Scilab Textbook Companion for Process Systems Analysis And Control by S. E. LeBlanc And D. R. Coughanowr[1]

Created by
K. Dheemanth
B.Tech. (pursuing)
Chemical Engineering
UCT (A), Osmania University
College Teacher
Dr. V. Ramesh Kumar, UCT (A), Osmania University
Cross-Checked by
Prashant Dave, IIT Bombay

August 12, 2013

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Process Systems Analysis And Control

Author: S. E. LeBlanc And D. R. Coughanowr

Publisher: McGraw - Hill International

Edition: 2

Year: 1991

ISBN: 0-07-100807-1

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

List of Figures

Chapter 2

The Laplace Transform

Scilab code Exa 2.1 Laplace transform

```
//Example 2.1
syms t s;
fs=laplace('1',t,s);
disp(fs,'f(s)=')
```

Scilab code Exa 2.3 Laplace transform

```
//Example 2.3
clc
s=%s;
xs=2/(s+3);
disp(xs,'x(s)=')
syms t;
xt=ilaplace(xs,s,t);
disp(xt,'x(t)=')
```

Chapter 3

Inversion by Partial Fractions

Scilab code Exa 3.1 Inverse laplace transform

```
//Example 3.1
clc
s=%s;
xs=1/(s*(s+1));
disp(xs,'x(s)=')
syms t;
[A]=pfss(xs)
F1=ilaplace(A(1),s,t);
F2=ilaplace(A(2),s,t);
xt=F1+F2;
disp(xt,'x(t)=')
```

Scilab code Exa 3.2 Inverse laplace transform

```
//Example 3.2
clc
s=%s;
syms t;
```

```
num=poly([-8 9 -6 0 1],'s','coeff');
den=s*(s-2)*poly([-2 -1 2 1],'s','coeff');
xs=syslin('c',num/den);
disp(xs,'x(s)=')
A=pfss(xs)
F1=ilaplace(A(1),s,t);
F2=ilaplace(A(2),s,t);
F3=ilaplace(A(3),s,t);
F4=ilaplace(A(4),s,t);
F5=ilaplace(A(5),s,t);
xt=F1+F2+F3+F4+F5;
disp(xt,'x(t)=')
```

Scilab code Exa 3.3 Inverse laplace transform

```
//Example 3.3
clc
s=%s;
syms t;
xs=2/(s*(s^2+2*s+2));
disp(xs,'x(s)=')
[A]=pfss(xs)
F1=ilaplace(A(1),s,t);
F2=ilaplace(A(2),s,t);
xt=F1+F2;
disp(xt,'x(t)=')
```

Scilab code Exa 3.4 Inverse laplace transform

```
//Example 3.4
clc
s=%s;
syms t;
```

```
xs=2/((s^2+4)*(s+1));
disp(xs,'x(s)=')
[A]=pfss(xs)
F1=ilaplace(A(1),s,t);
F2=ilaplace(A(2),s,t);
xt=F1+F2;
disp(xt,'x(t)=')
```

Scilab code Exa 3.5 Inverse laplace transform

```
//Example 3.5
clc
s=%s;
syms t;
xs=1/(s*(s^2-2*s+5));
disp(xs,'x(s)=')
[A]=pfss(xs)
F1=ilaplace(A(1),s,t);
F2=ilaplace(A(2),s,t);
xt=F1+F2;
disp(xt,'x(t)=')
```

Scilab code Exa 3.6 Inverse laplace transform

```
//Example 3.6
clc
s=%s;
syms t;
xs=1/(s*(s^3+3*s^2+3*s+1));
disp(xs,'x(s)=')
[A]=pfss(xs)
F1=ilaplace(A(1),s,t);
F2=ilaplace(A(2),s,t);
```

```
xt=F1+F2;
disp(xt,'x(t)=')
```

Chapter 4

Further Properties of Transforms

Scilab code Exa 4.1 Final value theorem

```
//Example 4.1
clc
s=%s;
num=poly(1,'s','coeff');
den=s*poly([1 3 3 1],'s','coeff');
xs=num/den;
disp(xs,'xs=')
syms s;
xt=limit(s*xs,s,0);//final value theorem
disp(xt,'x(t)=')
```

Scilab code Exa 4.2 Final value theorem

```
//Example 4.2
clc
s=%s;
```

```
num=poly([-8 0 9 -6 1],'s','coeff');
den=s*(s-2)*poly([-2 -1 2 1],'s','coeff')
xs=num/den;
disp(xs,'x(s)=')
disp(s*xs,'s*x(s)=')
[A]=pfss(s*xs)
printf("since xs becomes infinite for s=1 and s=2,
    the conditions of the final value theorem are not
    satisfied\n")
printf("Final value theorem is not applicable \n")
```

Scilab code Exa 4.4 Laplace transform

```
//Example 4.4
clc
syms t s a k;
xt=laplace('%e^(-a*t)*cos(k*t)',t,s);
disp(xt,'x(t)=')
x
```

Chapter 5

Response of First Order Systems

Scilab code Exa 5.1 First order systems

```
//Example 5.1
tau=0.1;//min
xs=90;//degrees
A=10;//degrees
Y_inf=10;//degrees
Y_t=8;//degrees
//Substituting into Eq.(5.12) the appropriate values
    of Y_t,A,and tau gives
t=-0.1*logm(1-(Y_t/A));//min
disp('min',t,'time=')
```

Scilab code Exa 5.2 First order systems

```
//Example 5.2
clear all
clc
```

```
tau=0.1;//min
xs=100;//Fahrenheit
ys=100;//Fahrenheit
A=2;//Fahrenheit
f=10/%pi;//cycles/min
w=2*%pi*f;//rad/min
//From Eq.(5.25), the amplitude of the response and
    the phase angle are calculated;thus
disp('Fahrenheit',A/sqrt((tau*w)^2+1),'A/sqrt((tau*w
    )^2+1)=')
phi=atan(-w*tau);//radians
phi=phi*180/%pi;//degrees
disp('degrees',phi,'phase lag=')
t=0:0.01:1;
//From Eq.(5.19), the input of the thermometer is
    therefore
disp("X(t)=2*sin(20*t)");
//or
xt=xs+2*sin(20*t);
//The response of the thermometer is therefore
disp("Y(t)=0.8944*sin(20*t-63.4349)")
//or
yt=ys+0.8944*sin(20*t-63.4349);
Lag=phi/(360*f);//min
Lag=abs(Lag);//min
disp('min',Lag,'Lag=')
clf;
plot(t,yt)
plot(t,xt)
xlabel('time')
ylabel('x(t) , y(t)')
title('x(t) , y(t) Vs time')
xgrid
```

