

# RBE 500 Homework #6

Arjan Gupta

## Question 1

Consider the following robot joint model

$$J\ddot{\theta}(t) + B\dot{\theta}(t) = u(t) + d(t)$$

where  $J$  is the inertia of the link,  $B$  is the effective damping on the link,  $\theta$  is the joint angle,  $u$  is the actuator torque (input), and  $d$  is the disturbance acting on the system.

First, assume that disturbance is zero and take  $J = 2$ ,  $B = 0.5$ . Design a PD controller such that the closed loop system is critically damped, and settling time is 2 second. Do not do this by tuning the gains; calculate the  $K_p$  and  $K_d$  gains using natural frequency and damping ratio.

## Solution

Since  $d(t) = 0$ ,  $J = 2$ ,  $B = 0.5$ , we have

$$2\ddot{\theta}(t) + 0.5\dot{\theta}(t) = u(t)$$

Transform to Laplace domain,

$$2\Theta(s)s^2 + 0.5\Theta(s)s = U(s)$$

$$\Theta(s)[2s^2 + 0.5s] = U(s) \quad (1)$$

$$\frac{\Theta(s)}{U(s)} = \frac{1}{2s^2 + 0.5s}$$

Let our PD controller model be

$$K_p e + K_d \dot{e} = u$$

Transform to Laplace domain,

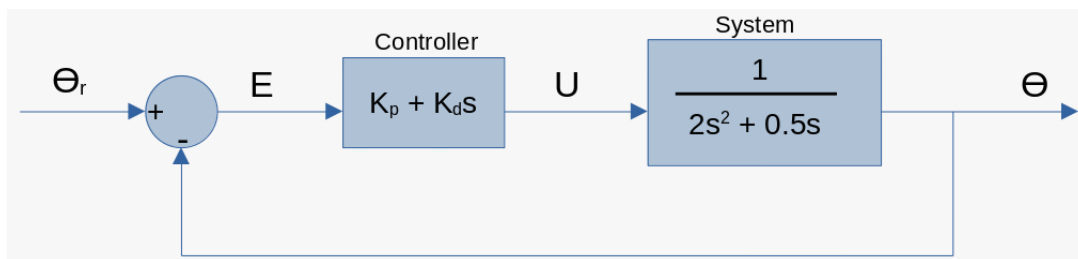
$$K_p E(s) + K_d E(s)s = U(s)$$

$$E(s)[K_p + K_d s] = U(s)$$

Therefore, the transfer function for the PD controller is

$$\frac{U(s)}{E(s)} = K_p + K_d s \quad (2)$$

Now we can draw the block diagram, as shown below.



From the block diagram, we can see that

$$E = \Theta_r - \Theta$$

Using equation 2,

$$\frac{U(s)}{K_p + K_d s} = \Theta_r - \Theta$$

Furthermore, using equation 1,

$$\begin{aligned}\frac{\Theta(s)[2s^2 + 0.5s]}{K_p + K_d s} &= \Theta_r - \Theta \\ \frac{\Theta[2s^2 + 0.5s]}{K_p + K_d s} + \Theta &= \Theta_r \\ \Theta \left( \frac{2s^2 + 0.5s}{K_p + K_d s} + 1 \right) &= \Theta_r \\ \Theta \left( \frac{2s^2 + 0.5s + K_p + K_d s}{K_p + K_d s} \right) &= \Theta_r\end{aligned}$$

Therefore,

$$\frac{\Theta}{\Theta_r} = \frac{K_p + K_d s}{2s^2 + s(0.5 + K_d) + K_p}$$

So our characteristic equation is,

$$\begin{aligned}2s^2 + s(0.5 + K_d) + K_p &= 0 \\ s^2 + s \frac{(0.5 + K_d)}{2} + \frac{K_p}{2} &= 0\end{aligned}$$

The general form of the characteristic equation is

$$s^2 + (2\xi\omega_n)s + \omega_n^2 = 0$$

Where  $\xi$  is the damping ratio and  $\omega_n$  is the natural frequency.

Hence, we have,

$$\omega_n^2 = \frac{K_p}{2} \tag{3}$$

and

$$2\xi\omega_n = \frac{(0.5 + K_d)}{2} \tag{4}$$

Also, we know that the natural frequency and settling time  $T_s$  are related by

$$\xi\omega_n T_s = 4$$

Since we are solving for a critically damped system, we set  $\xi = 1$ . We also want settling time  $T_s = 2$  seconds.

