

Scilab Textbook Companion for  
High Frequency and Microwave Engineering  
by E. Da Silva<sup>1</sup>

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

July 31, 2019

<sup>1</sup>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	4
1 Basic features of radio communication systems	5
2 Transmission lines	11
3 Smith charts and scattering parameters	21
5 Amplifier basics	36
6 High frequency transistor amplifiers	51
7 Microwave amplifiers	70
8 Oscillators and frequency synthesizers	89

# List of Scilab Codes

Exa 1.1	Example 1 . . . . .	5
Exa 1.2	Example 2 . . . . .	6
Exa 1.3	Example 3 . . . . .	6
Exa 1.4	Example 4 . . . . .	7
Exa 1.5	Example 5 . . . . .	8
Exa 1.6	Example 6 . . . . .	8
Exa 1.7	Example 7 . . . . .	9
Exa 1.8	Example 8 . . . . .	9
Exa 1.9	Example 9 . . . . .	10
Exa 2.2	Example 2 . . . . .	11
Exa 2.4	Example 4 . . . . .	11
Exa 2.5	Example 5 . . . . .	13
Exa 2.6	Example 6 . . . . .	14
Exa 2.7	Example 7 . . . . .	14
Exa 2.8	Example 8 . . . . .	14
Exa 2.9	Example 9 . . . . .	15
Exa 2.10	Example 10 . . . . .	16
Exa 2.11	Example 11 . . . . .	16
Exa 2.12	Example 12 . . . . .	17
Exa 2.13	Example 13 . . . . .	18
Exa 2.14	Example 14 . . . . .	19
Exa 2.15	Example 15 . . . . .	19
Exa 3.1	Example 1 . . . . .	21
Exa 3.2	Example 2 . . . . .	22
Exa 3.3	Example 3 . . . . .	23
Exa 3.4	Example 4 . . . . .	24
Exa 3.6	Example 6 . . . . .	25
Exa 3.8	Example 8 . . . . .	27

Exa 3.9	Example 9 . . . . .	28
Exa 3.10	Example 10 . . . . .	28
Exa 3.11	Example 11 . . . . .	28
Exa 3.13	Example 13 . . . . .	29
Exa 3.15	Example 15 . . . . .	31
Exa 3.16	Example 16 . . . . .	32
Exa 3.17	Example 17 . . . . .	33
Exa 3.18	Example 18 . . . . .	34
Exa 5.1	Example 1 . . . . .	36
Exa 5.3	Example 3 . . . . .	37
Exa 5.4	Example 4 . . . . .	37
Exa 5.6	Example 6 . . . . .	38
Exa 5.8	Example 8 . . . . .	39
Exa 5.10	Example 10 . . . . .	40
Exa 5.11	Example 11 . . . . .	41
Exa 5.12	Example 12 . . . . .	42
Exa 5.13	Example 13 . . . . .	43
Exa 5.14	Example 14 . . . . .	43
Exa 5.15	Example 15 . . . . .	44
Exa 5.16	Example 16 . . . . .	45
Exa 5.17	Example 17 . . . . .	45
Exa 5.18	Example 18 . . . . .	46
Exa 5.19	Example 19 . . . . .	46
Exa 5.20	Example 20 . . . . .	46
Exa 5.21	Example 21 . . . . .	47
Exa 5.22	Example 22 . . . . .	48
Exa 5.23	Example 23 . . . . .	48
Exa 5.24	Example 24 . . . . .	49
Exa 6.1	Example 1 . . . . .	51
Exa 6.2	Example 2 . . . . .	52
Exa 6.3	Example 3 . . . . .	52
Exa 6.4	Example 4 . . . . .	53
Exa 6.5	Example 5 . . . . .	54
Exa 6.6	Example 6 . . . . .	55
Exa 6.7	Example 7 . . . . .	56
Exa 6.8	Example 8 . . . . .	57
Exa 6.9	Example 9 . . . . .	57
Exa 6.10	Example 10 . . . . .	58

Exa 6.11	Example 11 . . . . .	59
Exa 6.12	Example 12 . . . . .	60
Exa 6.13	Example 13 . . . . .	60
Exa 6.14	Example 14 . . . . .	61
Exa 6.15	Example 15 . . . . .	62
Exa 6.16	Example 16 . . . . .	64
Exa 6.17	Example 17 . . . . .	64
Exa 6.18	Example 18 . . . . .	66
Exa 6.19	Example 19 . . . . .	68
Exa 7.1	Example 1 . . . . .	70
Exa 7.2	Example 2 . . . . .	72
Exa 7.3	Example 3 . . . . .	74
Exa 7.4	Example 4 . . . . .	75
Exa 7.5	Example 5 . . . . .	76
Exa 7.6	Example 6 . . . . .	79
Exa 7.7	Example 7 . . . . .	80
Exa 7.8	Example 8 . . . . .	82
Exa 7.9	Example 9 . . . . .	84
Exa 7.10	Example 10 . . . . .	85
Exa 7.12	Example 12 . . . . .	87
Exa 8.1	Example 1 . . . . .	89
Exa 8.2	Example 2 . . . . .	89
Exa 8.3	Example 3 . . . . .	90
Exa 8.4	Example 4 . . . . .	91
Exa 8.5	Example 5 . . . . .	91
Exa 8.6	Example 6 . . . . .	92
Exa 8.7	Example 7 . . . . .	93
Exa 8.8	Example 8 . . . . .	94

# Chapter 1

## Basic features of radio communication systems

Scilab code Exa 1.1 Example 1

```
1 //Chapter 1, Example 1.1
2 clc
3 E=10e-6 //electric field
   strength in volts/metre
4 Z=377 //wave impedance
   in ohm
5 A=5 //area in metre
   square
6
7 //calculating magnetic field strength
8 H=E/Z
9 //calculating incident power
10 P=E*H*A
11
12 printf("(a) Magnetic field strength = %.3f nA/m\n\n",
   H*10^9)
13 printf("(b) Incident power on a recieving aerial = %
   .3f pW\n\n",P*10^12)
```

---



### Scilab code Exa 1.2 Example 2

```
1 //Chapter 1, Example 1.2
2 clc
3 E=10e-6 //electric field
   strength in volts/metre
4 Z=377 //wave impedance
   in ohm
5 P=2.65*10^-13 //power incident
   on a surface area 1m2
6
7 D1=10 //distance in
   kilometre
8 D2=100 //distance in
   kilometre
9 P1=(D1/D2)^2 //power density
   factor
10 PD=P1*P //power density in
   W/m2
11
12 Erms=sqrt(PD*Z) //Erms at 100Km
13 Hrms=Erms/Z //Hrms at 100Km
14 printf("(a) Erms at 100km = %.3 f microV/m\n\n",Erms
   *10^6)
15 printf("(b) Hrms at 100km = %.3 f nA/m\n\n",Hrms
   *10^9)
```

---

### Scilab code Exa 1.3 Example 3

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

```

3 Vo=100 //transmitter output in
  volts
4 Zo=72 //transmitter output
  impedance in ohm
5 Zin=72 //antenna input
  impedance in ohm
6 R=72 //radiation resistance
  in ohm
7 eff=100 //antenna efficiency
8
9 Vrms=Vo/2
10 Pr=(Vrms)^2/R
11
12 printf("Power radiated = %.2f W",Pr)

```

---

#### Scilab code Exa 1.4 Example 4

```

1 //Chapter 1, Example 1.4
2 clc
3 n=105 //no of coil turns
4 a=8*10^-5 //cross sectional
  area in metre square
5 ur=230 //relative
  permeability
6 uo=4*pi*10^-7 //permeability of
  air
7 Erms=10*10^-6 //electric field
  strength
8 f=10^6 //frequency in hertz
9
10 //calculation
11 erms=n*2*pi*f*ur*uo*Erms*a*cos(0)
12
13 printf("r.m.s open circuit voltage = %.2f microVolt"
  ,erms*10^6)

```

---

Scilab code Exa 1.5 Example 5

```
1 //Chapter 1, Example 1.5
2 clc
3 n=100 //no of coil turns
4 a=8*10^-5 //effective cross-
    sectional area in metre
5 ur=200 //relative
    permeability
6 uo=4*%pi*10^-7 //permeability, of
    air
7 ao=60*%pi/180 //angle of incidence
    of magnetic field
8 f=10^6 //frequency in hertz
9 E=100*10^-6 //electric field
    strength in V/m
10 z=377 //wave impedance in
    ohm
11
12 // calculation
13 w=2*%pi*f
14 erms=n*w*ur*uo*a*(E/z)*cos(ao)
15
16 printf("r.m.s open circuit voltage induced = %.3f uV
    ",erms*10^6)
```

---

Scilab code Exa 1.6 Example 6

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

```

4 Zo=75                                //input impedance of
    each receiver
5
6 //calculating the value of resistor
7 R=((n-1)/(n+1))*Zo
8
9 printf("Value of the matching resistor = %d ohm",R)

```

---

#### Scilab code Exa 1.7 Example 7

```

1 //Chapter 1, Example 1.7, figure 1.23
2 clc
3 n=4                                    //no of recievers
4 Zo=50                                  //input impedance of
    each receiver
5
6 //calculating the value of resistor
7 R=((n-1)/(n+1))*Zo
8
9 printf("Value of the matching resistor = %d ohm",R)

```

---

#### Scilab code Exa 1.8 Example 8

```

1 //Chapter 1, Example 1.8, figure 1.23
2 clc
3 n=3                                    //no of recievers
4 Zo=50                                  //input impedance of
    each receiver
5 Van=100*10^-6                          //open-circuit voltage
    in aerial
6
7 //calculation
8 R=((n-1)/(n+1))*Zo

```

```

9 V=1/(2*n)*Van
10
11 printf("(a) Value of the matching resistor = %d ohm\
n\n\n",R)
12 printf("(b) Voltage at receiver input terminal = %.3
f uV",V*10^6)

```

---

### Scilab code Exa 1.9 Example 9

```

1 //Chapter 1, Example 1.9, Figure 1.25
2 clc
3 r1=75 //network splitter
and termination impedance
4 r2=43 //matching network
5
6 //calculation
7 Voc=(r1/(r2+r1))*(((r1+r2)/2)/(((43+75)/2)+r1))
8 Zin=((r1+r2)/2)
9 Zsr=r2+(((r1+r2)*r1)/((r2+r1)+r1))
10
11 printf("(a) Ratio Vout/Voc = %.2 f \n\n",Voc)
12 printf("(b) Input impedance to the network = %d ohm\
n\n",Zin)
13 printf("(c) Reciever source impedance = %.2 f ohm\n\n
",Zsr)

```

---

# Chapter 2

## Transmission lines

Scilab code Exa 2.2 Example 2

```
1 //Chapter 2, Problem 2
2 clc
3 D=300e-3 //distance in meter
   between the two wire
4 d=4e-3 //diameter in meter of a
   conductor
5 e=1 //relative dielectric
6
7 //calculating the characteristic impedance of the
   type of parallel transmission line
8 z0=(276/sqrt(e))*log10(2*D/d)
9 printf("Characteristic impedance, Z0 = %d ohm",z0)
```

---

Scilab code Exa 2.4 Example 4

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

```

4 R=23 //resistance in ohm
5 G=4*10^-3 //conductance in
    siemens
6 L=125*10^-6 //inductance in
    henry
7 C=48*10^-9 //capacitance in
    farad
8
9 //list of frequencies in hertz
10 f1=100
11 f2=500
12 f3=15e3
13 f4=5e6
14 f5=10e6
15
16 deff(' [a]=imp(R,G,L,C,f)', 'a=sqrt((R+(%i*2*%pi*f*L))
    /(G+(%i*2*%pi*f*C)))');
17 deff(' [b]=imp1(d)', 'b={(real(d)^2)+(imag(d)^2)}^0.5'
    );
18 deff(' [c]=imp2(e)', 'c=atan(imag(e),real(e))');
19 [Z01]=imp(R,G,L,C,f1)
20 [Z_mag1]=imp1(Z01)
21 [Z_ang1]=imp2(Z01)
22 [Z02]=imp(R,G,L,C,f2)
23 [Z_mag2]=imp1(Z02)
24 [Z_ang2]=imp2(Z02)
25 [Z03]=imp(R,G,L,C,f3)
26 [Z_mag3]=imp1(Z03)
27 [Z_ang3]=imp2(Z03)
28 [Z04]=imp(R,G,L,C,f4)
29 [Z_mag4]=imp1(Z04)
30 [Z_ang4]=imp2(Z04)
31 [Z05]=imp(R,G,L,C,f5)
32 [Z_mag5]=imp1(Z05)
33 [Z_ang5]=imp2(Z05)
34
35 printf(" Characteristic impedance Z0 for a given
    frequency is ,\n\n")

