Scilab Textbook Companion for High Frequency and Microwave Engineering by E. Da Silva[1]

Created by
Sufiyan Siddique
BE (pursuing)
Electronics Engineering
A.I Kalsekar Technical Campus, Mumbai University
College Teacher
Prof Chaya S
Cross-Checked by
Chaitanya Potti

June 1, 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: High Frequency and Microwave Engineering

Author: E. Da Silva

Publisher: Butterworth-heinemann, Oxford, UK

Edition: 1

Year: 2001

ISBN: 075065046X

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

Chapter 1

Basic features of radio communication systems

Scilab code Exa 1.1 Example 1

```
//Chapter 1, Example 1.1
clc
E=10e-6                                 //electric field strength in volts/metre
Z=377                                   //wave impedance in ohm
A=5                                     //area in metre square

//calculating magnetic field strength
H=E/Z
//calculating incident power
P=E*H*A

printf("(a) Magnetic field strength = %.3f nA/m\n\n",H*10^9)
printf("(b) Incident power on a recieving aerial = %.3f pW\n\n",P*10^12)
```

Scilab code Exa 1.2 Example 2

```
//Chapter 1, Example 1.2
clc
E=10e-6                               //electric field strength in volts/metre
Z=377                                 //wave impedance in ohm
P=2.65*10^-13                         //power incident on a surface area 1m2

D1=10                                 //distance in kilometre
D2=100                                //distance in kilometre
P1=(D1/D2)^2                          //power density factor
PD=P1*P                               //power dentity in W/m2

Erms=sqrt(PD*Z)                       //Erms at 100Km
Hrms=Erms/Z                           //Hrms at 100Km
printf("(a) Erms at 100km = %.3f microV/m\n\n",Erms*10^6)
printf("(b) Hrms at 100km = %.3f nA/m\n\n",Hrms*10^9)
```

Scilab code Exa 1.3 Example 3

```
//Chapter 1, Example 1.3
clc
```

```
Vo=100                        //transmitter output in
    volts
Zo=72                         //transmitter output
    impedance in ohm
Zin=72                        //antenna input
    impedance in ohm
R=72                          //radiation resistance
    in ohn
eff=100                       //antenna efficiency

Vrms=Vo/2
Pr=(Vrms)^2/R

printf("Power radiated = %.2f W",Pr)
```

Scilab code Exa 1.4 Example 4

```
//Chapter 1, Example 1.4
clc
n=105                             //no of coil turns
a=8*10^-5                         //cross sectional
    area in metre square
ur=230                            //relative
    permeability
uo=4*%pi*10^-7                    //permeability of
    air
Erms=10*10^-6                     //electric field
    strength
f=10^6                            //frequency in hertz

//calculation
erms=n*2*%pi*f*ur*uo*Erms*a*cos(0)

printf("r.m.s open circuit voltage = %.2f microVolt"
    ,erms*10^6)
```

Scilab code Exa 1.5 Example 5

```
//Chapter 1, Example 1.5
clc
n=100                              //no of coil turns
a=8*10^-5                          //effective cross-sectional area in metre
ur=200                             //relative permeability
uo=4*%pi*10^-7                     //permeability, of air
ao=60*%pi/180                      //angle of incidence of magnetic field
f=10^6                             //frequency in hertz
E=100*10^-6                        //electric field strength in V/m
z=377                              //wave impedance in ohm

//calculation
w=2*%pi*f
erms=n*w*ur*uo*a*(E/z)*cos(ao)

printf("r.m.s open circuit voltage induced = %.3f uV",erms*10^6)
```

Scilab code Exa 1.6 Example 6

```
//Chapter 1, Example 1.6
clc
n=2                          //no of recievers
```

```
Zo=75                          //input impedance of
   each receiver

//calculating the value of resistor
R=((n-1)/(n+1))*Zo

printf("Value of the matching resistor = %d ohm",R)
```

Scilab code Exa 1.7 Example 7

```
//Chapter 1, Example 1.7, figure 1.23
clc
n=4                            //no of recievers
Zo=50                          //input impedance of
   each receiver

//calculating the value of resistor
R=((n-1)/(n+1))*Zo

printf("Value of the matching resistor = %d ohm",R)
```

Scilab code Exa 1.8 Example 8

```
//Chapter 1, Example 1.8, figure 1.23
clc
n=3                            //no of recievers
Zo=50                          //input impedance of
   each receiver
Van=100*10^-6                  //open-circuit voltage
   in aerial

//calculation
R=((n-1)/(n+1))*Zo
```

```
V=1/(2*n)*Van

printf("(a) Value of the matching resistor = %d ohm\
   n\n\n",R)
printf("(b) Voltage at receiver input terminal = %.3
   f uV",V*10^6)
```

Scilab code Exa 1.9 Example 9

```
//Chapter 1, Example 1.9, Figure 1.25
clc
r1=75                              //network splitter
   and termination impedance
r2=43                              //matching network

//calculation
Voc=(r1/(r2+r1))*(((r1+r2)/2)/(((43+75)/2)+r1))
Zin=((r1+r2)/2)
Zsr=r2+(((r1+r2)*r1)/((r2+r1)+r1))

printf("(a) Ratio Vout/Voc = %.2f \n\n",Voc)
printf("(b) Input impedance to the network = %d ohm\
   n\n",Zin)
printf("(c) Reciever source impedance = %.2f ohm\n\n
   ",Zsr)
```

Chapter 2

Transmission lines

Scilab code Exa 2.2 Example 2

```
//Chapter 2, Problem 2
clc
D=300e-3                              //distance in meter
   between the two wire
d=4e-3                                //diameter in meter of a
    conductor
e=1                                   //relative dielectric

//calculating the characteristic impedance of the
   type of parallel transmission line
z0=(276/sqrt(e))*log10(2*D/d)
printf("Characteristic impedance, Z0 = %d ohm",z0)
```

Scilab code Exa 2.4 Example 4

```
//Chapter 2, Problem 4
clc
funcprot(0)
```

```
R=23                                //resistance in ohm
G=4*10^-3                           //conductance in siemens
L=125*10^-6                         //inductance in henry
C=48*10^-9                          //capacitance in farad

//list of frequencies in hertz
f1=100
f2=500
f3=15e3
f4=5e6
f5=10e6

deff('[a]=imp(R,G,L,C,f)','a=sqrt((R+(%i*2*%pi*f*L))/(G+(%i*2*%pi*f*C)))');
deff('[b]=imp1(d)','b={(real(d)^2)+(imag(d)^2)}^0.5');
deff('[c]=imp2(e)','c=atan(imag(e),real(e))');
[Z01]=imp(R,G,L,C,f1)
[Z_mag1]=imp1(Z01)
[Z_ang1]=imp2(Z01)
[Z02]=imp(R,G,L,C,f2)
[Z_mag2]=imp1(Z02)
[Z_ang2]=imp2(Z02)
[Z03]=imp(R,G,L,C,f3)
[Z_mag3]=imp1(Z03)
[Z_ang3]=imp2(Z03)
[Z04]=imp(R,G,L,C,f4)
[Z_mag4]=imp1(Z04)
[Z_ang4]=imp2(Z04)
[Z05]=imp(R,G,L,C,f5)
[Z_mag5]=imp1(Z05)
[Z_ang5]=imp2(Z05)

printf("Characteristic impedance Z0 for a given frequency is,\n\n")
```

```
printf("(a) for 100 Hz, \nZ0(magnitude) = %.2f ohm\n Z0(angle) = %f rad\n\n",Z_mag1,Z_ang1)
printf("(b) for 500 Hz, \n Z0(magnitude) = %.2f ohm\n Z0(angle) = %f rad\n\n",Z_mag2,Z_ang2)
printf("(c) for 15 KHz, \n Z0(magnitude) = %.2f ohm\n Z0(angle) = %f rad\n\n",Z_mag3,Z_ang3)
printf("(d) for 5 MHz, \nZ0(magnitude) = %.2f ohm\n Z0(angle) = %f rad\n\n",Z_mag4,Z_ang4)
printf("(e) for 10 MHz, \n Z0(magnitude) = %.2f ohm\n Z0(angle) = %f rad\n\n",Z_mag5,Z_ang5)
```

Scilab code Exa 2.5 Example 5

```
//Chapter 2, Problem 5
clc
f=1.6*10^6                              //frequency in hertz
Zoc_mag=900                             //magnitude in ohm of open circuit impedance
Zoc_ang=-30                             //angle in degree of open circuit impedance
Zsc_mag=400                             //magnitude in ohm of short circuit impedance
Zsc_ang=-10                             //angle in degree of short circuit impedance

//calculation of charactersitics impedance
Z0_mag=sqrt(Zoc_mag*Zsc_mag)
Z0_ang=Zoc_ang-Zsc_ang

printf("Z0 (magnitude) = %d ohm\n Z0(angle) = %d degree",Z0_mag,Z0_ang)
```

Scilab code Exa 2.6 Example 6

```
//Chapter 2, Problem 6
clc
a=2                                     //attenuation
    constant
l=10                                    //length in Km

//calculation of loss in dB
a10=a*l
loss=8.686*a10

printf("Loss in dB = %.2f dB",loss)
```

Scilab code Exa 2.7 Example 7

```
//Chapter 2, Problem 7
clc
Zl=80-%i*10                         //load impedance in
    complex form
Z0=50                               //characteristic
    impedance in ohm

//calculation of reflection coefficient
ref=(Zl-Z0)/(Zl+Z0)
ref_mag={(real(ref)^2)+(imag(ref)^2)}^0.5
ref_ang=atan(imag(ref)/real(ref))

printf("Reflection coefficient is given by \n
    magnitude = %.2f \n angle = %.2f degree",ref_mag,
    ref_ang*180/%pi)
```

Scilab code Exa 2.8 Example 8

```
//Chapter 2, Problem 8
clc
Z0=50                                           //
    characteristic impedance in ohm
//load impedance in ohm
Zl1=50
Zl2=0
Zl3=75

//calculation of reflection coefficient
ref1=(Zl1-Z0)/(Zl1+Z0)
ref2=1
ref3=(Zl2-Z0)/(Zl2+Z0)
ref4=(Zl3-Z0)/(Zl3+Z0)
printf("(a) with Zl = 50, reflection coefficient is
    \n magnitude = %f \n angle = %d degree\n\n",ref1,
    ref1)
printf("(b) with Zl = open circuit , reflection
    coefficient is \n magnitude = %f \n angle = %d
    degree\n\n",ref2,angle=0)
printf("(c) with Zl = short circuit, reflection
    coefficient is \n magnitude = %f \n angle = %d
    degree\n\n",ref3,angle2=0)
printf("(d) with Zl = 75, reflection coefficient is
    \n magnitude = %f \n angle = %d degree\n\n",ref4,
    angle3=0)
```

Scilab code Exa 2.9 Example 9

```
//Chapter 2, Problem 9
clc
Vin=100                         //incident voltage
Vref=10                         //reflected voltage

//calculation of voltage standing wave ratio
```

```
Vswr=(Vin+Vref)/(Vin-Vref)

printf("VSWR = %.2f",Vswr)
```

Scilab code Exa 2.10 Example 10

```
//Chapter 2, Problem 10
clc
ref=0.1                                    //reflection
   coefficient

//calculation of voltage standing wave ratio
Vswr=(1+ref)/(1-ref)

printf("VSWR = %.2f",Vswr)
```

