Scilab Textbook Companion for Op-Amps And Linear Integrated Circuits by R. A. Gayakwad[1]

Created by
Amar Ds
BE
Instrumentation Engineering
Sri Jayachamarajendra College Of Engineering
College Teacher
Rathnnakara S
Cross-Checked by
TechPassion

August 10, 2013

[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: Op-Amps And Linear Integrated Circuits

Author: R. A. Gayakwad

Publisher: PHI Learning Pvt. Ltd., New Delhi

Edition: 4

Year: 2004

ISBN: 978-81-203-2058-1

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 to Operational Amplifier

Scilab code Exa 1.1.a Collector current and dc voltage

```
//Chapter1
//Page.No-4,
//Example_1_1_a, Figure.No-1.2
//Collector current and dc voltage
//Given:
clear;clc;
Vcc=6;Vbe5=0.7;Vee=6;Vbe3=6.7;Vbe6=0.7;Vbe7=0.7; //
   Voltage in volts
Rc1=6.7*10^3; // Resistance in ohms
Ic1=rand();
Vc1=Vcc-Rc1*Ic1;
Ve4=Vc1-Vbe5;
I4=(Ve4+Vee)/(9.1*10^3+5.5*10^3);
Vb3=5.5*10^3*I4-Vee;
Ve3=Vb3-Vbe3;
Ie3=(Ve3+Vbe3)/3.3*10^3;
Ic1=1.08*10^-3/2.765; // Since Ie3=2*Ic1,
   substituting in above equation and simplifying
printf("\n Collector current Ic1 is = %.5f A \n",Ic1
```

```
) // Result
Vc1=Vcc-Rc1*Ic1;
printf("\n Voltage Vc1 is = %.2f V \n",Vc1) // Result
Ve4=Vc1-Vbe5;
printf("\n Voltage Ve4 is = %.2f V \n",Ve4) // Result
Ie4=(Ve4+Vee)/(29.2*10^3);
printf("\n Current Ie4 is = %.6f A \n",Ie4) // Result
Ic5=Ie4;
printf("\n Current Ic5 is = %.6f A \n",Ic5) // Result
Vc5=Vcc-3.8*10^3*Ic5;
printf("\n Voltage Vc5 is = %.2f V \n",Vc5) // Result
Ve6=Vc5-Vbe6;
printf("\n Voltage Ve6 is = %.2f V \n",Ve6) // Result
Ie6=(Ve6+Vee)/(15*10^3);
printf("\n Current Ie6 is = %.6f A \n",Ie6) // Result
Ve7=Ve6+Vbe7;
printf("\n Voltage Ve7 is = %.2f V \n",Ve7) // Result
I1=(Vcc-Ve7)/400;
printf("\n Current I1 is = %.6f A \n",I1) // Result
Ie8=I1;
printf("\n Current Ie8 is = %.6f A \n",Ie8) // Result
Ve8=-Vee+2*10^3*Ie8;
printf("\n Voltage Ve8 at the output terminal is = %.2f V \n",Ve8) // Result
```

Scilab code Exa 1.1.b Voltage gain of the opamp

```
//Chapter1
//Page.No-4,
//Example_1_1_b, Figure.No-1.2
//Voltage gain of the opamp
//Given:
clear;clc;
Ie1=0.39*10^-3;Ie4=0.298*10^-3;Ie6=0.678*10^-3; // Current in amps
Rc1=6.7*10^3;Rc5=3.8*10^3; // Resistance in ohms
beta_ac=150;
re1=(25*10^-3)/Ie1;
re2=re1;
re4=(25*10^-3)/Ie4;
re5=re4;
re6=(25*10^-3)/Ie6;
k=(Rc1*2*beta_ac*re4)/(Rc1+2*beta_ac*re4);
Ad1=k/re1;
printf("\n Voltage gain of the dual-input,balanced output-differential amplifier is = %.2f \n",Ad1) // Result
k1=(Rc5*beta_ac*(re6+15*10^3))/(Rc5+beta_ac*(re6+15*10^3));
Ad2=k1/(2*re5);
printf("\n Voltage gain of the dual-input,unbalanced output-differential amplifier is = %.1f \n",Ad2) // Result
Ad=82.55*22.6; // Using Ad=Ad1*Ad2
printf("\n Overall gain of the op-amp is = %.2f \n",Ad) // Result
```

Scilab code Exa 1.1.c Input resistance of the opamp

```
//Chapter1
//Page.No-4,
//Example_1_1_c, Figure.No-1.2
//Input resistance of the opamp
//Given:
clear;clc;
beta_ac=150;
re1=64.1; // Resistance in ohms
Ri=2*beta_ac*re1;
printf("\n Input resistance Ri is = %.1f ohm \n",Ri)
    // Result
```

