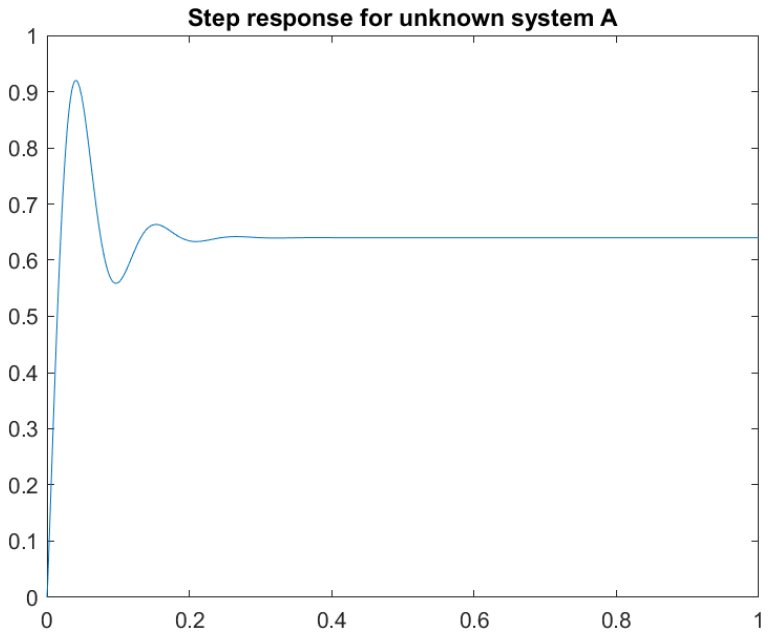
Take-Home Exam System and Control Theory 2023
Kobe Michiels (r0848543) – Transfer functie 9

Question 1 (System A)

Measure and visualize the step response and bode plot of your unknown system A.

Om de *step response* te visualiseren, leg ik een stapfunctie aan aan de input. Vervolgens plot ik de response van het Simulink-model in functie van de tijd.

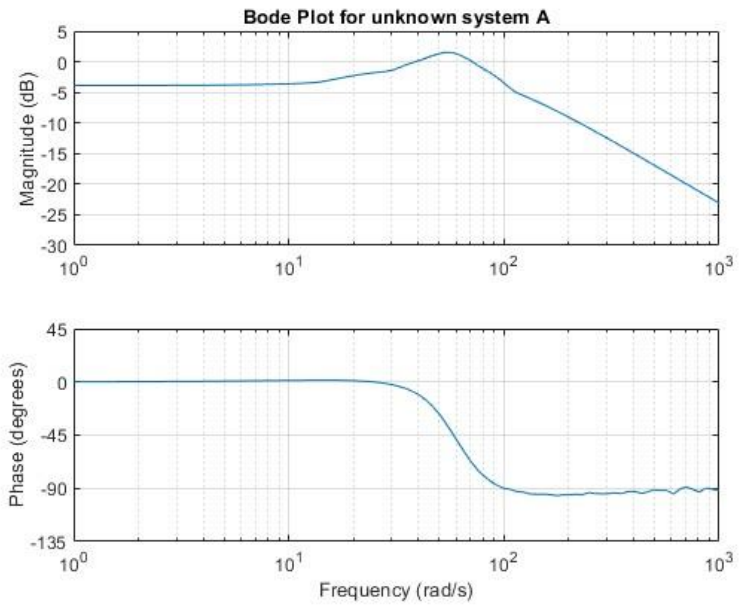
```
time_in = (0:1e-3:1)';
input = heaviside(time_in);
simOut = sim('transferfunction9A_harness.slx',time_in);
figure('Name', 'Generated step response for unknown system A');
plot(time_in,simOut.response)
title('Step response for unknown system A')
```



