UNIVERSITY OF WATERLOO

FACULTY OF ENGINEERING

ECE 488 - Project 1 & 2

Prepared by:

Jianxiang (Jack) Xu [20658861]

8 February 2021

1 Problem P1: Classical design for the SISO aiming system

Answer 1.1: (a) Verify the linearization

For the given nonlinear motion equations:

$$\frac{3}{2}\ddot{w}+\frac{1}{4}\left[\ddot{\theta}\cos(\theta)-(\dot{\theta})^2\sin(\theta)\right]=0 \tag{1}$$

$$\frac{1}{6}\ddot{\theta}+\frac{1}{4}\left[\ddot{w}\cos(\theta)-g\sin(\theta)\right]=\tau \tag{2}$$

We may linearize Equation (1) and Equation (2) about $\theta=0$ with Taylor series for small angle changes:

$$\frac{3}{2}\ddot{w}+\frac{1}{4}\left[\ddot{\theta}\cancelto{1}{\cos(\theta)}-\cancelto{0}{(\dot{\theta})^2}\cancelto{\theta}{\sin(\theta)}\right]=0 \tag{3}$$

$$\frac{1}{6}\ddot{\theta}+\frac{1}{4}\left[\ddot{w}\cancelto{1}{\cos(\theta)}-g\cancelto{\theta}{\sin(\theta)}\right]=\tau \tag{4}$$

Hence, we may obtain the linearized set of equations as following:

$$\frac{3}{2}\ddot{w}+\frac{1}{4}\ddot{\theta}=0 \tag{5}$$

$$\frac{1}{6}\ddot{\theta}+\frac{1}{4}\ddot{w}-\frac{1}{4}g\theta=\tau \tag{6}$$

From Equation (5), we may obtain:

$$\ddot{w}=-\frac{1}{6}\ddot{\theta} \tag{7}$$

Substitute Equation (7) into Equation (6), we may obtain:

$$\frac{1}{6}\ddot{\theta}-\frac{1}{24}\ddot{\theta}-\frac{1}{4}g\theta=\tau \quad\Rightarrow\quad \frac{1}{8}\ddot{\theta}-\frac{1}{4}g\theta=\tau \tag{8}$$

Applying Laplace transform, we may obtain the transfer function:

$$\because \quad \mathscr{L}\left\{\frac{1}{8}\ddot{\theta}-\frac{1}{4}g\theta=\tau\right\} \tag{9}$$

$$\therefore \quad \frac{1}{8}s^2\Theta(s)-\frac{g}{4}\Theta(s)=\mathbb{T}(s) \tag{10}$$

$$\therefore \quad P_1(s)=\frac{\Theta(s)}{\mathbb{T}(s)}=\frac{1}{\frac{1}{8}s^2-\frac{g}{4}}=\frac{8}{s^2-2g}=\frac{8}{(s-\sqrt{2g})(s+\sqrt{2g})} \tag{11}$$

Now, by sub in numerical value $g=9.8$, and we may obtain a numerical transfer function of the plant $P_1(s)$:

Final Linearized Plant $P_1(s)$

$$P_1(s)=\frac{\Theta(s)}{\mathbb{T}(s)}\approx\frac{8}{(s-4.427)(s+4.427)} \tag{12}$$

Q.E.D.

Answer 1.2: (b) Compute $W(s)/\tau(s)$

Recall Answer 1.1, we obtained the relationship between w and $theta$ in time domain as stated in Equation (7). We may apply Laplace transform, and obtain their s-domain relationship:

$$\mathbb{W}(s) = -\frac{1}{6}\mathbb{T}(s) \tag{13}$$

Hence, we may obtain the transfer function $P_2(s)$:

Final Linearized Plant $P_2(s)$

$$P_2(s) = \frac{\mathbb{W}(s)}{\mathbb{T}(s)} = -\frac{1}{6}\frac{\Theta(s)}{\mathbb{T}(s)} \approx -\frac{4/3}{(s-4.427)(s+4.427)} \tag{14}$$

Answer 1.3: (c) Classical Control Design

The goal is to design a controller $C_1(s)$ for the plant $P_1(s)$ such that all three specifications are met:

I The closed-loop sytem is stable
II For step reference signals, the steady-state tracking error $|e_\infty| \leq 10\%$
III The phase margin is at least 50°

As we may observed from the plant function Equation (12), there exists a pole in ORHP, hence the closed loop plant is unstable, as observed in the root locus plot and step response in `'sisotool'` (as shown Figure 1-1 below).

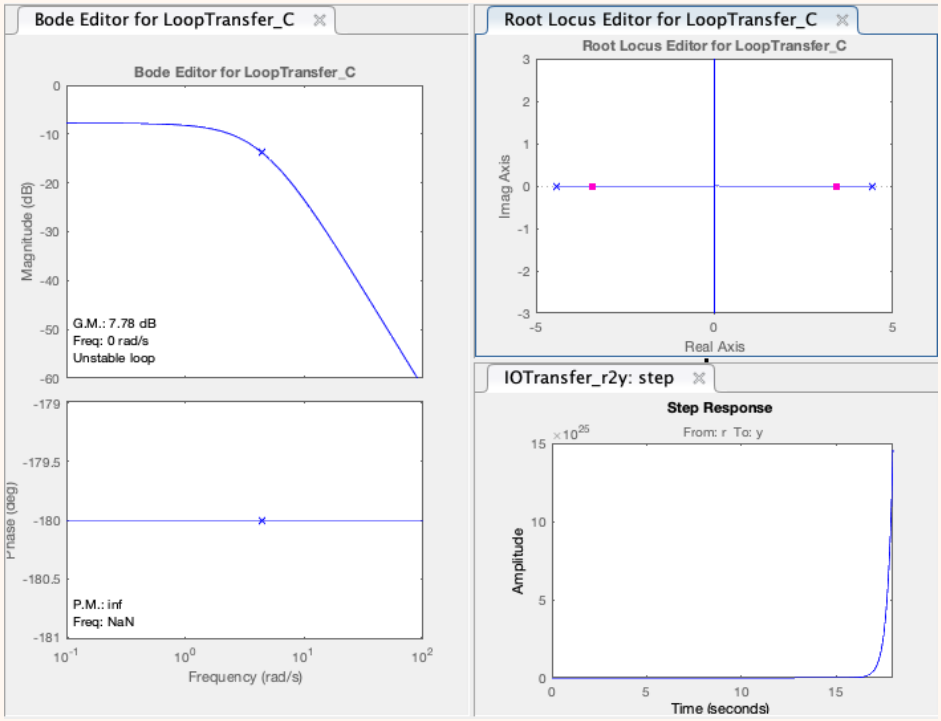

Figure 1-1. Control panel view of the SISOtool

Hence, we need a real zero in LHP for the loop gain $L_1(s) = C(s)P_1(s)$. To make the controller proper, we may need an extra pole in $C(s)$. Hence, we may possibly stabilize the P_1 plant (Spec I) with a first order controller.

From the bode plot in Figure 1-1, we may find the DC gain is below $0\,[\text{dB}]$ and the phase plot is a line at -180°. Hence, to satisfy Spec II and Spec III, we need a lead filter with zero dominant, so that we can pull

up the phase margin around the cross-over frequency w_{cg} to obtain proper phase margin.

