# Lab 4

## ECE 380 W21

Group 8

Arjun Bawa, a3bawa Andrew Tran, a89tran

# Table of Contents

## Declaration of Authorship

We acknowledge and promise that:

a) We are the sole authors of this lab report and associated simulation files/code.
b) This work represents our original work.
c) We have not shared detailed analysis or detailed design results, computer code, or Simulink diagrams with any other student.
d) We have not obtained or looked at lab reports from any other current or former student of ECE/SE 380, and we have not let any other student access any part of our lab work.
e) We have completely and unambiguously acknowledged and referenced all persons and aids used to help us with our work.

Student1 Name and Signature:

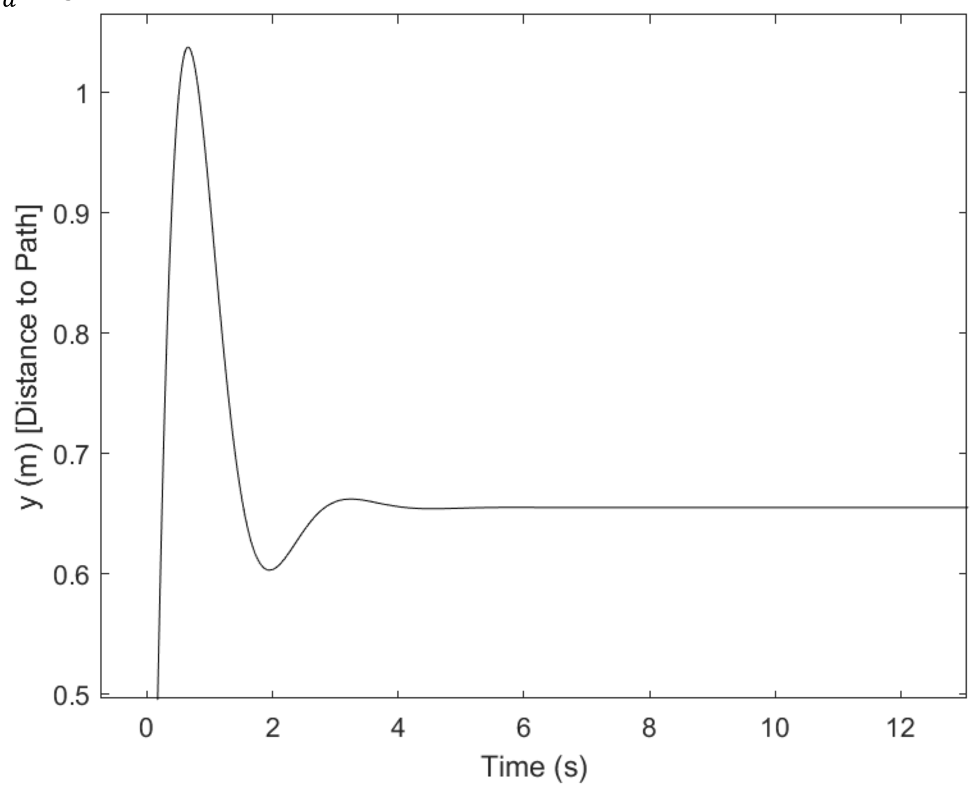
**Arjun Bawa**

Student2 Name and Signature:

