

Scilab Manual for  
Radio Frequency Circuit Design  
by Prof Rajiv Tawde  
Others  
Padmabhushan Vasantdada Patil  
Pratishthan's College of Engineering/Mumbai  
University<sup>1</sup>

Solutions provided by  
Mr MEHUL M. SOLANKI  
Others  
Mumbai University/PVPPCOE

June 15, 2026

<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
1 To design constant-k high pass filter.	4
2 To design prototype Band Pass filter.	6
3 Compute skin depth and ac & dc resistance of a wire at given frequency.	8
4 To design m-derived T and pi section low-pass filter.	11
5 To design m-derived T and pi section high-pass filter.	13
6 Compute width of trace, phase velocity and wavelegnth of microstripline	15
7 To design composite low-pass filter.	18
8 To design composite high-pass filter.	20
9 To design Butterworth low pass filter with 3dB ripple and plot response of filter.	22
10 To design Low pass Chebychev filter and plot response.	24
11 To convert low- pass filter to high- pass ,band -pass and band-stop.	26

# List of Experiments

Solution 1.0	Experiment number 1 . . . . .	4
Solution 2.0	Experiment number 2 . . . . .	6
Solution 3.0	Experiment number 3 . . . . .	8
Solution 4.0	Experiment number 4 . . . . .	11
Solution 5.0	Experiment number 5 . . . . .	13
Solution 6.0	Experiment number 6 . . . . .	15
Solution 7.0	Experiment number 7 . . . . .	18
Solution 8.0	Experiment number 8 . . . . .	20
Solution 9.0	Experiment number 9 . . . . .	22
Solution 10.0	Experiment number 10 . . . . .	24
Solution 11.0	Experiment number 11 . . . . .	26

# Experiment: 1

## To design constant-k high pass filter.

Scilab code Solution 1.0 Experiment number 1

```
1 //AIM:To design constant-k high pass filter
2 //Software version Scilab 5.5.2
3 //OS Windows 7
4
5 clear;
6 clc;
7 //We will design constant-k T section high pass
  filter.
8 R0=600;//Nominal characteristic impedance in ohms.
9 disp('ohms',R0,'Considered value of nominal
  characteristic impedance : R0 = ')
10 fc=10^4;//Cutoff frequency in Hz.
11 L=(R0)/(4*%pi*fc);
12 disp('H',L,'L=')
13 C=1/(4*%pi*fc*R0);
14 disp('F',C,'C=')
15 //Each capacitor in the series arm of T section is 2
  C
16 twoC=2*C;
```

```

17 disp('F',twoC,'2C=')
18
19 //(i): Computing Z0T:
20 //We will calculate the characteristic impedance and
    phase constant
21 // at (say) 25KHz
22 f=25*(10^3);
23 Z0T=R0*sqrt(1-((fc/f)^2));
24 disp('ohms',Z0T,'Z0T=')
25 b=2*asin(fc/f); //b=Beta
26 b_degrees=b*(180/%pi);
27 disp('degrees',b_degrees,'Beta=')
28
29 //(ii): Computing alpha:
30 f1=5*(10^3);
31 alpha=2*(acosh(fc/f1));
32 disp(alpha,'alpha=')
33
34 R01=sqrt(L/C);
35 disp('ohms',R01,'Calculated value of R0 = ')
36 if R01==R0 then
37     disp('Since the calculated value of the nominal
        characteristic impedance')
38     disp('is the same as the considered value of the
        nominal characteristic impedance,')
39     disp('our design is perfect')
40 else
41     disp('Design is not correct')
42 end