We shall find the proper gain and zeros first to satisfy the steady state error and phase margin (Spec II and Spec III), and then determine the proper pole after the cross over frequency ω_{gc} to maintain the margin within the Spec III.

Step 1: determine the zero location

As Figure 1-2 shown below, there are three cases of zero locations so that the closed loop would be stable: zero on the right of the plant LHP pole (Option A), zero on the left of the plant poles (option B), and zero at the LHP plant pole (option C). To note, the zero has to locate in LHP to ensure closed loop stable.

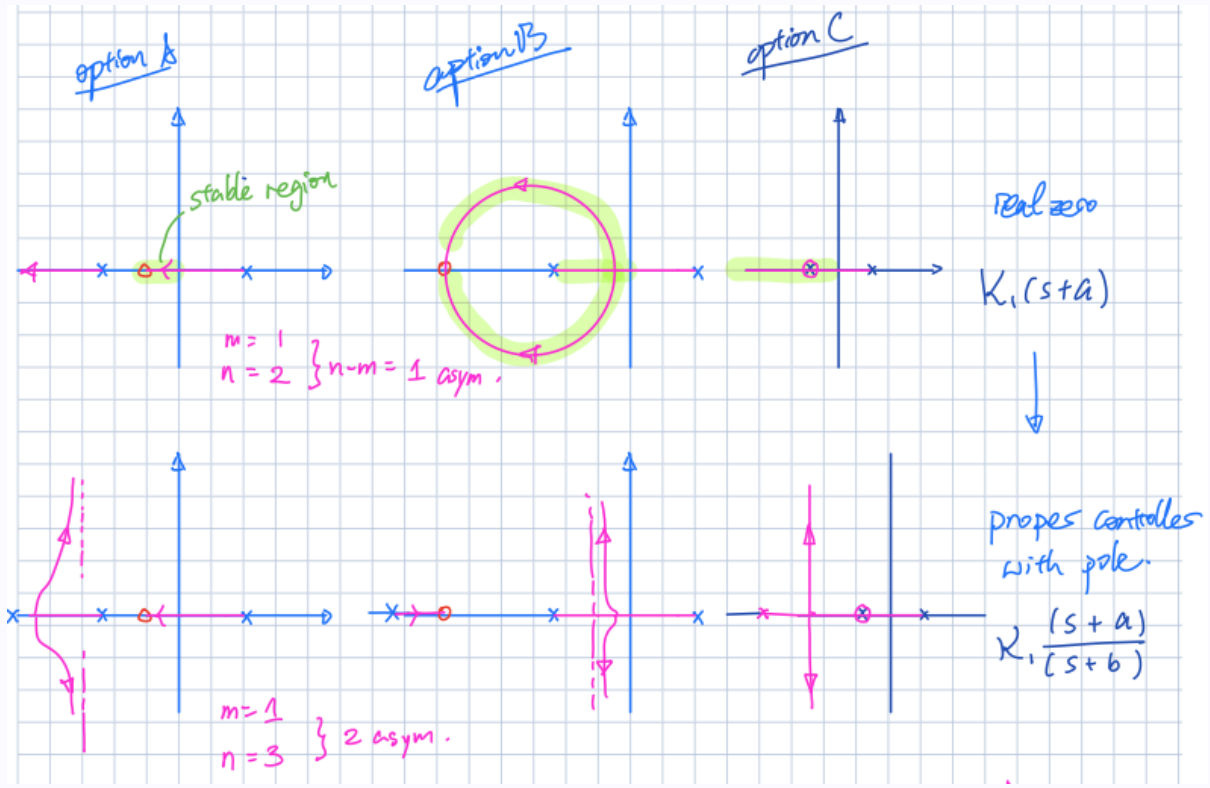

Figure 1-2. Zero placement in root locus

It is possible to design a proper controller with all three options to stabilize the plant (as required by Spec I), but there would be pros and cons in each option.

For simplicity, we proceed with option C, which results a pole cancellation, resulting a reduced order in the closed-loop. Hence, let $C_1' = K_1(s+a) = K_1(s+4.427)$ with pole at $s = -4.427$.

Step 2: determine the gain

To satisfy the steady state error (Spec II), we may determine the minimum gain required:

$$\left|\frac{1}{1+P_1(s=0)C_1(s=0)}\right| \leq e_{ss}^{max} = 10\% \tag{15}$$

$$\frac{1}{1+\left|\frac{8}{(s-4.427)(s+4.427)}\right| K_1(s+4.427)} \leq 0.1 \tag{16}$$

$$\frac{1}{1+\frac{8K_1}{4.427}} \leq 0.1 \tag{17}$$

$$9 \leq \frac{8}{4.427}K_1 \tag{18}$$

$$4.98 \leq K_1 \tag{19}$$

Hence, we need a minimum of 4.98 as the gain. For simplicity, let's assume $K_1 = 10$, so that we may compensate the later gain loss from the pole.

Now the controller is formulated as:

$$C_1(s)' = K_1(s+a) = 10(s+4.427) \tag{20}$$

Let's checkout the effect of the current improper controller $C_1'(s)$ designed above.