Om de bode plot te genereren, leg ik sinussen met verschillende frequenties aan aan de ingang. Voor elke frequentie bereken ik de magnitude en de faseverschuiving. De magnitude bereken ik als de verhouding tussen de output en de input. De fase bereken ik uit het tijdsverschil tussen de input en de output. Dit tijdsverschil bereken ik door een lokaal maxima van de input te zoeken, en vervolgens het eerst volgende lokaal maxima van de output te zoeken en dan de tijdsstippen waarop beide maxima plaatsvinden van elkaar af te trekken. Bij deze methode moet ik wel rekening houden met het gegeven dat er in het geval dat er geen faseverschil is, een faseverschil van 360° bekomen wordt doordat het volgende lokaal maxima dan 1 periode verder ligt. Wanneer ik voor alle frequenties magnitude en fase berekend heb, kan ik deze plotten ten opzichte van de frequentie, op een logaritmische schaal. Ik sla de data op in een MATLAB Data file zodat ik dit stuk code niet telkens opnieuw moet uitvoeren aangezien dit toch wel redelijk wat tijd kost.

```
% Initialisation of time vector
startTime = 0;
endTime = 50;
samplingTime = 0.0001;
time_in = (startTime:samplingTime:endTime)';

% Initialisation of frequency (in rad/s)
startFreq = 1;
endFreq = 1000;
numPoints = 100;
freq = logspace(log10(startFreq), log10(endFreq), numPoints);

% Initialisation of magnitude and phase vector to store generated data:
magnitude = zeros(1, numPoints);
phase = zeros(1, numPoints);

% Generation of Bode Plot data
for k = 1 : numPoints
    input = sin(freq(k)*time_in);
    simOut = sim("transferfunction9A_harness",time_in);
    output = simOut.response;

    % Magnitude
    magnitude(k) = max(abs(output)) / max(abs(input));

   % Phase
    % Local maximum of input after possible transition phenomena disappear:
    for i = (0.75*(length(time_in)-1)):length(time_in)
        if input(i-1) < input(i) && input(i+1) < input(i)
            input_index = i;
            input_time = input_index*samplingTime;
            break;
        end
    end

    % First next local maximum of output:
    for i = input_index:length(time_in)
        if output(i-1) < output(i) && output(i+1) < output(i)
            output_index = i;
            output_time = output_index*samplingTime;
            break;
        end
    end

    % Calculate phase with time delay:
    phase(k) = -(output_time-input_time)*freq(k)*180/pi;
    if phase(k) < -350
        phase(k) = phase(k) + 360;
    end
end
% Save data to MATLAB Data file:
save("transferfunction9A_BodePlotWorkspace.mat")
```

```
%% Generate Bode plot with MATLAB Data file data:
load("transferfunction9A_BodePlotWorkspace.mat")
figure('Name', 'Generated bode plot for unknown system A');
subplot(2,1,1);
semilogx(freq, 20*log10(magnitude));
title('Bode Plot for unknown system A')
ylabel('Magnitude (dB)');
xlim([1 1000]);
ylim([-30 5]);
yticks(-30:5:5)
grid on;
subplot(2,1,2);
semilogx(freq, phase);
xlabel('Frequency (rad/s)');
ylabel('Phase (degrees)');
xlim([1 1000]);
ylim([-135 45]);
yticks(-135:45:45)
grid on;
```



Analyze and interpret the figures. Discuss the stability (in detail), and the order.

Zoals in de opgave is aangegeven, is systeem A stabiel. Voordat het systeem zijn steady state bereikt, zijn er nog enkele oscillaties in de staprespons. Het systeem is dus ondergedempt ($0 < \zeta < 1$) en zal twee complex toegevoegde polen hebben met een reëel deel dat kleiner is dan 0. Verder zal er ook één nul aanwezig zijn. Dit leid ik af uit de fase respons die van 0° naar -90° gaat, terwijl die in het geval dat er enkel 2 polen en geen nullen waren naar -180° zou dalen. We hebben dus te maken met een tweede-orde systeem.

Question 2 (System A)

Reverse engineer the transfer function from your Bode Plot and/or step response.

In de magnituderespons is er eerst een toename van de amplitude zichtbaar, en die dan overgaat in een afname aan -20dB/dec. We weten al dat het systeem ondergedempt is en dus twee complex toegevoegde polen heeft. In de faserespons is er een overgang van 0° naar -90°. Zoals eerder aangehaald leid ik hieruit af dat ik te maken heb met een tweede orde-systeem met één nul en twee polen. De transferfunctie zal dus de volgende vereenvoudigde vorm hebben:

$T(s) = \frac{As+B}{s^2+Cs+D}$.

Ik begin met het bepalen van de constante D. Voor een tweede orde systeem is deze gelijk aan het kwadraat van de natuurlijke eigenfrequentie ${\omega_n}^2$. De fase daalt van 0° naar -90° en bereikt een waarde van -45° in ω=60. Hier leidt ik uit af dat de dubbele pool zich op deze frequentie bevindt en dus ω_n = 60, en dat er hier een nul dicht in de buurt zal liggen. D zal bijgevolg gelijk zijn aan 3600.

Vervolgens bepaal ik grafisch wat de 2% *settling time* is. Hiervoor bekom ik een waarde van 0,175s en deze gebruik ik om de dempingsfactor te berekenen.

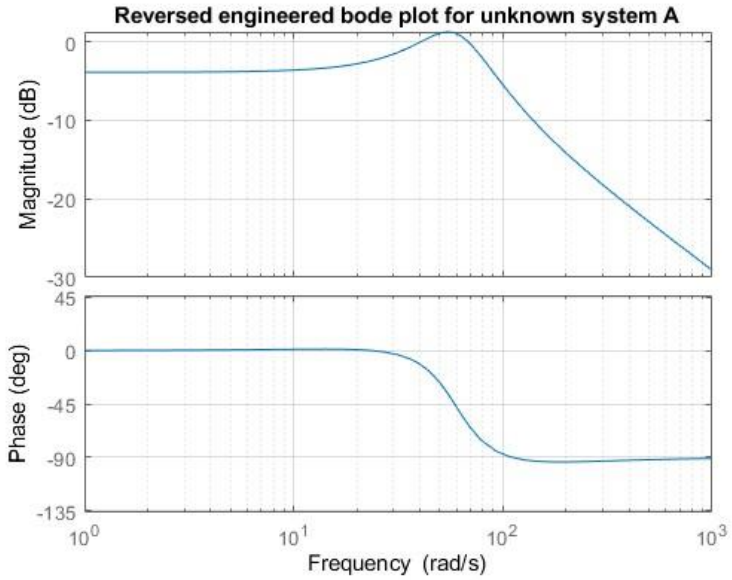
$$T_{s,2\%} \approx \frac{4}{\zeta\omega_n} \Leftrightarrow \zeta \approx \frac{4}{T_{s,2\%}\,\omega_n} = \frac{4}{0{,}175\,.\,60} = 0{,}38.$$

