Control of a Flexible Joint

Cours : Automatique (Fall 2019)

Name :	Gaspard Leroy	Sciper : 287178
Name :	Filip Slezak	Sciper : 286557

Section : MT

Module 1: Analysis of Feedback Control Systems

1.2: Give the transfer function of the flexible joint G(s) in zpk format :

$$\text{zpk}(G) = \frac{5609}{s\,(s+50.79)\,(s^2+3.211s+215.5)}$$

1.3.1: Write the transfer function S between r and e as a function of Dc and G and give its numerical values in zpk format.

$$\text{zpk}(S) = \text{zpk}\left(\frac{1}{1+D_cG}\right) = \frac{s\,(s+50.79)\,(s^2+3.211s+215.5)}{(s+50.7)\,(s+1.058)\,(s^2+2.237s+209.1)}$$

1.3.2: Write the transfer function U between r and u as a function of Dc and G and give its numerical values in zpk format.

$$\text{zpk}(U) = \text{zpk}\left(\frac{D_c}{1+D_cG}\right) = \frac{2\,s\,(s+50.79)\,(s^2+3.211s+215.5)}{(s+50.7)\,(s+1.058)\,(s^2+2.237s+209.1)}$$

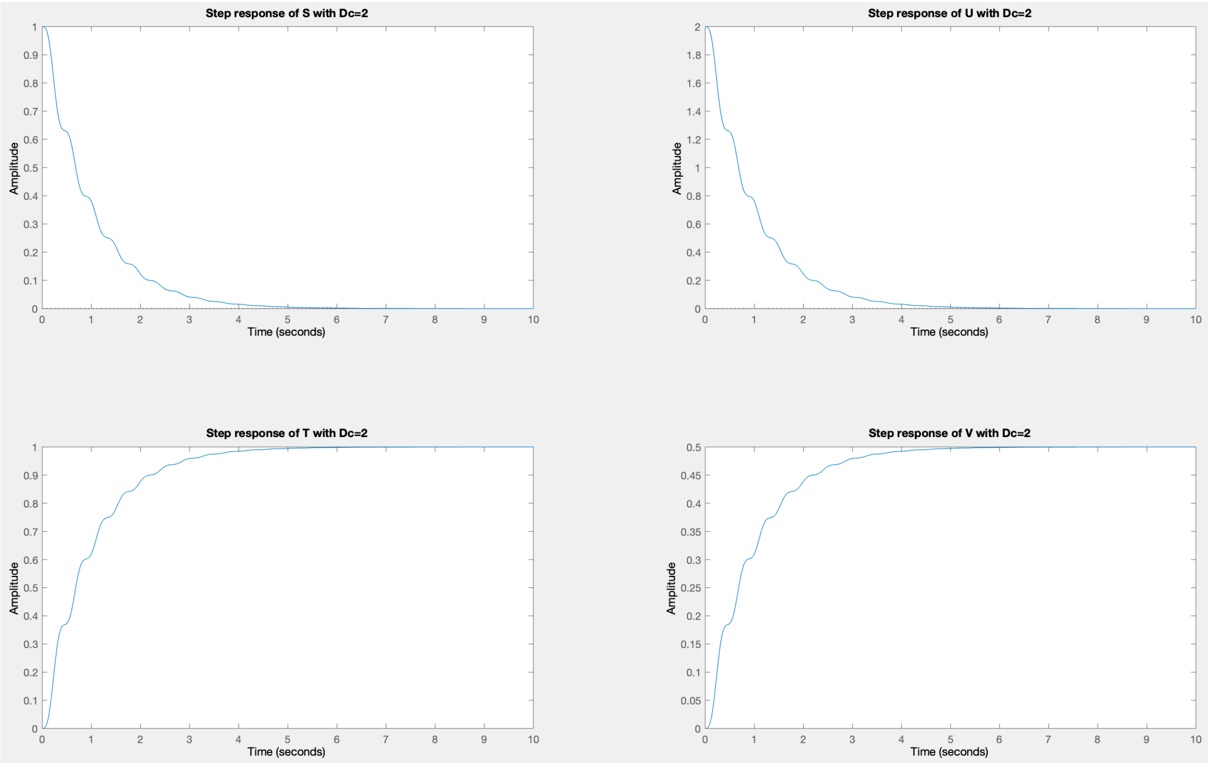
1.3.3: Write the transfer function T between r and y as a function of Dc and G and give its numerical values in zpk format.

$$\text{zpk}(T) = \text{zpk}\left(\frac{D_cG}{1+D_cG}\right) = \frac{11218\,s\,(s+50.79)\,(s^2+3.211s+215.5)}{s\,(s+50.79)\,(s+50.7)\,(s+1.058)\,(s^2+2.237s+209.1)\,(s^2+3.211s+215.5)}$$

1.3.4: Write the transfer function V between w and y as a function of Dc and G and give its numerical values in zpk format.

$$\text{zpk}(V) = \text{zpk}\left(\frac{G}{1+D_cG}\right) = \frac{5609\,s\,(s+50.79)\,(s^2+3.211s+215.5)}{s\,(s+50.79)\,(s+50.7)\,(s+1.058)\,(s^2+2.237s+209.1)\,(s^2+3.211s+215.5)}$$

1.3.a: Plot the step responses of the closed-loop system from reference signal to the output, from reference signal to the control signal (plant input), from reference signal to the tracking error signal (the input of the controller) and from disturbance signal to the output. All plots in one figure with appropriate scale (Use a 2 by 2 subplot).



1.3.b: Give the closed-loop poles.

The closed-loop poles of S and U are :

```
-50.7020 + 0.0000i
 -1.1186 +14.4185i
 -1.1186 -14.4185i
 -1.0579 + 0.0000i
```

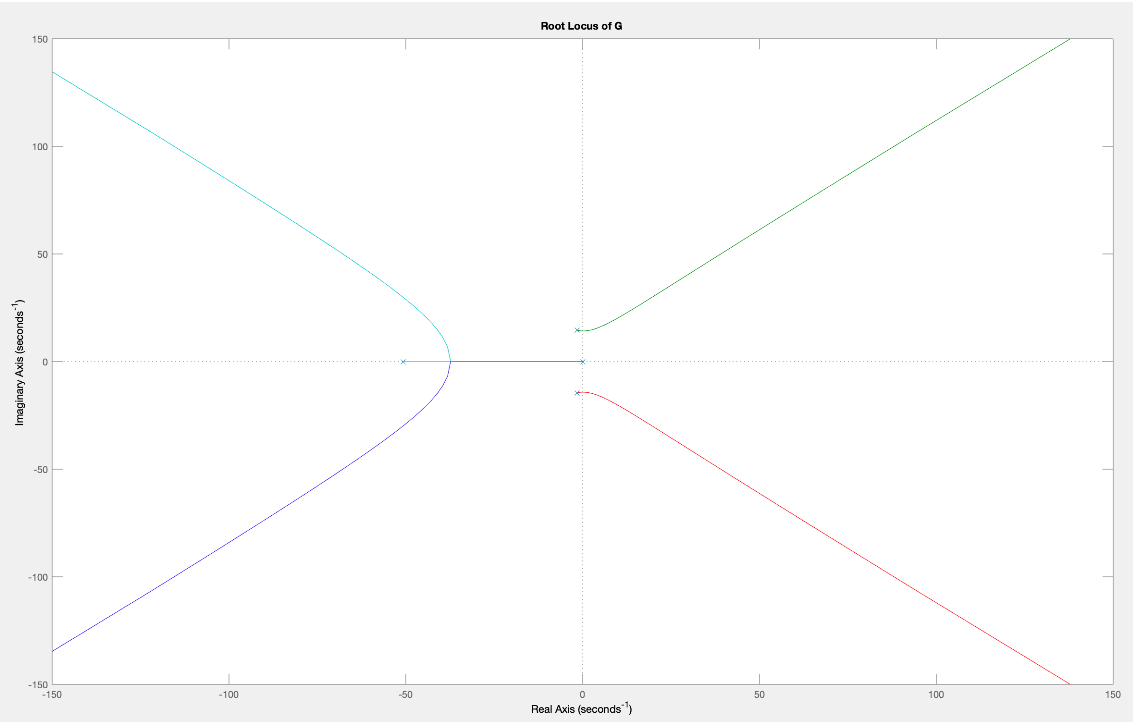
The closed-loop poles of T and V are :

```
  0.0000 + 0.0000i
-50.7863 + 0.0000i
-50.7020 + 0.0000i
 -1.6054 +14.5925i
 -1.6054 -14.5925i
 -1.1186 +14.4185i
 -1.1186 -14.4185i
 -1.0579 + 0.0000i
```

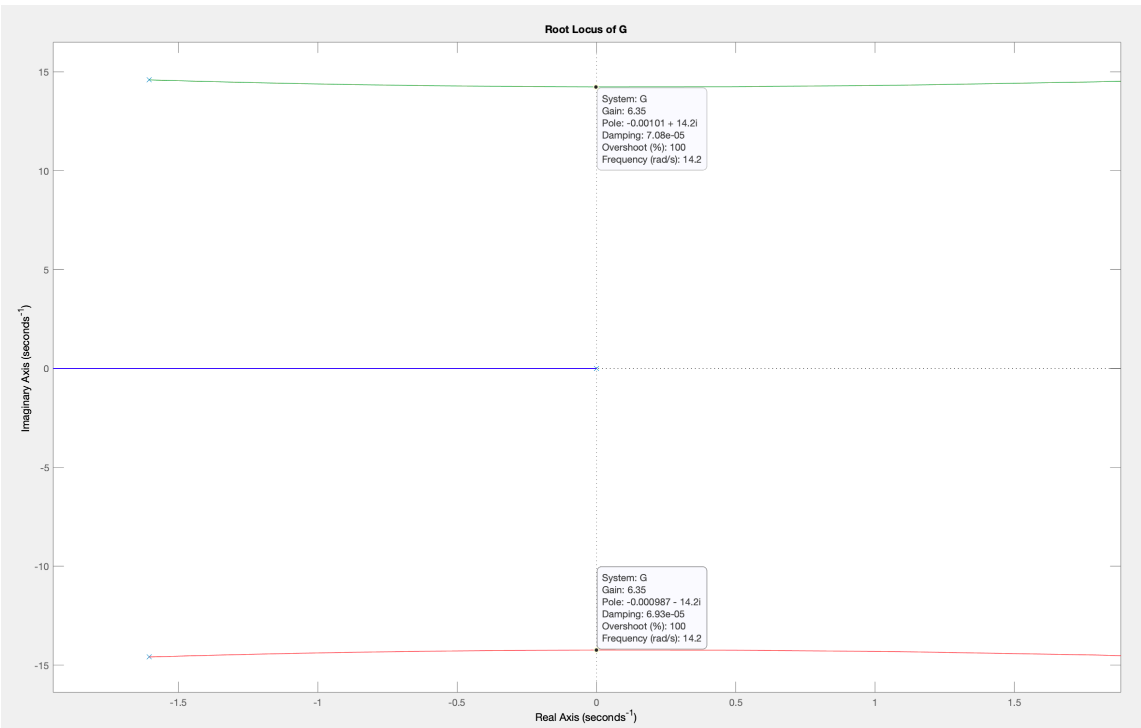
1.3.c: Is the closed-loop system stable? Why?

The pole at the origin is compensated by a zero so the closed-loop system is stable because all the other poles of the transfer functions have negative real parts, and are thus in the LHP.

1.4: Compute the ultimate gain using the plot of rlocus command of Matlab. Validate your result using the Routh stability criterion.



When zooming in this ***rlocus*** graph we can find out the ultimate gain $K_u = 6.35$.



