CDS 101/110 Homework #7 Solution

Problem 1 (CDS 101, CDS 110): (15 points)

From (11.10),

$$k_i = \frac{a}{P(a)} = a(1-a)^2$$

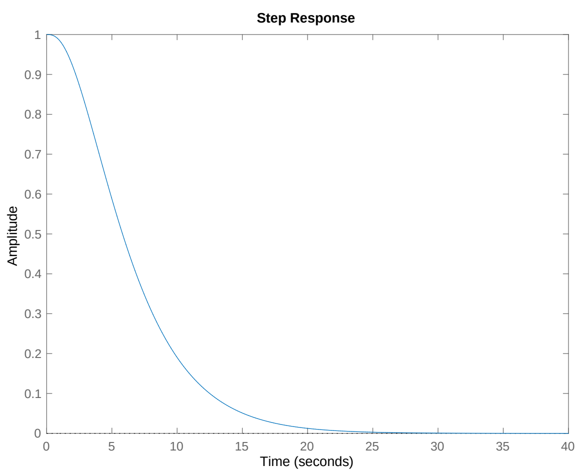
Note that the above equation is unbounded, so it does not make sense to talk about maximum in general. In this problem, we will assume that $a \leq 1$.

Then, the maximum is given by $a = 1/3$ which gives $k_i = 0.148$.

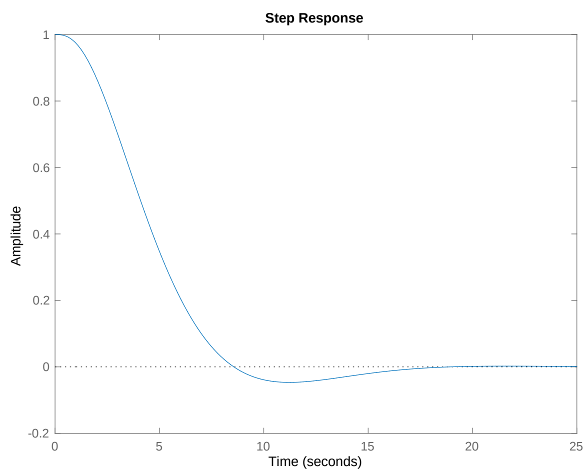
The closed loop poles of the system are $-1/3$, and $-4/3$.

The open loop poles are 0 an 1. Note that $-1/3$ is larger than 0 in magnitude, therefore it is not a dominant closed loop pole.

From (11.7), $k_i' = 0.25$.



(a) Using $a = 1/3$



(b) Using (11.7)

Note that there is a small overshoot when using k_i'.

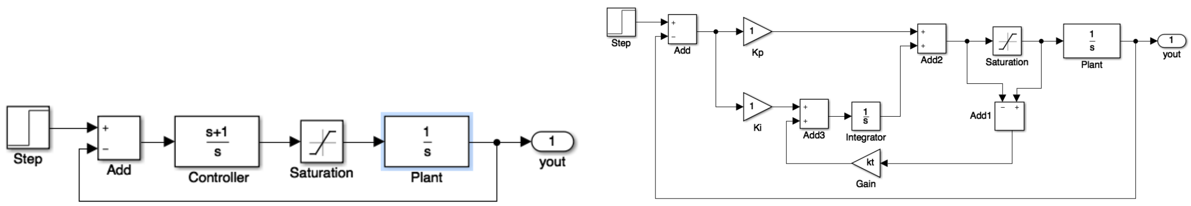
Code:

```
s = tf('s');
P = (s+1)^(-2);

% When a = 1/3
a = 1/3;
ki1 = a*(1-a)^2;
C1 = ki1/s;
CLP1 = 1/(1+P*C1);
```

```
figure
step(CLP1)

% When using (11.7)
ki2 = 0.25;
C2 = ki2/s;
CLP2 = 1/(1+P*C2);
figure
step(CLP2)
```

Problem 2 (CDS 101, CDS 110): (15 points)