Scilab code Exa 2.11 Example 11

```
//Chapter 2, Problem 11
clc
Vswr=1.07                          //voltage standing wave
   ratio
Z0=50                              //characteristic
   impedance in ohm

//calculation of power reflected in percent
ref=(Vswr-1)/(Vswr+1)
Pref=(ref^2*100)

printf("Reflected power = %.1f percentage of power
   incident",Pref)
```

Scilab code Exa 2.12 Example 12

```
//Chapter 2, Problem 12
clc
funcprot(0)
R=23                                          //resistance in
    ohm
G=4*10^-3                                     //conductance in
    siemens
L=125*10^-6                                   //inductance in
    henry
C=48*10^-9                                    //capacitance in
    farad

//list of frequencies in hertz
f1=100
f2=500
f3=15000
f4=5*10^6
f5=10*10^6

//calculation of characteristic impedance by
    declaring the function
function [c,d,e,f]=myfct(u,v,x,y,z)
  a=u+%i*2*%pi*z*v
  b=x+%i*2*%pi*z*y
  m=sqrt(a*b)
  n=sqrt(a/b)
  c=sqrt(real(m)^2+imag(m)^2)
  d=atan(imag(m),real(m))
  e=sqrt(real(n)^2+imag(n)^2)
  f=atan(imag(n),real(n))
endfunction

```

```
[y1,z1,x1,v1]=myfct(R,L,G,C,f1)
[y2,z2,x2,v2]=myfct(R,L,G,C,f2)
[y3,z3,x3,v3]=myfct(R,L,G,C,f3)
[y4,z4,x4,v4]=myfct(R,L,G,C,f4)
[y5,z5,x5,v5]=myfct(R,L,G,C,f5)

printf("(a)for 100 Hz \n\tPropagation constant is \n\t\t= %f (magnitude)\n\t\t= %f (angle)\n\n\tCharacteristic impedance Z0 is \n\t\t= %f ohm(magnitude)\n\t\t= %f (angle)\n\n",y1,z1,x1,v1)
printf("(b)for 500 Hz \n\tPropagation constant is \n\t\t= %f (magnitude)\n\t\t= %f (angle)\n\n\tCharacteristic impedance Z0 is \n\t\t= %f ohm(magnitude)\n\t\t= %f (angle)\n\n",y2,z2,x2,v2)
printf("(c)for 15 KHz \n\tPropagation constant is \n\t\t= %f (magnitude)\n\t\t= %f (angle)\n\n\tCharacteristic impedance Z0 is \n\t\t= %f ohm(magnitude)\n\t\t= %f (angle)\n\n",y3,z3,x3,v3)
printf("(d)for 5 MHz \n\tPropagation constant is \n\t\t= %f (magnitude)\n\t\t= %f (angle)\n\n\tCharacteristic impedance Z0 is \n\t\t= %f ohm(magnitude)\n\t\t= %f (angle)\n\n",y4,z4,x4,v4)
printf("(e)for 10 MHz \n\tPropagation constant is \n\t\t= %f (magnitude)\n\t\t= %f (angle)\n\n\tCharacteristic impedance Z0 is \n\t\t= %f ohm(magnitude)\n\t\t= %f (angle)\n\n",y5,z5,x5,v5)
```

Scilab code Exa 2.13 Example 13

```
//Chapter 2, Problem 13
clc
Z0=377                                    //characteristic impedance in ohm
l=1/7                                     //electrical length
```

```

//calculation of input impedance
Zin=%i*Z0*tan(2*%pi*l)

printf("Input impedance = j%.1f  ohm",imag(Zin))
```

Scilab code Exa 2.14 Example 14

```
//Chapter 2, Problem 14
clc
Z0=75                                         //
    characteristic impedance in ohm
l=1/5                                          //electrical
     length

//calculation of input impedance
Zin=-%i*Z0*cotg(2*%pi*l)

printf("Input impedance = %.1fj ohm",imag(Zin))
```

Scilab code Exa 2.15 Example 15

```
//Chapter 2, Problem 15
clc
Zl=20                            //load impedance in
    ohm
Z0=90                            //characteristic
    impedance in ohm
l=1/4                            //electrical length

//calculation of input impedance
Zin=(Z0)^2/Zl
printf("Input impedance = %d ohm ",Zin);
```

Chapter 3

Smith charts and scattering parameters

Scilab code Exa 3.1 Example 1

```
//Chapter 3, Problem 1, Figure 3.4
clc
//https://atoms.scilab.org/toolboxes/microwave
//Download and install the Microwave toolbox from
   above link and load it from scilab menubar >
   Toolboxes > microwave


//Plot the smith chart
uW_display_smith([.2 .5 1 2 5],12);
Z=[0.7-%i*0.2 0.7+%i*0.3 0.3-%i*0.5 0.3+%i*0.3]
                    //impedances in matrix form

R2=0
plot2d(real(R2),imag(R2),-1);

for n=1:length(Z)
    Z1=50*Z(n)
    G=(Z1-50)/(Z1+50);
```

```
plot2d(real(G), imag(G),-8);
xtitle("Smith Chart");
end;
```

Scilab code Exa 3.2 Example 2

```
//Chapter 3, Problem 2
clc
//https://atoms.scilab.org/toolboxes/microwave
//Download and install the Microwave toolbox from
   above link and load it from scilab menubar >
   Toolboxes > microwave

funcprot(0)

// A = p2z(R,Theta) - Convert from polar to
   rectangular form.
//    R is a matrix containing the magnitudes
//    Theta is a matrix containing the phase angles
   (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

// [R1, Theta1] = z2p(A1) - Display polar form of
   complex matrix.
function [R1, Theta1] = z2p(A1)
     Theta1 = atan(imag(A1),real(A1))*180/%pi;
     R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction

//Plot the smith chart
uW_display_smith([.2 .5 1 2 5],12);
Z=0.8-%i*1.6;                                  //
   impedance
```

```
Z1=50*Z;                                           //50 = characteristic impedance
[Zm,Za]=z2p(Z);
G=(Z1-50)/(Z1+50);                                 //reflection coefficient
Ym=1/Zm                                            //admittance (magnitude)
Ya=Za*(-1)                                         //admittance (angle)
Y=p2z(Ym,Ya)
Y1=50*Y;
R=(Y1-50)/(Y1+50);
R2=0
plot2d(real(R2),imag(R2),-1);

plot2d(real(G),imag(G),-8);
plot2d(real(R),imag(R),-8);
xtitle("Smith Chart");
printf("Admittance value, Y = %.2f + j%.1f",real(Y),imag(Y));
```

Scilab code Exa 3.3 Example 3

```
//Chapter 3, Problem 3
clc
//https://atoms.scilab.org/toolboxes/microwave
//Download and install the Microwave toolbox from above link and load it from scilab menubar > Toolboxes > microwave

//Plot the smith chart
uW_display_smith([.2 .5 1 2 5],12);
r=0.667                                               //radius of VSWR circle
Z=0.25-%i*0.5;                                     //
```

```
impedance
Z1=50*Z;                                    //50 = characteristic impedance
G=(Z1-50)/(Z1+50);                          //reflection coefficient
R2=0
plot2d(real(R2),imag(R2),-1);
plot2d(real(G),imag(G),-8);

//Plot a VSWR circle of radius 0.667
x=linspace(0,2*%pi,200);
plot2d(r*cos(x),r*sin(x))
xtitle("Smith Chart");


printf("From smith chart, The answer is %.3f (magnitude) and -124 degree (angle)",r)
disp("This is shown as point C in Figure 3.11.")
```

Scilab code Exa 3.4 Example 4

```
//Chapter 3, Problem 4
clc
//https://atoms.scilab.org/toolboxes/microwave
//Download and install the Microwave toolbox from above link and load it from scilab menubar > Toolboxes > microwave

//Plot the smith chart
uW_display_smith([.2 .5 1 2 5],12);
r=0.667                                     //radius of VSWR circle
Z=0.21+%i*0.21;                             //impedance
Z1=50*Z;                                    //50 =
```

```
characteristic impedance
G=(Z1-50)/(Z1+50);                                    //reflection coefficient
R2=0
plot2d(real(R2),imag(R2),-1);
plot2d(real(G),imag(G),-8);

//Plot a VSWR circle of radius 0.667
x=linspace(0,2*%pi,200);
plot2d(r*cos(x),r*sin(x))
xtitle("Smith Chart");


printf("From smith chart, The answer is %.2f + j%.2f",real(Z),imag(Z))
disp("This is shown as point E in Figure 3.11.")
```

Scilab code Exa 3.6 Example 6

```
//Chapter 3, Problem 6
clc
//https://atoms.scilab.org/toolboxes/microwave
//Download and install the Microwave toolbox from above link and load it from scilab menubar > Toolboxes > microwave

funcprot(0)

// [R1, Theta1] = z2p(A1) - Display polar form of complex matrix.
function [R1, Theta1] = z2p(A1)
     Theta1 = atan(imag(A1),real(A1))*180/%pi;
     R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction

```

```
function Zin = zin(d)
    B=2*%pi*d
    Zin=Zo*((Zl+(%i*Zo*tan(B)))/(Zo+(%i*Zl*tan(B))))
endfunction

Zo=50                              //characteristic impedance
Zl=40-%i*80                        //load impedance
//line
d1=0.096
d2=0.173
d3=0.206

refl=(Zl-Zo)/(Zl+Zo)                       //reflection coefficient
[reflm,refla]=z2p(refl)
SWR=(1+reflm)/(1-reflm)                    //standing wave ratio
Zin1=zin(d1)
Zin2=zin(d2)
Zin3=zin(d3)
////load impedance is expressed in normalised form
a=Zl/Zo
d=0.25-%i*0.5
f=0.2+%i*0
h=0.2+%i*0.2
j=0.25-%i*0.5


//Plot the smith chart
uW_display_smith([.2 .5 1 2 5],12);
Z=[a d f h j]
for n=1:length(Z)
    Z1=50*Z(n)
    G=(Z1-50)/(Z1+50);
plot2d(real(G), imag(G),-8);
end;
R2=0
```

```
plot2d(real(R2),imag(R2),-1);

//Plot a VSWR circle of radius 0.667
x=linspace(0,2*%pi,200);
plot2d(0.66*cos(x),0.66*sin(x))
xtitle("Smith Chart");

printf("The input impedance Zin of the terminated line \n\n")
printf("(a) 0.096h = %.2f %.2fj\n\n",real(Zin1),imag(Zin1))
printf("(a) 0.173h = %.2f + %.2fj\n\n",real(Zin2),imag(Zin2))
printf("(a) 0.206h = %.2f + %.2fj\n\n",real(Zin3),imag(Zin3))
```

Scilab code Exa 3.8 Example 8

```
//Chapter 3, Problem 8
clc

j=0.25+(%i*0.5)                    //Normalised impedance from smith chart
Zo=50                              //Characteristic impedance

//calculating line impedance at point J in Figure 3.15
Zl=j*Zo

printf("Line impedance = %.1f + j%d",real(Zl),imag(Zl))
```

