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

July 31, 2019

[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 4

2 The Laplace Transform 6

3 Inversion by Partial Fractions 7

4 Further Properties of Transforms 11

5 Response of First Order Systems 13

6 Physical Examples of First Order Systems 16

7 Response of First Order Systems in Series 18

10 Controllers and Final Control Elements 19

12 Closed Loop Transfer functions 21

14 Stability 23

15 Root Locus 27

16 Introduction To Frequency Response 30

17 Control System Design By Frequeny Response 35

18 Advanced Control Strategies 41

19 Controller Tuning And Process Identification 44

20 Control Valves 49

22 Sampling And Z Transforms 52

24 Stability 53

26 Sampled Data Control Of A First Order Process With Transport Lag 54

29 Transfer Function Matrix 56

30 Multivariable Control 58

List of Scilab Codes

Exa 2.1	Laplace transform	6
Exa 2.3	Laplace transform	6
Exa 3.1	Inverse laplace transform	7
Exa 3.2	Inverse laplace transform	7
Exa 3.3	Inverse laplace transform	8
Exa 3.4	Inverse laplace transform	8
Exa 3.5	Inverse laplace transform	9
Exa 3.6	Inverse laplace transform	9
Exa 4.1	Final value theorem	11
Exa 4.2	Final value theorem	11
Exa 4.4	Laplace transform	12
Exa 5.1	First order systems	13
Exa 5.2	First order systems	13
Exa 6.1	First order systems	16
Exa 7.1	First order systems	18
Exa 10.1	Control system	19
Exa 12.1	Transfer functions	21
Exa 12.2	Transfer functions	21
Exa 14.1	Stability	23
Exa 14.2	Stability	24
Exa 14.3	Stability	24
Exa 14.4	Stability	25
Exa 15.1	Root locus	27
Exa 15.2	Root locus	27
Exa 16.1	Frequency Response	30
Exa 16.2	Frequency Response	30
Exa 16.4	Bode diagram	31
Exa 16.5	Bode diagram	33

Exa 17.1	Frequency Response	35
Exa 17.3	Tuning Rules	36
Exa 17.4	Tuning Rules	37
Exa 18.3	Tuning Rules	41
Exa 18.5	Internal Model Control	41
Exa 18.6	Internal Model Control	42
Exa 19.1	Tuning Rules	44
Exa 19.2	Tuning Rules	45
Exa 20.1	Control Valves	49
Exa 20.2	Control Valves	49
Exa 20.3	Control Valves	50
Exa 22.1	Z transforms	52
Exa 22.2	Z transforms	52
Exa 24.1	Stability	53
Exa 26.1.a	Sampled data system	54
Exa 26.1.b	Sampled data system	55
Exa 29.1	Transfer function matrix	56
Exa 29.2	Transfer function matrix	56
Exa 30.1	Multivariable control	58
Exa 30.2	Multivariable control	59
Exa 30.3	Multivariable control	60

List of Figures

5.1 First order systems 15

15.1 Root locus 28
15.2 Root locus 29

16.1 Bode diagram 32
16.2 Bode diagram 34

17.1 Frequency Response 36
17.2 Tuning Rules 39
17.3 Tuning Rules 40

19.1 Tuning Rules 46
19.2 Tuning Rules 48

Chapter 2

The Laplace Transform

Scilab code Exa 2.1 Laplace transform

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

Scilab code Exa 2.3 Laplace transform

```
1 //Example 2.3
2 clc
3 s=%s;
4 xs=2/(s+3);
5 disp(xs,'x(s)=')
6 syms t;
7 xt=ilaplace(xs,s,t);
8 disp(xt,'x(t)=')
```

Chapter 3

Inversion by Partial Fractions

Scilab code Exa 3.1 Inverse laplace transform

```
1 //Example 3.1
2 clc
3 s=%s;
4 xs=1/(s*(s+1));
5 disp(xs,'x(s)=')
6 syms t;
7 [A]=pfss(xs)
8 F1=ilaplace(A(1),s,t);
9 F2=ilaplace(A(2),s,t);
10 xt=F1+F2;
11 disp(xt,'x(t)=')
```

Scilab code Exa 3.2 Inverse laplace transform

```
1 //Example 3.2
2 clc
3 s=%s;
4 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;
```

```
5  xs=2/((s^2+4)*(s+1));
6  disp(xs,'x(s)=')
7  [A]=pfss(xs)
8  F1=ilaplace(A(1),s,t);
9  F2=ilaplace(A(2),s,t);
10 xt=F1+F2;
11 disp(xt,'x(t)=')
```

Scilab code Exa 3.5 Inverse laplace transform

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

Scilab code Exa 3.6 Inverse laplace transform

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

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

Chapter 4

Further Properties of Transforms

Scilab code Exa 4.1 Final value theorem

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

Scilab code Exa 4.2 Final value theorem

```
1 //Example 4.2
2 clc
3 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

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

Scilab code Exa 5.2 First order systems