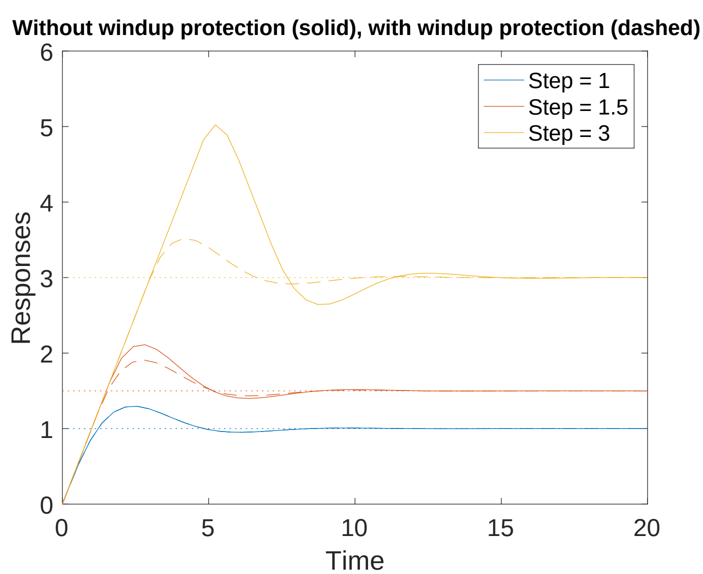
The systems are simulated using Simulink.



(c) Without anti-windup (d) With anti-windup

Figure 1: Block diagram of the feedback systems in Simulink

The responses are shown below when $k_t = 1$



With windup protection, the responses have a smaller overshoot. Furthermore, the responses are similar when the step input magnitude is 1 because the controller is not saturated in this case. When the step input magnitude becomes larger, the overshoot due to integral windup becomes larger, thus, the windup protection plays an more important role.

Code:

```
kt = 1;
stoptime = '20';

stepMagnitude = 1;
simOut = sim('problem11_10a','ReturnWorkspaceOutputs','on', ...
'StopTime', stoptime);
outputs = simOut.get('yout');
x1=(outputs.getElement(1).Values);

simOut = sim('problem11_10b','ReturnWorkspaceOutputs','on', ...
    'StopTime', stoptime);
outputs = simOut.get('yout');
y1=(outputs.getElement(1).Values);

stepMagnitude = 1.5;
simOut = sim('problem11_10a','ReturnWorkspaceOutputs','on', ...
    'StopTime', stoptime);
outputs = simOut.get('yout');
x2=(outputs.getElement(1).Values);

simOut = sim('problem11_10b','ReturnWorkspaceOutputs','on', ...
    'StopTime', stoptime);
outputs = simOut.get('yout');
y2=(outputs.getElement(1).Values);

stepMagnitude = 3;
simOut = sim('problem11_10a','ReturnWorkspaceOutputs','on', ...
    'StopTime', stoptime);
outputs = simOut.get('yout');
x3=(outputs.getElement(1).Values);

simOut = sim('problem11_10b','ReturnWorkspaceOutputs','on', ...
    'StopTime', stoptime);
outputs = simOut.get('yout');
y3=(outputs.getElement(1).Values);

cp = lines(3);

figure
plot(x1,'Color',cp(1,:)); hold on; plot(x2, 'Color',cp(2,:)); plot(x3, 'Color',cp(3,:));
plot([0 20],[1 1],':','Color',cp(1,:));
plot([0 20],[1.5 1.5],':','Color',cp(2,:));
plot([0 20],[3 3],':','Color',cp(3,:));
plot(y1,'--','Color',cp(1,:)); plot(y2,'--','Color',cp(2,:)); plot(y3,'--','Color',cp(3,:));
title('Without windup protection (solid), with windup protection (dashed)')
xlabel('Time');
ylabel('Responses');
legend('Step = 1', 'Step = 1.5', 'Step = 3')
```

Problem 3 (CDS 110): (30 points)

(a) The bode plot of the plant $P(s)$ is given by the blue line Figure 2. The triangle shows the steady state requirement at $20 \log 20 dB$. The blue box shows the tracking error requirement up to 2π rad/s at $20 \log 100 dB$. The red box shows the phase margin requirement at 30 degree.

(b) We want to design a controller such that $L(s) = P(s)C(s)$ is above the blue box with zero frequency gain at the triangle, and a phase margin above the read box.

