Scilab Textbook Companion for Microwave Engineering by D. M. Pozar[1]

Created by
Amardeep Kumar
B.TECH (pursuing)
Electronics Engineering
M.N.N.I.T, Allahabad
College Teacher
Rajeev Gupta
Cross-Checked by
Sonanya Tatikola, IIT Bombay

May 16, 2016

[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: Microwave Engineering

Author: D. M. Pozar

Publisher: Addison - Wesley Longman, Incorporated

Edition: 1

Year: 1993

ISBN: 0-201-50418-9

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

ELECTROMAGNETIC THEORY

Scilab code Exa 2.1 program to calculate wavelength phase velocity and

```
// example 2.1,page no.-24.
// program to calculate wavelength,phase velocity
   and wave impedence.
f=3*10^9;
mur=3;
muo=4*%pi*10^-7;
eipsilao=8.854*10^-12;
eipsilar=7;
mue=muo*mur;
eipsila=eipsilao*eipsilar;
Vp=sqrt(1/(mue*eipsila));
lamda=Vp/f;
eta=sqrt(mue/eipsila);
//Result
disp(Vp,'phase velocity in meter per second=')
         // phase velocity.
disp(lamda,'wavelength in meter=')   // wavelength.
disp(eta,'wave impedence in ohm=')    // wave
   impedence.
```

Scilab code Exa 2.2 progarm to find out skin depth

```
// example:-2.2.page no.-26.
// progarm to find out skin depth of aluminium, copper,gold and silver at frequency 10GHZ.
f=10*10^9;
muo=4*%pi*10^-7;  // permeability in free space.
omega=2*%pi*f;
sigma_aluminium=3.816*10^7;
sigma_copper=5.813*10^7;
sigma_gold=4.098*10^7;
sigma_silver=6.173*10^7;
delta1=sqrt(2/(omega*muo*sigma_aluminium));
delta2=sqrt(2/(omega*muo*sigma_copper));
delta3=sqrt(2/(omega*muo*sigma_gold));
delta4=sqrt(2/(omega*muo*sigma_silver));
//result
disp(delta1,'skin depth of aluminium in meter=')  // skin depth of aluminium.
disp(delta2,'skin depth of copper in meter=')  // skin depth of copper.
disp(delta3,'skin depth of gold in meter=')  //skin depth of gold.
disp(delta4,'skin depth of silver in meter=')  // skin depth of silver.
```

Scilab code Exa 2.3 program to find the resulting fields

```
// example:-2.3,page no.-31.
//program to find the resulting fields by assumibg plane waves on either side of the current sheet and enforcing the boundary conditions.
```

```
syms E x E1 E2 H1 H2 z Jo A B c N n d ko y;
sym('n*(E2-E1)=0');    //boundary condition to be satisfied at z=0
sym('z*(E2-E1)=0');    //   "        "          "         "
sym('n*(H2-H1)=Jo');   //   "        "          "         "
sym('z*(H2-H1)=Jo');   //   "        "          "         "
E1=A*N*exp(%i*ko*z)*x;   // x component of electric field (region z<0).
H1=A*N*exp(%i*ko*z)*(-y); // -y component of magnetic field (region z<0).
E2=B*N*exp(-%i*ko*z)*x;    // x component of electric field (region z>0).
H2=B*N*exp(-%i*ko*z)*y;    // y component of electric field (region z>0).
disp(E1,'for z<0, E1=')
disp(H1,'for z<0, H1=')
disp(E2,'for z>0, E2=')
disp(H2,'for z>0, H2=')
//from boundary conditions imposed.we get:-
c=[-1 -1;1 -1];
d=[A;B];
c*d==[Jo;0];
d=inv(c)*[Jo;0];
//result
// A=-Jo/2; B=-Jo/2.
disp(d)
```

Scilab code Exa 2.4 program to show decomposition in to RHCP and LHCP

```
// example:-2.4,page no.-34.
// program to show that a circularly polarized plane
```

```
wave can be decomposed in to RHCP and LHCP.
A=sym('A');
B=sym('B');
Eo=sym('Eo');
x=sym('x');
y=sym('y');
Ko=sym('Ko');
z=sym('z');
E=Eo*(x+2*y)*exp(-%i*Ko*z);    // given
// can be written as:=>E=A*(x-y)*exp(-%i*Ko*z)+B*(x+
   y)*exp(-%i*Ko*z),so
p=[1 1;-%i/2 %i/2];
q=[A;B];
r=[1;1];
p*q==Eo*r;
q=inv(p)*Eo*r;
//result
disp('value of A and B will be=')
disp(q)
disp(q(1,1)*(x-y)*exp(-%i*Ko*z)+q(2,1)*(x+y)*exp(-%i
   *Ko*z),'E=')
//conclusion:-any linearly polarized wave can be
   decomposed in to two circularly polarized waves.
```

Scilab code Exa 2.5 program to compute the poynting vector

```
// example:-2.5,page no.-36.
// program to compute the poynting vector for the
   plane wave field.
syms E Eo H k  s n N x r;
E=Eo*exp(-%i*k*r); // electric field.
H=(E/N)*n;  //N is intrinsic impedence,n is unit
   vector.
H1=conj(H)  // conjugate of magnetic field.
s=E*H1;
```

```
//result
disp(s,'poynting vector is(meter square)=')
disp('which shows that power density is flowing in
   the direction of propagation.')
```

Scilab code Exa 2.6 program to compute propagation constant and other

```
// example:-2.6,page no.-46.
// program to compute propagation constan,impedence,
   skin depth,reflection and transmission
   coefficient.
f=1*10^9;
omega=2*%pi*f;
sigma=5.813*10^7;  // for copper.
mue=4*%pi*10^-7;   // permeability in free space.
delta=sqrt(2/(mue*sigma*omega)); // skin depth.
gama=((1+%i)/delta);  //propagation constant.
eta=gama/sigma;  // impedence
etao=377;  //intrinsic impedence in free space.
tao=((eta-etao)/(eta+etao)); // reflection
   coefficient.
t=(2*eta)/(eta+etao);  //transmission coefficient.
// result
disp(delta,'skin depth in meter=')
disp(gama,'propagation constant=')
disp(eta,'intrinsic impedence in ohm=')
disp(tao,'reflection coefficient=')
disp(t,'transmission coefficient=')
```

Scilab code Exa 2.7 program to plot the reflection coefficients

```
// example:-2.7.page no.-50.
```

```
// program to plot the reflection coefficients for parallel and perpendicular polarized plane waves incident from free space on to a dielectric region with Er=2.55,versus incidence angle.
Er=2.55;  // relaitve permittivity of dielectric medium.
N1=377;  // intrinsic impedence
N2=N1/sqrt(Er);  // intrinsic impedence of dielectric medium.
xb=asin(sqrt(1/(1+1/2.55)));  // brewster angle valid only in case of parallel polarization.
xt=acos(sqrt(1-(1/Er)^2*sin(xb))); // angle of transmission.
xi=[0:0.01:%pi/2];  // incidence angle.
// for parallel polarization
N2=N2*cos(xt);
N1=N1*cos(xi);
Tpar=(N2-N1)./(N2+N1);
w=abs(Tpar);
// result
subplot(1,2,1)
xtitle("parallel polarization","xi(incidence angle)","Tpar(reflection coefficient)")
plot2d(xi,w,style=3,rect=[0,0,%pi/2,1])
// for perpendicular polarization.      //NOTE:- in case of this polarization.there is no brewster angle.
xt=acos(sqrt(1-(1/Er)^2*sin(xi)));
n1=377.*cos(xt);
n2=(377/sqrt(Er)).*cos(xi);
Tper=(n2-n1)./(n1+n2);
z=abs(Tper);
//result
subplot(1,2,2)
xtitle("perpendicular polarization","xi(inxidence angle)","Tper(reflection coefficient)")
plot2d(xi,z,style=2,rect=[0,0,%pi/2,1])
```

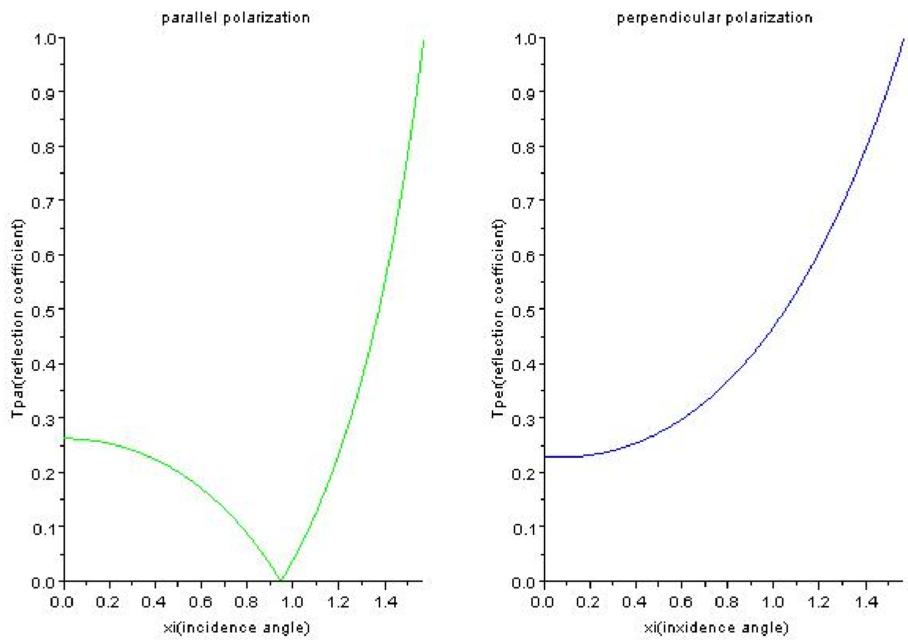

Figure 2.1: program to plot the reflection coefficients

Chapter 3

TRANSMISSION LINE THEORY

Scilab code Exa 3.1 program to determine transmission line parameters

```
//example:-3.1,page no.-72.
// program to determine transmission line parameters.
syms E H Vo P a b Io mue y z Q pi L eipsila G C R Rs w;
E=(Vo/(P*log(b/a)))*exp(-%i*y*z); // in radial direction.
H=(Io/(2*pi*P))*exp(-%i*y*z);   // in phi direction.
H=H*conj(H)*P;
E=E*conj(E)*P;
L=(mue/((Io)^2))*integ(integ(H,P),Q);   // surface integral in culindrical coordinate system
L=limit(L,P,b)-limit(L,P,a);  // limits when integrated w.r.t rho.
L=limit(L,Q,2*pi)-limit(L,Q,0);  // limits when integrated w.r.t phi.
C=(eipsila/(Vo^2))*integ(integ(E,P),Q);  // surface integral in culindrical coordinate system
C=limit(C,P,b)-limit(C,P,a);    // limits when
```

```
    integrated w.r.t rho.
C=limit(C,Q,2*pi)-limit(C,Q,0);  // limits when
    integrated w.r.t phi.
R=(Rs/(Io^2))*integ(H,Q);
R=limit(R,P,b)+limit(R,P,a);
R=limit(R,Q,2*pi)-limit(R,Q,0);  // limits when
    integrated w.r.t phi.
G=((w*eipsila)/(Vo^2))*integ(integ(E,P),Q);  //
    surface integral in culindrical coordinate system
G=limit(G,P,b)-limit(G,P,a);  // limits when
    integrated w.r.t rho.
G=limit(G,Q,2*pi)-limit(G,Q,0);  // limits when
    integrated w.r.t phi.
// result
disp(L,'self-inductance in H/m =')
disp(C,'capacitance in F/m =')
disp(R,'resistance in Ohm/m =')
disp(G,'shunt conductance in S/m =')
```

check Appendix AP 2 for dependency:

`smith_chart_tao.sci`

Scilab code Exa 3.2 program to find out load impedence

```
// example:-3.2,page no.-87.
// program to find out load impedence.
Zo=100;  // characteristic impedence.
tao=0.560+0.215*%i;  // reflection coefficient.
z=(1+tao)/(1-tao);  // normalized impedence(
    normalized w.r.t Zo)
Zl=z*Zo;
// result
disp(Zl,'load impedence = ')
// by smith chart.
smith_chart(tao)
```