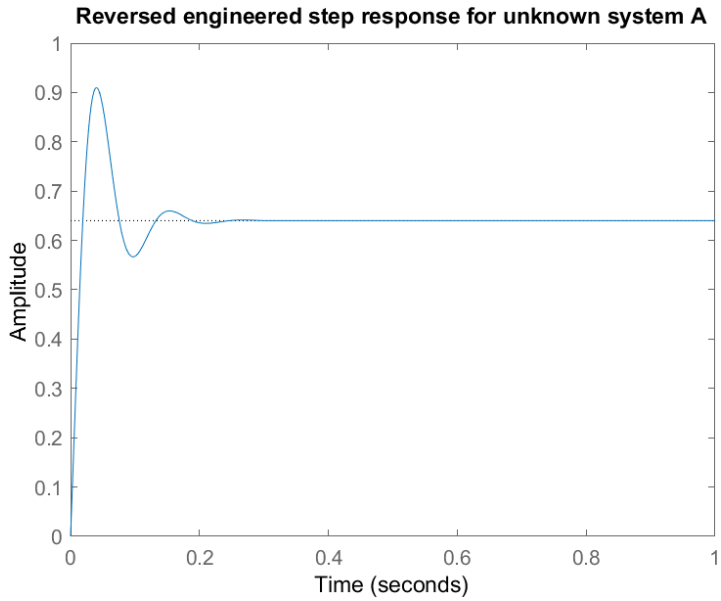
De noemer van een tweede orde transferfunctie heeft de volgende vorm: $s^2 + 2\zeta\omega_n s + {\omega_n}^2$. Hieruit volgt dat de term C uit mijn vereenvoudigde notatie gelijk is aan $C = 2\zeta\omega_n = 2.0{,}38.60 = 46$.

Uit de bode plot haal ik een DC-gain van -3,87dB. Hierdoor bekom ik $\frac{B}{D} = \frac{B}{3600} = 10^{-\frac{3{,}87}{20}} = 0{,}64 \Leftrightarrow B = 3600\,.0{,}64 = 2304$.

De waarde voor A bepaal ik aan de hand van mijn nul. Hier weet ik vrij weinig over, behalve dan dat deze dicht bij de polen zal liggen. Omdat er op de faserespons helemaal geen invloed van de nu zichtbaar is, buiten dan een verschil van 90°, veronderstel ik dat deze achter de polen zal liggen, en dat de oorspronkelijke stijging in de amplitude mede veroorzaakt wordt door resonantie. Ik probeer een waarde van 65 voor de locatie van de pool en bekom dan $A = \frac{2304}{65} = 35$. Dit levert mij volgende transferfunctie op $T(s) = \frac{35s+2304}{s^2+46s+3600}$, met een bode plot en *step response* die overeenkomen met mijn gegenereerde plots. Hierdoor veronderstel ik dat de gekozen positie van mijn nul vrij accuraat is.

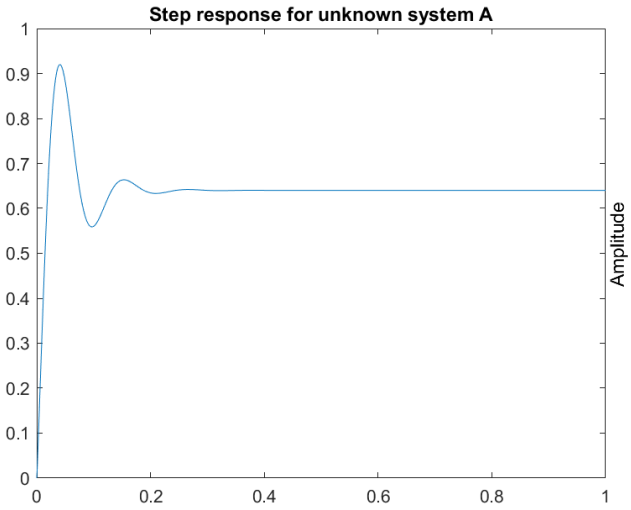
```
%Reverse engineered transfer function:
num_A = [35, 2304];
den_A = [1, 46, 3600];
TF_A = tf(num_A, den_A);

% Bode plot of reverse engineered transfer function:
figure('Name', 'Reversed engineered bode plot for unknown system A');
bode(TF_A)
title('Reversed engineered bode plot for unknown system A')
xlim([1 1000]);
grid on

% Step response of reverse engineered transfer function:
figure('Name', 'Reversed engineered step response for unknown system A');
step(TF_A)
title('Reversed engineered step response for unknown system A')
xlim([0 1])
```