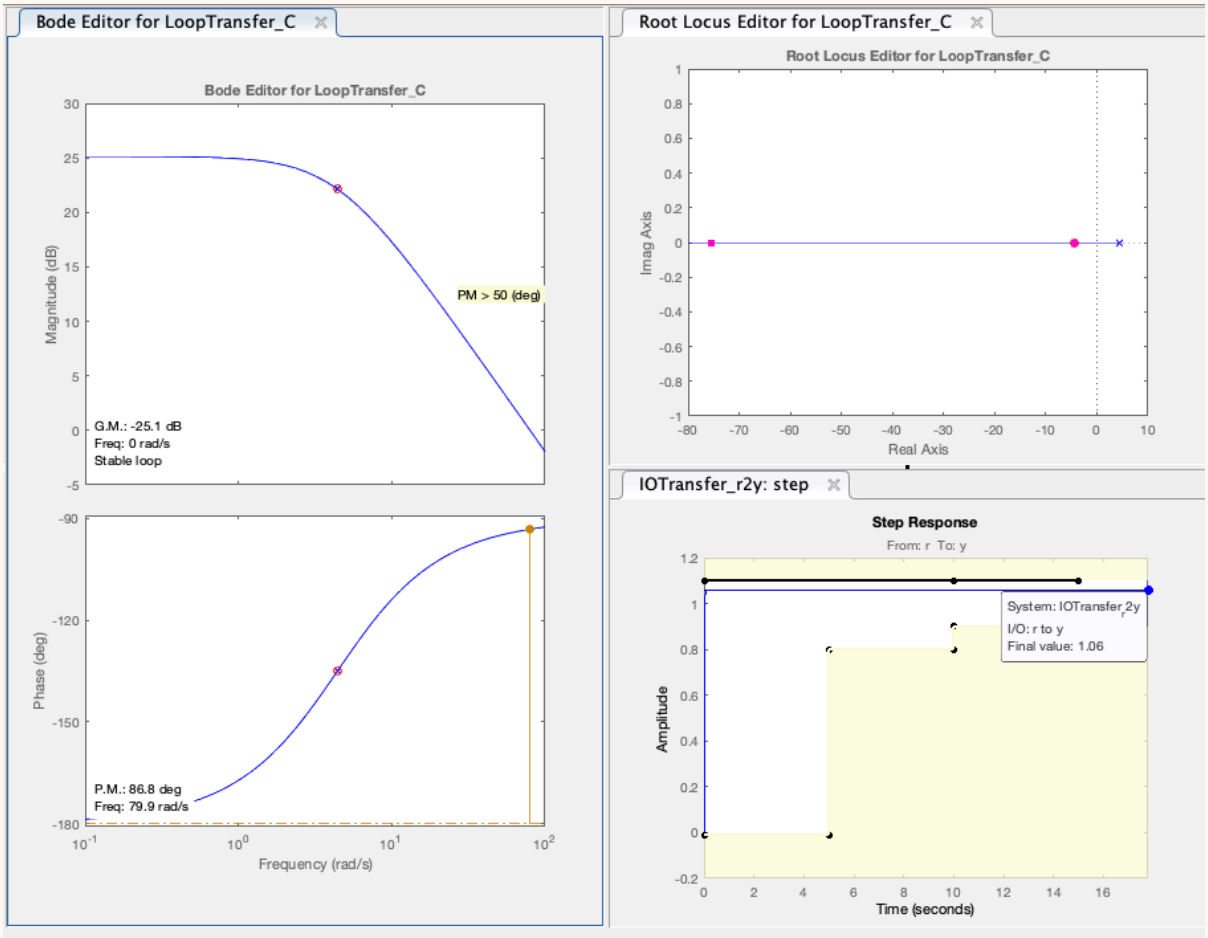

Figure 1-3. SISO Control Panel of the current design $C_1'(s)$

From Figure 1-3, we may observe we have satisfied the steady state spec Spec II, and we have also satisfied the phase margin requirement Spec III. Hence, additional lead compensator is not required.

Step 3: determine the pole

Now, we may determine the pole to make it proper, with $\frac{1}{\tau s+1}$.

Ideally, $\tau < \frac{1}{10\,\omega_{gc}} = \frac{1}{80[\text{rad/s}]\times 10} = 0.00125$

Hence, let's use $\tau = 0.001[s]$.

As a result, the final controller is:

Final Controller $C_1(s)$

$$C_1(s) = \frac{10(s+4.427)}{0.001s+1} \tag{21}$$

As observed in SISO tool (as shown in Figure 1-4 below), the controller is stable, the phase margin is $PM = 82.3°$ and the steady state error is $|e_{ss}| = 6\%$. Hence, all specs (Spec I, Spec II, Spec III) are met with Equation (21).

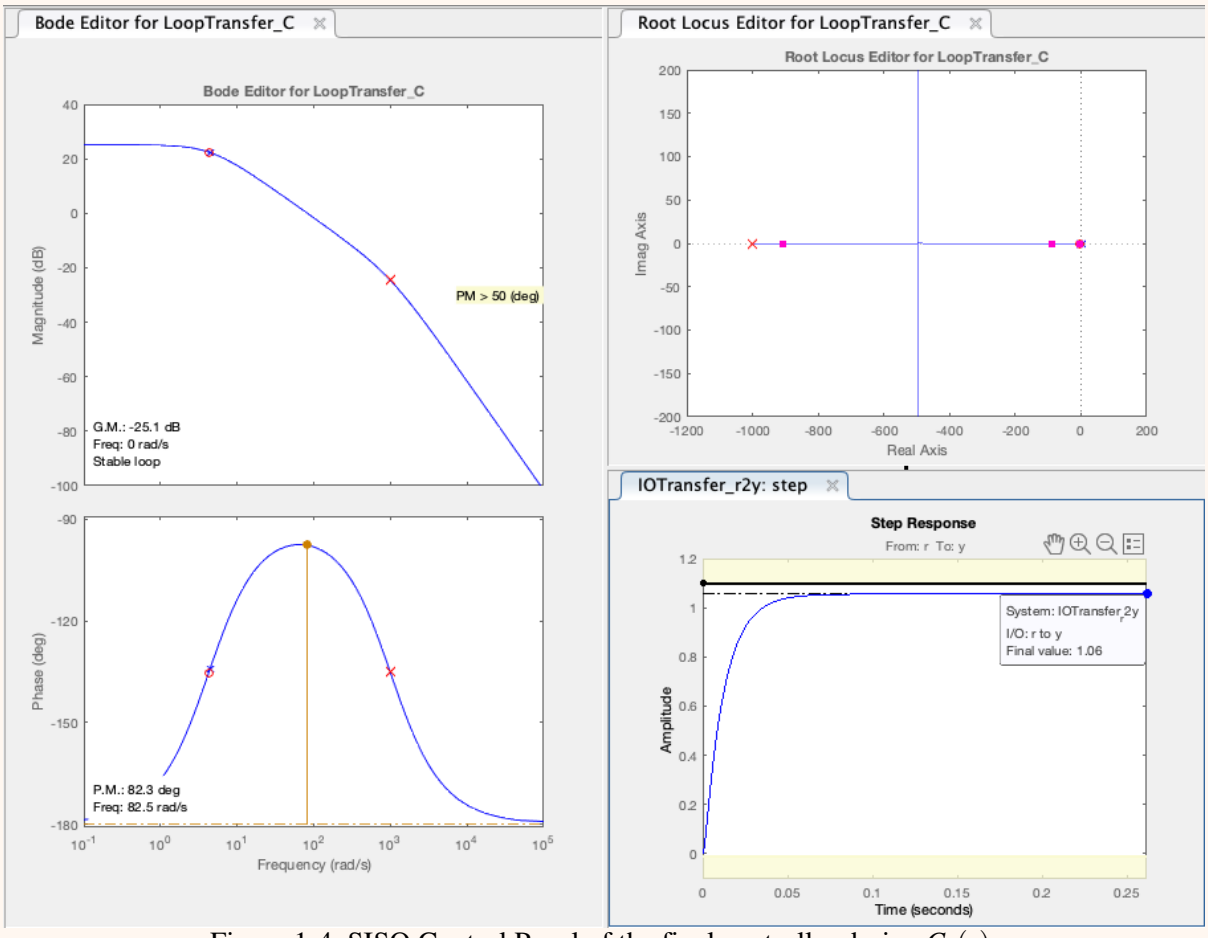

Figure 1-4. SISO Control Panel of the final controller design $C_1(s)$

Answer 1.4: (d) MATLAB : Performance Evaluation

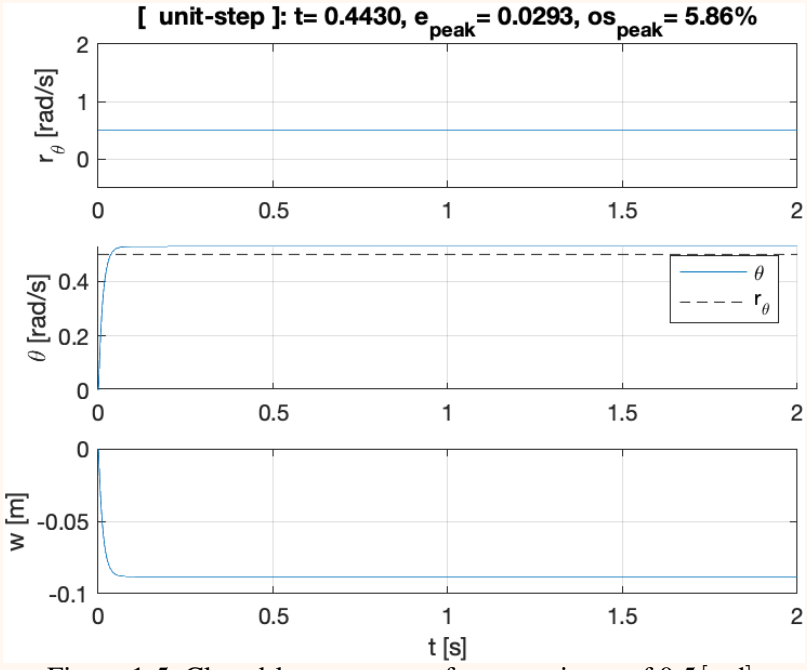

Figure 1-5. Closed-loop response for a step input of 0.5 [rad]

