

Scilab Manual for
Control Systems & Electrical Machines
by Dr Lochan Jolly
Electronics Engineering
Thakur College of Engineering & Technology¹

Solutions provided by
Dr Lochan Jolly
Electronics Engineering
Mumbai/TCET

April 26, 2024

¹Funded by a grant from the National Mission on Education through ICT, <http://spoken-tutorial.org/NMEICT-Intro>. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website <http://scilab.in>

Contents

List of Scilab Solutions	3
1 To plot pole zero plot of given system and comment on stability	5
2 To study transient response for given system	7
3 To sketch root locus, bode plot and nyquist plot for given system	10
4 To compare open loop and closed loop system.	12
5 To use block reduction technique to get the equivalent system	14
6 To implement state model reduction technique for the given system	16

List of Experiments

Solution 1.01	polezeroplot	5
Solution 2.02	transientresponse	7
Solution 3.03	stabilityanalysis	10
Solution 4.04	systemcomparison	12
Solution 5.05	blockreduction	14
Solution 6.06	statemodel	16

List of Figures

1.1	polezeroplot	6
2.1	transientresponse	9
3.1	stabilityanalysis	11
4.1	systemcomparison	13
5.1	blockreduction	15
6.1	statemodel	17

Experiment: 1

To plot pole zero plot of given system and comment on stability

Scilab code Solution 1.01 polezeroplot

```
1 //exp1 find pole zero plot of given transfer
  functions
2 //OS=Windows XP sp3
3 //Scilab version 5.4.0
4 //sample values
5 s=poly(0, 's');
6 n=[2+3*s+4*s^2]; //input numerator of transfer
  function
7 d=[1+s+s^2]; //input denominator of transfer function
8 h=syslin('c',n/d);
9 plzr(h);
```

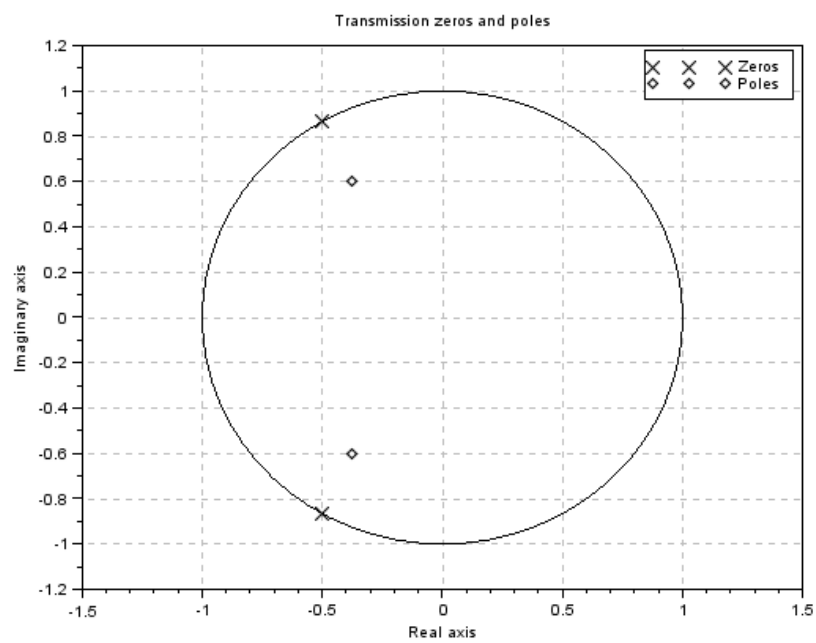


Figure 1.1: polezeroplot

Experiment: 2

To study transient response for given system

Scilab code Solution 2.02 transientresponse

```
1 //          2 To study transient response for given
   system1
2 //OS=Windows XP sp3
3 //Scilab version 5.4.0
4 //sample values
5 s=%s
6 //('Enter the value of damping z for under damped
   system ')
7 z=0.9
8 num=1;den =36+2*z*s+s^2;//input numerator and
   denominator of transfer function
9 TF = syslin('c',num,den)//transfer function
10 //disp(TF)
11 subplot(131)
12 t=0:0.1:10;
13 y1 = csim('step', t, TF);//time response
14 title('Underdamped system');
15 plot(t, y1)
16 //('Enter the value of damping z for critically
```



```

        damped system ')
17 z=6
18 num=1;den =36+2*z*s+s^2;
19 TF = syslin('c',num,den)
20 subplot(132)
21 t=0:0.1:10;
22 y1 = csim('step', t, TF);
23 title('Critically damped system');
24 plot(t, y1)
25 //('Enter the value of damping z for over damped
    system ')
26 z=20
27 num=1;den =36+2*z*s+s^2;
28 TF = syslin('c',num,den)
29 subplot(133)
30 t=0:0.1:10;
31 y1 = csim('step', t, TF);
32 title('Overdamped system');
33 plot(t, y1)