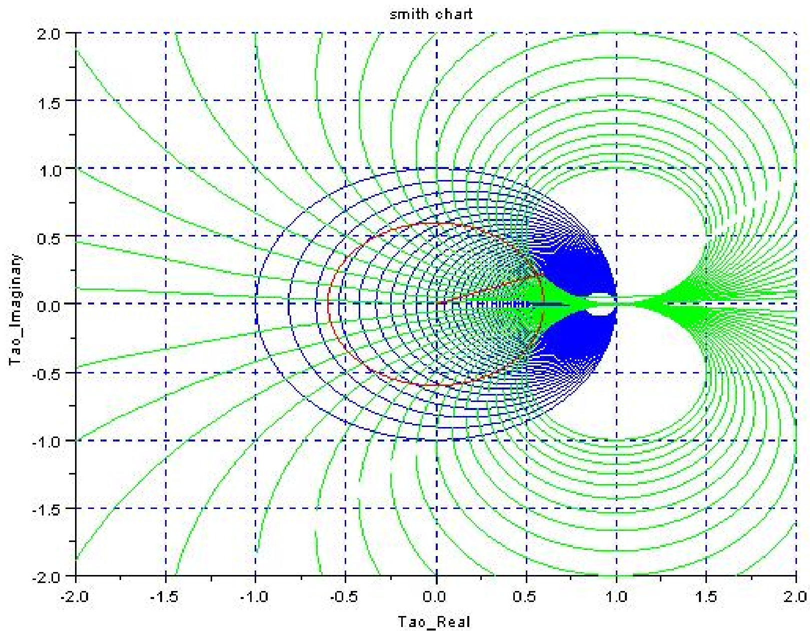

Figure 3.1: program to find out load impedence

```
// when analyse with the help of smith chart.see the
    angle from x=0 axis i.e Tao_real axis.if it is
   above this axis take angle anticlockwise and if
   it is below this axis.take angle clockwise from
   Tao_real axis below.
```

check Appendix AP 4 for dependency:

`reflection_coefficient.sci`

check Appendix AP 2 for dependency:

`smith_chart_tao.sci`

check Appendix AP 5 for dependency:

swr.sci

Scilab code Exa 3.3 program to find out return loss in dB and others

```
// example:-3.3,page no.-87.
// program to find out return loss in dB,SWR and
    reflection coefficient.
Zl=80-40*%i;  // load impedence.
Zo=50; // characteristic impedence.
z=Zl/Zo;  // normalized impedence.
tao=reflection_coefficient(Zl,Zo);
SWR=VSWR(abs(tao));
Rl=-20*log10(abs(tao));
disp(abs(tao),'reflection coefficient = ')
disp(SWR,'standing wave ratio = ')
disp(Rl,'return loss in dB = ')
smith_chart(tao)
// when analyse with the help of smith chart.see the
     angle from x=0 axis i.e Tao_real axis.if it is
    above this axis take angle anticlockwise and if
    it is below this axis.take angle clockwise from
    Tao_real axis below.
```

check Appendix AP 3 for dependency:

input_impedence.sci

check Appendix AP 4 for dependency:

reflection_coefficient.sci

check Appendix AP 5 for dependency:

swr.sci

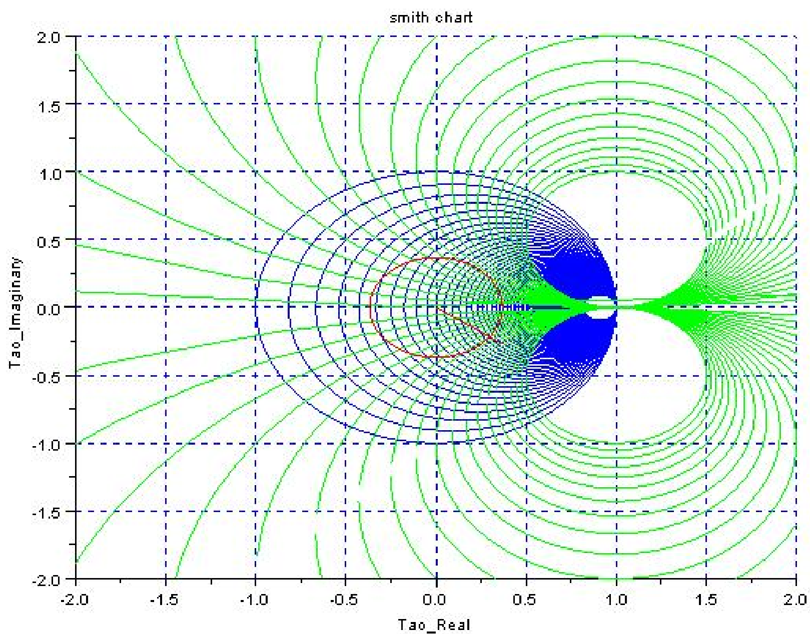

Figure 3.2: program to find out return loss in dB and others

Scilab code Exa 3.4 program to find input impedence and SWR of line

```
// example no.-3.4,page no.-88.
// program to find input impedence and SWR of line.
Zo=75;Zl=37.5+75*%i;l=0.02;eipsilar=2.56;f=3*10^9;c
   =3*10^8;
b=(2*%pi*f*sqrt(eipsilar))/c;  // beta
tao=reflection_coefficient(Zl,Zo);
Zin=input_impedence(tao,b,l,Zo);
// result
disp(Zin,'input impedence = ')
tao=abs(tao);
s=VSWR(tao);
// result
disp(s,'SWR of the line = ')
```

Scilab code Exa 3.5 program to find out load admittance and other

```
// example:-3.5,page no.-91.
// program to find out load admittance and input
   admittance of the line
syms lamda;
Zl=100+50*%i;
Zo=50;
le=0.15;  //electrical length(l/lamda).
b=(2*%pi);
tao=reflection_coefficient(Zl,Zo);
Zin=input_impedence(tao,b,le,Zo);
Yin=1/Zin;
Yl=1/Zl;
// result
disp(Yin,'input admittance = ')
disp(Yl,'load admittance = ')
```

Scilab code Exa 3.6 program to find out characteristic impedence

```
//example:-3.6,page no.-93.
// program to find out characteristic impedence and
   plot the magnitude of reflection coefficient
   versus normalized frequency.
Zl=100;// load impedence
Zi=50;//impedence of line which is to be matched
//as it is a quarter wave transformer so,Zi=(Zo)^2/
   zl;
Zo=sqrt(Zi*Zl);
disp(Zo,'characteristic impedence of tne matching
   section=')
syms f fo x;
x=f/fo;
x=0:0.001:4;
y=(%pi/2)*(x);
Zin=Zo*(((Zl*cos(y))+(Zo*%i*sin(y)))./((Zo*cos(y))+(
   Zl*%i.*sin(y))))
tao=((Zin-Zo)./(Zin+Zo));
tao=abs(real(tao)+imag(tao));
plot2d(x,tao,style=6,rect=[0,0,4,1])
xtitle("reflection coefficient versus normalized
   frequency for quarter wave transformer","f/fo","
   tao(reflection coefficient)")
```

check Appendix AP 2 for dependency:

smith_chart_tao.sci

Scilab code Exa 3.7 program to determine unknown load impedence

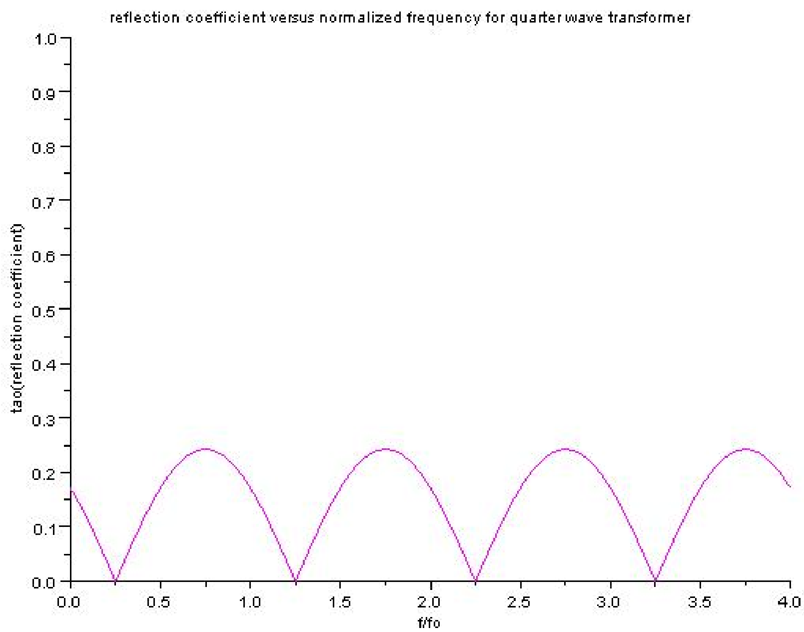

Figure 3.3: program to find out characteristic impedence

```
// example:-3.7,page no.-101.
// NOTE:-this example is a method for calculating unknown load impedence of slotted line section. all data are given and preassumed.
// program to determine unknown load impedence.
Zl=0;Zo=50; // for short circuitting the load.
SWR=%inf;
// short circuit is removed and replace with unknown load.
SWR=1.5;lamda=0.04;
lmin=4.2-2.72;
tao=(1.5-1)/(1.5+1);
theta=(%pi+((4*%pi)/4)*1.48);
tao=abs(tao)*exp(%i*theta);
Zl=50*((1+tao)/(1-tao));
// result
disp(Zl,'load impedence = ')
smith_chart(tao)
// when analyse with the help of smith chart.see the angle from x=0 axis i.e Tao_real axis.if it is above this axis take angle anticlockwise and if it is below this axis.take angle clockwise from Tao_real axis below.
```

Scilab code Exa 3.8 program to calculate attenuation constant

```
// example:-3.8,page no.-108.
// program to calculate attenuation constant.
syms alpha R Rs L G C eta a b w pi eipsila eipsilac mue eta;
//from example 3.1:-alpha=(R*(sqrt(C/L)+G*sqrt(L/C));
eta=sqrt(mue/eipsila);
```

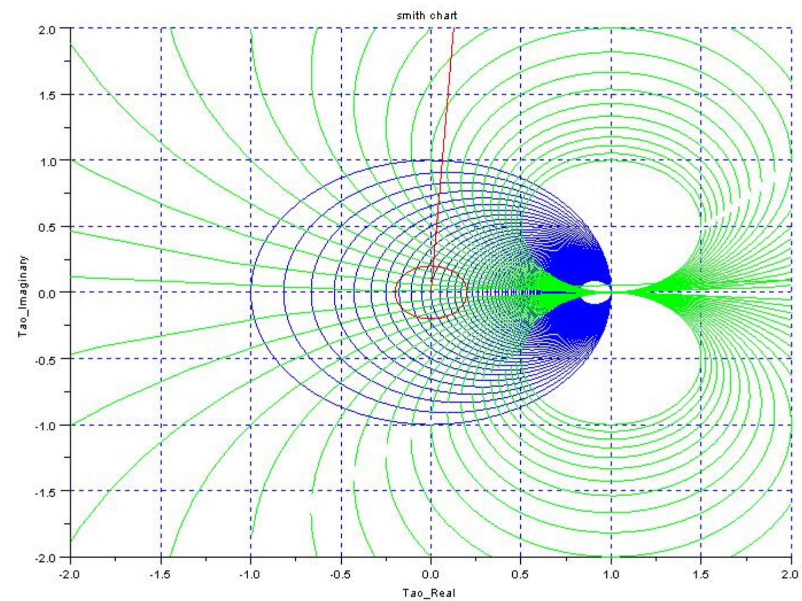

Figure 3.4: program to determine unknown load impedence

```
L=(mue/(2*pi))*(log(b/a));
C=(2*pi*eipsila)/log(b/a);
R=(Rs/(2*pi))*((1/a)+(1/b));
G=(2*pi*w*eipsilac)/log(b/a);
alpha=(1/2)*(R*sqrt(C/L)+G*sqrt(L/C));
disp(alpha,'attenuation constant = ')
```

Scilab code Exa 3.9 program to find the attenuation constant

```
// example:-3.9,page no.-111.
// program to find ht eattenuation constant of
   coaxial line.
syms E H Vo Zo P a b B z pi Po Q Rs Plc alpha Pld
   Plc w eipsila;
//Zo=(eta/(2*pi))*log(b/a);
E=(Vo/(P*(log(b)-log(a))))*exp(-%i*B*z);    //B=beta.
H=(Vo/(2*pi*P*Zo))*exp(-%i*B*z);
H=conj(H)*P; // for defining E cross H*.
Po=(1/2)*integ(integ((E*H),P),Q);
Po=limit(Po,P,b)-limit(Po,P,a);
Po=limit(Po,Q,2*pi)-limit(Po,Q,0);
disp(Po,'power flowing on the lossless line = ')
H=(H*conj(H))/P;  // for defining |H|^2;
Plc=(Rs/2)*integ(integ(H,z),Q);
Plc=limit(Plc,P,b)+limit(Plc,P,a);
Plc=limit(Plc,z,1)-limit(Plc,z,0);
Plc=limit(Plc,Q,2*pi)-limit(Plc,Q,0);
disp(Plc,'conductor loss = ')
E=E*conj(E)*P;
Pld=((w*eipsila)/2)*integ(integ(integ(E,P),Q),z);
Pld=limit(Pld,P,b)-limit(Pld,P,a);
Pld=limit(Pld,z,1)-limit(Pld,z,0);
Pld=limit(Pld,Q,2*pi)-limit(Pld,Q,0);
disp(Pld,'dielectric loss = ')
alpha=(Pld+Plc)/(2*Po); // attenuation constant.
```

```
disp(alpha,'attenuation constant = ')
```