```
1 //Example 5.2
2 clear
3 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
```

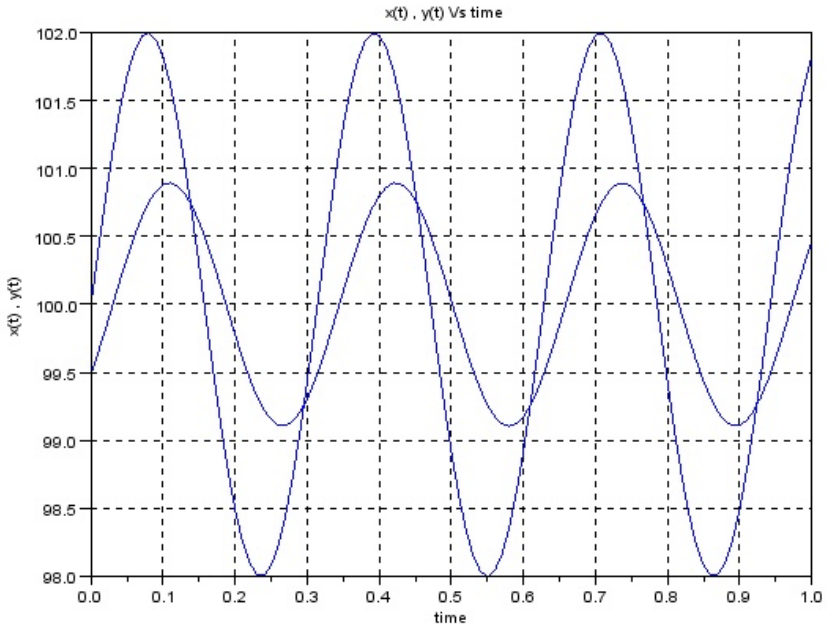

Figure 5.1: First order systems

Chapter 6

Physical Examples of First Order Systems

Scilab code Exa 6.1 First order systems

```
1 //Example 6.1
2 clc;
3 syms s t;
4 tau=1;//min
5 R=1/9;//ft/cfm
6 A=9;
7 //from Equation 6.8
8 g=R/(tau*s+1);
9 disp(g,'H(s)/Q(s)=')
10 //from Example 4.5
11 disp('Q(t)=90[u(t)-u(t-0.1)')
12 //where u(t) is a unit step function, the laplace
     transform of it gives
13 Qs=90*(1-exp(-0.1*s))/s
14 disp(Qs,'Q(s)=')
15 Hs=Qs*g;
16 disp(Hs,'H(s)=')
17 //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

```
1  //Example 7.1
2  clc
3  s=%s;
4  tau1=0.5;
5  tau2=1;
6  R2=1;
7  //From Eq.(7.8)
8  g=R2/((tau1*s+1)*(tau2*s+1))
9  disp(g,'H2(s)/Q(s)=')
10 Qs=1/s;
11 H2s=g*Qs;
12 disp(H2s,'H2(s)=')
13 syms t;
14 H2t=ilaplace(H2s,s,t);
15 disp(H2t,'H2(t)=')
```

Chapter 10

Controllers and Final Control Elements

Scilab code Exa 10.1 Control system

```
1  //Example 10.1
2  clear
3  clc
4  t1=60;//Fahrenheit
5  t2=100;//Fahrenheit
6  p1=3;//psi
7  p2=15;//psi
8  T1=71;//Fahrenheit
9  T2=75;//Fahrenheit
10 pb=((T2-T1)/(t2-t1))*100;
11 disp('%',pb,'proportional band=')
12 Gain=(p2-p1)/(T2-T1);
13 disp('psi/F',Gain,'Gain=')
14 //Assume pb is changed to 75% then
15 pb=75;//%
16 T=(pb*(t2-t1))/100;
17 disp('Fahrenheit',T,'T=')
18 Gain=(p2-p1)/T;
19 disp('psi/F',Gain,'Gain=')
```

Chapter 12

Closed Loop Transfer functions

Scilab code Exa 12.1 Transfer functions

```
1 //Example 12.1
2 clc
3 syms Gc G1 G2 G3 H1 H2 R U1;
4 G=Gc*G1*G2*G3*H1*H2;
5 g=Gc*G1*G2*G3/(1+G);
6 disp(g,'C/R=')
7 g1=G2*G3/(1+G);
8 disp(g1,'C/U1=')
9 g2=G3*H1*H2/(1+G);
10 disp(g2,'B/U2=')
11 C1=g*R;
12 C2=g1*U1;
13 disp(C1+C2,'C=')
```

Scilab code Exa 12.2 Transfer functions

```
1 //Example 12.2
2 clc
```

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

Chapter 14

Stability

Scilab code Exa 14.1 Stability

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

```
21 if (real(r(i,1))<0)
22 c=c+1;
23 end
24 end
25 if(c>=1)
26 printf("system is stable\n")
27 else ("system is unstable")
28 end
```

