Scilab Textbook Companion for Principles Of Communication Engineering by A. K. Gautam[1]

Created by
Prasoon Mishra
B.tech
Electrical Engineering
college of engineering roorkee
College Teacher
Vatsalya Sharma
Cross-Checked by
Lavitha Pereira

July 14, 2017

[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: Principles Of Communication Engineering

Author: A. K. Gautam

Publisher: S. K. Kataria & Sons, New Delhi

Edition: 3

Year: 2005

ISBN: 81-85749-11-6

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

INTRODUCTION

Scilab code Exa 1.1 Program to find noise voltage

```

//Exa:1.1
clc;
clear;
close;
//Given:
b_w=6000000;//bandwidth in hertz
T=290;//temperature in kelvin
k=1.36*10^-23;//constant
R=500//ohms
V_n=sqrt(4*k*T*b_w*R);
printf("\n\n\t noise voltage = %f v ",V_n);
```

Scilab code Exa 1.2 Program to find noise power

```

//Exa:1.2
clc;
clear;
close;
//Given:
bw=1000;//in ohms
t=298;//VSWR (unitless)
P=1.38*10^-23 * 298*1000*10^12;
printf("\n\n\t noise power = %f pW",P);
```

Scilab code Exa 1.3 Program to equivalent noise

```
//Exa:1.3
clc;
clear;
close;
//Given:
```

```
b_w=3000;//bandwidth in hertz
T=300;//temperature in kelvin
k=1.36*10^-23;//constant
Vn=300*10^-6//ohms
R=Vn*Vn/(4*k*T*b_w);
printf("\n\n\t equivalent noise resistance = %f ohm
    ",R);
```

Scilab code Exa 1.4 `Program to astimate maximum noise voltage amplitude`

```

//Exa:1.4
clc;
clear;
close;
//Given:
b_w=200000;//bandwidth in hertz
T=302;//temperature in kelvin
k=1.36*10^-23;//constant
R=1000//ohms
g=1000//Voltage gain
V_n=sqrt(4*k*T*b_w*R) * 10^6;
printf("\n\n\t noise voltage = %f uV ",V_n);
```

Scilab code Exa 1.5 `Program to find noise power`

```
//Exa:1.5
clc;
clear;
close;
//Given:
bw=1000;//in ohms
t=298;//VSWR (unitless)
k=1.38*10^-23;//constant
P=k*298*1000*10^12;
printf("\n\n\t noise power = %f pW",P);
```

Scilab code Exa 1.6 Program to equivalent noise

```
//Exa:1.6
clc;
clear;
close;
//Given:
b_w=3000;//bandwidth in hertz
T=300;//temperature in kelvin
k=1.36*10^-23;//constant
Vn=300*10^-6//ohms
R=Vn*Vn/(4*k*T*b_w);
printf("\n\n\t equivalent noise resistance = %f ohm
   ",R);
```

Chapter 2

AMPLITUDE MODULATION

Scilab code Exa 2.1 Program to calculate total power in the modulated signal

```

//Exa:2.1
clc;
clear;
close;
//Given:
Pc=500;//poer of carrier
m=0.50;//depth
Pt=Pc*(1+(m^2)/2)
printf("\n\n\t total power of modulated signal = %f
   W ",Pt);
```

Scilab code Exa 2.2 Program to calculate carrier frequency

```
//Exa:2.2
clc;
clear;
close;
//Given:
L=50*10^-6;//henry
C=10^-9;//in farads
f=1/(2*%pi*sqrt(L*C));
printf("\n\n\t total power of modulated signal = %f Hz ",f);
```

Scilab code Exa 2.3 Program to calculate power of carrier signal

```
//Exa:2.1
clc;
clear;
close;
//Given:
Pt=20000;//total power
m=0.50;//depth
Pc=Pt/(1+(m^2)/2)*10^-3;
printf("\n\n\t power of carrier signal = %f kW ",Pc);
```

Scilab code Exa 2.4 Program to determine antenna current

```
//Exa:2.1
clc;
clear;
close;
//Given:
It=10.25;//total power
Ic=9;
m=sqrt(2*(It/Ic)-1);
printf("\n\n\t depth signal = %f ",m);
printf("\n when percent of odulation changed to 6");
It=Ic*(1+(m^2)/2);
printf("\n\n\t the antenna current = %f A",It);
```