```

```

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

```

---

#### Scilab code Exa 2.5 Example 5

```

1 //Chapter 2, Problem 5
2 clc
3 f=1.6*10^6 //frequency in
      hertz
4 Zoc_mag=900 //magnitude in
      ohm of open circuit impedance
5 Zoc_ang=-30 //angle in
      degree of open circuit impedance
6 Zsc_mag=400 //magnitude in
      ohm of short circuit impedance
7 Zsc_ang=-10 //angle in
      degree of short circuit impedance
8
9 //calculation of charactersitics impedance
10 Z0_mag=sqrt(Zoc_mag*Zsc_mag)
11 Z0_ang=Zoc_ang-Zsc_ang
12
13 printf("Z0 (magnitude) = %d ohm\n Z0(angle) = %d
      degree",Z0_mag,Z0_ang)

```

---



### Scilab code Exa 2.6 Example 6

```
1 //Chapter 2, Problem 6
2 clc
3 a=2 //attenuation
    constant
4 l=10 //length in Km
5
6 //calculation of loss in dB
7 a10=a*l
8 loss=8.686*a10
9
10 printf("Loss in dB = %.2f dB",loss)
```

---

### Scilab code Exa 2.7 Example 7

```
1 //Chapter 2, Problem 7
2 clc
3 Z1=80-%i*10 //load impedance in
    complex form
4 Z0=50 //characteristic
    impedance in ohm
5
6 //calculation of reflection coefficient
7 ref=(Z1-Z0)/(Z1+Z0)
8 ref_mag={real(ref)^2+(imag(ref)^2)}^0.5
9 ref_ang=atan(imag(ref)/real(ref))
10
11 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

```

1 //Chapter 2, Problem 8
2 clc
3 Z0=50 //
   characteristic impedance in ohm
4 //load impedance in ohm
5 Z11=50
6 Z12=0
7 Z13=75
8
9 //calculation of reflection coefficient
10 ref1=(Z11-Z0)/(Z11+Z0)
11 ref2=1
12 ref3=(Z12-Z0)/(Z12+Z0)
13 ref4=(Z13-Z0)/(Z13+Z0)
14 printf("(a) with Zl = 50, reflection coefficient is
   \n magnitude = %f \n angle = %d degree\n\n",ref1,
   ref1)
15 printf("(b) with Zl = open circuit , reflection
   coefficient is \n magnitude = %f \n angle = %d
   degree\n\n",ref2,angle=0)
16 printf("(c) with Zl = short circuit , reflection
   coefficient is \n magnitude = %f \n angle = %d
   degree\n\n",ref3,angle2=0)
17 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

```

1 //Chapter 2, Problem 9
2 clc
3 Vin=100 //incident voltage
4 Vref=10 //reflected voltage
5
6 //calculation of voltage standing wave ratio

```

```
7 Vswr=(Vin+Vref)/(Vin-Vref)
8
9 printf("VSWR = %.2 f",Vswr)
```

---

#### Scilab code Exa 2.10 Example 10

```
1 //Chapter 2, Problem 10
2 clc
3 ref=0.1 //reflection
   coefficient
4
5 //calculation of voltage standing wave ratio
6 Vswr=(1+ref)/(1-ref)
7
8 printf("VSWR = %.2 f",Vswr)
```

---

#### Scilab code Exa 2.11 Example 11

```
1 //Chapter 2, Problem 11
2 clc
3 Vswr=1.07 //voltage standing wave
   ratio
4 Z0=50 //characteristic
   impedance in ohm
5
6 //calculation of power reflected in percent
7 ref=(Vswr-1)/(Vswr+1)
8 Pref=(ref^2*100)
9
10 printf(" Reflected power = %.1f percentage of power
   incident",Pref)
```

---

### Scilab code Exa 2.12 Example 12

```
1 //Chapter 2, Problem 12
2 clc
3 funcprot(0)
4 R=23 //resistance in
   ohm
5 G=4*10^-3 //conductance in
   siemens
6 L=125*10^-6 //inductance in
   henry
7 C=48*10^-9 //capacitance in
   farad
8
9 //list of frequencies in hertz
10 f1=100
11 f2=500
12 f3=15000
13 f4=5*10^6
14 f5=10*10^6
15
16 //calculation of characteristic impedance by
   declaring the function
17 function [c,d,e,f]=myfct(u,v,x,y,z)
18   a=u+%i*2*%pi*z*v
19   b=x+%i*2*%pi*z*y
20   m=sqrt(a*b)
21   n=sqrt(a/b)
22   c=sqrt(real(m)^2+imag(m)^2)
23   d=atan(imag(m),real(m))
24   e=sqrt(real(n)^2+imag(n)^2)
25   f=atan(imag(n),real(n))
26 endfunction
27
```

```

28 [y1,z1,x1,v1]=myfct(R,L,G,C,f1)
29 [y2,z2,x2,v2]=myfct(R,L,G,C,f2)
30 [y3,z3,x3,v3]=myfct(R,L,G,C,f3)
31 [y4,z4,x4,v4]=myfct(R,L,G,C,f4)
32 [y5,z5,x5,v5]=myfct(R,L,G,C,f5)
33
34 printf("(a) for 100 Hz \n\tPropagation constant is \n
\t\t= %f (magnitude)\n\t\t= %f (angle)\n\n\t
\tCharacteristic impedance Z0 is \n\t\t= %f ohm(
magnitude)\n\t\t= %f (angle)\n\n",y1,z1,x1,v1)
35 printf("(b) for 500 Hz \n\tPropagation constant is \n
\t\t= %f (magnitude)\n\t\t= %f (angle)\n\n\t
\tCharacteristic impedance Z0 is \n\t\t= %f ohm(
magnitude)\n\t\t= %f (angle)\n\n",y2,z2,x2,v2)
36 printf("(c) for 15 KHz \n\tPropagation constant is \n
\t\t= %f (magnitude)\n\t\t= %f (angle)\n\n\t
\tCharacteristic impedance Z0 is \n\t\t= %f ohm(
magnitude)\n\t\t= %f (angle)\n\n",y3,z3,x3,v3)
37 printf("(d) for 5 MHz \n\tPropagation constant is \n\t
\t\t= %f (magnitude)\n\t\t= %f (angle)\n\n\t
\tCharacteristic impedance Z0 is \n\t\t= %f ohm(
magnitude)\n\t\t= %f (angle)\n\n",y4,z4,x4,v4)
38 printf("(e) for 10 MHz \n\tPropagation constant is \n
\t\t= %f (magnitude)\n\t\t= %f (angle)\n\n\t
\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

```

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

```

```

5
6 //calculation of input impedance
7 Zin=%i*Z0*tan(2*%pi*1)
8
9 printf("Input impedance = j%.1f ohm",imag(Zin))

```

---

#### Scilab code Exa 2.14 Example 14

```

1 //Chapter 2, Problem 14
2 clc
3 Z0=75 //
   characteristic impedance in ohm
4 l=1/5 //electrical
   length
5
6 //calculation of input impedance
7 Zin=-%i*Z0*cotg(2*%pi*1)
8
9 printf("Input impedance = %.1fj ohm",imag(Zin))

```

---

#### Scilab code Exa 2.15 Example 15

```

1 //Chapter 2, Problem 15
2 clc
3 Z1=20 //load impedance in
   ohm
4 Z0=90 //characteristic
   impedance in ohm
5 l=1/4 //electrical length
6
7 //calculation of input impedance
8 Zin=(Z0)^2/Z1
9 printf("Input impedance = %d ohm ",Zin);

```



# Chapter 3

## Smith charts and scattering parameters

Scilab code Exa 3.1 Example 1

```
1 //Chapter 3, Problem 1, Figure 3.4
2 clc
3 //https://atoms.scilab.org/toolboxes/microwave
4 //Download and install the Microwave toolbox from
   above link and load it from scilab menubar >
   Toolboxes > microwave
5
6
7 //Plot the smith chart
8 uW_display_smith([.2 .5 1 2 5],12);
9 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
10
11 R2=0
12 plot2d(real(R2),imag(R2),-1);
13
14 for n=1:length(Z)
15     Z1=50*Z(n)
16     G=(Z1-50)/(Z1+50);
```



```

17 plot2d(real(G), imag(G),-8);
18 xtitle("Smith Chart");
19 end;

```

---

### Scilab code Exa 3.2 Example 2

```

1 //Chapter 3, Problem 2
2 clc
3 //https://atoms.scilab.org/toolboxes/microwave
4 //Download and install the Microwave toolbox from
   above link and load it from scilab menubar >
   Toolboxes > microwave
5
6 funcprot(0)
7
8 // A = p2z(R,Theta) - Convert from polar to
   rectangular form.
9 // R is a matrix containing the magnitudes
10 // Theta is a matrix containing the phase angles
   (in degrees).
11 function [A] = p2z(R,Theta)
12 A = R*exp(%i*%pi*Theta/180);
13 endfunction
14
15 // [R1, Theta1] = z2p(A1) - Display polar form of
   complex matrix.
16 function [R1, Theta1] = z2p(A1)
17 Theta1 = atan(imag(A1),real(A1))*180/%pi;
18 R1=sqrt(real(A1)^2+imag(A1)^2)
19 endfunction
20
21 //Plot the smith chart
22 uW_display_smith([.2 .5 1 2 5],12);
23 Z=0.8-%i*1.6; //
   impedance

```

```

24 Z1=50*Z; //50 =
    characteristic impedance
25 [Zm,Za]=z2p(Z);
26 G=(Z1-50)/(Z1+50); //
    reflection coefficient
27 Ym=1/Zm //
    admittance (magnitude)
28 Ya=Za*(-1) //
    admittance (angle)
29 Y=p2z(Ym,Ya)
30 Y1=50*Y;
31 R=(Y1-50)/(Y1+50);
32 R2=0
33 plot2d(real(R2),imag(R2),-1);
34
35 plot2d(real(G),imag(G),-8);
36 plot2d(real(R),imag(R),-8);
37 xtitle("Smith Chart");
38 printf("Admittance value , Y = %.2 f + j%.1 f",real(Y),
    imag(Y));

```

---

### Scilab code Exa 3.3 Example 3

```

1 //Chapter 3, Problem 3
2 clc
3 //https://atoms.scilab.org/toolboxes/microwave
4 //Download and install the Microwave toolbox from
    above link and load it from scilab menubar >
    Toolboxes > microwave
5
6 //Plot the smith chart
7 uW_display_smith([.2 .5 1 2 5],12);
8 r=0.667 //
    radius of VSWR circle
9 Z=0.25-%i*0.5; //

```

```

        impedance
10 Z1=50*Z;                                //50 =
        characteristic impedance
11 G=(Z1-50)/(Z1+50);                      //
        reflection coefficient
12 R2=0
13 plot2d(real(R2),imag(R2),-1);
14 plot2d(real(G),imag(G),-8);
15
16 //Plot a VSWR circle of radius 0.667
17 x=linspace(0,2*pi,200);
18 plot2d(r*cos(x),r*sin(x))
19 xtitle("Smith Chart");
20
21
22 printf("From smith chart , The answer is %.3f (
        magnitude) and -124 degree (angle)",r)
23 disp("This is shown as point C in Figure 3.11.")

```

---

#### Scilab code Exa 3.4 Example 4

```

1 //Chapter 3, Problem 4
2 clc
3 //https://atoms.scilab.org/toolboxes/microwave
4 //Download and install the Microwave toolbox from
        above link and load it from scilab menubar >
        Toolboxes > microwave
5
6 //Plot the smith chart
7 uW_display_smith([.2 .5 1 2 5],12);
8 r=0.667                                  //
        radius of VSWR circle
9 Z=0.21+%i*0.21;                          //
        impedance
10 Z1=50*Z;                                //50 =

```

```

        characteristic impedance
11 G=(Z1-50)/(Z1+50); //
        reflection coefficient
12 R2=0
13 plot2d(real(R2), imag(R2), -1);
14 plot2d(real(G), imag(G), -8);
15
16 //Plot a VSWR circle of radius 0.667
17 x=linspace(0,2*%pi,200);
18 plot2d(r*cos(x),r*sin(x))
19 xtitle("Smith Chart");
20
21
22 printf("From smith chart , The answer is %.2f + j%.2f
        ",real(Z), imag(Z))
23 disp("This is shown as point E in Figure 3.11.")

```

---

### Scilab code Exa 3.6 Example 6