```

---

## Experiment: 2

# To design prototype Band Pass filter.

Scilab code Solution 2.0 Experiment number 2

```
1 //Aim:To design prototype Band Pass filter .
2 //Software version Scilab 5.5.2
3 //OS Windows 7
4
5 clear;
6 clc;
7 R0=600;//Nominal characteristic impedance in ohms.
8 disp('ohms',R0,'Considered value of nominal
      characteristic impedance : R0 = ')
9 fc=10^4;//Cutoff frequency in Hz.
10 f2=5000;//Upper cutoff frequency in Hz.
11 f1=2000;//Lower cutoff frequency in Hz.
12 L1=R0/(%pi*(f2-f1));
13 disp('H',L1,'L1=')
14 L=L1/2;
15 disp('H',L,'L1/2=')
16 C1=(f2-f1)/(4*%pi*R0*f1*f2);
17 disp('F',C1,'C1=')
18 C=2*C1;
```

```

19 disp('F',C,'2*C1=')
20 L2=(R0*(f2-f1))/(4*pi*f1*f2);
21 disp('H',L2,'L2=')
22 C2=1/(pi*R0*(f2-f1));
23 disp('F',C2,'C2=')
24
25 //We will now check whether our design is correct or
    not
26 //For bandpass filter ,we have the relation :
27 // R0=sqrt(L2/C1)=sqrt(L1/C2)
28 //Let
29 R01=sqrt(L2/C1);
30 R02=sqrt(L1/C2);
31 disp('ohms',R01,'R01=')
32 disp('ohms',R02,'R02=')
33 if R01==R02 then
34     R0new=R01
35     disp('So, R01=R02')
36 else
37     disp('Design is not correct')
38 end
39 if R0new==R0 then
40     disp('Since R01=R02=R0 ,it indicates that the
        calculated values of the nominal
        characteristic impedance')
41     disp('are the same as the considered value of
        the nominal characteristic impedance,')
42     disp('and so our design is perfect')
43 else
44     disp('Design is not correct')
45 end

```

---

## Experiment: 3

Compute skin depth and ac & dc resistance of a wire at given frequency.

Scilab code Solution 3.0 Experiment number 3

```
1 //AIM: Compute skin depth and ac & dc resistance of
  a wire at given frequency.
2 //Software version Scilab 5.5.2
3 //OS Windows 7
4
5 clc;
6 clear;
7 //Now let us consider 1GHz and 10GHz be the given
  frequencies
8 //Let the length of the wire be 10cm with a diameter
  of 1mm
9 Length=10*(10^-2); //in metres
10 a=0.5*(10^-3); //Since diameter is 1mm,so the radius
  'a' will be 0.5*10^-3 metres
11 sigmaCu=64.516*10^6; //in mho/metres
12 sigmaAl=40*10^6; //in mho/metres
13 sigmaAu=48.544*10^6; //in mho/metres
```

```

14 //sigmacu ,sigmaAl ,sigmaAu are the conductivities of
    copper , aluminium
15 //and gold respectively .
16 //The formula for skin depth is 1/sqrt(%pi*f*mu0*
    sigma)
17 //Since we have to calculate the skin depth for 3
    values of frequency
18 // & 3 values of sigma ,we will first calculate the
    value of
19 //1/sqrt(%pi*mu0) only which will be required for
    initial calculations
20 mu0=4*%pi*(10^(-7)); //Standard value of permeability
    of free space (in H/m)
21 b=inv(sqrt(%pi*mu0));
22 //(i) At 1GHz
23 f1=1*(10^9); //in Hz
24 skindepthcu=b/(sqrt(f1*sigmacu));
25 skindepthAl=b/(sqrt(f1*sigmaAl));
26 skindepthAu=b/sqrt((f1*sigmaAu))
27 Rdccu=Length/(%pi*a*a*sigmacu)
28 RdcAl=Length/(%pi*a*a*sigmaAl);
29 RdcAu=Length/(%pi*a*a*sigmaAu);
30 Raccu=(Rdccu*a)/(2*skindepthcu);
31 RacAl=(RdcAl*a)/(2*skindepthAl);
32 RacAu=(RdcAu*a)/(2*skindepthAu);
33 disp(" At 1GHz")
34 printf(" Skindepth for Cu,Al,Au is as shown
    respectively=\n%g m\n%g m\n%g m\n",skindepthcu ,
    skindepthAl ,skindepthAu)
35 printf("DC resistance for Cu,Al,Au is as shown
    respectively=\n%g ohms\n%g ohms\n%g ohms\n",Rdccu
    ,RdcAl ,RdcAu)
36 printf("AC resistance for Cu,Al,Au is as shown
    respectively=\n%g ohms\n%g ohms\n%g ohms\n",Raccu
    ,RacAl ,RacAu)
37 //(ii) At 10GHz
38 f2=10*10^9; //in Hz
39 skindepthcu2=b/(sqrt(f2*sigmacu));