**Andrew Tran**

## 4.1

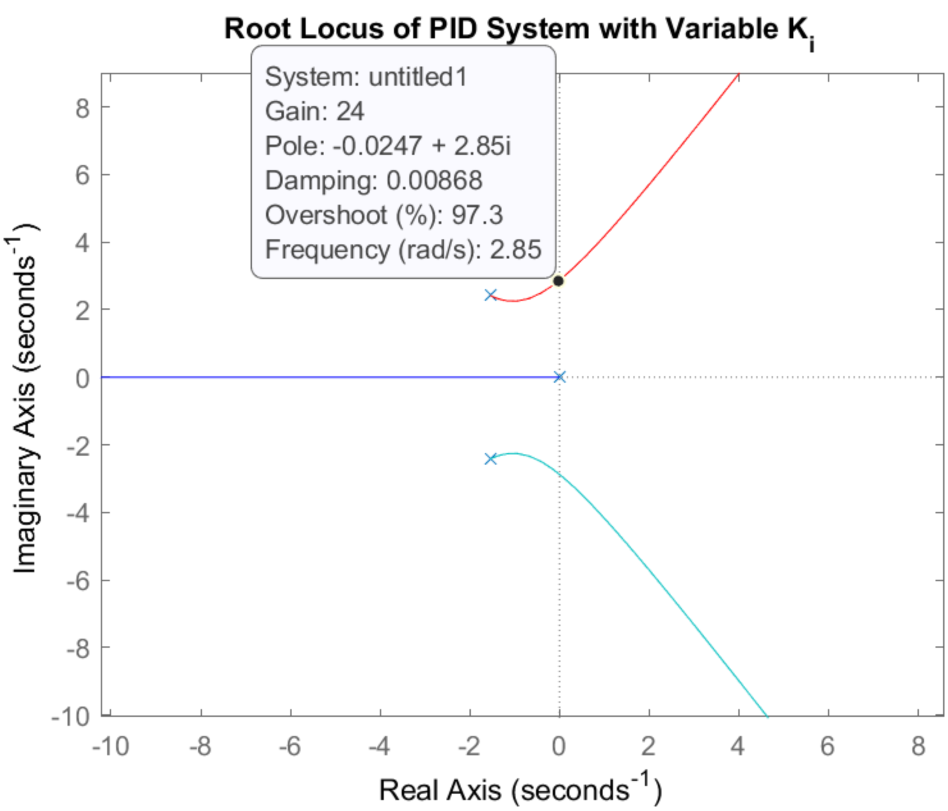
Choose $K_p = 8, K_d = 3$



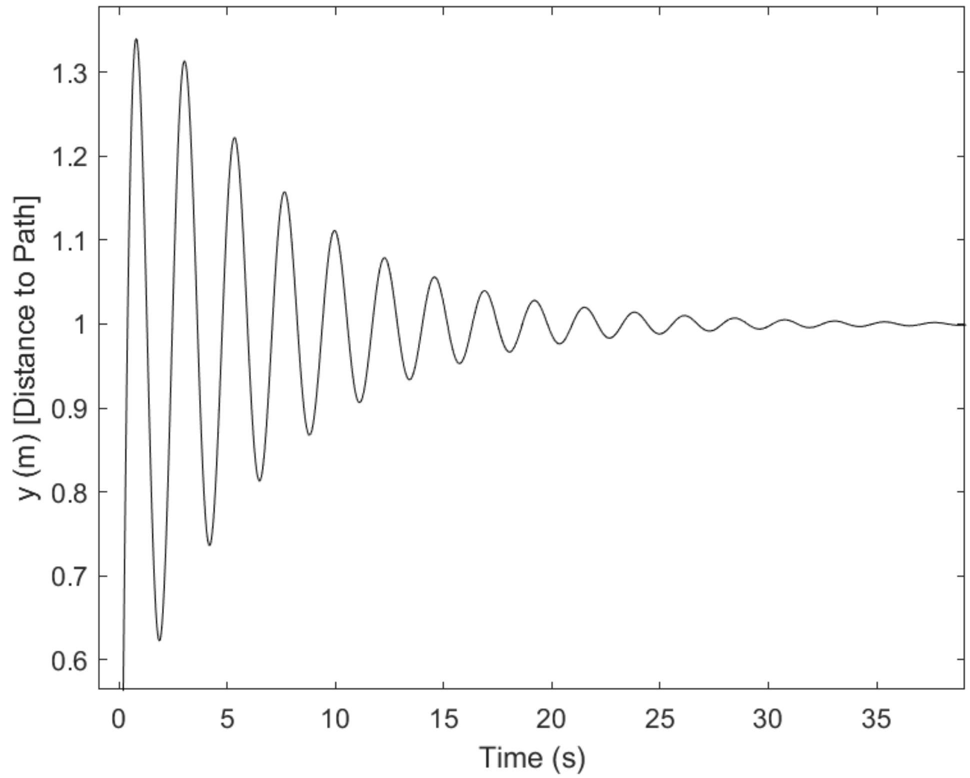

## 4.2

Choose $K_i = 20$

## 4.3

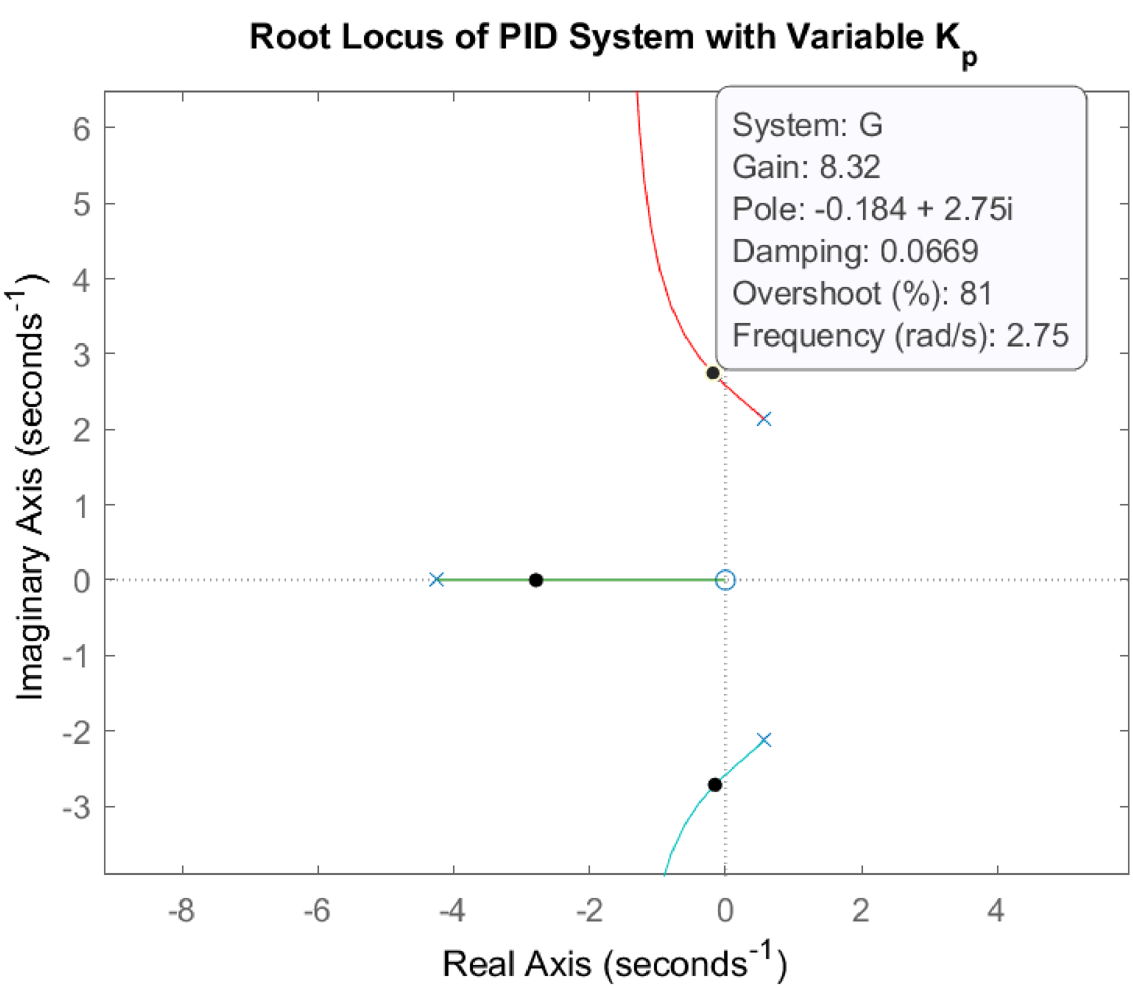