Scilab code Exa 2.5 Program to calculate modulation index due to second wave

```
//Exa:2.5
clc;
clear;
close;
//Given:
```

```
It=10;//current in amperes
m=0.50;//depth
IT=11.5;//current increased
Ic=It/sqrt(1+m*m/2);
printf("\n\n\t carrier signal current = %f A ",Ic);
mt=sqrt(2*[IT^2/Ic^2-1]);
printf("\n\n\t total modulation index = %f  ",mt);
ma=sqrt(mt*mt-m*m);
printf("\n\n\t modulation index due to second wave=
    %f  ",ma);
```

Scilab code Exa 2.6 Program to calculate total sideband power radiated

```

//Exa:2.6
clc;
clear;
close;
//Given:
Pc=50*10^3;//interms of watts
m=0.85;//depth
Pt=Pc*(1+(m^2)/2);
printf("\n\n\t total power of modulated signal = %f
    kW ",Pt/1000);
```

Scilab code Exa 2.7 Program to calculate total sideband power radiated

```
//Exa:2.7
clc;
clear;
close;
//Given:
m1=0.55;//modulation percent 1
m2=0.65;//modulation percent 2
Pc=360;// in watts
mt=sqrt(m1*m1+m2*m2);
printf("\n\n\t total modulation  = %f  ",mt);
Pb=(mt*mt)*Pc/2;
printf("\n\n\t total sideband power radiated  = %f
   W",Pb);
```

Scilab code Exa 2.8 Program to calculate modulation index radiated power

```
//Exa:2.8
clc;
clear;
close;
//Given:
Pc=6;//in terms of kw
Pt=7.5;//in terms of kw
m2=0.5;//depth
m1=sqrt(2*(Pt/Pc-1));
printf("\n\n\t modualtion index = %f",m1);
mt=sqrt(m1*m1 + m2*m2);
```

```
printf("\n\n\t total modulation index = %f",mt);
PT=Pc*(1+(mt^2)/2);
printf("\n\n\t total power of modulated signal = %f
   kW ",PT);
```

Scilab code Exa 2.9 Program to calculate total power in the modulated signal

```

//Exa:2.9
clc;
clear;
close;
//Given:
Pc=500;//in Watts
m=0.75;//depth
Pt=Pc*(1+(m^2)/2);
printf("\n\n\t total power of modulated signal = %f
   W ",Pt);
```

Scilab code Exa 2.10 Program to calculate power saving

```

//Exa:2.10
clc;
clear;
```

```
close;
//Given:
//Pb=(1+mt*mt/2)*Pc;
printf("\n\n\t (a)when m=1 Pt=1.5Pc");
printf("\n and carrier and one sideband supressed Pt=1.25Pc");
r=1.25/1.5;
printf("\n\n\t Percent power saving =%fpercent",r*100);
printf("\n\n\t (b)when m=0.5 Pt=1.125Pc");
printf("\n and carrier and one sideband supressed Pt=1.0625Pc");
r=1.0625/1.125;
printf("\n\n\t Percent power saving =%fpercent",r*100);
```

Scilab code Exa 2.11 Program to calculate signal parameters

```

//Exa:2.11
clc;
clear;
clc;
close;
//Given:
//Vt=[50+205msin(5*10^3*pi*t)]cos(2*pi*10^7)V
//comparing it with Vt=[Vc+Vesin(Wm*t)]cos(wc*t)V
Vc=50;
Ve=20;
wc=2*%pi*10^7;
wm=5*10^3;
```

```
fm=500/2;
fc=10^7;
m=Ve/Vc;
Pc=Vc*Vc;
Ps=(m*m/4)*Pc*2;
Pt=Pc+Ps;
printf("\n\n\t carrier frequency =%f",fc);
printf("\n\n\t modulating frequency =%f",fm);
printf("\n\n\t modulation index =%f",m);
printf("\n\n\t Total Power =%f",Pt);
```

Scilab code Exa 2.12 To determine if there is interference

```

clc;
clear;
//Given:
spacing=20//in Hz
bg=100//in Hz
bw=5//modulated by a signal Hz
printf("\n\n\t first sideband pair %d to 100Hz and
    100Hz to %d",(bg-bw),(bg+bw) );
//For second pair
bg2=120// in Hz
bw=5//modulated by a signal Hz
printf("\n\n\t second sideband pair %d to 120Hz and
    120Hz to %d",(bg2-bw),(bg2+bw) );
printf("No overlap occurs");
```