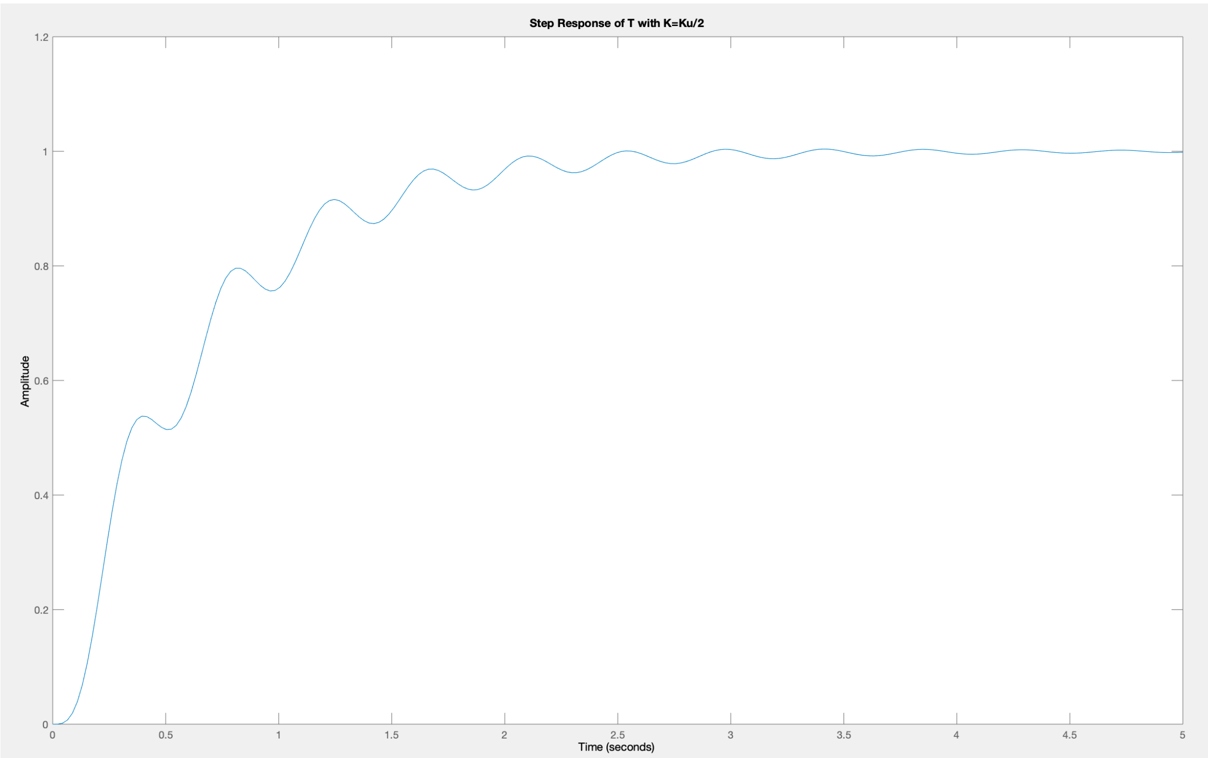
To validate the result we can use the Routh stability criterion. We evaluate the closed-loop poles of the system given by the roots of $1 + D_c G$ where $D_c = K_u$. The numerator is the polynomial $A(s) = s^4 + a_1 s^3 + a_2 s^2 + a_3 s + a_4 = s^4 + 54s^3 + 378.6s^2 + 10946s + 35620$.

Routh array from $A(s)$:

n = 4	1	378.6	35620
3	54	10946	0
2	175.8963	35620	
1	10.6930	0	
0	35620		

All the elements of the 1st column are positive; therefore, all the roots of $A(s)$ and poles of $1 + D_c G$ are in the LHP and the system is stable.

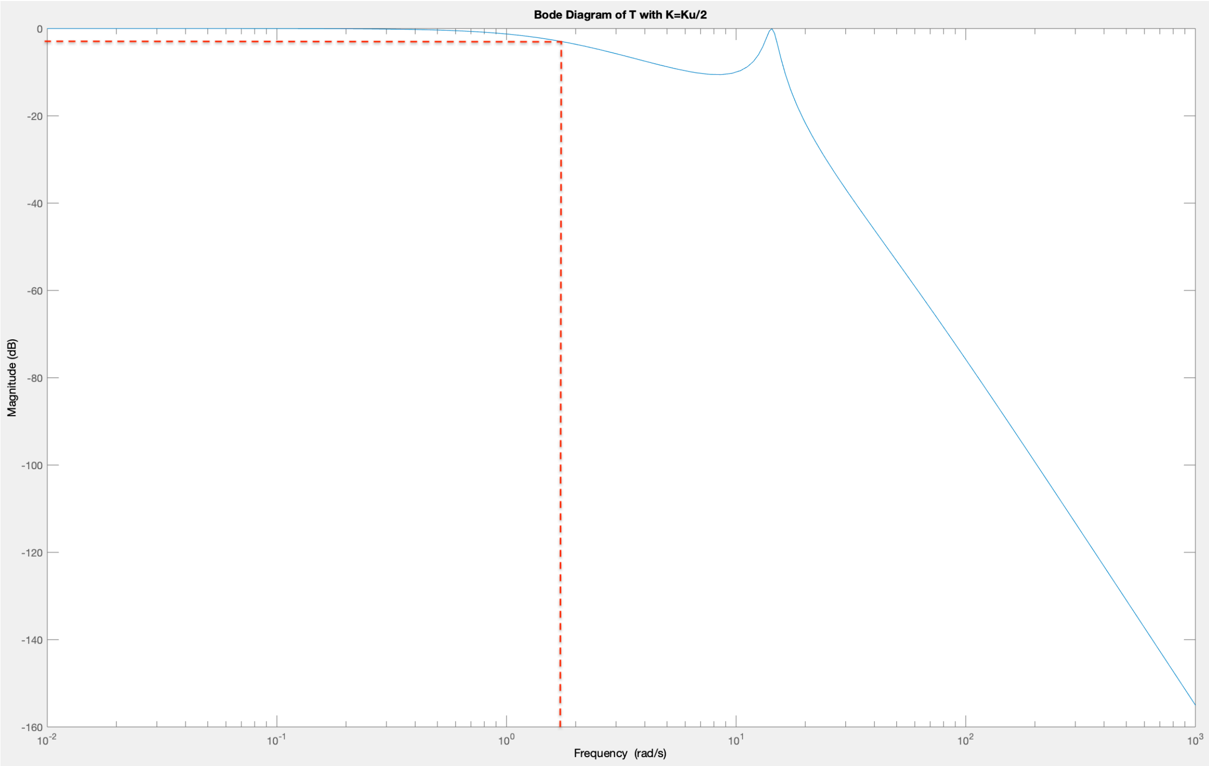
1.5: Plot the step response of the closed-loop system (between r and y). Print the rise-time, settling time and the overshoot (use stepinfo). Compute the closed-loop bandwidth (use bandwidth). Plot the magnitude Bode diagram of the closed-loop transfer function (use bodemag) and check the correctness of the bandwidth.



Using ***stepinfo*** we get :

```
        RiseTime: 1.0369
    SettlingTime: 2.7827
     SettlingMin: 0.8738
     SettlingMax: 1.0040
       Overshoot: 0.3961
      Undershoot: 0
            Peak: 1.0040
        PeakTime: 3.4117
```

The closed-loop bandwidth determined using ***bandwidth*** is 1.7483 rad/s .

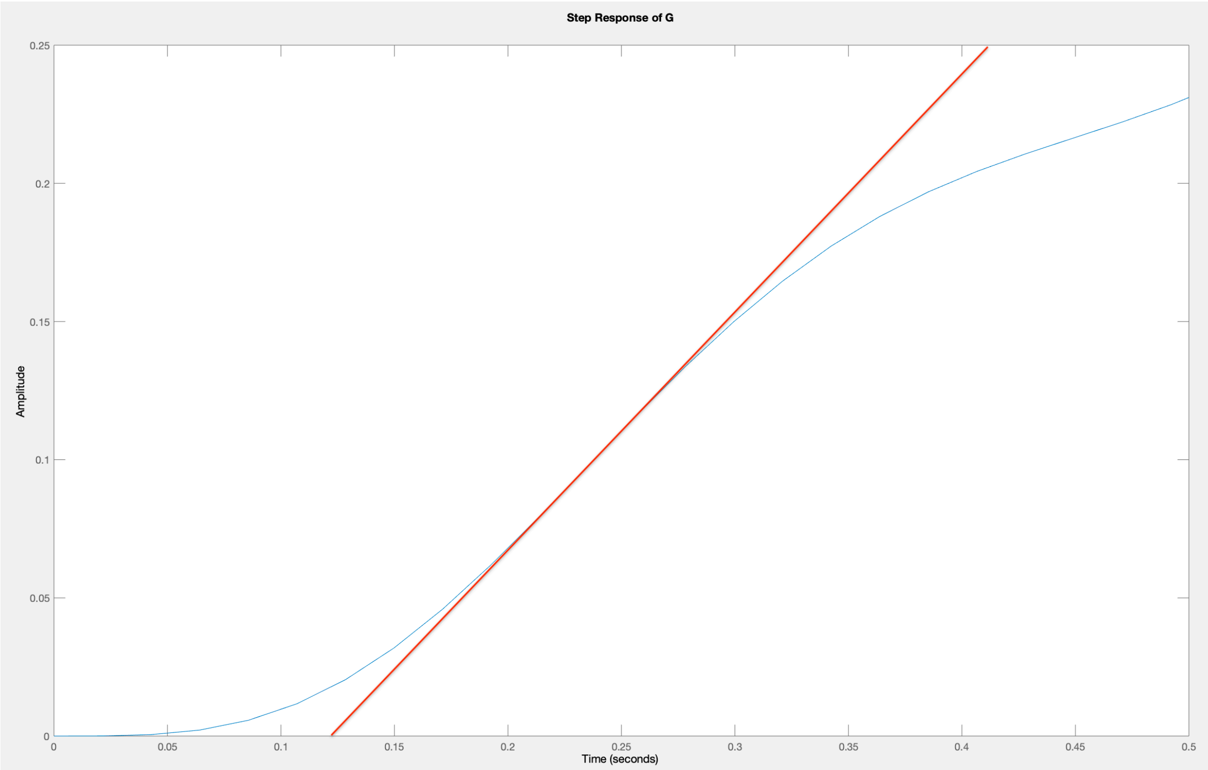


We see that this value corresponds to the frequency at the magnitude of $-3\,\mathrm{dB}$, from the steady state gain (here $0\,\mathrm{dB}$), on our bode diagram which is around $1.8\,\mathrm{rad/s}$.

Module 2: PID Controller Design

2.1 ZN First method

2.1.1: Plot the step response of G from 0 to 0.5 s together with the asymptote (or the tangent with the largest slope) for the first method of ZN.



The tangent with the largest slope is represented in red.

2.1.2: Give the value of L and R.

The intersection of the tangent with the time-axis is at $L = 0.12$, and we compute the tangent's slope $R = 0.87$.

2.1.3: Give the parameters of the PID controller.

The PID controller parameters we get using the 1st method of ZN for a mixed structure are :

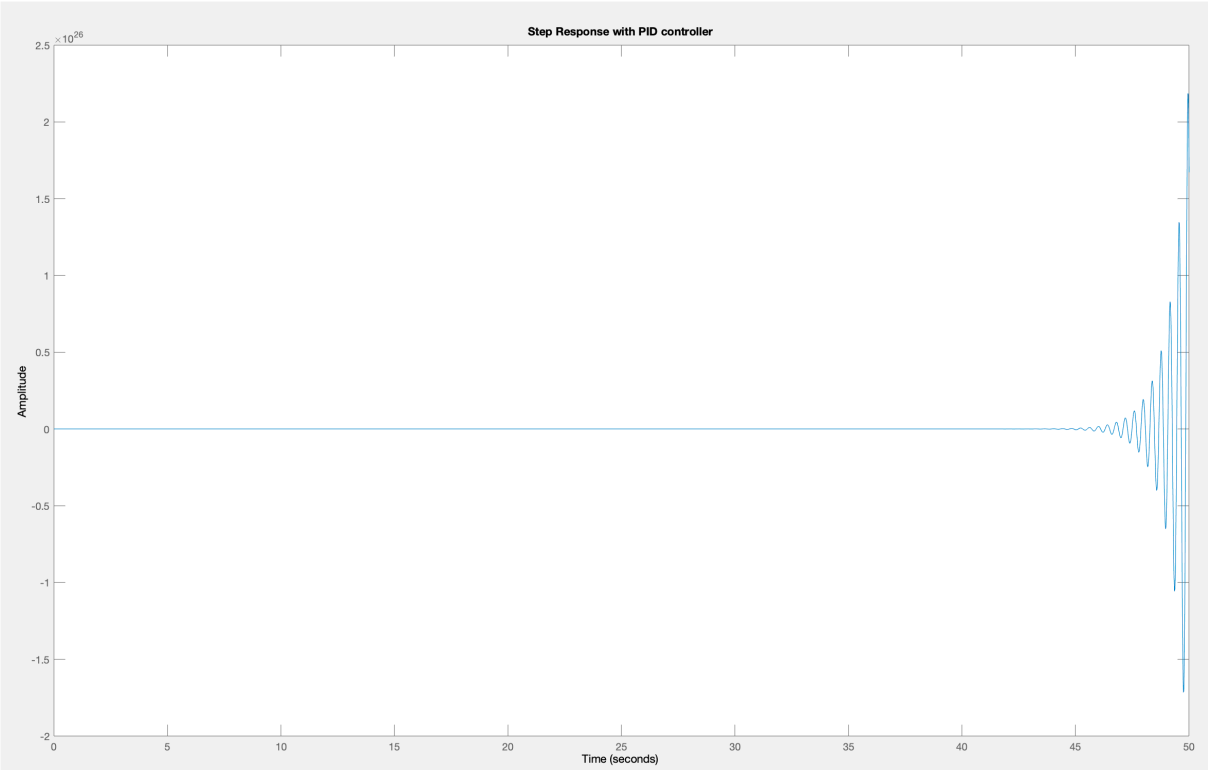
$$K_p = \frac{1.2}{RL} = 11.49$$

$$T_i = 2L = 0.24$$

$$T_d = 0.5L = 0.06$$

2.1.4: Is the closed-loop system stable with this PID controller?

