MXEN3004 – Dynamic Modelling and Control

Design Assignment

Semester 1, 2022

Brett Stevens – 1940 750

Executive Summary

This document aims to outline the simulation of a controller for an industrial plant that meets certain design criteria. This design criteria can be found in the 2022 MXEN3004 Design Assignment brief. As such, this report is intended to be read alongside the assignment brief. For ease of reading, each section in the body of the report pertains to a section of the assignments brief; for example, section 2.1 of this document corresponds to the first section of the assignment brief: *1 Process transfer function*, etc.

This report was completed with the help of *Matlab*. All graphs and diagrams were generated using *Matlab*. Any additional calculations were completed by hand, the methods of which will be included in the body of the report.

Contents

Table of Figures

1. Introduction

In this report we aim to simulate a controller for an industrial plant that meets certain design criteria given in the design brief. All relevant mathematical reasoning and work done in *Matlab* is included in the body of this report.

2. Design Tasks

2.1. Process transfer function

Using *Matlab* to generate individual parameters:

```
N=19407580

[k,a,b,c,d,wl,wh,ev]=systems_generator(N);
```

Figure 1 - Matlab code used to generate design parameters

Name	Value
a	23.0940
b	40
c	1.7321
d	75
ev	0.0115
k	19
N	19407580
wh	43.3013
wl	34.6410

Figure 2 - Parameters used for plant and control problem

Thus, the plant is governed by the following differential equation:

$$19\left(23.0940\,\frac{du(t)}{dt}+40u(t)\right)=\frac{d^3y(t)}{dt^3}+1.7321\frac{d^2y(t)}{dt^2}+75\frac{dy(t)}{dt}$$

Using Laplace transforms on the above differential equation, it follows that:

$$U(s).(438.786s+760)=\;Y(s).(s^3+1.7321s^2+75s)$$

Therefore,

$$G(s)=\frac{Y(s)}{U(s)}$$
$$=\frac{\mathbf{19(23.0940}\boldsymbol{s}\mathbf{+40)}}{\boldsymbol{s}\mathbf{(}\boldsymbol{s}^{\mathbf{2}}\mathbf{+1.7321}\boldsymbol{s}\mathbf{+75)}}$$

2.2. Proportional controller

Consider the controller:

$$C(s)=K_p$$

As stated as (1) of the design criteria in the assignment brief,

$$\omega_c\in[\omega_l,\omega_h]$$

Where ω_l and ω_h are 34.6410 and 43.3013, respectively (see Figure 1).

We first consider the lower limit of ω_c and find K_{p1} such that:

$$|C(j\omega_c)G(j\omega_c)|=1$$

Using the following *Matlab* script,

```
wc = wl;   % initial gain crossover frequency (lower limit)

% compute the frequency response of the system at the crossover frequency
g_wc = squeeze(freqresp(g_s, wc));
% the magnitude at wc
mag_g_wc = abs(g_wc);

% finding k_p1
k_p1 = 1 / mag_g_wc
```

Figure 3 - Matlab script used to produce K_p1

we derive: $K_{p1} = 2.5643$

Using this value to produce a Bode diagram of the system:

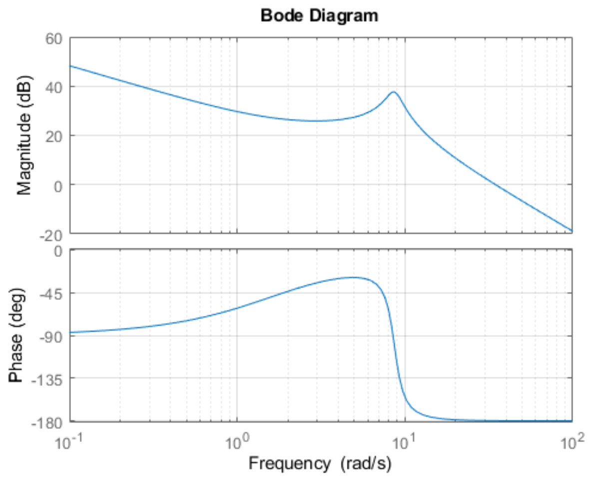

Figure 4 - Bode diagram where wc = 34.6410, Kp = 2.5675

The *phase margin* of this system is derived using the following *Matlab* script:

```
% the phase at wc
phase_g_wc = angle(g_wc);
% phase margin
pm = (phase_g_wc + pi)*180/pi
```