Scilab code Exa 2.13 Program to calculate modulation index

```

//Exa:2.13
clc;
clear;
close;
//Given:
Vp=10;//peak voltage in volts
Vu=8;//unmodulated carrier value
m=(Vp-Vu)/Vu
printf("\n\n\t modulation index = %f ",m);
```

Scilab code Exa 2.14 Program to calculate modulation index

```

//Exa:2.14
clc;
clear;
close;
//Given:
Smax=5;//maximum span in V
Smin=1;//minimum span in V
m=(Smax-Smin)/(Smax+Smin);
```

```
printf("\n\n\t modulation index = %f percent",m*100);
```

Scilab code Exa 2.15 Program to calculate modulation index

```

//Exa:2.15
clc;
clear;
close;
//Given:
Smax=8;//maximum span in V
Smin=0;//minimum span in V
m=(Smax-Smin)/(Smax+Smin);
printf("\n\n\t modulation index = %f percent",m*100);
```

Scilab code Exa 2.16 Program to calculate total power in the modulated signal

```

//Exa:2.16
clc;
clear;
close;
//Given:
```

```
Pc=250;//in Watts
m=0.9;//depth
Pt=Pc*(1+(m^2)/2);
printf("\n\n\t total power of modulated signal = %f
   W ",Pt);
```

Scilab code Exa 2.17 Program to calculate power in the carrier signal

```

//Exa:2.17
clc;
clear;
close;
//Given:
Pt=1000;//in Watts
m=0.95;//depth
Pc=Pt/(1+(m^2)/2);
printf("\n\n\t carrier power = %f W ",Pc);
```

Scilab code Exa 2.18 Program to calculate power in the carrier signal

```

//Exa:2.18
clc;
clear;
```

```
close;
//Given:
Pt=100;//in Watts
m=0.25;//depth
Pmax=100//maximum power transmission capable
Pc=Pt/(1+(m^2)/2);
printf("\n\n\t carrier power = %f W ",Pc);
printf("\n\n\t sidebands have the remaining %f W ",
    Pmax-Pc);
```

Scilab code Exa 2.19 Program to calculate modulation index

```

//Exa:2.19
clc;
clear;
close;
//Given:
Pc=850;//in terms of watts
Pt=1200;//in terms of watts
m=sqrt(2*Pt/Pc)-1;
printf("\n\n\t modualtion index = %f percent",m*100)
    ;
```

Scilab code Exa 2.20 Program to calculate modulation index

```
//Exa:2.20
clc;
clear;
close;
//Given:
Pc=10;//in terms of watts
Pt=12;//in terms of watts
m=sqrt(2*(Pt/Pc-1));
printf("\n\n\t modualtion index = %f percent",m*100)
    ;
```

Chapter 3

FREQUENCY AND PHASE MODULATION

Scilab code Exa 3.1 Program to calculate deviation and modulation index

```

//Exa:3.1
clc;
clear;
close;
//Given:
w1=4.8;//in KHz
v1=2.4;
v2=7.2;
fm=0.5;//in KHz
w2=w1*v2/v1;//in KHz
v3=10;
w3=w1*v3/v1;
m1=w1/fm;
m2=w2/fm;
m3=w3/fm;
printf("\n 1)deviation = %f KHz and modulation index
    = %f",w1,m1);
printf("\n 2)deviation = %f KHz and modulation index
```

```
= %f",w2,m2);
printf("\n 3)deviation = %f KHz and modulation index
= %f",w3,m3);
```

Scilab code Exa 3.2 Program to calculate Power dissipated

```

//Exa:3.2
clc;
clear;
close;
//Given:
//V=12*sin(6*10^8*t+5*sin(1250)*t);
//Compairing it with V=A*sin(wc*t+mf*sin(wm)*t);
wc=6*10^8;
wm=1250;
mf=5;
A=12;
R=10;
Vrms=A/sqrt(2);
fc=wc/2/%pi;
fm=wm/2/%pi;
P=Vrms^2/R;
printf("\n Fc = %f MHz",fc/10^6);
printf("\n Fm = %fHz",fm);
printf("\n Power = %fW",P);
```

Scilab code Exa 3.3 `Program to calculate modulation index`

```
//Exa:3.3
clc;
clear;
close;
//Given:
Bw=12;//in KHz
Mod_mn=300;
Mod_mx=3000;
dev=6;//in KHz
m1={(Bw-dev)*1000}/Mod_mn;
m2={(Bw-dev)*1000}/Mod_mx;
printf("\n 1)modulation indexat 300Hz = %f ",m1);
printf("\n 2)modulation indexat 3000Hz = %f ",m2);
```