Chapter 2

Interpretation of Data Sheets and Characteristics of an Opamp

Scilab code Exa 2.1.a Output voltage for Openloop differential amplifier

```
//Chapter2
//Page.No-45, Figure.No-2.9
//Example_2_1_a
//Output voltage for open-loop differential
    amplifier
//Given:
clear;clc;
vin1=5*10^-6;vin2=-7*10^-6;  // Both input voltages
    are in volts
A=200000; // Voltage gain
vo=A*(vin1-vin2); //Output voltage in volts
printf("\n Output voltage is vo = %.1f V dc \n",vo)
    // Result
```

Scilab code Exa 2.1.b Output voltage for openloop differential amplifier

```
//Chapter2
//Page.No-45, Figure.No-2.9
//EXAMPLE_2_1_b
//Output voltage for open-loop differential
    amplifier
//Given:
clear;clc;
vin1=10*10^-3;vin2=20*10^-3;  // Both input voltages
    are in volts
A=200000; // Voltage gain
vo=A*(vin1-vin2); // Output voltage in volts
printf("\n Output voltage is vo = %.f V rms \n",vo)
    // Result
```

Scilab code Exa 2.2.a Output voltage for inverting amplifier

```
//Chapter2
//Page.No-46, Figure.No-2.10
//Example_2_2_a
//Output voltage for inverting amplifier
//Given:
clear;clc;
vin=20*10^-3;  // Input voltage in volts
A=200000; // Voltage gain
vo=-(A*vin); // Output voltage in volts
```

```
printf("\n Output voltage is vo = %.f V \n",vo) // Result
```

Scilab code Exa 2.2.b Output voltage for inverting amplifier

```
//Chapter2
//Page.No-46, Figure.No-2.10
//Example_2_2_b
//Output voltage for inverting amplifier
//Given:
clear;clc;
vin=-50*10^-6;  // Input voltage in volts
A=200000; // Voltage gain
vo=-(A*vin); // Output voltage in volts
printf("\n Output voltage is vo = %.f V \n",vo) // Result
```

Chapter 3

An Opamp with Negative Feedback

Scilab code Exa 3.1 Parameters Of voltageseries feedback amplifier

```
// Chapter3
// Page.No-75, Figure.No-3.2
// Example_3_1
// Parameters of voltage-series feedback amplifier
// Given
clear;clc;
R1=1000;Rf=10000;
A=200000; // Open-loop voltage gain
Ri=2*10^6; // Input resistance without feedback
Ro=75; // Output resistance without feedback
fo=5; // Break frequency of an Op-amp
Vsat=13; // Saturation voltage
B=R1/(R1+Rf); // Gain of the feedback circuit
Af=A/(1+A*B);  // Closed-loop voltage gain
printf("\n Closed-loop voltage gain is Af = %.2f \n",Af) // Result
RiF=Ri*(1+A*B); // Input resistance with feedback
printf("\n Input resistance with feedback is RiF = %.2f ohms \n",RiF) // Result
```

```
RoF=Ro/(1+A*B); // Output resistance with feedback
printf("\n Output resistance with feedback is RoF = %f ohms \n",RoF) // Result
fF=fo*(1+A*B); // Bandwidth with feedback
printf("\n Bandwidth with feedback is vo = %.1f Hz \n",fF) // Result
VooT=Vsat/(1+A*B); // Total output offset voltage with feedback
printf("\n Total output offset voltage with feedback is VooT = %f V \n",VooT) // Result
```

Scilab code Exa 3.2 Parameters Of voltageseries feedback amplifier

```
// Chapter3
// Page.No-83, Figure.No-3.7
// Example_3_2
// Parameters of voltage-series feedback amplifier
// Given
clear;clc;
R1=1000;Rf=10000;
A=200000; // Open-loop voltage gain
Ri=2*10^6; // Input resistance without feedback
Ro=75; // Output resistance without feedback
fo=5; // Break frequency of an Op-amp
Vsat=13; // Saturation voltage
B=1; // Gain of the feedback circuit and it is equal to 1 for voltage follower
Af=A/(1+A*B); // Closed-loop voltage gain
printf("\n Closed-loop voltage gain is Af = %.f \n",Af) // Result
RiF=Ri*(1+A*B); // Input resistance with feedback
printf("\n Input resistance with feedback is RiF = %.1f ohms \n",RiF) // Result
```

```
RoF=Ro/(1+A*B); // Output resistance with feedback
printf("\n Output resistance with feedback is RoF =
    %f ohms \n",RoF) // Result
fF=fo*(1+A*B); // Bandwidth with feedback
printf("\n Bandwidth with feedback is vo = %.1f Hz \
    n",fF) // Result
VooT=Vsat/(1+A*B); // Total output offset voltage
    with feedback
printf("\n Total output offset voltage with feedback
     is VooT = %f V \n",VooT) // Result
```