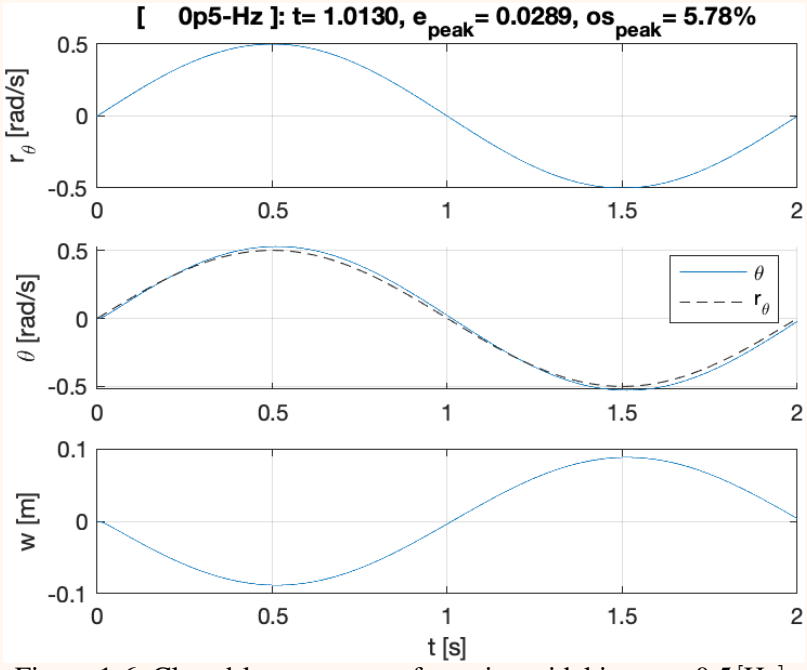

Figure 1-6. Closed-loop response for a sinusoidal input at 0.5 [Hz]

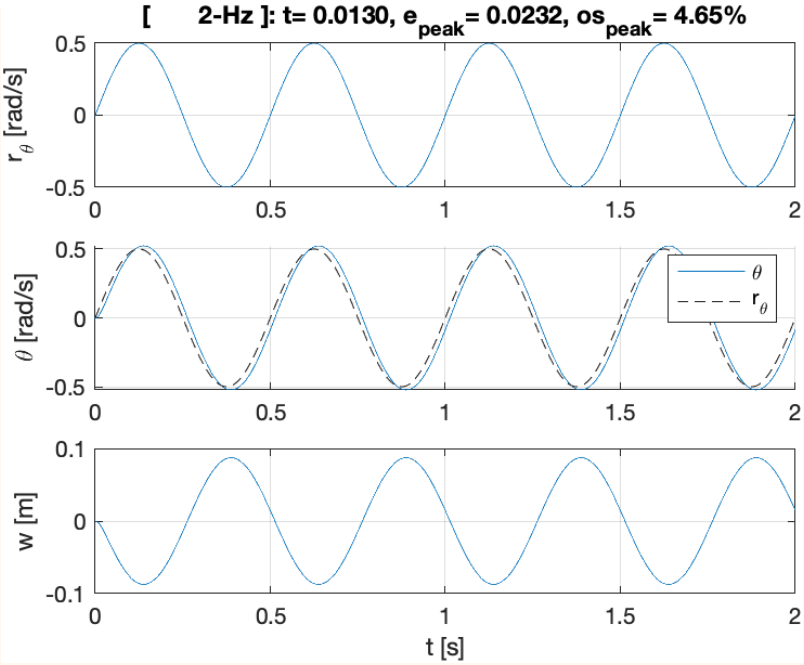

Figure 1-7. Closed-loop response for a sinusoidal input at 2 [Hz]

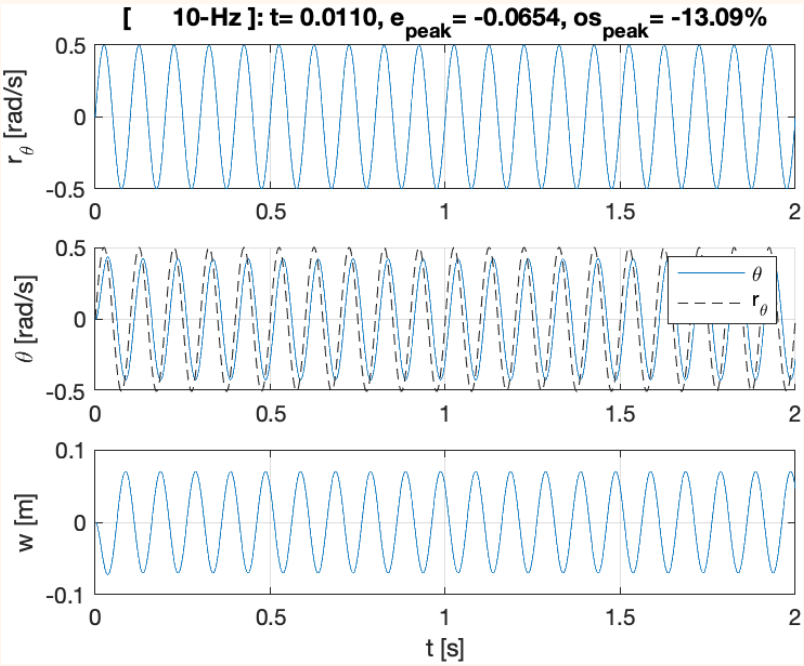

Figure 1-8. Closed-loop response for a sinusoidal input at 10 [Hz]

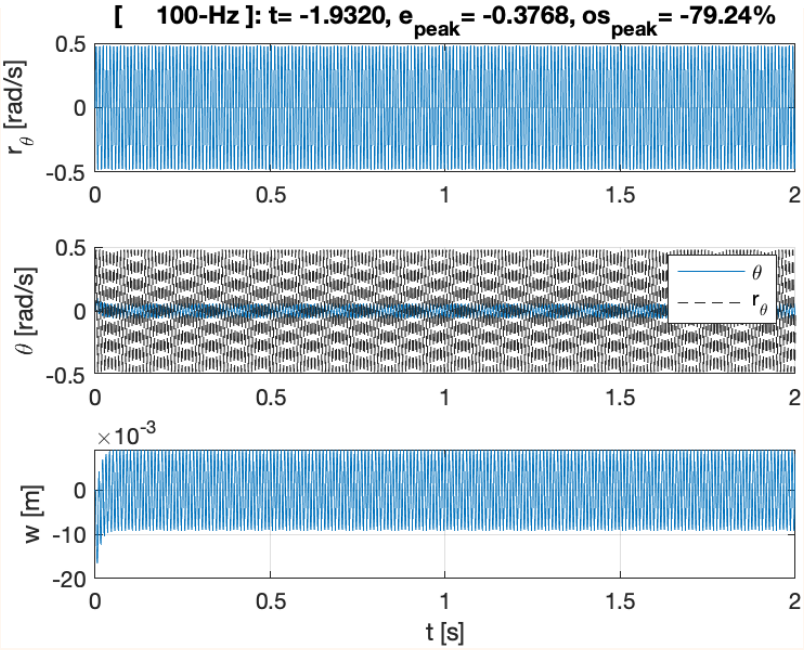

Figure 1-9. Closed-loop response for a sinusoidal input at 100 [Hz]