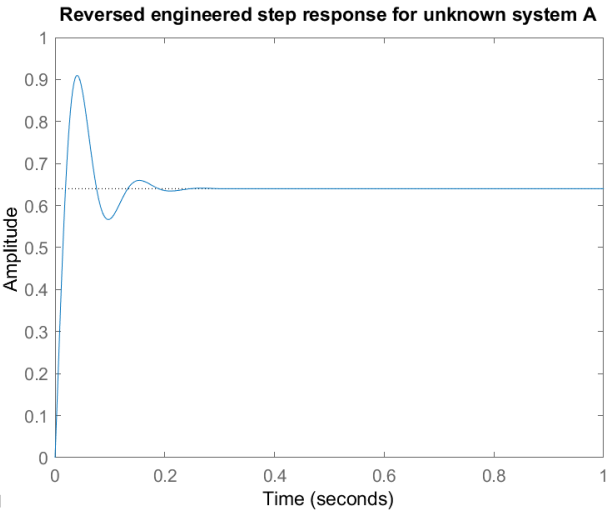


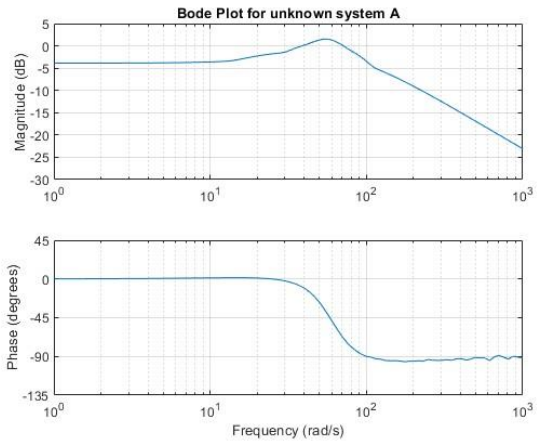


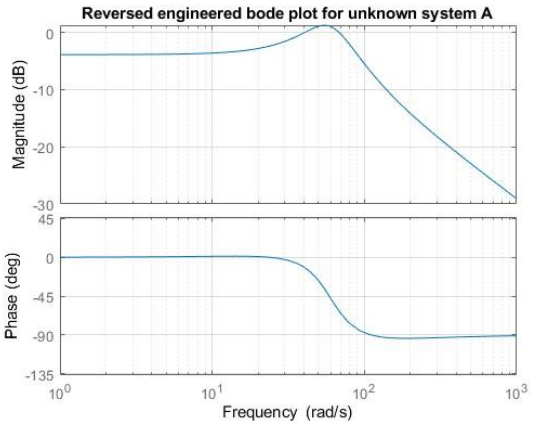
Analyze and discuss the mismatch between your transfer function and the protected transfer function.

De staprespons van mijn transferfunctie en die van de transferfunctie van het beschermd systeem komen redelijk goed overeen. Het enige verschil is dat de pieken van de oscillaties van het beschermd systeem iets hoger liggen dan deze van mijn transferfunctie. De verklaring die ik hiervoor vind, is dat ik de 2% settling time gebruikt hebt om mijn demping ratio te bepalen. De formule die ik hiervoor gebruikt heb, is deze voor een tweede-orde ondergedempt systeem. In mijn transferfunctie is echter ook een nul aanwezig, die zal zorgen voor een versterking van de amplitude van de oscillaties. Hierdoor zal ook de settling time afwijken van de effectieve waarde. Verder heb ik deze settling time grafisch afgeleid van mijn staprespons, wat ook kan leiden tot een lichte afwijking. Deze afwijking kan het verschil in de demping van de oscillaties verklaren. Verder heb ik ook een schatting gemaakt van de positie van de nul. Ook dit zal een invloed hebben op mijn uiteindelijk bodeplot en staprespons. Wanneer ik de bode plots vergelijk, zie ik dat ze een vrij gelijkaardige vorm hebben. De piek van de magnitudeplot van het beschermd systeem ligt iets hoger dan deze van mijn transfer functie. Ook zal de beschermde transfer functie minder snel afnemen in magnitude. Er zal dus in het algemeen een grotere versterking optreden bij die frequenties, wat ik ook verklaar door de lagere dempingsfactor.