Scilab code Exa 14.2 Stability

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

Scilab code Exa 14.3 Stability

```
1  //Example 14.3
2  clc
3  syms Kc s s3;
4  G1=1/((s+1)*(0.5*s+1));
5  H=3/(s+3);
6  G=Kc*G1*H;
7  G=simple(G);
8  //The characteristic equation is therefore
9  disp('1+G=0')
10 disp('=0',1+G,'1+G=');
11 //which is equivalent to
12 disp("s^3+6*s^2+11*s+6+6*Kc=0")
13 routh=[1 11;6 6+6*Kc]
14 routh=[routh;-det(routh(1:2,1:2))/routh(2,1),0]
15 routh=[routh;-det(routh(2:3,1:2))/routh(3,1),0]
16 routh=simple(routh)
17 disp('>0',routh(3,1))
18 disp('Kc<10')
19 Kc=10;
20 routh=horner(routh,Kc);
21 routh=dbl(routh)
22 C=routh(2,1);
23 D=routh(2,2);
24 p=poly([D 0 C],'s','coeff')
25 disp('6*s^2+66=0')
26 r=roots(p)
27 disp('=0',simple((s-r(1,1))*(s-r(2,1))*(s-s3)))
28 //On comparing with the equation
29 poly([6+6*Kc 11 6 1],'s','coeff')
30 //we get
31 s3=-6;
32 printf("s1=3.3166248*i,s2=3.3166248*i,,s3=6\n")
```

Scilab code Exa 14.4 Stability

```
1  //Example 14.4
2  clc
3  s=%s;
4  tau1=1;
5  tau2=1/2;
6  tau3=1/3;
7  taui=0.25;
8  Kc=5;
9  n=Kc/(tau1*tau2*tau3)*(taui*s+1);
10 d=taui*s*(s+(1/tau1))*(s+(1/tau2))*(s+(1/tau3));
11 G=syslin('c',n/d);
12 //The characteristic equation is therefore
13 disp('1+G=0')
14 disp('=0',1+G,'1+G=');
15 //which is equivalent to
16 disp("s^4+6*s^3+11*s^2+36*s+120=0")
17 h=poly([120 36 11 6 1],'s','coeff')
18 r=routh_t(h)
19 y=coeff(h);
20 n=length(y);
21 c=0;
22 for i=1:n
23 if (r(i,1)<0)
24 c=c+1;
25 end
26 end
27 if(c>=1)
28 printf("system is unstable\n")
29 else ("system is stable")
30 end
```