The steady state frequency of the $P(s)$ is approximately $-10dB$. So, we can set the proportional gain to $k_p = 300 \sim 50dB$ so that $L(s)$ is approximately $40dB$ in steady state. To obtain the required phase margin, we can include a derivative term into the controller at frequency $\omega_d = 20\pi$ rad/s (i.e. $k_d = k_p/\omega_d = 15/\pi$) that will shift the phase margin to approximately 36 degree.

So, controller is $C(s) = 300 + 15/\pi s$. We do not need the integral term in this case. The yellow line in Figure 2 shows that $L(s)$ satisfies all the requirements drawn for part(a).

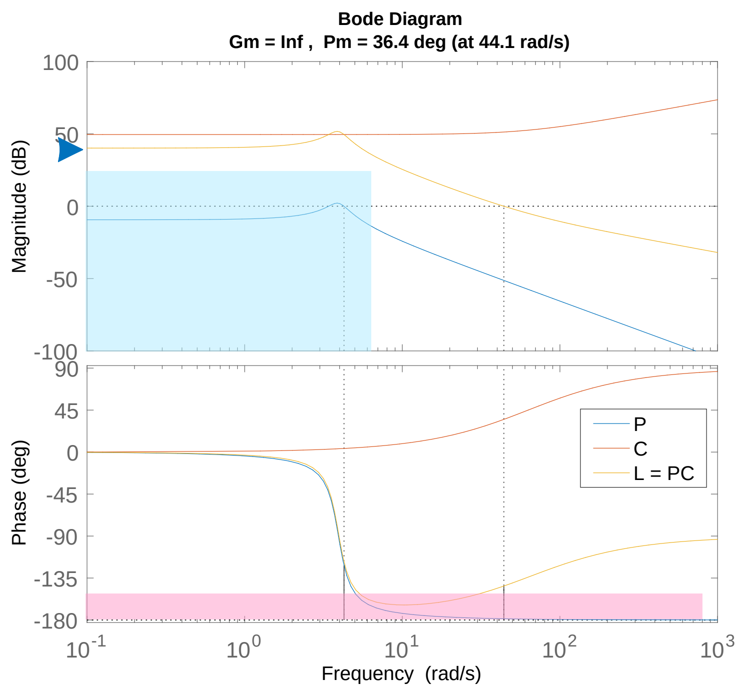
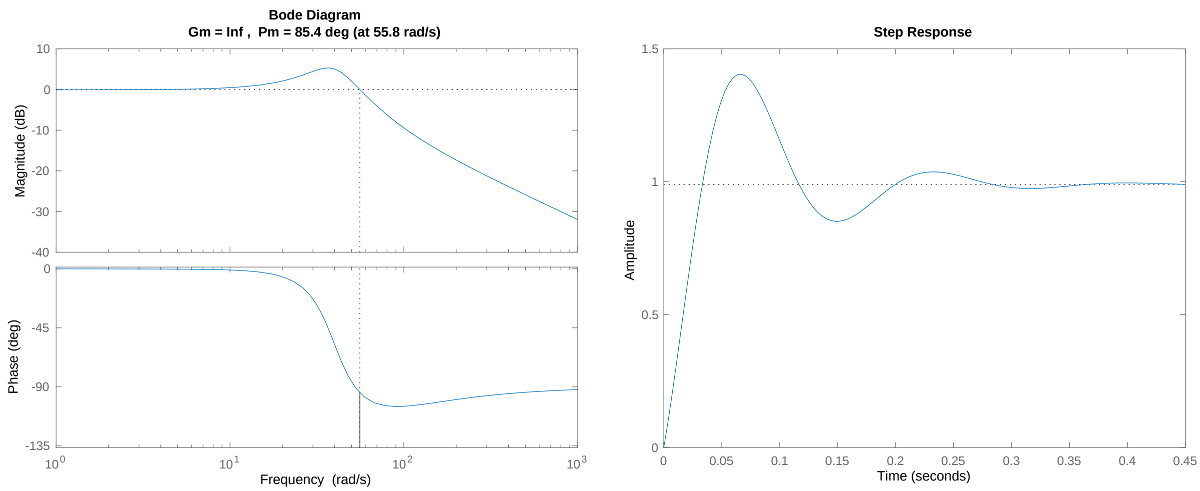


Figure 2

(c) The frequency response and step response are shown in the figure.