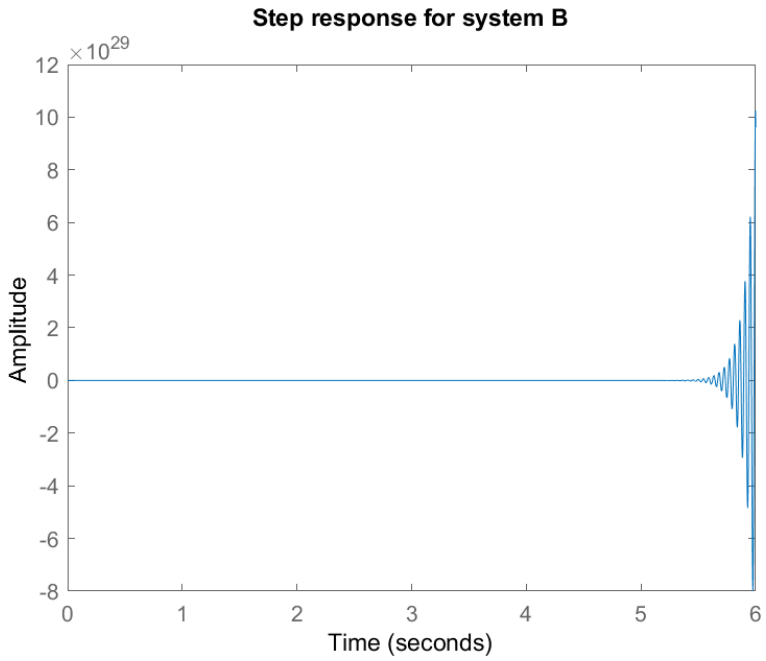


Question 3 (System B)

Discuss the (in)stability of the system. Make a root locus plot and find gain and phase margin

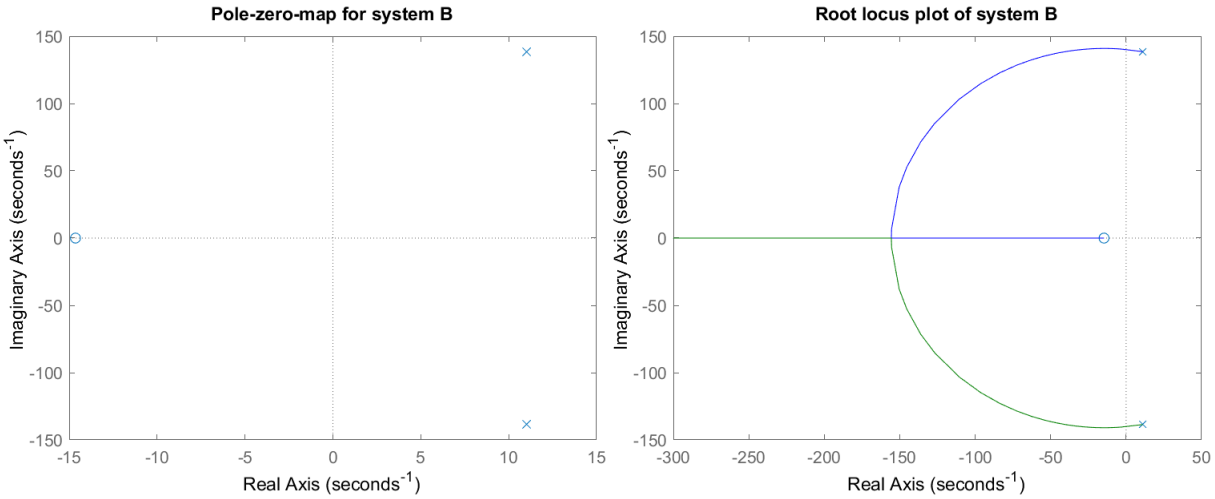
Mijn systeem B heeft de volgende transferfunctie: $T(s) = \frac{3114s+45598}{s^2-22s+19321}$. De negatieve waarde bij de s-factor in de noemer geeft aan dat er sprake is van een negatieve dempingsfactor. Dit is al een eerste indicatie dat het systeem onstabiel zal zijn. Wanneer ik de *step response* van dit systeem teken, ziet men duidelijk dat dit het geval is.

```
num_B = [3114, 45598];
den_B = [1, -22, 19321];
TF_B= tf(num_B, den_B);

% Step response:
figure('Name', 'Step response for system B');
step(TF_B)
title('Step response for system B')
```