Scilab code Exa 3.10 program to calculate attenuaton

```
// example:-3.10,page no.-114.
// program to calculate attenuaton due to conductor
   loss of a coaxial line using incremental
   inductance rule.
syms Zo eta pi a b Rs l alpha alpha_c alpha_dash
   delta alpha_c_dash sigma w mue;
sd=sqrt(2/(w*mue*sigma))
Zo=(eta*log(b/a))/(2*pi);
alpha_c=(Rs/(4*Zo*pi^2))*(diff(log(b/a),b)-diff(log(
   b/a),a));
disp(alpha_c,'attenuation due to conductor loss = ')
alpha_c_dash=alpha_c*(1+((2/pi)*atan((1.4*delta)/sd)
   ));
disp(alpha_c_dash,'attenuation corrected for surface
    roughness = ')
```

Chapter 4

TRANSMISSION LINE AND WAVEGUIDES

Scilab code Exa 4.1 program to find the cut off frequency

```
// example:-4.1,page no.-148.
// program to find the cut off frequency fo the first four propagating modes.
a=0.02286;b=0.01016;f=10*10^9;k=209.44;sigma=5.8*10^7;mue=4*%pi*10^-7;
c=3*10^8;
m=0;n=1;
fc=(c/(%pi*2))*sqrt(((%pi*m)/a)^2+((%pi*n)/b)^2);
fc=fc/(10^9);
disp(fc,'cut-off frequency for TE01 mode in GHZ=')
m=1;n=0;
fc=(c/(%pi*2))*sqrt(((%pi*m)/a)^2+((%pi*n)/b)^2);
fc=fc/(10^9);
disp(fc,'cut-off frequency for TE10 mode in GHZ=')
m=2;n=0;
fc=(c/(%pi*2))*sqrt(((%pi*m)/a)^2+((%pi*n)/b)^2);
fc=fc/(10^9);
disp(fc,'cut-off frequency for TE20 mode in GHZ=')
m=1;n=1;
```

```
fc=(c/(%pi*2))*sqrt(((%pi*m)/a)^2+((%pi*n)/b)^2);
fc=fc/(10^9);
disp(fc,'cut-off frequency for TE11 mode in GHZ=')
B=sqrt(k^2-(%pi/a)^2)  // for TE10 mode
Rs=sqrt(((2*%pi*f)*mue)/(2*sigma));   // surface resistance.
disp(Rs,'surface resistance in ohm=')
ac=(Rs/(a^3*b*B*k*377))*((2*b*%pi^2)+(a^3*k^2))  // attenuation constant.
ac=-20*(-ac)*log10(%e);
disp(ac,'attenuation constant in dB/m=')
```

Scilab code Exa 4.2 program to find the cut off frequency

```
//example:-4.2,page no.-160.
//program to find the cut off frequency of two propagating modes of a circular waveguide.
a=0.005;eipsilar=2.25;f=13*10^9;c=3*10^8;d=0.001;sigma=6.17*01^7;muo=4*%pi*10^-7;
m=1;n=1;
p11=1.841;p01=2.405;
fc=(p11*c)/(2*%pi*a*sqrt(eipsilar));
kc=p11/a;
fc=fc/(10^9);
disp(fc,'cut-off frequency for TE11 mode in GHZ')
m=0;n=1;
fc=(p01*c)/(2*%pi*a*sqrt(eipsilar));
fc=fc/(10^9);
disp(fc,'cut-off frequency for TE01 mode in GHZ')
// so,TE01 can't be propagating mode.only TE11 will be.
k=(2*%pi*f*sqrt(eipsilar))/c;
disp(k,'k in m-1=')
B=sqrt(k^2-kc^2);
disp(B,'propagation constant of TE11 mode')
```

```
ac=(k^2*d)/(2*B);
Rs=sqrt((2*%pi*f*muo)/(2*sigma));  // surface resistance.
acm=(Rs/(a*k*377*B))*(kc^2+((k^2)/(p11^2-1)));
a=ac+acm;
a=-20*(-0.547*0.5)*log10(%e);
disp(a,'total attenuation factor in dB=')
```

Scilab code Exa 4.3 program to find out the highest usable frequency

```
//example:-4.3,page no.-167.
//program to find out the highest usable frequency.
a=0.000889;b=0.0029464;eipsilar=2.2;c=3*10^8;
// here (b/a)=3.3,so for this kc*a=0.47
kc=0.47/a;
fc=(c*kc)/(2*%pi*sqrt(eipsilar))
fc=fc/(10^9);
fmax=0.95*fc;
disp(fmax,'maximum usable frequency in GHZ=')
```

Scilab code Exa 4.4 program to calculate and plot propagation constant

```
//example:-4.4,page no.-175.
// program to calculate and plot the propagation constant of first three propagating surface wave mode.
eipsilar=2.55;c=3*10^8;    // x=d/lamdao;
x=0.001:0.01:1.2;
for n=0:1:4
y=sqrt(eipsilar-((n^2)./(4.*(x^2)*(eipsilar-1))));  // y=beta/lamdao;
plot2d(x,y,style=2,rect=[0,0,1.2,1.6])
end
```

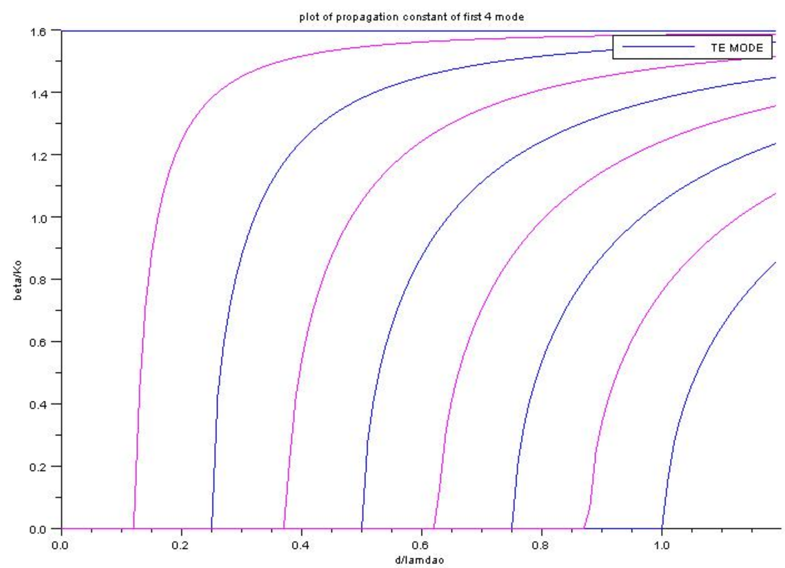

Figure 4.1: program to calculate and plot propagation constant

```
x=0.001:0.01:1.2;
for n=1:1:4
  y=sqrt(eipsilar-((((2.*n)-1)^2)./(16.*(x^2)*(
     eipsilar-1))))
  plot2d(x,y,style=6,rect=[0,0,1.2,1.6])
end
xtitle("plot of propagation constant of first 4 mode
   ","d/lamdao","beta/Ko");
legend("TE MODE")
```

Scilab code Exa 4.5 program to find width of a copper strip line

```
// example:-4.5,page no.-180.
```

```
// program to find width of a copper strip line conductor.
eipsilar=2.20;Zo=50;b=0.0032;d=0.001,f=10^10;t=0.00001;
c=3*10^8;Rs=0.026;A=4.74;
x=(30*%pi)/(sqrt(eipsilar)*Zo);
x=x-0.441;
w=b*x;
if ((sqrt(eipsilar)*Zo)<120)
    disp("width of copper strip line conductor is 0.00266m")
end
K=(2*%pi*f*sqrt(eipsilar))/c;
ad=(K*d)/2;
ac=(2.7*(10^-3)*Rs*eipsilar*Zo*A)/(30*%pi*(b-t));
a=ac+ad;
a=20*a*log10(%e);
lamda=c/(sqrt(eipsilar)*f);
alamda=lamda*a;
disp(K,'wave number=')
disp(ad,'dielectric aattenuation=')
disp(ac,'conductor attenuation=')
disp(a,'total attenuation constant=')
disp(alamda,'attenuation in dB/lamda=')
```

Scilab code Exa 4.7 program to calculate width and length

```
//example:-4.7,page no.-187.
//program to calculate the width and length of microstrip line.
eipsilae=1.87;//effective dielectric constant.
Zo=50;q=%pi/2;c=3*10^8;
f=2.5*10^9;
ko=(2*%pi*f)/c;
d=0.00127;
```

```
eipsilar=2.20;
// for w/d>2;
B=7.985;
w=3.081*d*100;
disp(w,'width in centi meter=')
l=(q*100)/(sqrt(eipsilae)*ko);
disp(l,'length of microstrip in centi meter=')
```

Scilab code Exa 4.9 program to calculate the group velocity

```
//example:-4.9,page no.-197.
//program to calculate the group velocity.
syms w c v;
B=sym('B');
ko=sym('ko');
kc=sym('kc');
ko=w/c;
B=sqrt(ko^2-kc^2);
v=diff(B,w);
vg=v^(-1);
vg=(c*B)/ko;
vp=w/B;
disp(vg,'group velocity=')
disp(vp,'phase velocity=')
disp('conclusion:-since B<ko,we have that vg<c<vp,
    which indicates that the phase velocity of a
    waveguide mode may be greater than the speed of
    light.but the group velocity will be lesser than
    the speed of light.')
```

Chapter 5

MICROWAVE NETWORK ANALYSIS

Scilab code Exa 5.1 program to find equivalent voltages and current

```
// example:-5.1,page no.-209.
//program to find the equivalent voltages and
   current.
syms a b A Zte V I C1 C2 P;
P=(a*b*A^2)/(4*Zte);
c=(1/2)*V*I;
d=(1/2)*(A^2)*C1*C2;
C1=sqrt((a*b)/2);  // on comparision.
C2=sqrt((a*b)/2)*Zte;  // on comparision.
c=[C1 C2];
disp(c)
disp("which completes the transmission line
   equivalence for the TE10 mode.")
```

Scilab code Exa 5.2 program to compute reflection coefficient

```
//example:-5.2,page no.-212.
//program to compute reflection coefficient.
a=0.03485;b=0.01580;eipsilao=8.854*10^-12;muo=4*%pi
    *10^-7;
f=4.5*10^9;
w=2*%pi*f;  // angular frequency.
// for z<0 region air filled.
eipsilar=2.56;  //for z>0 region.
ko=w*sqrt(muo*eipsilao);
k=ko*sqrt(eipsilar);
Ba=sqrt(ko^2-(%pi/a)^2); // propagation constant in
    air region z<0.
Bd=sqrt(k^2-(%pi/a)^2);  // propagation constant in
    dielectric region z>0.
Zoa=(ko*377)/Ba;
Zod=(ko*377)/Bd;
tao=(Zod-Zoa)/(Zod+Zoa);
disp(tao,'reflection coefficient')
```

Scilab code Exa 5.3 program to find z parameter of two port network

```
//example:-5.3,page no.-220.
// program to find the z parameter of the two port
    network.
syms Z11 Z12 Z22 Z21 V1 I1 V2 I2 Za Zb Zc;
Z11=Za+Zc;  // for I2=0.
Z12=(Zc/(Zb+Zc))*(Zb+Zc);  //for I1=0.
Z21=(Zc/(Za+Zc))*(Za+Zc);  // for I2=0.
Z22=Zb+Zc;   //for I1=0.
Z=[Z11 Z12;Z21 Z22];  // z-parameter matrix.
disp(Z,'Z-parameter of two port network = ')
```

