Scilab Textbook Companion for Feedback Circuits And Operational Amplifiers by D. H. Horrocks[1]

Created by
Naveen Varma Champathi
B.Tech (pursuing)
Electronics Engineering
National Institute of Technology, Nagpur
College Teacher
Eswar Rao
Cross-Checked by
Aditya Sengupta, IIT Bombay

May 18, 2016

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Feedback Circuits And Operational Amplifiers

Author: D. H. Horrocks

Publisher: Chapman And Hall

Edition: 2

Year: 1990

ISBN: 0-412-34270-7

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 2

GENERAL PROPERTIES OF FEEDBACK AMPLIFIERS

Scilab code Exa 2.1 feedback fraction

```
//example1:

printf(" Given:")
disp("value of A=1000")//To display given values
A=1000
disp("Magnitude of closed loop gain with feedback,Af=10")
Af=10
disp("Af=(A)/(1+A*b)")//standard formula for closed loop gain
disp("b=((A/Af)-1)*(1/A)")
b=((A/Af)-1)*(1/A)
printf(' The value of b is %f',b)
//if,A becomes 900angle(-30 degrees)
A=900*cos(-%pi/6)+%i*900*sin(-%pi/6)
disp("If,A becomes 900angle(-30 degrees)")
disp("Af=(A)/((1+A*b))")//standard formula for closed loop gain
Af=(A)/((1+A*b))
```

```
r=real(Af)//To get real part of Af
i=imag(Af)//To get imaginary part of Af
printf(' The value of Af is (%f)+i(%f)',r,i)//to
    display value of Af
t=tan(i/r)//to get angle in radians of Af
t=t*180/%pi//to get angle in degrees
m=abs(Af)//to get magnitude of Af
printf('\n Af=%fangle(%f degrees)',m,t)
printf('\n Af=10.004angle(-0.32 degrees)')//rounding
     to 3 digits
```

Scilab code Exa 2.2 Expected variations in closed loop gain

```
//To calculate the expected variations in closed
    loop gain
disp("Given:")
disp("Af=10+0.2% or 10-0.2%")//To display given
    values
disp("-->Af=10 and dAf/Af=0.2%")
disp("The available amplifiers are with gains
    50,500,5000")
disp("dA/A=20%")
disp("We have from formula")
disp("dAf/Af=(1/(1+A*b))*(dA/A)")//standard formula
    for closed loop gain
disp("let f=0.2%=(dAf/Af),a=20%=(dA/A),k=(1+A*b)")
disp("Therefore we have:")
Af=10;
printf(" Af=%d\n",Af)
f=0.2/100;
printf(" f=%f\n",f)
a=20/100;
printf(" a=%f",a)
disp("Therefore,the above formula becomes f=(1/k)*a"
    )
```

```
k=a/f
printf(" Therefore (1=A*b)>=%d,but,A=(1+A*b)*Af",k)
l=k*Af//store partial value
disp("A>=k*Af")//given condition
printf(" Therefore we got the value of A to be >=%d",l)//display value of A
printf("\n Therefore A=5000 satisfies the obtained condition.Therefore A=5000 is most economical")
A=5000
b=((A/Af)-1)*(1/A)
printf(" The value of b is %f",b)
disp("Pratical values of A are 4000 and 6000,since dA/A=(+-20%) and A=5000")
disp("Therefore ")
Afmin=(4000/(1+(4000*b)))//to print result
Afmax=(6000/(1+(6000*b)))//to print result
printf(" Afmin=%1.3f",Afmin)
devAfmin=(1-(Afmin/Af))*100;//deviation of Afmin from Af
printf("(%1.2f percent low)",devAfmin)//to display deviation from original value
printf("\n Afmax=%2.3f",Afmax)
devAfmax=(1-(Afmax/Af))*100;//deviation of Afmax from Af
dev=abs(devAfmax);//negative value of devAfmax indicates Afmax is greater than Af by abs(devAfmax)
printf("(%1.2f percent high) ",dev)//to display deviation from original value
```