```

1 //Chapter 3, Problem 6
2 clc
3 //https://atoms.scilab.org/toolboxes/microwave
4 //Download and install the Microwave toolbox from
   above link and load it from scilab menubar >
   Toolboxes > microwave
5
6 funcprot(0)
7
8 // [R1, Theta1] = z2p(A1) - Display polar form of
   complex matrix.
9 function [R1, Theta1] = z2p(A1)
10     Theta1 = atan(imag(A1),real(A1))*180/%pi;
11     R1=sqrt(real(A1)^2+imag(A1)^2)
12 endfunction
13

```

```

14 function Zin = zin(d)
15     B=2*%pi*d
16     Zin=Zo*((Zl+(%i*Zo*tan(B)))/(Zo+(%i*Zl*tan(B))))
17 endfunction
18
19 Zo=50 //characteristic
    impedance
20 Zl=40-%i*80 //load impedance
21 //line
22 d1=0.096
23 d2=0.173
24 d3=0.206
25
26 refl=(Zl-Zo)/(Zl+Zo) //
    reflection coefficient
27 [reflm,refla]=z2p(refl)
28 SWR=(1+reflm)/(1-reflm) //
    standing wave ratio
29 Zin1=zin(d1)
30 Zin2=zin(d2)
31 Zin3=zin(d3)
32 ////load impedance is expressed in normalised form
33 a=Zl/Zo
34 d=0.25-%i*0.5
35 f=0.2+%i*0
36 h=0.2+%i*0.2
37 j=0.25-%i*0.5
38
39
40 //Plot the smith chart
41 uW_display_smith([.2 .5 1 2 5],12);
42 Z=[a d f h j]
43 for n=1:length(Z)
44     Z1=50*Z(n)
45     G=(Z1-50)/(Z1+50);
46 plot2d(real(G), imag(G),-8);
47 end;
48 R2=0

```

```

49 plot2d(real(R2), imag(R2), -1);
50
51 //Plot a VSWR circle of radius 0.667
52 x=linspace(0,2*%pi,200);
53 plot2d(0.66*cos(x),0.66*sin(x))
54 xtitle("Smith Chart");
55
56 printf("The input impedance Zin of the terminated
    line \n\n")
57 printf("(a) 0.096h = %.2f %.2fj\n\n",real(Zin1),imag
    (Zin1))
58 printf("(a) 0.173h = %.2f + %.2fj\n\n",real(Zin2),
    imag(Zin2))
59 printf("(a) 0.206h = %.2f + %.2fj\n\n",real(Zin3),
    imag(Zin3))

```

---

#### Scilab code Exa 3.8 Example 8

```

1 //Chapter 3, Problem 8
2 clc
3
4 j=0.25+(%i*0.5) //Normalised
    impedance from smith chart
5 Zo=50 //Characteristic
    impedance
6
7 //calculating line impedance at point J in Figure
    3.15
8 Z1=j*Zo
9
10 printf("Line impedance = %.1f + j%d",real(Z1),imag(
    Z1))

```

---

### Scilab code Exa 3.9 Example 9

```
1 //Chapter 3, Problem 9
2 clc
3 Z0=50 //characteristic
   impedance in ohm
4 Z10=5 //ratio of Z/Z0
5
6 //calculation
7 Z0t=sqrt(Z10)
8 Z0t1=Z0*Z0t
9 printf("Impedance = %.2f ohm",Z0t1)
```

---

### Scilab code Exa 3.10 Example 10

```
1 //Chapter 3, Problem 10
2 clc
3 Z1o=0.2 //ratio of Z1/Zo
   impedance in ohm
4 Zo=50 //output
   impedance in ohm
5
6 //calculation of characteristic impedance
7 Z0t=Zo*sqrt(Z1o)
8
9 printf("Characteristic impedance, Z0t = %d ohm",Z0t)
```

---

### Scilab code Exa 3.11 Example 11

```
1 //Chapter 3, Problem 11, Figure 3.16(a), 3.18
2 clc
3 Zs=50+%i*0 //impedance in ohm
4 Z1=100+%i*0 //impedance in ohm
```

```

5
6 //using one quarter wave transformer in the circuit
   of figure 3.16(a)
7 Z0=sqrt(Zs*Zl)
8
9 //Use two quarter-wave transformers as in Figure
   3.18
10 Z0t=sqrt(Zl/Zs)
11 Z0t2=60
12 Z0t1=Z0t2*Z0t
13
14 //using table 3.3
15 x=[0.6,0.8,1.0,1.2,1.4]
16 y=[-13.83,-19.28,-60,-19.28,-13.83]
17 clf;
18 x1=[0.6,0.8,1.0,1.2,1.4]
19 y1=[-18.81,-32.09,-38.69,-32.09,-18.81]
20 plot(x,y,".r")
21 plot(x1,y1,".b")
22 legend("one h/4","Two h/4")
23
24 xtitle("reflection coefficient vs frequency","
   frequency (GHz)","Reflection coefficient (dB)");
25 printf("(a) Matching network using one /4
   transformer,\n\t Z0 = %.2f ohm\n\n",Z0)
26 printf("(b) Matching network using two /4
   transformers,\n Ratio of Z0t1 and Z0t2 = %.3f\n",
   Z0t)
27 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

```

1 //Chapter 3, Problem 13
2 clc

```



```

3
4 //https://atoms.scilab.org/toolboxes/microwave
5 //Download and install the Microwave toolbox from
   above link and load it from scilab menubar >
   Toolboxes > microwave
6
7 funcprot(0)
8 // [R1, Theta1] = z2p(A1) - Display polar form of
   complex matrix.
9 function [R1, Theta1] = z2p(A1)
10     Theta1 = atan(imag(A1),real(A1))*180/%pi;
11     R1=sqrt(real(A1)^2+imag(A1)^2)
12 endfunction
13
14 zin=100           //input resistance
   in ohm
15 zo=50           //amplifier input
   resistance in ohm
16 c1=5e-12        //capacitance in
   farad
17 f=10^9          //frequency in
   hertz
18 d=1+(%i*2.3)    //point C
19 h=0-(%i*2.3)    //point E
20
21 //Calculation
22 Yo=1/zo
23 Yl=(1/zin)+( %i*2*%pi*f*c1)
24 Y=Yl/Yo
25
26 //Plot the smith chart
27 uW_display_smith([.2 .5 1 2 5],12);
28 Y1=50*Y;
29 R=(Y1-50)/(Y1+50);
30 R2=0
31 [Rm ,Ra]=z2p(R)
32 plot2d(real(R) , imag(R) , -8);
33 plot2d(real(R2) , imag(R2) , -1);

```

```

34 y=[d h]
35 for n=1:length(y)
36     y1=50*y(n)
37     R1=(y1-50)/(y1+50);
38 plot2d(real(R1), imag(R1),-8);
39 end;
40
41 //Plot a VSWR circle of radius 0.667
42 x=linspace(0,2*%pi,200);
43 plot2d(Rm*cos(x),Rm*sin(x))
44 xtitle("Smith chart")
45
46 printf("Yl/Yo = %.1f + j %.2f\n\n",real(Y),imag(Y))

```

---

#### Scilab code Exa 3.15 Example 15

```

1 //Chapter 3, Prblem 15, figure 3.30
2 clc
3 //from figure 3.30
4 zo=50 //in ohm
5 z1=50 //in ohm
6
7 //calculating the S parameter
8 z1=zo+z1
9 s11=(z1-zo)/(z1+zo)
10 z2=zo+z1
11 s22=(z2-zo)/(z2+zo)
12 s21=(2*z1)/(50+zo+z1)
13 s12=(2*z1)/(50+zo+z1)
14
15 s=[s11 s12;s21 s22]
16
17 printf("s11 (magnitude) = %.3f \n\t(angle) = 0
18     degree\n\n",s11)
18 printf("s12 (magnitude) = %.3f \n\t(angle) = 0

```

```

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

```

---

### Scilab code Exa 3.16 Example 16

```

1 //Chapter 3, Prblem 16, figure 3.32
2 clc
3 //from figure 3.32
4 zo=50 //in ohm
5 z1=50 //in ohm
6
7 //calculating the S parameter
8 z1=(zo*z1)/(zo+z1)
9 s11=(z1-zo)/(z1+zo)
10 z2=(zo*z1)/(zo+z1)
11 s22=(z2-zo)/(z2+zo)
12 s21=(2*z1)/(50+z1)
13 s12=(2*z2)/(50+z2)
14
15 s=[s11 s12;s21 s22]
16
17 printf("s11 (magnitude) = %.3 f \n\t(angle) = 180
    degree\n\n",-s11)
18 printf("s12 (magnitude) = %.3 f \n\t(angle) = 0
    degree\n\n",s12)
19 printf("s21 (magnitude) = %.3 f \n\t(angle) = 0
    degree\n\n",s21)
20 printf("s22 (magnitude) = %.3 f \n\t(angle) = 180
    degree\n\n",-s22)
21 disp(s,"S (magnitude) = ")

```

---

### Scilab code Exa 3.17 Example 17

```
1 //Chapter 3, Prblem 17, figure 3.34
2 clc
3 funcprot(0)
4 // [R1, Theta1] = z2p(A1) - Display polar form of
   complex matrix.
5 function [R1, Theta1] = z2p(A1)
6     Theta1 = atan(imag(A1),real(A1))*180/%pi;
7     R1=sqrt(real(A1)^2+imag(A1)^2)
8 endfunction
9
10
11 //from figure 3.34
12 zo=50 //in ohm
13 z1=50 //in ohm
14 r1=30
15 zai=%i*20
16 za=(r1*zo)/(r1+zo)
17 z1=za+zai
18 s11=(z1-zo)/(z1+zo)
19 z2=(zo+zai)*r1/(zo+zai+r1)
20 s22=(z2-zo)/(z2+zo)
21 s21=za*2/(za+(zo+zai))
22 s12=zo*0.75/(za+zo+zai)
23
24
25 [s11m, s11a]=z2p(s11)
26 [s22m, s22a]=z2p(s22)
27 [s21m, s21a]=z2p(s21)
28 [s12m, s12a]=z2p(s12)
29
30 ret_loss=-20*log10(s11m)
31 ins_loss=-20*log10(s21m)
```

```

32
33 printf("(a) S parameters is , \n")
34 printf("s11 (magnitude) = %.3 f \n\t(angle) = %.2 f
    degree\n\n",s11m,s11a)
35 printf("s12 (magnitude) = %.3 f \n\t(angle) = %.2 f
    degree\n\n",s12m,s12a)
36 printf("s21 (magnitude) = %.3 f \n\t(angle) = %.2 f
    degree\n\n",s21m,s21a)
37 printf("s22 (magnitude) = %.3 f \n\t(angle) = %.2 f
    degree\n\n",s22m,s22a)
38 printf("(b) Return loss = %.3 f dB\n\n",ret_loss)
39 printf("(c) Insertion loss = %.3 f dB\n\n",ins_loss)

```

---

### Scilab code Exa 3.18 Example 18

```

1 //Chapter 3, Prblem 18,
2 clc
3 funcprot(0)
4 // A = p2z(R,Theta) - Convert from polar to
    rectangular form.
5 // R is a matrix containing the magnitudes
6 // Theta is a matrix containing the phase angles
    (in degrees).
7 function [A] = p2z(R,Theta)
8 A = R*exp(%i*%pi*Theta/180);
9 endfunction
10
11 // [R1, Theta1] = z2p(A1) - Display polar form of
    complex matrix.
12 function [R1, Theta1] = z2p(A1)
13 Theta1 = atan(imag(A1),real(A1))*180/%pi;
14 R1=sqrt(real(A1)^2+imag(A1)^2)
15 endfunction
16
17

```

```

18 //transistor S-parameter
19 s11=p2z(0.12,-10)
20 s12=p2z(0.002,-78)
21 s21=p2z(9.8,160)
22 s22=p2z(0.01,-15)
23
24 [s11m,s11a]=z2p(s11)
25 [s22m,s22a]=z2p(s22)
26 [s21m,s21a]=z2p(s21)
27 [s12m,s12a]=z2p(s12)
28
29 vswr=(1+s11m)/(1-s11m)
30 ret_loss=-20*log10(s11m)
31 Fig=20*log10(s21m)
32 Rig=20*log10(s12m)
33
34 printf("(a) Input VSWR = %.2f\n\n",vswr)
35 printf("(b) Return loss (dB) = %.2f dB\n\n",ret_loss
    )
36 printf("(c) Forward insertion gain = %.2f dB\n\n",
    Fig)
37 printf("(d) Reverse insertion gain = %.2f dB\n\n",
    Rig)