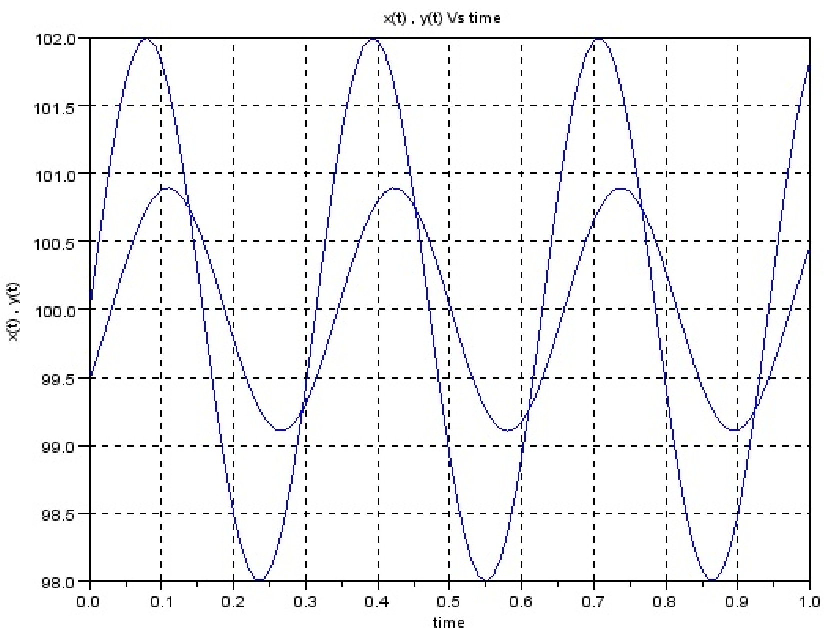

Figure 5.1: First order systems

Chapter 6

Physical Examples of First Order Systems

Scilab code Exa 6.1 First order systems

```
//Example 6.1
clc;
syms s t;
tau=1;//min
R=1/9;//ft/cfm
A=9;
//from Equation 6.8
g=R/(tau*s+1);
disp(g,'H(s)/Q(s)=')
//from Example 4.5
disp('Q(t)=90[u(t)-u(t-0.1)')
//where u(t) is a unit step function, the laplace
    transform of it gives
Qs=90*(1-exp(-0.1*s))/s
disp(Qs,'Q(s)=')
Hs=Qs*g;
disp(Hs,'H(s)=')
//taking first term for t<0.1, the second term goes
    equals to zero
```

```
Ht=ilaplace('10*(1/(s*(s+1)))',s,t);//t<0.1
disp(Ht,'H(t)=')
disp('H(t)=10(1-yexp(-(-t-0.1)))')//t>0.1
Ht=10*((1-exp(-t))-(1-exp(-(-t-0.1))));
disp(Ht,'H(t)=')
//from Eq.(5.16)
Ht=R*A*exp(-(t/tau));//impulse
disp(Ht,'H(t)=')
```

Chapter 7

Response of First Order Systems in Series

Scilab code Exa 7.1 First order systems

```
//Example 7.1
clc
s=%s;
tau1=0.5;
tau2=1;
R2=1;
//From Eq.(7.8)
g=R2/((tau1*s+1)*(tau2*s+1))
disp(g,'H2(s)/Q(s)=')
Qs=1/s;
H2s=g*Qs;
disp(H2s,'H2(s)=')
syms t;
H2t=ilaplace(H2s,s,t);
disp(H2t,'H2(t)=')
```

Chapter 10

Controllers and Final Control Elements

Scilab code Exa 10.1 Control system

```
//Example 10.1
clear
clc
t1=60;//Fahrenheit
t2=100;//Fahrenheit
p1=3;//psi
p2=15;//psi
T1=71;//Fahrenheit
T2=75;//Fahrenheit
pb=((T2-T1)/(t2-t1))*100;
disp('%',pb,'proportional band=')
Gain=(p2-p1)/(T2-T1);
disp('psi/F',Gain,'Gain=')
//Assume pb is changed to 75% then
pb=75;//%
T=(pb*(t2-t1))/100;
disp('Fahrenheit',T,'T=')
Gain=(p2-p1)/T;
disp('psi/F',Gain,'Gain=')
```

Chapter 12

Closed Loop Transfer functions

Scilab code Exa 12.1 Transfer functions

```
//Example 12.1
clc
syms Gc G1 G2 G3 H1 H2 R U1;
G=Gc*G1*G2*G3*H1*H2;
g=Gc*G1*G2*G3/(1+G);
disp(g,'C/R=')
g1=G2*G3/(1+G);
disp(g1,'C/U1=')
g2=G3*H1*H2/(1+G);
disp(g2,'B/U2=')
C1=g*R;
C2=g1*U1;
disp(C1+C2,'C=')
```

Scilab code Exa 12.2 Transfer functions

```
//Example 12.2
clc
```

```
syms Gc1 Gc2 G1 G2 G3 H1 H2;
Ga=Gc2*G1/.H2
Gb=G2*G3
g=Gc1*Ga*Gb/.H1;
g=simple(g);
disp(g,'C/R=')
```

Chapter 14

Stability

Scilab code Exa 14.1 Stability

```
//Example 14.1
clear
clc
s=%s;
G1=10*((0.5*s+1)/s);
G2=1/(2*s+1);
H=1;
G=G1*G2*H
//The characteristic equation is therefore
disp('1+G=0')
disp('=0',1+G,'1+G=');
//which is equivalent to
disp("s^2+3*s+5=0");
h=poly([5,3,1],'s','coeff');
r=roots(h)
disp(r,'roots=')
//Since the real part of roots are negative, the
   system is stable
n=length(r);
c=0;
for i=1:n
```

```
if (real(r(i,1))<0)
c=c+1;
end
end
if(c>=1)
printf("system is stable\n")
else ("system is unstable")
end
```

Scilab code Exa 14.2 Stability

```
//Example 14.2
clear;
clc
h=poly([2,4,5,3,1],'s','coeff');
r=routh_t(h)
//Since there is no change in sign in the first
   column, there are no roots having positive real
   parts, and the system is stable.
y=coeff(h);
n=length(y);
c=0;
for i=1:n
if (r(i,1)<0)
c=c+1;
end
end
if(c>=1)
printf("system is unstable")
else ("system is stable")
end
```