The system's closed-loop transfer function is $T = \dfrac{D_c G}{1 + D_c G}$ with $D_c = K_p\,(1 + \dfrac{1}{T_i s} + T_d s)$.



We see that the closed-loop system, with this PID controller, tuned using the first ZN method, is not stable for a step input.

2.2 ZN Second method

2.2.1: Give the value of the ultimate gain and the ultimate period.

The ultimate gain $K_u = 6.35$ occurs at $f = 14.2$ rad/s (cf. root locus of G in question 1.4) and we compute the ultimate period $P_u = \frac{2\pi}{f} = 0.44$ s .

2.2.2: Give the parameters of the PID controller.

The PID controller parameters we get using the 2nd method of ZN for a mixed structure are :

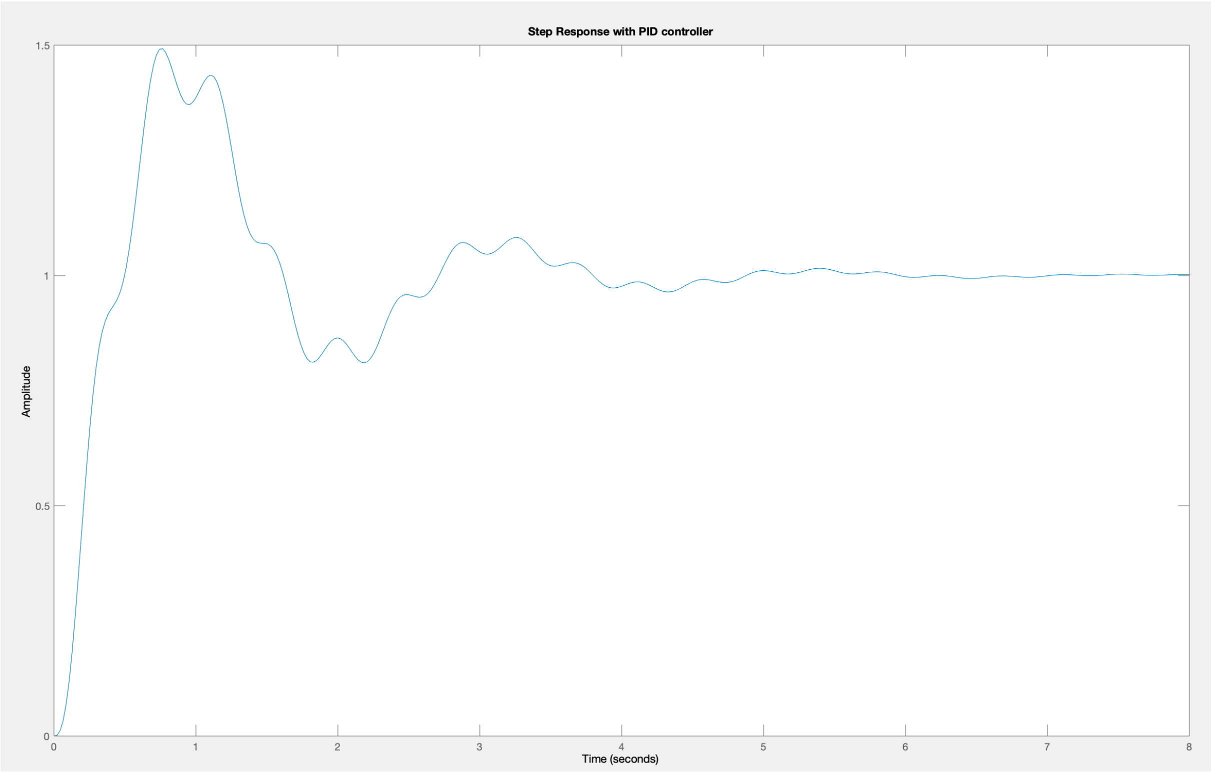
$$K_p = 0.6K_u = 3.81$$

$$T_i = 0.5P_u = 0.22$$

$$T_d = 0.125P_u = 0.055$$

2.2.3: Plot the step response of the closed-loop system from the reference signal to the output.

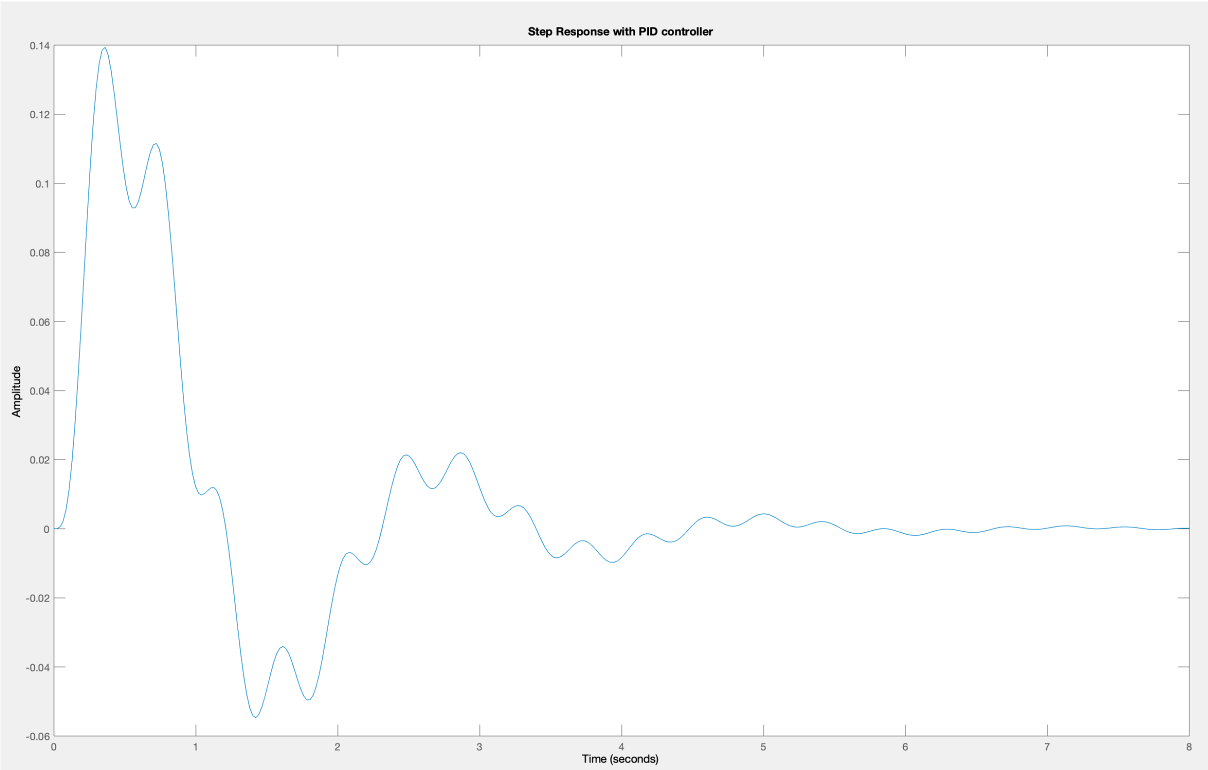
The system's closed-loop transfer function is $T = \frac{D_cG}{1 + D_cG}$ with $D_c = K_p(1 + \frac{1}{T_is} + T_ds)$.



We see that the closed-loop system, with this PID controller, tuned using the second ZN method, is stable when tracking a step input.

2.2.4: Plot the step response of the closed-loop system from the disturbance signal to the output.

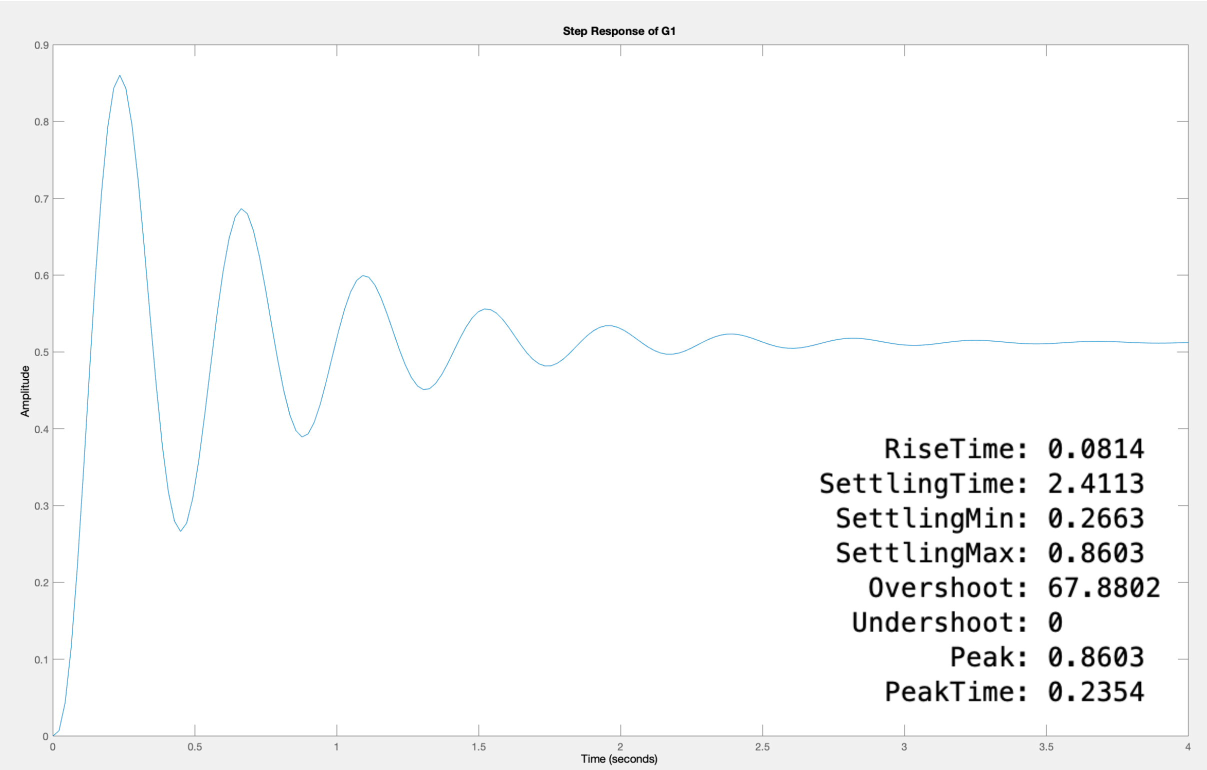
The system's closed-loop transfer function is $V = \dfrac{G}{1 + D_c G}$ with $D_c = K_p(1 + \dfrac{1}{T_i s} + T_d s)$.



We see that the closed-loop system, with this PID controller, tuned using the second ZN method, rejects a step disturbance.

2.3 Cascade Controller

2.3.1: Plot the step response of G1 and give the step information. From the step response, estimate the steady-state gain, the damping factor and the natural frequency of an approximate second-order model.



The peak value $y_{max} = 0.86$ is attained at $t_p = 0.24$ s while the steady state gain is $y_\infty = 0.51$.