Scilab code Exa 5.4 program to find the s parameter of 3 dB attenuator

```
// example:-5.4,page no.-221.
// program to find the s-parameter of 3-dB
   attenuator circuit.
Za=8.56;Zb=8.56,Zc=141.8;Zo=50;
S11=(((((Zo+Zb)*Zc)/(Zo+Zb+Zc))+Za)-Zo)/(((((Zo+Zb)*
   Zc)/(Zo+Zb+Zc))+Za)+Zo);  // reflection
   coefficient seen at port 1.
S22=(((((Zo+Za)*Zc)/(Zo+Za+Zc))+Zb)-Zo)/(((((Zo+Za)*
   Zc)/(Zo+Za+Zc))+Zb)+Zo);  // reflection
   coefficient seen at port 2.
S12=(((1/((((Zo+Za)*Zc)/(Zo+Za+Zc))+Zb))*(((Zo+Za)*
   Zc)/(Zo+Za+Zc)))*(Zo/(Zo+Za)));  // transmission
   coefficient from port 2 to 1.
S21=(((1/((((Zo+Zb)*Zc)/(Zo+Zb+Zc))+Za))*(((Zo+Zb)*
   Zc)/(Zo+Zb+Zc)))*(Zo/(Zo+Zb)));  // transmission
   coefficient from port 1 to 2.
S=[S11 S12;S21 S22];  // s-parameter matrix.
disp(S,'S-parameter of 3-db attenuator circuit is ='
   )
```

Scilab code Exa 5.5 program to determine reciprccity and losslessness

```
//example:-5.5,page no.-226.
//program to determine the reciprccity and lossless
   of two port network and find return loss.
syms S Rl tao;
S=[0.1 0.8*%i;0.8*%i 0.2]; // s-parameter matrix.
if (S(1,2)==S(2,1))
  disp("the network is reciprocal.")
else
  disp("the network is not reciprocal.")
end
if (S(1,1)^2+S(1,2)^2==1)
  disp("the network is lossless.")
else
```

```
  disp("the network is lossy.")
end
tao=S(1,1)-(S(1,2)*S(2,1))/(1+S(2,2));  //input
    reflection coefficient.
Rl=-20*log10(abs(tao));  // return loss in dB.
//result
disp(Rl,'return loss at port 1 in dB=')
```

Scilab code Exa 5.6 program to find ABCD parameter of two port network

```
// example:-5.6,page no.-232.
//program to find the ABCD parameter of a two-port
    network.
syms A B C D V1 V2 I1 I2 Z;
//A=V1/V2;  // for i2=0;
A=1;
B=V1/(V1/Z);
C=0;
D=I1/I1;
ABCD=[A B;C D];
// result
disp(ABCD,'abcd parameter')
```

Scilab code Exa 5.7 program to find admittance matrix for bridge T

```
// example:-5.7,page no.-238.
// program to find the admittance matrix for bridge-
    T network.
syms Za Z1 Z2 Z3 Y Ya Yb D;
Za=[Z1+Z2 Z2;Z2 Z1+Z2];
Yb=[1/Z3 -1/Z3;-1/Z3 1/Z3];
Y1=1/Z1;Y2=1/Z2;
```

```
Ya=1/Za;
Y=Ya+Yb;
D=((Z2+Z1)^2-Z2^2);
// result
disp(Y,'admittance matrix for bridge-T network=')
```

Scilab code Exa 5.8 program to compute power gains

```
// example:-5.8,page no.-243.
//program to compute power gains.
f=10^10;Zs=20;Zl=30;Zo=50;
S=[-0.39+%i*0.225 0.009848+%i*-0.001736;2.02+0.356*%i -0.3464-%i*0.2];
taos=(Zs-Zo)/(Zs+Zo);
taol=(Zl-Zo)/(Zl+Zo);
taoin=S(1,1)+((S(1,2)*S(2,1)*taol)/(1-S(2,2)*taol));
taoout=S(2,2)+((S(1,2)*S(2,1)*taos)/(1-S(1,1)*taos));
Ga=(abs(S(2,1)^2)*(1-abs(taos)^2))/((abs(1-S(1,1)*taos)^2)*(1-abs(taoout)^2));
Gt=(abs(S(2,1)^2)*(1-abs(taos)^2)*(1-abs(taol)^2))/((abs(1-S(2,2)*taol)^2)*abs(1-taos*taoin)^2);
G=(abs(S(2,1)^2)*(1-abs(taol)^2))/((abs(1-S(2,2)*taol)^2)*(1-abs(taoin)^2));
disp(G,'actual power gain=')
disp(Ga,'the available power gain=')
disp(Gt,'the transducer power gain=')
disp(taoin,'reflection coefficient looking at port 1=')
disp(taoout,'reflection coefficient looking at port 2=')
disp(taos,'reflection coefficient at the source=')
disp(taol,'reflection coefficient at the load=')
```

Scilab code Exa 5.9 program to derive the expression for taoin

```
// example:-5.9,page no.-248.
// program to derive the expression for taoin.
syms S S11 S22 S12 S21 taol taoin a1 a2 b1 b2 a b;
S=[S11 S12;S21 S22];
b=[b1;b2];
a=[a1;a2];
b=S*a;
disp(b)
//so, S11 will be the reflection coefficient i.e
    taoin.
taoin=S11+((S21*S12*taol)/(1-S22*taol));
// result
disp(taoin,'the expression for taoin will be=')
```

Scilab code Exa 5.10 program to find out expression for taoin

```
// example:-5.10.page no.-250.
//program to find out expression for taoin.
syms P1 P2 S11 S22 S12 S21 taol taoin L1 l2;
P1=S11; // path one.
P2=S21*S12*taol;  //path second.
L1=taol*S22; // loop gain for path 1.
L2=L1^2;  // loop gain taking two at a time.(but
    only one loop wiil exist.i.e=L1.)
L2=0;
// from mason's gain formula.
taoin=(S11*(1-taol*S22)+(S21*taol*S12))/(1-taol*S22)
    ;
// result
disp(taoin)
```

Scilab code Exa 5.11 determine amplitude of forward and backward wave

```
// example:-5.11,page no.-264.
// program to determine the amplitude of the forward and backward travelling TE10 modes and the input resistance.
syms Io a b x y z h1 e1 J P1 A1p A1m pi Z1 delta b1 j P Rin;
e1=sin(pi*x/a);  // in y direction.
h1=-sin(pi*x/a)/Z1; // in z direction.
P1=(2/Z1)*integ(integ(sin(pi*x/a)^2,x),y);
P1=limit(P1,x,a)-limit(P1,x,0);
P1=limit(P1,y,b)-limit(P1,y,0);
// taking sin(2*pi)=0. we get,
P1=a*b/Z1;
A1p=(-1/P1)*Io*y; // as for x,it will be one at x=a/2 and 1 for z at z=0;
A1p=limit(A1p,y,b)-limit(A1p,y,0);
A1m=(-1/P1)*Io*y; // as for x,it will be one at x=a/2 and 1 for z at z=0;
A1m=limit(A1m,y,b)-limit(A1m,y,0);
P=integ(integ(((A1p^2)/Z1)*sin(pi*x/a)^2,x),y);
P=limit(P,x,a)-limit(P,x,0);
P=limit(P,y,b)-limit(P,y,0);
// taking sin(2*pi)=0. we get,
P=(b*(Io^2)*Z1*pi)/(2*a*pi);
Rin=2*P/(Io^2);
disp(A1p,'amplitude of the forward travelling wave = ')
disp(A1m,'amplitude of the backward travelling wave = ')
disp(Rin,'input resistance seen by the probe = ')
```

Scilab code Exa 5.12 find excitation coefficient of forward wave TE10

```
// example:-5.12,page no.-265.
// program to find the excitation coefficient of the
    forward travelling TE10 mode.
syms M Pm uo w j a b Io x y z ro pi Z1 h1 A1p P1 no
   ko uo eo;
ko=w*sqrt(uo*eo);
no=sqrt(uo/eo);
h1=sin(pi*x/a)*(-1/Z1); // in x direction.
P1=(2/Z1)*integ(integ(sin(pi*x/a)^2,x),y);
P1=limit(P1,x,a)-limit(P1,x,0);
P1=limit(P1,y,b)-limit(P1,y,0);
// taking sin(2*pi)=0. we get,
P1=a*b/Z1;
Pm=Io*pi*(ro^2);  // defined at x=a/2,y=b/2 and z=0;
M=j*w*uo*Pm;
A1p=(1/P1)*(-(1/Z1)*M);
disp(A1p,'the forward wave excitation coefficient
    will be = ')
disp("!! NOTE:-replace w=sqrt(uo*eo) and no=sqrt(uo/
   eo),the answer will match !! ")
disp(" NOTE:- on integrating, x component will
   become one at x=a/2,y component will become one
   at y=b/2 and z component will become one at z=0."
   )
```

Chapter 6

IMPEDENCE MATCHING AND TUNNING

Scilab code Exa 6.1 program to design an L section matching network

```
// example:-6.1,page no.-284.
// program to design an L section matching network to match a series RC load.
Zl=200-%i*100; // load impedence.
Rl=200;Xl=-100;f=500*10^6;Zo=100;
B1=(Xl+sqrt(Rl/Zo)*sqrt(Rl^2+Xl^2-(Rl*Zo)))/(Rl^2+Xl^2);
B2=(Xl-sqrt(Rl/Zo)*sqrt(Rl^2+Xl^2-(Rl*Zo)))/(Rl^2+Xl^2);
C1=(B1/(2*%pi*f))*10^12;
L2=(-1/(B2*2*%pi*f))*10^9;
X1=(1/B1)+((Xl*Zo)/Rl)-(Zo/(B1*Rl));
X2=(1/B2)+((Xl*Zo)/Rl)-(Zo/(B2*Rl));
L1=(X1/(2*%pi*f))*10^9;
C2=(-1/(X2*2*%pi*f))*10^12;
disp(L1,'inductor of first circuit in nH = ')
disp(C1,'capacitor of the first circuit in pF = ')
disp(L2,'inductor of second circuit in nH = ')
disp(C2,'capacitor of the second circuit in pF = ')
```

```
disp("NOTE:-for above specific problem Rl>Zo, positive X implies inductor,negative X implies capacitor,positive B implies capacitor and negative B implies inductor.")
```

Scilab code Exa 6.5 design quarter wave matching transformer

```
//example:-6.5,page no.-304.
//program to design a single section quarter wave matching transformer.
Zl=10;  // load impedence.
Zo=50; // characteristic impedence.
fo=3*10^9;swr=1.5; // maximum limit of swr.
Z1=sqrt(Zo*Zl); // characteristic impedence of the matching section.
taom=(swr-1)/(swr+1);
frac_bw=2-(4/%pi)*acos((taom/sqrt(1-taom^2))*(2*sqrt(Zo*Zl)/abs(Zl-Zo)));  // fractional bandwidth.
disp(Z1,'charecteristic impedence of matching section = ')
disp(frac_bw,'fractional bandwidth = ')
```

Scilab code Exa 6.6 program to evaluate the worst case percent error

```
// example:-6.6,page no.-307.
// program to evaluate the worst case percent error in computing magnitude of reflection coefficient.
Z1=100;Z2=150;Zl=225;
tao_1=(Z2-Z1)/(Z2+Z1);
tao_2=(Zl-Z2)/(Zl+Z2);
tao_exact=(tao_1+tao_2)/(1+tao_1*tao_2); // this results as angle is taken zero.
```

```
tao_approx=tao_1+tao_2;  // this results as angle is taken zero.
eror=abs(((tao_exact-tao_approx)/tao_exact)*100);
disp(tao_approx,'approximate value of reflection coefficient is = ')
disp(eror,'the error in percent is about = ')
```

Scilab code Exa 6.7 design three section binomial transformer

```
// example:-6.7,page no.-312.
// program to design three section binomial transformer.
Zl=50;Zo=100;N=3;taom=0.05;
A=(2^-N)*abs((Zl-Zo)/(Zl+Zo));
frac_bw=2-(4/%pi)*acos(0.5*(taom/A)^2);
for c=1
  Z1=Zo*((Zl/Zo)^((2^-N)*(c^N)));
disp(Z1,'Z1 = ')
end
for c=3^(1/3)
  Z2=Z1*((Zl/Zo)^((2^-N)*(c^N)));
  disp(Z2,'Z2 = ')
end
for c=3^(1/3)
  Z3=Z2*((Zl/Zo)^((2^-N)*(c^N)));
  disp(Z3,'Z3 = ')
end
```

Scilab code Exa 6.8 design three section chebysev transfomer

```
// example:-6.8,page no.-316.
// program to design a three section chebysev transformer.
```

```
Zl=100;Zo=50;taom=0.05;N=3;A=0.05;
thetam=asec(cosh((1/N)*acosh((1/taom)*abs((Zl-Zo)/(Zl+Zo)))))*(180/%pi);
x=(cosh((1/N)*acosh((1/taom)*abs((Zl-Zo)/(Zl+Zo)))))
tao_o=A*(x^3)/2;
tao_1=(3*A*(x^3-x))/2;  // from symmetry tao_3=tao_0;
Z1=Zo*((1+tao_o)/(1-tao_o));
Z2=Z1*((1+tao_1)/(1-tao_1));
Z3=Zl*((1-tao_o)/(1+tao_o));
disp(Z1,Z2,Z3,'the characteristic impedences are = ')
```