Chapter 15

Root Locus

Scilab code Exa 15.1 Root locus

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

Scilab code Exa 15.2 Root locus

```
1  //Example 15.2
2  clc
```

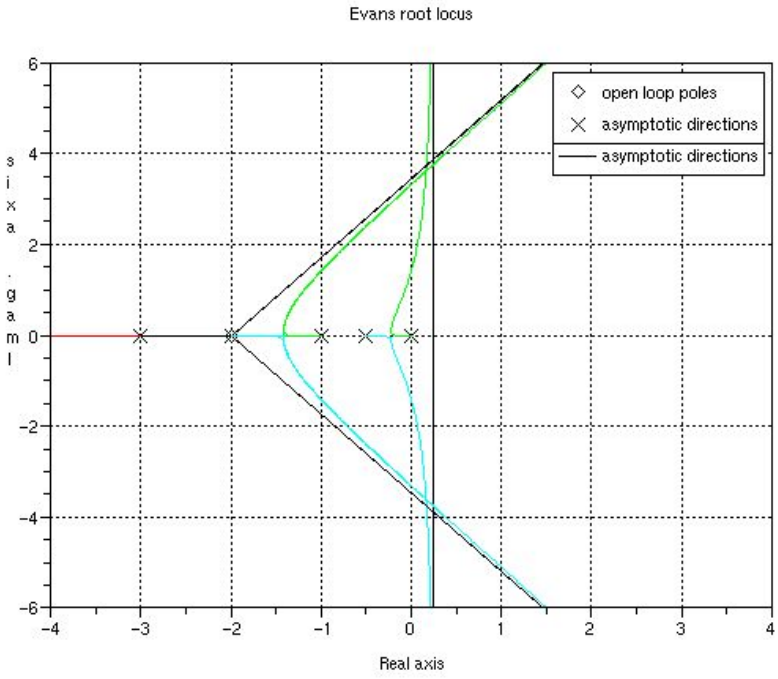

Figure 15.1: Root locus

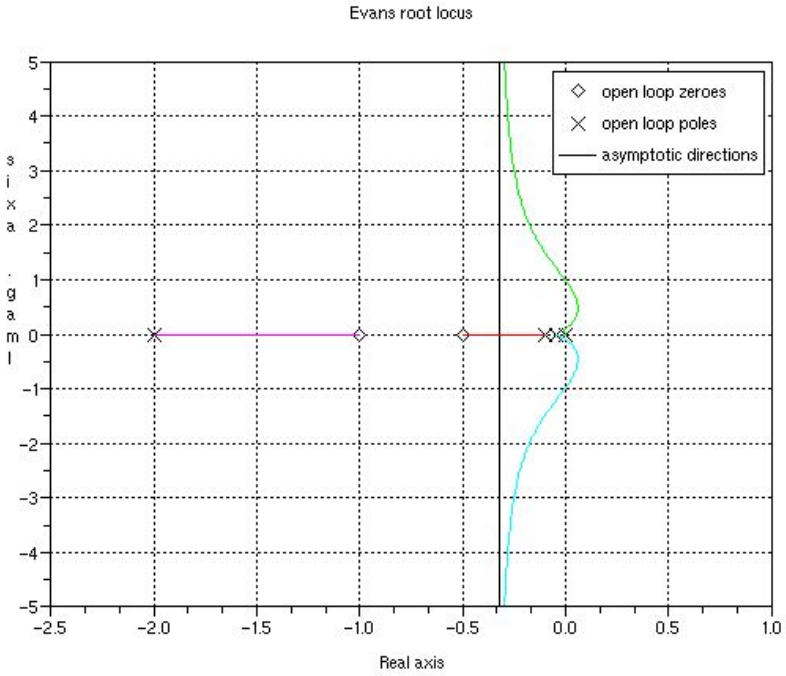

Figure 15.2: Root locus

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

Chapter 16

Introduction To Frequency Response

Scilab code Exa 16.1 Frequency Response

```
1  //Example 16.1
2  clc
3  s=%s;
4  j=%i;
5  f=10/%pi;
6  w=2*%pi*f;
7  G=1/(0.1*s+1);
8  s=w*j;
9  Gs=horner(G,s);
10 disp(Gs,'G(20j)=')
11 [r,theta]=polar(Gs)
12 theta=theta*180/%pi;
13 disp('degrees',theta,'theta=')
```

Scilab code Exa 16.2 Frequency Response

```
1  //Example 16.2
2  clc
3  syms tau s zeta w;
4  j=%i;
5  n=1;
6  d=tau^2*s^2+2*zeta*tau*s+1;
7  G=n/d
8  s=j*w;
9  G=1/(2*s*tau*zeta+s^2*tau^2+1)
10 [num den]=numden(G)
11 d=abs(den)
12 cof_a_0=coeffs(den,'%i',0)
13 cof_a_1=coeffs(den,'%i',1)
14 AR=1/d
15 theta=AR*atan(-cof_a_1/cof_a_0);
16 disp(theta,'Phase angle=')
```

Scilab code Exa 16.4 Bode diagram

```
1  //Example 16.4
2  clc
3  s=%s;
4  H=1/(s+1);
5  Hs=syslin('c',H)
6  J=1/(s+5);
7  Js=syslin('c',J)
8  G=Hs*Js;
9  Gs=syslin('c',G)
10 clf
11 bode([Hs;Js;Gs;])
12 legend(['1/(s+1)';'1/(s/5+1)';'1/(5*(s+1)*(s/5+1))'
   ])
```