Scilab code Exa 3.9 Example 9

```
//Chapter 3, Problem 9
clc
Z0=50                              //characteristic
   impedance in ohm
Zl0=5                             //ratio of Z/Z0

//calculation
Z0t=sqrt(Zl0)
Z0t1=Z0*Z0t
printf("Impedance = %.2f ohm",Z0t1)
```

Scilab code Exa 3.10 Example 10

```
//Chapter 3, Problem 10
clc
Zlo=0.2                                  //ratio of Zl/Zo
    impedance in ohm
Zo=50                                    //output
   impedance in ohm

//calculation of characteristic impedance
Z0t=Zo*sqrt(Zlo)

printf("Characteristic impedance, Z0t = %d ohm",Z0t)
```

Scilab code Exa 3.11 Example 11

```
//Chapter 3, Problem 11, Figure 3.16(a), 3.18
clc
Zs=50+%i*0                         //impedance in ohm
Zl=100+%i*0                        //impedance in ohm
```

```
//using one quarter wave transformer in the circuit of figure 3.16(a)
Z0=sqrt(Zs*Zl)

//Use two quarter-wave transformers as in Figure 3.18
Z0t=sqrt(Zl/Zs)
Z0t2=60
Z0t1=Z0t2*Z0t

//using table 3.3
x=[0.6,0.8,1.0,1.2,1.4]
y=[-13.83,-19.28,-60,-19.28,-13.83]
clf;
x1=[0.6,0.8,1.0,1.2,1.4]
y1=[-18.81,-32.09,-38.69,-32.09,-18.81]
plot(x,y,".r")
plot(x1,y1,".b")
legend("one h/4","Two h/4")

xtitle("reflection coefficient vs frequency","frequency (GHz)","Reflection coefficient (dB)");
printf("(a) Matching network using one  /4 transformer,\n\t Z0 = %.2f ohm\n\n",Z0)
printf("(b) Matching network using two  /4 transformers,\n Ratio of Z0t1 and Z0t2 = %.3f\n",Z0t)
printf("If I choose a value of 60 ohm for Z0t2, then Z0t1 = %.2f ohm\n\n",Z0t1)
```

Scilab code Exa 3.13 Example 13

```
//Chapter 3, Problem 13
clc
```

```

//https://atoms.scilab.org/toolboxes/microwave
//Download and install the Microwave toolbox from above link and load it from scilab menubar > Toolboxes > microwave

funcprot(0)
// [R1, Theta1] = z2p(A1) - Display polar form of complex matrix.
function [R1, Theta1] = z2p(A1)
    Theta1 = atan(imag(A1),real(A1))*180/%pi;
    R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction

zin=100                          //input resistance in ohm
zo=50                            //amplifier input resistance in ohm
cl=5e-12                         //capacitance in farad
f=10^9                            //frequency in hertz
d=1+(%i*2.3)                         //point C
h=0-(%i*2.3)                         //point E

//Calculation
Yo=1/zo
Yl=(1/zin)+(%i*2*%pi*f*cl)
Y=Yl/Yo

//Plot the smith chart
uW_display_smith([.2 .5 1 2 5],12);
Y1=50*Y;
R=(Y1-50)/(Y1+50);
R2=0
[Rm,Ra]=z2p(R)
plot2d(real(R),imag(R),-8);
plot2d(real(R2),imag(R2),-1);
```

```
y=[d h]
for n=1:length(y)
    y1=50*y(n)
    R1=(y1-50)/(y1+50);
plot2d(real(R1), imag(R1),-8);
end;

//Plot a VSWR circle of radius 0.667
x=linspace(0,2*%pi,200);
plot2d(Rm*cos(x),Rm*sin(x))
xtitle("Smith chart")

printf("Yl/Yo = %.1f + j %.2f\n\n",real(Y),imag(Y))
```

Scilab code Exa 3.15 Example 15

```
//Chapter 3, Prblem 15, figure 3.30
clc
//from figure 3.30
zo=50                              //in ohm
zl=50                              //in ohm

//calculating the S parameter
z1=zo+zl
s11=(z1-zo)/(z1+zo)
z2=zo+zl
s22=(z2-zo)/(z2+zo)
s21=(2*zl)/(50+zo+zl)
s12=(2*zl)/(50+zo+zl)

s=[s11 s12;s21 s22]

printf("s11 (magnitude) = %.3f \n\t(angle) = 0
   degree\n\n",s11)
printf("s12 (magnitude) = %.3f \n\t(angle) = 0
```

```
    degree\n\n",s12)
printf("s21 (magnitude) = %.3f \n\t(angle) = 0
    degree\n\n",s21)
printf("s22 (magnitude) = %.3f \n\t(angle) = 0
    degree\n\n",s22)
disp(s,"S = ")
```

Scilab code Exa 3.16 Example 16

```
//Chapter 3, Prblem 16, figure 3.32
clc
//from figure 3.32
zo=50                              //in ohm
zl=50                              //in ohm

//calculating the S parameter
z1=(zo*zl)/(zo+zl)
s11=(z1-zo)/(z1+zo)
z2=(zo*zl)/(zo+zl)
s22=(z2-zo)/(z2+zo)
s21=(2*z1)/(50+z1)
s12=(2*z2)/(50+z2)

s=[s11 s12;s21 s22]

printf("s11 (magnitude) = %.3f \n\t(angle) = 180
    degree\n\n",-s11)
printf("s12 (magnitude) = %.3f \n\t(angle) = 0
    degree\n\n",s12)
printf("s21 (magnitude) = %.3f \n\t(angle) = 0
    degree\n\n",s21)
printf("s22 (magnitude) = %.3f \n\t(angle) = 180
    degree\n\n",-s22)
disp(s,"S (magnitude) = ")
```

Scilab code Exa 3.17 Example 17

```
//Chapter 3, Prblem 17, figure 3.34
clc
funcprot(0)
// [R1, Theta1] = z2p(A1) - Display polar form of
   complex matrix.
function [R1, Theta1] = z2p(A1)
     Theta1 = atan(imag(A1),real(A1))*180/%pi;
     R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction


//from figure 3.34
zo=50                          //in ohm
zl=50                          //in ohm
r1=30
zai=%i*20
za=(r1*zo)/(r1+zo)
z1=za+zai
s11=(z1-zo)/(z1+zo)
z2=(zo+zai)*r1/(zo+zai+r1)
s22=(z2-zo)/(z2+zo)
s21=za*2/(za+(zo+zai))
s12=zo*0.75/(za+zo+zai)


[s11m,s11a]=z2p(s11)
[s22m,s22a]=z2p(s22)
[s21m,s21a]=z2p(s21)
[s12m,s12a]=z2p(s12)

ret_loss=-20*log10(s11m)
ins_loss=-20*log10(s21m)
```

```

printf("(a) S parameters is , \n")
printf("s11 (magnitude) = %.3f \n\t(angle) = %.2f degree\n\n",s11m,s11a)
printf("s12 (magnitude) = %.3f \n\t(angle) = %.2f degree\n\n",s12m,s12a)
printf("s21 (magnitude) = %.3f \n\t(angle) = %.2f degree\n\n",s21m,s21a)
printf("s22 (magnitude) = %.3f \n\t(angle) = %.2f degree\n\n",s22m,s22a)
printf("(b) Return loss = %.3f dB\n\n",ret_loss)
printf("(c) Insertion loss = %.3f dB\n\n",ins_loss)
```

Scilab code Exa 3.18 Example 18

```
//Chapter 3, Prblem 18,
clc
funcprot(0)
// A = p2z(R,Theta) - Convert from polar to rectangular form.
//    R is a matrix containing the magnitudes
//    Theta is a matrix containing the phase angles (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

// [R1, Theta1] = z2p(A1) - Display polar form of complex matrix.
function [R1, Theta1] = z2p(A1)
     Theta1 = atan(imag(A1),real(A1))*180/%pi;
     R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction


```

```
//transistor S-parameter
s11=p2z(0.12,-10)
s12=p2z(0.002,-78)
s21=p2z(9.8,160)
s22=p2z(0.01,-15)

[s11m,s11a]=z2p(s11)
[s22m,s22a]=z2p(s22)
[s21m,s21a]=z2p(s21)
[s12m,s12a]=z2p(s12)

vswr=(1+s11m)/(1-s11m)
ret_loss=-20*log10(s11m)
Fig=20*log10(s21m)
Rig=20*log10(s12m)

printf("(a) Input VSWR = %.2f\n\n",vswr)
printf("(b) Return loss (dB) = %.2f dB\n\n",ret_loss
    )
printf("(c) Forward insertion gain = %.2f dB\n\n",
    Fig)
printf("(d) Reverse insertion gain = %.2f dB\n\n",
    Rig)
```

Chapter 5

Amplifier basics

Scilab code Exa 5.1 Example 1

```
//Chapter 5, Problem 1
clc
R=3                               //resistance in ohm
L=20*10^-9                        //inductance in henry
f0=500e6                          //frequency in hertz

//calculation
Z=R
C=(1/(2*%pi*f0*sqrt(L)))^2
Q=2*%pi*f0*L/R
B=f0/Q

printf("(a) Impedance at resonance = %d ohm\n\n",Z)
printf("(b) Value of series capacitor = %.3f pF\n\n",C*10^12)
printf("(c) Q of the circuit at resonance = %.3f\n\n",Q)
printf("(d) 3 dB bandwidth of the circuit = %.3f Mhz\n\n",B/10^6)
```

Scilab code Exa 5.3 Example 3

```
//Chapter 5, Problem 3
clc
f1=260*10^6                                     //frequency in hertz
f2=100*10^6                                     //frequency in hertz
A=40                                            //minimum attenuation in dB

//calculation
fr=f1/f2
n=A/(20*log10(fr))

printf("Number of arms = %f\n i.e 5 arms",n)
```

Scilab code Exa 5.4 Example 4

```
//Chapter 5, Problem 4
clc
z0=50                             //characteristic impedance in ohm
fp=500*10^6                       //passband limit frequency in hertz

//Butterworth normalised values
g1=0.618
g2=1.618
g3=2
g4=1.618
g5=0.618
```

```
w=2*%pi*fp

//calculation of component values
c1=g1/(w*z0)
l2=g2*z0/w
c3=g3/(w*z0)
l4=g4*z0/w
c5=g5/(w*z0)

printf("Component values are\n\tC1 = %.2f pF",c1
    *10^12)
printf("\n\tL2 = %.2f nH\n\tC3 = %.2f pF\n\tL4 = %.2
    f nH\n\tC5 = %.2f pF",l2*10^9,c3*10^12,l4*10^9,c5
    *10^12)
```

Scilab code Exa 5.6 Example 6

```
//Chapter 5, Problem 6
clc
z0=50                               //characteristic
    impedance in ohm
fp=500*10^6                         //passband limit
    frequency in hertz

//Butterworth normalised values
g1=1/0.618
g2=1/1.618
g3=1/2
g4=1/1.618
g5=1/0.618
w=2*%pi*fp

//calculation of component values
l1=g1*z0/(w)
c2=g2/(w*z0)
```

```
l3=g3*z0/w
c4=g4/(w*z0)
l5=g5*z0/(w)

printf("Component values are\n\tL1 = %.2f nH",l1
    *10^9)
printf("\n\tC2 = %.2f pF\n\tL3 = %.2f nH\n\tC4 = %.2
    f pF\n\tL5 = %.2f nH",c2*10^12,l3*10^9,c4*10^12,
    l5*10^9)
```