Scilab code Exa 14.3 Stability

```
//Example 14.3
clc
syms Kc s s3;
G1=1/((s+1)*(0.5*s+1));
H=3/(s+3);
G=Kc*G1*H;
G=simple(G);
//The characteristic equation is therefore
disp('1+G=0')
disp('=0',1+G,'1+G=');
//which is equivalent to
disp("s^3+6*s^2+11*s+6+6*Kc=0")
routh=[1 11;6 6+6*Kc]
routh=[routh;-det(routh(1:2,1:2))/routh(2,1),0]
routh=[routh;-det(routh(2:3,1:2))/routh(3,1),0]
routh=simple(routh)
disp('>0',routh(3,1))
disp('Kc<10')
Kc=10;
routh=horner(routh,Kc);
routh=dbl(routh)
C=routh(2,1);
D=routh(2,2);
p=poly([D 0 C],'s','coeff')
disp('6*s^2+66=0')
r=roots(p)
disp('=0',simple((s-r(1,1))*(s-r(2,1))*(s-s3)))
//On comparing with the equation
poly([6+6*Kc 11 6 1],'s','coeff')
//we get
s3=-6;
printf("s1=3.3166248*i,s2=3.3166248*i,,s3=6\n")
```

Scilab code Exa 14.4 Stability

```
//Example 14.4
clc
s=%s;
tau1=1;
tau2=1/2;
tau3=1/3;
taui=0.25;
Kc=5;
n=Kc/(tau1*tau2*tau3)*(taui*s+1);
d=taui*s*(s+(1/tau1))*(s+(1/tau2))*(s+(1/tau3));
G=syslin('c',n/d);
//The characteristic equation is therefore
disp('1+G=0')
disp('=0',1+G,'1+G=');
//which is equivalent to
disp("s^4+6*s^3+11*s^2+36*s+120=0")
h=poly([120 36 11 6 1],'s','coeff')
r=routh_t(h)
y=coeff(h);
n=length(y);
c=0;
for i=1:n
if (r(i,1)<0)
c=c+1;
end
end
if(c>=1)
printf("system is unstable\n")
else ("system is stable")
end
```

Chapter 15

Root Locus

Scilab code Exa 15.1 Root locus

```
//Example 15.1
clc
s=%s;
syms K;
N=1;
D=poly([-1 -2 -3],'s','roots');
G=syslin('c',N/D);
disp(K*G,'G=')
evans(G)
v=[-3.5 3.5 -6 6];
mtlb_axis(v);
xgrid
```

Scilab code Exa 15.2 Root locus

```
//Example 15.2
clc
```

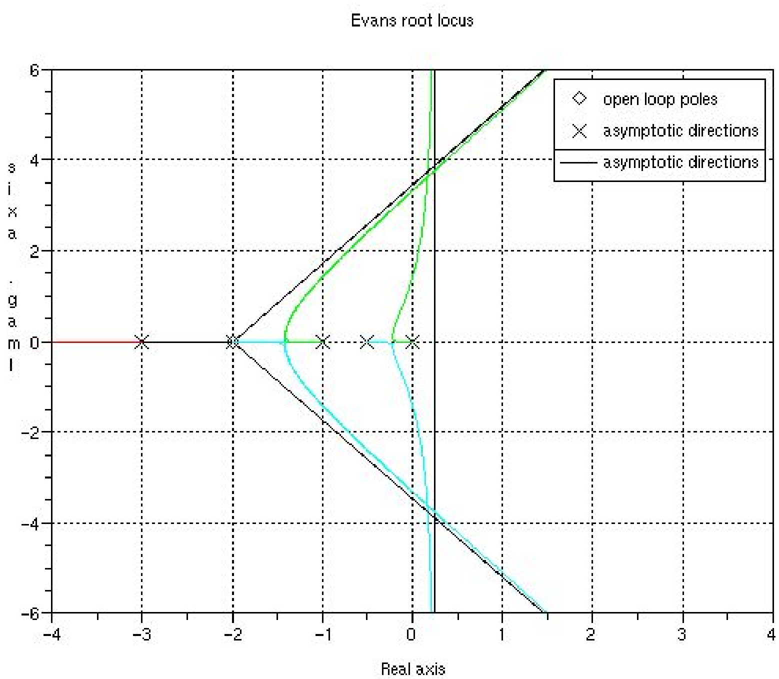

Figure 15.1: Root locus

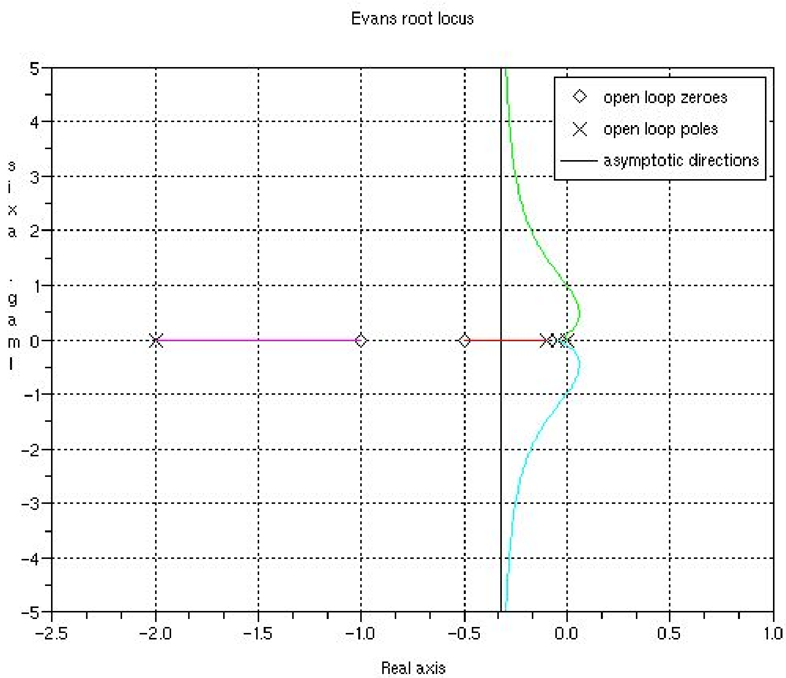

Figure 15.2: Root locus

```
s=%s;
syms Kc;
N=1+(2*s/3)+1/(3*s);
D=(20*s+1)*(10*s+1)*(0.5*s+1);
G=N/D;
G=syslin('c',G);
disp(Kc*G,'G=')
clf
evans(G)
v=[-2.5 1 -5 5];
mtlb_axis(v);
xgrid
```