## 4.4

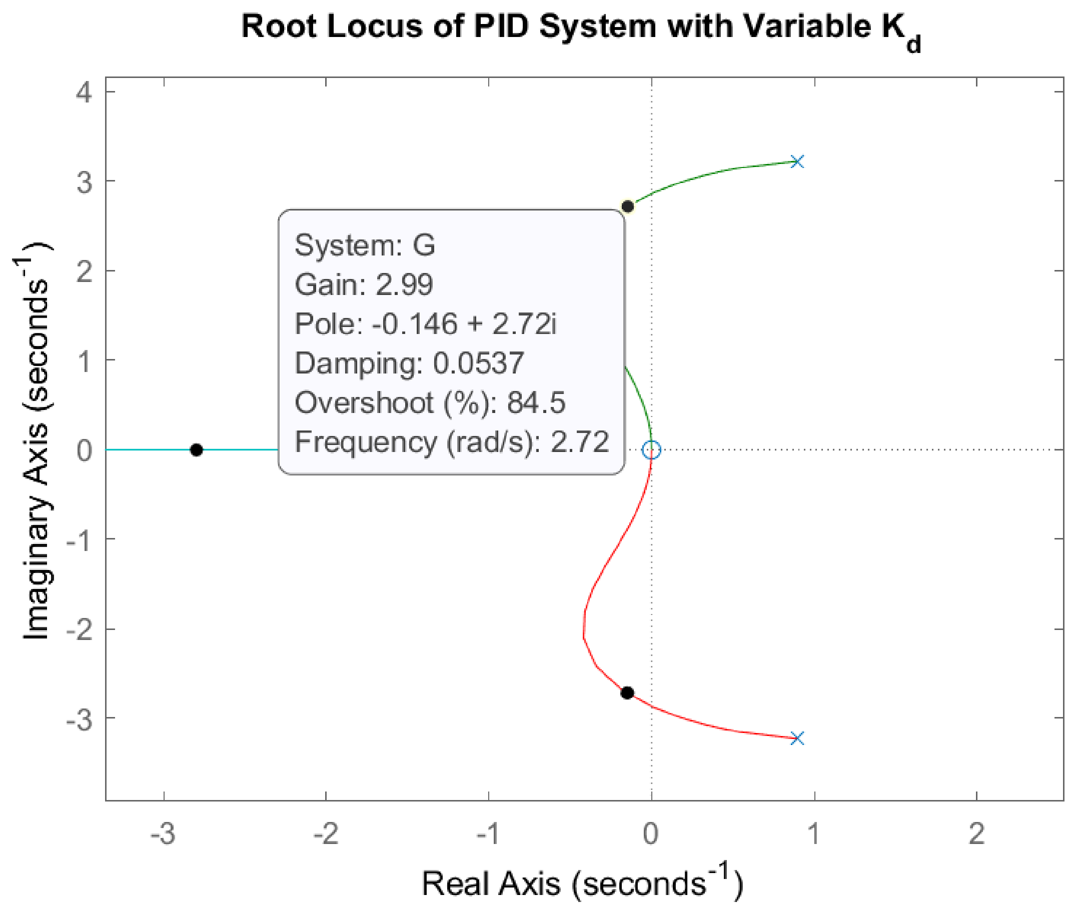


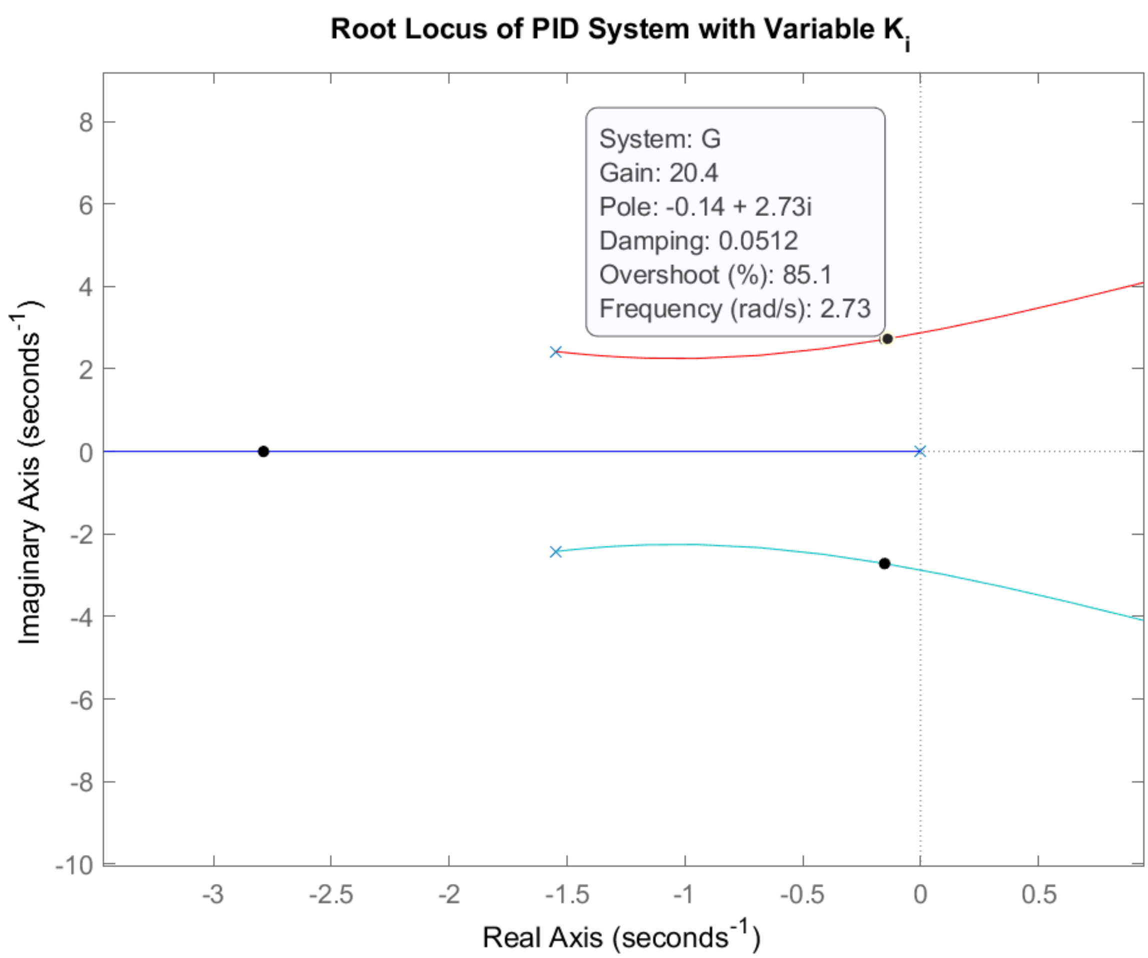
y (m) [Distance to Path]
Time (s)


## 4.5



Root Locus of PID System with Variable $K_p$
System: G
Gain: 8.32
Pole: -0.184 + 2.75i
Damping: 0.0669
Overshoot (%): 81
Frequency (rad/s): 2.75
Imaginary Axis (seconds$^{-1}$)
Real Axis (seconds$^{-1}$)


Root Locus of PID System with Variable $K_d$
System: G
Gain: 2.99
Pole: -0.146 + 2.72i
Damping: 0.0537
Overshoot (%): 84.5
Frequency (rad/s): 2.72
Imaginary Axis (seconds$^{-1}$)
Real Axis (seconds$^{-1}$)