Scilab code Exa 3.4 `Program to Bandwidth required`

```
//Exa:3.4
clc;
clear;
close;
```

```
//Given:
w=10;//in KHz
wm=2;//in KHz
mf=w/wm;
Bw=wm*8*2;//for 5 highest coefficient of J in Bessel
    function is 8
printf("\n modulation index = %f ",mf);
printf("\n Band width required = %f KHz",Bw);
```

Scilab code Exa 3.5 Program to determine equations of FM and PM

```

//Exa:3.5
clc;
clear;
close;
//Given:
fc=25*10^6;//in Hz
fm=400;//in Hz
A=4;//in volts
wc=2*%pi*fc;
wm=2*%pi*fm;
w=10000;
mf=w/fm;
printf("\n a)Eq:FM \nV=%f*sin(%f*t+%f*sin(%f)*t)",A,
   wc,mf,wm);
printf("\n\n b)Eq:PM \nV=%f*sin(%f*t+%f*sin(%f)*t)",
   A,wc,mf,wm);
fm2=5*fm;
mf12=mf/5;
mf22=mf;
```

```
wm=2*%pi*fm2;
printf("\n\n c)Eq:FM \nV=%f*sin(%f*t+%f*sin(%f)*t)",
   A,wc,mf12,wm);
printf("\n\n d)Eq:FM \nV=%f*sin(%f*t+%f*sin(%f)*t)",
   A,wc,mf22,wm);
```

Scilab code Exa 3.6 Program to modulation index

```

//Exa:3.6
clc;
clear;
close;
//Given:
dev=100;//in KHz
mod_f=15;//in KHz
m1=dev/mod_f;
m2=3*dev/mod_f;
t=m1/m2;
printf("\n modulation index = %f ",m1);
printf("\n modulation index = %f ",m2);
printf("\n when deviation is tripled m becomes =
   %ftimes ",t);
```

Scilab code Exa 3.7 Program to Bandwidth required

```
//Exa:3.7
clc;
clear;
close;
//Given:
dev=100;//in KHz
mod_f=15;//in KHz
Bw=2*(dev+mod_f);
printf("\n Band width required = %f KHz",Bw);
```

Scilab code Exa 3.8 Program to calculate deviation to be used

```
//Exa:3.8
clc;
clear;
close;
//Given:
Bw=150;//in KHz
mod_f=10;//in KHz
dev=Bw/2 - mod_f;
printf("\n deviation to be used = %f KHz",dev);
```

Scilab code Exa 3.9 Program to calculate modulation index

```
//Exa:3.9
clc;
clear;
close;
//Given:
Bw=12;//in KHz
Mod_mn=300;
Mod_mx=3000;
dev=6;//in KHz
m1={(Bw-dev)*1000}/Mod_mn;
m2={(Bw-dev)*1000}/Mod_mx;
printf("\n 1)modulation indexat 300Hz = %f ",m1);
printf("\n 2)modulation indexat 3000Hz = %f ",m2);
```

Scilab code Exa 3.10 Program to determine realtive amplitude of each sideband

```
//Exa:3.10
clc;
clear;
close;
//Given:
mf=1.0;
Jo=0.77;
//According the bessels Function table
j1=0.44;
j2=0.11;
j3=0.02;
printf("\n Firstsideband pairs :J1 = %f",j1);
printf("\n Secondsideband pairs :J2 = %f",j2);
```

```
printf("\n Thirdsideband pairs :J3 = %f",j3);
```

Scilab code Exa 3.11 `Program to determine realtive amplitude of each sideband`

```

//Exa:3.11
clc;
clear;
close;
//Given:
mf=1.0;
fc=1;//in MHz
fm=10;//in KHz
//According the bessels Function table
j0=0.22
j1=0.58;
j2=0.35;
j3=0.13;
j4=0.03
printf("\n Firstsideband pairs :J1 = %f",j1);
printf("\n Secondsideband pairs :J2 = %f",j2);
printf("\n Thirdsideband pairs :J3 = %f",j3);
printf("\n Forthsideband pairs :J4 = %f",j4);
```