```

---

# Chapter 5

## Amplifier basics

Scilab code Exa 5.1 Example 1

```
1 //Chapter 5, Problem 1
2 clc
3 R=3 //resistance in ohm
4 L=20*10^-9 //inductance in henry
5 f0=500e6 //frequency in hertz
6
7 //calculation
8 Z=R
9 C=(1/(2*%pi*f0*sqrt(L)))^2
10 Q=2*%pi*f0*L/R
11 B=f0/Q
12
13 printf("(a) Impedance at resonance = %d ohm\n\n",Z)
14 printf("(b) Value of series capacitor = %.3f pF\n\n"
    ,C*10^12)
15 printf("(c) Q of the circuit at resonance = %.3f\n\n"
    ",Q)
16 printf("(d) 3 dB bandwidth of the circuit = %.3f Mhz
    \n\n",B/10^6)
```

---

### Scilab code Exa 5.3 Example 3

```
1 //Chapter 5, Problem 3
2 clc
3 f1=260*106 //frequency
   in hertz
4 f2=100*106 //frequency
   in hertz
5 A=40 //minimum
   attenuation in dB
6
7 //calculation
8 fr=f1/f2
9 n=A/(20*log10(fr))
10
11 printf("Number of arms = %f\n i.e 5 arms",n)
```

---

### Scilab code Exa 5.4 Example 4

```
1 //Chapter 5, Problem 4
2 clc
3 z0=50 //characteristic
   impedance in ohm
4 fp=500*106 //passband limit
   frequency in hertz
5
6 //Butterworth normalised values
7 g1=0.618
8 g2=1.618
9 g3=2
10 g4=1.618
11 g5=0.618
```



```

12 w=2*%pi*fp
13
14 //calculation of component values
15 c1=g1/(w*z0)
16 l2=g2*z0/w
17 c3=g3/(w*z0)
18 l4=g4*z0/w
19 c5=g5/(w*z0)
20
21 printf("Component values are\n\tC1 = %.2f pF",c1
        *10^12)
22 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

```

1 //Chapter 5, Problem 6
2 clc
3 z0=50 //characteristic
        impedance in ohm
4 fp=500*10^6 //passband limit
        frequency in hertz
5
6 //Butterworth normalised values
7 g1=1/0.618
8 g2=1/1.618
9 g3=1/2
10 g4=1/1.618
11 g5=1/0.618
12 w=2*%pi*fp
13
14 //calculation of component values
15 l1=g1*z0/(w)
16 c2=g2/(w*z0)

```

```

17 l3=g3*z0/w
18 c4=g4/(w*z0)
19 l5=g5*z0/(w)
20
21 printf("Component values are\n\tL1 = %.2f nH",l1
      *10^9)
22 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

```

1 //Chapter 5, Problem 8
2 clc
3 z0=50 //characteristic
      impedance in ohm
4 fb=525e6
5 fa=475e6
6 fp=fb-fa //passband limit
      frequency in hertz
7
8 f0=sqrt(fb*fa)
9
10 //Butterworth normalised values
11 g1=0.618
12 g2=1.618
13 g3=2
14 g4=1.618
15 g5=0.618
16 w=2*%pi*fp
17
18 //calculation of component values
19 c1=g1/(w*z0)
20 l2=g2*z0/w
21 c3=g3/(w*z0)

```

```

22 l4=g4*z0/w
23 c5=g5/(w*z0)
24
25 printf("Component values are\n\tC1 = %.2f pF",c1
        *10^12)
26 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 5.10 Example 10

```

1 //Chapter 5, Problem 10
2 clc
3 z0=50 //characteristic
        impedance
4 fb=525e6
5 fa=475e6
6 fp=fb-fa //passband limit
        frequency
7
8 f0=sqrt(fb*fa)
9
10 //Butterworth normalised values
11 g1=1/0.618
12 g2=1/1.618
13 g3=1/2
14 g4=1/1.618
15 g5=1/0.618
16 w=2*%pi*fp
17
18 //calculation of component values
19 l1=g1*z0/(w)
20 c2=g2/(w*z0)
21 l3=g3*z0/w
22 c4=g4/(w*z0)

```

```

23 l5=g5*z0/(w)
24
25 printf("Component values are\n\tL1 = %f nH",l1*10^9)
26 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

```

1 //Chapter 5, Problem 11
2 clc
3 z0=50 //characteristic
impedance
4 fp=50*10^6 //passband limit
frequency
5 //from Figure 5.39, it is seen that about five arms
will be required
6 n=5
7
8
9 //Butterworth normalised values
10 g1=1.1468
11 g2=1.3721
12 g3=1.9760
13 g4=1.3712
14 g5=1.1468
15 w=2*%pi*fp
16
17 //calculation of component values
18 c1=g1/(w*z0)
19 l2=g2*z0/w
20 c3=g3/(w*z0)
21 l4=g4*z0/w
22 c5=g5/(w*z0)
23 printf("(a) Number of arms of low pass filter = %.2f
\n",n)

```

```

24 printf("Component values are\n\tC1 = %.2f pF",c1
    *10^12)
25 printf("\n\tL2 = %.2f nH\n\tC3 = %.2f pF\n\tL4 = %.2
    f nH\n\tC5 = %.2f pF",12*10^9,c3*10^12,14*10^9,c5
    *10^12)

```

---

### Scilab code Exa 5.12 Example 12

```

1 //Chapter 5, Problem 12
2 clc
3 z0=75 //characteristic
    impedance
4 fp=500*10^6 //passband limit
    frequency
5 f1=260*10^6 //frequency in hertz
6 //from Figure 5.39, it is seen that about five arms
    will be required
7 n=5
8
9 //Butterworth normalised values
10 g1=1/1.382
11 g2=1/1.326
12 g3=1/2.209
13 g4=1/1.326
14 g5=1/1.382
15 w=2*%pi*fp
16
17 //calculation of component values
18 l1=g1*z0/(w)
19 c2=g2/(w*z0)
20 l3=g3*z0/w
21 c4=g4/(w*z0)
22 l5=g5*z0/(w)
23 printf("(a) Number of arms of low pass filter = %.2f
    \n",n)

```

```

24 printf("(b) Component values are\n\tL1 = %f nH", l1
    *10^9)
25 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

```

1 //Chapter 5, Problem 13
2 clc
3 n1=16 //no of turns on
    primary
4 n2=8 //no of turns on
    secondary
5 zs=16 //terminating
    resistance
6
7 //calculation of effective resistance
8 zp=zs*(n1/n2)^2
9 printf("Effective resistance at the primary = %d ohm
    ", zp)

```

---

#### Scilab code Exa 5.14 Example 14

```

1 //Chapter 5, Problem 14
2 clc
3 n1=160 //no of turn
4 n2=40 //no of turns
5 n3=8 //no of turns
6 n4=150 //no of turns
7 n5=50 //no of turns
8 r1=2000 //load
    resistance in ohms
9

```

```

10 // calculation
11 r11=((n1+n2)/n3)^2*r1
12 req=(n2/(n1+n2))^2*r11
13 r12=((n4+n5)/n3)^2*r1
14 req2=(n5/(n4+n5))^2*r12
15 printf("(a) Transistor load impedance at resonance =
    %d Kohm\n\n",req/1000)
16 printf("(b) New transistor load impedance at
    resonance = %.2f Kohm\n\n",req2/1000)

```

---

#### Scilab code Exa 5.15 Example 15

```

1 //Chapter 5, Problem 15
2 clc
3 n1=160 //no of turn
4 n2=40 //no of turns
5 n3=8 //no of turns
6 n4=150 //no of turns
7 n5=50 //no of turns
8 r1=2000 //load
    resistance in ohms
9 rt=100e3 //output
    impedance of transistor
10 Q=100 //Q factor
11 Ct=180*10^-12 //capacitance
    in farad
12 f=465e3 //resonant
    frequency
13
14 r11=((n1+n2)/n3)^2*r1
15 rtr=((n1+n2)/n2)^2*rt
16 rckt=Q/(2*pi*Ct*f)
17 req=r11*rckt/(r11+rckt)
18 r12=((n2/(n1+n2))^2)*req
19

```

```
20 printf(" Transistor load impedance at resonance = %.2
    f Kohm\n\n",r12/1000)
```

---

#### Scilab code Exa 5.16 Example 16

```
1 //Chapter 5, Problem 16
2 clc
3 req=125e3 //effective
    resistance in ohm
4 f=465e3 //resonant frequency
    in hertz
5 L=650e-6 //tuning inductance
    in inductance
6
7 //calculation
8 Q=req/(2*pi*f*L)
9 B=f/Q
10 printf("Q = %.1f \n\n Bandwidth = %d Hz",Q,B)
```

---

#### Scilab code Exa 5.17 Example 17

```
1 //Chapter 5, Problem 17
2 clc
3 z1=22e3 //reactance in ohm
4 c1=10 //capacitance in picofarad
5 c2=100 //capacitance in picofarad
6
7 //calculation
8 z2=z1*(c1/(c1+c2))
9 printf("transformed value of reactance = %d Kohm",z2
    /1000)
```

---



Scilab code Exa 5.18 Example 18

```
1 //Chapter 5, Problem 18
2 clc
3 f=100e6 //frequency in hertz
4 cp=100e-12 //capacitance in
   farad
5 rp=15e3 //resistance in ohm
6
7 //calculation
8 qp=2*%pi*f*cp*rp
9 printf("Quality factor Qp = %.2f",qp)
```

---

Scilab code Exa 5.19 Example 19

```
1 //Chapter 5, Problem 19
2 clc
3 f=800e3 //frequency in hertz
4 Ls=365e-6 //capacitance in
   farad
5 Rs=8 //resistance in ohm
6
7 //calculation
8 Qs=(2*%pi*f*Ls)/Rs
9 printf("Quality factor Qs = %d",Qs)
```

---

Scilab code Exa 5.20 Example 20

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

```

2  clc
3  f=10e6                                //frequency in hertz
4  Ls=15e-6                              //capacitance in farad
5  Rs=2                                  //resistance in ohm
6
7  // calculation
8  Qs=(2*%pi*f*Ls)/Rs
9  Rp=Rs*(1+(Qs^2))
10 Lp=((1+Qs^2)/Qs^2)*Ls
11
12 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

```

1  //Chapter 5, Problem 21, figure 5.55
2  clc
3  Rp=500                                //equals to load
    resistance
4  Rs=50                                  //equals to generator
    resistance
5  f=100e6                                //frequency in hertz
6
7  w=2*%pi*f
8  Qs=sqrt((Rp/Rs)-1)
9  Ls=(Rs*Qs)/w
10 Xs=w*Ls
11 Ca=1/(w*Xs)
12 Lp=((1+Qs^2)/Qs^2)*Ls
13 printf(" Capacitor Ca = %.2 f pF\n\n Inductor Lp = %.2 f
    nH" ,Ca/10^-12,Lp/10^-9)

```

---

### Scilab code Exa 5.22 Example 22

```
1 //Chapter 5, Problem 22, figure 5.58
2 clc
3 f=100e6 //supply
   frequency in hertz
4 Rs=50 //resistance in
   ohms
5 Csh=42e-12 //shunt
   capacitance in ohm
6 Rl=500 //load
   resistance in ohm
7 Rp=Rl
8
9 //calculation
10 w=2*%pi*f
11 Qs=sqrt((Rp/Rs)-1)
12 Ls=(Rs*Qs)/w
13 Xs=w*Ls
14 Ca=1/(w*Xs)
15 Lp=((1+Qs^2)/Qs^2)*Ls
16
17 L=1/(w^2*Csh)
18 Lcom=(Lp*L)/(Lp+L)
19
20 printf("Matching network component value are,\n Ca =
   %.1f pF \n L (combined) = %d nH\n\n",Ca*10^12,
   Lcom*10^9)
21 disp("For the final network, shown in figure 5.61")
```