```

```
40 skindepthAl2=b/sqrt((f2*sigmaAl));
41 skindepthAu2=b/sqrt((f2*sigmaAu));
42 Raccu=Rdccu*a/(2*skindepthcu2);
43 RacAl=RdcAl*a/(2*skindepthAl2);
44 RacAu=RdcAu*a/(2*skindepthAu2);
45 disp(" At 10GHz")
46 printf("skindepth for Cu,Al,Au is as shown
         respectively=\n%g m\n%g m\n%g m\n",skindepthcu2,
         skindepthAl2,skindepthAu2)
47 printf("AC resistance for Cu,Al,Au is as shown
         respectively=\n%g ohms\n%g ohms\n%g ohms",Raccu,
         RacAl,RacAu)
```

---

## Experiment: 4

# To design m-derived T and pi section low-pass filter.

Scilab code Solution 4.0 Experiment number 4

```
1 //Aim:To design m-derived T and pi section low-pass
  filter .
2 //Software version Scilab 5.5.2
3 //OS Windows 7
4
5 clear;
6 clc;
7 fc=1800;//fc is the cutoff frequency
8 f_infinity=2000;//f_infinity is the infinite
  attenuation frequency
9 R0=600;//R0 is the nominal characteristic impedance
10
11 //Computing the value of m:
12 m=sqrt(1-((fc/f_infinity)^2));
13 disp(m, 'm=')
14 L=R0/(%pi*fc);//L is the series arm inductance
15 disp('H',L, 'L=')
16 C=1/(%pi*R0*fc);//C is the shunt arm capacitance
17 disp('F',C, 'C=')
```

```

18
19 //Computation of values of the elements for the T
    section
20 //of the m-derived filter :
21 a=(m*L/2);
22 disp('H',a,'mL/2=')
23 b=m*C;
24 disp('F',b,'mC=')
25 c=((1-(m^2))/(4*m))*L;
26 disp('H',c,'((1-(m^2))/(4*m))*L=')
27
28 //Computation of the values of the elements for the
    pi-section
29 //of the m-derived filter :
30 d=(m*C)/2;
31 disp('F',d,'mC/2=')
32 e=m*L;
33 disp('H',e,'mL=')
34 f=((1-(m^2))/(4*m))*C;
35 disp('F',f,'((1-(m^2))/(4*m))*C=')

```

---

## Experiment: 5

### To design m-derived T and pi section high-pass filter.

Scilab code Solution 5.0 Experiment number 5

```
1 //Aim:To design m-derived T and pi section high-pass
    filter .
2 //Software version Scilab 5.5.2
3 //OS Windows 7
4
5 clear;
6 clc;
7 f_infinity=3.6*(10^3);//Frequency at infinite
    attenuation .
8 fc=4*(10^3);//Cut-off frequency .
9 R0=600;//Nominal characteristic impedance.
10
11 //Computing the values of m:
12 m=sqrt(1-((f_infinity/fc)^2));
13 disp(m, 'm=')
14 C=1/(4*%pi*fc*R0);//C is the series arm capacitance.
15 disp('Farads',C, 'C=')
16 L=R0/(4*%pi*fc);//L is the shunt arm inductance.
17 disp('Henry',L, 'L=')
```

```

18
19 //Computation of the values of the circuit elements
    for T-section
20 //of m-derived filter :
21 a=(2*C)/m;
22 disp('Farads',a,'2C/m=')
23 b=L/m;
24 disp('Henry',b,'L/m=')
25 c=((4*m)/(1-(m^2)))*C;
26 disp('Farads',c,'((4*m)/(1-(m^2)))*C=')
27 //Computation of the values of the circuit elements
    for pi-section
28 //of m-derived filter :
29 d=(2*L)/m;
30 disp('Henry',d,'2L/m=')
31 e=C/m;
32 disp('Farads',e,'C/m=')
33 f=((4*m)/(1-(m^2)))*L;
34 disp('Henry',f,'((4*m)/(1-(m^2)))*L=')