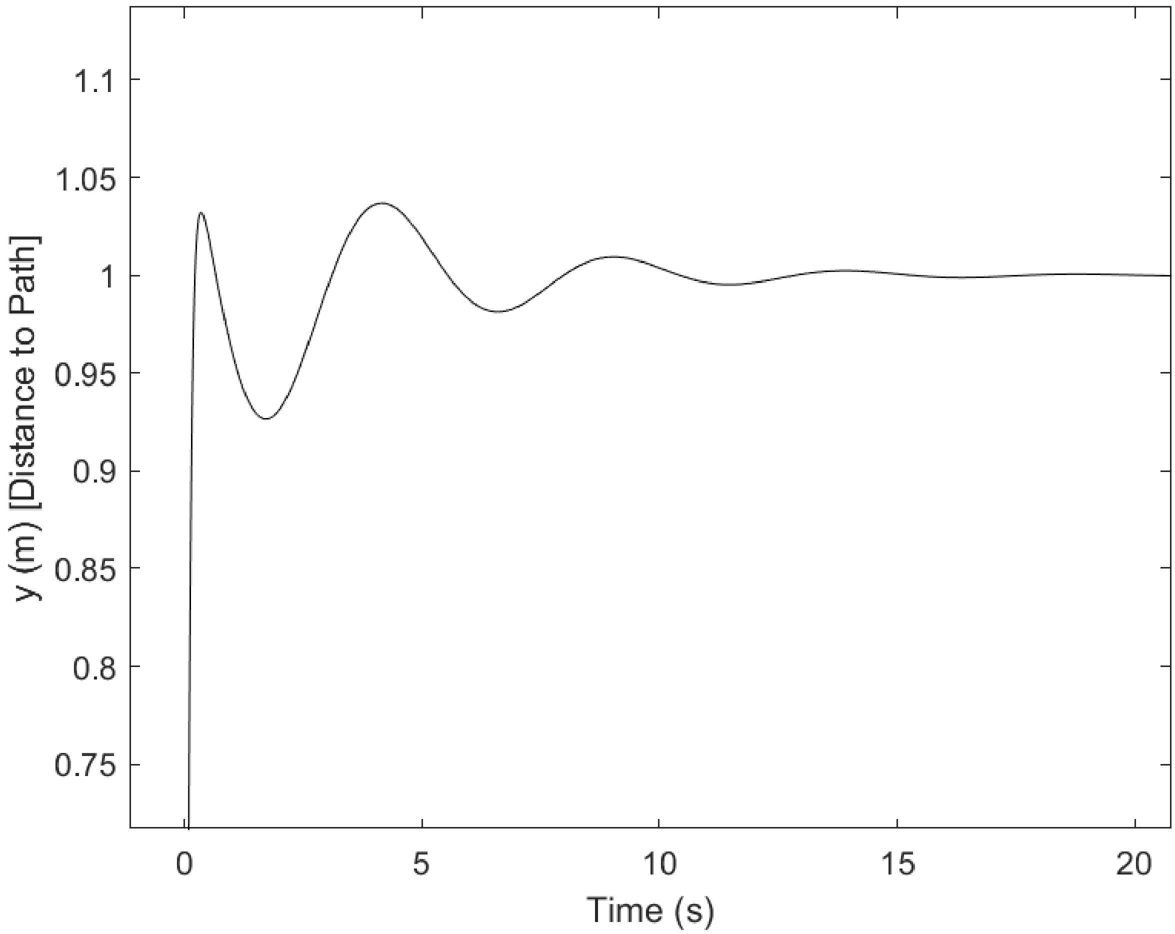

Root Locus of PID System with Variable $K_i$
System: G
Gain: 20.4
Pole: -0.14 + 2.73i
Damping: 0.0512
Overshoot (%): 85.1
Frequency (rad/s): 2.73
Imaginary Axis (seconds$^{-1}$)
Real Axis (seconds$^{-1}$)

## 4.6

Increase $K_d$ from 3 to 12 to reduce overshoot.

$$K_p = 8, K_d = 12, K_i = 20$$

## 4.7



4.8

1

(1) Like in Lab 3, the primary output was a (relative) position $y(t)$. Unlike Lab 3, we do not observe the velocity explicitly but, instead, use the (practical) differentiator in the PID controller to estimate velocity (derivative of the relative position is velocity) and apply that in closed-loop feedback. Unfortunately, the explicit use of a differentiator in parallel to the proportional error feedback introduced zeros in the final transfer function $G(s)$ from $r(t)$ to $y(t)$. In particular

$$G(s) = \frac{(NK_d + K_p)s^2 + (NK_p + K_i)s + NK_i}{s^4 + Ns^3 + (NK_d + K_p)s^2 + (NK_p + K_i)s + NK_i}.$$

For your final choice of $K_p$, $K_i$ and $K_d$, **determine** where the zeros of $G(s)$ are located. *You can inspect the MATLAB scripts or Simulink model to find the filter coefficient $N$.*

Filter Coeff. $:= N = 100$

$K_p = 8, \quad K_d = 12, \quad K_i = 20$

$$\Rightarrow G(s) = \frac{(100(12)+8)s^2 + (100(8)+20)s + 100(20)}{s^4 + 100s^3 + (100(12)+8)s^2 + (100(8)+20)s + 100(20)}$$

$$= \frac{1208s^2 + 820s + 2000}{s^4 + 100s^3 + 1208s^2 + 820s + 2000}$$

$G(s) := \frac{P(s)}{D(s)} \Rightarrow$ Zeros of $G(s)$: values of $s$ where $P(s) = 0$