Figure 5 - Matlab script to produce phase margin (pm)

$$Therefore, \qquad PM = 0.1905 < 55°$$

The *phase margin* calculated is less than the required *phase margin* of 55° given as the 2nd design criteria in the design brief.

Now consider a *proportional gain* that satisfies the 3rd design criteria given in the design brief. Given the value of velocity error is known (see Figure 1), we find the appropriate *proportional gain* by performing the following calculations:

$$K_v = \lim_{s \to 0} \left(s\, K_{p2}\, L(s) \right)$$
$$= K_{p2} \frac{19(40)}{75}$$
$$Given\ that, \qquad K_v = \frac{1}{e_v}$$
$$K_{p2} = \frac{1}{e_v} \frac{75}{19(40)} = 8.5463$$

Using this value to produce a Bode diagram of the system:

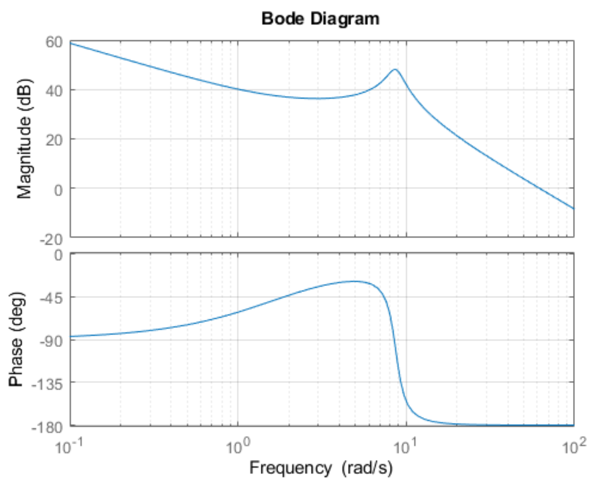

Figure 6 - Bode diagram where wc = 34.6410, Kp = 8.5463

We can also derive the ramp tracking-error of the system using the following *Matlab* script:

```
% Visualise the ramp tracking-error
figure(3); set(gcf,'Visible','on'); % Prepare a figure window and set to a visible state
step(1/s*k_p1*g_s);
grid on
hold on
```

Figure 7 - Malab script used to produce ramp tracking-error

The following graph represents the ramp tracking-error:

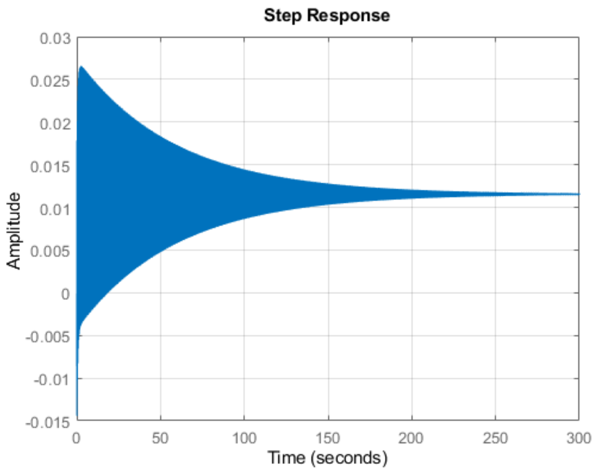

Figure 8 - Ramp tracking-error for proportional controller

2.3. Phase-lead compensator

Considering the phase lead compensator:

$$C(s) = K_p \frac{\tau s + 1}{\alpha \tau s + 1}$$

We select the same K_p as found in section 2.2 (K_{p2}), as to that satisfy the 3rd design criteria given in the design brief. We can obtain variables, τ and α, using the following *Matlab* script, where *phase margin* (pm) = 70°:

```
% Compute the frequency response of the system at the crossover frequency wc
p_wc = squeeze(freqresp(g_s, wc));

% Magnitude at wc
mag_p_wc = abs(p_wc);
% Phase at wc
phase_p_wc = angle(p_wc);

% find phi
phi = pm*pi/180 - pi - phase_p_wc
% Compute tz
tz = (mag_p_wc-cos(phi))/(wc*mag_p_wc*sin(phi));
%find alpha
alpha = (mag_p_wc*cos(phi)-1)/mag_p_wc*(mag_p_wc-cos(phi));
```

Figure 9 - Matlab script to find variables for phase-lead compensator

The bode diagram of the system with the phase-lead compensator:

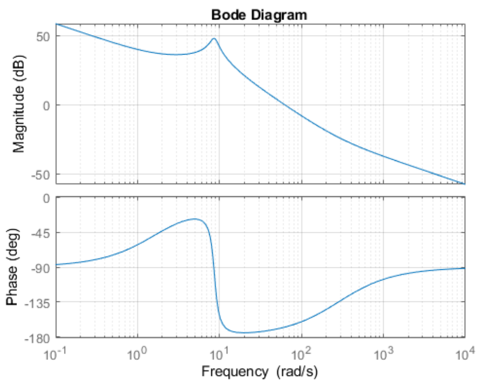

Figure 10 - Bode diagram of system given phase-lead compensator

The closed-loop step response of the system:

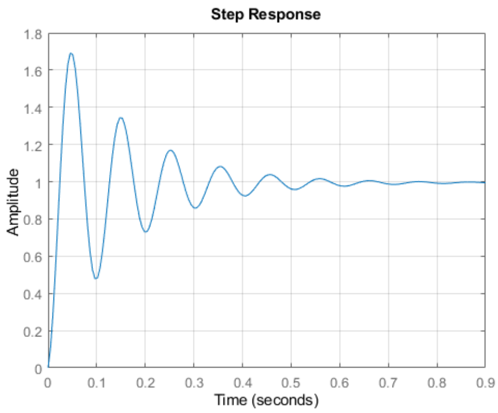

Figure 11 - Closed-loop step response of the system given a phase-lead compensator

The following graph represents the ramp tracking-error of the system with a phase-lead compensator, as you can see the ramp error requirement is not yet met:

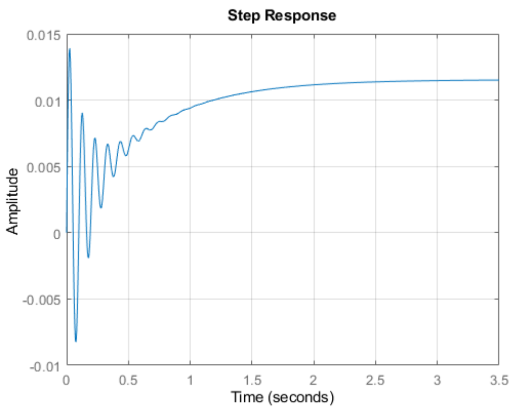

Figure 12 - Ramp tracking error of the system given a phase-lead compensator

2.4. Lead-lag compensator

Consider the lead-lag compensator:

$$C(s) = K_p \frac{\tau s + 1}{\alpha \tau s + 1} \frac{\beta T s + 1}{T s + 1}$$