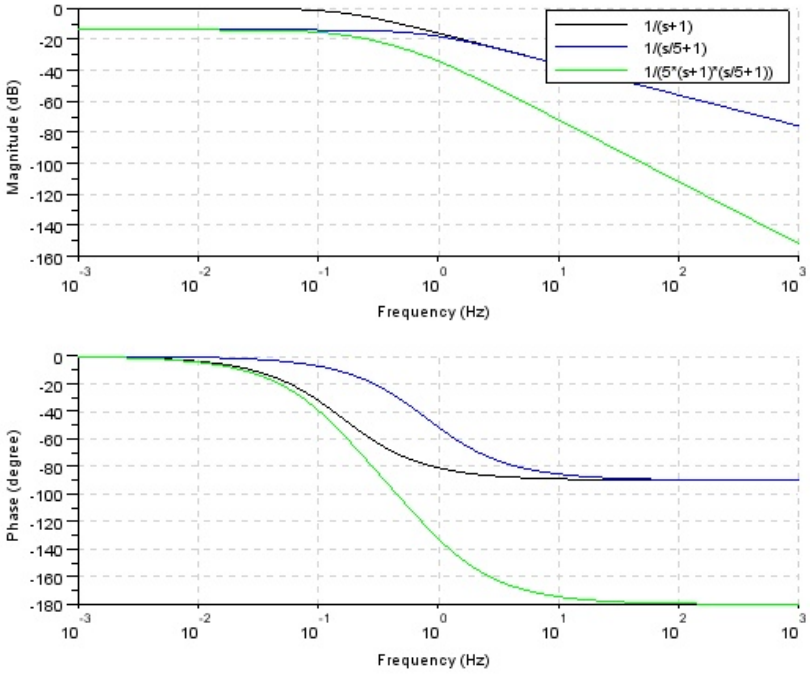

Figure 16.1: Bode diagram

Scilab code Exa 16.5 Bode diagram

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

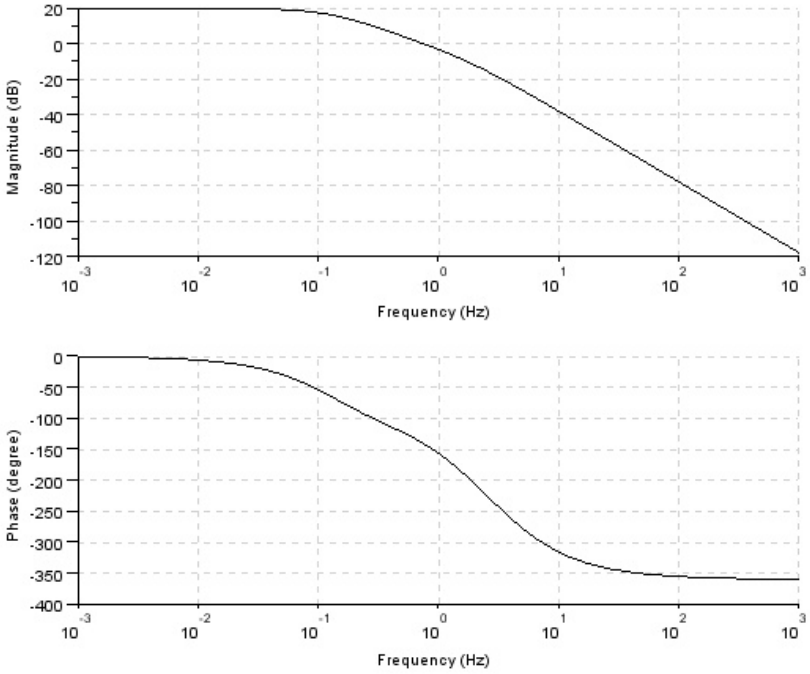

Figure 16.2: Bode diagram

Chapter 17

Control System Design By Frequeny Response

Scilab code Exa 17.1 Frequency Response

```
1  //Example 17.1
2  clc
3  s=%s;
4  syms Kc
5  tau=1;
6  taum=1;
7  wC=1;
8  g1=Kc;
9  g2=1/(s+1);
10 g3=1/(s+1);
11 G1=g2*g3;
12 G1=syslin('c',G1)
13 G=g1*g2/.g3;
14 disp(G,'C(s)/R(s)=')
15 //This equation can be written in the form of Kc*(s
     +1)/((1+Kc)*(tau2^2*s^2+2*tau2*zeta2*s+1)
16 tau2=sqrt(1/(1+Kc))
17 zeta2=sqrt(1/(1+Kc))
18 clf
```

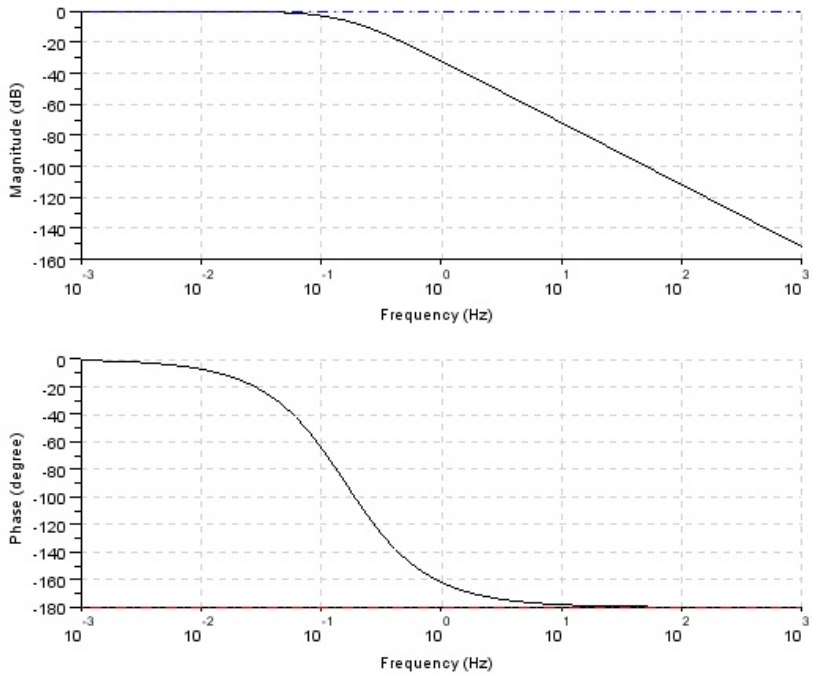

Figure 17.1: Frequency Response

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

Scilab code Exa 17.3 Tuning Rules

```
1  //Example 17.3
2  clc;
3  syms Kc tauI s;
4  g1=Kc*(1+1/(tauI*s));
5  g2=1/(s+1);
6  g2=exp(-1.02*s)
7  G=g1*g2*g3//Openloop transfer function
8  //By solving the equation -180=-atan(w)-57.3*1.02*w,
       we get
9  wc0=2;//rad/min
10 disp('AR=Kcu/sqrt(1+wc0^2)')
11 AR=1;
12 Kcu=AR*sqrt(1+wc0^2);
13 //From Ziegler-Nicholas rules
14 Kc=Kcu*0.45//ultimate gain
15 Pu=2*%pi/wc0;//ultimate period
16 tauI=Pu/1.2;
17 disp('min',tauI,'tauI=')
```

Scilab code Exa 17.4 Tuning Rules

```
1  //Example 17.4
2  clc
3  s=%s;
4  syms Kc K1 tauI tauD
5  K=0.09;
6  Kc=K1/K;
7  Gc=K1*(1+1/(tauI*s)+tauD*s)
8  g1=1/((s+1)*(s+2));
9  //g2=exp(-0.5*s), we can write it as g2=(2-0.5*s)
       /(2+0.5*s). Therefore,
10 g2=(2-0.5*s)/(2+0.5*s);
11 G=g1*g2;
12 G=syslin('c',G)
13 clf
```

```
14 bode(G)
15 show_margins(G)
16 //From the bode diagrams we get
17 wc0=1.56;//rad/min
18 A=0.145;
19 Ku=1/A
20 Pu=2*%pi/wc0
21 //By Z-N rules
22 //For P controller
23 K1=0.5*Ku
24 Gc=K1
25 G1=Gc*G/K1
26 //For PI controller
27 K1=0.45*Ku
28 tauI=Pu/1.2
29 Gc=K1*(1+1/(tauI*s))
30 G2=Gc*G/K1
31 //For PID controller
32 K1=0.6*Ku
33 tauI=Pu/2
34 tauD=Pu/8
35 Gc=K1*(1+1/(tauI*s)+tauD*s)
36 G3=Gc*G/K1
37 clf
38 bode([G1;G2;G3])
39 legend(['G1';'G2';'G3']);
```