From there we compute the overshoot M_p , the damping factor ζ and the natural frequency ω_n.

$$M_p = \frac{y_{max} - y_\infty}{y_\infty} = 0.68 \text{ which corresponds to the value given by } \textbf{\textit{stepinfo}}$$

$$\zeta = \sqrt{\frac{\ln(M_p)^2}{\pi^2 + \ln(M_p)^2}} = 0.12$$

$$\omega_n = \frac{\pi}{t_p\sqrt{1-\zeta^2}} = 13.75 \text{ rad/s}$$

2.3.2: Give the reference model for the inner loop. Give the PID controller parameters of the inner loop.

We approximate G_1 by the second order model $\frac{\gamma\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$ where $\gamma = y_\infty$.

Therefore the reference model $M(s)$ to use for the inner loop is the one proposed in the 4th case in chapter 4 (lecture slide 54).

The reference model is $M(s) = \dfrac{1}{\tau_m s + 1}$.

The bandwidth determined using ***bandwidth*** is 50 rad/s, therefore $\tau_m = \dfrac{1}{50} = 0.02$ rad/s.

From $k_p = \dfrac{2\zeta}{\gamma\omega_n\tau_m}$, $k_I = \dfrac{1}{\gamma\tau_m}$ and $k_D = \dfrac{1}{\gamma\omega_n^2\tau_m}$we obtain the PID controller parameters :

$$K_p = k_P = 1.7$$

$$T_i = \frac{k_P}{k_I} = 0.02$$

$$T_d = \frac{k_D}{k_P} = 0.3$$

2.3.3: Give the reference model for the outer loop. Give the proportional controller for the outer loop.

The inner loop is so much faster dynamics than the outer loop that we can ignore it in the design of the outer loop controller.

We make a first ordre approximation and choose the reference model $M = \dfrac{1}{\tau_m s + 1}$.

The closed-loop transfer function between reference signal and the output is $T = \dfrac{D_c/s}{1 + D_c/s}$.

The bandwidth determined , again, using ***bandwidth*** is 5 rad/s, therefore $\tau_m = \dfrac{1}{5} = 0.2$ rad/s.

We now have to solve $M = T$ for $K_p = D_c$ and the resulting solution is $K_p = 5$.

2.3.4: Give the transfer function between R and Y in terms of Dc, Dc' and G. Give also the transfer function with its numerical values in zpk format.

The transfer function is $T = \dfrac{D_c \frac{D_c' G_1}{1 + D_c' G_1} \frac{1}{s}}{1 + D_c \frac{D_c' G_1}{1 + D_c' G_1} \frac{1}{s}} = \dfrac{1}{1 + \frac{1 + D_c' G s}{D_c D_c' G}}$

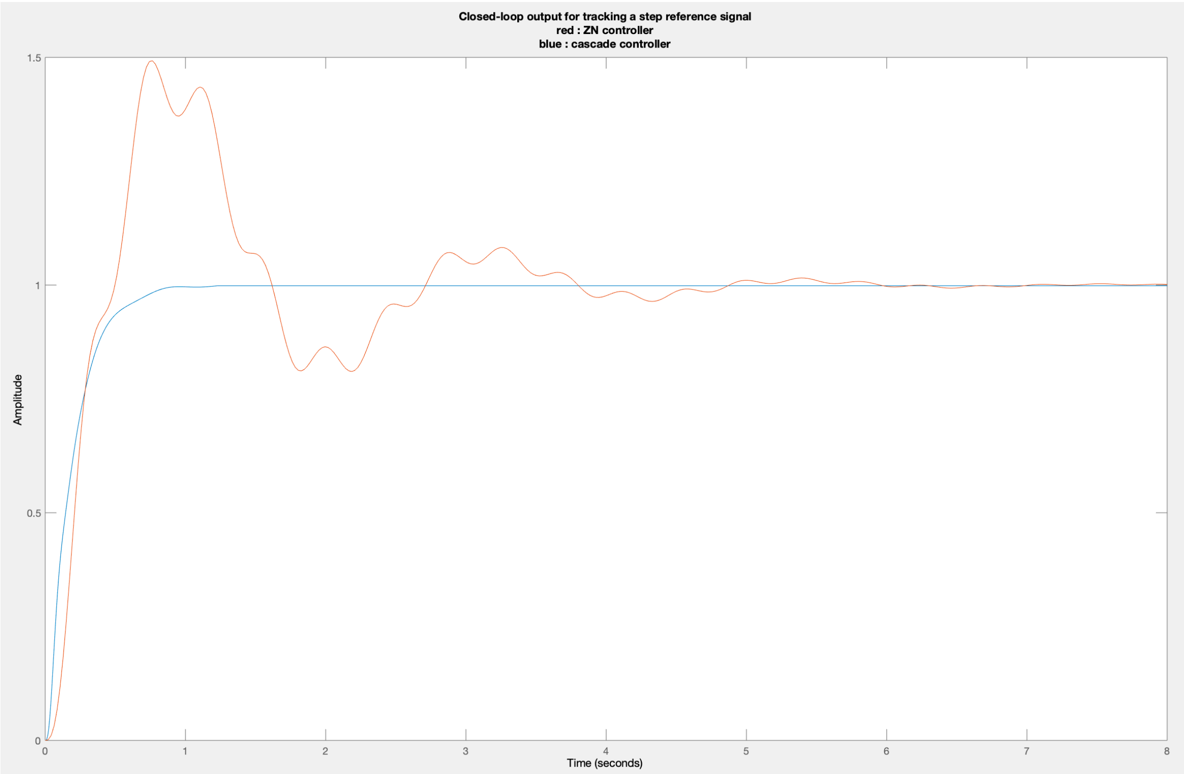
$$\text{zpk}(T) = \frac{14528\, s^2 (s + 50.79)\,(s^2 + 3.274s + 189.3)\,(s^2 + 3.211s + 215.5)}{s^2 (s + 50.79)\,(s + 5.541)\,(s^2 + 3.734s + 186.4)\,(s^2 + 3.211s + 215.5)\,(s^2 + 44.72s + 2662)}$$

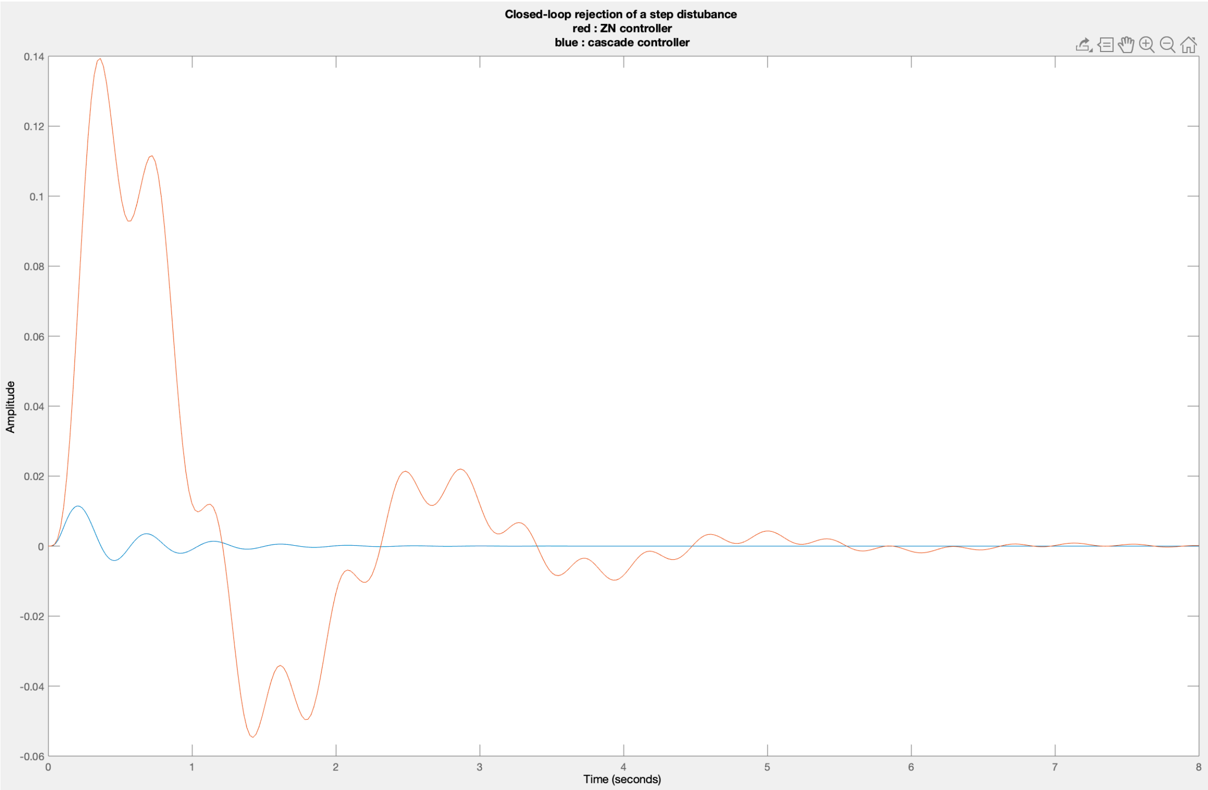
2.3.5: Give the transfer function between W and Y in terms of Dc, Dc' and G. Give also the transfer function with its numerical values in zpk format.

The transfer function is $V = \dfrac{\frac{G_1}{s}}{1 + D_c D_c' \frac{G_1}{s}\left(\frac{1}{D_c/s} + 1\right)} = \dfrac{G}{1 + D_c D_c' G\left(\frac{1}{D_c/s} + 1\right)}$

$$\mathrm{zpk}(V) = \frac{5609\, s^2 (s + 50.79)\,(s^2 + 3.211s + 215.5)}{s\,(s + 50.79)\,(s + 5.542)\,(s^2 + 3.767s + 186.2)\,(s^2 + 3.211s + 215.5)\,(s^2 + 44.69s + 2664)}$$

2.3.6: Plot the closed-loop output for tracking a step reference signal for the cascade controller and the ZN controller in the same figure. Plot the closed-loop output for the rejection of a step disturbance for the cascade controller and the ZN controller in the same figure.





We see that the cascade controller (blue) is much better in both tracking and disturbance rejection. There are fewer oscillations, little overshoot and it has a great settling time. This explains why the cascade controller is so dominant in the industry.

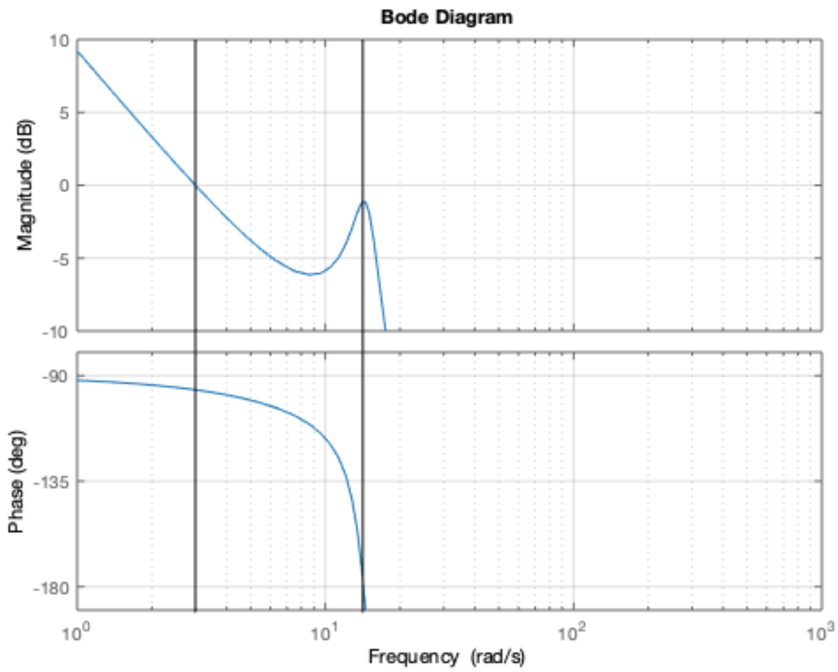
Module 3: Loop Shaping Method

3.1 (Proportional controller)

3.1.1: Give the value of kP and explain how you compute it from the Bode diagram of G. Give the values of gain margin and phase margin using the Bode diagram of the open-loop transfer function. Check your results using the command margin of Matlab.

First we need to plot the bode diagram and find out the current gain at the desired crossover frequency $\omega_c = 3\,\text{rad/s}$, which is $-15\,\text{dB}$. Therefore we need to shift the curve by $+15\,\text{dB}$ which is our k_P. After conversion $k_P = +\,15\,\text{dB} = 10^{+15/20} = 5.62$.