Scilab code Exa 6.9 design triangular taper and a klopfenstein taper

```
//example:-6.9,page no.-323.
//program to designa triangular taper and a klopfenstein taper.
taom=0.02;Zl=50;Zo=100;
tao_o=0.5*log(Zl/Zo);
A=acosh(tao_o/taom);
A=real(A);
disp(tao_o,'tao_o = ')
disp(A,'A = ')
```

Chapter 7

MICROWAVE RESONATORS

Scilab code Exa 7.1 program to compare the Q factor

```
// example:-7.1,page no.-339.
//program to compare the Q of an air filled and
    teflon filled coaxial line resonator.
sigma=5.813*10^7;muo=4*%pi*10^-7;f=5*10^9;eta=377;a
    =1*10^-3;b=4*10^-3;
omega=2*%pi*f;ko=104.7;B=104.7;alpha=0.022;
Rs=sqrt((omega*muo)/(2*sigma));
alphaca=(Rs/(2*eta*log(b/a)))*((1/a)+(1/b)); //
    attenuation due to conductor loss for air filled
    line.
eipsilar=2.08;tandelta=0.0004;  // for teflon filled
     line.
alphact=((Rs*sqrt(2.08)*0.01)/(2*eta*log(b/a)))*((1/
    a)+(1/b)); // attenuation due to conductor loss
    for teflon filled line.
alphada=0; // for air filled line.
alphadt=ko*(sqrt(eipsilar)/2)*tandelta;
Qair=B/(2*alpha);
B=B*sqrt(eipsilar);
alpha=0.062;
Qteflon=B/(2*alpha);
```

```
disp(Qair,'Qair = ')
disp(Qteflon,'Qteflon = ')
disp("conclusion:-Qair is almost twice that of Qteflon")
```

Scilab code Exa 7.2 program to compute length and Q of the resonator

```
//example:-7.2,page no.-342.
// program to compute the length of the line for resonance at 5 GHZ and the Q of the resonator.
W=0.0049;c=3*10^8;f=5*10^9;Zo=50;eipsilar=2.2;ko=104.7;tandelta=0.001;
Rs=0.0184; // taken from example 7.1.
eipsilae=1.87;  // effective permittivity.
l=c/(2*f*sqrt(eipsilae)); // resonator length.
B=(2*%pi*f*sqrt(eipsilae))/c;
alphac=Rs/(Zo*W);
alphad=(ko*eipsilar*(eipsilae-1)*tandelta)/(2*sqrt(eipsilae)*(eipsilar-1));
alpha=alphac+alphad;
Q=B/(2*alpha);
disp(l,'length of the line in meter = ')
disp(Q,'Q of the resonator = ')
```

Scilab code Exa 7.3 program to find required length and other

```
// example:-7.3,page no.-347.
// program to find required length,d and Q for l=1 and l=2 resonator mode.
a=0.04755;b=0.02215;eipsilar=2.25;tandelta=0.0004;f=5*10^9;c=3*10^8;
k=(2*%pi*f*sqrt(eipsilar))/c // wave number.
for l=1:1:2
```

```
d=(l*%pi)/sqrt((k^2)-((%pi/b)^2)); // m=1 & n=0 mode
    .
disp(d,'d in meter = ')
end
eta=377/sqrt(eipsilar);
Qc1=3380;// l=1.
Qc2=3864;// l=2.
Qd=2500;  // Q due to dielectric loss only.
Q1=((1/Qc1)+(1/Qd))^-1; // for l=1.
Q2=((1/Qc2)+(1/Qd))^-1; // for l=2.
disp(Q1,'Q1 = ');
disp(Q2,'Q2 = ')
```

Scilab code Exa 7.4 program to find dimension and Q

```
//example:-7.4,page no.-353.
// program to find dimension and Q;
f=5*10^9;c=3*10^8;p01=3.832;sigma=5.813*10^7;muo=4*
    %pi*10^-7;
eipsilar=2.25;
// mode TE011. and d=2a.
omega=2*%pi*f;
eta=377;
lamda=c/f;
k=(2*%pi)/lamda;
// f=(c/(2*%pi))*sqrt((p01/a)^2+(%pi/(2*a))^2); as d
    =2a given
a=sqrt((p01)^2+(%pi/2)^2)/k;
Rs=sqrt((omega*muo)/(2*sigma))
Qc=(k*a*eta)/(2*Rs); // for m=l=1,n=0 and d=2a.
disp(a,'a in meter = ')
disp(Qc,'Qc = ')
```

Scilab code Exa 7.5 program to find resonant frequency and Q

```
//example:-7.5,page no.-358.
// program to find the resonant frequency and Q for
    TE01delta mode.
delta=0.001;eipsilar=95;a=0.413;L=0.008255;c=3*10^8;
//tan((B*L)/2)=alpha/beta.
//ko=(2*%pi*f)/c;
alpha=sqrt((2.405/a)^2-(ko)^2);
B=sqrt((eipsilar*(ko)^2)-(2.405/a)^2); // beta
f1=((c*2.405)/(2*%pi*sqrt(eipsilar)*a))*10^-7;
f2=((c*2.405)/(2*%pi*a))*10^-7;
disp(f1,'f1 in GHZ = ')
disp(f2,'f2 in GHZ = ')
Q=1/tan(delta);
disp(Q,'approx. value of Q due to dielectric loss =
    ')
```

Scilab code Exa 7.6 program to find the mode number and Q

```
// example:-7.6,page no.-361.
// program to find the mode number and Q of given
    resonator.
fo=94*10^9;d=0.04;c=3*10^8;muo=4*%pi*10^-7;sigma
    =5.813*10^7;
l=(2*d*fo)/c; // mode number.
Rs=sqrt((2*%pi*fo*muo)/(2*sigma));
Q=(%pi*l*377)/(4*Rs);
disp(l,'mode number = ')
disp(Q,'Q = ')
```

Scilab code Exa 7.7 program to find value of the coupling capacitor

```
// example:-7.7,page no.-367.
// program to find the value of the coupling
   capacitor required for critical coupling.
l=0.02175;Zo=50;eipsilae=1.9;c=3*10^8;
fo=c/(2*l*sqrt(eipsilae)); // first resonant
   frequency will occur when the resonator ia about
   l=lamdag/2 in length.
lamdag=c/fo;
alpha=1/8.7; // in Np/m.
Q=%pi/(2*l*alpha);
bc=sqrt(%pi/(2*Q));
C=bc/(2*%pi*fo*Zo)*10^12;
disp(C,'coupling capacitor in pF = ')
C=bc/(2*%pi*fo*Zo);
w1=atan(2*%pi*fo*C*Zo)*c/(l*sqrt(eipsilae));   //
   from equation tan(B*l)=-bc;
w1=w1*10^-8;
disp(w1,'frequency in GHZ = ')
```

Scilab code Exa 7.8 derive expression for change in resonant frequency

```
// example:-7.8,page no.-373.
// program to derive an expression for the change in
    resonant frequency.
syms Ey Hx Hz A Zte n a pi x z d j k t y er eo c wo
   w b;
Ey=A*sin((pi*x)/a)*sin((pi*z)/d);
Hx=((-j*A)/Zte)*sin((pi*x)/a)*cos((pi*z)/d);
Hz=((j*pi*A)/(k*n*a))*cos((pi*x)/a)*sin((pi*z)/d);
Ey=Ey^2;
//c=(er-1)*eo;
w=c*integ(integ(integ(Ey,z),y),x);
w=limit(w,z,d)-limit(w,z,0);
w=limit(w,y,t)-limit(w,y,0);
w=limit(w,x,a)-limit(w,x,0);
```

```
disp(w)
// as sin(2*pi)=0;  then last term of above result
   will be:-
w=(c*A^2*a*t*d)/4;
disp(w,'on taking sin(2*pi)=0  , w becomes = ')
wo=((a*b*d*eo)/2)*A^2;
deltaw=(w-wo)/wo;
disp(deltaw,'fractional change in resonant frequency
   = ')
```

Scilab code Exa 7.9 derive expression for change in resonant frequency

```
// example:-7.9,page no.-376.
// program to derive an expression for the change in
   resonant frequency.
syms Ey Hx Hz A Zte n a pi x z d j eo c wo w b l ro;
Ey=A*sin((pi*x)/a)*sin((pi*z)/d);
Hx=((-j*A)/Zte)*sin((pi*x)/a)*cos((pi*z)/d);
Hz=((j*pi*A)/(k*n*a))*cos((pi*x)/a)*sin((pi*z)/d);
Ey=A;// at x=a/2,y,z=d/2;
Hx=0;// at x=a/2,y,z=d/2;
Hz=0;// at x=a/2,y,z=d/2;
//where w is perturbed resonant frequency and wo is
   unperturbed resonant frequency.
w=-eo*A^2*pi*l*ro^2;
wo=(a*b*eo*d*A^2)/2;
deltaw=(w-wo)/wo;
disp(deltaw,'the perturbation in resonant frequency
   w.r.t wo = ')
```

Chapter 8

POWER DIVIDERS DIRECTIONAL COUPLERS AND HYBRIDS

check Appendix AP 1 for dependency:

parallel_impedence.sce

Scilab code Exa 8.1 program to compute the reflection coefficients

```
// example:-8.1,page no.-392.
// program to compute the reflection coefficients
   seen looking in to the output port.
// as the power is divided in to 2:1 ratio. and Pin
   =(1/2)*Vo^2/Zo;
// so,P1=(1/3)*Pin;and P2=(2/3)*Pin ............(i)
Zo=50;
Z1=3*Zo;  // from above condition............(i)
Z2=(3/2)*Zo;
Zin=parallel_impedence(Z1,Z2);  // input impedence
   to the junction.
if Zin==Zo
  disp("input is matched to the 50 ohm sources")
```

```
else
  disp("not matched")
end
Zin1=parallel_impedence(Zo,Z2); // looking in to the
    150 ohm source.
Zin2=parallel_impedence(Zo,Z1); // looking in to the
    75 ohm source.
tao1=(Zin1-Z1)/(Zin1+Z1);
tao2=(Zin2-Z2)/(Zin2+Z2);
disp(tao1,'reflection coefficient looking at 150 ohm
    line = ')
disp(tao2,'reflection coefficient looking at 75 ohm
    line = ')
```

Scilab code Exa 8.2 design equi split wilkinson power divider

```
//example:-8.2,page no.-398.
// program to design an equi-split wilkinson power
    divider for 50 ohm system impedence.
Zo=50;
Z=sqrt(2)*Zo; // impedence of quarter wave
    transmission line.
R=2*Zo; // shunt resistor.
disp(R,'the shunt resistance value should be in ohm
    = ')
disp(Z,'the quarter wave transmission line in the
    divide should have a characteristic impedence in
    ohm = ')
```

Scilab code Exa 8.3 design bethe hole coupler for x band waveguide

```
// example:-8.3,page no.-404.
```

```
// program to design bethe-hole coupler for x-band wave guide.
f=9*10^9;C=20;a=0.02286;b=0.01016;Ko=188.5;B=129;Z10=550.9;P10=4.22*10^-7;lamdao=0.0333;uo=4*%pi*10^-7;eo=8.854*10^-12;w=2*%pi*f;
s=(a/%pi)*asin(lamdao/sqrt(2*(lamdao^2-a^2)))*10^3;
// a=10*b;// as C=20db; // take x=a/b; so x=10;
ro=(P10/((10*w)*((((2*eo/3)+(4*uo)/(3*Z10^2))*0.944)-((4*%pi^2*uo*0.056)/(3*B^2*a^2*Z10^2)))))^(1/3)*10^3;
disp(s,'the aperture position in mm = ')
disp(ro,'the aperture size in mm = ')
disp("NOTE:-the above shown results completes the design of the betha hole coupler.")
```