Chapter 16

Introduction To Frequency Response

Scilab code Exa 16.1 Frequency Response

```
//Example 16.1
clc
s=%s;
j=%i;
f=10/%pi;
w=2*%pi*f;
G=1/(0.1*s+1);
s=w*j;
Gs=horner(G,s);
disp(Gs,'G(20j)=')
[r,theta]=polar(Gs)
theta=theta*180/%pi;
disp('degrees',theta,'theta=')
```

Scilab code Exa 16.2 Frequency Response

```
//Example 16.2
clc
syms tau s zeta w;
j=%i;
n=1;
d=tau^2*s^2+2*zeta*tau*s+1;
G=n/d
s=j*w;
G=1/(2*s*tau*zeta+s^2*tau^2+1)
[num den]=numden(G)
d=abs(den)
cof_a_0=coeffs(den,'%i',0)
cof_a_1=coeffs(den,'%i',1)
AR=1/d
theta=AR*atan(-cof_a_1/cof_a_0);
disp(theta,'Phase angle=')
```

Scilab code Exa 16.4 Bode diagram

```
//Example 16.4
clc
s=%s;
H=1/(s+1);
Hs=syslin('c',H)
J=1/(s+5);
Js=syslin('c',J)
G=Hs*Js;
Gs=syslin('c',G)
clf
bode([Hs;Js;Gs;])
legend(['1/(s+1)';'1/(s/5+1)';'1/(5*(s+1)*(s/5+1))'
  ])
```

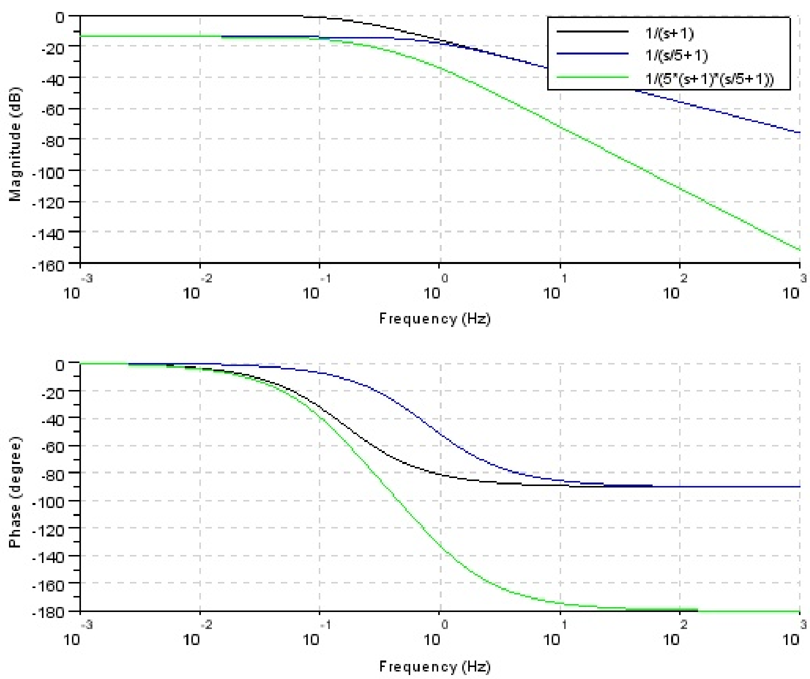

Figure 16.1: Bode diagram

Scilab code Exa 16.5 Bode diagram

```
//Example 16.5
clc
s=poly(0,'s');
disp("G=10*(0.5*s+1)*exp(-s/10)/(((s+1)^2)*(0.1*s+1))")
printf("exp(-0.1*s)=(2-0.1*s)/(2+0.1*s)\n")
G=10*(0.5*s+1)*(2-0.1*s)/(((s+1)^2)*(0.1*s+1)*(2+0.1*s));
Gs=syslin('c',G)
clf
bode(Gs)
```

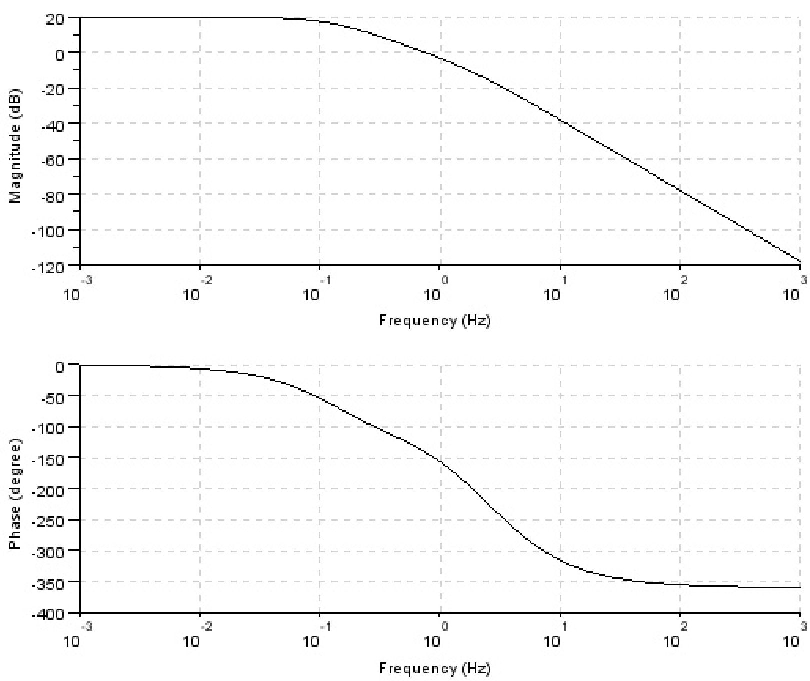

Figure 16.2: Bode diagram

Chapter 17

Control System Design By Frequeny Response

Scilab code Exa 17.1 Frequency Response

```
//Example 17.1
clc
s=%s;
syms Kc
tau=1;
taum=1;
wC=1;
g1=Kc;
g2=1/(s+1);
g3=1/(s+1);
G1=g2*g3;
G1=syslin('c',G1)
G=g1*g2/.g3;
disp(G,'C(s)/R(s)=')
//This equation can be written in the form of Kc*(s
   +1)/((1+Kc)*(tau2^2*s^2+2*tau2*zeta2*s+1)
tau2=sqrt(1/(1+Kc))
zeta2=sqrt(1/(1+Kc))
clf
```