Scilab code Exa 3.3 Parameters of Voltageshunt feedback amplifier

```
// Chapter3
// Page.No-86, Figure.No-3.8
// Example_3_3
// Parameters of voltage-shunt feedback amplifier
// Given
clear;clc;
R1=470;Rf=4.7*10^3;
A=200000; // Open-loop voltage gain
Ri=2*10^6;// Input resistance without feedback
Ro=75; // Output resistance without feedback
fo=5; // Break frequency of an Op-amp
Vsat=13; // Saturation voltage
K=Rf/(R1+Rf); // Voltage attenuation factor
B=R1/(R1+Rf); // Gain of the feedback circuit
Af=-(A*K)/(1+A*B);  // Closed-loop voltage gain
printf("\n Closed-loop voltage gain is Af = %.f \n",
    Af) // Result
X=Rf/(1+A);
RiF=R1+(X*Ri)/(X+Ri); // Input resistance with
    feedback
```

```
printf("\n Input resistance with feedback is RiF = %.1f ohms \n",RiF) // Result
RoF=Ro/(1+A*B); // Output resistance with feedback
printf("\n Output resistance with feedback is RoF = %f ohms \n",RoF) // Result
fF=fo*(1+A*B)/K; // Bandwidth with feedback
printf("\n Bandwidth with feedback is vo = %.2f Hz \n",fF) // Result
VooT=Vsat/(1+A*B); // Total output offset voltage with feedback
printf("\n Total output offset voltage with feedback is VooT = %f V \n",VooT) // Result
```

Scilab code Exa 3.4 Output voltage of voltage shunt feedback amplifier

```
// Chapter3
// Page.No-86, Figure.No-3.8
// Example_3_4
// Output voltage of voltage-shunt feedback amplifier
// Given
clear;clc;
R1=470;Rf=4.7*10^3;
A=200000; // Open-loop voltage gain
vin=1; // Input voltage in volts
K=Rf/(R1+Rf); // Voltage attenuation factor
B=R1/(R1+Rf); // Gain of the feedback circuit
Af=-(A*K)/(1+A*B);  // Closed-loop voltage gain
vo=Af*vin; // Output voltage
printf("\n Output voltage is vo = %.f V  \n",vo) // Result
t=0:0.1:2*%pi;
vo=-10*sin(t);
```

```
plot(t,vo);
title('Output Voltage');
xlabel('t');
ylabel('Vo');
```

Scilab code Exa 3.5.a Gain input resistance of the amplifier

```
// Chapter3
// Page.No-96, Figure.No-3.14
// Example_3_5_a
// Gain Input resistance of the amplifier
// Given
clear;clc;
R1=1000;R2=1000;
Rf=10*10^3;R3=10*10^3;
AD=-Rf/R1; // Voltage gain
printf("\n Voltage gain is AD = %.f \n",AD) //
   Result
RiFx=R1; // Input resistance of inverting amplifier
printf("\n Input resistance of inverting amplifier
   is RiFx = %.f ohms \n",RiFx) // Result
RiFy=R2+R3; // Input resistance of non-inverting
   amplifier
printf("\n Input resistance of non-inverting
   amplifier is RiFy = %.f ohms \n",RiFy) // Result
```

Scilab code Exa 3.5.b Output voltage of an Opamp

```
// Chapter3
```

```
// Page.No-96, Figure.No-3.14
// Example_3_5_b
// Output voltage of an Op-amp
// Given
clear;clc;
vx=2.7;vy=3; //Both input voltages are in volts
Rf=10*10^3;R1=1000; // Both are in ohms
AD=-Rf/R1; // Voltage gain
vxy=vx-vy;
vo=AD*vxy; // Output voltage
printf("\n Output voltage is vo = %.f V \n",vo) // Result
```

Scilab code Exa 3.6.a Voltage gain and input resistance of the Opamp

```
// Chapter3
// Page.No-99, Figure.No-3.16
// Example_3_6_a
// Voltage gain and input resistance of Op-amp
// Given
clear;clc;
R1=680;R3=680; // Both are in ohms
RF=6800;R2=6800; // Both are in ohms
Ri=2*10^6; // Open-loop input resistance of the op-amp
vx=-1.5;vy=-2; // Both are in volts
A=200000; // Open-loop Gain
AD=1+RF/R1; // Voltage gain
printf("\n Voltage gain is AD = %.f \n",AD) // Result
B=R2/(R2+R3);
RiFy=Ri*(1+A*B); // Input resistance of first stage amplifier
```

```
printf("\n Input resistance of first stage amplifier is RiFy = %.1f ohms \n",RiFy) // Result
B=R1/(R1+RF);
RiFx=Ri*(1+A*B); // Input resistance of second stage amplifier
printf("\n Input resistance of second stage amplifier is RiFx = %.1f ohms \n",RiFx) // Result
```