Chapter 3

AMPLIFIERS WITHOUT FEEDBACK

Scilab code Exa 3.1 Output voltage and gain of two stage amplifier

```
//example1:

printf(" Given:")
disp("The value of Rin=Rout=100kohms,Rload=1kohm,Kv
   =100")//To display given values
Rin=100//assigning given values
Rout=100
Rload=1
Kv=100
disp("All resistances are in kohms")
disp("The whole two stage amplifier is fed by a
   generator having a voltage of Eg=1mV when no load
    and having self resistance Rg=20kohms")
Eg=(1/1000)//Eg converted to volts
Rg=20//all resistances are in kohms
disp("Since amplifiers are identical")//given
disp("Rin1=Rin")
disp("Rin2=Rin")
disp("Rout1=Rout")
```

```
disp("Rout2=Rout")
Rin1=Rin//assigning given values
Rin2=Rin
Rout1=Rout
Rout2=Rout
disp("Vout2 is given by equation")
disp("Vout2=((Eg)*(Rin1/(Rg+Rin1)))*(Kv)*(Rin2/(Rout1+Rin2))*(Kv)*(Rload/(Rout2+Rload))")
Vout2=((Eg)*(Rin1/(Rg+Rin1)))*(Kv)*(Rin2/(Rout1+Rin2))*(Kv)*(Rload/(Rout2+Rload))//equation for Vout2
printf(" Therefore Vout2=%f",Vout2)
disp("Vin is given by equation ")
disp("Vin=(Eg*(Rin1/Rg+Rin1))")
Vin=(Eg*(Rin1/(Rg+Rin1)))//Equation for Vin
disp("Now overall gain ")
disp("Av=(Vout2/Vin)")
Av=(Vout2/Vin)//Equation for Av
printf(" Therefore Vout2=%f and overall voltage gain Av=%f",Vout2,Av)//To print the required values.
```

Scilab code Exa 3.2 single stage amplifier parameters

```

rin=10000;//in ohms
Kv=100;//constant for voltage-controlled voltage-source
rout=1000;//in ohms
//load resistance Rload
Rload=4000;//in ohms
C2=50*(10**-12);//in Farads
disp('Vout=Vin*(input voltage coupling)*Kv*(output voltage coupling)');
disp('(i)Mid band gain');
disp('Av=(Vout/Vin)=Kv*(Rload/(rout+Rload))');
Av=Kv*(Rload/(rout+Rload));
```

```
disp('(ii)For the upper half-power frequency');
disp('fu=(1/(2*%pi))*(1/(C2*(rout||Rload)))');
fu=(1/(2*%pi))*(1/(C2*((rout*Rload)/(rout+Rload))));
//Let coupling capacitance be C
disp('(iii)For coupling capacitance C ');
disp('fl=(1/(2*%pi))*(1/(C*rin))');
fl=10;//given lower half-power frequency fl
disp('C=1/(2*%pi*fl*rin)');
C=1/(2*%pi*fl*rin);
disp('RESULTS:\n');
printf('(i)Mid-band gain=%d\n',Av);
printf('(ii)Upper half-power frequency=%1.2f MHz\n',
    fu/(10**6));//fu divided by 10^6 to convert into
    MHz
printf('(iii)Coupling capacitance C for fl to be 10
    Hz is %1.2f uF\n',C*(10**6));// C multiplied by
    10^6 by convert to microFarads
```

Scilab code Exa 3.3 Common mode rejection ratio and differential mode

```

disp("given:")
disp("The voltage generated by an transducer which
    is connected to differential amplifier is Vdm=50
    mV ")
Vdm=50*(1/1000)//Vdm in volts
disp("Vcm=5V")
Vcm=5
disp("Vout should be equal to 10V")
Vout=10
disp("The unwanted output component owing to the
    common-mode input is to be less than 1% of the
    wanted component i.e Voutcm=1% of Voutdm")
Voutcm=(1/100)*10
printf(" Voutcm=%f",Voutcm)
```

```
disp("Let amplifier differential-mode gain be Adm")
disp("Vout,Admand Vdm are related as")
disp("Vout=Adm*Vdm------(1)")
disp("Adm=Vout/Vdm")
Adm=Vout/Vdm
printf(" Therefore amplifier differential-mode gain(Adm)=%d",Adm)
disp("Equation (1)can also be used for calculating unwanted output component")
disp("Unwanted components arise when operating in common-mode")
disp("Acm=Voutcm/Vcm")
Acm=Voutcm/Vcm
printf(" Common-mode gain Acm=%f",Acm)
disp("Now CMRR can be calculated using relation")
disp("CMRR=Adm/Acm")
CMRR=Adm/Acm
printf(" There fore CMRR of amplifier should be greater than or equal to %f",CMRR)
printf(" CMRR&gt;=%d",CMRR)
```