Chapter 4

PRINCIPLES OF AM MODULATION

Scilab code Exa 4.1 Program to determine percent increase in signal power of a car

```
//Exa:4.1
clc;
clear;
close;
//Given:
//Ec^2=Pk
printf("\n After 100percent modulation energy gets
   doubled");
printf("\n So, \n (2Ec)^2=4*Pk");
printf("4*Pk/2=2Pk The SIGNAL POWER GETS DOUBLED \N
   THIS IS TYPICAL OF DIGITAL MODULATION SYSTEM WITH
    ON-OFF Keys OR OOK signal");
```

Scilab code Exa 4.2 Program to calculate turn ratio for modulation transformer

```
//Exa:4.2
clc;
clear;
close;
//Given:
Ebb=2000;//in volts
i=2.5;//in Amps
Pr=1600;//in ohms
Rm=Ebb/i;
tr=Pr/Rm;
printf("\n turn ratio = %d:1 ",tr);
```

Scilab code Exa 4.3 Program to calculate total RF power

```
//Exa:4.3
clc;
clear;
close;
//Given:
diss=1000;//in watts
```

```
eff=0.75;
m=0.50;
Po=eff*diss*4;
Dc=Po+diss;
printf("\n DC power = %fW ",Dc);
Pt=Po*(1+m/4);
printf("\n Total RF power = %fW ",Pt);
```

Scilab code Exa 4.4 Program to calculate carrier power efficiency carrier amplitud

```

//Exa:4.3
clc;
clear;
close;
//Given:
Pt=50000;//in watts
m=0.7071;
Sx=m^2/2;
Sc=Pt/(1+Sx)/1000;
printf("\n 1)carrier power = %fkW ",Sc);
n=Sc*Sx/(Sc+Sc*Sx);
printf("\n 2)efficiency = %f percent",n*100);
Ac=sqrt(2*Pt*Sc);
printf("\n 3)carrier amplitude = %f kv",Ac/1000);
```

Scilab code Exa 4.5 Program to calculate equivalent resistance inductor capacitor

```
//Exa:4.5
clc;
clear;
close;
//Given:
Vcc=12.5;//in volts
Po=2.5;//in watts
Rl=50;
f=27.5*10^6;//in hz
Ri=Vcc^2/2/Po;
n=Rl/Ri;
printf("\n 1)equivalent ressitance = %fohm ",Ri);
Xl=Ri*sqrt(n-1);
l=Xl/(2*%pi*f);
printf("\n 2)indutance = %f microhenry",l*10^6);
Xc=Ri*n/sqrt(n-1);
c=1/{Xc*(2*%pi*f)};
printf("\n 3)capacitance = %f pF",c*10^12);
```

Scilab code Exa 4.6 Program to calculate carrier power contents

```
//Exa:4.6
clc;
clear;
close;
//Given:
Pc=1500;//in watts
m1=0.7;
m2=0.5;
P=m1^2*Pc/4;
printf("\n 1)Pusb=Plsb= %fW ",P);
P=m2^2*Pc/4;
printf("\n 2)Pusb=Plsb= %fW ",P);
```

Scilab code Exa 4.7 Program to calculate required frequency in two cases

```
//Exa:4.7
clc;
clear;
close;
//Given:
Fc=540;//in kHz
Fif=455;//in kHz
Flo=Fc+Fif;
printf("\n 1)when local oscillator tracks above
   frequency of received frequency \n Flo= %fKHz ",
   Flo);
Flo=Fc-Fif;
printf("\n 1)when local oscillator tracks below
   frequency of received frequency \n Flo= %fKHz ",
```

```
   Flo);
```

Chapter 5

PRINCIPLES OF FM MODULATION

Scilab code Exa 5.1 Program to calculate capacitive reactance

```

//Exa:5.1
clear;
close;
//Given:
n=9;
gm=12*10^-3;
X=n/gm;
printf("\n capacitive reactance = %fohm ",X);
```

Scilab code Exa 5.2 Program to calculate capacitive reactance

```
//Exa:5.1
clear;
close;
//Given:
fn=5*10^7;//in ohms
gm=(9*10^-3)/8;
C=50*10^-12;
Cx=gm/2*%pi*fn;
r=sqrt(1+Cx/C);
u=0.0173*fn;
Fv=2*u;
printf("\n frequency variation = %f MHz",Fv/10^6);
```