```

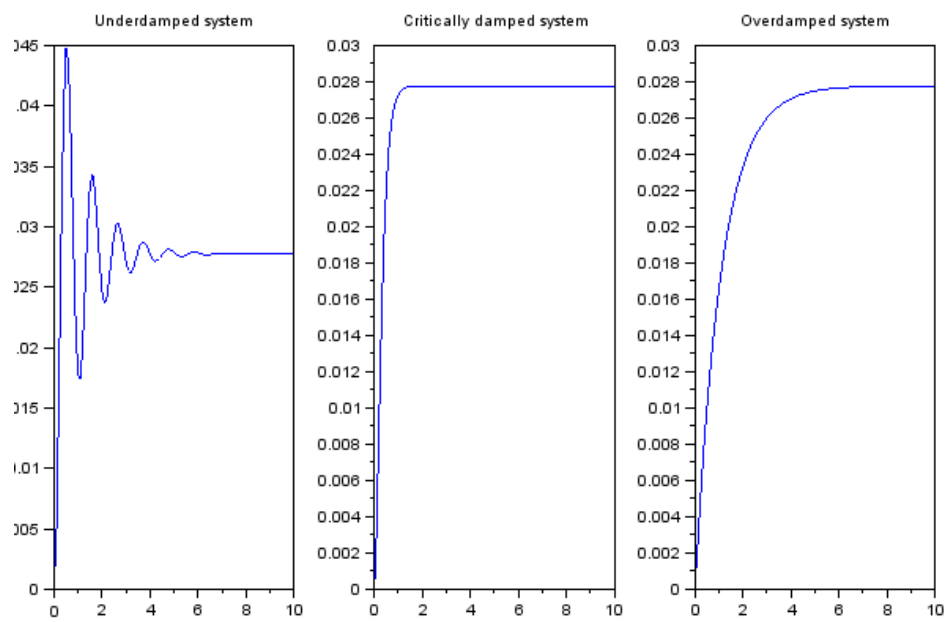


Figure 2.1: transientresponse

Experiment: 3

To sketch root locus, bode plot and nyquist plot for given system

Scilab code Solution 3.03 stabilityanalysis

```
1 //3 To sketch root locus , bode plot and nyquist
  plot for given system
2 //OS=Windows XP sp3
3 //Scilab version 5.4.0
4 //sample values
5 s=%s ;
6 T=syslin('c',25+30*s+5*s^2,168+206*s+89*s^2+16*s^3+s
  ^4); //transfer function
7 subplot(131)
8 title('Bode plot')
9 bode(T) // bode plot
10 subplot(132)
11 nyquist(T) // nyquist plot
12 subplot(133)
13 evans(T); //rootlocus
```

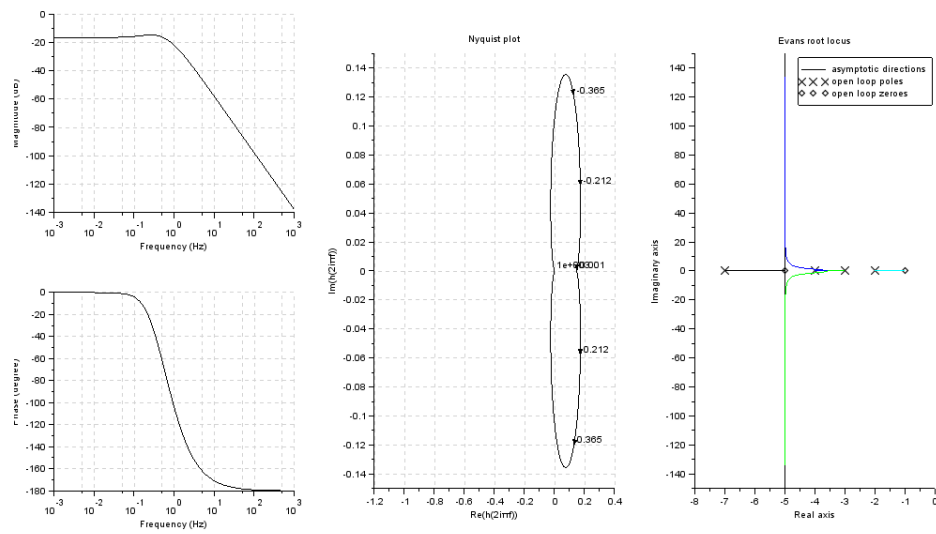


Figure 3.1: stabilityanalysis

Experiment: 4

To compare open loop and closed loop system.

