

Övning 11 (Datorövning 3)

Introduktion

Varmt välkomna till tredje datorövningen i Reglerteknik AK!

Håkan Terelius hakante@kth.se

Endast en kort genomgång, sedan hjälper vi till och svarar på frågor då ni arbetar på med övningarna.

Längst bak i exempelsamlingen finns en lista på användbara kommandon i MATLAB.

Repetition

Definiera överföringsfunktioner genom att först definiera symbolen s :

```
s = tf('s');  
G = (2*s+1)/(3*s^2+s+4)
```

MATLAB har inbyggd hjälp-dokumentation, använd kommandona *help* och *doc*

```
help tf  
doc tf
```

Första gången använde vi funktionerna *step*, *pole* och *tzero*

```
step(G)  
pole(G)  
tzero(G)
```

Förra gången ritade vi rotorter med *rlocus*. Identifiera polynomet $P(s) + KQ(s) = 0$, och använd sedan

```
rlocus(Q/P)
```

Nyquistkurvan och Bodediagrammet ritades med kommandona

```
nyquist(G)  
bode(G)  
margin(G)
```

Genomgång

För att simulera med en godtycklig insignal kan ni använda kommandot *lsim*

```
t = 0:0.01:10
ref = t.*t
lsim(G, ref, t)
```

För att generera tillståndsåterkopplingen används kommandot *ss*

```
Gss = ss(G)
Gss = ss(A,B,C,D)
```

Och för att gå tillbaka till överföringsfunktionen används *tf*

```
tf(Gss)
```

Polplacering för ett system kan göras med kommandot *place*

```
L = place(A, B, poles)
```

Den statiska förstärkningen fås ut med *dcgain*

```
dcgain(Gc)
```

Övningar

- 5.13
- 6.10
- 9.14ab

5.13

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

```
%% Part A
G = 725 / ( (s+1)*(s+2.5)*(s+25) );
F = 1;
```

```
figure(1);
margin(F*G)
```

```
%% Part B
```

```

% Increase phase margin by 60-27+6 = 39 degrees
wcd = 5;
beta = 0.22;
taud = 1 / (wcd * sqrt(beta));
% Gain K * 1/sqrt(beta) * abs(G(5i)) = 1
K = sqrt(beta);

% Lag-part
gamma = 0;
taui = 10/wcd;

Flead = (taud*s+1)/(taud*beta*s+1);
Flag = (taui*s+1)/(taui*s+gamma);
Fleadlag = K*Flead*Flag;
Gc = feedback(Fleadlag*G,1);

figure(2)
margin(Fleadlag*G);
figure(3);
step(Gc);

%% Part C
Gc0 = feedback(F*G,1);
figure(4);
bode(Gc0, Gc)

%% Part D
S = 1 / (1+Fleadlag*G); % Sensitivity function
t = 0:0.05:50; % Time vector
r = t; % Ramp input
y = lsim(S,r,t); % Output

figure(5);
plot(t,y);

```

6.10

```

s = tf('s');

G = 725 / ( (s+1)*(s+2.5)*(s+25) );
wcd = 5;
beta = 0.22;
taud = 1 / (wcd * sqrt(beta));
K = sqrt(beta);
gamma = 0;
taui = 10/wcd;

```

```

Flead = (taud*s+1)/(taud*beta*s+1);
Flag = (taui*s+1)/(taui*s+gamma);
Fleadlag = K*Flead*Flag;

Gc = feedback(Fleadlag*G,1);

%% Part A
deltaG = -s/(s+1);

%% Part B
figure(1);
bode(1/deltaG, Gc, feedback(G,1));
legend('1/deltaG', 'lead-lag', 'proportional');

```

9.14ab

```

s = tf('s');

%% Part A
G = 1 / ( s*(s+1) );
S = ss(G);

% x2 = angle
% x1 = d(x2)/dt = angular velocity

%% Part B
L = place(S.a, S.b, [-2.2 -2.1]);
Gc = ss(S.a - S.b*L, S.b, S.c, S.d);
l0 = 1/dcgain(Gc);
Gc = Gc*l0;

[y,t,x] = step(Gc,5);
u = 10 - x*L.';

figure(1);
plot(t,y,t,u);
axis([0 5 -1 5]);

L = place(S.a, S.b, [-1+i -1-i]);
Gc = ss(S.a - S.b*L, S.b, S.c, S.d);
l0 = 1/dcgain(Gc);
Gc = Gc*l0;

[y,t,x] = step(Gc,5);
u = 10 - x*L.';

```

```

figure(2);
plot(t,y,t,u);
axis([0 5 -1 5]);

%% Part C
polesC = zeros(2,3);
% Case i)
Q = diag([0 1]);
R = 1;
L = lqr(S, Q, R);
Gc = ss(S.a - S.b*L, S.b, S.c, S.d);
l0 = 1/dcgain(Gc);
Gci = Gc*l0;

polesC(:,1) = pole(Gci);

% Case ii)
Q = diag([0 10]);
R = 1;
L = lqr(S, Q, R);
Gc = ss(S.a - S.b*L, S.b, S.c, S.d);
l0 = 1/dcgain(Gc);
Gcii = Gc*l0;

polesC(:,2) = pole(Gcii);

% Case iii)
Q = diag([0 0.1]);
R = 1;
L = lqr(S, Q, R);
Gc = ss(S.a - S.b*L, S.b, S.c, S.d);
l0 = 1/dcgain(Gc);
Gciii = Gc*l0;

polesC(:,3) = pole(Gciii);

figure(3)
step(Gci,Gcii,Gciii)
axis([0 20 0 1.4])

%% Part D

figure(4);
clf
polesD = zeros(2,10);
for R = 1:10
    Q = diag([0 10]);

```

```

    L = lqr(S, Q, R);
    Gc = ss(S.a - S.b*L, S.b, S.c, S.d);
    l0 = 1/dcgain(Gc);
    Gc = Gc*l0;
    step(Gc);
    hold on
    polesD(:,R) = pole(Gc);
end

axis([0 20 0 1.4])

%% Part E
R = 1;

figure(5);
clf;
polesE = zeros(2,10);
for W = 1:10
    Q = diag([W 1]);
    L = lqr(S, Q, R);
    Gc = ss(S.a - S.b*L, S.b, S.c, S.d);
    l0 = 1/dcgain(Gc);
    Gc = Gc*l0;
    step(Gc);
    hold on
    polesE(:,W) = pole(Gc);
end

```