Remark 1.1: Steady-state tracking error verification

As suggested by Spec II, we shall have a steady state error less than 10%, it is verified with a unit step input as shown in Figure 1-5 which has shown the steady state error is less than 5.86%. To note, the os_{peak} here indicates the percent overshoot from the input peak. Since the system is under-damp as designed, the peak here would be the steady-state value, and the overshoot percentage indicates the steady state error. Hence, we may conclude, the Spec II has been satisfied, and the controller is indeed stable.

Remark 1.2: Comment on the tracking performance for oscillatory signals

As observed from Figure 1-7, it performs the best with smaller steady state error and faster response. In addition, as observed from Figure 1-6, we also see slight improvements in both steady state error and rise time delay. However, at higher frequency (Figure 1-8 and Figure 1-9), we see the steady state error and rise time delay has been dramatically worsen as the frequency increase.

This is as what we expected. As the frequency increases, the gain should be attenuated. At the low frequency range, the gain is reduced slightly, whereas the gain is reduced much dramatically at higher frequency after (about 6Hz). We shall expect the peak value reduced and the steady state error enlarged as the frequency increased. As observed, the peak value is reduced throughout all frequency as frequency increased. Since the steady state error was originated from the overshoot of the target value at DC step input, this overshoot would be reduced as the frequency increases at low frequency band, resulting an increased performance in steady state error at low frequency. As observed, the steady state error is reduced slightly in low frequency band and increased dramatically in high frequency band.

The rise time should improve at low frequency before the crossover frequency $\omega_{gc} = 82.5\,[\text{rad/s}] = 13.13[Hz]$, since the phase increases at low frequency band as the frequency increases. The rise time shall be worsen as the frequency continues to increase in high frequency.

Optional: Simulation

Thanks to Professor's provided simulation, we may now see how it works!

(Please enable by `'EN_SIM'`)

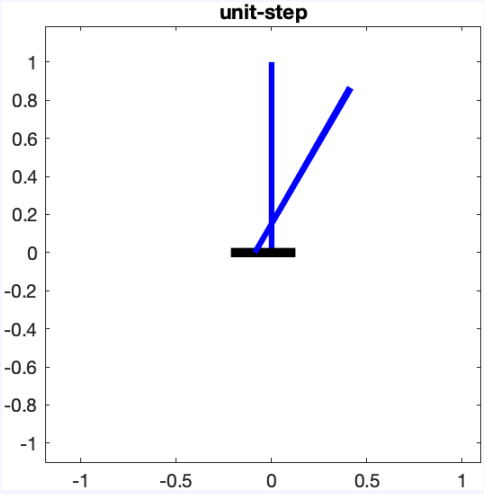

Figure 1-10. Simulation result of unit-step input

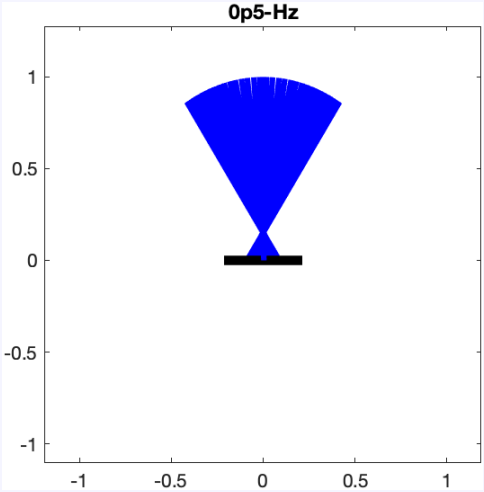

Figure 1-11. Simulation result of sinusoidal input at 0.5 [hz]

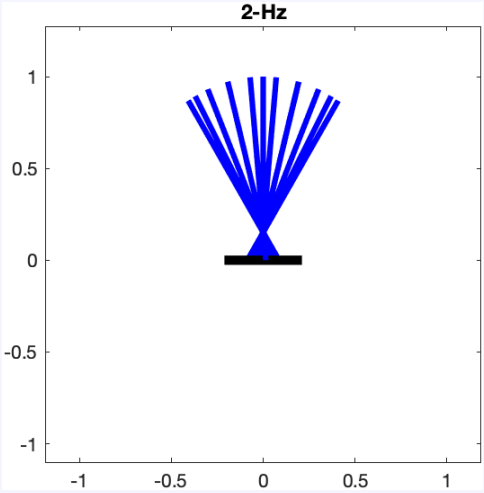

Figure 1-12. Simulation result of sinusoidal input at 2 [hz]

2 Problem P2: Further analysis of the SISO aiming system controller

Answer 2.1: (a) Loop Gain $L_1(s) = P_1(s)C_1(s)$ Bode Plot

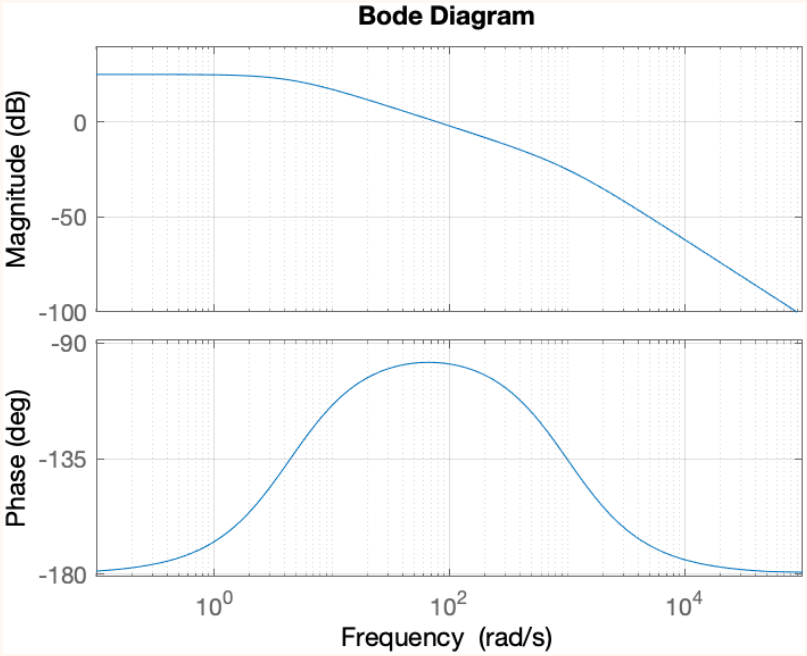

Figure 2-1. Bode plot of the loop gain ($L_1(s) = P_1(s)C_1(s)$)

Recall a typical good loopshape has $|L(j\omega)|$ large at low frequencies and small at high frequency. As observed in Figure 2-1, the loop gain indeed appears such characteristic, hence, it is a "typical good loopshape".