Scilab code Solution 4.04 systemcomparison

```
1 // 4 To compare open loop and closed loop system.
2 //OS=Windows XP sp3
3 //Scilab version 5.4.0
4 //sample values
5 s=%s
6 z=0.1
7 num=1;den =36+2*z*s+s^2;//input numerator and
   denominator of transfer function
8 TF = syslin('c',num,den)//transfer function of
   system
9 disp(TF)
10 subplot(121)
11 t=0:0.1:30;
12 y1 = csim('step', t, TF);//transient response
13 title('Open Loop system');
14 plot(t, y1)
15 s=%s
16 num1=s;den1 =1+s;
17 TF1 = syslin('c',num1,den1)
```

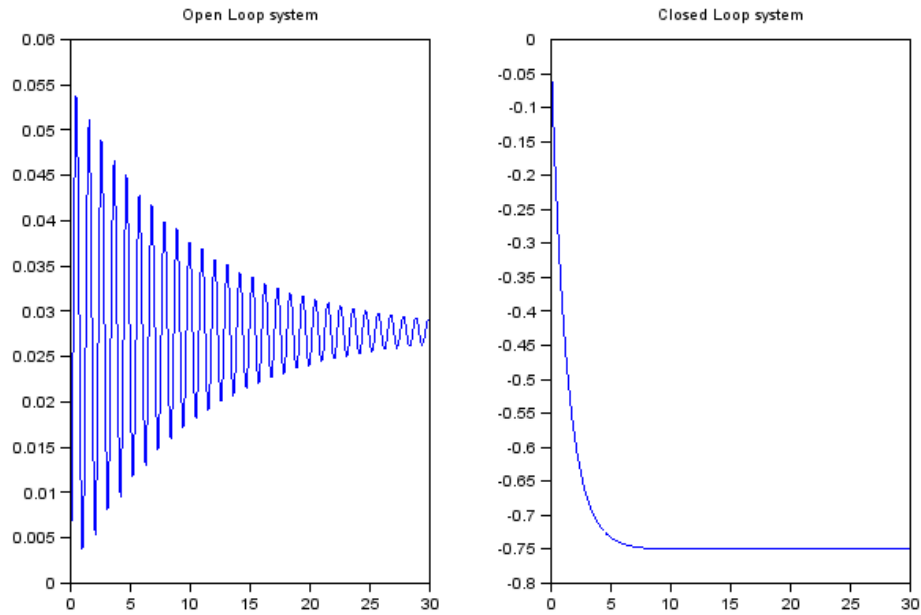


Figure 4.1: systemcomparison

```

18 num2=1;den2 =3;
19 TF2 = syslin('c',num2,den2)
20 S1=TF1/.TF2
21 disp(S1)
22 subplot(122)
23 t=0:0.1:30;
24 y1 = csim('step', t, S1);
25 title('Closed Loop system');
26 plot(t, y1)

```

This code can be downloaded from the website www.scilab.in

Experiment: 5

To use block reduction technique to get the equivalent system

Scilab code Solution 5.05 blockreduction

```
1 //          5 To use block reduction technique to get
   the equivalent system
2 // this is is a typical case where for given block
   daigram where h3 and h4 form a loop which is in
   series to h1 and h3 is negative feedback to the
   combination
3 //OS=Windows XP sp3
4 //Scilab version 5.4.0
5 //sample values
6 clc
7 s=poly(0, 's');
8 n1=[2*s]; //input numerator and denomenator of
   transfer function
9 d1=[3+s^2];
10 h1=syslin('c',n1/d1); //transient response
11 n2=[2];
12 d2=[s+4];
```