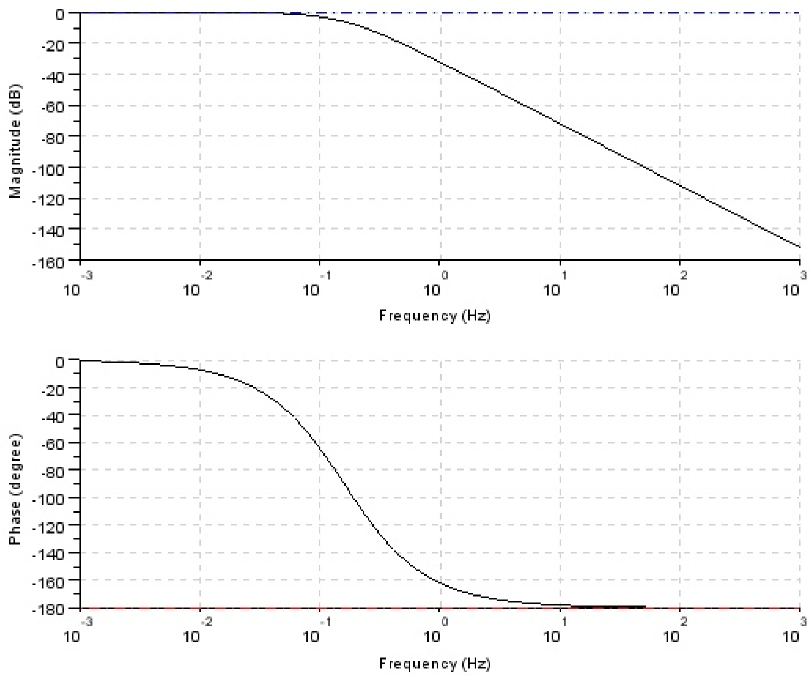

Figure 17.1: Frequency Response

```
bode(G1)
show_margins(G1)
//To make the open loop gain 1 at w=4
phaseangle=-152//degrees
phasemargin=180+phaseangle//degrees
//At this phase margin, the gain margin is
A=0.062//gain margin
Kc=1/A
zeta2=dbl(zeta2)
```

Scilab code Exa 17.3 Tuning Rules

```
//Example 17.3
clc;
syms Kc tauI s;
g1=Kc*(1+1/(tauI*s));
g2=1/(s+1);
g2=exp(-1.02*s)
G=g1*g2*g3//Openloop transfer function
//By solving the equation -180=-atan(w)-57.3*1.02*w, we get
wc0=2;//rad/min
disp('AR=Kcu/sqrt(1+wc0^2)')
AR=1;
Kcu=AR*sqrt(1+wc0^2);
//From Ziegler-Nicholas rules
Kc=Kcu*0.45//ultimate gain
Pu=2*%pi/wc0;//ultimate period
tauI=Pu/1.2;
disp('min',tauI,'tauI=')
```

Scilab code Exa 17.4 Tuning Rules

```
//Example 17.4
clc
s=%s;
syms Kc K1 tauI tauD
K=0.09;
Kc=K1/K;
Gc=K1*(1+1/(tauI*s)+tauD*s)
g1=1/((s+1)*(s+2));
//g2=exp(-0.5*s), we can write it as g2=(2-0.5*s)/(2+0.5*s). Therefore,
g2=(2-0.5*s)/(2+0.5*s);
G=g1*g2;
G=syslin('c',G)
clf
```

```
bode(G)
show_margins(G)
//From the bode diagrams we get
wc0=1.56;//rad/min
A=0.145;
Ku=1/A
Pu=2*%pi/wc0
//By Z-N rules
//For P controller
K1=0.5*Ku
Gc=K1
G1=Gc*G/K1
//For PI controller
K1=0.45*Ku
tauI=Pu/1.2
Gc=K1*(1+1/(tauI*s))
G2=Gc*G/K1
//For PID controller
K1=0.6*Ku
tauI=Pu/2
tauD=Pu/8
Gc=K1*(1+1/(tauI*s)+tauD*s)
G3=Gc*G/K1
clf
bode([G1;G2;G3])
legend(['G1';'G2';'G3']);
```