So,

$$\begin{aligned}\xi\omega_n T_s &= 4 \\ 1 \cdot \omega_n \cdot 2 &= 4 \\ \omega_n &= 2\end{aligned}$$

Plugging this into equation 3, we have

$$(2)^2 = \frac{K_p}{2}$$

$$4 = \frac{K_p}{2}$$

$$\boxed{K_p = 8}$$

Also, plugging in values into equation 4, we have

$$2(1)(2) = \frac{0.5 + K_d}{2}$$

$$8 = 0.5 + K_d$$

$$\boxed{K_d = 7.5}$$

## Question 2

Follow steps in the assignment PDF file. Explain the process and be sure to include the plot to your report.

### Solution

First we write the following MATLAB script that contains the system and controller values of the closed loop system. Here,  $J$  is the inertia of the link in the system model,  $B$  is the effective damping on the link in the system model,  $K_p$  is the proportional gain for the controller, and  $K_d$  is the derivative gain for the controller.

```
1 % Code for Question 2 of HW6 for RBE500
2
3 clear; close all; clc;
4
5 % System model
6 J = 2;
7 B = 0.5;
8
9 % Controller
10 K_p = 8;
11 K_d = 7.5;
```

Next, we run our script so that the variables are loaded into our base workspace in MATLAB. We then type `simulink` into the MATLAB command window to launch Simulink. Now we use the Library Browser to begin to construct our closed-loop block diagram. Our process is as follows:

- First, we choose the step function as the input block. We double-click on this block and set the step time as 0 instead of 1, because we want to provide an instantaneous signal.
- Then we choose gain blocks to construct our PD controller. By double-clicking on the gain blocks, we are able to indicate the variables `K_p` and `K_d` from our base workspace. In the case of `K_d`, we also add a derivative block before the gain block.
- Now we add a summation block to add these gains.
- Next, we add a transfer function block and write  $[J \ B \ 0]$  in the denominator to accurately represent our transfer function.
- Now we connect the output of the controller summation into our system model.
- We can now add a scope to monitor the output of our controlled system. We connect the transfer function output to the scope.
- Now we add a summation after the input, and edit it by double-clicking the summation and making sure it says  $| \ + \ -$ .
- We can now represent our feedback error. For this, we right-click the connector leading to the scope, and draw a line into the minus part of the summation after the input.
- The input is fed into the plus part of the summation right after the input, so we draw that connector.

- The output of the summation right after the input feeds into our controller, so we draw that connector as well.
- We are almost ready to run our simulation. However as explained in the lectures, in order to be more accurate, we want to edit the solver in our model settings such that the solver-selection is fixed-step, and ode3 (Bogacki-Shampine).
- Now our closed loop system is ready to run. We click the play button to run the simulation.
- Our plot is now ready, which can be viewed by double-clicking the scope block.

Figure 1 below shows the end-result of the closed loop system we constructed in Simulink.

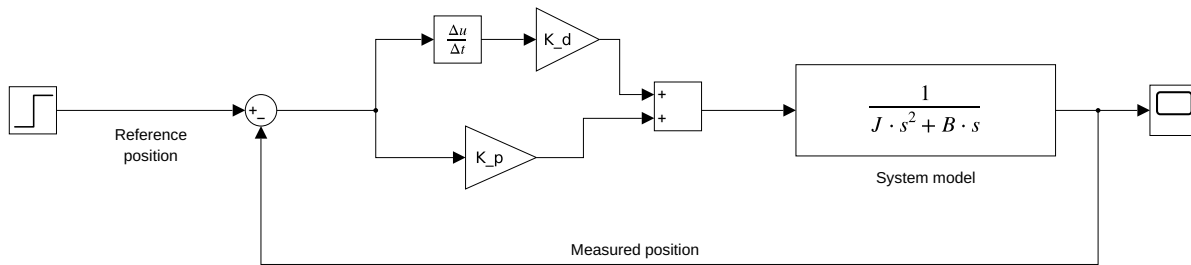


Figure 1: Block diagram for Question 2

After constructing the block diagram, we obtained the following plot (Figure 2).