$$\Rightarrow 1208s^2 + 820s + 2000 = 0$$

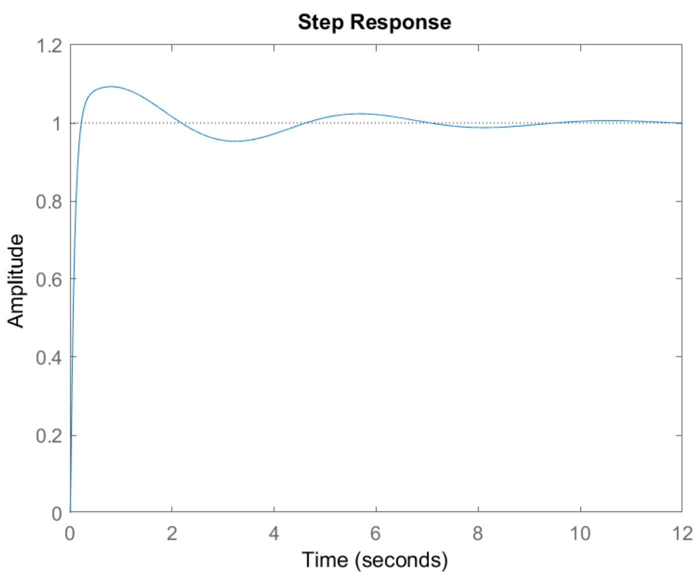
$$\Rightarrow \boxed{s \doteq -0.3394 \pm j1.2411}$$

## 2

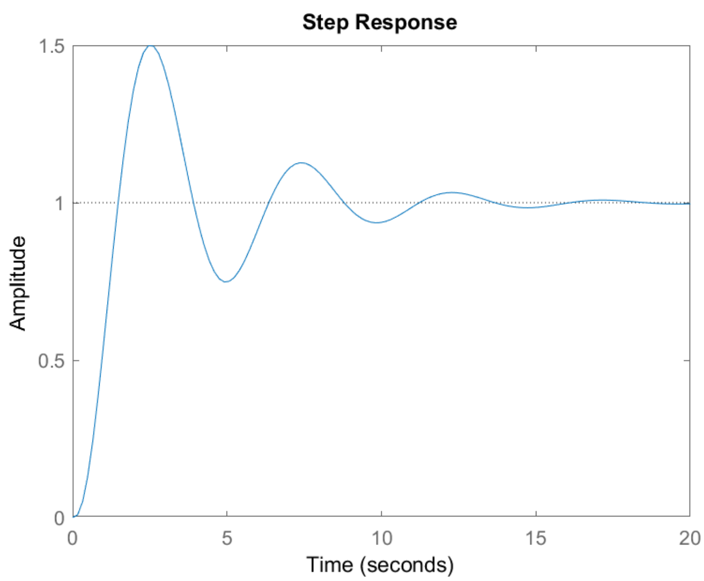
```
G =

          1208 s^2 + 820 s + 2000
  ---------------------------------------
  s^4 + 100 s^3 + 1208 s^2 + 820 s + 2000

Continuous-time transfer function.

>> zero(G)

ans =

  -0.3394 + 1.2411i
  -0.3394 - 1.2411i

>> pole(G)

ans =

 -86.0729 + 0.0000i
 -13.3645 + 0.0000i
  -0.2813 + 1.2882i
  -0.2813 - 1.2882i
```

```
>> 4*abs(unique(real(zero(G)))) <= abs(unique(real(pole(G))))

ans =

  3×1 logical array

   1
   1
   0
```

Therefore, zeros affect system's response.




Step response for $G(s)$

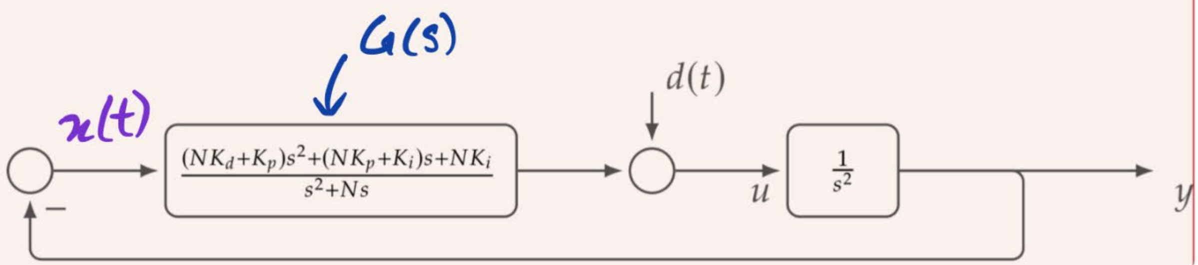



Step response for $\tilde{G}(s)$

We see that the zeros had a dampening effect on the step response of the system, dramatically reducing overshoot and settling time.