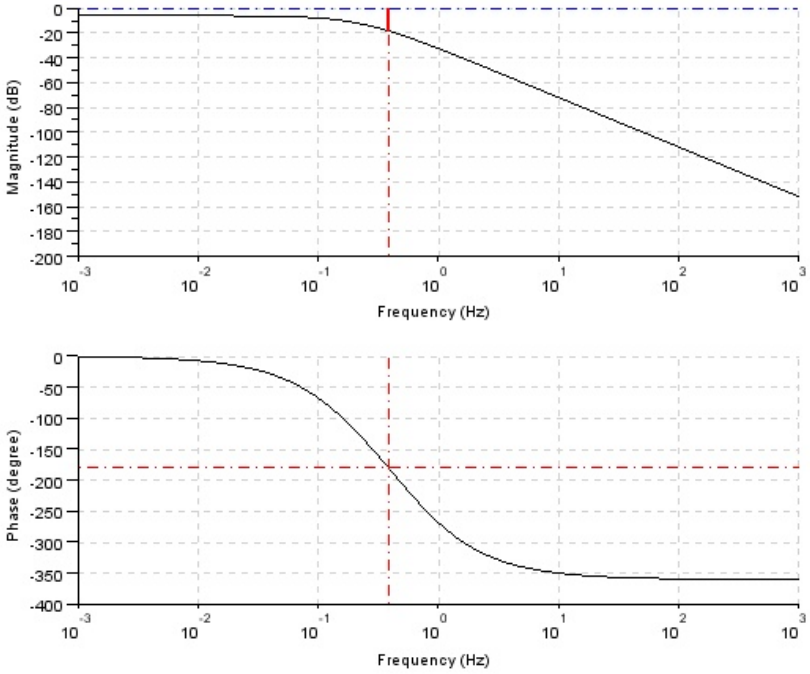

Figure 17.2: Tuning Rules

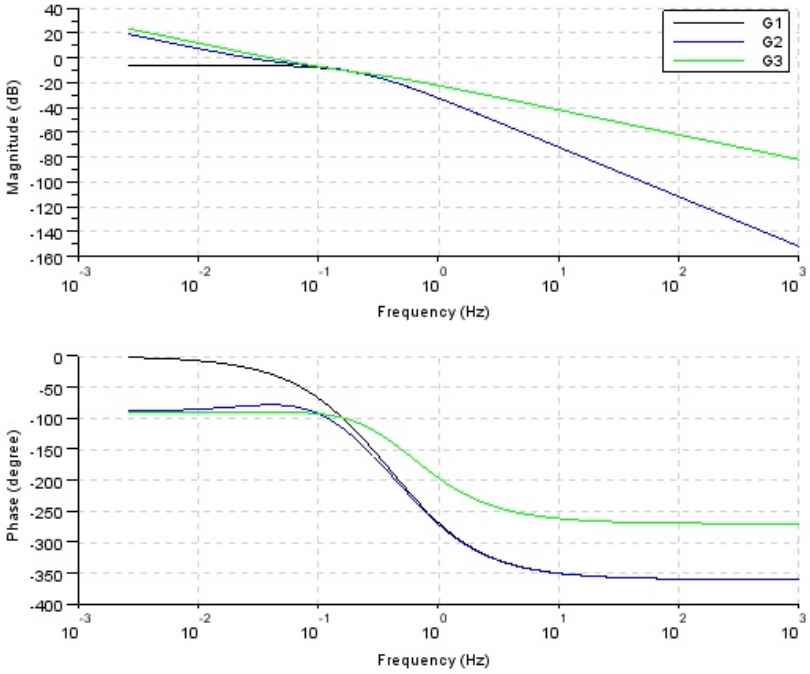

Figure 17.3: Tuning Rules

Chapter 18

Advanced Control Strategies

Scilab code Exa 18.3 Tuning Rules

```
1 //Example 18.3
2 clc
3 s=%s;
4 Kf=-1;
5 tp=2;
6 //Applying feedforward control rules
7 T1=1.5*tp
8 T2=0.7*tp
9 Gfs=Kf*(T1*s+1)/(T2*s+1);
10 disp(Gfs,'Gf(s)=')
```

Scilab code Exa 18.5 Internal Model Control

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

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

Scilab code Exa 18.6 Internal Model Control

```
1  //Example
2  clc
3  syms K taud s tau t
4  G=K*exp(-taud*s)/(tau*s+1)
5  //we can use an approximation that
6  printf("exp(-taud*s)=(2-taud*s/2)/(2+taud*s)\n")
7  Gm=K*(2-taud*s/2)/((2+taud*s)*(tau*s+1));//here Gm=G
8  //For this model
9  Gma=(2-taud*s/2)/(2+taud*s);
10 Gmm=K/(tau*s+1);
11 Gm=Gma*Gmm;
12 GI=1/Gmm
13 f=1/(l*s+1);
14 //In order to be able to implement this transfer
      function let f(s)=1/(l*s+1)
15 //Thus IMC becomes
16 GI=f/Gmm
17 Gc=GI/(1-GI*Gm)
18 //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