Scilab code Exa 5.4 Program to calculate number of stations

```
//Exa:5.4
clc;
clear;
close;
//Given:
Fd=40;//in kHz
Fc=101.6;//in MHz
Fm=8;//in KHz
Fs=2*Fd;
mf=Fd/Fm;
FH=(Fc*1000+Fd)/1000;
FL=(Fc*1000-Fd)/1000;
printf("\n\t carrer swing = %f",Fs);
printf("\n modulation index = %f",mf);
```

```
printf("\n\t Highest frequency = %f MHz",FH);
printf("\n\t lowest frequency = %f MHz",FL);
```

Scilab code Exa 5.5 Program to calculate carrier swing

```

//Exa:5.5
clc;
clear;
close;
//Given:
Fmx=107.218;//in MHz
Fmn=107.196;//in MHz
fm=4;//in Khz
swing=Fmx-Fmn;//in MHz
fd=swing/2;
fc=Fmx-fd;
m=(fd*10^3)/fm;
printf("\n\t carrer swing = %f MHz",swing);
printf("\n frequency deviation = %f KHz",fd*10^3);
printf("\n career frequency = %f",fc);
printf("\n modulation index = %f",m);
```

Scilab code Exa 5.6 Program to calculate percentage modulation

```
//Exa:5.6
clc;
clear;
close;
//Given:
Fmx=88;//in MHz
Fmn=108;//in MHz
fd=15;//in kHz
m1=fd/75;
m2=fd/25;
printf("\n percentage modulation = %f",m1*100);
printf("\n percentage modulation = %f",m2*100);
```

Scilab code Exa 5.7 `Program to calculate value of capacitance`

```
//Ex:5.7
clc;
clear;
close;
//Given:
Ri=450;//in ohms
B=80;
b1=60;
b2=100;
R2=20000;//inthe question it is given 2Kohm but 20
    Kohm is used
C2=500*10^-12;
Ceq=B*R2*C2/(Ri+R2);
lv=b1*R2*C2/(Ri+R2);
hv=b2*R2*C2/(Ri+R2);
```

```
printf("\n\t value of capacitance = %f nF",Ceq*10^9);
printf("\n lower value of capacitance = %f nF",lv*10^9);
printf("\n higher value of capacitance = %f nF",hv*10^9);
```

Scilab code Exa 5.8 Program to determine range of capacitance

```
//EX:5.8
clc;
clear;
close;
//Given:
R=90*10^3;//in Kohms
C=100*10^-12;//in pF
g1=2800*10^-6;
g2=4300*10^-6;
lv=g1*R*C;
hv=g2*R*C;
printf("\n lower value of capacitance = %f nf",lv*10^9);
printf("\n higher value of capacitance = %f nF",hv*10^9);
```

Scilab code Exa 5.9 Program to calculate frequency deviation

```

//Exa:5.9
clc;
clear;
close;
//Given:
Pd=10;//in degrees
Fm=100;//in Hz
fd=Pd*(%pi/180)*Fm;
printf("\n frequency deviation = %f KHz",fd*10^-3);
```

Scilab code Exa 5.10 Program to calculate bandwidth

```

//Exa:5.10
clear;
close;
//Given:
m=2.4;
BW=2*(m+1);
printf("\n bandwidth = %ffm ",BW);
```

Chapter 6

DEMODULATION OF AM WAVES

Scilab code Exa 6.1 Program to calculate image frequency rejection ratio

```

//Exa:6.1
clc;
clear;
close;
//Given:
f1=1000;//in KHz
f2=25;//in MHz
IF=455//in KHz
Q=100;
fs1=f1+2*IF;
p1=fs1/f1 - f1/fs1;
a1=sqrt(1+ Q^2*p1^2);
printf("\n\n\t (a)image frequency is %f KHz \n
    rejection ratio is %f dB",fs1,20*log10(a1));
fs2=f2+2*IF/1000;
p2=fs2/f2 - f2/fs2;
a2=sqrt(1+ Q^2*p2^2);
printf("\n\n\t (b)image frequency is %f KHz \n
```

```
rejection ratio is %f dB",fs2,20*log(a2));
```

Scilab code Exa 6.2 Program to calculate maximum modulation index

```

//Exa:6.2
clc;
clear;
close;
//Given:
R1=110;//in Kohm
R2=220;//in kohm
R3=470;//in kohm
R4=1;//in Mohm
Rc=R1+R2;
Zin=(R4*1000*R2*R3)/(R2*R3+R3*R4*1000+R4*R2*1000) +
    R1;
Mmax=Zin/Rc;
printf("\n\n\t maximum modulation index = %f percent
    ",Mmax*100);
```