Chapter 4

FEEDBACK AMPLIFIER CIRCUITS

Scilab code Exa 4.1 input resistance of a feedback amplifier

```
//input resistance in Kohm
Rin=1;
//voltage gain
Av=1000;
//feedback fraction
Bv=0.1;
//Let input resistance after feedback is applied be
   Zif
AB=Av*Bv;
Zif=Rin*(1+AB);//Zif in Kohms
printf("RESULTS:\n");
printf("Input resistance after feedback is applied
   Zif=%dKohms",Zif);
```

Scilab code Exa 4.2 input resistance of a feedback amplifier

```
//input impedance in ohms
Rin=10000;
//Trans-resistance in ohms
Rt=(10)^(5);
//feedback fraction Bg
Bg=(10)^(-3);
AB=Rt*Bg;
//input impedance after feedback is applied Zif in
   ohms
Zif=Rin/(1+AB);
printf("RESULTS:\n");
printf("Input impedance after feedback applied is
   Zif=%d Ohms",Zif);
```

Scilab code Exa 4.3 parameters of a feedback amplifier

```
//Trans-resistance Rtf in Kohms
Rtf=(-10);
//given
R1=20;//in Kohms
R2=5;
Rc=1;
//transistor parameters in kohms
hie=1;
hfe=0.1;
//feedback fraction Bg
Bg=1/Rtf;
R=(-Rtf);
//input resistance ri in ohms
a=(R1*R2*hie)/((R1*R2)+(R1*hie)+(R2*hie));
ri=(a);
//Output resistance in ohms
ro=1000*Rc;
b=(R1*R2)/(R1+R2);
//let
```

```
i1=1;
ib=i1*b/(b+hie);
ic=hfe*ib;
//output voltage Vo
Vo=(-Rc*ic);
Rt=Vo/i1*1000;
//Feedback factor F
F=1+(Rt*Bg);
//closed loop gain Rtf
Rtf=Rt/F;
//closed loop input resistance rif
rif=ri/F;
//closed loop output resistance rof
rof=ro/F;
printf("RESULTS\n");
printf("closed-loop gain,Rtf=%1.2fKohms\n",Rtf);
printf("closed-loop input resistance rif=%2.0f Ohms\n",1000*rif);//in ohms
printf("closed-loop output resistance rof=%d Ohms",rof);
```

Scilab code Exa 4.4 series voltage feedback circuit

```
//caption:series voltage feedback circuit
//given in ohms
R1=10000;
R2=2000;
Rc=2000;
//transistor parameters in Ohms
hie=1000;//in ohms
hfe=100;
//loop gain Avf
Avf1=(R1+R2)/R2;//there is corrected equation.in text book,it needs a correction
Bv=1/Avf1;
```

```
//forward voltage gain Av
Av=(0.5)*hfe*Rc/hie;
//feedback factor F
F=(1+Av*Bv);
//closed loop gain Avf
Avf=Av/F;
//input resistance ri
ri=2*hie;
//input resistance after feed back rif
rif=ri*F;
//output resistance ro
ro=Rc;
//output resistance after feedback rof
rof=ro/F;
printf("RESULTS\n");
printf("(i)voltage amplification Avf=%1.2f, This accords with the previous value of Avf=%d\n",Avf,Avf1);
printf("(ii)input resistance after feed back rif=%2.1fKohms\n",rif/1000);//to convert ohms to Kohms
printf("(iii)input resistance after feed back rof=%dKohms\n",rof);//to convert ohms to Kohms
```