Answer 2.2: (b) Bandwidth of the control system

From the MATLAB function `'bandwidth'`, we obtained the closed loop bandwidth as $BW = 81.93\,[\text{rad/s}]$. Since the unstable plant pole is located at $p = 4.427\,[\text{rad/s}]$, the rule of thumb is indeed satisfied:

$$BW = 81.93\,[\text{rad/s}] \gg 2p = 2 \times 4.427\,[\text{rad/s}] \tag{22}$$

Answer 2.3: (c) Robustness analysis of $C_1(s)$

To evaluate the performance of the closed-loop systems for original and modified plant, we shall plot their bode plot and root locus for the open loop, and corresponding closed-loop unit-step response.

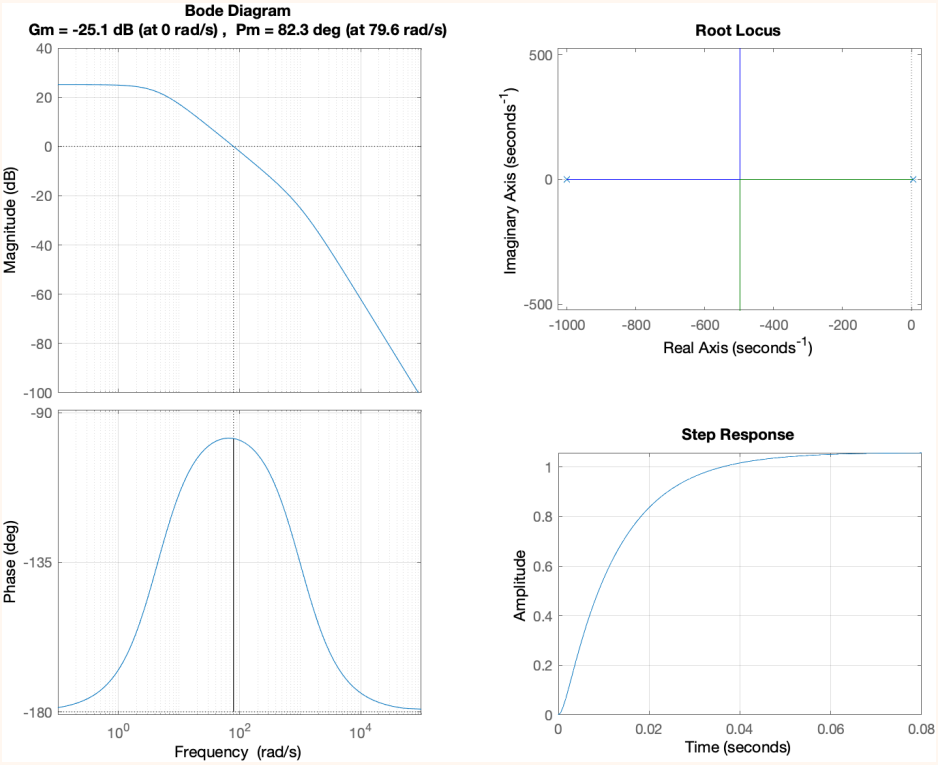

Figure 2-2. Characteristic plots with the original pant (P_1)

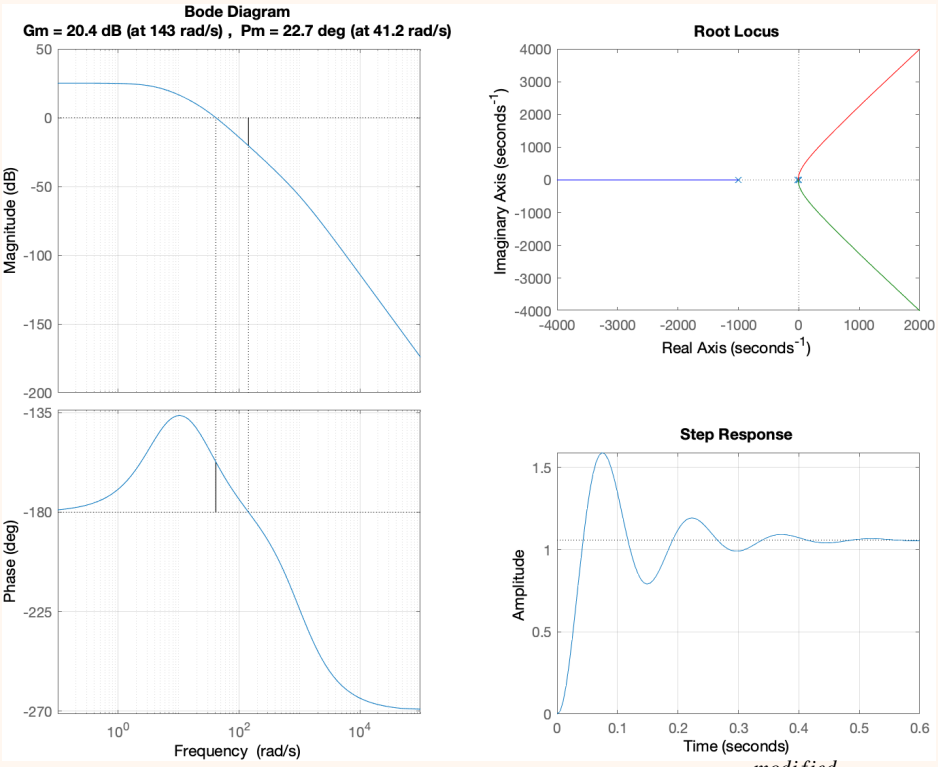

Figure 2-3. Characteristic plots with the modified pant ($P_1^{modified}$)

i) Comments on Modified Plant

As observed in Figure 2-3, the closed-loop system appear to be stable. The step response (also shown in Figure 2-4 below) becomes under-damped with some oscillations and significant overshoot (which was not observed in Figure 2-2). The phase margin is reduced significantly from 82 [deg] to 23 [deg].

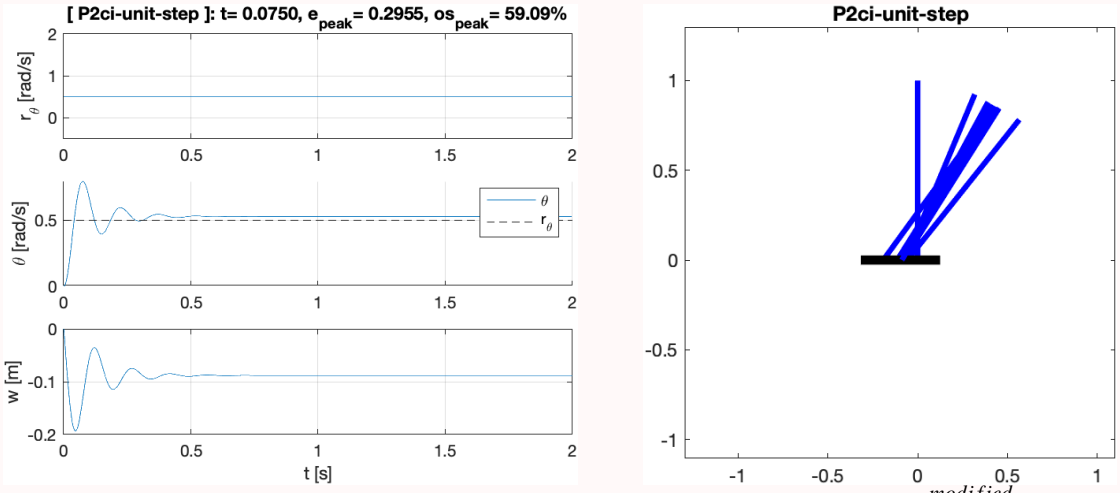

Figure 2-4. Unit step response and simulation with the modified pant ($P_1^{modified}$)

This is as expected with our intuition, since the modified plant introduced another pole at 25 [rad/s], which would pull down the the gain and phase before the original crossover frequency (79.6 [rad/s]), introducing a significant phase lag to the system. In addition, the extra pole makes the asymptote into complex plane of root-locus, resulting oscillation in the step response.