(a) Frequency response

(b) Step response

The steady state under step input is given by

$$\lim_{s\to 0} \frac{L(s)}{1+L(s)} = \lim_{s\to 0} \frac{0.0567s^3 + 3.622s^2 + 4.627s + 55.13}{0.002256s^4 + 0.06145s^3 + 3.695s^2 + 4.701s + 55.67} \sim 0.9903$$

Thus, the steady state error is less than 1%.

The rise time is 0.0259s (10% - 90%), the overshoot is 41.81, and the settling time is 0.2656s (2%). These values can be estimated from the step response.

Code:

```
s = tf('s');

g = 9.8;
m = 1.5;
c = 0.05;
l = 0.05;
J = 0.0475;
r = 0.25;

P = r/(J*s^2 + c*s + m*g*l);

% part (a)
figure(1)
margin(P)

% part (b)
kp = 300;
wd = 2*pi*10;
kd = kp/wd;
wi = 0;
ki = kp*wi;

C = kp+kd*s+ki/s;
```

```
figure(2)
margin(P*C)

% part (c)
CLP = P*C/(1+P*C);

figure(3)
margin(CLP)

figure(4)
step(CLP)
stepinfo(CLP)
```

Problem 4 (CDS 110): (20 points)

The transfer function of the full system is given by

$$P(s) = C(sI - A)^{-1}B = \frac{k_I(cs + k)}{s^2(J_1J_2s^2 + k_d(J_1 + J_2)s + k(J_1 + J_2))}$$

For low frequencies (s is small), the transfer function can be approximated by

$$P(s) = \frac{k_I}{(J_1 + J_2)s^2} \triangleq \frac{b_p}{s^2}$$

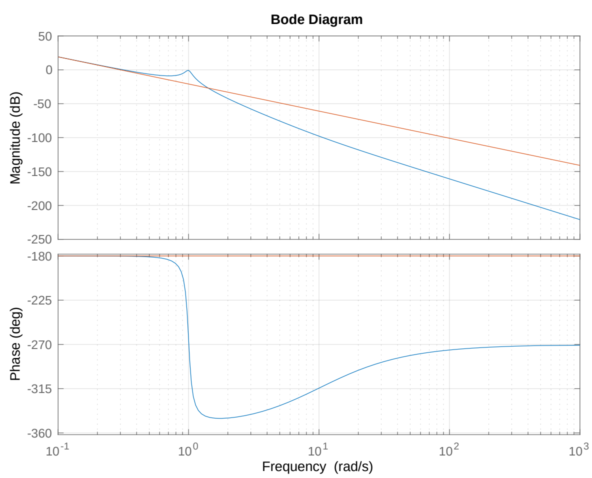


Figure 3: Blue - real model, red - approximate model

Note that the approximated model and the real model are pretty similar up to about $\omega = 0.12$ rad/s.

A PD controller has the form $C(s) = k_p + k_d s$. Thus, the characteristics polynomial of the closed loop transfer function is given by the numerator of $1 + P(s)C(s)$ which is

$$s^2 + b_p k_d s + b_p k_p$$

The desired closed loop characteristic polynomial from the given poles is $s^2 + 2\zeta\omega_0 s + \omega_0^2$. Thus,