---

### Scilab code Exa 5.23 Example 23

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

```

4 R1=1000 //resistance in ohm
5 Q=15 //Q factor
6
7 //calculation
8 Rv=R1/(Q^2+1)
9 Xp2=R1/Q
10 Xs2=Q*Rv
11 Q1=sqrt((Rs/Rv)-1)
12 Xp1=Rs/Q1
13 Xs1=Q1*Rv
14
15 printf("Zs = %d ohm\nXp1 = %.3 f ohm \nXs1 = %.3 f ohm
\n",Rs,Xp1,Xs1)
16 printf("Xs2 = %.3 f ohm\n Xp2 = %.3 f ohm\n Zl = %d
ohm\n\n",Xs2,Xp2,R1)
17 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

```

1 //Chapter 5, Problem 24, figure 5.72
2 clc
3 Rs=10 //resistance in ohm
4 R1=50 //resistance in ohm
5 Q=10 //Q factor
6
7 //calculation
8 Rv=Rs*(Q^2+1)
9 Xs1=Q*Rs
10 Xp1=Rv/Q
11 Q2=sqrt((Rv/R1)-1)
12 Xp2=Rv/Q2
13 Xs2=Q2*R1
14
15 printf("Zs = %d ohm\nXp1 = %.3 f ohm \nXs1 = %.3 f ohm

```

```
    \n",Rs,Xp1,Xs1)
16 printf("Xs2 = %.3 f ohm\n Xp2 = %.3 f ohm\n Zl = %d
    ohm\n\n",Xs2,Xp2,R1)
17 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

```
1 //Chapter 6, Problem 1
2 clc
3 hfe=200 //dc current
   gain
4 vcc=10 //supply voltage
5 vbe=0.7 //base to
   emitter voltage
6 ic=1e-3 //collector
   current
7 vc=5 //collector
   voltage
8
9 //calculation of operating point
10 ib=ic/hfe
11 rf=(vc-vbe)/ib
12 rc=(vcc-vc)/(ic+ib)
13 printf(" Operating point Rc = %.3f Kohm",rc/1000)
```

---

### Scilab code Exa 6.2 Example 2

```
1 //Chapter 6, Problem 2
2 clc
3 hfe=250 //dc current
   gain
4 vcc=24 //supply voltage
5 vbe=0.7 //base to
   emitter voltage
6 ic=2e-3 //collector
   current
7 vc=12 //collector
   voltage
8
9 //calculation of operating point
10 ib=ic/hfe
11 rf=(vc-vbe)/ib
12 rc=(vcc-vc)/(ic+ib)
13 printf("Operating point Rc = %.3f Kohm",rc/1000)
```

---

### Scilab code Exa 6.3 Example 3

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

```

7 hfe=150                                //dc current
   gain
8 vbb=2
9 ibb=1e-3
10
11 //calculating the biasing resistors
12 ib=ic/hfe
13 rb=(vbb-vbe)/ib
14 r1=vbb/ibb
15 rf=((vc-vbb)/(ibb+ib))
16 rc=((vcc-vc)/(ic+ib+ibb))
17
18 disp("Biasing resistors is given by")
19 printf("R1 = %d Kohm\nRb = %.2 f Kohm\nRc = %.2 f Kohm
   \nRf = %.2 f Kohm",r1/1000 ,rb/1000 ,rc/1000 ,rf
   /1000)

```

---

#### Scilab code Exa 6.4 Example 4

```

1 //Chapter 6, Problem 4
2 clc
3 vbe=0.7                                //base to
   emitter voltage
4 ic=1e-3                                //collector
   current
5 vc=6                                    //collector
   voltage
6 hf1=100                                //dc current
   gain
7 hf2=250                                //dc current
   gain
8 vcc=12                                  //supply voltage
9 vbb=1.5
10 ibb=0.5e-3
11

```



```

12 //calculating the biasing resistors
13 hfe=sqrt(hf1*hf2)
14 ib=ic/hfe
15 rb=((vbb-vbe)/ib)
16 r1=vbb/ibb
17 rf=((vc-vbb)/(ibb+ib))
18 rc=((vcc-vc)/(ic+ib+ibb))
19
20 disp(" Biasing resistor is given by\n")
21 printf("R1 = %d Kohm\nRb = %.2 f Kohm\nRc = %.2 f Kohm
        \nRf = %.2 f Kohm" ,r1/1000 ,rb/1000 ,rc/1000 ,rf
        /1000)

```

---

#### Scilab code Exa 6.5 Example 5

```

1 //Chapter 6, Problem 5
2 clc
3 vbe=0.7 //base to
        emitter voltage
4 ic=10e-3 //collector
        current
5 vc=10 //collector
        voltage
6 vcc=20 //supply voltage
7 hfe=50 //dc current gain
8
9
10 //calculating the biasing resistors
11 ie=ic //assuming for
        high gain transistor
12 ve=(10/100)*vcc
13 re=ve/ie
14 rc=((vcc-vc)/ic)
15 ib=ic/hfe
16 vbb=ve+vbe

```

```

17  ibb=10*ib
18  r2=vbb/ibb
19  r1=((vcc-vbb)/(ibb+ib))
20
21  disp("Biasing resistor is given by\n")
22  printf("R1 = %d ohm\nR2 = %.2 f ohm\nRc = %.2 f ohm\
        nRe = %.2 f ohm",r1,r2,rc,re)

```

---

### Scilab code Exa 6.6 Example 6

```

1  //Chapter 6, Problem 6
2  clc
3  vbe=0.7 //base to
    emitter voltage
4  ic=1e-3 //collector
    current
5  vc=6 //collector
    voltage
6  vcc=12 //supply voltage
7  hf1=100 //dc current
    gain
8  hf2=250 //dc current
    gain
9  ibb=0.5e-3
10
11 //calculating the biasing resistors
12 ie=ic
13 hfe=sqrt(hf1*hf2)
14 ve=(10/100)*vcc
15 re=ve/ie
16 rc=((vcc-vc)/ic)
17 ib=ic/hfe
18 vbb=ve+vbe
19 r2=vbb/ibb
20 r1=((vcc-vbb)/(ibb+ib))

```

```

21
22 disp(" Biasing resistor is given by\n")
23 printf("R1 = %.2 f Kohm\nR2 = %.2 f Kohm\nRc = %.2 f
      Kohm\nRe = %.2 f Kohm", r1/1000, r2/1000, rc/1000, re
      /1000)

```

---

### Scilab code Exa 6.7 Example 7

```

1 //Chapter 6, Problem 7, figure 6.13
2 clc
3 vcc=24 //supply
   voltage
4 vds=10 //drain to
   source voltage
5 id=5e-3 //drain
   current
6 vgs=2.3 //gate to
   source voltage
7 vs=2.3 //source
   voltage
8 vp=-8 //pinch-off
   voltage
9 idss=10e-3 //
   drain source current when the gate and source
   are shorted
10
11 //calculating the biasing resistors
12 rs=vgs/id
13 vd=vds+vs
14 rd=(vcc-vd)/id
15 vgs=vp*(1-sqrt(id/idss))
16
17 disp(" Since IG = 0, RG = 1 Mohm (approx)")
18 printf("Rs = %.2 f ohm\nRd = %.2 f ohm\n\n", rs, rd)

```

---

### Scilab code Exa 6.8 Example 8

```
1 //Chapter 6, Problem 8
2 clc
3 id=2e-3 //drain current
   in ampere
4 vds=12 //drain to
   source voltage
5 vcc=24 //supply voltage
6 idss=8e-3 // d r a i n source
   current when the gate and source are shorted
7 vp=-6 //pinch-off
   voltage
8
9 //calculating the biasing resistors
10 vgs=vp*(1-sqrt(id/idss))
11 rs=-vgs/id
12 vs=-vgs
13 vd=vds+vs
14 rd=((vcc-vd)/id)
15
16 disp("Since IG = 0, RG = 1 Mohm (approx)")
17 printf("Rs = %d ohm\nRd = %.2f ohm\n\n",rs,rd)
```

---

### Scilab code Exa 6.9 Example 9

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

```

5 vcc=-24 //supply
   voltage
6 idss=8e-3 // d r a i n source
   current when the gate and source are shorted
7 vp=-6 //pinch-off
   voltage
8 vgs=2.3 //gate to source
   voltage
9
10 //calculating the biasing resistors
11 rs=-vgs/id
12 vs=-vgs
13 vd=vds+vs
14 rd=((vcc-vd)/id)
15
16 disp(" Since IG = 0, RG = 1 Mohm (approx)")
17 printf("\nRs = %.2f ohm\nRd = %.2f ohm\n\n",rs,rd)

```

---

#### Scilab code Exa 6.10 Example 10

```

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

```

```

9
10 //calculating the biasing resistors
11 vgs=vp*(1-sqrt(id/idss))
12 rs=-vgs/id
13 vd=-vds+vs
14 rd=((vcc+vd)/id)
15
16 disp(" Since IG = 0, RG = 1 Mohm (approx)")
17 printf("\nRs = %d ohm\nRd = %.2f ohm\n\n",rs,rd)

```

---

#### Scilab code Exa 6.11 Example 11

```

1 //Chapter 6, Problem 11
2 clc
3 id=5e-3 //drain current
   in ampere
4 vds=10 //drain to
   source voltage
5 vcc=18 //supply
   voltage
6 vs=0.1*vcc //source
   voltage
7 vgs=3.2 //gate to
   source voltage
8 r2=220e3 //resistance in
   ohm based upon d.c. input resistance needs
9
10 //calculating the biasing resistors
11 rs=vs/id
12 vg=vgs+vs
13 r1=(r2*(vcc-vg)/vg)
14 vd=vds+vs
15 rd=((vcc-vd)/id)
16
17 printf("\nRs = %d ohm\nRd = %.2f ohm\n\n",rs,rd)

```

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

---

### Scilab code Exa 6.12 Example 12

```
1 //Chapter 6, Problem 12
2 clc
3 id=2e-3 //drain current
    in ampere
4 vds=6 //drain to source
    voltage
5 vcc=12 //supply
    voltage
6 vs=0.1*vcc //source
    voltage
7 vgs=1.8 //gate to
    source voltage
8 r2=220e3 //resistance in
    ohm based upon d.c. input resistance needs
9
10 //calculating the biasing resistors
11 rs=vs/id
12 vg=vgs+vs
13 r1=(r2*(vcc-vg)/vg)
14 vd=vds+vs
15 rd=((vcc-vd)/id)
16
17 printf("\nRs = %d ohm\nRd = %.2 f ohm\n\n",rs,rd)
18 printf("\nR1 = %d Kohm\nR2 = %.2 f Kohm\n\n",r1/1000,
    r2/1000)
```

---

### Scilab code Exa 6.13 Example 13

```

1 //Chapter 6, Problem 13
2 clc
3 funcprot(0)
4 //using Y-parameters given in the case study
5 Yin=(18.33+%i*11.59)*10^-3 //in complex
    form
6 y21=(1.09-%i*17.51)*10^-3 //in complex
    form
7 y22=(0.3+%i*1.57)*10^-3 //in complex
    form
8 Y1=3.33e-3
9
10 //defining a funcion
11 deff(' [b]=imp1(d) ', 'b={{(real(d)^2)+(imag(d)^2)}^0.5'
    );
12 deff(' [c]=imp2(e) ', 'c=atan(imag(e)/real(e))*180/%pi'
    );
13
14 a=y21*Y1
15 b=Yin*(y22+Y1)
16
17 //calling a function
18 [a1]=imp1(a)
19 [a2]=imp2(a)
20 [b1]=imp1(b)
21 [b2]=imp2(b)
22
23 Ai1=a1/b1
24 Ai2=a2-b2
25
26 printf(" Ai (magnitude) = %.2f amp\n\n",Ai1)
27 printf(" Ai (angle) = %.2f degree",Ai2)

```

---

Scilab code Exa 6.14 Example 14



```

1 //Chapter 6, Problem 14
2 clc
3 //transistor parameter
4 yi=(16+%i*11.78)*10^-3 //in complex
   form
5 y0=(1.55+%i*5.97)*10^-3 //in complex
   form
6 gi=16e-3 //input
   conductance
7 go=0.19e-3 //output
   conductance
8 yr_mag=1.55e-3 //magnitude of
   yr parameter
9 yr_ang=258 //angle of yr
   parameter
10 yf_mag=45e-3 //magnitude of
   yf parameter
11 yf_ang=285 //angle of yf
   parameter
12
13 //calculation of stability factor
14 a=yr_mag*yf_mag
15 b=(2*gi*go)+(yr_mag*yf_mag)
16 C=a/b
17
18 printf("Linvill stability factor C = %.2f ",C)