Chapter 5

MORE ABOUT FEEDBACK AMPLIFIERS

Scilab code Exa 5.1 Voltage shunt feedback circuit

```
//caption:shunt voltage feedback circuit
//example5.1
printf("Given:");
printf("R=10 Kohms,\nR1=20Kohms,\nR2=5Kohms,\nhfe
    =100Kohms,hie=1Kohm,\nRc=1Kohm");
R=10000;
R1=20000;//in ohms
R2=5000;
Rc=1000;
hfe=100;
Bg=-1/R;
hie=1000;
printf("Therefore Bg=%1.5f\n",Bg);
printf("RT=input current coupling factor*-hfe*
    effective cdollector resistance\n");
printf("RT=(R1||R2||R)/((R1||R2||R)+hfe)*hfe*(Rc||R)
    \n");
a=(R1*R2*R)/((R1*R)+(R*R2)+(R1*R2));
b=(Rc*R)/(Rc+R);
```

```
RT=(-hfe*a*b)/(a+hie)/1000;
printf("RT=%2.1fKohms\n",RT);
c=1+(RT*1000*Bg);
d=(1/R1)+(1/R2)+(1/R)+(1/hie);
Zi=1/d;
Zo=b;
RTf=RT/c;
rif=Zi/c;
rof=Zo/c;
printf("RESULTS:\n");
printf("RT=%2.1fKohms\n",RT);
printf("Zi=%3.1fohms\n",Zi);
printf("Zo=%dohms\n",Zo);
printf("Closed-loop gain,RTf=%1.2fKohms\n",RTf);
printf("Closed-loop input resistance,rif=%dohms\n",
   rif);
printf("Closed-loop output resistance,rof=%dohms\n",
   rof);
```

Scilab code Exa 5.2 amplifier parameters

```
//caption:amplifier parameters
//given values of resistances
R1=10000;//in ohms
R2=2000;//in ohms
Rc=2000;//in ohms
hie=1000;//in ohms
hfe=100;
ri1=2*hie;
//unloaded feed-backfraction Bv
Bv=R2/(R1+R2);
disp("Av=ri1/(ri1+(R1||R2))*(Vo/Via)");
disp('Av=(1*hfe/2)*(ri1/(ri1+(R1||R2)))*(Rc||(R1+R2)
   )/hie');
Av=(1*hfe/2)*(ri1/(ri1+(R1*R2/(R1+R2))))*(Rc*(R1+R2)
```

```
   /(Rc+R1+R2))/hie;
C=1+(Av*Bv);
//open−loop input resistance ri
ri=ri1+(R1*R2/(R1+R2));
//open−loop output resistance ro
ro=Rc*(R1+R2)/(Rc+R1+R2);
//closed−loop gain,Avf
disp('closed−loop gain, Avf=Av/C,where C=1+Av*Bv');
Avf=Av/C;
//closed−loop input resistance,rif
disp('closed−loop input resistance, rif=ri*C');
rif=ri*C;
//closed−loop output resistance, rof
disp('closed−loop output resistance, rof=ro/C,where
   C=1+Av*Bv');
rof=ro/C;
printf("RESULTS:\n");
printf("Closed−loop gain,Avf=%1.2f(previously 5.66)\
   n",Avf);
printf("Closed−loop input resistance,rif=%2.1f Kohms
   (previously 35.3Kohms)\n",rif/1000);//divided by
   1000 to convert to Kohms
printf("Closed−loop output resistance,rof=%3.0f Ohms
   (previously 113 ohms)\n",rof);
```