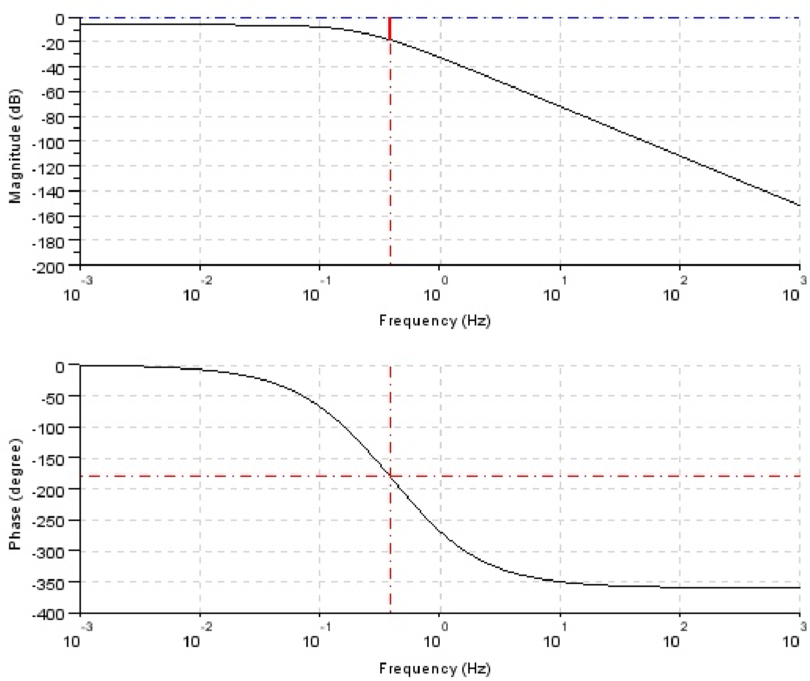

Figure 17.2: Tuning Rules

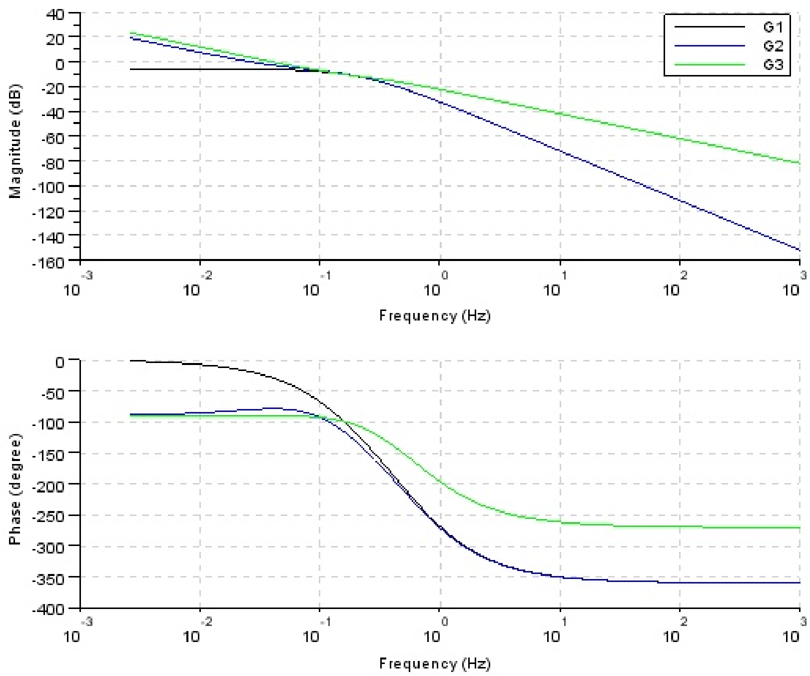

Figure 17.3: Tuning Rules

Chapter 18

Advanced Control Strategies

Scilab code Exa 18.3 Tuning Rules

```
//Example 18.3
clc
s=%s;
Kf=-1;
tp=2;
//Applying feedforward control rules
T1=1.5*tp
T2=0.7*tp
Gfs=Kf*(T1*s+1)/(T2*s+1);
disp(Gfs,'Gf(s)=')
```

Scilab code Exa 18.5 Internal Model Control

```
//Example 18.5
clc
syms K tau s l;
Gm=K/(tau*s+1);
//For this case
```

```
Gma=1;
Gmm=K/(tau*s+1);
Gm=Gma*Gmm;
GI=1/Gmm
f=1/(l*s+1);
//In order to be able to implement this transfer
   function let f(s)=1/(l*s+1)
//Thus IMC becomes
GI=f/Gmm
Gc=GI/(1-GI*Gm)
//On simplification , it will be in the form of
Gc=tau*(1+1/(tau*s))/(l*s*K)
printf("The result is in the form of PI controller")
```

Scilab code Exa 18.6 Internal Model Control

```
//Example
clc
syms K taud s tau t
G=K*exp(-taud*s)/(tau*s+1)
//we can use an approximation that
printf("exp(-taud*s)=(2-taud*s/2)/(2+taud*s)\n")
Gm=K*(2-taud*s/2)/((2+taud*s)*(tau*s+1));//here Gm=G
//For this model
Gma=(2-taud*s/2)/(2+taud*s);
Gmm=K/(tau*s+1);
Gm=Gma*Gmm;
GI=1/Gmm
f=1/(l*s+1);
//In order to be able to implement this transfer
   function let f(s)=1/(l*s+1)
//Thus IMC becomes
GI=f/Gmm
Gc=GI/(1-GI*Gm)
//This may be reduced algebraically to the form
```

```
   given by Eq.(18.21) with
printf("Kc=(2*tau+taud)/(2*l+taud)\n")
printf("tauI=tau+taud/2\n")
printf("tau*taud)/(2*tau+taud)\n")
printf("tau1=l*taud/2*(l+taud)\n")
```

Chapter 19

Controller Tuning And Process Identification

Scilab code Exa 19.1 Tuning Rules

```
//Example 19.1
clc
s=poly(0,'s');
syms tauI Kc
Gc=1+1/(tauI*s);
g1=1/(s+1);
//g2=exp(-s);
//we can write exp(-s) as (2-s)/(2+s).Therefore,
g2=(2-s)/(2+s);
G=g1*g2;
G=syslin('c',G)
Gp=Kc*Gc*G
Gs=Gp/(1+Gp)//Overall transfer function
//Ziegler Nicholas method
scf(1);
clf
bode(G)
show_margins(G)
//From bode diagrams we get
```

```
wc0=2.03
Kcu=2.26
Pu=2*%pi/wc0
//Since Gc is a PI controller, by Z-N rules
Kc=0.45*Kcu
tauI=Pu/1.2
//Cohen-Coon method
//Comaparing G with Eq.(19.6), we get
T=1;
Td=1;
Kp=1;
Kc=T*(0.9+Td/(12*T))/(Kp*Td)
tauI=Td*(30+3*Td/T)/(9+20*Td/T)
```

Scilab code Exa 19.2 Tuning Rules

```
//Example 19.2
clc
s=%s;
syms t Kc tauI;
Gc=Kc*(1+1/(tauI*s))
G=1/(s+1)^4;
G=syslin('c',G)
Gs=Gc*G/(1+Gc*G)//Overall transfer function
Us=1/s;
Cs=G*Us;
//Cohen-Coon method
Ct=ilaplace(Cs,s,t)
Ct1=diff(Ct,t)
Ct2=diff(Ct1,t)
disp('=0',Ct2)
//On solving the equation we get
t=linsolve(-1,3)
```