Scilab code Exa 5.8 Example 8

```
//Chapter 5, Problem 8
clc
z0=50                                   //characteristic
    impedance in ohm
fb=525e6
fa=475e6
fp=fb-fa                                //passband limit
    frequency in hertz

f0=sqrt(fb*fa)

//Butterworth normalised values
g1=0.618
g2=1.618
g3=2
g4=1.618
g5=0.618
w=2*%pi*fp

//calculation of component values
c1=g1/(w*z0)
l2=g2*z0/w
c3=g3/(w*z0)
```

```
l4=g4*z0/w
c5=g5/(w*z0)

printf("Component values are\n\tC1 = %.2f pF",c1*10^12)
printf("\n\tL2 = %.2f nH\n\tC3 = %.2f pF\n\tL4 = %.2f nH\n\tC5 = %.2f pF",l2*10^9,c3*10^12,l4*10^9,c5*10^12)
```

Scilab code Exa 5.10 Example 10

```
//Chapter 5, Problem 10
clc
z0=50                                    //characteristic impedance
fb=525e6
fa=475e6
fp=fb-fa                                 //passband limit frequency

f0=sqrt(fb*fa)

//Butterworth normalised values
g1=1/0.618
g2=1/1.618
g3=1/2
g4=1/1.618
g5=1/0.618
w=2*%pi*fp

//calculation of component values
l1=g1*z0/(w)
c2=g2/(w*z0)
l3=g3*z0/w
c4=g4/(w*z0)
```

```
l5=g5*z0/(w)

printf("Component values are\n\tL1 = %f nH",l1*10^9)
printf("\n\tC2 = %f pF\n\tL3 = %f nH\n\tC4 = %f pF\n
    \tL5 = %f nH",c2*10^12,l3*10^9,c4*10^12,l5*10^9)
```

Scilab code Exa 5.11 Example 11

```
//Chapter 5, Problem 11
clc
z0=50                             //characteristic
    impedance
fp=50*10^6                        //passband limit
    frequency
//from Figure 5.39, it is seen that about five arms
    will be required
n=5


//Butterworth normalised values
g1=1.1468
g2=1.3721
g3=1.9760
g4=1.3712
g5=1.1468
w=2*%pi*fp

//calculation of component values
c1=g1/(w*z0)
l2=g2*z0/w
c3=g3/(w*z0)
l4=g4*z0/w
c5=g5/(w*z0)
printf("(a) Number of arms of low pass filter = %.2f
     \n",n)
```

```
printf("Component values are\n\tC1 = %.2f pF",c1
    *10^12)
printf("\n\tL2 = %.2f nH\n\tC3 = %.2f pF\n\tL4 = %.2
    f nH\n\tC5 = %.2f pF",l2*10^9,c3*10^12,l4*10^9,c5
    *10^12)
```

Scilab code Exa 5.12 Example 12

```
//Chapter 5, Problem 12
clc
z0=75                           //characteristic
    impedance
fp=500*10^6                     //passband limit
    frequency
f1=260*10^6                     //frequency in hertz
//from Figure 5.39, it is seen that about five arms
    will be required
n=5

//Butterworth normalised values
g1=1/1.382
g2=1/1.326
g3=1/2.209
g4=1/1.326
g5=1/1.382
w=2*%pi*fp

//calculation of component values
l1=g1*z0/(w)
c2=g2/(w*z0)
l3=g3*z0/w
c4=g4/(w*z0)
l5=g5*z0/(w)
printf("(a) Number of arms of low pass filter = %.2f
     \n",n)
```

```
printf("(b) Component values are\n\tL1 = %f nH",l1
    *10^9)
printf("\n\tC2 = %f pF\n\tL3 = %f nH\n\tC4 = %f pF\n
    \tL5 = %f nH",c2*10^12,l3*10^9,c4*10^12,l5*10^9)
```

Scilab code Exa 5.13 Example 13

```
//Chapter 5, Problem 13
clc
n1=16                              //no of turns on
    primary
n2=8                               //no of turns on
    secondary
zs=16                              //terminating
    resistance

//calculation of effective resistance
zp=zs*(n1/n2)^2
printf("Effective resistance at the primary = %d ohm
    ",zp)
```

Scilab code Exa 5.14 Example 14

```
//Chapter 5, Problem 14
clc
n1=160                                 //no of turn
n2=40                                  //no of turns
n3=8                                   //no of turns
n4=150                                 //no of turns
n5=50                                  //no of turns
rl=2000                                //load
    resistance in ohms

```

```
//calculation
rl1=((n1+n2)/n3)^2*rl
req=(n2/(n1+n2))^2*rl1
rl2=((n4+n5)/n3)^2*rl
req2=(n5/(n4+n5))^2*rl2
printf("(a)Transistor load impedance at resonance =
   %d Kohm\n\n",req/1000)
printf("(b)New transistor load impedance at
   resonance = %.2f Kohm\n\n",req2/1000)
```

Scilab code Exa 5.15 Example 15

```
//Chapter 5, Problem 15
clc
n1=160                                  //no of turn
n2=40                                   //no of turns
n3=8                                    //no of turns
n4=150                                  //no of turns
n5=50                                   //no of turns
rl=2000                                 //load
   resistance in ohms
rt=100e3                                //output
   impedance of transistor
Q=100                                   //Q factor
Ct=180*10^-12                              //capacitance
    in farad
f=465e3                                     //resonant
   frequency

rl1=((n1+n2)/n3)^2*rl
rtr=((n1+n2)/n2)^2*rt
rckt=Q/(2*%pi*Ct*f)
req=rl1*rckt/(rl1+rckt)
rl2=((n2/(n1+n2))^2)*req

```

```
printf("Transistor load impedance at resonance = %.2
    f Kohm\n\n",rl2/1000)
```

Scilab code Exa 5.16 Example 16

```
//Chapter 5, Problem 16
clc
req=125e3                          //effective
    resistance in ohm
f=465e3                            //resonant frequency
     in hertz
L=650e-6                           //tuning inductance
    in inductance

//calculation
Q=req/(2*%pi*f*L)
B=f/Q
printf("Q = %.1f \n\n Bandwidth = %d Hz",Q,B)
```

Scilab code Exa 5.17 Example 17

```
//Chapter 5, Problem 17
clc
z1=22e3                      //reactance in ohm
c1=10                        //capacitance in picofarad
c2=100                      //capacitance in picofarad

//calculation
z2=z1*(c1/(c1+c2))
printf("tranformed value of reactance = %d Kohm",z2
    /1000)
```

Scilab code Exa 5.18 Example 18

```
//Chapter 5, Problem 18
clc
f=100e6                                 //frequency in hertz
cp=100e-12                              //capacitance in
   farad
rp=15e3                                 //resistance in ohm

//calculation
qp=2*%pi*f*cp*rp
printf("Quality factor Qp = %.2f",qp)
```

Scilab code Exa 5.19 Example 19

```
//Chapter 5, Problem 19
clc
f=800e3                                 //frequency in hertz
Ls=365e-6                              //capacitance in
   farad
Rs=8                                 //resistance in ohm

//calculation
Qs=(2*%pi*f*Ls)/Rs
printf("Quality factor Qs = %d",Qs)
```

Scilab code Exa 5.20 Example 20

```
//Chapter 5, Problem 20
```

```
clc
f=10e6                          //frequency in hertz
Ls=15e-6                       //capacitance in farad
Rs=2                          //resistance in ohm

//calculation
Qs=(2*%pi*f*Ls)/Rs
Rp=Rs*(1+(Qs^2))
Lp=((1+Qs^2)/Qs^2)*Ls

printf("Resistance Rp = %d Kohm\n\n Inductance Lp =
    %d uH\n\n Quality factor Qp = %d",Rp/1000,Lp
    *10^6,Qs)
```

Scilab code Exa 5.21 Example 21

```
//Chapter 5, Problem 21, figure 5.55
clc
Rp=500                        //equals to load
    resistance
Rs=50                         //equals to generator
    resistance
f=100e6                       //frequency in hertz

w=2*%pi*f
Qs=sqrt((Rp/Rs)-1)
Ls=(Rs*Qs)/w
Xs=w*Ls
Ca=1/(w*Xs)
Lp=((1+Qs^2)/Qs^2)*Ls
printf("Capacitor Ca = %.2f pF\n\nInductor Lp = %.2f
    nH",Ca/10^-12,Lp/10^-9)
```

Scilab code Exa 5.22 Example 22

```
//Chapter 5, Problem 22, figure 5.58
clc
f=100e6                                    //supply
   frequency in hertz
Rs=50                                      //resistance in
   ohms
Csh=42e-12                                 //shunt
   capacitance in ohm
Rl=500                                     //load
   resistance in ohm
Rp=Rl

//calculation
w=2*%pi*f
Qs=sqrt((Rp/Rs)-1)
Ls=(Rs*Qs)/w
Xs=w*Ls
Ca=1/(w*Xs)
Lp=((1+Qs^2)/Qs^2)*Ls

L=1/(w^2*Csh)
Lcom=(Lp*L)/(Lp+L)

printf("Matching network component value are,\n Ca =
    %.1f pF \n L (combined) = %d nH\n\n",Ca*10^12,
   Lcom*10^9)
disp("For the final network, shown in figure 5.61")
```

Scilab code Exa 5.23 Example 23

```
//Chapter 5, Problem 23, figure 5.65
clc
Rs=100                          //resistance in ohm
```

```
Rl=1000                              //resistance in ohm
Q=15                                 //Q factor

//calculation
Rv=Rl/(Q^2+1)
Xp2=Rl/Q
Xs2=Q*Rv
Q1=sqrt((Rs/Rv)-1)
Xp1=Rs/Q1
Xs1=Q1*Rv

printf("Zs = %d ohm\nXp1 = %.3f ohm \nXs1 = %.3f ohm
    \n",Rs,Xp1,Xs1)
printf("Xs2 = %.3f ohm\n Xp2 = %.3f ohm\n Zl = %d
    ohm\n\n",Xs2,Xp2,Rl)
disp("Four types of matching network is shown in
    figure 5.66, 5.67, 5.68, 5.69.")
```

Scilab code Exa 5.24 Example 24

```
//Chapter 5, Problem 24, figure 5.72
clc
Rs=10                              //resistance in ohm
Rl=50                              //resistance in ohm
Q=10                                 //Q factor

//calculation
Rv=Rs*(Q^2+1)
Xs1=Q*Rs
Xp1=Rv/Q
Q2=sqrt((Rv/Rl)-1)
Xp2=Rv/Q2
Xs2=Q2*Rl

printf("Zs = %d ohm\nXp1 = %.3f ohm \nXs1 = %.3f ohm
```

```
    \n",Rs,Xp1,Xs1)
printf("Xs2 = %.3f ohm\n Xp2 = %.3f ohm\n Zl = %d
    ohm\n\n",Xs2,Xp2,Rl)
disp("Four types of matching network is shown in
    figure 5.66, 5.67, 5.68, 5.69.")
```