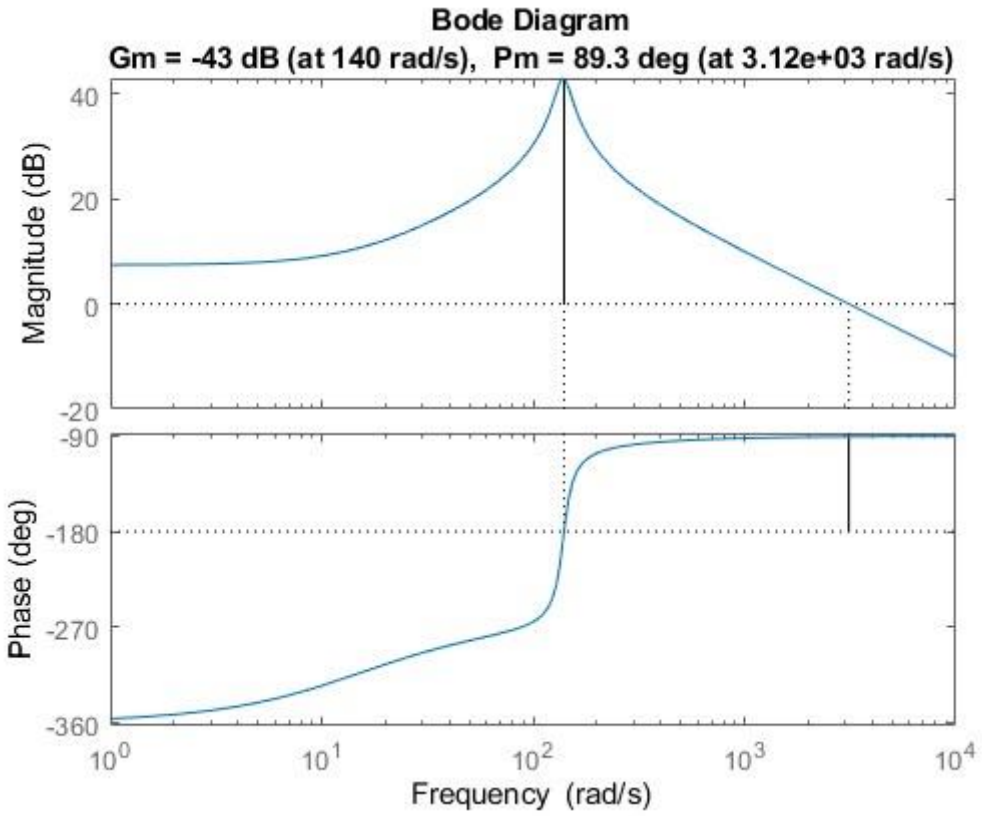
Om de stabiliteit van het systeem verder te onderzoeken, bekijk ik eerst de polen van dit systeem. Dit doe ik aan de hand van een pole-zero-map en de root locus plot. Beiden plots tonen aan dat de polen een positief reëel deel hebben. Dit geeft aan dat het systeem onstabiel is.

```
% Pole-zero-map:
figure('Name', 'Pole-zero-map for system B');
pzmap(TF_B)
title('Pole-zero-map for system B')
% Root locus plot:
figure('Name', 'Root locus plot of system B');
rlocus(TF_B)
title('Root locus plot of system B')
```

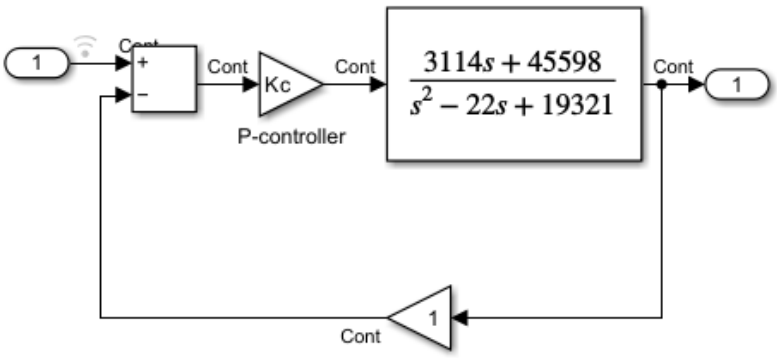


Het systeem heeft een *phase margin* van -43dB. Deze negatieve *phase margin* bevestigd een vierde keer dat het systeem onstabiel zal zijn.

```
%Gain and phase margin:
figure('Name', 'Gain and phase margin of system B');
margin(TF_B)
```



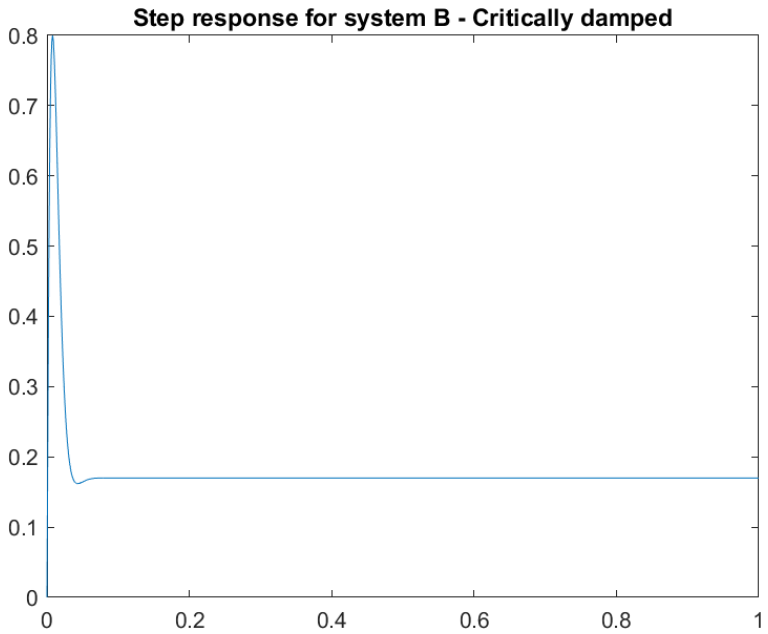
Build a P controller around the unprotected transfer function (B) to stabilize the system.