The following *Matlab* script describes the variables used:

```
pm = 70; % >55
wc = wl % lower limit of wc
k_p = k_p2; %k_p satisfies velocity error

% Compute the frequency response of the system at the crossover frequency wc
p_wc = squeeze(freqresp(g_s, wc));
% Magnitude at wc
mag_p_wc = abs(p_wc);
% Phase at wc
phase_p_wc = angle(p_wc);

% find phi
phi = pm*pi/180 - pi - phase_p_wc
% Compute tz
tz = (mag_p_wc-cos(phi))/(wc*mag_p_wc*sin(phi));
%find alpha
alpha = (mag_p_wc*cos(phi)-1)/mag_p_wc*(mag_p_wc-cos(phi));
% Compute the Tz
Tz = (mag_p_wc*cos(phi)-1)/(wc*mag_p_wc*sin(phi));
%find beta
beta = mag_p_wc*(cos(phi)-mag_p_wc)/(1-mag_p_wc*cos(phi));

% compensator
c_s = k_p*(tz*s+1)/(alpha*tz+1)*(beta*Tz+1)/(Tz*s+1);
```

Figure 13 – Method of derivation of variable used for lead-lag compensator

The bode diagram of the system with the lead-lag compensator:

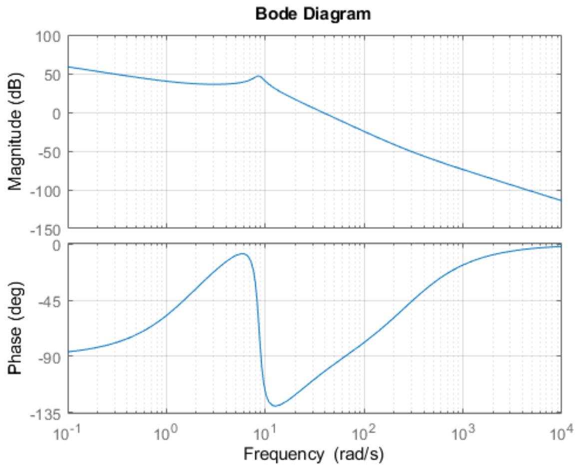

Figure 14 - Bode diagram for lead-lag compensator

The closed-loop step response of the system:

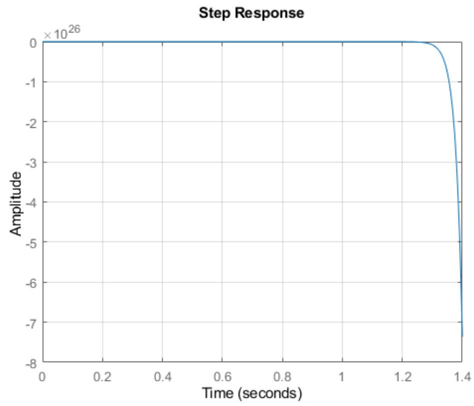

Figure 15 - Closed loop step response of the system given a lead-lag compensator

The following graph represents the ramp tracking-error of the system with a lead -lag compensator:

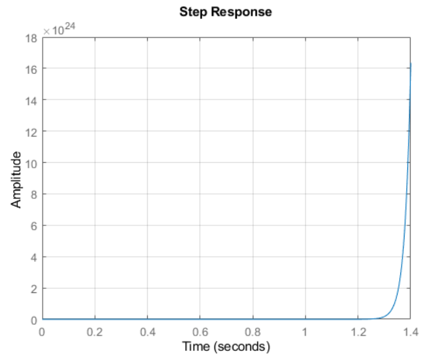

Figure 16 - Ramp tracking error of the system given a lead-lag compensator

3. Conclusion

All design tasks were completed to varying degrees of success. As you can see in section 2.4, the four criteria given in the design brief were successfully achieved: (1) the gain crossover frequency lay within the acceptable range; (2) the phase margin was 70, which was greater than the required 55; (3) the velocity error was smaller than the given ev; and, (4) the steady-state error in response to step inputs is zero (but only for certain values of time). Unfortunately, the final lead-lag compensator did not achieve an asymptotically stable closed-loop system.