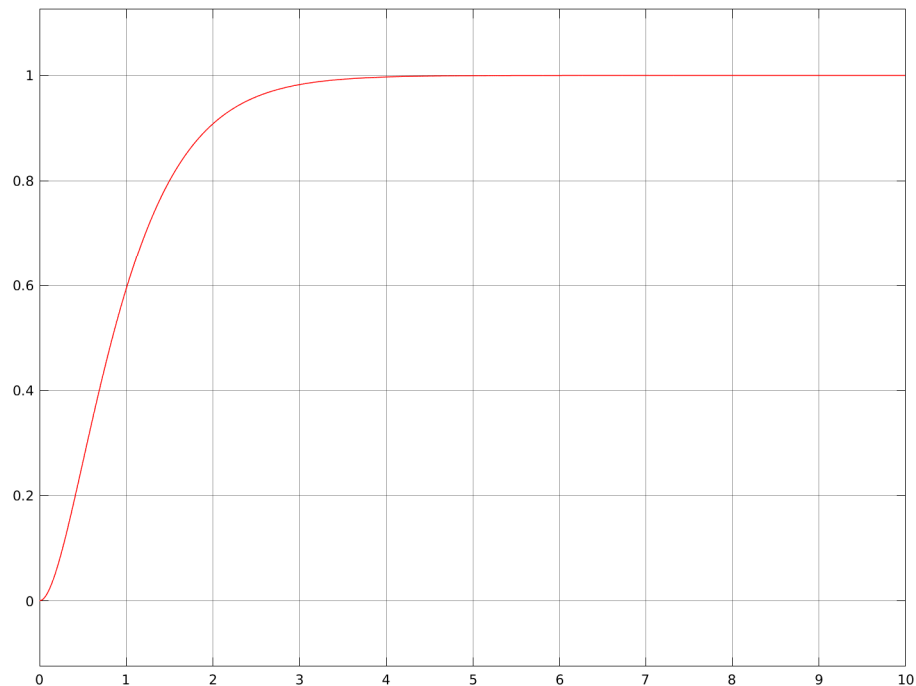


Figure 2: Generated plot for Question 2

## Question 3

You may have noticed that the system performance (overshoot and settling time) is quite close to your goal, but it might not be exactly there. Report the current settling time in your report (you may need to zoom in to the plot to see the exact settling time). This is because the equations defining the relations between the damping ratio, natural frequency, settling time and overshoot, are approximations, not exact relations. Specifically, the settling time equation is only accurate when damping ratio is  $\ll 1$ . Now, tune the values of  $K_p$  and  $K_d$  so that you get closer to your goal performance. Please remember the  $K_p$  and  $K_d$  to the system performance from our slides and discussions. Explain your process and include a plot showing the results after tuning. If your values have already been good and do not require tuning, report that as well.

## Solution

When we zoom into the plot we obtained in Question 2, we get the following plot (Figure 3).