```
1  //Example 19.1
2  clc
3  s=poly(0,'s');
4  syms tauI Kc
5  Gc=1+1/(tauI*s);
6  g1=1/(s+1);
7  //g2=exp(-s);
8  //we can write exp(-s) as (2-s)/(2+s).Therefore,
9  g2=(2-s)/(2+s);
10 G=g1*g2;
11 G=syslin('c',G)
12 Gp=Kc*Gc*G
13 Gs=Gp/(1+Gp)//Overall transfer function
14 //Ziegler Nicholas method
15 scf(1);
16 clf
17 bode(G)
18 show_margins(G)
19 //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)
```

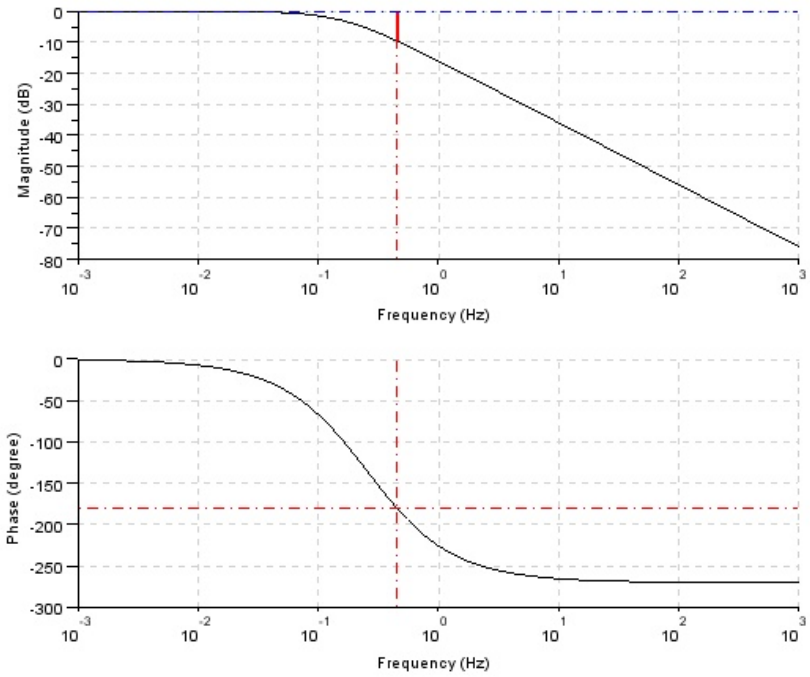

Figure 19.1: Tuning Rules

```
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)
```

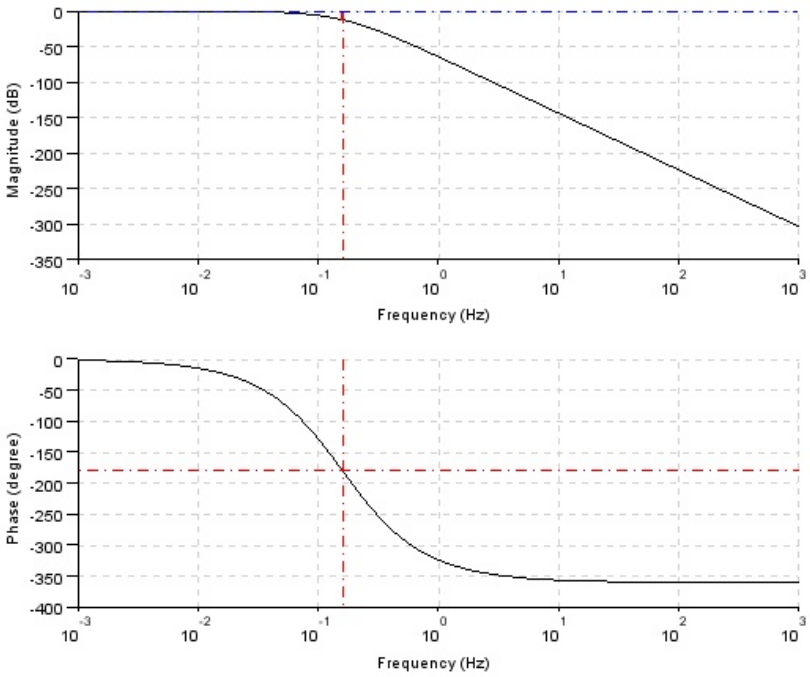

Figure 19.2: Tuning Rules

Chapter 20

Control Valves

Scilab code Exa 20.1 Control Valves

```
1 //Example 20.1
2 clc
3 Cv=4;
4 G=1.26;
5 P=100;//psi
6 q=Cv*sqrt(P/G);
7 disp('gpm',q,'q=')
```

Scilab code Exa 20.2 Control Valves

```
1 //Example 20.2
2 clc
3 L=100;//ft
4 D=1;//ft
5 D1=D/12;//inches
6 D2=D1*2.42;//centimetres
7 rho=62.4;//lb/ft^3
8 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
```

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

Chapter 22

Sampling And Z Transforms

Scilab code Exa 22.1 Z transforms

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

Scilab code Exa 22.2 Z transforms

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

Chapter 24

Stability

Scilab code Exa 24.1 Stability

