

Scilab Manual for  
Control System Engineering  
by Dr R. Anjana  
Electrical Engineering  
Laxmi Institute of Technology<sup>1</sup>

Solutions provided by  
Dr R. Anjana  
Electrical Engineering  
Gujarat Technological University

March 10, 2025

<sup>1</sup>Funded 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
2 To obtain overall transfer function of a system connected in series, parallel and feedback configuration	4
4 Routh Stability Criterion	7
12 DC motor Response	11

# List of Experiments

Solution 2.02	Block Diagram Reduction . . . . .	4
Solution 4.04	Routh Hurwitz Criterion . . . . .	7
Solution 12.0	DC motor Response . . . . .	11

## Experiment: 2

# To obtain overall transfer function of a system connected in series, parallel and feedback configuration

Scilab code Solution 2.02 Block Diagram Reduction

```
1 // VERSION: Scilab: 5.5.2
2 // OS: windows 7
3 // CAPTION: DC motor Response
4
5 //To obtain overall transfer function of a system
   connected in series , parallel and feedback
   configuration
6
7
8 clc;
9 clear all;
10 p = input('enter the Co-efficient of numerator
   polynomial for p:');
11 a = input('enter the Co-efficient of denominator
   polynomial for p:');
```

```

12 q=input('enter the Co-efficient of numerator
    polynomial for q:');
13 b=input('enter the Co-efficient of denominator
    polynomial for q:');
14
15 C = poly([p], 's','coeff')
16 D = poly([q], 's','coeff')
17 X = poly([a], 's','coeff')
18 Y =poly([b], 's','coeff')
19
20 G1 = syslin('c',C,X)
21 G2 = syslin('c',D,Y)
22
23 x = G1*G2;
24 y = G1 + G2
25 z = G1 /. G2
26 disp(x, 'the series function of G1 & G2 is:')
27 disp(y, 'the Parallel function of G1 & G2 is:')
28 disp(z, 'the Feedback function of G1 & G2 is:')
29
30
31 // Sample output
32 //enter the Co-efficient of numerator polynomial for
    p:[1]
33 //enter the Co-efficient of denominator polynomial
    for p:[1 2]
34 //enter the Co-efficient of numerator polynomial for
    q:[4]
35 //enter the Co-efficient of denominator polynomial
    for q:[1]
36
37 // the series function of G1 & G2 is:
38
39 //      4
40 //      ———
41 // 1 + 2s
42
43 //the Parallel function of G1 & G2 is:

```

```
44
45 //      5 + 8s
46 //      -----
47 //      1 + 2s
48
49 // the Feedback function of G1 & G2 is :
50
51 //      1
52 //      -----
53 //      5 + 2s
```

---

# Experiment: 4

## Routh Stability Criterion

Scilab code Solution 4.04 Routh Hurwitz Criterion

```
1 // VERSION: Scilab: 5.4.2
2 // OS: windows 7
3 // CAPTION: Routh-Hurwitz Criterion
4
5 clc;
6 disp(' ')
7 D=input('Input coefficients of characteristic
      equation, i.e: [an an-1 an-2 ... a0]= ');
8 l=length (D);
9
10 disp(' ')
11 disp('-----')
12 disp('Roots of characteristic equation is:')
13 roots(D)
14 if modulo(l,2) = 0
15     m=zeros(1,l/2);
16     [cols,rows]=size(m);
17     for i=1:rows
18         m(1,i)=det(1,(2*i)-1);
19         m(2,i)=det(1,(2*i));
20     end
```



```

21 else
22     m=zeros(1,(l+1)/2);
23     [cols,rows]=size(m);
24     for i=1:rows
25         m(1,i)=D(1,(2*i)-1);
26     end
27     for i=1:((l-1)/2)
28         m(2,i)=D(1,(2*i));
29     end
30 end
31
32 for j=3:cols
33
34     if m(j-1,1)==0
35         m(j-1,1)=0.001;
36     end
37
38     for i=1:rows-1
39         m(j,i)=(-1/m(j-1,1))*det([m(j-2,1) m(j-2,i
40             +1);m(j-1,1) m(j-1,i+1)]);
41     end
42
43 disp('—————The Routh–Hurwitz array is:—————',
44     m)
45 //% —————End of Bulding array
46
47 //% Checking for sign change
48 Temp=sign(m);a=0;
49 for j=1:cols
50     a=a+Temp(j,1);
51 end
52 if a==cols
53     disp('          —————> System is Stable <————')
54 else
55     disp('          —————> System is Unstable <————')

```

```

56 end
57
58     ///Sample Input for unstable system
59
60     //input coefficients of characteristic equation ,
        i.e:[an an-1 an-2 ... a0]= [1 2 3 4 5]
61
62
63
64 // -----
65
66 // Roots of characteristic equation is:
67
68 //     1.     3.     5.
69 //     2.     4.     0.
70 //     1.     5.     0.
71 //    - 6.     0.     0.
72 //     5.     0.     0.
73
74 // -----The Routh-Hurwitz array is:-----
75
76 //           -----> System is Unstable <-----
77
78
79 // == Check for stable system ==
80
81 //Input coefficients of characteristic equation ,i.e
        :[an an-1 an-2 ... a0]= [ 1 1 1]
82
83
84 // -----
85
86 // Roots of characteristic equation is:
87
88 //     1.     1.
89 //     1.     0.
90 //     1.     0.
91

```

```
92 // -----The Routh-Hurwitz array is:-----  
93  
94 //           -----> System is Stable <-----
```

---

# Experiment: 12

## DC motor Response

Scilab code Solution 12.0 DC motor Response

```
1 // VERSION: Scilab: 5.5.2
2 // OS: windows 7
3 // CAPTION: DC motor Response
4
5
6 clc;
7 clear;
8 format('v',7)
9 // given data :
10 V=input('Enter the value of voltage:'); // enter 230
    vs
11 Vm=sqrt(2)*V; // in volts
12 Ka=input('Enter the value of ka:'); // enter Ka = 1;
13 QR=input('Enter the value of QR:'); // enter QR = 1;
14 Ra=input('Enter the value of Ra:'); // enter Ra =
    0.05
15 Alpha=input('Enter the angle'); // enter angle = 30 (
    in degree)
16 Y=(60/(2*%pi)); //
```

```

17 Z=((Vm/%pi)*(1+cosd(Alpha))); //
18 X=(Ra/(0.5)^2)
19 for i=1:8
20 Wm(i)= (Z -(i)*X)*Y; //
21 end
22 Wm=[(Y*Z);Wm(1);Wm(2);Wm(3);Wm(4);Wm(5);Wm(6);Wm(7);
23 Wm(8)]
24 disp(Wm ," Varoius values of speed in RPM is ")
25 T=[0;1;2;3;4;5;6;7;8];
26 plot2d(T,Wm)
27 xlabel("Torque ,N*m")
28 ylabel(" Speed ( rpm) for alpha=30 degree ")
29
30 // == Sample output ===//
31
32 //Enter the value of voltage:230
33 //Enter the value of ka:1
34 //Enter the value of QR:1
35 //Enter the value of ra:0.05
36 //Enter the angle30
37
38 // Varoius values of speed in RPM is
39
40 // 1844.9
41 // 1843.
42 // 1841.1
43 /// 1839.2
44 // 1837.3
45 // 1835.4
46 // 1833.5
47 // 1831.6
48 // 1829.7

```

---