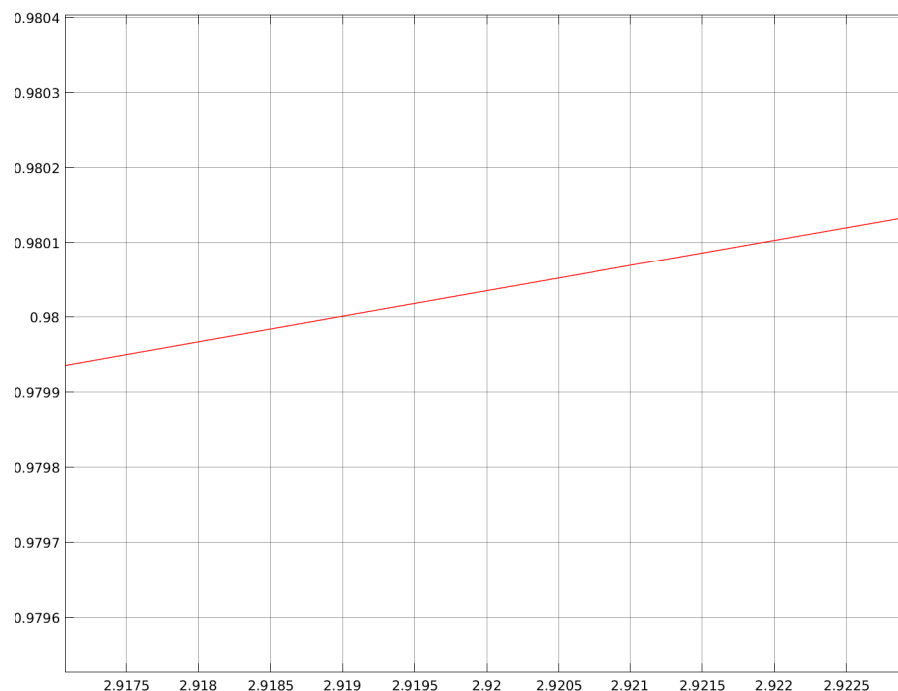


Figure 3: Zoomed-in plot for Question 3

It is clear from this plot that the system's response enters the 2% tolerance zone (i.e., above 0.98 and below 1.02) at 4.2 seconds. Therefore, the settling time  $T_s$  is 2.919 seconds.

Now we take the following steps to manually tune our  $K_p$  and  $K_d$  values to achieve our goal settling time of 2 seconds.

- First  $K_p$  was increased to 10. This decreased the settling time to about 2.2 seconds with no overshoot.
- Then  $K_p$  was increased to 12. This decreased the settling time to about 1.6 seconds but introduced an overshoot of about 1%.
- $K_p$  was reduced to 11. This decreased the settling time to 1.8, and reduced overshoot to 0.5%.
- Next, we increased  $K_d$  to 15. This drastically worsened our settling time (around 4 sec) while removing the overshoot.

- $K_d$  was then decreased to 6. This decreased our settling time to 1.3, and caused an overshoot of 5%.
- We then set  $K_d$  to 8. This decreased our overshoot to about 0.1%, and decreased our settling time to 2.034 seconds.

Therefore, by tuning our controller gains to  $K_p = 11$  and  $K_d = 8$ , we have achieved an improved controller.

The following plots show the improved system response and settling-time of the system.