Scilab code Exa 5.4 closed loop current gain input output resistance

```

//example5.4
//Shunt circuit feedback configuration//given
R1=10000;//in ohms
R2=2000;//in ohms
Rc1=5000;//in ohms
hie=1000;//in ohms
hfe=100;
```

```
//unloaded feedback fraction B1
B1=R2/(R1+R2);
disp('open loop forward current amplification A1=io/
   i');
iia=1;//let
disp('ib=input-current coupling factor*iia');
disp('ib=(R1+R2)*iia/(R1+R2+hie)');
ib=(R1+R2)*iia/(R1+R2+hie);
disp('V2=(-hfe)*(Rc1||ri2)*ib');
disp('ri2=hie+(1+hfe)*(R1||R2)');
ri2=hie+(1+hfe)*(R1*R2/(R1+R2));
V2=(-hfe)*(Rc1*ri2/(Rc1+ri2))*ib;
disp('io/V2=C=-hfe/(hie+(1+hfe)*(R1||R2))');
C=-hfe/(hie+(1+hfe)*(R1*R2/(R1+R2)));
disp('Open loop current gain=A1=io/i=(ib/i)*(V2/ib)
   *(io/V2)');
A1=(ib/iia)*(V2/ib)*(C);
disp('Open-loop input resistance, ri=(R1+R2)||hie');
ri=(R1+R2)*hie/(R1+R2+hie);
disp('Open-loop output resistance ro as seen by the
   load resistance Rc2 is infinitely large since the
    load is in series with the infinitely large
   collector resistance of the transistor.');
disp('Closed-loop current gain,A1f=A1/(1+A1*B1)');
A1f=A1/(1+A1*B1);
disp(A1f);
disp('Closed-loop input resistance,rif=ri/(1+A1*B1)'
   );
rif=ri/(1+A1*B1);
disp(rif);
disp('Closed-loop output resistance,rof=ro(1+A1*B1),
    since ro is infinitely large,the rof is also
   large infinitely..');
printf(" RESULTS:\n\n");
printf(" A1f=%1.2f,\n\n",A1f);
printf(" rif=%2.1f ohms,\n\n",rif);//approximately
printf(" rof=infinite\n\n");
```

Scilab code Exa 5.5 Voltage shunt feedback circuit

```
//important:In this example ro1 value is contradicting in text book,sometimes they used 1 K0hm and sometimes 4Kohms,the code below used ro1=2KOhms as specified in question
//example5.5
//caption:Shunt voltage feedback circuit
//input resistance ri1
ri1=1000;//in ohms
//output resistance ro1
ro1=2000;//in ohms
//trans resistance Kr
Kr=-10^6;//in ohms
//Feedback resistor R
R=10000;//in ohms
//current source Jg
Jg=0.001;//in Amps
//source internal resistance rg
rg=2000;//in ohms
//load resistance rl
rl=5000;//in ohms
//unloaded feedback fraction
Bg=-1/R;
disp('Unloaded feedback fraction,Bg=-1/R');
disp('open loop gain,RTs=Vo/Jg=input current coupling factor*Kr*output voltage coupling factor');
RTs=((rg*R/(rg+R))/(ri1+(rg*R/(rg+R))))*Kr*((rl*R/(rl+R))/(ro1+(rg*R/(rg+R))));
printf("RTs=%d ohms",RTs/1000);
disp('Open-loop input resistance,ris=rg||R||ri1');
ris=rg*R*ri1/(rg*R+R*ri1+rg*ri1);
printf("\nris=%d ohms",ris);
```

```
disp('Open-loop output resistance,ros=ro1||R||rl');
ros=rl*R*ro1/(rl*R+R*ro1+rl*ro1);
printf("ros=%d Ohms",ros);
disp('Closed-loop gain,RTf=RTs/(1+RTs*Bg)');
RTfs=RTs/(1+RTs*Bg);
printf("\nRTfs=%1.2f Kohms",RTfs/1000);
disp('Closed-loop input resistance,rifs=ris/(1+RTs*Bg)');
rifs=ris/(1+RTs*Bg);
printf("rifs=%d Ohms",rifs);
disp('Closed-loop output resistance,rof=ros/(1+RTs*Bg)');
rofs=ros/(1+RTs*Bg);
printf("rofs=%2.1f Ohms",rofs);
printf("RESULTS:\n");
Vout=RTfs*Jg;
printf("\n(i)The output voltage=%1.2f V",Vout);
disp('rifs=rif*rg/(rif+rg)==>rif=rg*rifs/(rg-rifs)');
rif=rg*rifs/(rg-rifs);
printf("\n(ii)The input resistance seen by the actual signal source is %2.1f Ohms",rif);
disp('rofs=rof*rl/(rof+rl)==>rof=rl*rofs/(rl-rofs)');
rof=rl*rofs/(rl-rofs)
printf("\n(iii)The output resistance seen by the load is %2.1f Ohms",rof);
printf("(iV)The closed-loop gain of the amplifier");
disp('RTf=Vo/i1,where')
disp('i1=current coupling factor*Jg=rg*Jg/(rg+rif)');
RTf=(rg+rif)*RTfs/rg;
printf("(iV)The closed-loop gain of the amplifier circuit is %1.2f Kohms",RTf/1000);//divided by 1000 to convert ohms to Kohms.
```