Observing 4.7, we see the zeros affected the response drastically until the 15 second mark. The zeros dampened the response relative to the response of $\tilde{G}(s)$, to the point of causing undershoot at $0 < t < 5\ s$.

(3) The PD controller of Part I produced a steady-state constant error. This was explained, in Part II, as being due to a step input disturbance to the plant $P(s)$. **Prove** that if $K_i \neq 0$ then step input disturbances to the plant are perfectly rejected. To do so, first find the transfer function from $d(t)$ to $y(t)$ in the diagram




$$x(t) = -y(t) \Rightarrow X(s) = -Y(s)$$

$$\Rightarrow U(s) = D(s) + G(s)X(s) = D(s) - G(s)Y(s)$$

$$Y(s) = \frac{1}{s^2}U(s) = \frac{1}{s^2}\left(D(s) - G(s)Y(s)\right)$$

$$\Rightarrow Y(s)\left(1 + \frac{1}{s^2}G(s)\right) = \frac{1}{s^2}D(s)$$

$$\Rightarrow \frac{Y(s)}{D(s)} = \frac{1}{s^2\left(1 + \frac{1}{s^2}G(s)\right)} = \frac{1}{s^2 + G(s)}$$

$$= \frac{1}{s^2 + \frac{(NK_d+K_p)s^2 + (NK_p+K_i)s + NK_i}{s^2 + Ns}}$$

$$\Rightarrow G_d(s) := \frac{s^2 + Ns}{s^4 + Ns^3 + (NK_d+K_p)s^2 + (NK_p+K_i)s + NK_i}$$

$$d(t) = A \cdot 1(t) \Rightarrow D(s) = \frac{A}{s}$$

$$\Rightarrow Y(s) = G_d(s) \cdot D(s) = \frac{A}{s} G_d(s)$$

Assuming all poles of $AG_d(s)$ in OLHP ($K_i \neq 0$ required), using FVT:

$$\lim_{t \to \infty} y(t) = \boxed{\lim_{s \to 0} sY(s) = 0} \quad \blacksquare$$

4

Increasing $K_p$ generally results in decrease of settling time. An increase in $K_p$ generally corresponds with an increase in $\theta$ from the origin to each pole, creating an increase in overshoot. An increase in $K_p$ also corresponds to a decrease in rise time and peak time.

Increasing $K_d$ generally results in decrease of settling time. An increase in $K_d$ generally corresponds with a decrease in $\theta$ from the origin to each pole, creating a decrease in overshoot. An increase in $K_d$ also corresponds to a decrease in rise time and peak time.

Increasing $K_i$ (up till its upper limit) generally results in decrease of settling time. An increase in $K_p$ generally corresponds with an increase in $\theta$ from the origin to each pole, creating a decrease in overshoot. An increase in $K_i$ also corresponds to a decrease in rise time and peak time.

5

From the proof in part 3, we know that an integrator term creates perfect rejection of any type of step-input disturbances. When analyzing a system with $K_i = 0$, a constant error cannot be mitigated by the controller, meaning a steady-state error can't be fixed. When the integrator term is included (i.e. $K_i \neq 0$), the time a constant error *persists* is accounted for when error-correcting. This means over time, the integrator will get rid of constant error that persists, leading to no steady-state error by step-input disturbance.

The trade-off of the inclusion of the integrator term is that it makes the system more "sluggish" by increasing overshoot and settling time.

## 6

This change would not change the answer in a significant way other than there being no poles at $s = 0$ with this plant. A pole would need to be added at $s = 0$ for steady-state error tracking.

## 7

We aimed to improve overshoot by increasing $K_d$. The reason $K_d$ was chosen to increase was simply that it had the best benefits-to-drawbacks ratio out of all the possible changes to any of the gains. Increasing $K_d$ not only reduced significant overshoot, but improved rise time as well without increasing settling time. An increase in $K_d$ causes a decrease in $\theta$ from the origin, decreasing overshoot.