Proof 2.1: ii) The perturbed plant in Answer 2.3 a particular case of the uncertainty model

Let's say there exist some $\{\Delta(s) : |\Delta(j\omega)| < 1, \forall\omega \in \mathbb{R}^+\}$, such that $(1+\Delta(s)W(s)) = \frac{25}{s+25}$.

$$(1+\Delta(s)W(s)) = \frac{25}{s+25} \tag{23}$$

$$\Delta(s) = \frac{\frac{25}{s+25} - 1}{W(s)} \tag{24}$$

$$\Delta(s) = -\frac{1}{10}\frac{s+200}{s+25} \tag{25}$$

Let's see if the condition ($|\Delta(j\omega)| < 1, \forall\omega \in \mathbb{R}^+$) still holds:

$$|\Delta(j\omega)| = \left|-\frac{1}{10}\frac{j\omega+200}{j\omega+25}\right| \tag{26}$$

$$\because \quad |\Delta(j\omega)|_{\omega\to\infty} = \left|-\frac{1}{10}\frac{j\omega+200}{j\omega+25}\right|^{1} = 0.1 \tag{27}$$

$$|\Delta(j\omega)|_{\omega\to 0} = \left|-\frac{1}{10}\frac{200}{25}\right|^{1} = 0.8 \tag{28}$$

$$\therefore \quad |\Delta(j\omega)| < 1\forall\omega \tag{29}$$

Hence, there indeed exists such $\{\Delta(s) : |\Delta(j\omega)| < 1, \forall\omega \in \mathbb{R}^+\}$, such that $(1+\Delta(s)W(s)) = \frac{25}{s+25}$.

More specifically, $P_{1,mode}(s) = \frac{25}{s+25} \cdot P_1(s)$ is a particular case of $P_{1,uncertain}(s) = (1+\Delta(s)W(s)) \cdot P_1(s)$, with $\Delta(s) = -\frac{1}{10}\frac{s+200}{s+25}$ for $W(s) = \frac{10s}{s+200}$

Q.E.D.

iii) Small Gain Theorem

For a multiplicative uncertainty model, the Small Gain Theorem can be simplified to the following: The system is robustly stable, if $|M(j\omega)| = |\frac{L_1(s)W(s)}{1+L_1(s)}|_{s=j\omega} \leq 1 \, \forall \omega$

Hence, we just need to plot the bode-plot of the transfer function $M(s) = \frac{L_1(s)W(s)}{1+L_1(s)}$, and make sure the magnitude is less than $0\,[\text{dB}]$ for all frequency.

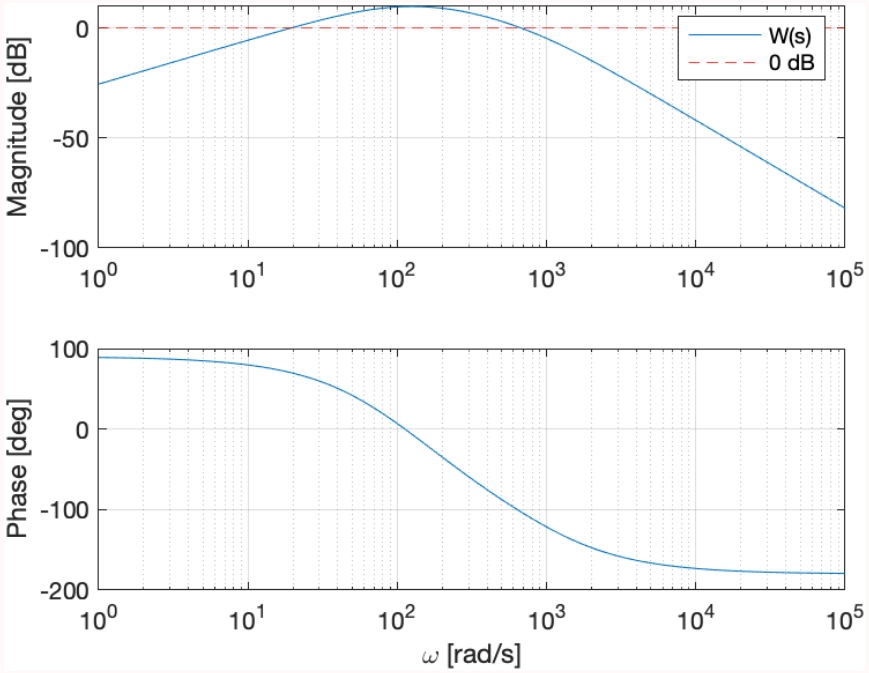

Figure 2-5. Bode plot of $M(s)$

As Figure 2-5 shown, the magnitude of $M(s)$ actually over the $0\,[\text{dB}]$ for some of the frequency. In short, the system is not robustly stable.

iv) Answer 2.3 and Answer 2.3 are not contradictory

As discovered in Answer 2.3, the system is stable for that particular case (as proved in Proof 2.1), but it does not necessarily mean the system is robustly stable for all cases (as Answer 2.3 discovered).

Glossary

LHP Left Hand Plane.

ORHP Open Right Hand Plane.

RHP Right Hand Plane.