Scilab code Exa 8.4 program to design a four hole chebysev coupler

```
//example:-8.4,page no.-410.
// program to design a four hole chebysev coupler in x-band wave guide using round aperture located at s=a/4.
a=0.02286;b=0.01016;lamdao=0.0333;ko=188.5;bta=129;Z10=550.9;P10=4.22*10^-7;f=9*10^9;no=377;N=3;
s=a/4;
kf=((2*ko)/(3*no*P10))*((sin(%pi*s/a)^2)-(2*(bta^2)/(ko^2))*((sin(%pi*s/a)^2)+((%pi^2)/((bta^2)*(a^2)))*(cos(%pi*s/a)^2)));
kf=abs(kf)
kb=((2*ko)/(3*no*P10))*((sin(%pi*s/a)^2)+(2*(bta^2)/(ko^2))*((sin(%pi*s/a)^2)-((%pi^2)/((bta^2)*(a^2)))*(cos(%pi*s/a)^2)));
kb=abs(kb)
x=cosh(acosh(100)/3);  // x=sec(thetam).
thetam=asec(x)*180/%pi; // so,thetam=70.6 and 109.4 at the band edge.
```

```
k=10^(-171.94/20);
ro=(((k/2)^(1/3))*x)*1000;
r1=(1.5*k*((x^3)-x))^(1/3)*1000;
disp(kf,'kf = ')
disp(kb,'kb = ')
disp(thetam,'thetam in degree = ')
disp(ro,'ro in mm = ')
disp(r1,'r1 in mm = ')
```

Scilab code Exa 8.5 design 50 ohm branchline quadrature hybrid junc

```
// example:-8.5,page no.-415.
// program to design a 50 ohm branch-line quadrature
   hybrid junction.
Zo=50;
Z=Zo/sqrt(2);
disp(Z,'the branch line impedence in ohm will be = '
   )
```

Scilab code Exa 8.6 determine even and odd mode characteristic impeden

```
// example:-8.6,page no.-419.
// program to determine  the even and odd mode
   characteristic impedence.
syms C A d W C11 C12 Ce Co v eo er s b uo Zoe Zoo
   eipsila;
C=A*eipsila/d;
C11=(eo*er*W)/((b-s)/2)+(eo*er*W)/((b+s)/2);
C12=er*eo*W/s;
Ce==C11;
Co=C11+2*C12
v=1/sqrt(er*eo*uo);
Zoe=1/(v*C11); // as Ce=C11;
```

```
Zoo=1/(v*Co);
disp(Zoe,'Zoe = ')
disp(Zoo,'zoo = ')
```

Scilab code Exa 8.7 design a 20 db single section coupled line coupler

```
// example:-8.7,page no.-425.
//design a 20 db single section coupled line coupler
    in stripline.
C=10^(-20/20);f=3*10^9;eipsila=2.56;Zo=50;b=0.00158;
Zoe=Zo*sqrt((1+C)/(1-C));
Zoo=Zo*sqrt((1-C)/(1+C));
Zoe=eipsila*Zoe;
Zoo=eipsila*Zoo;
x=0.72; //x=w/b.
y=0.34; // y=s/b.
w=0.72*b*100;
s=0.34*b*100;
disp(w,'conductor width in cm = ')
disp(s,'conductor seperation in cm = ')
```

Scilab code Exa 8.8 design a three section 20 db coupler

```
// example:-8.8,page no.-428.
// design a three section 20 db coupler with a
    binomial response.
Zo=50;f=3*10^9;N=3;
syms C C1 C2 theta;
C=10^(-20/20);
disp("for a maximally flat response for a three=
    section coupler doupble derivative of C will be
    zero.")
C1=0.0125;C2=0.125;C3=0.0125;
```

```
Zoe1=Zo*sqrt((1+C1)/(1-C1));
Zoe3=Zo*sqrt((1+C3)/(1-C3));
Zoo1=Zo*sqrt((1-C1)/(1+C1));
Zoo3=Zo*sqrt((1-C1)/(1+C1));
Zoe2=Zo*sqrt((1+C2)/(1-C2));
Zoo2=Zo*sqrt((1+C2)/(1-C2));
disp("the even and odd mode characteristic impedences for each section are = ")
disp(Zoe1,'Zoe1 = ')
disp(Zoo1,'Zoo1 = ')
disp(Zoe2,'Zoe2 = ')
disp(Zoo2,'Zoo2 = ')
disp(Zoe3,'Zoe3 = ')
disp(Zoo3,'Zoo3 = ')
```

Scilab code Exa 8.9 design a 3 dB 50 ohm langer coupler

```
//example:-8.9,page no.-434.
// program to design a 3 dB 50 ohm langer coupler for operation at 5 GHZ.
f=5*10^9;C=10^(-3/20);
Zo=50;
Zoe=(((4*C)-3+sqrt(9-(8*C^2)))/((2*C)*sqrt((1-C)/(1+C))))*Zo;
Zoo=(((4*C)+3-sqrt(9-(8*C^2)))/((2*C)*sqrt((1+C)/(1-C))))*Zo;
disp(Zoe,'even mode characteristic impedence of a pair of adjacent coupled lines is = ')
disp(Zoo,'even mode characteristic impedence of a pair of adjacent coupled lines is = ')
```

Scilab code Exa 8.10 design 180 deg ring hybrid for 50 ohm system imped

```
// example:-8.10,page no.-440.
// design a 180 deg. ring hybrid for a 50 ohm system
    impedence.
Zo=50;
Z=sqrt(2)*Zo;
disp(Z,'the characteristic impedence of the ring
    transmission line in ohm is = ')
```

Scilab code Exa 8.11 calculate even and odd mode characteristic impeden

```
// example:-8.11,page no.-444.
// calculate the even and odd-mode characteristic
    impedences for a tapered coupled line 180 deg.
    hybrid for a 3 db coupling ratio and a 50 ohm
    characteristic impedence.
alpha=0.707;bta=0.707;Zo=50;
k=(1-alpha)/(1+alpha);
Zoe=Zo/k;
Zoo=k*Zo;
disp(Zoo,'Zoo =  ')
disp(Zoe,'at (Z=L) the characteristic impedences of
    the coupled line must be = ')
disp('at Z=0,there will be no coupling')
```

Chapter 9

MICROWAVE FILTERS

Scilab code Exa 9.1 program to compute the propagation constant

```
// example:-9.1,page no.-462.
// program to compute the propagation constant,phase
    velocity and bloch impedence.
Co=2.666*10^-12;
d=0.01;c=3*10^8;
Zo=50;f=3*10^9;
p=(Co*Zo*c)/(2*d); // constant of equation given
    below.
y=0:0.001:0.96;
x=acos(cos(y)-p.*y.*sin(y)); // x=ko*d; and y=beta*d
    ;
subplot(2,1,1)
plot2d(x,y,style=2,rect=[-%pi,0,%pi,0.96])
plot2d(-x,y,style=2,rect=[-%pi,0,%pi,0.96])
xtitle("k-beta diagram for first pass band ","beta*d
    ","ko*d")
y=3:0.001:4;
x=acos(cos(y)-p.*y.*sin(y)); // x=ko*d; and y=beta*d
    ;
subplot(2,1,2)
plot2d(x,y,style=3,rect=[-%pi,3,%pi,4])
```

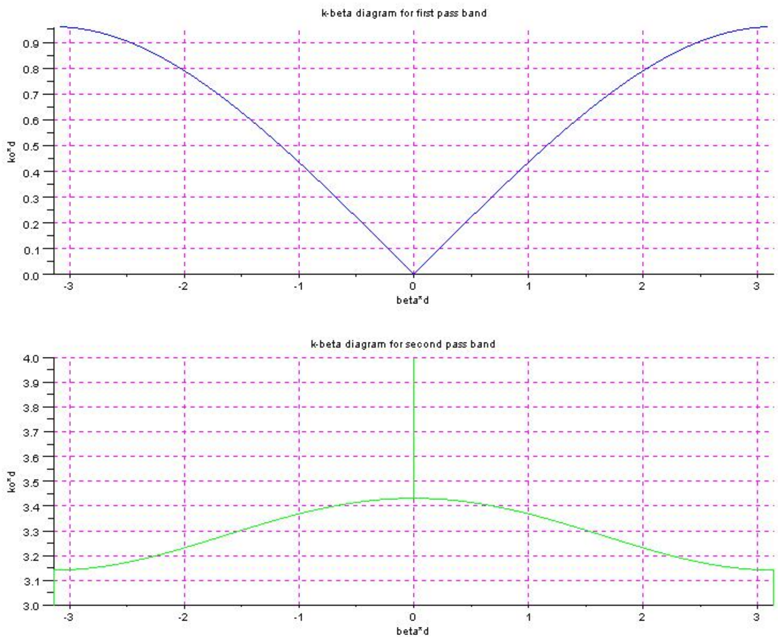

Figure 9.1: program to compute the propagation constant

```
plot2d(-x,y,style=3,rect=[-%pi,3,%pi,4])
xtitle("k−beta diagram for second pass band ","beta*d","ko*d")
bta=(acos(cos(ko*d)-p*ko*d*sin(ko*d)))/d;
ko=(2*%pi*f)/c;
vp=(ko*c)/150; // phase velocity.
b=2*%pi*f*Co*Zo;
A=cos(ko*d)-(b/2)*sin(ko*d);
B=%i*(sin(ko*d)+(b/2)*cos(ko*d)-(b/2));
Zb=(B*Zo)/sqrt(A^2-1); // bloch impedence.
disp(Zb,'Bloch impedence = ')
disp(vp,'phase velocity = ')
disp(bta,'propagation constant = ')
```

Scilab code Exa 9.2 program to design a low pass composite filter

```
// example:-9.2,page no.-473.
// program to design a low pass composite filter
   with cutoff frequency of 2 MHZ.
fc=2*10^6;f=2.05*10^6;Ro=75;
L=(2*Ro)/(2*%pi*fc);
C=2/(Ro*2*%pi*fc);
for m=sqrt(1-(fc/f)^2)
x=m*L/2;
y=m*C;
z=((1-m^2)/(4*m))*L;  // x,y,z are design parameter
   assumed.
disp(x,y,z,'design parameter for m=0.2195 ')
end
for m=0.6
 x=m*L/2;
 y=m*C/2;
 z=((1-m^2)/(2*m))*L;  // x,y,z are design parameter
    assumed.
 disp(x,y,z,'design parameter for m=0.6 ')
end
```

Scilab code Exa 9.3 program to find out number of filter elements

```
// example:-9.3,page no.-482.
// program to find out number of filter elements
   required.
fc=8*10^9;f=11*10^9;
w=2*%pi*f;
wc=2*%pi*fc;
x=abs(w/wc)-1;
```

```
disp(x,"from table we see that an attenuation of 20 db at this frequency requires that N>=8  for x = ")
```

Scilab code Exa 9.4 program to design a maximum flat low pass filter

```
// example:-9.4,page no.-488.
// program to design a maximum flat low pass filter with cut off frequency of 2 GHZ.
fc=2*10^9;f=3*10^9;
w=2*%pi*f;
wc=2*%pi*fc;
x=abs(w/wc)-1;
// from table we can see that N=5 will be sufficient.
// then prototype element values are:-
g1=0.618;g2=1.618;g3=2.000;g4=1.618;g5=0.618;
disp(g1,'g1 = ')
disp(g2,'g2 = ')
disp(g3,'g3 = ')
disp(g4,'g4 = ')
disp(g5,'g5 = ')
```

Scilab code Exa 9.5 program to design a band pass filter

```
// example:-9.5,page no.-492.
// design a band pass filter having a 0.5 db equal ripple respnse with N=3.
N=3;Zo=50;f=1*10^9;delta=1*10^8;
L1=1.596;L3=1.5963;C2=1.0967;Rl=1.000;
L_1=(L1*Zo)/(2*%pi*f*delta);
C_1=delta/(2*%pi*f*L1*Zo);
L_2=(delta*Zo)/(2*%pi*f*C2);
```

```
C_2=C2/(2*%pi*f*delta*Zo);
L_3=(L3*Zo)/(2*%pi*f*delta);
C_3=delta/(2*%pi*f*L3*Zo);
disp(L_1)
disp(L_2)
disp(C_1)
disp(C_2)
disp(L_3)
disp(C_3)
```

Scilab code Exa 9.6 design a low pass filter using micrstrip lines

```
// example:-9.6,page no.-498.
// design a low pass filter for fabrication using
   micrstrip lines.
disp("from table,the normalized low pass prototype
   element values are = ")
L1=3.3487;C2=0.7117;L3=3.3487;Rl=1.0000;
n=1+(1/3.3487);
disp(L1)
disp(Rl)
disp(C2)
disp(L3)
disp(n)
```