```

---

#### Scilab code Exa 6.15 Example 15

```

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

```

```

6 //      Theta is a matrix containing the phase angles
      (in degrees).
7 function [A] = p2z(R,Theta)
8 A = R*exp(%i*%pi*Theta/180);
9 endfunction
10
11 // [R, Theta] = z2p(A) - Display polar form of
      complex matrix.
12 function [R, Theta] = z2p(A)
13     Theta = atan(imag(A),real(A))*180/%pi;
14     R=sqrt(real(A)^2+imag(A)^2)
15 endfunction
16
17 //transistor parameter
18 yi=(4.8+%i*4.52)*10^-3           //in complex
      form
19 y0=(0.05+%i*2.26)*10^-3         //in complex
      form
20 gi=4.8e-3                       //input
      conductance
21 go=0.05e-3                       //output
      conductance
22 yr=p2z(0.90e-3,265)              //in polar
      form to complex form
23 yf=p2z(61e-3,325)                //in polar
      form to complex form
24
25 [yrm,yra]=z2p(yr)                 //in complex
      form to polar form
26 [yfm,yfa]=z2p(yf)                 //in complex
      form to polar form
27
28 Zs=50+%i*0
29 Zl=1000+%i*0
30 Ys=1/Zs
31 Yl=1/Zl
32 a=2*(gi+Ys)*(go+Yl)
33 b=(yfm*yrm)+real(yr*yf)

```

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

---

#### Scilab code Exa 6.16 Example 16

```
1 //Chapter 6, Problem 16
2 clc
3 //transistor Y parameter
4 yi=(16+%i*11.78)*10^-3 //in complex
   form
5 yf_mag=45e-3
6 yf_ang=285
7 yr_mag=1.55e-3
8 yr_ang=258
9 yo=0.19+%i*5.97 //in complex
   form
10 gi=16e-3 //input
   conductance in siemens
11 go=0.19e-3 //output
   conductance in siemens
12
13 //calculating maximum available gain
14 MAG=yf_mag^2/(4*gi*go)
15 MAG_db=10*log10(MAG)
16 printf("Maximum available gain = %.2f \n\n",MAG)
17 printf("Maximum available gain in dB = %.2f dB",
   MAG_db)
```

---

#### Scilab code Exa 6.17 Example 17

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

```

3 // A = p2z(R,Theta) - Convert from polar to
  rectangular form.
4 //   R is a matrix containing the magnitudes
5 //   Theta is a matrix containing the phase angles
  (in degrees).
6 function [A] = p2z(R,Theta)
7   A = R*exp(%i*%pi*Theta/180);
8 endfunction
9
10 //transistor Y parameter
11 yi=(17.37+%i*11.28)*10^-3           //in
  complex form
12 yr_mag=1.17e-3
13 yf_mag=130.50e-3
14 yr=p2z(1.17e-3,-91)
15 yf=p2z(130.50e-3,-69)
16 yo=(0.95+%i*3.11)*10^-3           //in
  complex form
17 f=300e6                             //
  frequency in hertz
18 Vce=5                                //base to
  emitter voltage
19 Ic=2e-3                               //
  collector current
20 gi=17.37e-3                           //input
  conductance
21 go=0.95e-3                            //output
  conductance
22
23 //to calculate linvill stability factor
24 a=yf_mag*yr_mag
25 b=(2*gi*go)-real(yf*yr)
26 c=a/b
27
28 //to calculate maximum available gain
29 MAG=yf_mag^2/(4*gi*go)
30 MAG_db=10*log10(MAG)
31

```

```

32 //to calculate conjugate input admittance
33 m=sqrt(((2*gi*go)-real(yf*yr))^2-(yf_mag*yr_mag)^2)
34 n=2*go
35 Gs=m/n
36 Bs=-imag(yi)+(imag(yf*yr)/(2*go))
37 Gsi=Gs+%i*Bs
38
39 //to calculate conjugate output admittance
40 Gl=Gs*go/gi
41 Bl=-imag(yo)+(imag(yf*yr)/(2*gi))
42 Gsl=Gl+%i*Bl
43
44 //to calculate Stern stability factor
45 u=2*(gi+Gs)*(go+Gl)
46 v=(yf_mag*yr_mag)+real(yf*yr)
47 K=u/v
48
49 printf("(1) Linvill stability factor C = %.2f\n\n",c
    )
50 printf("(2) Maximum available gain (MAG) = %.2f dB\n
    \n",MAG_db)
51 printf("(3) Conjugate input admittance \n\treal = %
    .2f mS \n\timaginary = %.2f mS\n\n",real(Gsi)
    *1000,imag(Gsi)*1000)
52 printf("(4) Conjugate output admittance \n\treal = %
    .2f mS \n\timaginary = %.2f mS\n\n",real(Gsl)
    *1000,imag(Gsl)*1000)
53 printf("(5) Stern stability factor K = %.2f",K)

```

---

#### Scilab code Exa 6.18 Example 18

```

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

```

```

    rectangular form.
5 //      R is a matrix containing the magnitudes
6 //      Theta is a matrix containing the phase angles
    (in degrees).
7 function [A] = p2z(R,Theta)
8   A = R*exp(%i*pi*Theta/180);
9 endfunction
10
11 // [R1, Theta1] = z2p(A1) - Display polar form of
    complex matrix.
12 function [R1, Theta1] = z2p(A1)
13   Theta1 = atan( imag(A1),real(A1))*180/%pi;
14   R1=sqrt(real(A1)^2+imag(A1)^2)
15 endfunction
16
17
18 //transistor Y parameter
19 yi=(17.37+%i*11.28)*10^-3 //in
    complex form
20 yr_mag=1.17e-3
21 yf_mag=130.50e-3
22 yr=p2z(1.17e-3,-91)
23 yf=p2z(130.50e-3,-69)
24 yo=(0.95+%i*3.11)*10^-3 //in
    complex form
25 f=300e6 //
    frequency in hertz
26 Vce=5 //base to
    emitter voltage
27 Ic=2e-3 //
    collector current
28 gi=17.37e-3 //input
    conductance
29 go=0.95e-3 //output
    conductance
30
31 //to calculate conjugate input admittance
32 m=sqrt(((2*gi*go)-real(yf*yr))^2-(yf_mag*yr_mag)^2)

```

```

33 n=2*go
34 Gs=m/n
35 Bs=-imag(yi)+(imag(yf*yr)/(2*go))
36 Gsi=Gs+%i*Bs
37
38 //to calculate conjugate output admittance
39 Gl=Gs*go/gi
40 Bl=-imag(yo)+(imag(yf*yr)/(2*gi))
41 Gsl=Gl+%i*Bl
42
43
44 num=4*Gs*Gl*(yf_mag)^2
45 den=((yi+Gsi)*(yo+Gsl))-(yf*yr)
46 [denm,dena]=z2p(den)
47 Gt=num/denm^2
48 Gt_db=10*log10(Gt)
49 printf("Transducer gain = %.2 f dB",Gt_db)

```

---

#### Scilab code Exa 6.19 Example 19

```

1 //Chapter 6, Problem 18
2 clc
3 // A = p2z(R,Theta) - Convert from polar to
  //      rectangular form.
4 //      R is a matrix containing the magnitudes
5 //      Theta is a matrix containing the phase angles
  //      (in degrees).
6 function [A] = p2z(R,Theta)
7   A = R*exp(%i*%pi*Theta/180);
8 endfunction
9
10 //transistor parameter
11 yi=(2.25+%i*7.2)*10^-3
12 yr=p2z(0.70e-3,-85.9)
13 yf=p2z(44.72e-3,-26.6)

```

```

14 yo=(0.4+%i*1.9)*10^-3
15 yr_mag=0.70e-3
16 yf_mag=44.72e-3
17 Rs=250
18 Gs=1/Rs
19 K=3                                     //stern
    stability factor
20 gi=2.25e-3                             //input
    conductance
21 go=0.4e-3                              //output
    conductance
22 a=K*((yr_mag*yf_mag)+real(yf*yr))
23 b=2*(gi+Gs)
24 Gl=(a/b)-go
25
26 Bl=-imag(yo)
27 Yl=Gl+%i*Bl
28
29 yin=yi-((yr*yf)/(yo+Yl))
30
31 Bs=-imag(yin)
32 Ys=Gs+%i*Bs
33
34 num=4*Gs*Gl*(yf_mag)^2
35 den=real(((yi+Ys)*(yo+Yl))-(yf*yr))^2
36 Gt=num/den
37 Gt_db=10*log10(Gt)
38
39 printf("(a) Load admittance Yl \n\treal = %.2f mS \n
    \timaginary = %.2f mS\n\n",real(Yl)*1000,imag(Yl)
    *1000)
40 printf("(b) Source admittance Ys \n\treal = %.2f mS
    \n\timaginary = %.2f mS\n\n",real(Ys)*1000,imag(
    Ys)*1000)
41 printf("(c) Transducer gain = %.2f dB",Gt_db)

```

---



# Chapter 7

## Microwave amplifiers

Scilab code Exa 7.1 Example 1

```
1 //Chapter 7, Problem 1
2 clc
3 funcprot(0)
4 // A = p2z(R,Theta) - Convert from polar to
   rectangular form.
5 //   R is a matrix containing the magnitudes
6 //   Theta is a matrix containing the phase angles
   (in degrees).
7 function [A] = p2z(R,Theta)
8   A = R*exp(%i*%pi*Theta/180);
9 endfunction
10
11 // [R, Theta] = z2p(A) - Display polar form of
   complex matrix.
12 function [R, Theta] = z2p(A)
13   Theta = atan(imag(A),real(A))*180/%pi;
14   R=sqrt(real(A)^2+imag(A)^2)
15 endfunction
16
17 //transistor s-parameter
18 s11=p2z(0.3,160)
```

```

19 s12=p2z(0.03,62)
20 s21=p2z(6.1,65)
21 s22=p2z(0.40,-38)
22 R=50 //resistance in ohms
23 f=150e6 //frequency in hertz
24 vce=12 //base to emitter
    voltage
25 ic=8e-3 //collector current
26
27 Ds=(s11*s22)-(s12*s21)
28 [Dmag,Dang]=z2p(Ds)
29 [s11m,s11a]=z2p(s11)
30 [s22m,s22a]=z2p(s22)
31 [s21m,s21a]=z2p(s21)
32 [s12m,s12a]=z2p(s12)
33 K=(1+Dmag^2-s11m^2-s22m^2)/(2*s21m*s12m)
34 B1=1+s11m^2-s22m^2-Dmag^2
35 MAG=10*log10(s21m/s12m)+10*log10(K-sqrt(K^2-1))
36 C2=s22-(Ds*conj(s11))
37 [C2m,C2a]=z2p(C2)
38 B2=1+s22m^2-s11m^2-Dmag^2
39 reflm=(B2-sqrt(B2^2-4*C2m^2))/(2*C2m)
40 refla=-C2a
41 refl=p2z(reflm,refla)
42 refs=conj(s11+((s12*s21*refl)/(1-(s22*refl))))
43 [refsm,refsa]=z2p(refs)
44
45 //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.
46 y=1.33
47 r=0.34
48 C1=y/(2*pi*f*R)
49 L1=r*R/(2*pi*f)
50

```

```

51 //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.
52 y1=0.8
53 r1=1.1
54 C2=1/(2*%pi*f*R*r1)
55 L2=R/(2*%pi*f*y1)
56
57 printf("For input matching network,\n\n")
58 printf("C1 = %.2 f pF\n",C1*10^12)
59 printf("L1 = %.2 f nH\n\n",L1*10^9)
60 printf("For output matching network,\n\n")
61 printf("C2 = %.2 f pF\n",C2*10^12)
62 printf("L2 = %.2 f nH\n\n",L2*10^9)
63 printf("The completed design (minus biasing network)
    is shown in Figure 7.3")

```

---

#### Scilab code Exa 7.2 Example 2

```

1 //Chapter 7, Problem 2
2 clc
3 funcprot(0)
4 // A = p2z(R,Theta) - Convert from polar to
    rectangular form.
5 // R is a matrix containing the magnitudes
6 // Theta is a matrix containing the phase angles
    (in degrees).
7 function [A] = p2z(R,Theta)
8 A = R*exp(%i*%pi*Theta/180);
9 endfunction
10
11 // [R, Theta] = z2p(A) - Display polar form of
    complex matrix.
12 function [R, Theta] = z2p(A)