We define our P controller as $D_c = K_P = k_P$. And plot the bode diagram of D_cG.



The phase margin is the phase that supplements the current phase at the crossover frequency where the magnitude is $0\,\text{dB}$. In equation form it can be written as $\text{PM} = \text{phase}(\omega_c) + 180°$, which from the diagram results in $\text{PM} \approx 85°$.

The gain margin is the gain difference in dB at the frequency where the phase is at $-180°$. In equation form it can be written as $\text{GM} = 0\,\text{dB} - 20\log\left(\text{mag}(\omega_{cr})\right)$, which from the diagram is somewhere close to $\text{GM} \approx 1.2\,\text{dB}$.

The ***margin*** command returns $\text{GM} = 1.1303\,\text{dB}$ and $\text{PM} = 83.9488°$ which is fairly close to our estimates.

3.2 Lead-Lag controller

3.2.1: Give the controller and explain in details how did you compute it.

The lead-lag controller is defined as $D_c(s) = K\frac{1}{s^l}\prod_{i=1}^{m}\frac{1+\alpha_i\tau_i s}{1+\tau_i s}$. Now, let's design it.

1) Determine how many integrators we need and compute K.

We want perfect rejection of a step disturbance so we need one integrator, so $l = 1$.

K is then computed using the final value theorem for a steady-state error of 0.08.

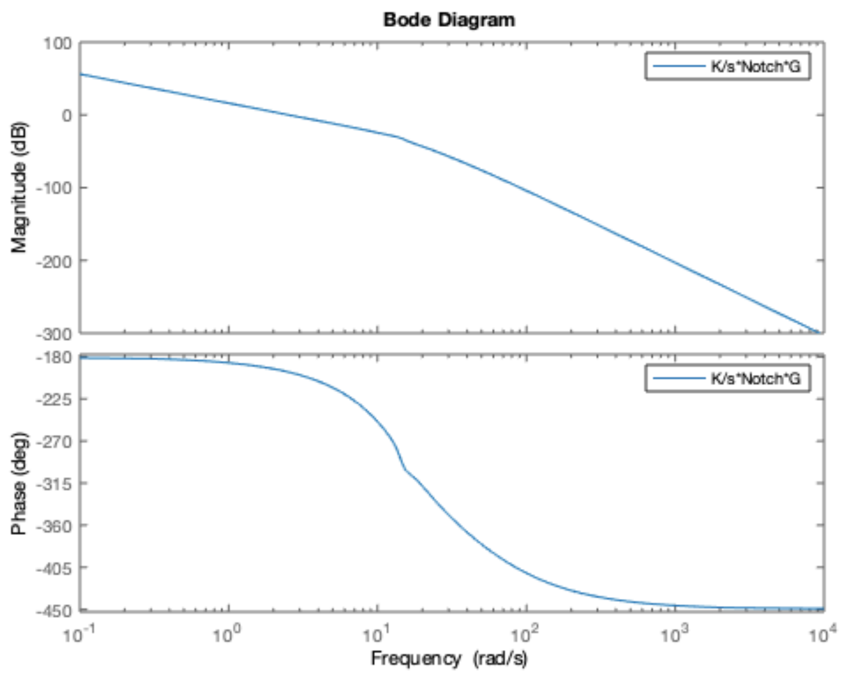
$$\lim_{s\to 0} sE(s) = \lim_{s\to 0} sV(s)W(s) = \lim_{s\to 0} s\frac{G}{1+GD_c}\frac{1}{s^2} = 0.08 \iff K = 12.5$$

Where V is the transfer function between E and W of the system.

2) Add a notch filter to damp the resonant frequency.

The notch filter's general equation is $C(s) = \frac{s^2 + 2\zeta_1\omega_1 s + \omega_1^2}{s^2 + 2\zeta_2\omega_2 s + \omega_2^2}$.

After tuning of the parameters we decide to keep $\zeta_1 = 0.125$ and $\zeta_2 = 0.7$ at $\omega_1 = \omega_2 = 15\,\text{rad/s}$ which is the frequency of the resonant peak; and get the following bode diagram.



3) Determine whether we need a lead or lag controller.

From the bode we determine that at the desired crossover frequency $\omega_c = 5\,\text{rad/s}$, the gain is $-11.85\,\text{dB}$ and the phase is $-212.78°$. To get that ω_c we need to shift the magnitude curve upwards by $11.85\,\text{dB}$. We also want a phase margin $\text{PM} = 50°$ so we need to shift the phase upwards by $82.78°$ and for that we need a lead compensator.

4) Design the lead compensator.

The phase shift we need is too big to be done by a single compensator. Let's divide it into two 41.39° phase contributions and for simplicity do the same with the magnitude and make two 5.925 dB contributions.

We need to determine α and τ, for that we define $c = 10^{\frac{5.925}{20}2} = 3.91$, the square of the magnitude contribution, and $p = \tan(41.39°) = 0.88$, the tangent of the phase contribution.

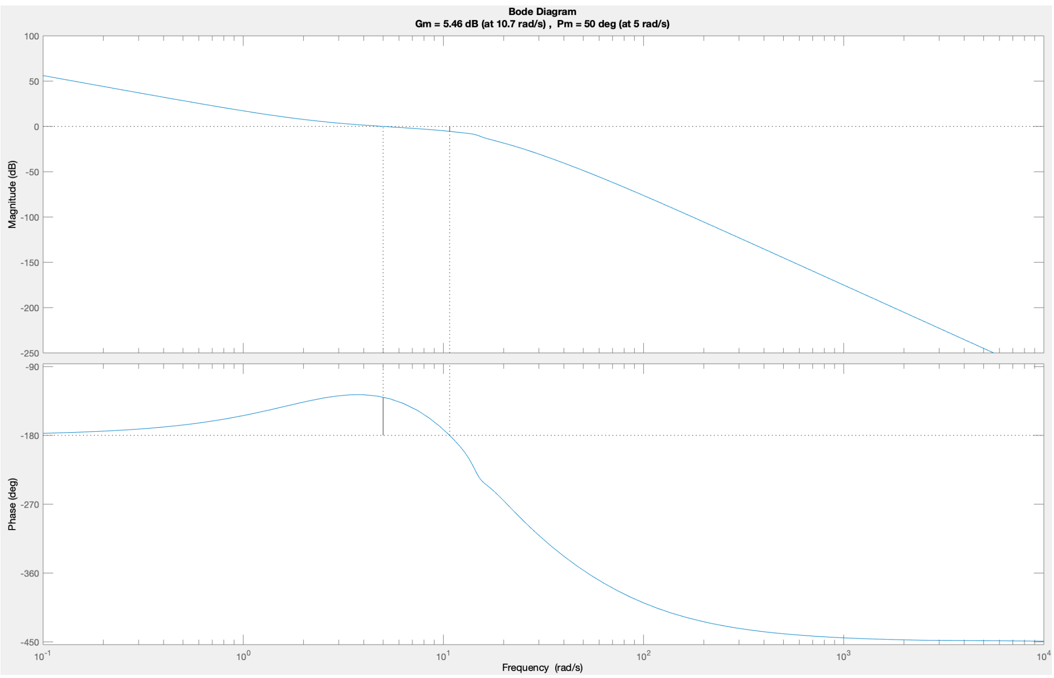
In the case of a lead compensator $\alpha > 1$ and to determine it we need to solve the equation $\left(p^2 - c + 1\right)\alpha^2 + 2p^2c\alpha + p^2c^2 + c^2 - c = 0$ which results in $\alpha = 5.02$.

As for τ, we get it directly from the formula $\tau = \dfrac{1}{\omega_c}\sqrt{\dfrac{1-c}{c-\alpha^2}} = 0.074$.

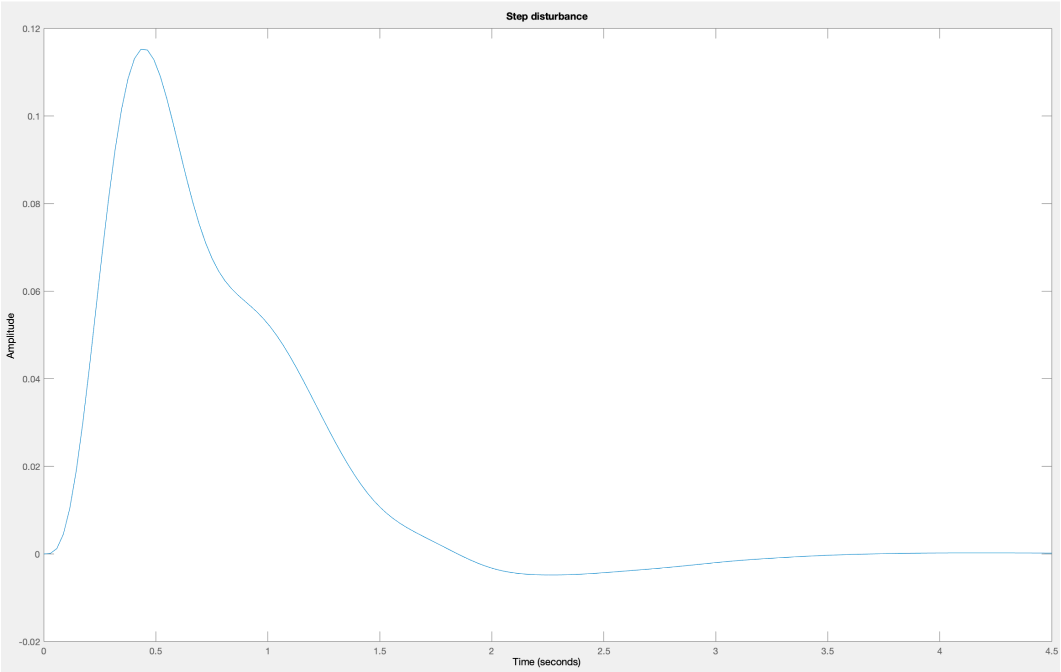
The lead compensator is then $D_c(s) = \dfrac{K}{s}\left(\dfrac{1+\alpha\tau s}{1+\tau s}\right)^2 = \dfrac{12.5}{s}\left(\dfrac{1+0.37s}{1+0.074s}\right)^2$.

3.2.2: Check your results (closed-loop bandwidth, phase margin).

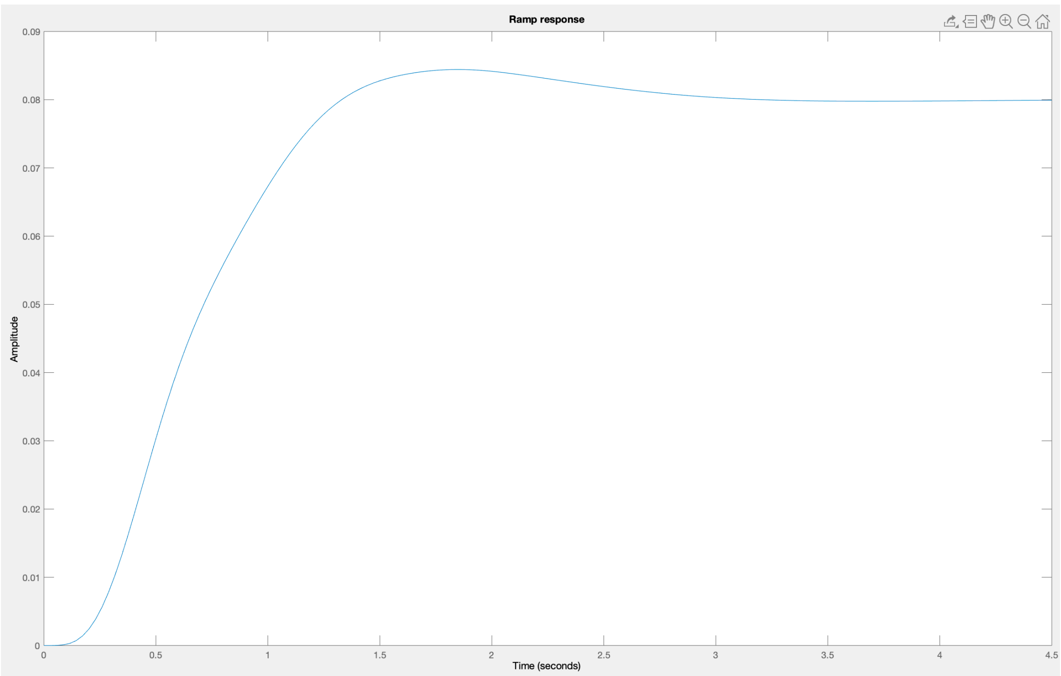
We need to verify that $\omega_c = 5$ rad/s and that PM = 50°. Then we have to make sure that the system rejects a step disturbance and has a steady state error of 0.08 (max) when suffering a ramp disturbance.