Scilab code Exa 6.3 Program to calculate number of stations

```

//Exa:6.3
```

```
clc;
clear;
close;
//Given:
BWt=100; //in kHz
Fh=5; //in KHz
n=BWt/(2*Fh);
printf("\n\t number of stations = %f",n);
```

Scilab code Exa 6.4 Program to calculate number of stations

```

//Exa:6.4
clc;
clear;
close;
//Given:
Fmax=3; //in kHz
Bw=20; //in MHz
Bs=2*Fmax*1000;
n=(20*1000000)/Bs;
printf("\n\t number of stations = %f",n);
```

Chapter 7

DEMODULATION OF FM WAVES

Scilab code Exa 7.1 Program to determine slope detector

```

//Exa:7.1
clc;
clear;
close;
//Given:
deviation=3;//in KHz
roll_off=6;//in KHz
full_dev=deviation*roll_off;
half_dev=full_dev/2;
printf("\n full deviation = %f dB\tcorresponds to
    12.6mV",-full_dev);
printf("\n half deviation = %f dB\tcorresponds to
    35.5mV",-half_dev);
```

Scilab code Exa 7.2 Program to determine minimum maximum capture and lock frequenc

```
//Exa:7.2
clc;
clear;
close;
//Given:
free_f=5.8;//in MHz
lock1=17;//in percent
lock2=23;//in percent
lock_feq=lock1*free_f/100;
min1=free_f-lock_feq;
max1=free_f+lock_feq;
printf("\n 1)Lock range is from %fMHz to %fMHz",min1,max1);
lock_feq=lock2*free_f/100;
min2=free_f-lock_feq;
max2=free_f+lock_feq;
printf("\n 2)Lock range is from %fMHz to %fMHz",min2,max2);
```

Scilab code Exa 7.3 Program to calculate total gain

```
//Exa:7.3
clc;
clear;
close;
//Given:
lim_signal=30*10^-3;//in mV
in_sig=5*10^-6;//in uV
to=lim_signal/in_sig;
t=20*log10(to);
printf("\n total gain= %fdB",t);
```

Scilab code Exa 7.4 `Program to calculate total gain`

```
//Exa:7.4
clc;
clear;
close;
//Given:
lim_signal=20*10^-3;//in mV
in_sig=2*10^-6;//in uV
to=lim_signal/in_sig;
t=20*log10(to);
printf("\n total gain= %fdB",t);
```

Scilab code Exa 7.5 Program to calculate number of stations

```
//Exa:7.5
clc;
clear;
close;
//Given:
F=6*10^6;
each=400*10^3;
n=F/each;
printf("\n total number of stations= %fstations",n);
```

Scilab code Exa 7.6 Program to determine slope detector

```
//Exa:7.6
clc;
clear;
close;
//Given:
deviation=3;//in KHz
roll_off=6;//in KHz
full_dev=deviation*roll_off;
```

```
half_dev=full_dev/2;
printf("\n full deviation = %f dB\tcorresponds to
   12.6mV",-full_dev);
printf("\n half deviation = %f dB\tcorresponds to
   35.5mV",-half_dev);
```

Chapter 8

PULSE MODULATION SYSTEMS

Scilab code Exa 8.1 Program to calculate nyquist sampling rate

```

//Exa:8.1
clc;
clear;
close;
//Given:
Bw=12;//in KHz
Nr=2*Bw;
printf("\n Nyquist sampling rate = %f KHz",Nr);
```

Scilab code Exa 8.2 Program to calculate nyquist sampling rate

```
//Exa:8.2
clc;
clear;
close;
//Given:
Bw=3300;//in Hz
resol=10;//bits per sample
Nr=2*Bw;//samples per second
Brate=Nr*resol;
DigBw=5*Brate;
printf("\n Nyquist sampling rate = %f Hz",Nr);
printf("\n Bit rate is = %f bits/sec",Brate);
printf("\n Digital bandwidth = %f KHz",DigBw/1000);
```

Scilab code Exa 8.3 Program to calculate nyquist sampling rate elements transmitte

```
//Exa:8.3
clc;
clear;
close;
//Given:
Bw=25;//in KHz
Nr=2*Bw*1000;
printf("\n Nyquist sampling rate = %f samples/
   seconds",Nr/1000);
br=8;//bits
Ne=br*Nr;
printf("\n Transmitted elements = %f elements",Ne);
```