```

---

## Experiment: 6

# Compute width of trace, phase velocity and wavelegngth of microstripline

Scilab code Solution 6.0 Experiment number 6

```
1 //AIM:Compute width of trace ,phase velocity and
   wavelength of microstripline.
2 //Software version Scilab 5.5.2
3 //OS Windows 7
4
5 clc;
6 clear;
7 Zo=50; //Considered characteristic impedance Zo=50
   ohms
8 printf(" Considered characteristic impedance=%g ohms\
   n\n",Zo)
9 //We know that ,for characteristic impedence of 50
   ohms,the value of w/h is 1.9
10 mu0=4*%pi*(10^(-7));
11 epsilon0=8.85*(10^(-12));
12 Zf=sqrt(mu0/epsilon0); //in ohms
13 printf(' Value of Zf=%g ohms\n\n',Zf)
```

```

14 h=40; //in mil
15 f=2*10^9; //in Hz
16 //Er=epsilon_r
17 Er=4.6;
18 //taking A=2*pi*(Zo/Zf)*(sqrt(Er+1/2))+((Er-1)/(Er
    +1))*(0.23+0.11/Er)
19 A=2*pi*(Zo/Zf)*(sqrt((Er+1)/2))+((Er-1)/(Er+1))
    *(0.23+0.11/Er);
20 printf("Value of A=%g \n\n",A)
21 a=(8*exp(A))/(exp(2*A)-2); //a means the calculation
    of w/h
22 //Eeff=effective dielectric constant.
23 if a>1 then
24 Eeff=((Er+1)/2)+((Er-1)/2)*((1+12*(a^-1))^-0.5);
25 printf("Effective permittivity=%g\n\n",Eeff)
26 Zo1=Zf/(sqrt(Eeff)*(1.393+a+(2/3)*log(a+1.444)));
27 else
28 Eeff=((Er+1)/2)+((Er-1)/2)*(((1+12*(a^-1))^-0.5)
    +(0.04(1-a)^2));
29 printf("Effective permittivity=%g\n\n",Eeff)
30 Zo1=(Zf/(2*pi*sqrt(Eeff)))*log((8*(a^-1))+(0.25*a))
    ;
31 end
32 Zo1=int(Zo1);
33 printf("Computed characteristic impedance,Zo1=%g
    ohms\n\n",Zo1)
34 if Zo1==Zo then
35     printf("The computed value of Zo1 is same as the
        considered value of Zo\n")
36     printf("Hence our design is correct.\n\n")
37 else
38     printf("Design is incorrect.\n\n")
39 end
40 w=a*h;
41 printf("Width of the trace=%g mils\n\n",w)
42 c=3*10^8;
43 vp=c/(sqrt(Eeff));
44 printf("Phase velocity=%g m/s\n\n",vp)

```

```
45 lambda=vp/f;  
46 printf("Wavelength=%g m",lambda)
```

---

# Experiment: 7

## To design composite low-pass filter.

Scilab code Solution 7.0 Experiment number 7

```
1 //Aim:To design composite low-pass filter.
2 //Software version Scilab 5.5.2
3 //OS Windows 7
4
5 clear;
6 clc;
7 R0=75;//R0 is the nominal impedance(in ohms).
8 fc=2*(10^6);//fc is the cut-off frequency in Hz.
9 f_infinity=2.05*(10^6);//f_infinity represents that
   the infinite attenuation.
10 //pole is placed at 2.05 MHz.
11
12 //Constant k section :
13 L=R0/(%pi*fc);
14 disp('H',L,'L=')
15 C=1/(%pi*R0*fc);
16 disp('F',C,'C=')
17
18 //m-derived section :
```

```

19 m=sqrt(1-((fc/f_infinity)^2));
20 disp(m, 'm=')
21 a=(m*L)/2;
22 disp('H',a, 'For m-derived section , mL/2=')
23 b=m*C;
24 disp('F',b, 'For m-derived section , mC=')
25 c=((1-(m^2))/(4*m))*L;
26 disp('H',c, 'For m-derived section , ((1-(m^2))/(4*m))
    )*L=')
27
28 //Matching sections: with m=0.6
29 //Let us call this new 'm' as 'm1'.
30 m1=0.6;
31 d=(m1*L)/2;
32 disp('H',d, 'For matching section , m1L/2=')
33 e=(m1*C)/2;
34 disp('F',e, 'For matching section , m1C/2=')
35 f=((1-(m1^2))/(2*m1))*L;
36 disp('H',f, 'For matching section , ((1-(m1^2))/(2*m1)
    ))*L=')