Scilab code Exa 5.6 feed back fraction

```

//given
//current gain of transistor A
A=100;
//upper half-power frequency fh
fh=2*(10^6);//in Hz
//closed-loop current gain Acl
Acl=10;
//band width BW
BW=10*(10^6);//in Hz
//gain bandwidth product ABWp
ABWp=Acl*BW;//in Hz
//gain bandwidth product of transistor ABWpt
ABWpt=A*fh;//in Hz
printf("The gain bandwidth product of the transistor is %d*(10^8),which is greater the minimum required i.e. %d*(10^8) and therefore the transistor is suitable\n",ABWpt/(10^8),ABWp/(10^8))
disp('Acl=A/(1+A*B)==&gt; B =(1/A)*(A/Acl-1)\n');
B =(1/A)*(A/Acl-1);
printf("Required value of B is %1.2f",B);
```

Chapter 6

THE OP AMPS BASIC IDEAS AND CIRCUITS

Scilab code Exa 6.1 design values of an inverting amplifier

```
R1=1;
//voltage gain of an inverting amplifier Av
Av=-100;
printf("Av=(-R2/R1)\nTherefore R2=-Av*R1");
R2=-Av*R1;
printf("\nRESULTS:\n");
printf("The design values are R1=%d Kohms and R2=%d
    Kohms",R1,R2);
```

Scilab code Exa 6.2 gain of a negative feedback amplifier

```
//Typical 741 type op-amp Differential gain A
A=200000;
R1=1;//in Kohms
```

```
R2=10;
//Circuit gain in negative feedback Av
Av=-(R2/R1)*(A*(R1/(R1+R2))/(1+(A*(R1/(R1+R2)))));
printf("Circuit gain Av=%2.4f",Av);
```

Scilab code Exa 6.4 non inverting amplifier

```

Av=100;
printf("Av=1+(R1/R2)\n");
//Let R2=1 Kohms
R2=1;//in Kohms
printf("R1=(Av-1)*R2");
R1=(Av-1)*R2;
printf("\nRESULTS:\n");
printf("The design values are R1=%d Kohms and R2=%d
    Kohm",R1,R2);
```

Chapter 7

OP AMP NON IDEALITIES

Scilab code Exa 7.1 Output voltage

```
A=200000;
//Offset voltage Vos in mV
Vos=5;
//output voltage Vout=A*Vos in mV
printf("\nVout=A*Vos\n");
Vout=A*Vos;
printf("RESULTS:\n")
printf("The maximum output voltage Vout=%d V",Vout
   /1000);//in volts
```

Scilab code Exa 7.2 output offset voltage

```
printf("A=1+(R1/R2)\n");
A=10;
//out put voltage of an op-amp owing to voltage
   offset effects Vout
```

```
printf("Vout=-A*Vos");
Vos=5;//in mV
Vout=-A*Vos;
printf("\nRESULTS:\n");
printf("Output offset voltage is Vout=%d mV",Vout);
```

Scilab code Exa 7.4 inverting voltage amplifier

```
Av=-10;
disp('Av=-(R2/R1)*(A*B/1+A*B), where A-differential gain and B=R1/R1+R2');
disp('When A*B&gt;&gt;1,Av=-(R2/R1)=Av1');
Av1=Av;
disp('abs(A)&gt;&gt;1+(R2/R1)=1+abs(Av1)');
A=1+abs(Av1);
disp('Because the cross over point occurs at a significantly higher frequency than the dominant frequency at 10Hz ');
disp('The phase angle of A is very close to -90 degrees=%pi/2 radians');
disp('Using the j-notation the gain A at the crossover point is equal to %i*11');
A=%i*11;
disp('1+abs(Av1)=(R1+R2)/R1');
Av=(-10)*(A/(1+abs(Av1)))/(1+(A/(1+abs(Av1))));
Av2=abs(Av);
M=Av/Av2;
printf(" The closed-loop gain at the cross over point is thus reduced by a factor %f and therefore point corresponds to upper half-power frequency which is 90KHz",M)
disp('Therefore fh=90KHz, fl=0KHz');
fh=90;//in KHz
fl=0;//in KHz
disp('Bandwidth BW=fh-fl');
```

```
BW=fh-fl;
printf(" RESULTS:\n");
printf(" The closed-loop bandwidth is %d KHz",BW);
```