Chapter 6

High frequency transistor amplifiers

Scilab code Exa 6.1 Example 1

```
//Chapter 6, Problem 1
clc
hfe=200                              //dc current gain
vcc=10                               //supply voltage
vbe=0.7                              //base to emitter voltage
ic=1e-3                              //collector current
vc=5                                 //collector voltage

//calculation of operating point
ib=ic/hfe
rf=(vc-vbe)/ib
rc=(vcc-vc)/(ic+ib)
printf("Operating point Rc = %.3f Kohm",rc/1000)
```

Scilab code Exa 6.2 Example 2

```
//Chapter 6, Problem 2
clc
hfe=250                                    //dc current gain
vcc=24                                     //supply voltage
vbe=0.7                                    //base to emitter voltage
ic=2e-3                                    //collector current
vc=12                                       //collector voltage

//calculation of operating point
ib=ic/hfe
rf=(vc-vbe)/ib
rc=(vcc-vc)/(ic+ib)
printf("Operating point Rc = %.3f Kohm",rc/1000)
```

Scilab code Exa 6.3 Example 3

```
//Chapter 6, Problem 3
clc
vbe=0.7                                    //base to emitter voltage
vcc=20                                     //supply voltage
vc=10                                       //collector voltage
ic=5e-3                                    //collector current
```

```
hfe=150                                  //dc current
   gain
vbb=2
ibb=1e-3

//calculating the biasing resistors
ib=ic/hfe
rb=(vbb-vbe)/ib
r1=vbb/ibb
rf=((vc-vbb)/(ibb+ib))
rc=((vcc-vc)/(ic+ib+ibb))

disp("Biasing resistors is given by")
printf("R1 = %d Kohm\nRb = %.2f Kohm\nRc = %.2f Kohm
   \nRf = %.2f Kohm",r1/1000,rb/1000,rc/1000,rf
   /1000)
```

Scilab code Exa 6.4 Example 4

```
//Chapter 6, Problem 4
clc
vbe=0.7                                  //base to
   emitter voltage
ic=1e-3                                 //collector
   current
vc=6                                    //collector
   voltage
hf1=100                                  //dc current
   gain
hf2=250                                  //dc current
   gain
vcc=12                                   //supply voltage
vbb=1.5
ibb=0.5e-3

```

```
//calculating the biasing resistors
hfe=sqrt(hf1*hf2)
ib=ic/hfe
rb=((vbb-vbe)/ib)
r1=vbb/ibb
rf=((vc-vbb)/(ibb+ib))
rc=((vcc-vc)/(ic+ib+ibb))

disp("Biasing resistor is given by\n")
printf("R1 = %d Kohm\nRb = %.2f Kohm\nRc = %.2f Kohm
    \nRf = %.2f Kohm",r1/1000,rb/1000,rc/1000,rf
    /1000)
```

Scilab code Exa 6.5 Example 5

```
//Chapter 6, Problem 5
clc
vbe=0.7                                    //base to
    emitter voltage
ic=10e-3                                   //collector
    current
vc=10                                      //collector
    voltage
vcc=20                                     //supply voltage
hfe=50                                    //dc current gain


//calculating the biasing resistors
ie=ic                                      //assuming for
    high gain transistor
ve=(10/100)*vcc
re=ve/ie
rc=((vcc-vc)/ic)
ib=ic/hfe
vbb=ve+vbe
```

```
ibb=10*ib
r2=vbb/ibb
r1=((vcc-vbb)/(ibb+ib))

disp("Biasing resistor is given by\n")
printf("R1 = %d ohm\nR2 = %.2f ohm\nRc = %.2f ohm\
    nRe = %.2f ohm",r1,r2,rc,re)
```

Scilab code Exa 6.6 Example 6

```
//Chapter 6, Problem 6
clc
vbe=0.7                                    //base to
    emitter voltage
ic=1e-3                                   //collector
    current
vc=6                                      //collector
    voltage
vcc=12                                     //supply voltage
hf1=100                                    //dc current
    gain
hf2=250                                    //dc current
    gain
ibb=0.5e-3

//calculating the biasing resistors
ie=ic
hfe=sqrt(hf1*hf2)
ve=(10/100)*vcc
re=ve/ie
rc=((vcc-vc)/ic)
ib=ic/hfe
vbb=ve+vbe
r2=vbb/ibb
r1=((vcc-vbb)/(ibb+ib))
```

```
disp("Biasing resistor is given by\n")
printf("R1 = %.2f Kohm\nR2 = %.2f Kohm\nRc = %.2f
    Kohm\nRe = %.2f Kohm",r1/1000,r2/1000,rc/1000,re
    /1000)
```

Scilab code Exa 6.7 Example 7

```
//Chapter 6, Problem 7, figure 6.13
clc
vcc=24                                          //supply
    voltage
vds=10                                          //drain to
    source voltage
id=5e-3                                         //drain
    current
vgs=2.3                                         //gate to
    source voltage
vs=2.3                                          //source
    voltage
vp=-8                                           //pinch-off
    voltage
idss=10e-3                                      //
    drain source current when the gate and source
    are shorted

//calculating the biasing resistors
rs=vgs/id
vd=vds+vs
rd=(vcc-vd)/id
vgs=vp*(1-sqrt(id/idss))

disp("Since IG = 0, RG = 1 Mohm (approx)")
printf("Rs = %.2f ohm\nRd = %.2f ohm\n\n",rs,rd)
```

Scilab code Exa 6.8 Example 8

```
//Chapter 6, Problem 8
clc
id=2e-3                                  //drain current in ampere
vds=12                                   //drain to source voltage
vcc=24                                   //supply voltage
idss=8e-3                                //drain source current when the gate and source are shorted
vp=-6                                    //pinch-off voltage

//calculating the biasing resistors
vgs=vp*(1-sqrt(id/idss))
rs=-vgs/id
vs=-vgs
vd=vds+vs
rd=((vcc-vd)/id)

disp("Since IG = 0, RG = 1 Mohm (approx)")
printf("Rs = %d ohm\nRd = %.2f ohm\n\n",rs,rd)
```

Scilab code Exa 6.9 Example 9

```
//Chapter 6, Problem 9
clc
id=-5e-3                                 //drain current in ampere
vds=-10                                  //drain to source voltage
```

```
vcc=-24                                          //supply voltage
idss=8e-3                                        //drain source current when the gate and source are shorted
vp=-6                                            //pinch-off voltage
vgs=2.3                                          //gate to source voltage

//calculating the biasing resistors
rs=-vgs/id
vs=-vgs
vd=vds+vs
rd=((vcc-vd)/id)

disp("Since IG = 0, RG = 1 Mohm (approx)")
printf("\nRs = %.2f ohm\nRd = %.2f ohm\n\n",rs,rd)
```

Scilab code Exa 6.10 Example 10

```
//Chapter 6, Problem 10
clc
id=-2e-3                                          //drain current in ampere
vds=-8                                           //drain to source voltage
vcc=-14                                           //supply voltage
vs=2.1                                              //source voltage
vp=5                                                //pinch-off voltage
idss=-6e-3                                        //drain source current when the gate and source are shorted
```

```

//calculating the biasing resistors
vgs=vp*(1-sqrt(id/idss))
rs=-vgs/id
vd=-vds+vs
rd=((vcc+vd)/id)

disp("Since IG = 0, RG = 1 Mohm (approx)")
printf("\nRs = %d ohm\nRd = %.2f ohm\n\n",rs,rd)
```

Scilab code Exa 6.11 Example 11

```
//Chapter 6, Problem 11
clc
id=5e-3                                  //drain current in ampere
vds=10                                   //drain to source voltage
vcc=18                                   //supply voltage
vs=0.1*vcc                               //source voltage
vgs=3.2                                  //gate to source voltage
r2=220e3                                 //resistance in ohm based upon d.c. input resistance needs

//calculating the biasing resistors
rs=vs/id
vg=vgs+vs
r1=(r2*(vcc-vg)/vg)
vd=vds+vs
rd=((vcc-vd)/id)

printf("\nRs = %d ohm\nRd = %.2f ohm\n\n",rs,rd)
```

```
printf("\nR1 = %d Kohm\nR2 = %.2f Kohm\n\n",r1/1000,r2/1000)
```

Scilab code Exa 6.12 Example 12

```
//Chapter 6, Problem 12
clc
id=2e-3                                           //drain current in ampere
vds=6                                             //drain to source voltage
vcc=12                                            //supply voltage
vs=0.1*vcc                                        //source voltage
vgs=1.8                                           //gate to source voltage
r2=220e3                                          //resistance in ohm based upon d.c. input resistance needs

//calculating the biasing resistors
rs=vs/id
vg=vgs+vs
r1=(r2*(vcc-vg)/vg)
vd=vds+vs
rd=((vcc-vd)/id)

printf("\nRs = %d ohm\nRd = %.2f ohm\n\n",rs,rd)
printf("\nR1 = %d Kohm\nR2 = %.2f Kohm\n\n",r1/1000,r2/1000)
```

Scilab code Exa 6.13 Example 13

```
//Chapter 6, Problem 13
clc
funcprot(0)
//using Y-parameters given in the case study
Yin=(18.33+%i*11.59)*10^-3                    //in complex form
y21=(1.09-%i*17.51)*10^-3                     //in complex form
y22=(0.3+%i*1.57)*10^-3                       //in complex form
Yl=3.33e-3

//defining a funcion
deff('[b]=imp1(d)','b={(real(d)^2)+(imag(d)^2)}^0.5');
deff('[c]=imp2(e)','c=atan(imag(e)/real(e))*180/%pi');

a=y21*Yl
b=Yin*(y22+Yl)

//calling a function
[a1]=imp1(a)
[a2]=imp2(a)
[b1]=imp1(b)
[b2]=imp2(b)

Ai1=a1/b1
Ai2=a2-b2

printf("Ai (magnitude) = %.2f amp\n\n",Ai1)
printf("Ai (angle) = %.2f degree",Ai2)
```

Scilab code Exa 6.14 Example 14

```
//Chapter 6, Problem 14
clc
//transistor parameter
yi=(16+%i*11.78)*10^-3                    //in complex
    form
y0=(1.55+%i*5.97)*10^-3                   //in complex
    form
gi=16e-3                                  //input
    conductance
go=0.19e-3                                //output
    conductance
yr_mag=1.55e-3                            //magnitude of
    yr parameter
yr_ang=258                                //angle of yr
    parameter
yf_mag=45e-3                              //magnitude of
    yf parameter
yf_ang=285                                 //angle of yf
    parameter

//calculation of stabilty factor
a=yr_mag*yf_mag
b=(2*gi*go)+(yr_mag*yf_mag)
C=a/b

printf("Linvill stability factor C = %.2f ",C)
```

Scilab code Exa 6.15 Example 15

```
//Chapter 6, Problem 15
clc
funcprot(0)
// A = p2z(R,Theta) - Convert from polar to
    rectangular form.
//    R is a matrix containing the magnitudes
```

```
//     Theta is a matrix containing the phase angles (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

// [R, Theta] = z2p(A) - Display polar form of complex matrix.
function [R, Theta] = z2p(A)
     Theta = atan(imag(A),real(A))*180/%pi;
     R=sqrt(real(A)^2+imag(A)^2)
endfunction

//transistor parameter
yi=(4.8+%i*4.52)*10^-3                    //in complex form
y0=(0.05+%i*2.26)*10^-3                   //in complex form
gi=4.8e-3                                  //input conductance
go=0.05e-3                                //output conductance
yr=p2z(0.90e-3,265)                          //in polar form to complex form
yf=p2z(61e-3,325)                           //in polar form to complex form

[yrm,yra]=z2p(yr)                             //in complex form to polar form
[yfm,yfa]=z2p(yf)                             //in complex form to polar form

Zs=50+%i*0
Zl=1000+%i*0
Ys=1/Zs
Yl=1/Zl
a=2*(gi+Ys)*(go+Yl)
b=(yfm*yrm)+real(yr*yf)
```

```
K=a/b
printf("Stern stability factor , K = %.2f ",K)
```