```

---

# Experiment: 8

## To design composite high-pass filter.

Scilab code Solution 8.0 Experiment number 8

```
1 //Aim:To design composite high-pass filter .
2 //Software version Scilab 5.5.2
3 //OS Windows 7
4
5 clear;
6 clc;
7 R0=75;//R0 is the nominal impedance (in ohms)
8 fc=50*(10^3);//fc is the cut-off frequency (in Hz)
9 f_infinity=48*(10^3);
10
11 //Constant k section :
12 L=R0/(4*%pi*fc);
13 disp('H',L,'L=')
14 C=1/(4*%pi*R0*fc);
15 disp('F',C,'C=')
16 twoC=2*C;
17 disp('F',twoC,'2C=')
18
19 //m-derived section :
```

```

20
21 //Computing m
22 m=sqrt(1-((f_infinity/fc)^2));
23 disp(m, 'm=')
24 a=(2*C)/m;
25 disp('F', a, '2C/m=')
26 b=L/m;
27 disp('H', b, 'L/m=')
28 c=(4*m*C)/(1-(m^2));
29 disp('F', c, '(4*m*C)/(1-(m^2))=')
30
31 //Matching sections: with m=0.6
32 //Let us call this new 'm' as m1
33 m1=0.6;
34 d=(2*C)/m1;
35 disp('F', d, '2C/m1=')
36 e=(2*L)/m1;
37 disp('H', e, '2L/m1=')
38 f=(2*m1*C)/(1-(m1^2));
39 disp('F', f, '(2*m1*C)/(1-(m1^2))=')
40 //Some of the capacitors will be in series
41 C1=(d*twoC)/(d+twoC);
42 disp('F', C1, 'C1=')
43 C2=(twoC*a)/(twoC+a);
44 disp('F', C2, 'C2=')
45 C3=(a*d)/(a+d);
46 disp('F', C3, 'C3=')

```

---

## Experiment: 9

To design Butterworth low pass filter with 3dB ripple and plot response of filter.

Scilab code Solution 9.0 Experiment number 9

```
1 //AIM:To design Butterworth low pass filter with 3dB
   ripple and plot response of filter.
2 //Software version Scilab 5.5.2
3 //OS Windows 7
4
5 clc;
6 clear;
7 f=2; //f=normalized frequency.
8 IL=20; //IL is the insertion loss (in dB).
9 E=10^0.3-1;
10 N=0.5*(log(10^(0.1*IL)-1)/log((f)+log(E)));
11 A=N-int(N);
12 if(A>0)
13     N=int(N)+1;
14 end;
15 disp(N,"Order of filter=")
16 printf("values of filter element are\n")
```

```

17 for i=1:N
18     g(i)=2*sin((((i*2)-1)*pi)/(2*N));
19     printf("g(%g)=%g\n",i,g)
20 end
21
22 //IL=10log(1+(f)^2*N)
23 //Now plotting IL Vs normalized frequency graph.
24 f=0:0.1:5;
25 IL=10*log10(1+f.^(2*N));
26 title('Response of Butterworth low pass filter in
        terms of ''Insertion-loss(IL) versus frequency''
        ');
27 ylabel("IL in dB");
28 xlabel("normalized frequency");
29 plot2d(f,IL);
30 //representing values of IL in console.
31 f=0:0.5:5;
32 IL=10*log10(1+f.^(2*N));
33 disp(IL,"IL in dB",f,"normalized frequency")