```

```

13     Theta = atan(imag(A),real(A))*180/%pi;
14     R=sqrt(real(A)^2+imag(A)^2)
15 endfunction
16
17 //transistor s-parameter
18 s11=p2z(0.3,160)
19 s12=p2z(0.03,62)
20 s21=p2z(6.1,65)
21 s22=p2z(0.40,-38)
22 R=50 //resistance in ohms
23 f=150e6 //frequency in hertz
24 vce=12 //base to emitter
    voltage
25 ic=8e-3 //collector current
26
27 Ds=(s11*s22)-(s12*s21)
28 [Dmag,Dang]=z2p(Ds)
29 [s11m,s11a]=z2p(s11)
30 [s22m,s22a]=z2p(s22)
31 [s21m,s21a]=z2p(s21)
32 [s12m,s12a]=z2p(s12)
33 K=(1+Dmag^2-s11m^2-s22m^2)/(2*s21m*s12m)
34 B1=1+s11m^2-s22m^2-Dmag^2
35 MAG=10*log10(s21m/s12m)+10*log10(K-sqrt(K^2-1))
36 C2=s22-(Ds*conj(s11))
37 [C2m,C2a]=z2p(C2)
38 B2=1+s22m^2-s11m^2-Dmag^2
39 reflm=(B2-sqrt(B2^2-4*C2m^2))/(2*C2m)
40 refla=-C2a
41 refl=p2z(reflm,refla)
42 refs=conj(s11+((s12*s21*refl)/(1-(s22*refl))))
43 [refsm,refsa]=z2p(refs)
44
45 a=s21m^2*(1-refsm^2)*(1-reflm^2)
46 b=((1-s11*refs)*(1-s22*refl))-(s12*s21*refl*refs)
47 [bm,ba]=z2p(b)
48 Gt=a/bm^2
49 Gt_db=10*log10(Gt)

```

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

---

### Scilab code Exa 7.3 Example 3

```
1 //Chapter 7, Problem 3
2 clc
3 funcprot(0)
4 // A = p2z(R,Theta) - Convert from polar to
   rectangular form.
5 // R is a matrix containing the magnitudes
6 // Theta is a matrix containing the phase angles
   (in degrees).
7 function [A] = p2z(R,Theta)
8 A = R*exp(%i*%pi*Theta/180);
9 endfunction
10
11 // [R1, Theta1] = z2p(A1) - Display polar form of
   complex matrix.
12 function [R1, Theta1] = z2p(A1)
13 Theta1 = atan(imag(A1),real(A1))*180/%pi;
14 R1=sqrt(real(A1)^2+imag(A1)^2)
15 endfunction
16
17 //transistor s-parameter
18 s11=p2z(0.3,140)
19 s12=p2z(0.03,65)
20 s21=p2z(2.1,62)
21 s22=p2z(0.40,-38)
22
23 f=5e9 //
   frequency in hertz
24 vce=15 //base
   to emitter voltage
25 ic=10e-3 //
   collector current in ampere
```

```

26
27 Ds=(s11*s22)-(s12*s21)
28 [Dmag,Dang]=z2p(Ds)
29 [s11m,s11a]=z2p(s11)
30 [s22m,s22a]=z2p(s22)
31 [s21m,s21a]=z2p(s21)
32 [s12m,s12a]=z2p(s12)
33 K=(1+Dmag^2-s11m^2-s22m^2)/(2*s21m*s12m)
34 B1=1+s11m^2-s22m^2-Dmag^2
35 MAG=10*log10(s21m/s12m)+10*log10(K-sqrt(K^2-1))
36
37 printf("Maximum available gain (MAG) = %.1f dB ",MAG
)

```

---

#### Scilab code Exa 7.4 Example 4

```

1 //Chapter 7, Problem 4
2 clc
3 funcprot(0)
4 // A = p2z(R,Theta) - Convert from polar to
   // rectangular form.
5 // R is a matrix containing the magnitudes
6 // Theta is a matrix containing the phase angles
   // (in degrees).
7 function [A] = p2z(R,Theta)
8 A = R*exp(%i*%pi*Theta/180);
9 endfunction
10
11 // [R1, Theta1] = z2p(A1) - Display polar form of
   // complex matrix.
12 function [R1, Theta1] = z2p(A1)
13 Theta1 = atan(imag(A1),real(A1))*180/%pi;
14 R1=sqrt(real(A1)^2+imag(A1)^2)
15 endfunction
16

```

```

17 //transistor s-parameter
18 s11=p2z(0.3,140)
19 s12=p2z(0.03,65)
20 s21=p2z(2.1,62)
21 s22=p2z(0.40,-38)
22 refs=p2z(0.463,-140)
23 refl=p2z(0.486,38)
24 [s11m,s11a]=z2p(s11)
25 [s22m,s22a]=z2p(s22)
26 [s21m,s21a]=z2p(s21)
27 [s12m,s12a]=z2p(s12)
28 [refsm,refsa]=z2p(refs)
29 [reflm,refla]=z2p(refl)
30
31 //calculation
32 a=(s21m^2)*(1-refsm^2)*(1-reflm^2)
33 b=((1-(s11*refs))*(1-(s22*refl))-(s12*s21*refl*refs)
    )^2
34 Gt=a/real(b)
35 Gt1=10*log10(Gt)
36 printf("Amplifier transducer gain = %.2f dB ",Gt1)

```

---

#### Scilab code Exa 7.5 Example 5

```

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

```

```

10
11 // [R1, Theta1] = z2p(A1) - Display polar form of
    complex matrix.
12 function [R1, Theta1] = z2p(A1)
13     Theta1 = atan(imag(A1),real(A1))*180/%pi;
14     R1=sqrt(real(A1)^2+imag(A1)^2)
15 endfunction
16
17 //transistor s-parameter
18 s11=p2z(0.28,-58)
19 s12=p2z(0.08,92)
20 s21=p2z(2.1,65)
21 s22=p2z(0.8,-30)
22 f=1e9 //frequency in
    hertz
23 vce=15 //collector to
    emitter voltage
24 ic=5e-3 //collector
    current in ampere
25 Zs=35-%i*60 //source
    impedance in ohm
26 Zl=50-%i*50 //load impedance
    in ohm
27 K=1.168 //Rollett
    stability factor
28 g=7.94 //desired gain
29 R=50 //resistance in
    ohm
30
31 [s11m,s11a]=z2p(s11)
32 [s22m,s22a]=z2p(s22)
33 [s21m,s21a]=z2p(s21)
34 [s12m,s12a]=z2p(s12)
35
36 Ds=s11*s22-s12*s21
37 [Dsm,Dsa]=z2p(Ds)
38 D2=s22m^2-Dsm^2
39 C2=s22-(Ds*conj(s11))

```



```

40 G=g/s21m^2
41 ro=(G*conj(C2))/(1+(D2*G))
42 po=sqrt(1-(2*K*s12m*s21m*G)+(s12m*s21m)^2*G^2)/(1+(
    D2*G))
43
44 //The Smith chart construction is shown in Figure
    7.5. The transistors 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
45 r=2
46 y=0.4
47 C1=1/(2*pi*f*r*R)
48 L1=R/(2*pi*f*y)
49
50 //For a conjugate match at the input to the
    transistor, the desired source reflection
    coefficient must be calculated as follows
51 refl=p2z(0.82,13)
    //point
    C in figure 7.5
52 refs=conj(s11+((s12*s21*refl)/(1-(s22*refl))))
53 [refsm,refsa]=z2p(refs)
54
55 //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
56
57 y1=0.63
58 r1=1.08
59 y2=2.15
60
61 C2=y1/(2*pi*f*R)

```

```

62 L2=r1*R/(2*%pi*f)
63 C3=y2/(2*%pi*f*R)
64
65 printf("For output matching network,\n\n")
66 printf("C1 = %.2 f pF\n",C1*10^12)
67 printf("L1 = %.1 f nH\n\n",L1*10^9)
68 printf("For input matching network,\n\n")
69 printf("C2 = %.1 f pF\n",C2*10^12)
70 printf("L2 = %.1 f nH\n\n",L2*10^9)
71 printf("C3 = %.1 f pF\n",C3*10^12)
72 printf("The completed design (minus biasing network)
        is shown in Figure 7.7")

```

---

#### Scilab code Exa 7.6 Example 6

```

1 //Chapter 7, Problem 6
2 clc
3 funcprot(0)
4 // A = p2z(R,Theta) - Convert from polar to
    rectangular form.
5 // R is a matrix containing the magnitudes
6 // Theta is a matrix containing the phase angles
    (in degrees).
7 function [A] = p2z(R,Theta)
8 A = R*exp(%i*%pi*Theta/180);
9 endfunction
10
11 // [R1, Theta1] = z2p(A1) - Display polar form of
    complex matrix.
12 function [R1, Theta1] = z2p(A1)
13 Theta1 = atan(imag(A1),real(A1))*180/%pi;
14 R1=sqrt(real(A1)^2+imag(A1)^2)
15 endfunction
16
17 //transistor s-parameter

```

```

18 s11=p2z(0.28,-58)
19 s12=p2z(0.08,92)
20 s21=p2z(2.1,65)
21 s22=p2z(0.8,-30)
22 Ds=p2z(0.333,-60.66)
23 C2=p2z(0.719,-33.42)
24 D2=0.529 //angle in
    degree //Rollett
25 K=1.168 //desired
    stability factor //gain
26 A=6.31
27 [s11m,s11a]=z2p(s11)
28 [s22m,s22a]=z2p(s22)
29 [s21m,s21a]=z2p(s21)
30 [s12m,s12a]=z2p(s12)
31
32 //calculating the radius of constant gain circle of
    9 dB
33 G=A/s21m^2
34 ro=conj(G*C2)/(1+(D2*G))
35 [rom,roa]=z2p(ro)
36 po=sqrt(1-(2*K*G*s12m*s21m)+((s12m*s21m)^2*G^2))
    /(1+(D2*G))
37
38 printf("Constant gain circle of 8 dB = %.3f ",po)

```

---

#### Scilab code Exa 7.7 Example 7

```

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

```

```

6 //      Theta is a matrix containing the phase angles
      (in degrees).
7 function [A] = p2z(R,Theta)
8   A = R*exp(%i*pi*Theta/180);
9 endfunction
10
11 // [R1, Theta1] = z2p(A1) - Display polar form of
      complex matrix.
12 function [R1, Theta1] = z2p(A1)
13     Theta1 = atan( imag(A1), real(A1))*180/%pi;
14     R1=sqrt(real(A1)^2+imag(A1)^2)
15 endfunction
16
17 //transistor S-parameter
18 s11=p2z(0.4,280)
19 s12=p2z(0.048,65)
20 s21=p2z(5.4,103)
21 s22=p2z(0.78,345)
22 f=200e6 //frequency
      in hertz
23 vce=6 //collector
      to emitter voltage
24 ic=5e-3 //collector
      current in ampere
25
26 [s11m, s11a]=z2p(s11)
27 [s22m, s22a]=z2p(s22)
28 [s21m, s21a]=z2p(s21)
29 [s12m, s12a]=z2p(s12)
30
31 Ds=(s11*s22)-(s12*s21)
32 [Dmag, Dang]=z2p(Ds)
33 K=(1+Dmag^2-s11m^2-s22m^2)/(2*s21m*s12m)
34 C1=s11-(Ds*conj(s22))
35 [C1m, C1a]=z2p(C1)
36 C2=s22-(Ds*conj(s11))
37 [C2m, C2a]=z2p(C2)
38 rs1=conj(C1)/(s11m^2-Dmag^2)

```

```

39 [rs1m,rs1a]=z2p(rs1)
40 ps1=s12*s21/(s11m^2-Dmag^2)
41 [ps1m,ps1a]=z2p(ps1)
42
43 rs2=conj(C2)/(s22m^2-Dmag^2)
44 [rs2m,rs2a]=z2p(rs2)
45 ps2=s12*s21/(s22m^2-Dmag^2)
46 [ps2m,ps2a]=z2p(ps2)
47
48 printf("Centre of input stability circle (magnitude)
      = %.3f \n\t\t\t\t\t(angle) = %.2f degree\n",rs1m,
      rs1a)
49 printf("Radius of input stability circle = %.2f \n\
      n",ps1m)
50 printf("Centre of output stability circle (magnitude
      ) = %.3f \n\t\t\t\t\t(angle) = %.2f degree\n",rs2m,
      rs2a)
51 printf("Radius of output stability circle = %.2f \n\
      \n",ps2m)
52 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")
53 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