Scilab code Exa 6.16 Example 16

```
//Chapter 6, Problem 16
clc
//transistor Y parameter
yi=(16+%i*11.78)*10^-3                    //in complex form
yf_mag=45e-3
yf_ang=285
yr_mag=1.55e-3
yr_ang=258
yo=0.19+%i*5.97                          //in complex form
gi=16e-3                                 //input conductance in siemens
go=0.19e-3                               //output conductance in siemens

//calculating maximum available gain
MAG=yf_mag^2/(4*gi*go)
MAG_db=10*log10(MAG)
printf("Maximum available gain = %.2f \n\n",MAG)
printf("Maximum available gain in dB = %.2f dB",
   MAG_db)
```

Scilab code Exa 6.17 Example 17

```
//Chapter 6, Problem 17
clc
```

```
// A = p2z(R,Theta) - Convert from polar to rectangular form.
//     R is a matrix containing the magnitudes
//     Theta is a matrix containing the phase angles (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

//transistor Y parameter
yi=(17.37+%i*11.28)*10^-3                          //in complex form
yr_mag=1.17e-3
yf_mag=130.50e-3
yr=p2z(1.17e-3,-91)
yf=p2z(130.50e-3,-69)
yo=(0.95+%i*3.11)*10^-3                            //in complex form
f=300e6                                            //frequency in hertz
Vce=5                                              //base to emitter voltage
Ic=2e-3                                            //collector current
gi=17.37e-3                                        //input conductance
go=0.95e-3                                         //output conductance

//to calculate linvill stability factor
a=yf_mag*yr_mag
b=(2*gi*go)-real(yf*yr)
c=a/b

//to calculate maximum available gain
MAG=yf_mag^2/(4*gi*go)
MAG_db=10*log10(MAG)

```

```
//to calculate conjugate input admittance
m=sqrt(((2*gi*go)-real(yf*yr))^2-(yf_mag*yr_mag)^2)
n=2*go
Gs=m/n
Bs=-imag(yi)+(imag(yf*yr)/(2*go))
Gsi=Gs+%i*Bs

//to calculate conjugate output admittance
Gl=Gs*go/gi
Bl=-imag(yo)+(imag(yf*yr)/(2*gi))
Gsl=Gl+%i*Bl

//to calculate Stern stability factor
u=2*(gi+Gs)*(go+Gl)
v=(yf_mag*yr_mag)+real(yf*yr)
K=u/v

printf("(1) Linvill stability factor C = %.2f\n\n",c)
printf("(2) Maximum available gain (MAG) = %.2f dB\n\n",MAG_db)
printf("(3) Conjugate input admittance \n\treal = %.2f mS \n\timaginary = %.2f mS\n\n",real(Gsi)*1000,imag(Gsi)*1000)
printf("(4) Conjugate output admittance \n\treal = %.2f mS \n\timaginary = %.2f mS\n\n",real(Gsl)*1000,imag(Gsl)*1000)
printf("(5) Stern stability factor K = %.2f",K)
```

Scilab code Exa 6.18 Example 18

```
//Chapter 6, Problem 18
clc
funcprot(0)
// A = p2z(R,Theta) - Convert from polar to
```

```
   rectangular form.
//     R is a matrix containing the magnitudes
//     Theta is a matrix containing the phase angles (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

// [R1, Theta1] = z2p(A1) - Display polar form of complex matrix.
function [R1, Theta1] = z2p(A1)
     Theta1 = atan(imag(A1),real(A1))*180/%pi;
     R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction


//transistor Y parameter
yi=(17.37+%i*11.28)*10^-3                           //in complex form
yr_mag=1.17e-3
yf_mag=130.50e-3
yr=p2z(1.17e-3,-91)
yf=p2z(130.50e-3,-69)
yo=(0.95+%i*3.11)*10^-3                             //in complex form
f=300e6                                             //frequency in hertz
Vce=5                                               //base to emitter voltage
Ic=2e-3                                             //collector current
gi=17.37e-3                                         //input conductance
go=0.95e-3                                          //output conductance

//to calculate conjugate input admittance
m=sqrt(((2*gi*go)-real(yf*yr))^2-(yf_mag*yr_mag)^2)
```

```
n=2*go
Gs=m/n
Bs=-imag(yi)+(imag(yf*yr)/(2*go))
Gsi=Gs+%i*Bs

//to calculate conjugate output admittance
Gl=Gs*go/gi
Bl=-imag(yo)+(imag(yf*yr)/(2*gi))
Gsl=Gl+%i*Bl


num=4*Gs*Gl*(yf_mag)^2
den=((yi+Gsi)*(yo+Gsl))-(yf*yr)
[denm,dena]=z2p(den)
Gt=num/denm^2
Gt_db=10*log10(Gt)
printf("Transducer gain = %.2f dB",Gt_db)
```

Scilab code Exa 6.19 Example 19

```
//Chapter 6, Problem 18
clc
// A = p2z(R,Theta) - Convert from polar to
   rectangular form.
//    R is a matrix containing the magnitudes
//    Theta is a matrix containing the phase angles
   (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

//transistor parameter
yi=(2.25+%i*7.2)*10^-3
yr=p2z(0.70e-3,-85.9)
yf=p2z(44.72e-3,-26.6)
```

```
yo=(0.4+%i*1.9)*10^-3
yr_mag=0.70e-3
yf_mag=44.72e-3
Rs=250
Gs=1/Rs
K=3                                         //stern stability factor
gi=2.25e-3                                  //input conductance
go=0.4e-3                                   //output conductance
a=K*((yr_mag*yf_mag)+real(yf*yr))
b=2*(gi+Gs)
Gl=(a/b)-go

Bl=-imag(yo)
Yl=Gl+%i*Bl

yin=yi-((yr*yf)/(yo+Yl))

Bs=-imag(yin)
Ys=Gs+%i*Bs

num=4*Gs*Gl*(yf_mag)^2
den=real(((yi+Ys)*(yo+Yl))-(yf*yr))^2
Gt=num/den
Gt_db=10*log10(Gt)

printf("(a) Load admittance Yl \n\treal = %.2f mS \n\timaginary = %.2f mS\n\n",real(Yl)*1000,imag(Yl)*1000)
printf("(b) Source admittance Ys \n\treal = %.2f mS\n\timaginary = %.2f mS\n\n",real(Ys)*1000,imag(Ys)*1000)
printf("(c) Transducer gain = %.2f dB",Gt_db)
```

Chapter 7

Microwave amplifiers

Scilab code Exa 7.1 Example 1

```
//Chapter 7, Problem 1
clc
funcprot(0)
// A = p2z(R,Theta) - Convert from polar to
   rectangular form.
//    R is a matrix containing the magnitudes
//    Theta is a matrix containing the phase angles
   (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

// [R, Theta] = z2p(A) - Display polar form of
   complex matrix.
function [R, Theta] = z2p(A)
     Theta = atan(imag(A),real(A))*180/%pi;
     R=sqrt(real(A)^2+imag(A)^2)
endfunction

//transistor s-parameter
s11=p2z(0.3,160)
```

```
s12=p2z(0.03,62)
s21=p2z(6.1,65)
s22=p2z(0.40,-38)
R=50                          //resistance in ohms
f=150e6                       //frequency in hertz
vce=12                        //base to emitter voltage
ic=8e-3                       //collector current

Ds=(s11*s22)-(s12*s21)
[Dmag,Dang]=z2p(Ds)
[s11m,s11a]=z2p(s11)
[s22m,s22a]=z2p(s22)
[s21m,s21a]=z2p(s21)
[s12m,s12a]=z2p(s12)
K=(1+Dmag^2-s11m^2-s22m^2)/(2*s21m*s12m)
B1=1+s11m^2-s22m^2-Dmag^2
MAG=10*log10(s21m/s12m)+10*log10(K-sqrt(K^2-1))
C2=s22-(Ds*conj(s11))
[C2m,C2a]=z2p(C2)
B2=1+s22m^2-s11m^2-Dmag^2
reflm=(B2-sqrt(B2^2-4*C2m^2))/(2*C2m)
refla=-C2a
refl=p2z(reflm,refla)
refs=conj(s11+((s12*s21*refl)/(1-(s22*refl))))
[refsm,refsa]=z2p(refs)

//since we get  source and load reflection coefficient. we plot this source reflection coefficient on smith chart for getting input matching network in figure 7.1. By plotting , we get Arc AB = shunt C = j1.33 S and Arc BC = series L = j0.34 ohm.
y=1.33
r=0.34
C1=y/(2*%pi*f*R)
L1=r*R/(2*%pi*f)

```

```
//we plot this load reflection coefficient on smith
   chart for getting input matching network in
   figure 7.2. By plotting , Arc AB = series C =
     j1.1 ohm and Arc BC = shunt L =    j0.8 S.
y1=0.8
r1=1.1
C2=1/(2*%pi*f*R*r1)
L2=R/(2*%pi*f*y1)

printf("For input matching network,\n\n")
printf("C1 = %.2f pF\n",C1*10^12)
printf("L1 = %.2f nH\n\n",L1*10^9)
printf("For output matching network,\n\n")
printf("C2 = %.2f pF\n",C2*10^12)
printf("L2 = %.2f nH\n\n",L2*10^9)
printf("The completed design (minus biasing network)
    is shown in Figure 7.3")
```

Scilab code Exa 7.2 Example 2

```
//Chapter 7, Problem 2
clc
funcprot(0)
// A = p2z(R,Theta) - Convert from polar to
   rectangular form.
//    R is a matrix containing the magnitudes
//    Theta is a matrix containing the phase angles
   (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

// [R, Theta] = z2p(A) - Display polar form of
   complex matrix.
function [R, Theta] = z2p(A)
```

```
    Theta = atan(imag(A),real(A))*180/%pi;
    R=sqrt(real(A)^2+imag(A)^2)
endfunction

//transistor s-parameter
s11=p2z(0.3,160)
s12=p2z(0.03,62)
s21=p2z(6.1,65)
s22=p2z(0.40,-38)
R=50                          //resistance in ohms
f=150e6                       //frequency in hertz
vce=12                        //base to emitter voltage
ic=8e-3                       //collector current

Ds=(s11*s22)-(s12*s21)
[Dmag,Dang]=z2p(Ds)
[s11m,s11a]=z2p(s11)
[s22m,s22a]=z2p(s22)
[s21m,s21a]=z2p(s21)
[s12m,s12a]=z2p(s12)
K=(1+Dmag^2-s11m^2-s22m^2)/(2*s21m*s12m)
B1=1+s11m^2-s22m^2-Dmag^2
MAG=10*log10(s21m/s12m)+10*log10(K-sqrt(K^2-1))
C2=s22-(Ds*conj(s11))
[C2m,C2a]=z2p(C2)
B2=1+s22m^2-s11m^2-Dmag^2
reflm=(B2-sqrt(B2^2-4*C2m^2))/(2*C2m)
refla=-C2a
refl=p2z(reflm,refla)
refs=conj(s11+((s12*s21*refl)/(1-(s22*refl))))
[refsm,refsa]=z2p(refs)

a=s21m^2*(1-refsm^2)*(1-reflm^2)
b=((1-s11*refs)*(1-s22*refl))-(s12*s21*refl*refs)
[bm,ba]=z2p(b)
Gt=a/bm^2
Gt_db=10*log10(Gt)
```

```
printf("Transducer gain = %.2f dB",Gt_db)
```