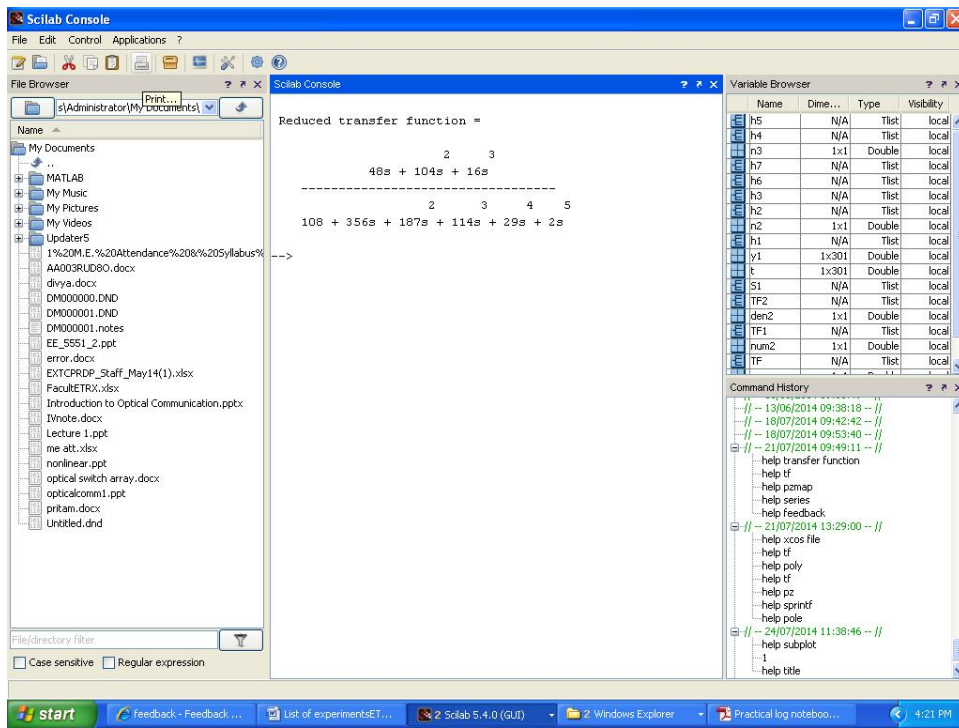


Figure 5.1: blockreduction

```

13 h2=syslin('c',n2/d2);
14 n3=[4];
15 d3=[s+6];
16 h3=syslin('c',n3/d3);
17 n4=[s];
18 d4=[1+2*s];
19 h4=syslin('c',n4/d4);
20 h5=h3/.h4
21 h6=h1*h5
22 h7=h6/.h3
23 disp(h7, "Reduced transfer function =")

```

Experiment: 6

To implement state model reduction technique for the given system

Scilab code Solution 6.06 statemodel

```
1 //          6 To implement state model reduction
   technique for the given system
2 //OS=Windows XP sp3
3 //Scilab version 5.4.0
4 //sample values
5 clc
6 clear all
7 s=poly(0,'s');
8 X1=[0 1;-6 -5];//X1, X2 X3 represent A B C matrix of
   state space representation
9 X2=[0 ; 1];
10 X3=[8 1 ]
11 [n1, n2] = size(X1)
12 I=eye(n1,n2)//identity matrix
13 X=s*I-X1
14 phi=inv(X)//inverse of matrix
15 Y=X3*phi
```

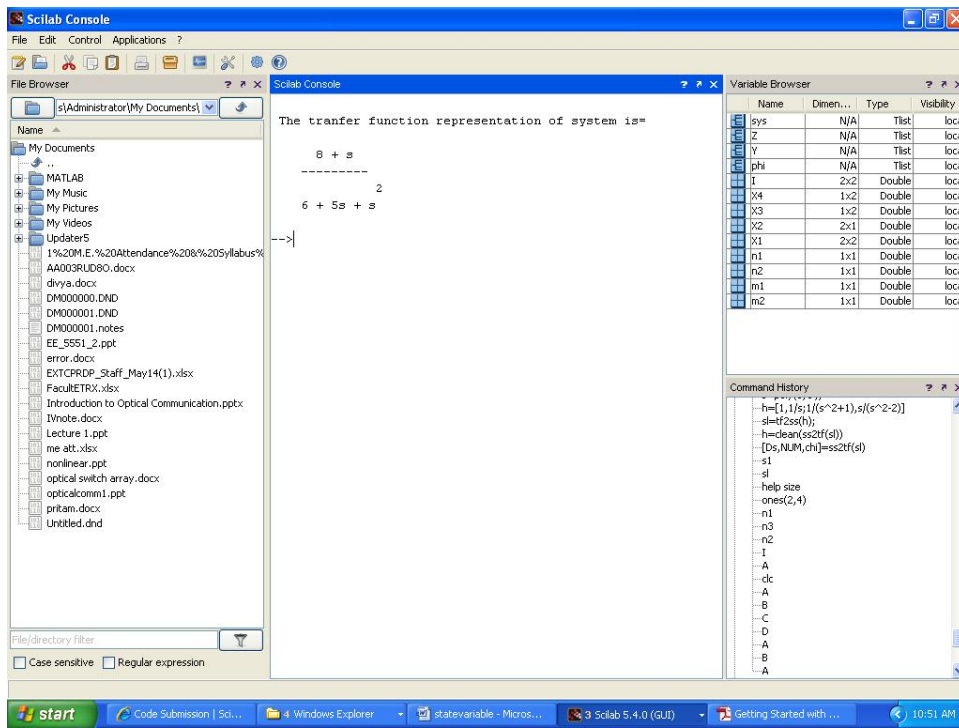


Figure 6.1: statemodel

```

16 Z=Y*X2
17 // sys=tf2ss(Z)
18 disp(Z,"The tranfer function representation of
    system is=")
19 //disp(sys)

```