Scilab code Exa 7.5 Full power bandwith frequency

```

SR=0.5*(10^6);
//given operating power supplies +-12V
printf("\nGiven operating power supplies +-12V\n");
printf("For 741 op-amp,the output can work typically
    to within 1V of the power supplies.\nSo,Vom=11V\
    n");
Vom=11;//in volts
printf("f=(1/(2*pi))*(SR/Vom)\n");
f=(1/(2*%pi))*(SR/Vom);
printf("RESULTS:\n");
printf("Full-power bandwidth frequency f=%d Hz",f);
```

Chapter 8

SELECTED OP AMP APPLICATIONS

Scilab code Exa 8.1 basic differential amplifier

```

Adm=100;
printf("Gain of an difference amplifier Adm=(1+(R2/
    R1))/(1+(R3/R4))\n");
printf("Let (R2/R1)=A and (R3/R4)=B\n");
printf("Therefore Adm=(1+A)/(1+B)\n");
printf("A=Adm,");
printf("gives B=1/Adm\n");
A=Adm;
B=1/Adm;
printf("By suitable selection of resistors say R1=R2
    =1 Kohm");
R1=1;//in Kohm
R3=1;
R2=Adm*R1;
R4=Adm*R3;
printf("\nRESULTS:\n");
printf("Reasonable set of resistors R1=%d Kohm,R2=%d
     Kohms,R3=%d Kohm,R4=%d Kohms.",R1,R2,R3,R4);
```

Scilab code Exa 8.2 Instrumental amplifier

```

Adm=100;
printf("Adm=100=10*10,therefore it requires(R2/R1)
    =10 Kohms and (R4/R3)=10 Kohms\n")
printf("Therefore if R1=R3=10 Kohms, then R2=R4=100
    kohms\n");
R1=10;//in Kohms
R2=100;
R3=10;
R4=100;
printf("Adm=(1+2(Rb/Ra))*R4/R3\n");
printf("(Rb/Ra)=B\n");
printf("Adm=(1+2*B)*R4/R3\n");
B=((Adm/(R4/R3))-1)/2;
printf("Rb/Ra=%1.1f\n",B);
printf("If Ra=10 Kohms\n");
Ra=10;//in kohms
Rb=B*Ra;
printf("\nRESULTS:\n");
printf("The value of resistors are R1=%d Kohms,R2=%d
     Kohms,R3=%d Kohms,R4=%d Kohms,Ra=%d Kohms,Rb=%d
    Kohms",R1,R2,R3,R4,Ra,Rb);
```

Scilab code Exa 8.3 wein bridge oscillator

```
//caption:wein bridge oscillator
//To design a wein bridge oscillator of frequency fo
    =10KHz
fo=10000;//in Hz
```

```
printf("Capacitance C,Resistance R and Frequency fo are related as C*R=1/(2*pi*fo)\n");
printf("If C*R=A\n");
A=1/(2*%pi*fo);
C=0.01*(10^-6);//in farads
R=A/C;//in ohms
B=R/1000;//to convert to Kohms
printf("Taking C=0.01microfarad , we get R=%1.2f Kohms\n",B);//in Kohms
printf("Gain of a non-inverting amplifier should be 3 i.e. Av=1+(R1/R2)=3\n");
printf("This gives (R1/R2)=2, by selecting R2=10 Kohms, we get R1=20 Kohms\n");
R2=10;
R1=20;//in Kohms
printf("The gain must be restricted between 2.8 to 3.2, by selecting proper resistors Ra and Rb such that R1=Ra||Rb this can be achieved.\n");
printf("These values of Ra and Rb comes out to be Ra=22Kohms and Rb=220Kohms\n");
printf("RESULTS:\n");
printf("design values are R=%1.2f Kohms, C=0.01 microFarad,Ra=22 Kohms,Rb=220 Kohms\n",B);
```