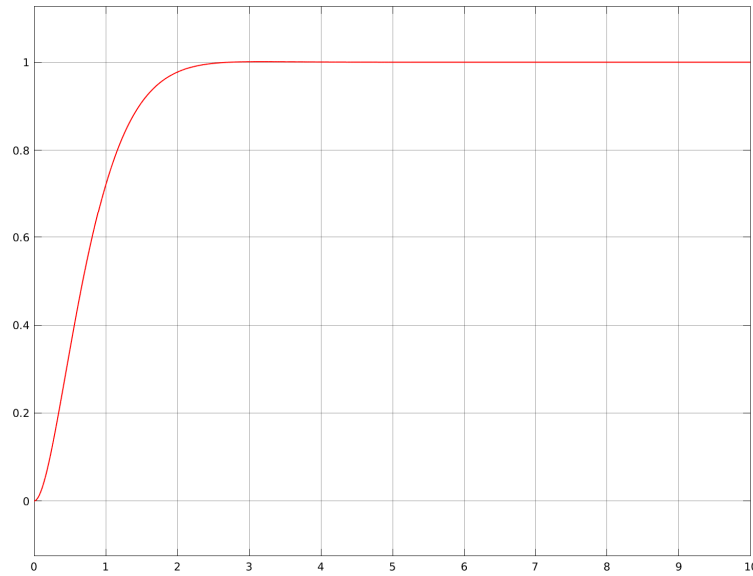


Figure 4: Tuned system reponse for Question 3

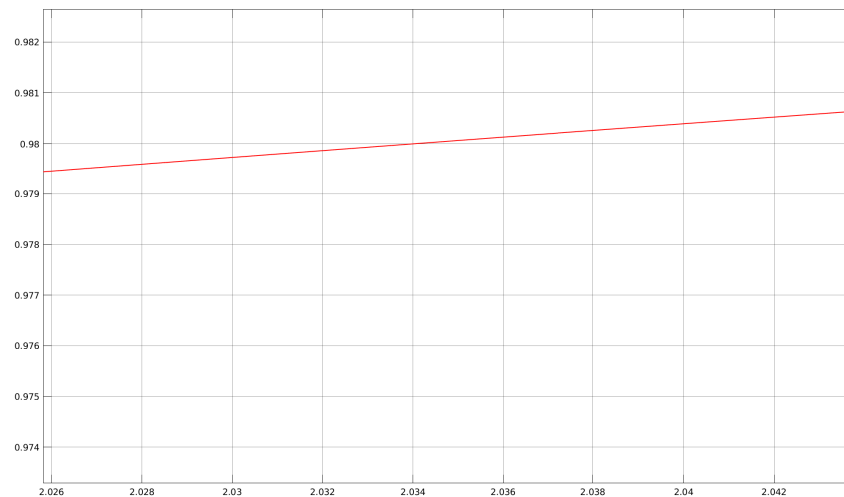


Figure 5: Zoomed-in plot to show settling time for Question 3



## Question 4

Now add a constant disturbance to the system, (Please check the lecture slides for where to add this disturbance from to your simulink block diagram), by using “constant” block from “Sources”. Assign the value  $D$  to the block. Define this same parameter in the script and set its value to  $D = 0.5$ . Provide the plot for theta with the same gain values to the previous question. You will see some steady state error. Report the amount of steady state error.

## Solution

Our new block diagram, with disturbance  $D$  added, is shown in Figure 6.

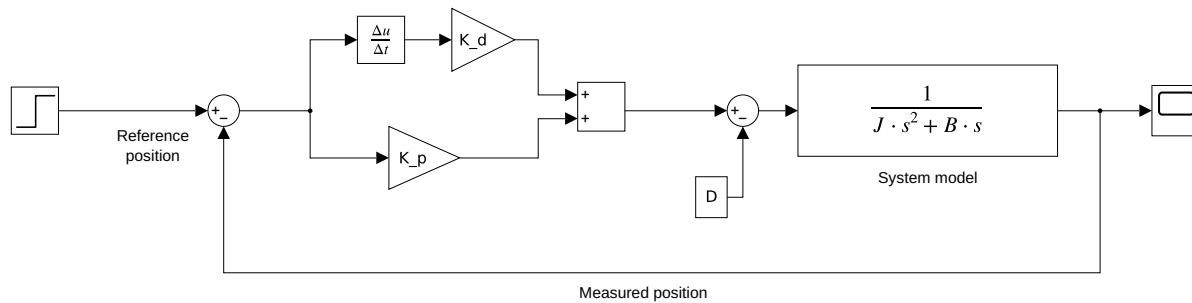


Figure 6: Block diagram with disturbance

At this point, our MATLAB script looks as follows.

```

1  % Code for HW6 of RBE500
2
3  clear; close all; clc;
4
5  % System model
6  J = 2;
7  B = 0.5;
8
9  % Controller
10 K_p = 11;
11 K_d = 8;
12
13 % Disturbance
14 D = 0.5;

```

Now we run our simulation again, and obtain the plot shown on the next page (Figure 7).

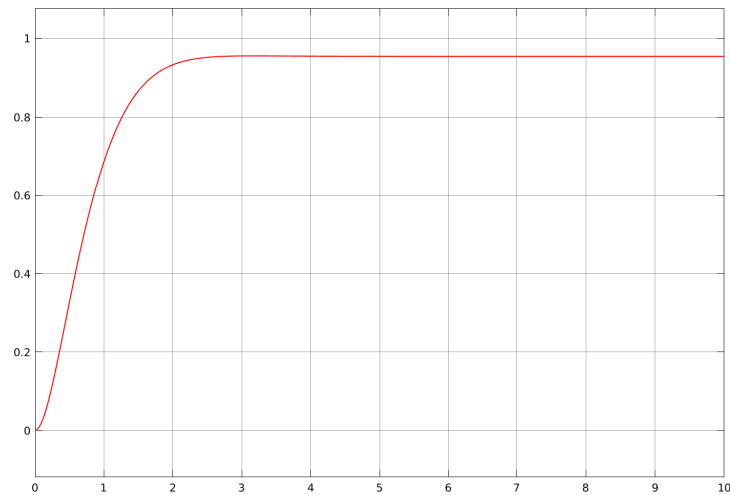


Figure 7: Disturbed system response for Question 4

Clearly, there is a steady state error because the response stabilizes before it reaches the target value of 1. Therefore it is a negative error.

To see the magnitude of the error, let us zoom into the plot as shown in Figure 8.