We see from this bode plot that the crossover frequency and phase margin are respected.



The system rejects the step disturbance as expected.

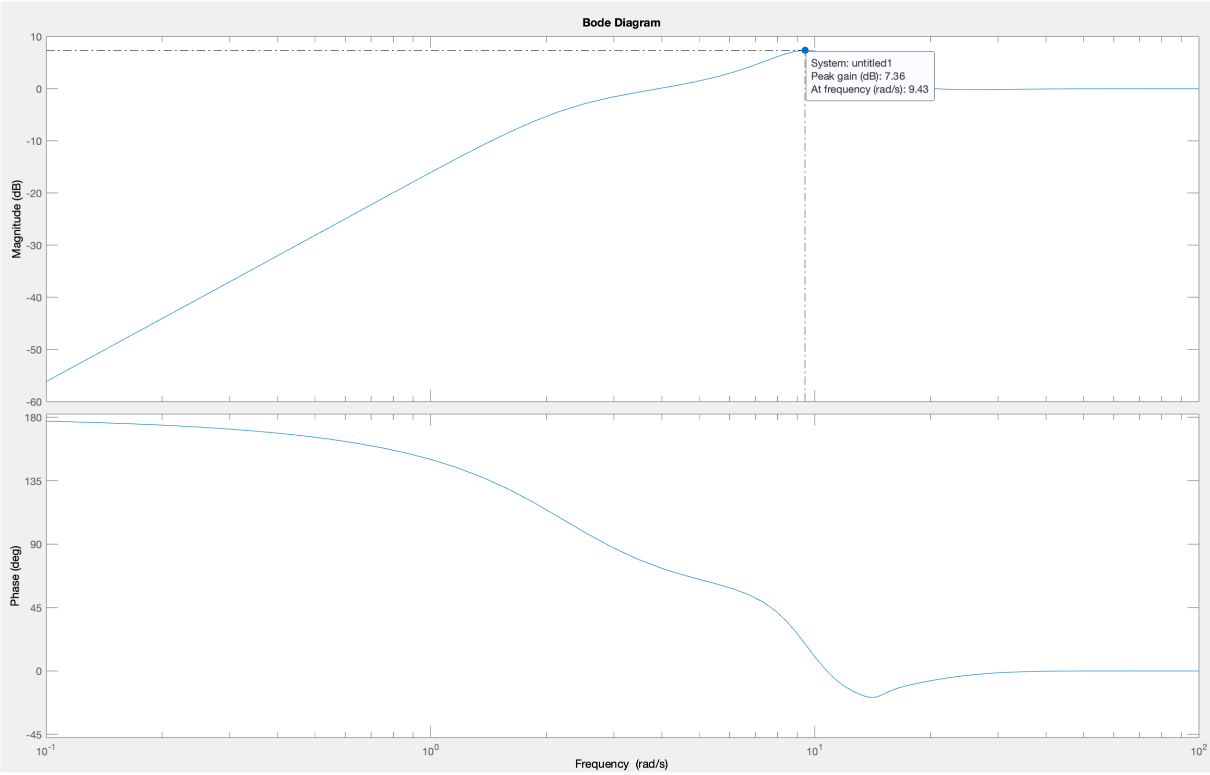


We have a steady state error of 0.08 for a ramp disturbance.

All the controller specifications are met.

3.2.3: Compute the modulus margin from the magnitude Bode diagram of the closed-loop sensitivity function.

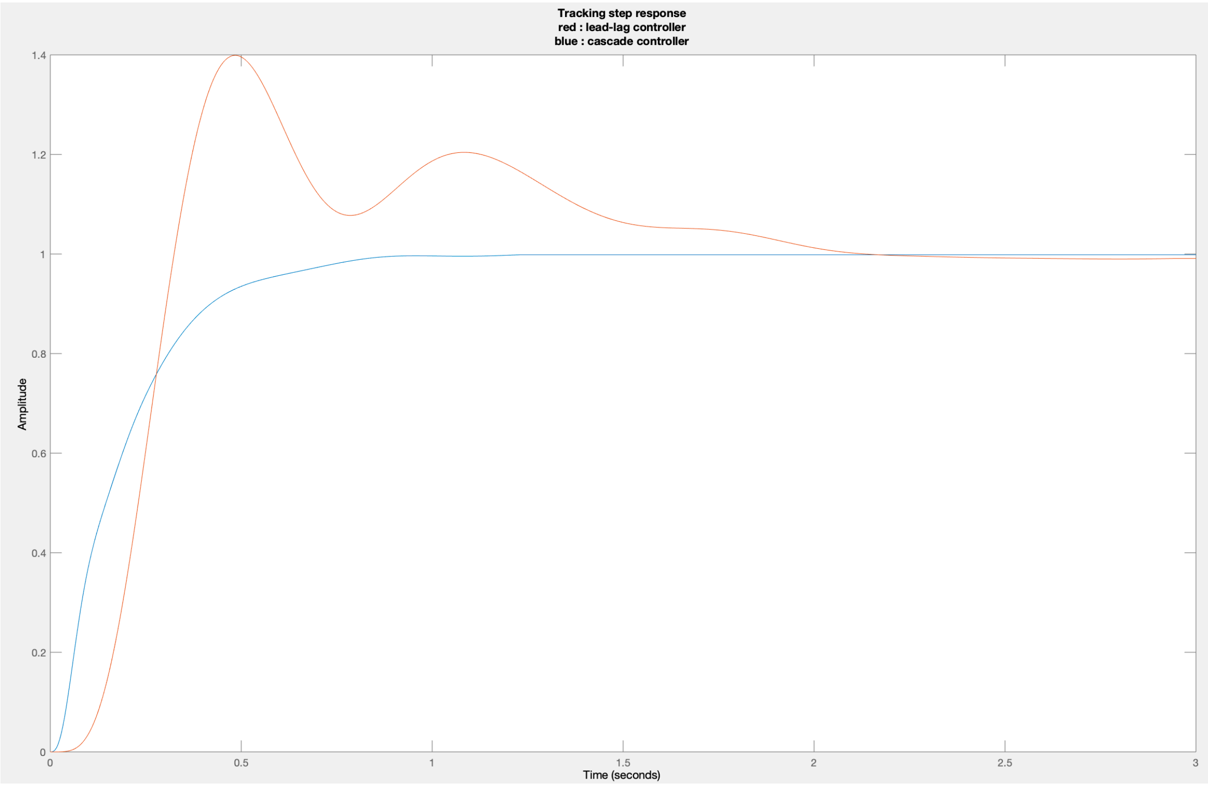
The closed-loop sensitivity function is $S = \dfrac{1}{1 + D_c C G}$ and its bode plot follows.



The peak gain is 7.36 and the modulus margin is the inverse of the peak gain, so $m = 0.13$.

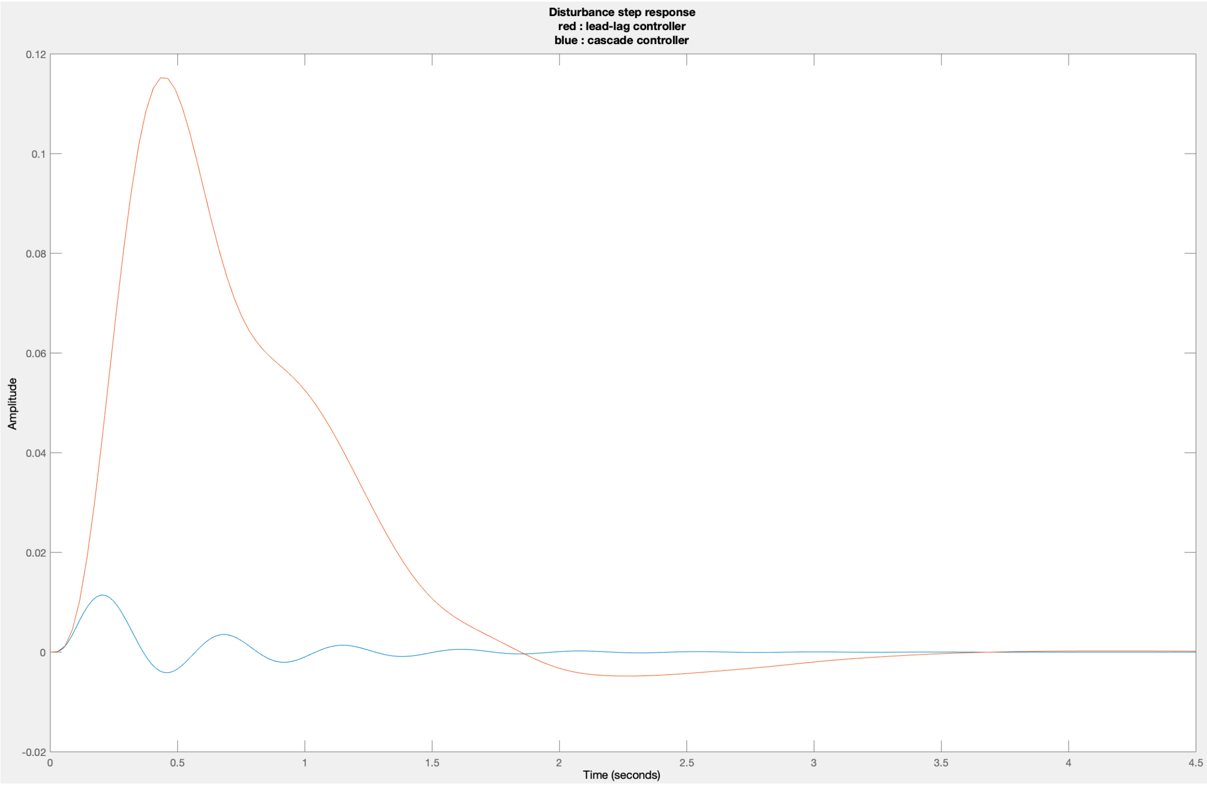
3.3 Comparison with cascade controller

3.3.1: Compare the tracking step response of the lead-lag controller and the cascade controller (plot both responses in the same figure).



The step response in tracking is better for the cascade controller (blue) compared to the lead controller (red); no overshoot, no oscillations and faster settling time are all in its favour.

3.3.2: Compare the disturbance step response of the lead-lag controller and the cascade controller (plot both responses in the same figure).



It is clear that disturbance rejection capabilities of the cascade controller (blue) are much better than the lead controller's (red). The damping is much more efficient.

Module 4: State-Space Method

4.1 State-Space Model

4.1.1: Give the state space equation of the system and the state space model in matrix form.

And the state space equations are :

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t)$$
$$y(t) = \mathbf{C}\mathbf{x}(t) + Du(t)$$

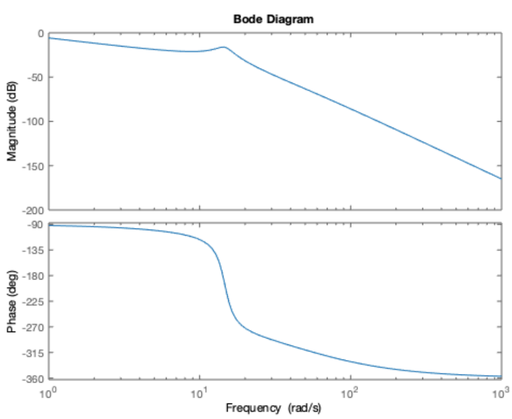
We directly compute the matrices A, B, C, D in control canonical form using ***tf2ss***.

$$\mathbf{A} = \begin{pmatrix} -54 & -379 & -10945 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \mathbf{C} = (0 \quad 0 \quad 0 \quad 5609) \qquad D = 0$$

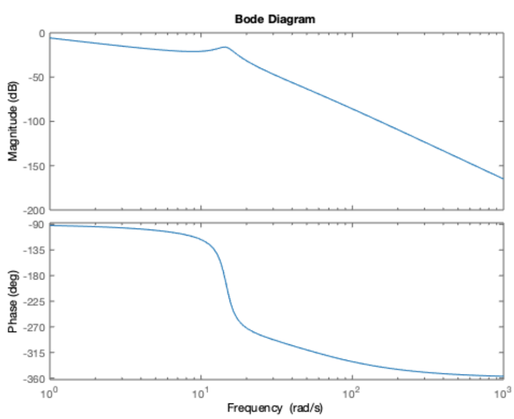
The coefficients are familiar to those we get when expanding G using ***minreal***.