Scilab code Exa 7.3 Example 3

```
//Chapter 7, Problem 3
clc
funcprot(0)
// A = p2z(R,Theta) - Convert from polar to
   rectangular form.
//    R is a matrix containing the magnitudes
//    Theta is a matrix containing the phase angles
   (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

// [R1, Theta1] = z2p(A1) - Display polar form of
   complex matrix.
function [R1, Theta1] = z2p(A1)
     Theta1 = atan(imag(A1),real(A1))*180/%pi;
     R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction

//transistor s-parameter
s11=p2z(0.3,140)
s12=p2z(0.03,65)
s21=p2z(2.1,62)
s22=p2z(0.40,-38)

f=5e9                                         //
   frequency in hertz
vce=15                                        //base
   to emitter voltage
ic=10e-3                                      //
   collector current in ampere
```

```

Ds=(s11*s22)-(s12*s21)
[Dmag,Dang]=z2p(Ds)
[s11m,s11a]=z2p(s11)
[s22m,s22a]=z2p(s22)
[s21m,s21a]=z2p(s21)
[s12m,s12a]=z2p(s12)
K=(1+Dmag^2-s11m^2-s22m^2)/(2*s21m*s12m)
B1=1+s11m^2-s22m^2-Dmag^2
MAG=10*log10(s21m/s12m)+10*log10(K-sqrt(K^2-1))

printf("Maximum available gain (MAG) = %.1f dB ",MAG)
```

Scilab code Exa 7.4 Example 4

```
//Chapter 7, Problem 4
clc
funcprot(0)
// A = p2z(R,Theta) - Convert from polar to rectangular form.
//    R is a matrix containing the magnitudes
//    Theta is a matrix containing the phase angles (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

// [R1, Theta1] = z2p(A1) - Display polar form of complex matrix.
function [R1, Theta1] = z2p(A1)
     Theta1 = atan(imag(A1),real(A1))*180/%pi;
     R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction

```

```
//transistor s-parameter
s11=p2z(0.3,140)
s12=p2z(0.03,65)
s21=p2z(2.1,62)
s22=p2z(0.40,-38)
refs=p2z(0.463,-140)
refl=p2z(0.486,38)
[s11m,s11a]=z2p(s11)
[s22m,s22a]=z2p(s22)
[s21m,s21a]=z2p(s21)
[s12m,s12a]=z2p(s12)
[refsm,refsa]=z2p(refs)
[reflm,refla]=z2p(refl)

//calculation
a=(s21m^2)*(1-refsm^2)*(1-reflm^2)
b=((1-(s11*refs))*(1-(s22*refl))-(s12*s21*refl*refs)
   )^2
Gt=a/real(b)
Gtl=10*log10(Gt)
printf("Amplifier transducer gain = %.2f dB ",Gtl)
```

Scilab code Exa 7.5 Example 5

```
//Chapter 7, Problem 5
clc
funcprot(0)
// A = p2z(R,Theta) - Convert from polar to
   rectangular form.
//    R is a matrix containing the magnitudes
//    Theta is a matrix containing the phase angles
   (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction
```

```

// [R1, Theta1] = z2p(A1) - Display polar form of complex matrix.
function [R1, Theta1] = z2p(A1)
     Theta1 = atan(imag(A1),real(A1))*180/%pi;
     R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction

//transistor s-parameter
s11=p2z(0.28,-58)
s12=p2z(0.08,92)
s21=p2z(2.1,65)
s22=p2z(0.8,-30)
f=1e9                                    //frequency in hertz
vce=15                                   //collector to emitter voltage
ic=5e-3                                  //collector current in ampere
Zs=35-%i*60                              //source impedance in ohm
Zl=50-%i*50                              //load impedance in ohm
K=1.168                                  //Rollett stability factor
g=7.94                                   //desired gain
R=50                                     //resistance in ohm

[s11m,s11a]=z2p(s11)
[s22m,s22a]=z2p(s22)
[s21m,s21a]=z2p(s21)
[s12m,s12a]=z2p(s12)

Ds=s11*s22-s12*s21
[Dsm,Dsa]=z2p(Ds)
D2=s22m^2-Dsm^2
C2=s22-(Ds*conj(s11))
```

```
G=g/s21m^2
ro=(G*conj(C2))/(1+(D2*G))
po=sqrt(1-(2*K*s12m*s21m*G)+(s12m*s21m)^2*G^2)/(1+(
    D2*G))

//The Smith chart construction is shown in Figure
    7.5. The transistor s output network must
    transform the actual load impedance into a value
    that falls on the constant gain 9 dB circle. By
    plotting, we get Arc AB = series C =   j2.0 ohm
    and Arc BC = shunt L =   j0.41 S
r=2
y=0.4
C1=1/(2*%pi*f*r*R)
L1=R/(2*%pi*f*y)

//For a conjugate match at the input to the
    transistor, the desired source reflection
    coefficient must be calculated as follows
refl=p2z(0.82,13)                                    //point
     C in figure 7.5
refs=conj(s11+((s12*s21*refl)/(1-(s22*refl))))
[refsm,refsa]=z2p(refs)

//The point is plotted as point D in Figure 7.6. The
     actual normalised source impedance is plotted at
     point A (0.7     j1.2) ohm. The input network
    must transform the actual impedance at point A to
     the desired impedance at point D. we get Arc AB
    = shunt C2 = j0.63 S, Arc BC = series L2 = j1.08
    ohm, Arc CD = shunt C3 = j2.15 S

y1=0.63
r1=1.08
y2=2.15

C2=y1/(2*%pi*f*R)
```

```
L2=r1*R/(2*%pi*f)
C3=y2/(2*%pi*f*R)

printf("For output matching network,\n\n")
printf("C1 = %.2f pF\n",C1*10^12)
printf("L1 = %.1f nH\n\n",L1*10^9)
printf("For input matching network,\n\n")
printf("C2 = %.1f pF\n",C2*10^12)
printf("L2 = %.1f nH\n\n",L2*10^9)
printf("C3 = %.1f pF\n",C3*10^12)
printf("The completed design (minus biasing network) is shown in Figure 7.7")
```

Scilab code Exa 7.6 Example 6

```
//Chapter 7, Problem 6
clc
funcprot(0)
// A = p2z(R,Theta) - Convert from polar to rectangular form.
//    R is a matrix containing the magnitudes
//    Theta is a matrix containing the phase angles (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

// [R1, Theta1] = z2p(A1) - Display polar form of complex matrix.
function [R1, Theta1] = z2p(A1)
     Theta1 = atan(imag(A1),real(A1))*180/%pi;
     R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction

//transistor s-parameter
```

```
s11=p2z(0.28,-58)
s12=p2z(0.08,92)
s21=p2z(2.1,65)
s22=p2z(0.8,-30)
Ds=p2z(0.333,-60.66)
C2=p2z(0.719,-33.42)
D2=0.529                                                  //angle in
    degree
K=1.168                                                   //Rollett
    stability factor
A=6.31                                                    //desired
    gain
[s11m,s11a]=z2p(s11)
[s22m,s22a]=z2p(s22)
[s21m,s21a]=z2p(s21)
[s12m,s12a]=z2p(s12)

//calculating the radius of constant gain circle of
    9 dB
G=A/s21m^2
ro=conj(G*C2)/(1+(D2*G))
[rom,roa]=z2p(ro)
po=sqrt(1-(2*K*G*s12m*s21m)+((s12m*s21m)^2*G^2))
    /(1+(D2*G))

printf("Constant gain circle of 8 dB = %.3f ",po)
```

Scilab code Exa 7.7 Example 7

```
//Chapter 7, Problem 7
clc
funcprot(0)
// A = p2z(R,Theta) - Convert from polar to
    rectangular form.
//    R is a matrix containing the magnitudes
```

```
//     Theta is a matrix containing the phase angles (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

// [R1, Theta1] = z2p(A1) - Display polar form of complex matrix.
function [R1, Theta1] = z2p(A1)
     Theta1 = atan(imag(A1),real(A1))*180/%pi;
     R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction

//transistor S-parameter
s11=p2z(0.4,280)
s12=p2z(0.048,65)
s21=p2z(5.4,103)
s22=p2z(0.78,345)
f=200e6                                    //frequency in hertz
vce=6                                      //collector to emitter voltage
ic=5e-3                                    //collector current in ampere

[s11m,s11a]=z2p(s11)
[s22m,s22a]=z2p(s22)
[s21m,s21a]=z2p(s21)
[s12m,s12a]=z2p(s12)

Ds=(s11*s22)-(s12*s21)
[Dmag,Dang]=z2p(Ds)
K=(1+Dmag^2-s11m^2-s22m^2)/(2*s21m*s12m)
C1=s11-(Ds*conj(s22))
[C1m,C1a]=z2p(C1)
C2=s22-(Ds*conj(s11))
[C2m,C2a]=z2p(C2)
rs1=conj(C1)/(s11m^2-Dmag^2)
```

```
[rs1m,rs1a]=z2p(rs1)
ps1=s12*s21/(s11m^2-Dmag^2)
[ps1m,ps1a]=z2p(ps1)

rs2=conj(C2)/(s22m^2-Dmag^2)
[rs2m,rs2a]=z2p(rs2)
ps2=s12*s21/(s22m^2-Dmag^2)
[ps2m,ps2a]=z2p(ps2)

printf("Centre of input stability circle (magnitude)
    = %.3f \n\t\t\t\t(angle) = %.2f degree\n",rs1m,
   rs1a)
printf("Radius of input  stability circle = %.2f \n\
   n",ps1m)
printf("Centre of output stability circle (magnitude
   ) = %.3f \n\t\t\t\t(angle) = %.2f degree\n",rs2m,
   rs2a)
printf("Radius of output  stability circle = %.2f \n
   \n",ps2m)
printf("Using these points, plotting a circle on
   smith chart as shown on Fig 7.9,\n,with the help
   of these we will get\n")
printf("load reflection coefficient = 0.89 (
   magnitude) , 70 (degree)\n Source reflection
   coefficient = 0.678 (magnitude), 79.4 (degree)")
```

