
enae432 hw09

Table of Contents

problem 1	1
part c	2
part e	2
problem 2	2
problem 3	2
problem 4	3
part a	3
part b	3
part c	3
part d	3
part e	4

vai srivastava

problem 1

```
I = 5;
b = 1;
Km = 3;

s = tf('s');
G = Km/(I*s + b);
% part a

K_test = 1;

H_0 = K_test;
L_0 = H_0*G;
T_0 = feedback(L_0, 1)
disp(pole(T_0))
% part b

t_s_des = 2;
omega_d = 50;

sigma_desired = 4/t_s_des
K = (I*sigma_desired - 1)/Km

H = K;
L = H*G;
T = feedback(L, 1)
disp(stepinfo(T))

y_ss = dcgain(T) * omega_d
e_ss = omega_d - y_ss
```

part c

```
syms Kp_sym Ki_sym s_sym

G_sym = Km/(I*s_sym + b);
H_sym = (Kp_sym*s_sym + Ki_sym)/s_sym;
L_sym = H_sym * G_sym;
T_sym = L_sym/(1 + L_sym);

dc_gain = limit(T_sym, s_sym, inf)
```

part e

```
Ki = 20/3;
Kp = 19/3;

H = pid(Kp, Ki);
L = H*G;
T = feedback(L, 1);
disp(stepinfo(T))
```

problem 2

```
s = tf('s');
G = 10/(s*(s+4)^2);

[Gm, Pm, Omegacg, Omegacp] = margin(G)

Ku = Gm
Tu = 2*pi/Omegacp

Kp = 3/5*Ku;
Ki = 2*Kp/Tu;
Kd = Kp*Tu/8;

H = pid(Kp, Ki, Kd)
L = H*G;
T = feedback(L, 1);
disp(stepinfo(T))
```

problem 3

```
s = tf('s');
G = 10/(s*(s+4)^2);

[H, info] = pidtune(G, 'PID', 4, pidtuneOptions(PhaseMargin = 40, DesignFocus
= "disturbance-rejection"))

L = H*G;
```

```
T = feedback(L, 1)
disp(stepinfo(T))
```

problem 4

```
Kp = 9.81;
Ki = 9.14;
Kd = 2.63;

s = tf('s');
G = 10/(s*(s+4)^2);
H = pid(Kp, Ki, Kd)
L = H*G
T = feedback(L, 1);
```

part a

```
Dm = allmargin(L).DelayMargin
Ur_min = 1/Dm
```

part b

```
tau = 0.05
Ld = L * exp(-s*tau)

[Gm, Pm, Omegacg, Omegacp] = margin(Ld)
Gm_mag = 20*log10(Gm)
```

part c

```
Td = G/(1+L);

omega = logspace(-2,3,20000);
mag = abs(freqresp(Td, omega));

omega_err = omega(mag > 0.01);
band = [min(omega_err), max(omega_err)]

[mag_max, i_max] = max(mag)
omega_peak = omega(i_max)
```

part d

```
Ljomega = squeeze(freqresp(L, omega));
Ljomegad = squeeze(freqresp(Ld, omega));

dist = abs(Ljomega + 1);
dstd = abs(Ljomegad + 1);
```

```
[minDist, i] = min(dist)
[minDistd, id] = min(distd)
```

part e

```
a_vals = logspace(-3,2,200);
ok = false(size(a_vals));

for k = 1:numel(a_vals)
    a = a_vals(k);
    Delta = a*s/(a*s+1);
    M = abs(freqresp(Delta*T, omega));
    if max(M) < 1
        ok(k) = true;
    end
end

a_max = max(a_vals(ok))
```

Published with MATLAB® R2024b