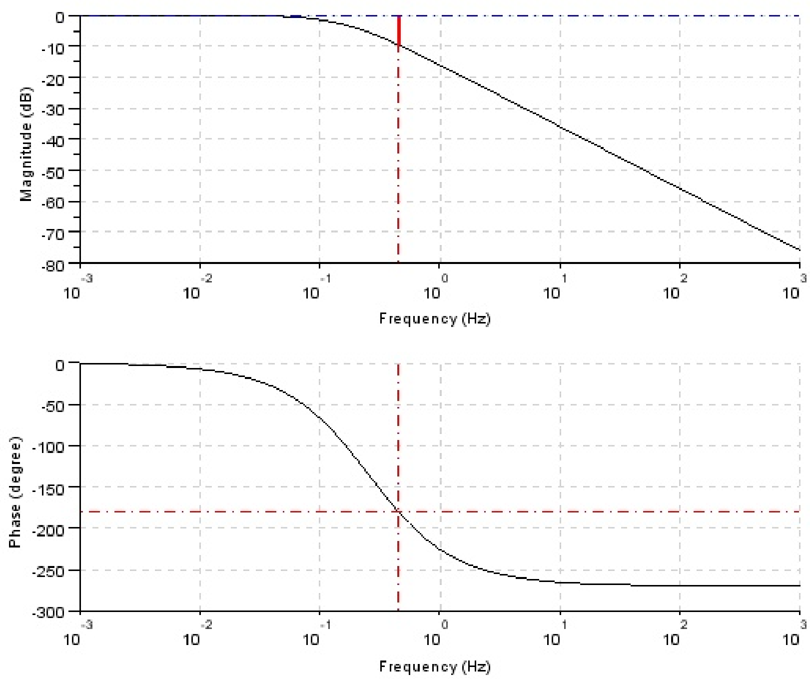

Figure 19.1: Tuning Rules

```
S=dbl(Ct1)
C3=dbl(Ct)
//From the figure 19.10 (B Vs t)
y2=0.353;
y1=0;
x2=3;
Td=3-(y2-y1)/S
Bu=1;//ultimate value of B
//From Eq.(19.4)
T=Bu/S
Kp=1;
//From Table 19.2
Kc=T*(0.9+Td/(12*T))/(Kp*Td)
tauI=Td*(30+3*Td/T)/(9+20*Td/T)
//By Z-N method
clf
bode(G)
show_margins(G)
//From Bode diagrams we get
Kcu=4;
Pu=2*%pi;
//Since Gc is a PI controller, by Z-N rules
Kc=0.45*Kcu
tauI=Pu/1.2
//By fitting the process reaction curve to a first
   order wit transport lag model by means of a least
    square fitting procedure. Applying the least
   square fit procedure out to t=5 produced the
   following results
Td=1.5;
T=3;
//By applying Cohen-Coon rules, we get
Kc=T*(0.9+Td/(12*T))/(Kp*Td)
tauI=Td*(30+3*Td/T)/(9+20*Td/T)
```

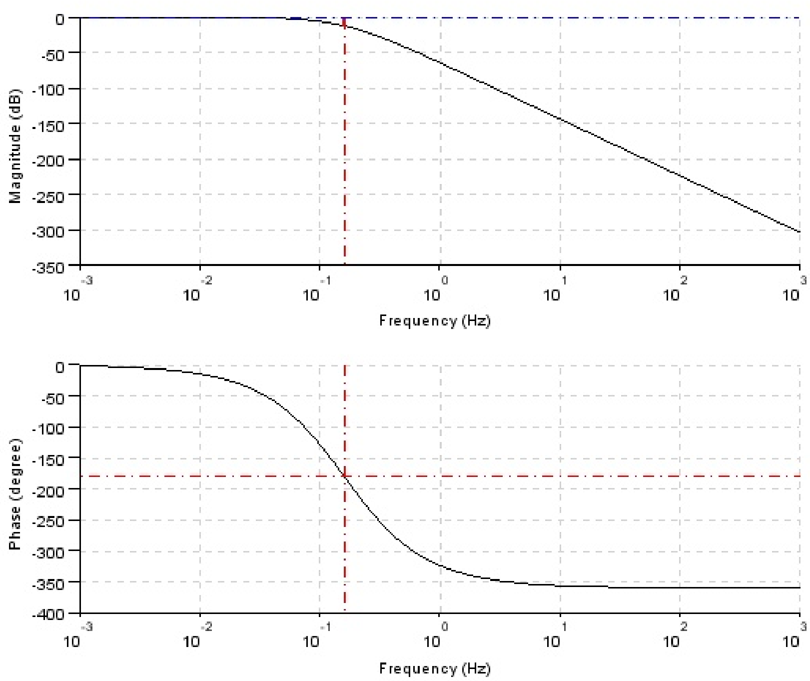

Figure 19.2: Tuning Rules

Chapter 20

Control Valves

Scilab code Exa 20.1 Control Valves

```
//Example 20.1
clc
Cv=4;
G=1.26;
P=100;//psi
q=Cv*sqrt(P/G);
disp('gpm',q,'q=')
```

Scilab code Exa 20.2 Control Valves

```
//Example 20.2
clc
L=100;//ft
D=1;//ft
D1=D/12;//inches
D2=D1*2.42;//centimetres
rho=62.4;//lb/ft^3
mu=1.5;//cp
```

```
Cv=4;
pv=100;//psi
G=1;
q=Cv*sqrt(pv/G);//maximum flow
disp('gpm',q,'q=')
printf("Let us start flow from q=30 gpm\n")
q=30;//gpm
q1=q/(60*7.48);//ft^3/sec
q2=q1*60*60;//ft^3/hr
Re=4*q2*rho/(%pi*mu*D2)//Reynolds number
//For this value of Reynolds number and for smooth
   pipe fanning friction factor is 0.005
f=0.005;//fanning friction factor
gc=32.2;
p=32*f*L*rho*q1^2/(144*%pi^2*gc*D1^5);//psi
P=pv-p
qmax=Cv*sqrt(P/G);
disp('gpm',qmax,'qmax=')
x=q/qmax//lift
```

Scilab code Exa 20.3 Control Valves

```
//Example 20.3
clc
L=200;//ft
D=1;//ft
D1=D/12;//inches
D2=D1*2.42;//centimetres
rho=62.4;//lb/ft^3
mu=1.5;//cp
pv=100;//psi
G=1;
q=30;//maximum flow
disp('gpm',30,'q=')
q1=q/(60*7.48);//ft^3/sec
```

```
q2=q1*60*60;//ft^3/hr
Re=4*q2*rho/(%pi*mu*D2)//Reynolds number
//For this value of Reynolds number and for smooth
    pipe fanning friction factor is 0.005
f=0.005;//fanning friction factor
gc=32.2;
p=32*f*L*rho*q1^2/(144*%pi^2*gc*D1^5);//psi
P=pv-p
Cv=q/sqrt(P/G)
//For q=20
q=20;//gpm
q1=q/(60*7.48);//ft^3/sec
p=32*f*L*rho*q1^2/(144*%pi^2*gc*D1^5);//psi
P=pv-p
qmax=Cv*sqrt(P/G);
disp('gpm',qmax,'qmax=')
x=q/qmax//lift
```