Chapter 9

FURTHER OP AMP APPLICATIONS

Scilab code Exa 9.1 Design of phase shifter

```
//caption:Design of phase shifter
//example9.1
disp("Given frequency fo=10 KHz,Vrms=5 V,phi=10
   degrees\n");
disp("Taking A=C3*R4\n");
phi=10;//in degrees
fo=1000;//in Hz
disp("phi=180-2(atan(2*%pi*f*A))");
A=tan((180-10)*%pi/(180*2))/(2*fo*%pi);
printf("Therefore A=C3*R4=%f sec.\n",A);
R4=10000;//let (in ohms)
printf("C3 and R4 values are selected such that
   their product equals or greater than %f, The
   above values are preferable for low cost and bias
    compensation",A);
C3=A/R4;
printf("\nC3=%f uF",C3*10^6);
disp("To lower the cost of design,the preferred
   value is C31=0.22uF");
```

```
C31=0.22;//let Such that C31>C3
disp("since,C31*R4>A,C31 can be preferred")
printf("Similarly, R1 and R2 values should be of Good matching to obtain accurate unity gain modulus ");
printf("RESULTS:\n");
printf("(i)Resistors, R1=R2=10Kohms\n");
printf("(iii)R4=%d Kohms\n",R4/1000);//divided by 1000 to display in Kohms
printf("(iii)Capacitor, C3=%1.2f uF\n",C31);
```

Scilab code Exa 9.2 design of non inverting amplifier

```
//caption:design of non inverting amplifier
//to design a non inverting amplifier
//mid-band gain Av
Av=100;
//input impedance Zin
Zin=1000000;//in ohms
//cut-off frequencies fl1 and fl2
fl1=10;//in Hz
fl2=10;//in Hz
//From problem it follows
R3=Zin;
disp('fl2=1/(2*%pi*C2*R3)');
disp('C2=1/(2*%pi*fl2*R3)');
C2=1/(2*%pi*fl2*R3);
disp('Av=1+(R2/R1)');
disp('Bias compensation can also be obtained by taking R2=R3=1MHz');
R2=R3;
disp('R1=1/(Av-1)*R2');
R1=1/(Av-1)*R2;
printf('R1=%1.1f Kohms, preferred and standard value is 10Kohms\n',R1/1000);
```

```
R=R1;
R1=10000;//in ohms
disp('fl1=1/(2*%pi*C1*R3)');
disp('C1=1/(2*%pi*fl1*R3)');
C1=1/(2*%pi*fl1*R1);
printf('RESULTS: Design summary\n');
printf('R1=%1.1f Kohms, preferred and standard value is 10Kohms\n',R/1000);
printf('R2=R3=%d Mohm\n',R3/10**6);//divided by 10^6 to convert to Mohms
printf('C1=%1.2f uF but standard and preferred value is 2.2 uF\n',C1*(10**6));
printf('C2=%2.1f nF but standard and preferred value is 22 uF\n',C2*(10**9));
```

Scilab code Exa 9.3 low pass second order filter

```
//caption:low pass second order filter
//to design a low pass second order filter
//since it is butterworth filter
Q=1/sqrt(2);
Wo=100;//in rad/sec.
H=-2;
bo=H;
a1=1/(Q*Wo);
a2=1/(Wo**2);
printf("It is convenient to choose R=100Kohms\n");
R=100;//in Kohms
R3=R;
R4=R;
R1=R4/(-bo);
a3=R3+R4+((R3*R4)/R1);
C5=(a1/a3)*(10**6);//in uF
C2=a2/(R3*R4*C5)*(10**9);//in nF
printf("RESULTS:\n");
```

```
printf("R=%dKohms,\nR1=%dKohms,\nR3=%dKohms,\nR4=
    %dKohms,\nC2=%1.3f  nF,\nC5=%2.1fuF",R,R1,R3,R4,C2
    ,C5);
```