```

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

```

```

    (in degrees).
7  function [A] = p2z(R,Theta)
8  A = R*exp(%i*%pi*Theta/180);
9  endfunction
10
11 // [R1, Theta1] = z2p(A1) – Display polar form of
    complex matrix.
12 function [R1, Theta1] = z2p(A1)
13     Theta1 = atan(imag(A1),real(A1))*180/%pi;
14     R1=sqrt(real(A1)^2+imag(A1)^2)
15 endfunction
16
17 //transistor S-parameter
18 s11=p2z(0.4,280)
19 s12=p2z(0.048,65)
20 s21=p2z(5.4,103)
21 s22=p2z(0.78,345)
22 ro=p2z(0.287,24) //centre of
    gain circle
23 po=0.724 //radius of
    12dB constant gain circle
24 f=200e6 //frequency
    in hertz
25 vce=6 //base to
    emitter voltage
26 ic=5e-3 //
    collector current
27 A=15.85 //gain
    desired
28 K=0.802 //
    Rollett's stability factor
29 C2=p2z(0.651,24.1)
30
31 [s11m,s11a]=z2p(s11)
32 [s22m,s22a]=z2p(s22)
33 [s21m,s21a]=z2p(s21)
34 [s12m,s12a]=z2p(s12)
35 Ds=(s11*s22)-(s12*s21)

```

```

36 [Dmag , Dang]=z2p(Ds)
37 G=A/s21m^2
38 D2=s22m^2-Dmag^2
39 ro=C2*G/(1+(D2*G))
40 [rom , roa]=z2p(ro)
41 po1=sqrt(1-(2*K*G*abs(s12m*s21m))+((abs(s12m*s21m))^2*G^2))/(1+(D2*G))
42
43 printf("ro (magnitude) = %.3f\n\t (angle) = %.3f
         degree\n\n",rom , roa)
44 printf("po = %.3f \n\n",po1)
45 disp("There the values are correct")

```

---

#### Scilab code Exa 7.9 Example 9

```

1 //Chapter 7, Problem 9
2 clc
3 funcprot(0)
4 // A = p2z(R,Theta) - Convert from polar to
   rectangular form.
5 //   R is a matrix containing the magnitudes
6 //   Theta is a matrix containing the phase angles
   (in degrees).
7 function [A] = p2z(R,Theta)
8   A = R*exp(%i*%pi*Theta/180);
9 endfunction
10
11 // [R1, Theta1] = z2p(A1) - Display polar form of
   complex matrix.
12 function [R1, Theta1] = z2p(A1)
13   Theta1 = atan(imag(A1),real(A1))*180/%pi;
14   R1=sqrt(real(A1)^2+imag(A1)^2)
15 endfunction
16
17 //transistor S parameter

```

```

18 s11=p2z(0.4,280)
19 s12=p2z(0.048,65)
20 s21=p2z(5.4,103)
21 s22=p2z(0.78,345)
22 rs=p2z(0.678,79.4) //source
    reflection coefficient
23 r1=p2z(0.89,70) //load
    reflection coefficient
24
25 Rs=conj(s11+((s12*s21*r1)/(1-(s22*r1))))
26 [Rsm,Rsa]=z2p(Rs)
27 printf("Source reflection coefficient , magnitude = %
    .3f \n\t\t\t\t\t angle = %.1f degree",Rsm,Rsa)

```

---

#### Scilab code Exa 7.10 Example 10

```

1 //Chapter 7, Problem 10
2 clc
3 funcprot(0)
4 // A = p2z(R,Theta) - Convert from polar to
    rectangular form.
5 // R is a matrix containing the magnitudes
6 // Theta is a matrix containing the phase angles
    (in degrees).
7 function [A] = p2z(R,Theta)
8 A = R*exp(%i*%pi*Theta/180);
9 endfunction
10
11 // [R1, Theta1] = z2p(A1) - Display polar form of
    complex matrix.
12 function [R1, Theta1] = z2p(A1)
13 Theta1 = atan(imag(A1),real(A1))*180/%pi;
14 R1=sqrt(real(A1)^2+imag(A1)^2)
15 endfunction
16

```



```

17 //transistor S-parameter
18 s11=p2z(0.35,165)
19 s12=p2z(0.035,58)
20 s21=p2z(5.9,66)
21 s22=p2z(0.46,-31)
22 refs=p2z(0.68,142) //source
    reflection coefficient
23 f=300e6 //
    frequency in hertz
24 vce=12 //base
    to emitter voltage
25 ic=4e-3 //
    collector current in ampere
26 [s11m,s11a]=z2p(s11)
27 [s22m,s22a]=z2p(s22)
28 [s21m,s21a]=z2p(s21)
29 [s12m,s12a]=z2p(s12)
30
31 Ds=(s11*s22)-(s12*s21)
32 [Dmag,Dang]=z2p(Ds)
33 K=(1+Dmag^2-s11m^2-s22m^2)/(2*s21m*s12m)
34
35 y=1.65
36 r=0.85
37 y1=0.62
38 r1=1.2
39 R=50
40
41 //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
42 C1=y/(2*pi*f*R)
43 L1=(r*R)/(2*pi*f)
44
45 refl=conj(s11+((s12*s21*refs)/(1-(s22*refs))))
46
47 //The design values of the input matching network

```

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

```

48
49 C2=1/(2*pi*f*r1*R)
50 L2=R/(2*pi*f*y1)
51
52 printf("For input matching network,\n\n")
53 printf("C1 = %.2f pF\n",C1*10^12)
54 printf("L1 = %.2f nH\n\n",L1*10^9)
55 printf("For output matching network,\n\n")
56 printf("C2 = %.2f pF\n",C2*10^12)
57 printf("L2 = %.2f nH\n\n",L2*10^9)
58 printf("The completed design is shown in Figure 7.12
    ")

```

---

#### Scilab code Exa 7.12 Example 12

```

1 //Chapter 7, Problem 12
2 clc
3
4 Vg=200e-3 //generator voltage
5 Zg=2000 //resistance in ohm
6 //transistor Y-parameter
7 Y11=1/1200
8 Y12=0
9 Y21=70/1200
10 Y22=1/40000
11 Yl=1/1000
12 Y1=[Y11 Y12;Y21 Y22]
13 Yf=[1/10000 -1/10000;-1/10000 1/10000]
14
15 //calculation
16 Yc=Y1+Yf
17 dely=Yc(1,1)*Yc(2,2)-Yc(1,2)*Yc(2,1)

```

```
18 Zin=(Yc(2,2)+Yl)/(dely+(Yc(1,1)*Yl))
19 vin=(Zin*Vg)/(Zin+Zg)
20 Av=-Yc(2,1)/(Yc(2,2)+Yl)
21 vout=vin*Av
22
23 printf("(a) Input impedance (Zin) = %.1f ohm\n\n",
        Zin)
24 printf("(b) Gain (Av) of the circuit = %.2f\n\n",Av)
25 printf("(c) Output voltage = %d mV\n\n",vout*10^3)
```

---

# Chapter 8

## Oscillators and frequency synthesizers

Scilab code Exa 8.1 Example 1

```
1 //Chapter 8, Problem 1
2 clc
3 L=630e-9 //inductance in
   henry
4 C=400e-12 //capacitance in
   farad
5
6 //calculating the resonating frequency
7 fos=1/(2*pi*sqrt(L*C))
8
9 printf("Resonant frequency = %.3f MHz",fos/10^6)
```

---

Scilab code Exa 8.2 Example 2

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

```

3 r1=100e3 //resistance in
   ohm
4 r2=10e3 //resistance in
   ohm
5 c1=10e-9 //capacitance in
   farad
6 c2=100e-9 //capacitance
   in farad
7
8 //calculation
9 w=sqrt(1/(c1*c2*r1*r2))
10 f=w/(2*%pi)
11 g=1+(r1/r2)+(c2/c1)
12
13 printf("(a) Frequency of oscillation = %.2f Hz\n\n",
   f)
14 printf("(b) Minimum gain of the amplifier = %d ",g)

```

---

### Scilab code Exa 8.3 Example 3

```

1 //Chapter 8, Problem 3
2 clc
3 c1=10e-12 //capacitance
   in farad
4 c2=100e-12 //capacitance
   in farad
5 f=100e6 //frequency
   in hertz
6
7 //calculation
8 w=2*%pi*f
9 L=(1/w^2)*((1/c1)+(1/c2))
10 g=1+(c2/c1)
11
12 printf("(a) Value of inductor = %.2f nH\n\n",L*10^9)

```

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

---

#### Scilab code Exa 8.4 Example 4

```
1 //Chapter 8, Problem 4
2 clc
3 ct=15e-12 //capacitance
   in farad
4 c1=47e-12 //capacitance
   in farad
5 c2=100e-12 //capacitance
   in farad
6 L=300e-9 //inductance in
   henry
7
8 //calculation
9 w1=(1/(L*ct))*(1+(ct/c1)+(ct/c2))
10 w=sqrt(w1)
11 fos=w/(2*pi)
12
13 printf("Approximate frequency = %.2f MHz",fos/10^6)
```

---

#### Scilab code Exa 8.5 Example 5

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

```

6 Ko=-40e3 //transfer gain
  in hertz/volt
7 fi=100e3 //input
  frequency in hertz
8 fo=120e3 //oitput
  frequency in hertz from VCO
9
10 //calculation
11 Ka=(Rf/R1)+1
12 Kl=Kphi*Ka*Ko*2*%pi
13 Kl_dB=real(20*log10(Kl))
14 fd=fi-fo
15 Vo=fd/Ko
16 Vd=Vo/Ka
17 theta=Vd/Kphi
18 fd1=-Kl/(2*%pi)
19 Vd1=Kphi*%pi/2
20
21 printf("(a) Voltage gain (ka) for the op-amp = %d\n\n",Ka)
22 printf("(b) Loop gain (kL) = %.1f s^-1\n\t\t= %.1f
  dB\n\n",Kl,Kl_dB)
23 printf("(c) With S1 open as shown, there is no phase
  lock and the beat frequency = %d kHz\n\n",fd
  /1000)
24 printf("(d)(i) fo = %d kHz\n",fi/1000)
25 printf(" (ii) Static phase error = %.3f rad\n",
  theta)
26 printf(" (iii) Vo = %.1f V\n\n",Vo)
27 printf("(e) Hold-in range Df = %.2f kHz\n\n",fd1
  /1000)
28 printf("(f) Maximum value of vd = %.3f V d.c",Vd1)

```

---

Scilab code Exa 8.6 Example 6

```

1 //Chapter 8, Problem 6
2 clc
3 f1=70e6           //section 1, frequency in
   hertz
4 f2=5e6           //section 2, frequency in
   hertz
5 f3=400e3         //section 3, frequency in
   hertz
6 f4=80e3         //section 4, frequency in
   hertz
7
8 //calculation
9 F3h=f3+f4
10 F3l=f3-f4
11
12 F2h=f2+F3h
13 F2l=f2-F3h
14
15 F1h=f1+F2h
16 F1l=f1-F2h
17
18 printf("Mixer 3 : %d Khz and %d Khz\n",F3h/1000,F3l
   /1000)
19 printf("After filter 3 : %d Khz\n\n",F3h/1000)
20 printf("Mixer 2 : %.2f Mhz and %.2f Mhz\n",F2h/10^6,
   F2l/10^6)
21 printf("After filter 2 : %.2f Mhz\n\n",F2h/10^6)
22 printf("Mixer 1 : %.2f Mhz and %.2f Mhz\n",F1h/10^6,
   F1l/10^6)
23 printf("After filter 1 : %.2f Mhz\n\n",F1h/10^6)

```

---

#### Scilab code Exa 8.7 Example 7

```

1 //Chapter 8, Problem 7
2 clc

```



```

3 f1=511e6           //lowest frequency at the
   divider
4 res=1e6           //resolution
5 fh=887e6         //highest frequency at the
   divider
6
7 //calculation of division factor
8 N=f1/res
9 N2=fh/res
10
11 printf("Lowest value of division factor , N = %d \n\n",N)
12 printf("Highest value of division factor , N = %d ",
   N2)

```

---

#### Scilab code Exa 8.8 Example 8

```

1 //Chapter 8, Problem 8
2 clc
3 f1=18.7e6         //lowest frequency at
   the divider
4 fo=50e3          //divider output
5 f12=38.7e6       //highest frequency at
   the divider
6
7 //calculation of division factor
8 N=f1/fo
9 N2=f12/fo
10
11 printf("Lowest value of division factor , N = %d \n\n",N)
12 printf("Highest value of division factor , N = %d ",
   N2)

```

---