4.1.2: Validate your model by comparison of the Bode diagram of the state-space model and G(s) of the first module.

Frequency Response



State Space



They are indeed identical.

4.1.3: Is the system controllable? Why?

Yes, $\det\left[\left(\mathbf{B}\ \mathbf{AB}\ \mathbf{A}^2\mathbf{B}\ \mathbf{A}^3\mathbf{B}\right)\right] = 1 \neq 0$

4.1.4: Is the system observable? Why?

Yes, $\det\left[\left(\mathbf{C}\ \mathbf{CA}\ \mathbf{CA}^2\ \mathbf{CA}^3\right)\right] = 9.90 \cdot 10^{14} \neq 0$

4.2 State-Space Controller Design

4.2.1: Give the codes for state-feedback controller design and the final numerical values of the controller K.

We set the closed-loop dynamic as being $s^2 + 2\zeta\omega_n s + \omega_n^2$ with $\omega_n = 5$ rad/s and $\zeta = 0.8$. The poles resulting from this equation are $s = -4 \pm 3j$. As we have a fourth ordre system we choose two more, much faster, poles $fp = -40$ to complete the matrix dimensions.

```
%control design by pole placement
wn = 5;
zeta = 0.8;
p3 = -40; %fast pole -> choose 10*p1,2
p4 = -40; %fast pole -> choose 10*p1,2
Pc = [roots([1 2*zeta*wn wn^2]); p3; p4]'; %Matrix 1x4
K = acker(A, B, Pc);
```

Which yields $\mathbf{K} = \begin{pmatrix} 34 & 1886 & 3855 & 40000 \end{pmatrix}$.

4.2.2: Give the codes for state estimator design and the final numerical values for L.

Again, we estimate the system by a second order model. The estimator's dynamic should be about 3-10 times faster than the controller so we chose it 5 times faster and set $\hat{\omega}_n = 5\omega_n$. From there we calculate the two poles and add another two, 10 times further.

```
%estimator design by pole placement
wne = 5*wn;  %dynamics of the estimator should be 3 to 10
             %times faster than the control dynamics
             %-> choose a factor 5
Pe = [roots([1 2*zeta*wne wne^2]); -200; -200]'; %fast poles
                                                  %-> 10*roots1,2

Lt = acker(A', C', Pe);
L = Lt';
```

The result is $\mathbf{L} = \begin{pmatrix} 3420.3 & -39 & 6.3 & 0.1 \end{pmatrix}^T$.

4.2.3: Give the code for computing the feed-forward gain and its final numerical value.

We want to design the feedforward gain in order to have zero steady state error when tracking a step reference signal. So we choose the closed-loop equation in chapter 7 (lecture slide 59) :

$$\overline{N} = -\left(\mathbf{C}_{cl}\,\mathbf{A}_{cl}^{-1}\begin{pmatrix}\mathbf{B}\\ \mathbf{B}\end{pmatrix}\right)^{-1} \text{ where } \mathbf{A}_{cl} = \begin{pmatrix}\mathbf{A} & -\mathbf{BK}\\ \mathbf{LC} & \mathbf{A}-\mathbf{BK}-\mathbf{LC}\end{pmatrix} \text{ and } \mathbf{C}_{cl} = \begin{pmatrix}\mathbf{C} & 0\end{pmatrix}$$

```
%compute the feedforward gain
Acl = [A -B*K; L*C A-B*K-L*C];
Ccl = [C zeros(1, 4)];
Nbar = -1/(Ccl*inv(Acl)*[B; B]);
```

The value we get is $\overline{N} = 7.1314$.

4.2.4: Give the state-space representation for the closed-loop system between r and y. Give the numerical values of the transfer function.

The state-space representation for the closed-loop system between r and y is :

$$\begin{aligned}
&\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t)\\
&y(t) = \mathbf{C}\mathbf{x}(t)\\
&\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}u(t) + \mathbf{L}\left(y(t) - \mathbf{C}\hat{\mathbf{x}}(t)\right)\\
&u(t) = -\mathbf{K}\hat{\mathbf{x}}(t) + \overline{N}r(t)
\end{aligned}$$

After manipulating the equations to get to the state space equations presented in 4.1.1 we get :

$$\mathbf{A}_{cl} = \begin{pmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{LC} & \mathbf{A} - \mathbf{BK} - \mathbf{LC} \end{pmatrix} \qquad \mathbf{B}_{cl} = \begin{pmatrix} \mathbf{B}\,\overline{N} \\ \mathbf{B}\,\overline{N} \end{pmatrix} \qquad \mathbf{C}_{cl} = (\mathbf{C} \quad \mathbf{0}) \qquad D_{cl} = 0$$

Finally we compute $T = \mathbf{C}_{cl}\,(s\mathbf{I} - \mathbf{A}_{cl})^{-1}\mathbf{B}_{cl}$ but to avoid numerical errors we use the ***ss*** function.

```
%Compute the transfer function between the reference signal r and the output y
T = ss(Acl, Bcl, Ccl, Dcl);
tf(T);
```

$$\text{zpk}(T) = \frac{40000\,(s+200)^2\,(s^2+40s+625)}{(s+200)^2\,(s+40)^2\,(s^2+8s+25)\,(s^2+40s+625)}$$

4.2.5: Give the state-space representation for the closed-loop system between r and u. Give the numerical values of the transfer function.

The state-space representation for the closed-loop system between r and u is :

$$\begin{aligned}
&\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t)\\
&y(t) = u(t)\\
&\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}u(t) + \mathbf{L}\left(y(t) - \mathbf{C}\hat{\mathbf{x}}(t)\right)\\
&u(t) = -\mathbf{K}\hat{\mathbf{x}}(t) + \overline{N}r(t)
\end{aligned}$$

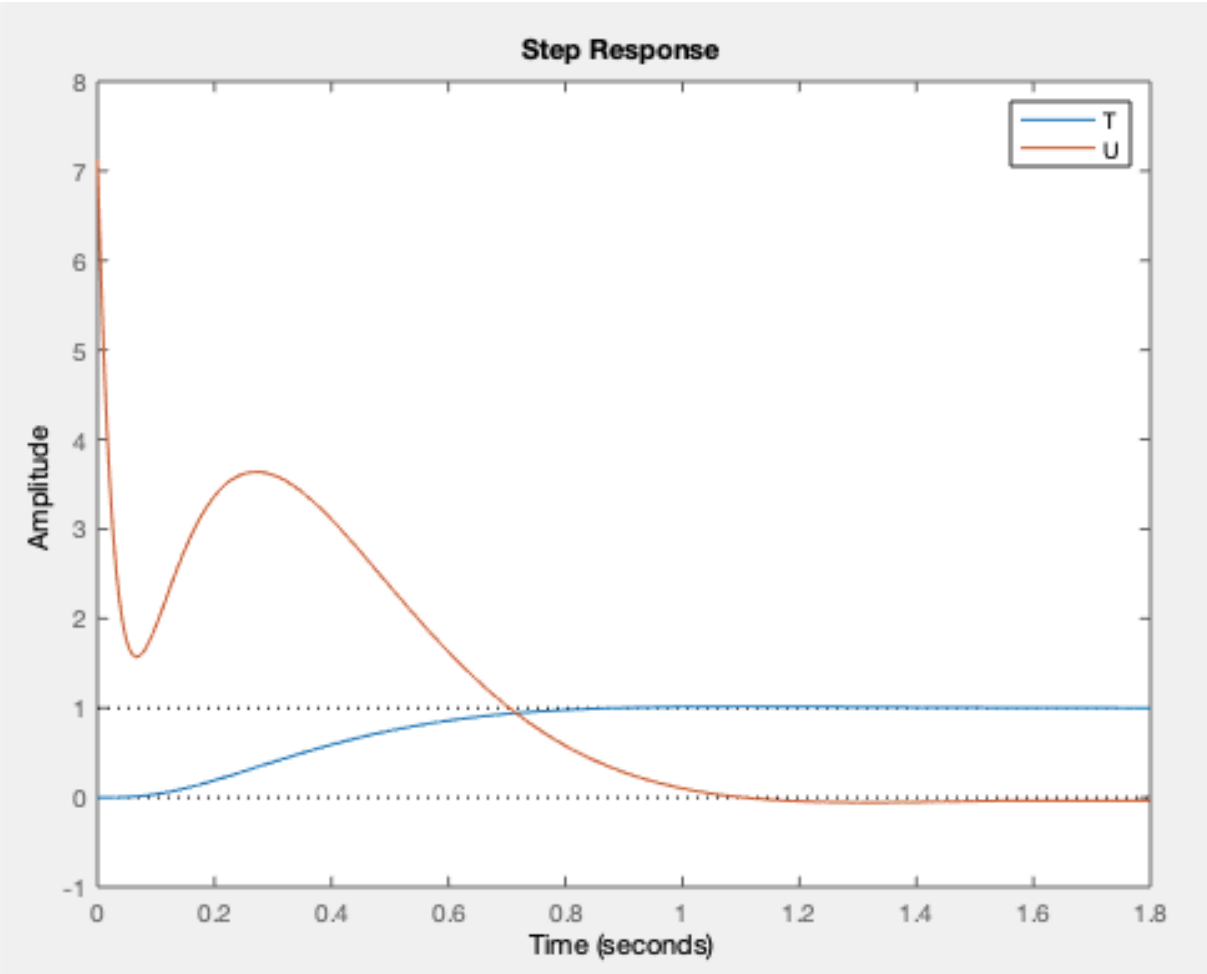
We then get the following matrices :

$$\mathbf{A}_{cl} = \begin{pmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{LC} & \mathbf{A} - \mathbf{BK} - \mathbf{LC} \end{pmatrix} \qquad \mathbf{B}_{cl} = \begin{pmatrix} \mathbf{B}\,\overline{N} \\ \mathbf{B}\,\overline{N} \end{pmatrix} \qquad \mathbf{C}_{cl} = (\mathbf{0} \quad -\mathbf{K}) \qquad D_{cl} = \overline{N}$$

Finally we compute $U = \mathbf{C}_{cl}\,(s\mathbf{I} - \mathbf{A}_{cl})^{-1}\mathbf{B}_{cl} + D_{cl}$ but to avoid numerical errors we use ***ss***, in the same manner as previously.

$$\text{zpk}(U) = \frac{7.1314\,s\,(s+200)^2\,(s+50.79)\,(s^2+3.211s+215.5)\,(s^2+40s+625))}{(s+200)^2\,(s+40)^2\,(s^2+8s+25)\,(s^2+40s+625)}$$

4.2.6: Plot the control signal u(t) and the output y(t) for a unit step reference signal.



4.3 State-Space Controller with integrator

4.3.1: Give the codes and the numerical values for the augmented model.

Adding an integrator adds the equation $\dot{x}_I(t) = -\mathbf{C}\mathbf{x} + r(t)$ to the system, the augmented matrices are :

```
%Find augmented state space model
Abar = [A zeros(4, 1); -C 0];
Bbar = [B; 0];
```

$$\overline{\mathbf{A}} = \begin{pmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{pmatrix} = \begin{pmatrix} -54 & -379 & -10945 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -5906 & 0 \end{pmatrix} \qquad \overline{\mathbf{B}} = \begin{pmatrix} \mathbf{B} \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

4.3.2: Give the codes for state-feedback controller design and the final controller K.

We use the LQR method to determine our new $\mathbf{K} = (\mathbf{K}_0 \ K_1)$ using the following code.

```
%Augmented control design by LQR
%define Q and R
rho = 100;
Q = zeros(5, 5);
Q(5,5) = rho;
R = 1;
K_augmented = lqr(Abar, Bbar,Q, R);
```

For the sake of agreement we took $\rho = 100$ but it could have been more for a better response time, finally we get $\mathbf{K} = \begin{pmatrix} 3 & 181 & 1513 & 37384 & -10 \end{pmatrix}$.

4.3.3: Give the codes for state estimator design and the final gain L.

We want to design an estimator using the pole placement method considering only the states of the plant model (no need to estimate the integrator, it is known) so we can simply take the one from 4.2.2 : $\mathbf{L} = (3420.3 \quad -39 \quad 6.3 \quad 0.1)^T$.