Tune the P controller so the system is critically damped.

```
Kc = 0.0867;
t = (0:1e-3:1)';
input = heaviside(t);
out = sim("transferfunction9B.slx",t);
figure('Name', 'Step response for system B - Critically damped');
plot(t,out.y)
title('Step response for system B - Critically damped')
```

De transferfunctie van de gesloten lus is gelijk aan $T_{\mathrm{cl},r\to z}(s) = \frac{3114\,K_C\,s+45598K_C}{s^2+(3114K_C-22)\,s+(45598K_C+19321)}$. Wanneer we met een kritische demping te maken hebben weten we dat de dempingsfactor ζgelijk is aan 1. Uit de formule voor een tweede orde transfer functie volgt dat $2\omega_n = 3114K_C - 22$ en ${\omega^2}_n = 45598K_C + 19321$. Wanneer we beide functies omvormen naar ${\omega^2}_n$ en gelijkstellen aan elkaar, kunnen we de K_C-waarde berekenen die tot kritische demping leidt. Deze zal gelijk zijn aan 0,0867 en levert onderstaande staprespons op. Dezelfde waarde kan bekomen worden door de discriminant van de veelterm in de noemer gelijk te stellen aan 0.

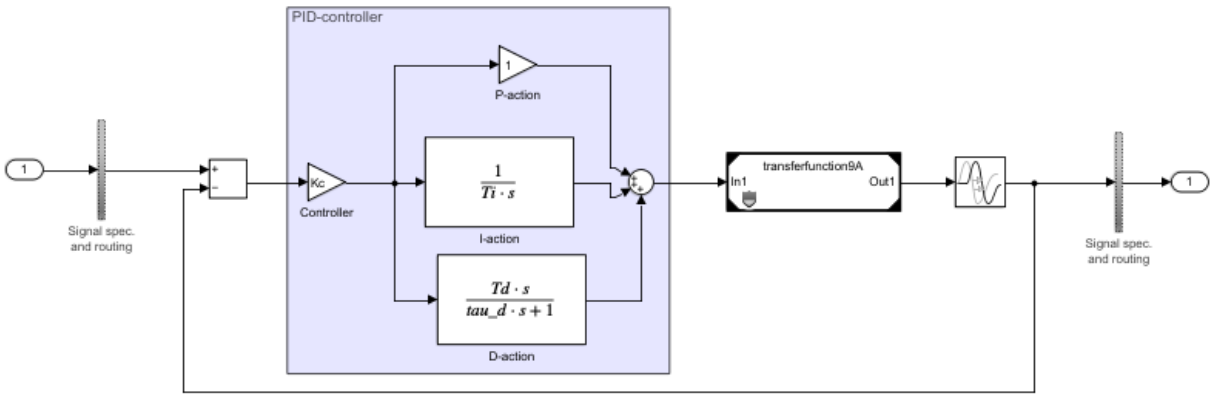


Deze staprespons gaat snel naar de evenwichtspositie, maar er is wel sprake van een grote overshoot. Om deze overshoot en de steady-state error op te lossen is een PID-controller nodig.

Question 4 (System A)

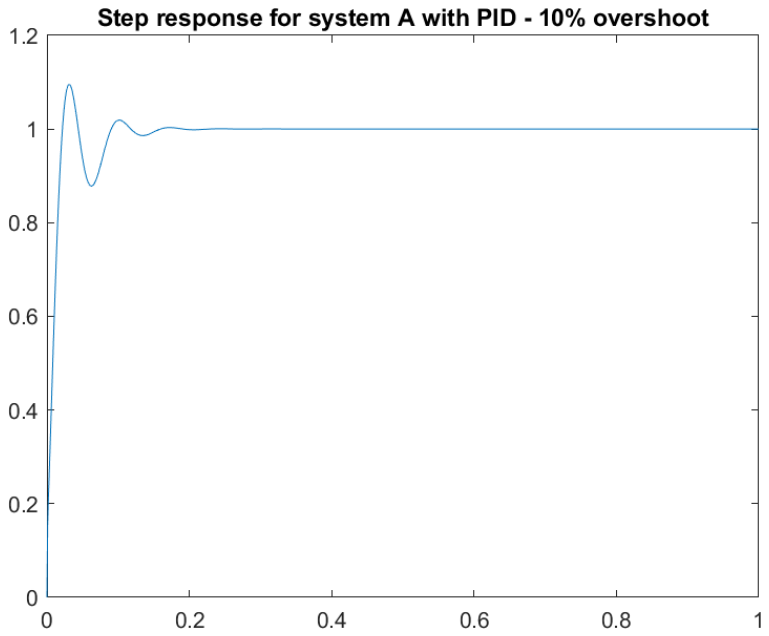
Build a control loop around the protected transfer function.

Hiervoor maak ik een nieuwe Simulink model aan, zodat de staprespons en bode plot die uit de vorige opgaven niet beïnvloed kunnen worden door de parameters uit de *control loop*. Ik voeg wel al de *time delay* uit vraag 5 toe, maar stel deze gelijk aan 0.