Appendix A Code for Main

```
close all;
clear all;
clc;

%% init.
helper.createFolder("output", false);
helper.createFolder("output/q1", false);
helper.createFolder("output/q2", false);

%% User Param:
ENV_P1 = false;
ENV_P2 = true;
ENV_P2c_i = true;


% Conditional Params
if ENV_P1 || ENV_P2c_i
    EN_SIM  = true;
    T       = 0.001;
    T_END   = 2; %[s]
end


%% TF : Common for P1 and P2
s = tf('s');
P1 = 8/(s-4.427)/(s+4.427);
P2 = -8/6/(s-4.427)/(s+4.427);
C1 = 10*(s+4.427)/(0.001*s + 1);

L1 = minreal(P1 * C1);
TF_r2theta = minreal((L1) / (1 + L1));
TF_r2w = minreal(P2 * (C1) / (1 + L1));
zpk(C1)
zpk(TF_r2theta)
zpk(TF_r2w)


% Test Data
if ENV_P1 || ENV_P2c_i
    t = 0:T:T_END;
    r_theta = 0.5 * ones(1, size(t,2)); % step @ 0.5 [rad/s]
    r_theta_sin_lf = 0.5 * sin((t ./ 2) * 2 * pi); % sin wave @ 0.5 [Hz]
    r_theta_sin_hf = 0.5 * sin((t .* 2) * 2 * pi); % sin wave @ 2 [Hz]
    r_theta_sin_hf2 = 0.5 * sin((t .* 10) * 2 * pi); % sin wave @ 10 [Hz]
    r_theta_sin_hf100 = 0.5 * sin((t .* 100) * 2 * pi); % sin wave @ 100 [Hz]
end

%% Problem 1 Solution %%%%%%%%%%%%%%%%%%%%%%
if ENV_P1
    % siso plot
    helper.sisoPlot(L1, [800,600], "q1", "L1")

    % simulation
    simulation_and_plot(TF_r2theta, TF_r2w, r_theta, t, "unit-step", EN_SIM, "q1")
    simulation_and_plot(TF_r2theta, TF_r2w, r_theta_sin_lf, t, "0p5-Hz", EN_SIM, "q1")
    simulation_and_plot(TF_r2theta, TF_r2w, r_theta_sin_hf, t, "2-Hz", EN_SIM, "q1")
    simulation_and_plot(TF_r2theta, TF_r2w, r_theta_sin_hf2, t, "10-Hz", EN_SIM, "q1")
    simulation_and_plot(TF_r2theta, TF_r2w, r_theta_sin_hf100, t, "100-Hz", EN_SIM, "q1")
end

%% Problem 2 Solution %%%%%%%%%%%%%%%%%%%%%%
```

```
if ENV_P2
    L1 = minreal(P1 * C1);
    zpk(L1)

    %% 2.a) bode
    bode(L1)
    grid on
    helper.saveFigure([400, 300], "q2", "bode_plot_L1")

    %% 2.b) BW
    fprintf("> Bandwidth: %f\n", bandwidth(TF_r2theta))

    %% 2.c)
    if ENV_P2c_i
        % modified plant
        P2ci_P1 = P1 * (25/(s+25));
        P2ci_L1 = minreal(P2ci_P1 * C1);
        P2ci_TF_r2theta = minreal((P2ci_L1) / (1 + P2ci_L1));
        P2ci_TF_r2w = minreal(P2 * (C1) / (1 + P2ci_L1));
        helper.sisoPlot(P2ci_L1, [800,600], "q2", "p2ci-L1-modified")

        % simulate
        simulation_and_plot(P2ci_TF_r2theta, P2ci_TF_r2w, r_theta, t, "P2ci-unit-step", EN_SIM, "q2")
        simulation_and_plot(P2ci_TF_r2theta, P2ci_TF_r2w, r_theta_sin_lf, t, "P2ci-0p5-Hz", EN_SIM, "q2")
        simulation_and_plot(P2ci_TF_r2theta, P2ci_TF_r2w, r_theta_sin_hf100, t, "P2ci-100-Hz", EN_SIM, "q2")
    end

    %% 2.c) - iii
    W = 10*s/(s+200);
    M = minreal((L1 * W)/(1 + L1));

    zpk(M)

    % custom bode plot
    [mag,phase,wout] = bode(M);
    figure()
    subplot(2,1,1)
    semilogx(wout, mag2db(abs(mag(:))))
    grid on
    ylabel("Magnitude [dB]")
    hold on
    yline(0, 'r--')
    legend(["W(s)", "0 dB"])

    subplot(2,1,2)
    semilogx(wout, phase(:))
    grid on
    ylabel("Phase [deg]")
    xlabel("\omega [rad/s]")

    helper.saveFigure([400, 300], "q2", "c-iii")
end

%% Helper
function simulation_and_plot(TF_r2theta, TF_r2w, r_theta, t, tag, ifsim, FOLDER)
    %% sim
    y_theta = lsim(TF_r2theta, r_theta, t);
    y_w = lsim(TF_r2w, r_theta, t);

    % analysis
    rinfo = stepinfo(r_theta,t);
    yinfo = stepinfo(y_theta,t);
    t_delay = (yinfo.PeakTime - rinfo.PeakTime);
    e_peak = (yinfo.Peak - rinfo.Peak);
    os = e_peak/ rinfo.Peak * 100;
    info_str = sprintf("[ %10s ]: t= %.4f, e_{peak}= %.4f, os_{peak}= %.2f%%", ...
        tag, t_delay, e_peak, os)

    % Plot
    figure()
```

```
132
133     subplot(3, 1, 1)
134     plot(t, r_theta)
135     grid on;
136     ylabel("r_{\theta} [rad/s]")
137     title(info_str)
138
139     subplot(3, 1, 2)
140     hold on;
141     plot(t, y_theta)
142     plot(t, r_theta, '--', 'color', '#222222')
143     grid on;
144     ylabel("\theta [rad/s]")
145     legend(["\theta", "r_{\theta}"])
146
147     subplot(3, 1, 3)
148     plot(t, y_w)
149     grid on;
150     ylabel("w [m]")
151     xlabel("t [s]")
152
153     helper.saveFigure([400, 300], FOLDER, sprintf("step_response_%s", tag))
154
155    % Simulate
156     if ifsim
157         figure()
158         single_pend_fancy_sim(t, [y_w, y_theta], [zeros(length(t),1) r_theta'], 1, 50, tag);
159         helper.saveFigure([300, 300], FOLDER, sprintf("sim_%s", tag))
160     end
161 end
```

Code 1: Main Lab Contents

Appendix B Code for Helper Class

```
1  %% Helper Functions %%
2  classdef helper
3      methods(Static)
4          function createFolder(path, clear_folder)
5              if ~exist(path)
6                  mkdir(path)
7                  fprintf("[HELPER] Folder created!\n");
8              else
9                  if ~isempty(path) & clear_folder
10                     rmdir(path, 's');
11                     mkdir(path);
12                     fprintf("[HELPER] Folder is emptied, %s\n", path);
13                 else
14                     fprintf("[HELPER] Folder already existed!\n");
15                 end
16             end
17         end
18         function RH_criterion(coeffs) % [ n , .... , 0 ]
19             num = size(coeffs,2);
20             n = ceil(num / 2);
21
22             if mod(num,2) == 1 % odd number
23                 A = [coeffs, 0];
24             else
25                 A = coeffs;
26             end
27
28             RH_mat = reshape(A, 2, n);
29
30             for j = 1:n
31                 b = sym(zeros(1, n));
32                 for i = 1:n-1
33                     b(i) = RH_mat(j, 1) * RH_mat(j+1, i+1);
34                     b(i) = RH_mat(j+1, 1) * RH_mat(j, i+1) - b(i);
35                     b(i) = b(i)/RH_mat(j+1, 1);
36                     b(i) = simplifyFraction(b(i))
```

```
                end
                RH_mat = [RH_mat; b];
            end
            disp(RH_mat)
        end
        function saveFigure(DIMENSION, FOLDER, FILE_NAME)
            set(gcf,'units','points','position',[0, 0, DIMENSION(1), DIMENSION(2)]);
            exportgraphics(gcf,sprintf('output/%s/%s.png', ...
                FOLDER, FILE_NAME),'BackgroundColor','white');
        end
        function sisoPlot(L_TF, DIMENSION, FOLDER, TAG)
            % Plot
            figure()

            subplot(2, 2, [1,3])
            margin(L_TF)
            grid on

            subplot(2, 2, 2)
            rlocus(L_TF)

            subplot(2, 2, 4)
            G_TF = minreal(L_TF/(L_TF + 1));
            step(G_TF)
            grid on

            helper.saveFigure(DIMENSION, FOLDER, sprintf("siso_plot_%s", TAG))
        end
    end
end
```

Code 2: Helper and commonly used functions by main