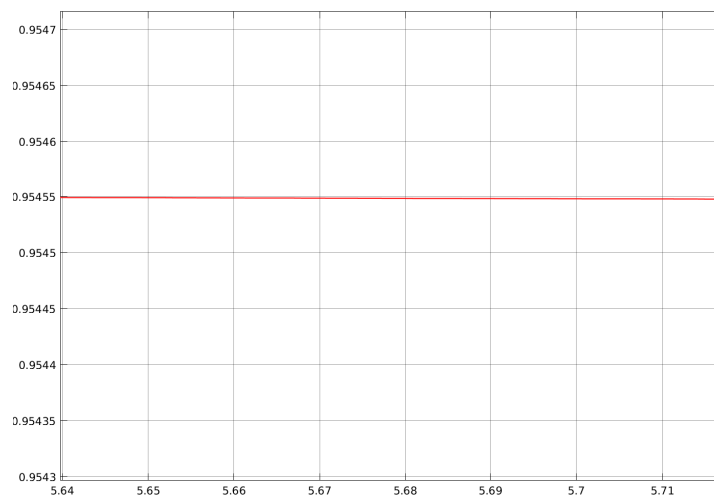


Figure 8: Disturbed system response for Question 4

This shows that the system settles around 0.95455.

Therefore, the steady state error is  $0.95455 - 1 = -0.04545$ , or a  $-4.545\%$  error.

## Question 5

Keeping the disturbance the same, add the integral term to your controller (using the integral Simulink blocks) and tune the P, I and D gains to obtain the desired performance and remove the steady state error even under the disturbance effect. Report the current  $K_p$ ,  $K_d$  and  $K_i$  values together with your new plot.

### Solution

First, we edit our MATLAB script so that it looks as follows.

```

1 % Code for HW6 of RBE500
2
3 clear; close all; clc;
4
5 % System model
6 J = 2;
7 B = 0.5;
8
9 % Controller
10 K_p = 11;
11 K_d = 8;
12 K_i = 20;
13
14 % Disturbance
15 D = 0.5;

```

Now we edit our Simulink block diagram so that it looks as follows (Figure 9).

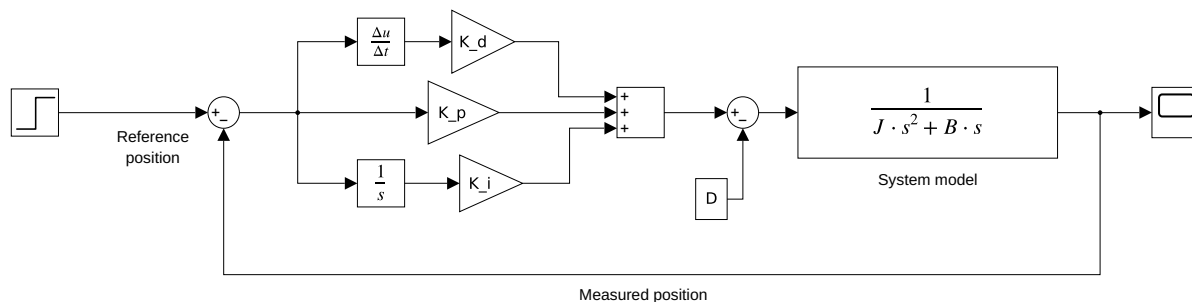


Figure 9: Block diagram with PID controller

When we plot the output of this current system, we get the plot shown on the next page (Figure 10).

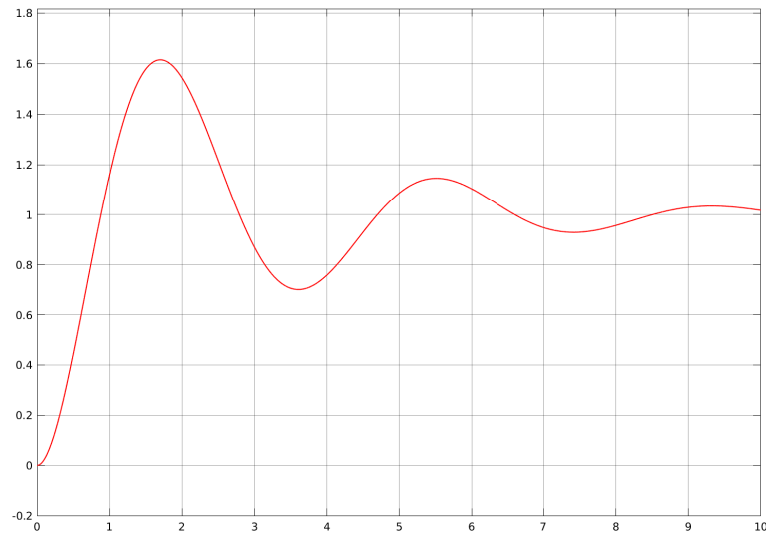


Figure 10: System response for PID controller in Question 5

Now we begin tuning the gain values for the controller.