Scilab code Exa 9.7 design a stepped impedence low pass filter

```
// example:-9.7,page no.-503.
// design a stepped-impedence low pass filter having
    a maximally flat response and a cut-off
   frequency of 2.5 GHZ.
w=4*10^9;wc=2.5*10^9;Zh=150;Ro=50;Zl=10;
```

```
C1=0.517;L2=1.414;C3=1.932;L4=1.932;C5=1.414;L6
    =0.517;
// above values are taken from table.
// for finding electrical lengths.
x1=(C1*Zl/Ro)*(180/%pi);
x2=(L2*Ro/Zh)*(180/%pi);
x3=(C3*Zl/Ro)*(180/%pi);
x4=(L4*Ro/Zh)*(180/%pi);
x5=(C5*Zl/Ro)*(180/%pi);
x6=(L6*Ro/Zh)*(180/%pi);
disp(x1)
disp(x2)
disp(x3)
disp(x4)
disp(x5)
disp(x6)
```

Scilab code Exa 9.8 design a coupled line band pass filter

```
// example:-9.8,page no.-516.
// design a coupled line band pass filter with N=3.
delta=0.1;f=1.8*10^9;fo=2*10^9;Zo=50;fc=1;
f=(1/delta)*((f/fo)-(fo/f));
x=abs(f/fc)-1; // the value on the horizontal scale.
attntn=20; // from above values.
disp(attntn,'attenuation in db = ')
```

Scilab code Exa 9.9 design a bandpass filter

```
// example:-9.9,page no.-521.
// design a bandpass filter using three quarter wave
    open circuit stubs.
f=2*10^9;delta=0.15;Zo=50;N=3;gn=1.5963;
```

```
Zon=4*Zo/(%pi*gn*delta);
Z_on=(%pi*Zo*delta)/(4*gn);
disp(Zon,'the cahrracteristic impedence of a bandpass filter is = ')
disp(Z_on,'for a bandpass filter using short circuited stub resonators,the corresponding result is = ')
```

Scilab code Exa 9.10 design a bandpass filter using capacitive coupled

```
// example:-9.10,page no.-524.
// design a bandpass filter using capacitive coupled resonators,with a 0.5 db equal passband haracteristic.
fo=2*10^9;delta=0.1;Zo=50;f=2.2*10^9;g1=1.5963;g2=1.0967;g3=1.5963;g4=1;
f=(1/delta)*((f/fo)-(fo/f));
x=abs(f/fc)-1; // the value on the horizontal scale.
x0=sqrt((%pi*delta)/(2*g1))/Zo; // x0=ZoJ1;
x1=((%pi*delta)/(2*sqrt(g1*g2)))/Zo; // x0=ZoJn;
B0=x0/(1-(Zo*x0)^2)
B1=x1/(1-(Zo*x1)^2)
theta0=(%pi-0.5*(atan(2*Zo*B0)+atan(2*Zo*B1)))*(180/%pi);
C0=(B0/(2*%pi*fo))*10^12;
disp(theta0,'thetao in degree = ')
disp(C0,'the coupling capacitor value in PF = ')
```

Chapter 10

THEORY AND DESIGN OF FERRIMAGNETIC COMPONENTS

Scilab code Exa 10.1 calculate and plot phase and attenuation constants

```
// example:-10.1;page no.-547.
// problem to calculate and plot the phase and
   attenuation constants for RHCP & LHCP plane wave.
M=1800;// M=4*pi*Ms;
deltaH=75;eo=8.854*10^-12;muo=4*%pi*10^-7;c=3*10^8;
Ho=3570;er=14;tandelta=0.001;
fo=(2.8*10^9)/Ho; // IN GHZ.
wo=2*%pi*fo;
fm=(2.8*10^9)/M;  // IN GHZ.
wm=2*%pi*fm;
mue=muo*(1+(wo*wm)/(wo^2-wm^2));
e=eo*er*(1-%i*tandelta);
f=0:1000000:20*10^9;
w=2*%pi.*f;
k=muo*((w.*wm)/(wo^2-w^2));
gama=%i*w*sqrt(e.*(mue-k));
alpha=abs(real(gama));
```

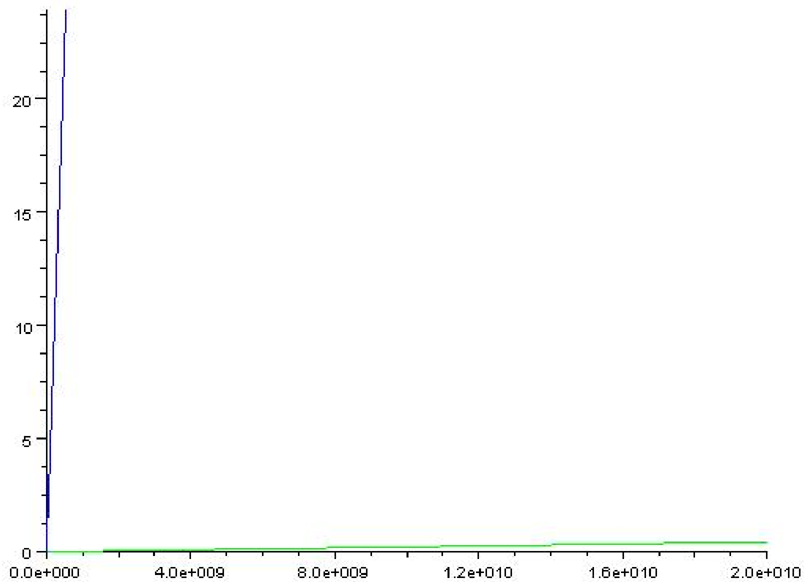

Figure 10.1: calculate and plot phase and attenuation constants

```
bta=abs(imag(gama));
plot2d(f,gama,style=3,rect=[0,0,20*10^9,24])
plot2d(f,bta,style=2,rect=[0,0,20*10^9,24])
```

Scilab code Exa 10.2 program to design an e plane resonance isolator

```
//example:-10.2,page no.-559.
// program to design an e plane resonance isolatorin
    x band waveguide.
er=13;revatt=30;
deltaH=200;x=1700;   // x=4*%pi*Ms.
```

```
f=10*10^9;alpha_=12.4; // from graph 10.13.
L=revatt/alpha_;
alpha_1=27/L;
disp(L,'for total reverse attenuation of 20 db,the
   length of the slab in cm must be = ')
disp(alpha_1,'for total reverse attenuation to be at
    least 27 db, alpha_ in db/cm be > ')
```

Scilab code Exa 10.3 program to design a resonance isolator

```
//example:-10.3.page no.-560.
// program to design a resonance isolator using the
   H-plane ferrite slab geometry in x-band.
f=10*10^9;delta_sbys=0.01;forpims=1700;deltaH=200;
revatt=30;ko=(2*%pi*f)/(3*10^8);
Ho=f/(2.8*10^9);
// for x-band waveguide, a=2.286 cm.
a=2.286;
kc=(%pi*100)/a;
betao=sqrt(ko^2-kc^2);
x=(1/%pi)*atan(kc/betao); // x=c/a.
L=revatt/2;
disp(L,'the slab length required for 30db total
   reverse attenuation in cm = ')
disp(kc,'cut-off wave number in m-1 = ')
disp(betao,'propagation constant = ')
```

Scilab code Exa 10.5 design a two slab remanent phase shifter

```
//example:10.5,page no.-567.
// program to design a two slab remanent phase
   shifter.
```

```
forpims=1786;er=13;f=10*10^9;uo=4*%pi*10^-7;ko=(2*%pi*f)/(3*10^10);
fm=2.8;s=0.1;// s and a in cm.
x=(2*%pi*fm*forpims)/(2*%pi*f);// x=wm/w = k/uo.
a=2.286; // for x-band.
t=.274; //from figure 10.19;
diffphaseshift=0.4*ko*(180/%pi); // differential phase shift.
L_1=180/diffphaseshift;
L_2=90/diffphaseshift;
disp(L_1,'the ferrite length required for the 180 deg. phase shift section in cm = ')
disp(L_2,'the ferrite length required for the 90 deg. phase shift section in cm = ')
```

Chapter 11

ACTIVE MICROWAVE CIRCUITS

Scilab code Exa 11.1 determine equivalent noise temperature of amplifie

```
// example:-11.1,page no.-589.
// program to determine the equivalent noise
   temperature of the amplifier.
T1=290;P1=-62;G=100;B=10^9;k=1.38*10^-23;
T2=77;P2=-64.7;Ts=450;
Y=P1-P2;  // Y-factor in db.
Y=10^0.27;
Te=(T1-Y*T2)/(Y-1);
Po=G*k*B*(Ts+Te);
Po=10*log10(Po/0.001); /// converting in to dBm.
disp(Te,'the equivalent noise temperature in kelwin
   = ')
disp(Po,'the total noise power out of the amplifier
   in dBm will be = ')
```

Scilab code Exa 11.2 find the dynamic range of the amplifier

```
//example:-11.2,page no.-591.
// program to find the dynamic range of the amplifier.
G=20;F=3.5; // in db.
k=1.38*10^-23;To=290;B=2*10^9;
// output noise power => No=G*F*k*To*B.so in dbm it will be-
No=20+3.5+10*log10((k*To*B)/0.001);
DR=10-No;
disp(DR,'the dynamic range in dB = ')
```

Scilab code Exa 11.3 program to calculate the noise figure

```
// example:-11.3,page no.-593.
// program to calculate the noise figure ig anteena is replaced by amplifier.
L=10^0.2;T=300;To=290;Te=150;
Fl=1+(L-1)*(T/To);
Fld=10*log10(Fl); // converting in to dBm.
Fa=1+(Te/To)
Fad=10*log10(Fa);// converting in to dBm.
Fcas=Fl+L*(Fa-1);
Fcasd=10*log10(Fcas);// converting in to dBm.
disp(Fcasd,'the noise figure of the cascade in dB = ')
disp(Fad,'the noise figure of the amplifier in dB = ')
disp(Fld,'the noise figure of the line in dB = ')
```

Scilab code Exa 11.4 calculate the impedence of the diode

```
//example:-11.4,page no.-596.
//program to calculate the impedence of the diode.
```

```
Cp=0.1*10^-12;Lp=2*10^-9;Cj=0.15*10^-12;Rs=10;Is=0.1*10^(-6);
Io1=0;Io2=60*10^(-6);alpha=(1/25)*(10^3);
R1j=1/(alpha*(Io1+Is)); // for Io=0.
R2j=1/(alpha*(Io2+Is)); // for Io=60 mA.
disp(R1j,'junction resistance for Io=0, in ohm = ')
disp(R2j,'junction resistance for Io=0, in ohm = ')
```

Scilab code Exa 11.5 determine the stability of the transistor

```
//example:-11.5,page no.-617.
//program to determine the stability of the transistor by calculating k and |delta|.
s11=0.894*expm(%i*(-60.6)*%pi/180);
s21=3.122*expm(%i*(123.6)*%pi/180);
s12=0.02*expm(%i*(62.4)*%pi/180);
s22=0.781*expm(%i*(-27.6)*%pi/180);
delta=(s11*s22)-(s12*s21);
[mag_delta,theta_delta]=polar(delta);
k=(1+(abs(delta)^2)-(abs(s11)^2)-(abs(s22)^2))/(2*abs(s12*s21));
Cl=conj(s22-delta*conj(s11))/(abs(s22)^2-abs(delta)^2);
[mag_Cl,theta_Cl]=polar(Cl);
Rl=abs(s12*s21)/(abs(s22)^2-abs(delta)^2);
Cs=conj(s11-delta*conj(s22))/(abs(s11)^2-abs(delta)^2);
[mag_Cs,theta_Cs]=polar(Cs);
Rs=abs(s12*s21)/(abs(s11)^2-abs(delta)^2);
disp([mag_Cl,theta_Cl])
disp([mag_Cs,theta_Cs])
disp(Rl)
disp(Rs)
disp("NOTE:-theta is in radian")
```

Scilab code Exa 11.6 design an amplifier for maximum gain

```
// example:11.6,page no.-620.
// program to design an amplifier for maximum gain
   at 4 GHZ using single stub matching section.
s11=0.72*expm(%i*(-116)*%pi/180);
s22=0.73*expm(%i*(-54)*%pi/180);
s12=0.03*expm(%i*(57)*%pi/180);
s21=2.6*expm(%i*(76)*%pi/180);
delta=(s11*s22)-(s12*s21)
k=(1+(abs(delta)^2)-(abs(s11)^2)-(abs(s22)^2))/(2*
   abs(s12*s21))
B1=1-(abs(delta)^2)+(abs(s11)^2)-(abs(s22)^2);
B2=1-(abs(delta)^2)-(abs(s11)^2)+(abs(s22)^2);
C1=s11-delta*conj(s22);
C2=s22-delta*conj(s11);
taos=(B1-sqrt(B1^2-4*abs(C1)^2))/(2*C1);
[mag_taos,theta_taos]=polar(taos);
taol=(B2-sqrt(B2^2-4*abs(C2)^2))/(2*C2);
[mag_taol,theta_taol]=polar(taol);
Gs=1/(1-abs(taos)^2);
Gs=10*log10(Gs);
Go=abs(s21)^2;
Go=10*log10(Go);
Gl=(1-abs(taol)^2)/(abs(1-s22*taol)^2);
Gl=10*log10(Gl);
Gtmax=Gs+Go+Gl;
disp(Gs,'Gs = ')
disp(Go,'Go = ')
disp(Gl,'Gl = ')
disp(Gtmax,'the over all transducer gain in dB will
   be = ')
Gs=1/(1-abs(taos)^2);
Gs=10*log10(Gs);
```