Tune the PI(D) controller to have less than 10% overshoot. The D-action may or may not be necessary.

```
time_in = (0:1e-3:1)';
input = heaviside(time_in);
Kc = 1.5;
Ti = 0.01;
Td = 0.003;
tau_d = Td/(10*Kc);
time_delay = 1e-12;
simOut = sim('transferfunction9A_harness_with_control_loop.slx',time_in);
figure('Name', 'Step response for system A with PID - 10% overshoot');
plot(time_in,simOut.response)
title('Step response for system A with PID - 10% overshoot')
```



Discuss your tuning method, parameters and results. Discuss whether you implemented a D-action, and why (not)?

In het hoorcollege hebben we de tuning methode van Ziegler-Nichols gezien. Deze maakt gebruik van de K-waarde bij randstabiliteit. Omdat het systeem gegarandeerd stabiel is en deze methode bovendien vaak tot grote overshoot leidt, heb ik ervoor gekozen om deze methode niet toe te passen. Ik heb de parameters uit mijn PID-controler berekend via trial-and-error. Ik heb eerst de waarde voor T_I als 10^9 ingesteld en die van T_D als 10^{-9}, zodat de I-actie en D-actie geen invloed hebben. Vervolgens heb ik de waarde van K_C in kleine stappen doen toenemen, zoals we in de oefenzitting 11 ook gedaan hebben, maar ik merkte direct een grote overshoot op. Hierdoor heb ik de I-actie ingevoerd en ben ik de waarde van T_I systematisch gaan verkleinen, tot ik op de combinatie K_C = 1.5 en T_I = 0.01 geen steady-state meer had en maar een lichte overshoot. Om de oscillaties en bijgevolg de overshoot te verminderen, heb ik ook een D-actie ingevoerd. Opnieuw via trial-and-error kwam ik voor een waarde voor T_D van 0.003 bovenstaande staprespons uit, die een overshoot van minder dan 10% heeft.

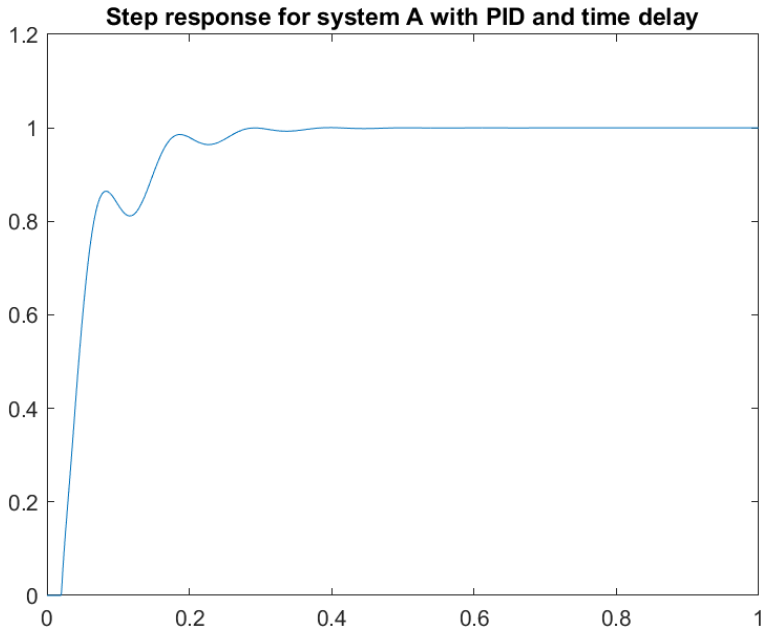
Question 5 (System A)

In Simulink, add a time delay in series with system A. If the delay becomes too large, this destabilizes the system (discuss!).

Een tijdsvertraging t_d in het tijdsdomein, zorgt in het Laplacedomein voor een extra factor $e^{-t_d s}$. De tijdsvertraging zal er dus voor zorgen dat de fase exponentieel en oneindig afneemt en dit effect zal toenemen bij een stijgende frequentie. De maximale tijdsvertraging die kan worden toegevoegd is een tijdsvertraging die berekend wordt uit de phase margin met zijn bijhorende frequentie ($t_d = \frac{PM.\pi}{180°.\omega_{PM}}$). Wanneer deze tijdsvertraging aanwezig is, is het systeem randstabiel. Een grotere tijdsvertraging zal tot instabiliteit leiden.

Tune the PI(D) controller again so the output is desirable even with this delay.

```
time_in = (0:1e-3:1)';
input = heaviside(time_in);
Kc = 0.25;
Ti = 0.01;
Td = 0.01;
tau_d = Td/(10*Kc);
time_delay = 0.02;
simOut = sim('transferfunction9A_harness_with_control_loop.slx',time_in);
figure('Name', 'Step response for system A with PID and time delay');
plot(time_in,simOut.response)
title('Step response for system A with PID and time delay')
```



Calculate or find the maximum delay for which the system can be made stable.

Het maximale tijdsverschil waarvoor het systeem stabiel kan zijn, is het tijdsverschil waarvoor het faseverschil gelijk is aan de *phase margin*. In dit geval is dit een tijdsvertraging van 0,0299s.

```
[Gm,Pm,Wcg,Wcp] = margin(tf([35, 2304], [1, 46, 3600]));
max_time_delay = Pm*pi/(180*Wcp)
```