```

---

## Experiment: 10

### To design Low pass Chebychev filter and plot response.

Scilab code Solution 10.0 Experiment number 10

```
1 //AIM:To design Low pass Chebyshev filter and plot
  response.
2 //Software version Scilab 5.5.2
3 //OS Windows 7
4
5 clc;
6 clear;
7 //ripple
8 rp=6;//ripple in dB.
9 ohm=2;//ohm=normalized frequency.
10 IL=50;//IL is the insertion loss (in dB).
11 a=sqrt(10^(rp/10)-1)
12 //To obtain order of filter N
13 // Using equation  $IL=10*\log(1+a^2*(T^2(\quad)))$ 
14 num=acosh(sqrt((10^(0.1*IL)-1)/(a^2)))
15 dem=acosh(ohm)
16 N=num/dem
17 x=N-int(N)
18 if(x>0);
```

```
19     N=int(N)+1;
20 end
21 disp(N," Order of filter=");
22 ohm=0:0.01:1;
23 T=cos(N*(acos(ohm)));
24 IL=10*log10(1+(T.^2).*a^2);
25 title('Response of low pass Chebychev filter');
26 ylabel("IL in dB");
27 xlabel("normalized frequency");
28 plot2d(ohm,IL);
29 ohm=0:0.1:1;
30 T=cos(N*(acos(ohm)));
31 IL=10*log10(1+(T.^2).*a^2);
32 disp(IL," IL in dB",ohm," normalized frequency");
```

---

## Experiment: 11

To convert low- pass filter to high- pass ,band -pass and band-stop.

Scilab code Solution 11.0 Experiment number 11

```
1 //Aim:To convert low- pass filter to high- pass ,
   band -pass and band-stop.
2 //Software version Scilab 5.5.2
3 //OS Windows 7
4
5 clc;
6 clear;
7 //we are considering order of filter as 5 hence
   there will be three
8 //inductors and two capacitors
9 L=[1.7058 2.5408 1.7058]
10 c=[1.2296 1.2296]
11 fc=60*10^6
12 wc=2*%pi*fc
13 fu=2*10^9
14 wu=2*%pi*fu
15 fl=1.9*10^9
```

```

16 w1=2*%pi*f1
17 wd=wu-w1;
18 wo=sqrt(wu*w1)
19 del=wd/wo;
20 //high pass filter.
21 disp("Converting values to High pass filter")
22 disp("Here inductor is converted into capacitor &
      capacitor is converted into inductor")
23 disp("For capacitor")
24 c1=(L*wc).^-1;
25 disp(c1)
26 disp("For inductor")
27 L1=(wc*c).^-1;
28 disp(L1)
29 //band pass filter
30 disp("Converting values to band pass filter")
31 disp("Here inductor is converted into series
      inductance and capacitance value are as follows")
32 disp("For inductor")
33 L2=L*(del*wo).^-1;
34 disp(L2)
35 disp("For capacitor")
36 c2=((L*wo).^-1)*del;
37 disp(c2)
38 disp("Here capacitor is converted into series
      inductance and capacitance value are as follows")
39 disp("For inductor")
40 L2=del/(wo*c);
41 disp(L2)
42 disp("For capacitor")
43 c2=c/(wo*del);
44 disp(c2)
45 //band stop filter.
46 disp("Conversion values of band stop filter")
47 disp("Here inductor is converted into parallel
      inductance and capacitance value are as follows")
48 disp("For inductor")
49 L3=(L*del)/wo;

```

```
50 disp(L3)
51 disp("For capacitor")
52 c3=(wo*L*del).^ -1;
53 disp(c3)
54 disp("Here capacitance is covered into serie
      inductance and cpacitance value are as follows")
55 disp("For inductor")
56 L3=(wo*c*del).^ -1;
57 disp(L3)
58 disp("For capacitor")
59 c3=(c*del)/wo;
60 disp(c3)
```

---