$$k_p = \frac{\omega_o^2}{b_p} \qquad k_d = \frac{2\zeta\omega_0}{b_p}$$

With low pass filter, the controller becomes

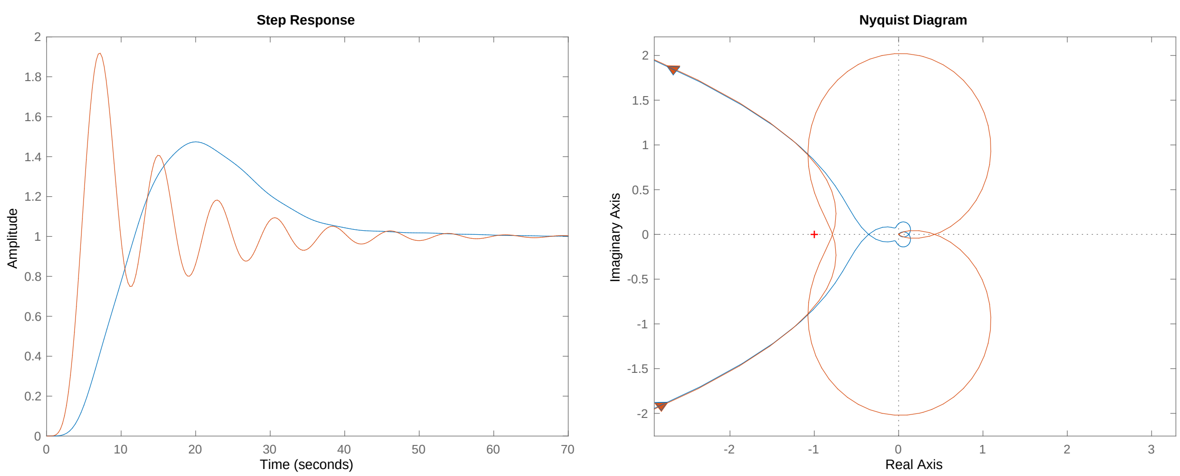
$$C(s) = \frac{k_p + k_d s}{1 + sT_f + s^2 T_f^2/2}$$

where $T_f = k_d/(Nk_p)$ and $N = 5$ is chosen for the simulation. This is arbitrary.

To obtain a well damped response we choose $\zeta = 0.8$ and it remains to find a suitable value of the design parameter ω_0. A large value of ω_0 gives a fast response. When the second order approximated model is valid, response time increases inversely with ω_0 and so does the control signal, because the real part of the poles of both $P(s)$ and $C(s)$ scale linearly with ω_0.

The frequency range where the model is valid ($\omega \leq 0.12$) gives a bound on possible values of ω_0. Note the oscillation frequency is $\omega_0\sqrt{1 - 0.8^2} = 0.6\omega_0$. Requiring that this frequency be less than 0.12 we find that a reasonable upper limit of the design parameter is $\omega_0 = 0.2$ and a 2% settling time of about $50s$.

To check the results, simulate the closed loop system with full model when $\omega_0 = 0.08, 0.2$.



(a) Step response

(b) Nyquist diagram

Figure 4: Blue - $\omega_0 = 0.08$, red - $\omega_0 = 0.2$

You can see that the systems are stable for both cases but less so when ω_0 is larger. Oscillation is observed when $\omega_0 = 0.2$. The stability margin is smaller when $\omega_0 = 0.2$. The settling time is aprroximately 47s when $\omega_0 = 0.08$, and 50s when $\omega_0 = 0.2$. We can conclude that the system can be controlled with a PD controller provided that the requirements on the response time are not too stringent.

Comparing with the step response using state feedback from Exercise 7.12, the response time using state feedback is approximately one order of magnitude smaller.

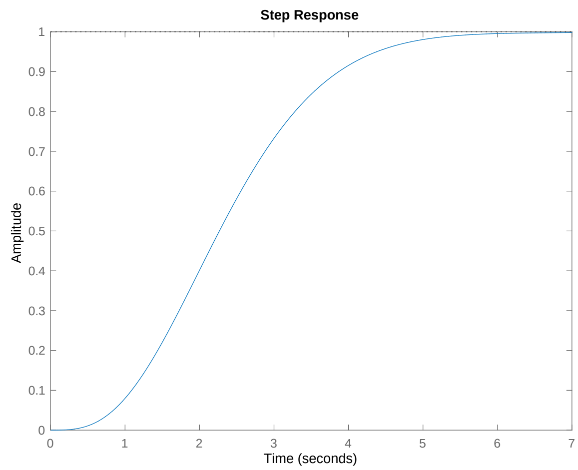


Figure 5: Step response from Exercise 7.12

Note that the model used for state feedback is valid for higher frequency. Therefore, the controller can make the system response faster with larger magnitude of u.