Chapter 22

Sampling And Z Transforms

Scilab code Exa 22.1 Z transforms

```
//Example 22.1
clc
disp("f(t)=u(t)=1")
disp("f(nT)=1")//for n>=0
syms z n
//From Eq.(22.8)
Z=symsum(z^(-n),n,0,%inf)
```

Scilab code Exa 22.2 Z transforms

```
//Example 22.2
clc
syms T tau z n
disp("f(t)=exp(-t/tau)")
ft=exp(-n*T/tau)*z^(-n);
Z=symsum(ft,n,0,%inf)
```

Chapter 24

Stability

Scilab code Exa 24.1 Stability

```
//Example 24.1
clc
syms K b z w;
Gz=K*(1-b)/(z-b)
//where b=exp(-T/tau)
//From Eq.(24.4)
z=w+1/w-1;
Gz=eval(Gz)
disp('=0',1+Gz,'1+G(z)=')
//which is equivalent to
disp('(K+1)*(1-b)*w+(1+b)-K(1-b)=0')
routh=[(K+1)*(1-b);(1+b)-K*(1-b)]
//b is always positive and less than one and K is positive
//The first element in the array is positive
//For stability, the Routh test requires that all elements of the first column be positive
//Therefore,
disp('>0',routh(2,1))
disp('K<(1+b)/(1-b)')
```

Chapter 26

Sampled Data Control Of A First Order Process With Transport Lag

Scilab code Exa 26.1.a Sampled data system

```
//Example 26.1(a)
clc
T=1;
tau=1.25;
b=exp(-T/tau)
//For quarter decay ratio
alpha=0.5
K=(alpha+b)/(1-b)
//Ultimate value of C is
Ci=K/(K+1);
disp(Ci,'C(inf)=')
Ri=1;
Offset=Ri-Ci
Period=2*T
```

Scilab code Exa 26.1.b Sampled data system

```
//Example 26.1(a)
clc
T=0.5;
tau=1.25;
b=exp(-T/tau)
//For quarter decay ratio
alpha=0.5
K=(alpha+b)/(1-b)
//Ultimate value of C is
Ci=K/(K+1);
disp(Ci,'C(inf)=')
Ri=1;
Offset=Ri-Ci
Period=2*T
```

Chapter 29

Transfer Function Matrix

Scilab code Exa 29.1 Transfer function matrix

```
//Example 29.1
clc
syms t tau
A=[-1 1;0 -2]
B=[0;1]
x0=[-1;0]
printf("x1=Ax+Bu(t)")
//On solving given equation
//let X=exp(A*t)
X=[exp(-t) exp(-t)-exp(-2*t);0 exp(-2*t)]
//Y=exp(A*(t-tau))
Y=[exp(-(t-tau) ) exp(-(t-tau))-exp(-2*(t-tau));0
   exp(-2*(t-tau))]
//From Eq.(29.4)
xt=X*x0+integ(Y*B,tau,0,t)
```

Scilab code Exa 29.2 Transfer function matrix

```
//Example 29.2
clc
A=[-2 0;4 -3]
B=[1 0;0 2]
syms s H1s H2s  U1s U2s
I=eye(2,2)
Gs=inv(s*I-A)*B
Hs=[H1s;H2s]
Us=[U1s;U2s]
Hs=Gs*Us
//On comparing
H1s=Hs(1,1)
H2s=Hs(2,1)
U2s=0;
U1s=1/s;
H1s=eval(H1s)
H2s=eval(H2s)
//On inverse laplace transformations
H1t=ilaplace(H1s,s,t)
H2s=ilaplace(H2s,s,t)
```

Chapter 30

Multivariable Control

Scilab code Exa 30.1 Multivariable control

```
//Example 30.1
clc
A1=1;
A2=1/2;
R1=1/2;
R2=2;
R3=1;
A=[-1/(R1*A1)-1/(R3*A1) 1/(A1*R1);1/(R1*A2) -1/(R2*
    A2)-1/(A2*R1)]
B=[1/A1 0;0 1/A2]
syms s M1 M2;
I=eye(2,2)
Gp=inv(s*I-A)*B
G11=Gp(1,1)
G12=Gp(1,2)
G21=Gp(2,1)
G22=Gp(2,2)
M=[M1;M2]
Cs=inv(s*I-A)*B*M
M1=1/s;
M2=0;
```

```
Cs=eval(Cs)
M1=0;
M2=1/s;
Cs=eval(Cs)
```

Scilab code Exa 30.2 Multivariable control

```
//Example 30.2
clc
syms s K1 K2
Gc11=K1;
Gc22=K2;
A1=1;
A2=1/2;
R2=2;
R3=1;
//In this problem ,Gv is a unit diagonal matrix i.e
    .,
Gv1=1;
Gv2=1;
A=[-1/(R1*A1)-1/(R3*A1) 1/(A1*R1);1/(R1*A2) -1/(R2*
    A2)-1/(A2*R1)]
B=[1/A1 0;0 1/A2]
I=eye(2,2)
Gp=inv(s*I-A)*B
G11=Gp(1,1)
G12=Gp(1,2)
G21=Gp(2,1)
G22=Gp(2,2)
Gc12=-G12*Gv2*Gc22/(G11*Gv1)
Gc21=-G21*Gv1*Gc11/(G22*Gv2)
Gv=[Gv1 0;0 Gv2]
Gc=[Gc11 Gc12;Gc21 Gc22]
Go=Gp*Gv*Gc;
Go=simple(Go)
```

Scilab code Exa 30.3 Multivariable control

```
//Example 30.3
clc
A1=1;
A2=1/2;
R1=1/2;
R2=2;
R3=1;
Gc11=K1;
Gc22=K2;
Gc12=0;
Gc21=0;
A=[-1/(R1*A1)-1/(R3*A1) 1/(A1*R1);1/(R1*A2) -1/(R2*
   A2)-1/(A2*R1)]
B=[1/A1 0;0 1/A2]
syms s;
I=eye(2,2)
Gp=inv(s*I-A)*B
G11=Gp(1,1)
G12=Gp(1,2)
G21=Gp(2,1)
G22=Gp(2,2)
Gv1=1;
Gv2=1;
Gm=I
Gv=[Gv1 0;0 Gv2]
Gc=[Gc11 Gc12;Gc21 Gc22]
Go=Gp*Gv*Gc;
Go=simple(Go)
//From Eq.(30.32)
P=det(I+Go*Gm)
disp('=0',simple(P))
```