4.3.4: Give the state-space representation for the closed-loop system between r and y. Give the numerical values of the transfer function.

The state-space representation for the closed-loop system between r and y is :

$$\begin{aligned} &\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \\ &y(t) = \mathbf{C}\mathbf{x}(t) \\ &\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}u(t) + \mathbf{L}\left(y(t) - \mathbf{C}\hat{\mathbf{x}}(t)\right) \\ &u(t) = -\mathbf{K_0}\hat{\mathbf{x}}(t) - K_1 x_I(t) \end{aligned}$$

We then get the following matrices :

$$\mathbf{A}_{cl} = \begin{pmatrix} \mathbf{A} & -\mathbf{B}\mathbf{K}_0 & -\mathbf{B}K_1 \\ \mathbf{L}\mathbf{C} & \mathbf{A}-\mathbf{B}\mathbf{K}_0-\mathbf{L}\mathbf{C} & -\mathbf{B}K_1 \\ -\mathbf{C} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad \mathbf{B}_{cl} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} \quad \mathbf{C}_{cl} = (\mathbf{C} \quad \mathbf{0} \quad 0) \quad D_{cl} = 0$$

Which corresponds to the following transfer function T :

$$\mathrm{zpk}(T) = \frac{56090\,(s+200)^2\,(s^2+40s+625)}{(s+200)^2\,(s+50.79)\,(s^2+3.236s+5.121)\,(s^2+3.226s+215.7)\,(s^2+40s+625)}$$

4.3.5: Give the state-space representation for the closed-loop system between r and u. Give the numerical values of the transfer function.

The state-space representation for the closed-loop system between r and u is :

$$\begin{aligned} &\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \\ &y(t) = u(t) \\ &\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}u(t) + \mathbf{L}\left(y(t) - \mathbf{C}\hat{\mathbf{x}}(t)\right) \\ &u(t) = -\,\mathbf{K_0}\hat{\mathbf{x}}(t) - K_1 x_I(t) \end{aligned}$$

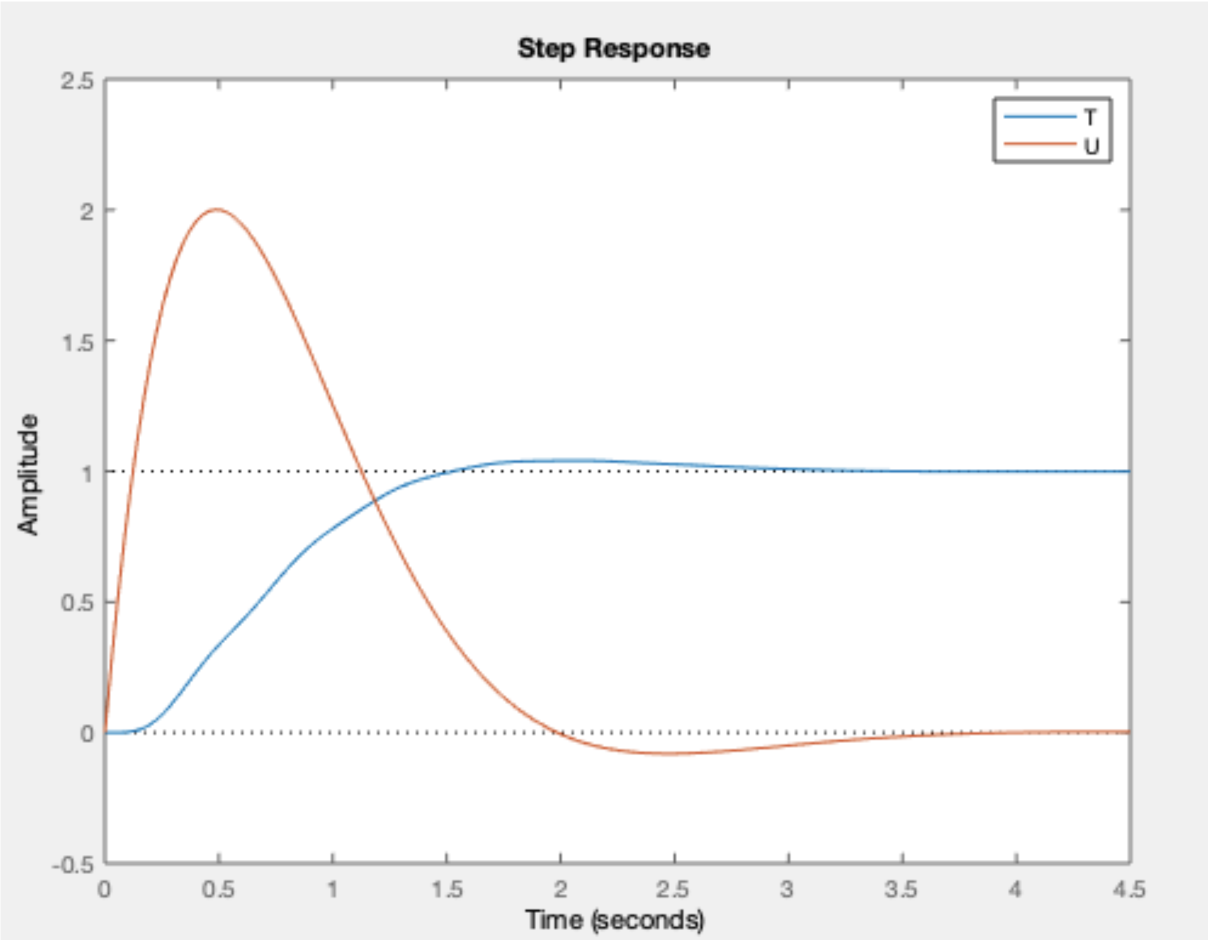
We then get the following matrices :

$$\mathbf{A}_{cl} = \begin{pmatrix} \mathbf{A} & -\mathbf{B}\mathbf{K}_0 & -\mathbf{B}K_1 \\ \mathbf{L}\mathbf{C} & \mathbf{A}-\mathbf{B}\mathbf{K}_0-\mathbf{L}\mathbf{C} & -\mathbf{B}K_1 \\ -\mathbf{C} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad \mathbf{B}_{cl} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} \quad \mathbf{C}_{cl} = \begin{pmatrix} \mathbf{0} & -\mathbf{K}_0 & -K_1 \end{pmatrix} \quad D_{cl} = 0$$

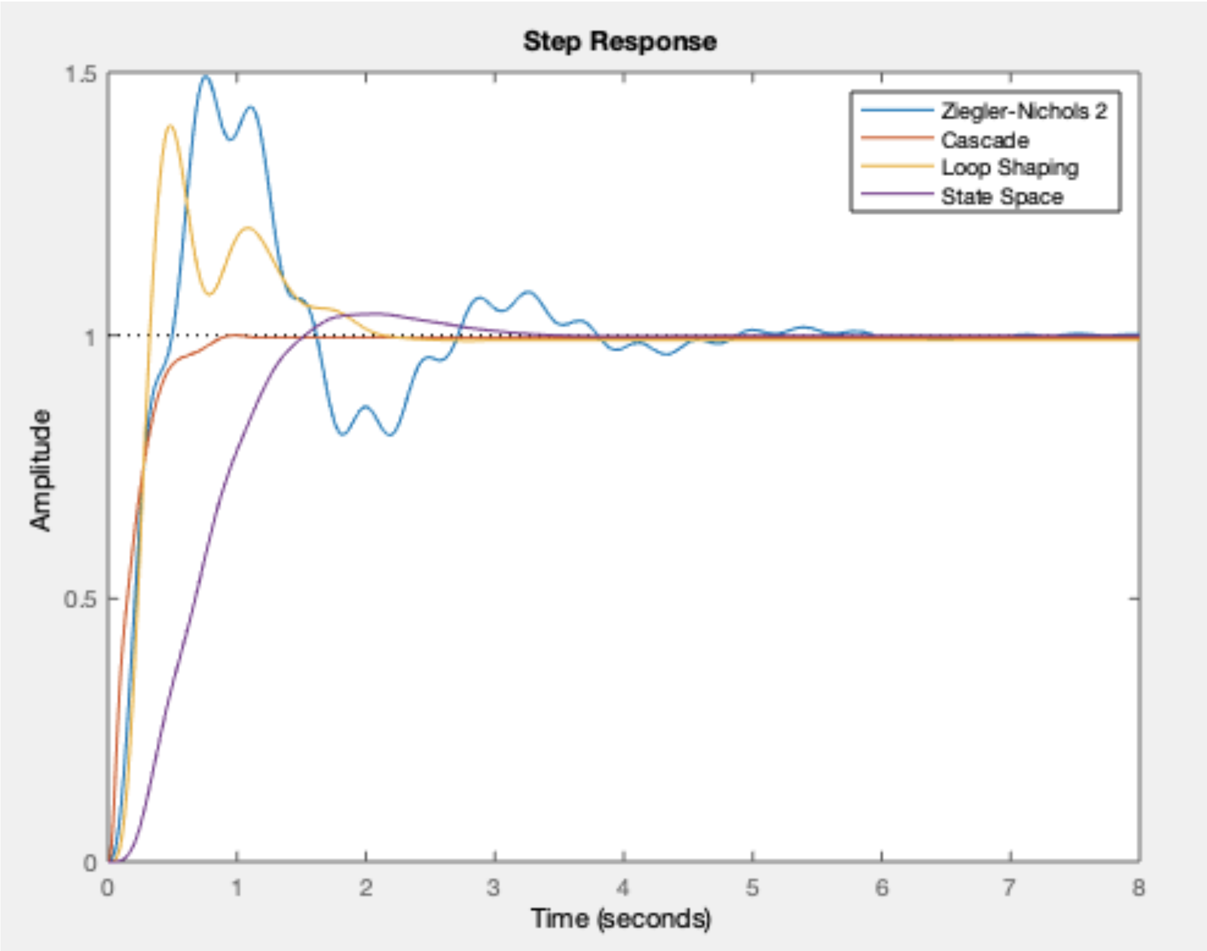
Which corresponds to the following transfer function U :

$$\mathrm{zpk}(U) = \frac{10\,s\,(s+200)^2\,(s+50.79)\,(s^2+3.211s+215.5)\,(s^2+40s+625)}{(s+200)^2\,(s+50.79)\,(s^2+3.236s+5.121)\,(s^2+3.226s+215.7)\,(s^2+40s+625)}$$

4.3.6: Plot the control signal u(t) and the output y(t) for a unit step reference signal.



4.3.7: Compare the four methods: ZN method, cascade PID, loop shaping (with integrator) and state space (with integrator) in terms of performance in tracking by superposition of the time responses.



The Ziegler-Nichols tuning method is the worst choice for this robotic arm, it needs 6s to actually settle to its final value and does so after quite a few high amplitude oscillations compared to the other methods.

The loop shaping method is second best for more or less the same reasons as Ziegler-Nichols.

As for the state space method it is great, the step response has a nice, smooth shape, it settles pretty fast and doesn't oscillate. We could tune it to be even faster by changing that ρ we set to 100, after adding the integrator, the final value won't alter.

The cascade controller beats them all. Its rise and settling times are simply excellent. Appart from that strange bump, right before reaching the final value, there is nothing to comment.

4.3.8: Compare these methods in terms of the facility and clarity of the design method, their advantages and disadvantages.

Ziegler-Nichols is very easy to compute however the performance is not so great, partly due to its origin from statistics that don’t apply to every system. It is great for forming an idea of what to expect and try to beat when designing another controller.

The loop shaping is pretty annoying to compute, there are values to gather from the bode diagram, some not so evident calculations and if we need multiple compensators it can easily get messy. Its advantage though is that we can divide and design/choose the physical control device as we like.

The cascade controller is great, its design can easily be divided into separate systems to be tuned by independently by different teams and we see from this example that its end performance is truly great. The downside is that with many loops, the division is not easy to do and the transfer functions get long and complicated.

Nowadays this is surely handled by computers in most cases so it’s not a big deal, but, as for the state space method precision errors might kick in and we must take that into account.

About the state space model, it is hard to make abstraction of the system and derive the matrices but afterwards we can pretty easily augment them when adding other factors of complexity. The computer basically doesn’t care but this also means it is restricted to computers; people prefer, and are more efficient, at tuning parameters. The good thing about state space is that the solution computed by this method is optimal, and can be easily be extended to multivariable systems.