- Upon increasing  $K_i$  to 30, the oscillation amplitude increased and the settling time become more delayed.
- Upon decreased  $K_i$  to 15, we shrunk our oscillations and dereased settling time, but to performance was still poor.
- Upon decreasing  $K_i$  to 1, we removed oscillations but now saw a positive error in steady state.
- Upon decreasing  $K_i$  to 0.75, we saw a lower postive steady-state error with no oscillation.
- For  $K_i = 6$  we had negligible steady state error and no oscillation. However our settling time was now 2.15 seconds.
- We now start to slightly increase  $K_p$ . We find that increasing in increments of 0.1 improved our results, until 11.6, which put our settling time right before two. Hence, we chose 11.55 as a satisfactory value for  $K_p$  because it puts our settling time at 2.01 seconds.

Therefore, our new tuned values are  $K_p = 11.55$ ,  $K_d = 8$ , and  $K_i = 0.6$ .

The plots on the next page show the new overall system response (Figure 11) and the zoomed-in plot to show the settling time (Figure 12).

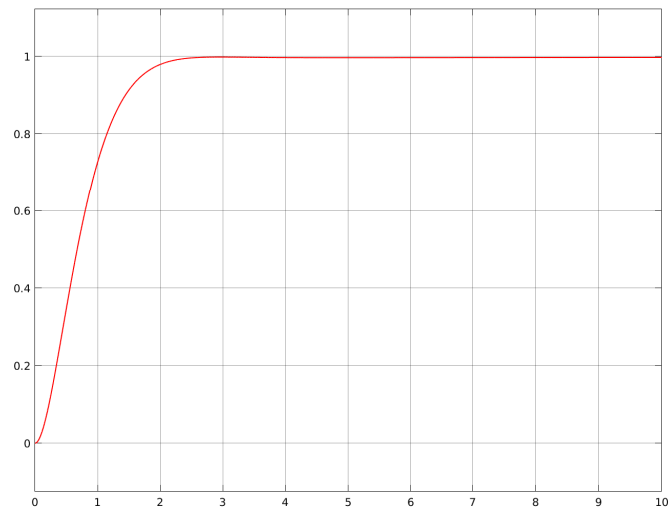


Figure 11: Tuned system response for PID controller

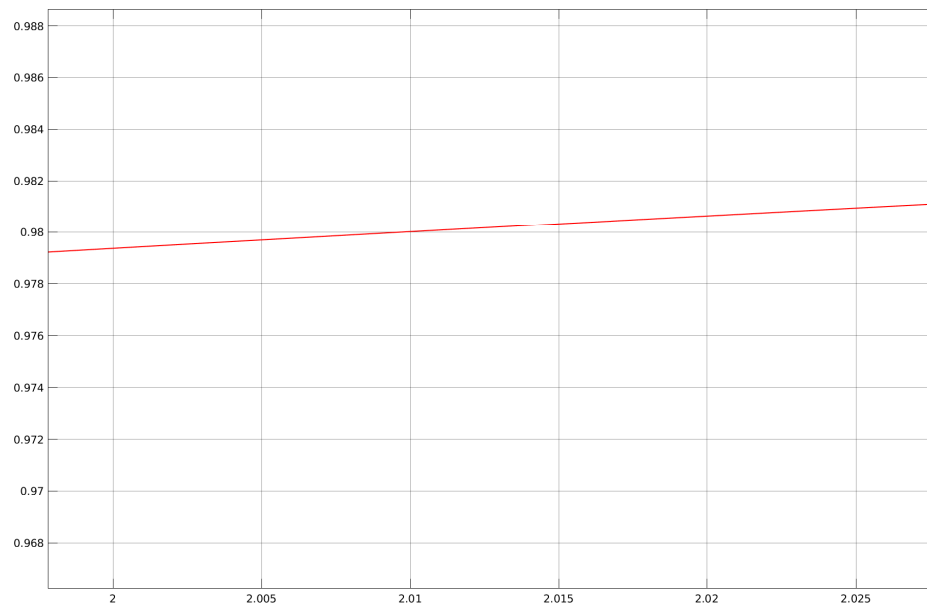


Figure 12: Zoomed in version of Figure 11, showing settling time of 2.01 seconds