Scilab code Exa 7.8 Example 8

```
//Chapter 7, Problem 8
clc
funcprot(0)
// A = p2z(R,Theta) - Convert from polar to
   rectangular form.
//     R is a matrix containing the magnitudes
//     Theta is a matrix containing the phase angles
```

```
(in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

// [R1, Theta1] = z2p(A1) - Display polar form of complex matrix.
function [R1, Theta1] = z2p(A1)
     Theta1 = atan(imag(A1),real(A1))*180/%pi;
     R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction

//transistor S-parameter
s11=p2z(0.4,280)
s12=p2z(0.048,65)
s21=p2z(5.4,103)
s22=p2z(0.78,345)
ro=p2z(0.287,24)                                    //centre of gain circle
po=0.724                                            //radius of 12dB constant gain circle
f=200e6                                             //frequency in hertz
vce=6                                               //base to emitter voltage
ic=5e-3                                             //collector current
A=15.85                                             //gain desired
K=0.802                                             //Rollett's stability factor
C2=p2z(0.651,24.1)

[s11m,s11a]=z2p(s11)
[s22m,s22a]=z2p(s22)
[s21m,s21a]=z2p(s21)
[s12m,s12a]=z2p(s12)
Ds=(s11*s22)-(s12*s21)
```

```
[Dmag,Dang]=z2p(Ds)
G=A/s21m^2
D2=s22m^2-Dmag^2
ro=C2*G/(1+(D2*G))
[rom,roa]=z2p(ro)
po1=sqrt(1-(2*K*G*abs(s12m*s21m))+((abs(s12m*s21m))
    ^2*G^2))/(1+(D2*G))

printf("ro (magnitude) = %.3f\n\t (angle) = %.3f
    degree\n\n",rom,roa)
printf("po = %.3f \n\n",po1)
disp("There the values are correct")
```

Scilab code Exa 7.9 Example 9

```
//Chapter 7, Problem 9
clc
funcprot(0)
// A = p2z(R,Theta) - Convert from polar to
    rectangular form.
//    R is a matrix containing the magnitudes
//    Theta is a matrix containing the phase angles
    (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

// [R1, Theta1] = z2p(A1) - Display polar form of
    complex matrix.
function [R1, Theta1] = z2p(A1)
     Theta1 = atan(imag(A1),real(A1))*180/%pi;
     R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction

//transistor S parameter
```

```
s11=p2z(0.4,280)
s12=p2z(0.048,65)
s21=p2z(5.4,103)
s22=p2z(0.78,345)
rs=p2z(0.678,79.4)                              //source
    reflection coefficient
rl=p2z(0.89,70)                                  //load
   reflection coefficient

Rs=conj(s11+((s12*s21*rl)/(1-(s22*rl))))
[Rsm,Rsa]=z2p(Rs)
printf("Source reflection coefficient, magnitude = %
   .3f \n\t\t\t angle = %.1f degree",Rsm,Rsa)
```

Scilab code Exa 7.10 Example 10

```
//Chapter 7, Problem 10
clc
funcprot(0)
// A = p2z(R,Theta) - Convert from polar to
   rectangular form.
//    R is a matrix containing the magnitudes
//    Theta is a matrix containing the phase angles
   (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

// [R1, Theta1] = z2p(A1) - Display polar form of
   complex matrix.
function [R1, Theta1] = z2p(A1)
     Theta1 = atan(imag(A1),real(A1))*180/%pi;
     R1=sqrt(real(A1)^2+imag(A1)^2)
endfunction

```

```
//transistor S-parameter
s11=p2z(0.35,165)
s12=p2z(0.035,58)
s21=p2z(5.9,66)
s22=p2z(0.46,-31)
refs=p2z(0.68,142)                              //source reflection coefficient
f=300e6                                         //frequency in hertz
vce=12                                          //base to emitter voltage
ic=4e-3                                       //collector current in ampere
[s11m,s11a]=z2p(s11)
[s22m,s22a]=z2p(s22)
[s21m,s21a]=z2p(s21)
[s12m,s12a]=z2p(s12)

Ds=(s11*s22)-(s12*s21)
[Dmag,Dang]=z2p(Ds)
K=(1+Dmag^2-s11m^2-s22m^2)/(2*s21m*s12m)

y=1.65
r=0.85
y1=0.62
r1=1.2
R=50

//The design values of the input matching network are shown in Figures 7.10. By plotting, we get Arc AB = shunt C = j1.65 S and Arc BC = series L = j0.85 ohm
C1=y/(2*%pi*f*R)
L1=(r*R)/(2*%pi*f)

refl=conj(s11+((s12*s21*refs)/(1-(s22*refs))))

//The design values of the input matching network
```

```
    are shown in Figures 7.12. By plotting, we get
    Arc AB = shunt L =   j0.72 S and Arc BC = series
     C =   j1.07 ohm

C2=1/(2*%pi*f*r1*R)
L2=R/(2*%pi*f*y1)

printf("For input matching network,\n\n")
printf("C1 = %.2f pF\n",C1*10^12)
printf("L1 = %.2f nH\n\n",L1*10^9)
printf("For output matching network,\n\n")
printf("C2 = %.2f pF\n",C2*10^12)
printf("L2 = %.2f nH\n\n",L2*10^9)
printf("The completed design is shown in Figure 7.12
    ")
```

Scilab code Exa 7.12 Example 12

```
//Chapter 7, Problem 12
clc

Vg=200e-3                           //generator voltage
Zg=2000                             //resistance in ohm
//transistor Y-parameter
Y11=1/1200
Y12=0
Y21=70/1200
Y22=1/40000
Yl=1/1000
Y1=[Y11 Y12;Y21 Y22]
Yf=[1/10000 -1/10000;-1/10000 1/10000]

//calculation
Yc=Y1+Yf
dely=Yc(1,1)*Yc(2,2)-Yc(1,2)*Yc(2,1)
```

```
Zin=(Yc(2,2)+Yl)/(dely+(Yc(1,1)*Yl))
vin=(Zin*Vg)/(Zin+Zg)
Av=-Yc(2,1)/(Yc(2,2)+Yl)
vout=vin*Av

printf("(a) Input impedance (Zin) = %.1f ohm\n\n",
   Zin)
printf("(b) Gain (Av) of the circuit = %.2f\n\n",Av)
printf("(c) Output voltage = %d mV\n\n",vout*10^3)
```

Chapter 8

Oscillators and frequency synthesizers

Scilab code Exa 8.1 Example 1

```
//Chapter 8, Problem 1
clc
L=630e-9                                    //inductance in
   henry
C=400e-12                                   //capacitance in
    farad

//calculating the resonating frequency
fos=1/(2*%pi*sqrt(L*C))

printf("Resonant frequency = %.3f MHz",fos/10^6)
```

Scilab code Exa 8.2 Example 2

```
//Chapter 8, Problem 2
clc
```

```
r1=100e3                                //resistance in ohm
r2=10e3                                 //resistance in ohm
c1=10e-9                                //capacitance in farad
c2=100e-9                               //capacitance in farad

//calculation
w=sqrt(1/(c1*c2*r1*r2))
f=w/(2*%pi)
g=1+(r1/r2)+(c2/c1)

printf("(a) Frequency of oscillation = %.2f Hz\n\n",f)
printf("(b) Minimum gain of the amplifier = %d ",g)
```

Scilab code Exa 8.3 Example 3

```
//Chapter 8, Problem 3
clc
c1=10e-12                               //capacitance in farad
c2=100e-12                              //capacitance in farad
f=100e6                                 //frequency in hertz

//calculation
w=2*%pi*f
L=(1/w^2)*((1/c1)+(1/c2))
g=1+(c2/c1)

printf("(a) Value of inductor = %.2f nH\n\n",L*10^9)
```

```
printf("(b) Minimum voltage gain = %d ",g)
```

Scilab code Exa 8.4 Example 4

```
//Chapter 8, Problem 4
clc
ct=15e-12                                    //capacitance
    in farad
c1=47e-12                                    //capacitance
    in farad
c2=100e-12                                    //capacitance
    in farad
L=300e-9                                    //inductance in
    henry

//calculation
w1=(1/(L*ct))*(1+(ct/c1)+(ct/c2))
w=sqrt(w1)
fos=w/(2*%pi)

printf("Approximate frequency = %.2f MHz",fos/10^6)
```

Scilab code Exa 8.5 Example 5

```
//Chapter 8, Problem 5, figure 8.28
clc
Rf=9e3                                    //resistance in
    ohm
R1=1e3                                    //resistance in
    ohm
Kphi=0.12                                    //transfer gain
    in volt/radian
```

```
Ko=-40e3                              //transfer gain in hertz/volt
fi=100e3                              //input frequency in hertz
fo=120e3                              //oitput frequency in hertz from VCO

//calculation
Ka=(Rf/R1)+1
Kl=Kphi*Ka*Ko*2*%pi
Kl_dB=real(20*log10(Kl))
fd=fi-fo
Vo=fd/Ko
Vd=Vo/Ka
theta=Vd/Kphi
fd1=-Kl/(2*%pi)
Vd1=Kphi*%pi/2

printf("(a) Voltage gain (ka) for the op-amp = %d\n\n",Ka)
printf("(b) Loop gain (kL) = %.1f s^-1\n\t\t= %.1f dB\n\n",Kl,Kl_dB)
printf("(c) With S1 open as shown, there is no phase lock and the beat frequency = %d kHz\n\n",fd/1000)
printf("(d)(i) fo = %d kHz\n",fi/1000)
printf("   (ii) Static phase error = %.3f rad\n",theta)
printf("   (iii) Vo = %.1f V\n\n",Vo)
printf("(e) Hold-in range Df = %.2f kHz\n\n",fd1/1000)
printf("(f) Maximum value of vd = %.3f V d.c",Vd1)
```

Scilab code Exa 8.6 Example 6

```
//Chapter 8, Problem 6
clc
f1=70e6                    //section 1, frequency in
   hertz
f2=5e6                     //section 2, frequency in
   hertz
f3=400e3                    //section 3, frequency in
   hertz
f4=80e3                    //section 4, frequency in
   hertz

//calculation
F3h=f3+f4
F3l=f3-f4

F2h=f2+F3h
F2l=f2-F3h

F1h=f1+F2h
F1l=f1-F2h

printf("Mixer 3 : %d Khz and %d Khz\n",F3h/1000,F3l
   /1000)
printf("After filter 3 : %d Khz\n\n",F3h/1000)
printf("Mixer 2 : %.2f Mhz and %.2f Mhz\n",F2h/10^6,
   F2l/10^6)
printf("After filter 2 : %.2f Mhz\n\n",F2h/10^6)
printf("Mixer 1 : %.2f Mhz and %.2f Mhz\n",F1h/10^6,
   F1l/10^6)
printf("After filter 1 : %.2f Mhz\n\n",F1h/10^6)
```

Scilab code Exa 8.7 Example 7

```
//Chapter 8, Problem 7
clc
```

```
fl=511e6                    //lowest frequency at the divider
res=1e6                     //resolution
fh=887e6                    //highest frequency at the divider

//calculation of division factor
N=fl/res
N2=fh/res

printf("Lowest value of division factor, N = %d \n\n",N)
printf("Highest value of division factor, N = %d ",N2)
```

Scilab code Exa 8.8 Example 8

```
//Chapter 8, Problem 8
clc
fl=18.7e6                   //lowest frequency at the divider
fo=50e3                     //divider output
fl2=38.7e6                  //highest frequency at the divider

//calculation of division factor
N=fl/fo
N2=fl2/fo

printf("Lowest value of division factor, N = %d \n\n",N)
printf("Highest value of division factor, N = %d ",N2)
```