```
1 //Example 24.1
2 clc
3 syms K b z w;
4 Gz=K*(1-b)/(z-b)
5 //where b=exp(-T/tau)
6 //From Eq.(24.4)
7 z=w+1/w-1;
8 Gz=eval(Gz)
9 disp('=0',1+Gz,'1+G(z)=')
10 //which is equivalent to
11 disp('(K+1)*(1-b)*w+(1+b)-K(1-b)=0')
12 routh=[(K+1)*(1-b);(1+b)-K*(1-b)]
13 //b is always positive and less than one and K is
      positive
14 //The first element in the array is positive
15 //For stability, the Routh test requires that all
      elements of the first column be positive
16 //Therefore,
17 disp('>0',routh(2,1))
18 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

```
1  //Example 26.1(a)
2  clc
3  T=1;
4  tau=1.25;
5  b=exp(-T/tau)
6  //For quarter decay ratio
7  alpha=0.5
8  K=(alpha+b)/(1-b)
9  //Ultimate value of C is
10 Ci=K/(K+1);
11 disp(Ci,'C(inf)=')
12 Ri=1;
13 Offset=Ri-Ci
14 Period=2*T
```

Scilab code Exa 26.1.b Sampled data system

```
1 //Example 26.1(a)
2 clc
3 T=0.5;
4 tau=1.25;
5 b=exp(-T/tau)
6 //For quarter decay ratio
7 alpha=0.5
8 K=(alpha+b)/(1-b)
9 //Ultimate value of C is
10 Ci=K/(K+1);
11 disp(Ci,'C(inf)=')
12 Ri=1;
13 Offset=Ri-Ci
14 Period=2*T
```

Chapter 29

Transfer Function Matrix

Scilab code Exa 29.1 Transfer function matrix

```
1  //Example 29.1
2  clc
3  syms t tau
4  A=[-1 1;0 -2]
5  B=[0;1]
6  x0=[-1;0]
7  printf("x1=Ax+Bu(t)")
8  //On solving given equation
9  //let X=exp(A*t)
10 X=[exp(-t) exp(-t)-exp(-2*t);0 exp(-2*t)]
11 //Y=exp(A*(t-tau))
12 Y=[exp(-(t-tau) ) exp(-(t-tau))-exp(-2*(t-tau));0
      exp(-2*(t-tau))]
13 //From Eq.(29.4)
14 xt=X*x0+integ(Y*B,tau,0,t)
```

Scilab code Exa 29.2 Transfer function matrix

```
1  //Example 29.2
2  clc
3  A=[-2 0;4 -3]
4  B=[1 0;0 2]
5  syms s H1s H2s  U1s U2s
6  I=eye(2,2)
7  Gs=inv(s*I-A)*B
8  Hs=[H1s;H2s]
9  Us=[U1s;U2s]
10 Hs=Gs*Us
11 //On comparing
12 H1s=Hs(1,1)
13 H2s=Hs(2,1)
14 U2s=0;
15 U1s=1/s;
16 H1s=eval(H1s)
17 H2s=eval(H2s)
18 //On inverse laplace transformations
19 H1t=ilaplace(H1s,s,t)
20 H2s=ilaplace(H2s,s,t)
```

Chapter 30

Multivariable Control

Scilab code Exa 30.1 Multivariable control

```
1  //Example 30.1
2  clc
3  A1=1;
4  A2=1/2;
5  R1=1/2;
6  R2=2;
7  R3=1;
8  A=[-1/(R1*A1)-1/(R3*A1) 1/(A1*R1);1/(R1*A2) -1/(R2*
       A2)-1/(A2*R1)]
9  B=[1/A1 0;0 1/A2]
10 syms s M1 M2;
11 I=eye(2,2)
12 Gp=inv(s*I-A)*B
13 G11=Gp(1,1)
14 G12=Gp(1,2)
15 G21=Gp(2,1)
16 G22=Gp(2,2)
17 M=[M1;M2]
18 Cs=inv(s*I-A)*B*M
19 M1=1/s;
20 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

```
1  //Example 30.3
2  clc
3  A1=1;
4  A2=1/2;
5  R1=1/2;
6  R2=2;
7  R3=1;
8  Gc11=K1;
9  Gc22=K2;
10 Gc12=0;
11 Gc21=0;
12 A=[-1/(R1*A1)-1/(R3*A1) 1/(A1*R1);1/(R1*A2) -1/(R2*
       A2)-1/(A2*R1)]
13 B=[1/A1 0;0 1/A2]
14 syms s;
15 I=eye(2,2)
16 Gp=inv(s*I-A)*B
17 G11=Gp(1,1)
18 G12=Gp(1,2)
19 G21=Gp(2,1)
20 G22=Gp(2,2)
21 Gv1=1;
22 Gv2=1;
23 Gm=I
24 Gv=[Gv1 0;0 Gv2]
25 Gc=[Gc11 Gc12;Gc21 Gc22]
26 Go=Gp*Gv*Gc;
27 Go=simple(Go)
28 //From Eq.(30.32)
29 P=det(I+Go*Gm)
30 disp('=0',simple(P))
```