2647 -932.1091
     -0.6482   -0.8049  -33.7945

ans = 3×1
     13.4042
    -11.8128
    -57.6505

num = 2×4
           0         0   31.8100    3.0765
           0    1.1500    0.0000 -310.6955

den = 1×4
10^3 ×
      0.0010    0.0561   -0.2501   -9.1285
```



Intro:

In this lab we learned about basic PID control and designed a controller for our Balbot. We filled in our equations with the variables given and were able to compute the values of the proportional, integral, and derivative constants to achieve our desired characteristic equation. We were then able to evaluate the feedback from this controller using MATLAB.

Conclusion:

For this lab we learned the basics of PID control and how to achieve a desired characteristic equation using either the state space or transfer function of our system. We observed what happened when we input different values for the constants Kp Ki and Kd. We were able to understand how to design controllers for basic dynamical systems using P and PID control and design controllers for real robotic systems (the BalBot) using state feedback