Chapter 9

DIGITAL PULSE MODULATION

Scilab code Exa 9.1 Program to determine output band and minimum required bandwidt

```
//Exa:9.1
clc;
clear;
close;
//Given:
Fm=80;//in MHz
Fc=60;//in MHz
Fb=20;//in Mbps
MI=(Fm-Fc)/Fb;
printf("\n Modulation index is = %f",MI);
printf("\n Minimum required bandwidth = %fMHz",Fc);
printf("\n Baud rate = %fMegabaud",Fb);
```

Scilab code Exa 9.2 Program to determine output band and minimum required bandwidt

```
//Exa:9.2
clc;
clear;
close;
//Given:
//Output = (sin(wa)t)(sin(wc)t)
//        =[sin(2pi5*10^6t)][sin(2pi70*10^6t)]
//        =1/2[cos(2pi*65*10^6t)-cos(2pi*75*10^6t)]
        '
w1=65;//in MHz
w2=75;//in MHz
F=w2-w1;
printf("\n Minimum lower side frequecy = %fMHz",w1);
printf("\n Maximum upper side frequency = %fMHz",w2)
    ;
printf("\n Minimum nyquist bandwidth = %fMHz",F);
printf("\n Baud rate = %fMegabaud",F);
```

Scilab code Exa 9.3 Program to determine bitrate

```
//Exa:9.3
clc;
clear;
close;
//Given:
Bw=4000;//in Hz
nQl=128;//i.e. =2^7;//quantizing levels
NyqR=2*Bw;
n=7;
Total_bpersmple=8;
total_smples=NyqR*Total_bpersmple;
printf("\n nyquist sampling rate = %fHz",NyqR);
printf("\n Bit rate = %fKb/sec",total_smples);
```

Scilab code Exa 9.4 Program to determine maximum value of bitrate

```

//Exa:9.4
clc;
clear;
close;
//Given:
Pe=10^-5;
v=0.5;
n=2*10^-6;
x=3.02;//at erfc(x)=2*10^-5 at x=3.02
T=(x^2*n)/(4*v^2);
B=1/T;
printf("\n Minimum Time Period= %f 10^-6",T*10^6);
printf("\n Maximum Bit rate = %fKb/sec",B/10^3);
```

Scilab code Exa 9.5 Program to determine expression for signal

```

//Exa:9.5
clc;
clear;
close;
//Given:
//v1(t)=10cos(2000*pi*t)+4sin(200*pi*t);
//BPF=800Hz to 1200Hz
printf("Since v2=v1+0.1*v1^2 and");
printf("\n BPF=800Hz to 1200Hz ");
printf("\n So,v3(t)=10*cos(2000*pi*t)+4*sin(2200*pi*
    t)-4*sin(1800*pi*t)");
```

Scilab code Exa 9.6 Program to print expression for frequency spectrum

```

//Exa:9.6
clc;
clear;
close;
//Given:
Ts=3;//in dB
```

```
fs=1/Ts;
printf("\n Ms(f)=p(f)[fs*sigma{M(f-Kfs)}]");
printf("\n 1/(1/30B)*integ(Ae^-j2*pi*f*t)dt from -60
   dB to 60dB");
printf("\n AsinC(2*pi*f/60B)");
```

Scilab code Exa 9.7 Program to determine maximum signal to quantisation noise rati

```

//Exa:9.7
clc;
clear;
close;
//Given:
c=3500;
b=60000;
l=b/2/c;
M=2^l;
Vrms=0.2;//in volts
s=4/M;
SQNR=20*[log10(Vrms/s)]+10*log10(12);
printf("\nThe signal to quantisation ratio is(SQNR)=
   %fdB",SQNR);
```

Scilab code Exa 9.8 Program to number of bits required per sample minimum bandwidt

```

//Exa:9.8
clc;
clear;
close;
//Given:
SQNR=20;
Nq=3*10^-4;//in Watts
S=sqrt(12*3*10^-4);
M=(3.8+3.8)/S;//-3.8V to 3.8V signal variation
n=log2(M);
printf("Number of bits per sample %f",n);
```

Scilab code Exa 9.9 Program to calculate the capacity of a standard telephone chan

```

//Exa:9.9
clc;
clear;
close;
//Given:
SNR=32;
Actual_snr=10^(SNR/10);
w1=300;//in Hz
w2=3400;//in Hz
w=w2-w1;
c=w*log2(1+Actual_snr);
printf("\n Capacity=%f bits/sec",c);
```