Scilab code Exa 11.7 design an amplifier to have a gain of 11 dB

```
// example:-11.7,page no.-625.
// program to design an amplifier to have a gain of
   11 dB at 4 GHZ.
s11=0.75*expm(%i*(-120)*%pi/180);
s21=2.5*expm(%i*(80)*%pi/180);
s12=0;
s22=0.6*expm(%i*(-70)*%pi/180);
Gsmax=1/(1-abs(s11)^2);
Gsmax=10*log10(Gsmax);
Glmax=1/(1-abs(s22)^2);
Glmax=10*log10(Glmax);
Go=abs(s21)^2;
Go=10*log10(Go);
Gtumax=Gsmax+Glmax+Go;
disp(Gsmax,'the maximum matching section gain in dB
   = ')
disp(Glmax,'the maximum matching section gain in dB
   = ')
disp(Go,'the gain of the mismatched transistor in dB
    = ')
disp(Gtumax,'the maximum unilateral transducer gain
   in dB = ')
```

Scilab code Exa 11.8 calculate maximum error in Gt and design amplifier

```
// example:-11.8,page no.-629.
// program to maximum error in Gt and design an
   amplifier having a 2 dB noise figure with the
   maximum gain that is compatible with the noise
   figure.
```

```
s11=0.6*expm(%i*(-60)*%pi/180);
s21=1.9*expm(%i*(81)*%pi/180);
s12=0.05*expm(%i*(26)*%pi/180);
s22=0.5*expm(%i*(-60)*%pi/180);
Fmin=1.6;F=1.58;Zo=50;
Fmin1=10^0.16
tao_opt=0.62*expm(%i*(100)*%pi/180);
atan(imag(tao_opt)/real(tao_opt))
Rn=20;
U=abs(s12*s21*s11*s22)/((1-abs(s11)^2)*(1-abs(s22)
    ^2));
x=1/(1+U)^2;
y=1/(1-U)^2;
disp("x<(Gt/Gtu)<y")
N=(((F-Fmin1)*Zo)/(4*Rn))*abs(1+tao_opt)^2
Cf=tao_opt/(N+1);
[mag_Cf,theta_Cf]=polar(Cf);
Rf=sqrt(N*(N+1-abs(tao_opt)^2))/(N+1);
disp(N,'N = ')
disp([mag_Cf,theta_Cf],'center of the 2 db noise
    figure circle = ')
disp(Rf,'the radius of the 2 dB noise figure circle
    = ')
Gl=1/(1-abs(s22)^2);
Gl=10*log10(Gl);
Go=abs(s21)^2;
Go=10*log10(Go);
Gs=1.7; // all Gl,Go,Gtu are in dB.
Gtu=Gs+Go+Gl;
disp(Gtu,'the over all transducer gain in db will be
    = ')
```

Scilab code Exa 11.9 design a load matching network

```
// example:-11.9,page no.-635.
```

```
// program to design a load matching network for a
   50 ohm load impedence.
Zo=50;f=6*10^9;taoin=1.25*expm(%i*(40)*%pi/180);
Zin=((1+taoin)/(1-taoin))*Zo;
Zl=-Zin;
disp(Zl,'the load impedence = ')
```

Scilab code Exa 11.10 program to design a transistor oscillator

```
//example:11.10,page no.-637.
// program to design a transistor oscillator at 4
   GHZ using a GaAs FET in common gate configuration
   .
s11=2.18*expm(%i*(-35)*%pi/180);
s21=2.75*expm(%i*(96)*%pi/180);
s12=1.26*expm(%i*(18)*%pi/180);
s22=0.52*expm(%i*(155)*%pi/180);// all are s
   parameter that are applicable for transistor in
   common gate configuration with aseries inductor.
delta=s12*s21-s11*s22;
Ct=conj(s22-delta*conj(s11));
Rt=abs((s12*s21)/(abs(s22)^2-abs(delta)^2))
taot=0.59*expm(%i*(-104)*%pi/180);
taoin=s11+(s12*s21*taot)/(1-s22*taot);
[mag_taoin,theta_taoin]=polar(taoin)
Zin=((1+taoin)/(1-taoin))*Zo;
Zl=-(real(Zin)/3)-(%i*imag(Zin));
disp([mag_taoin,theta_taoin])
disp(Zl,'the load impedence will be = ')
```

Scilab code Exa 11.11 obtain the greatest ratio of off to on attenuation

```
//example:-11.11,page no.-642.
```

```
// program to obtain the greatest ratio of  off to
   on attenuation.
Cj=0.1*10^-12;Rr=1;Rf=5;Li=0.4*10^-9;f=5*10^9;Zo=50;
w=2*%pi*f;
Zr=Rr+%i*((w*Li)-(1/(w*Cj)));
Zf=Rf+(%i*w*Li);
// for series circuit.
ILon=-20*log10(abs((2*Zo)/(2*Zo+Zf)));
ILoff=-20*log10(abs((2*Zo)/(2*Zo+Zr)));
// for shunt circuit.
ILon1=-20*log10(abs((2*Zr)/(2*Zr+Zo)));
ILoff1=-20*log10(abs((2*Zf)/(2*Zf+Zo)));
disp(ILon,'for series circuit = ')
disp(ILoff,'for series circuit = ')
disp(ILon1,'for shunt circuit = ')
disp(ILoff1,'for shunt circuit = ')
```

Chapter 12

INTRODUCTION TO MICROWAVE SYSTEMS

Scilab code Exa 12.1 compute directivity radiation intensity and others

```
// example:-12.1,page no.-668.
//program to compute directivity ,radiation intensity
    ,F,the effective area.
syms Etheta Hphi ko no Io l r pi theta C phi lamda;
Etheta=((%i*ko*no*Io*l)/(4*pi*r))*sin(theta)*exp(-%i
    *ko*r);
Hphi=((%i*ko*Io*l)/(4*pi*r))*sin(theta)*exp(-%i*ko*r
    );
F=(r^2)*(Etheta*conj(Hphi));
Prad=C*integ(integ(sin(theta)^3,theta),phi);
Prad=limit(Prad,theta,pi)-limit(Prad,theta,0);
Prad=limit(Prad,phi,2*pi)-limit(Prad,phi,0); // take
     cos(pi)=-1;
Prad=8*pi*C/3;
D=4*pi*C/Prad;
Ac=((lamda^2)*D)/(4*pi);
disp(F,'the radiation intensity is given by = ')
disp(D,'directivity is given by = ')
disp(Ac,'the effective area of the dipole = ')
```

Scilab code Exa 12.2 program to find the reactive power in dbm

```
// example:-12.2,page no.-674.
// program to find the reactive power in dbm.
Pt=120;f=6*10^9;
Gt=10^4.2;Gr=10^3.1;
lamda=0.05;R=3.59*10^7;
Pr=(Pt*Gt*Gr*(lamda^2))/((4*%pi*R)^2);
Pr=10*log10(Pr/0.001);
disp(Pr,'received power in dBm will be = ')
```

Scilab code Exa 12.3 calculate the input and output SNR

```
// example:-12.3,page no.-677.
// program to calculate the input and output SNR.
f=4*10^9;B=1*10^6;Grf=10^2;Gif=10^3;Lt=10^0.15;Lm
    =10^0.6;To=290;
Fm=10^0.7;Tm=(Fm-1)*To;Tp=300;Tb=200;eta=0.9;
Frf=10^0.3;Fif=10^0.11;k=1.38*10^-23;
Trf=(Frf-1)*To;
Tif=(Fif-1)*To;
Trec=Trf+(Tm/Grf)+((Tif*Lm)/Grf);
Ttl=(Lt-1)*Tp;
Ta=eta*Tb+(1-eta)*Tp;
Ni=k*B*Ta;
Ni=10*log10(Ni/0.001);  // converting in to dBm.
si=-80; // in dBm.
SNRi=si-Ni; // input SNR.
Tsys=Ta+Ttl+Lt*Trec;
SNRo=si-10*log10((k*B*Tsys)/0.001);
disp(SNRi,'input SNR in dB = ')
disp(SNRo,'output SNR in dB = ')
```

Scilab code Exa 12.4 program to find the maximum range of radar

```
// example:-12.4,page no.-683.
// program to find the maximum range of radar.
G=10^2.8;Pt=2000;sigma=12;
Pmin=10^-12;lamda=0.03;
Rmax=((Pt*(G^2)*sigma*(lamda^2))/(((4*%pi)^3)*Pmin))^(0.25);
disp(Rmax,'the maximum range of the radar in meter = ')
```

Scilab code Exa 12.5 program to find the J by S ratio

```
//example:-12.5,page no.-702.
// program to find the J/S ratio.
Gr=10^3.5;Pj=1000;R=3000;Br=1*10^6;Bj=20*10^6;
Gj=10;lamda=0.03;Pt=10^5;sigma=4;Rj=10000;
x=(Pj/Pt)*((4*%pi*(R^2)*Gj)/(sigma*Gr))*(Br/Bj); // x=J/S
x=10*log10(x);
Grsl=10^(3.5-2);// radar anteena gain in its sidelobe region.
x1=(Pj/Pt)*(((R^4)*Gj*Grsl)/((Gr^2)*(Rj^2)))*(Br/Bj);
x1=10*log10(x1);
disp(x,'THE J/S ration for the SSJ case in dB is = ')
disp(x1,'THE J/S ratio for the SOJ case in dB is = ')
```

Scilab code Exa 12.6 calculate power density of 20 m from anteena

```
// example:-12.6,page no.-704.
// program to calculate the power density of 20 m
   from the anteena.
G=10^4;Pin=5;R=20;
S=(Pin*G)/(4*%pi*(R^2))*0.1;
disp(S,'the power density in the main beam of the
   anteena at a distance of 20 m in mw/cm^2 = ')
```

Appendix

Scilab code AP 1 equivalent of two resistances in parallel

```
//function example:-5.4,page no.-221.
function[Z]=parallel_impedence(Z1,Z2)
  Z=(Z1*Z2)/(Z1+Z2);
endfunction
```

Scilab code AP 2 smith chart for finding load impedence when reflection coefficient is given.

```
// function for smith chart for finding load
   impedence when reflection coefficient is given.
function[]=smith_chart(tao)
theta=0:0.1:2*%pi;
for r=0:0.1:10
x=(1/(1+r))*cos(theta)+(r/(1+r));
y=(1/(1+r))*sin(theta);
plot2d(x,y,style=2,rect=[-2,-2,2,2])
end
for X=-2:0.1:2
   if X==0
    X=0.01;
  end
x=1+(1/X)*cos(theta);
y=(1/X)*sin(theta)+(1/X);
plot2d(x,y,style=3,rect=[-2,-2,2,2])
xgrid(2)
xtitle("smith chart","Tao_Real","Tao_Imaginary")
```

```
end
x=abs(tao)*cos(theta);
y=abs(tao)*sin(theta);
plot2d(x,y,style=5,rect=[-2,-2,2,2])
theta=-%pi/2:0.1:%pi/2;
x=abs(tao)*cos(theta);
[r angle]=polar(tao);
tao=[r angle]
y=x*tan(tao(1,2));
plot2d(x,y,style=5,rect=[-2,-2,2,2])
endfunction
```

Scilab code AP 3 function for input impedence

```
// function for input impedence.
function[Zin]=input_impedence(tao,b,l,Zo)
  Zin=Zo*((1+(tao*exp(-2*%i*b*l)))/(1-(tao*exp(-2*%i*b*l))))
endfunction
```

Scilab code AP 4 function for reflection coefficient

```
"Tao_Real","Tao_Imaginary""Tao_Real","Tao_Imaginary" // function for reflection coefficient.
function[tao]=reflection_coefficient(Zl,Zo)
  tao=(Zl-Zo)/(Zl+Zo);
endfunction
```

Scilab code AP 5 function to find SWR

```
// function to find SWR,
function[SWR]=VSWR(tao)
  SWR=(1+tao)/(1-tao)
endfunction
```