Scilab code Exa 3.6.b Output voltage of the Opamp

```
// Chapter3
// Page.No-99, Figure.No-3.16
// Example_3_6_b
// Output voltage of the Op-amp
// Given
clear;clc;
R1=680;RF=6800 // Both are in ohms
vx=-1.5;vy=-2; // Both input voltages are in volts
AD=1+RF/R1; // Voltage gain
vxy=vx-vy;
vo=AD*vxy; // Output voltage
printf("\n Output voltage is vo = %.1f V \n",vo) // Result
```

Chapter 4

The Practical Opamp

Scilab code Exa 4.1 Design of Compensating Network

```
// Chapter4
// Page.No-114
// Example_4_1
// Design of Compensating Network
// Given
clear;clc;
V=10 // Supply voltage
Vio=10*10^-3; // Input offset voltage
Rc=10; // Assumption
Rb=(V/Vio)*Rc;
printf("\n Resistance Rb is = %.f ohms \n",Rb) //
    Result
Ra=Rb/2.5; // Since Rb>Rmax,let us choose Rb=10*Rmax
     where Rmax=Ra/4
printf("\n Resistance Ra is = %.f ohms \n",Ra) //
    Result
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 4.2 Max Output offset voltage

```
// Chapter4
// Page.No-121, Figure.No-4.13
// Example_4_2
// Max Output offset voltage
// Given
clear;clc;
R1=1*10^3;Rf=10*10^3;
Vio=10*10^-3; // Input offset voltage
Aoo=1+Rf/R1; // To find max value of Voo,we reduce
    input voltage vin to zero.
Voo=Aoo*Vio; // Max output offset voltage
printf("\n Max output offset voltage is = %.3f V dc
    \n",Voo) // Result
```

Scilab code Exa 4.3 Design of input offset voltage compensating network

```
// Chapter4
// Page.No-121, Figure.No-4.14
// Example_4_3
// Design of input offset voltage-compensating
    network
// Given
clear;clc;
```

```
R1=1*10^3;Rf=10*10^3;Rc=10;
Af=1+Rf/(R1+Rc); // Closed loop gain of non-
    inverting amplifier
printf("\n Closed loop gain of non-inverting
    amplifier is = %.1f \n",Af) // Result
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 4.4.a Max Output offset voltage

```
// Chapter4
// Page.No-127, Figure.No-4.19
// Example_4_4_a
// Max Output offset voltage
// Given
clear;clc;
R1=470;Rf=47*10^3;
Vio=6*10^-3;
Ib=500*10^-9;
Vs=15;
// Max output offset voltage due to input offset
    voltage,Vio is :
Voo=(1+Rf/R1)*Vio; // Max output offset voltage
printf("\n Max output offset voltage is = %.3f V dc
    \n",Voo) // Result
// Max output offset voltage due to input offset
    voltage,Ib is :
VoIb=Rf*Ib; // Max output offset voltage
printf("\n Max output offset voltage due to input
    offset current,Ib is = %.6f V dc \n",VoIb) //
    Result
```

Scilab code Exa 4.4.b Effect of input bias current

```
// Chapter4
// Page.No-127, Figure.No-4.19
// Example_4_4_b
// Effect of input bias current
// Given
clear;clc;
R1=470;Rf=47*10^3;
ROM=R1*Rf/(R1+Rf); // Parallel combination of R1 and
    Rf
printf("\n Parallel combination of R1 and Rf,i.e ROM
    is = %.1f ohm \n",ROM) // Approximately the
    value is 47 ohm
```

Scilab code Exa 4.5.a Max Output offset voltage

```
// Chapter4
// Page.No-127, Figure.No-4.19
// Example_4_5_a
// Max Output offset voltage
// Given
clear;clc;
R1=1*10^3;Rf=100*10^3;
Vio=6*10^-3;
Ib=500*10^-9;
Vs=15;
```

```
// Max output offset voltage due to input offset voltage,Vio is :
Voo=(1+Rf/R1)*Vio; // Max output offset voltage
printf("\n Max output offset voltage due to input offset voltage,Vio is = %.4f V dc \n",Voo) // Result
// Max output offset voltage due to input offset voltage,Ib is :
VoIb=Rf*Ib; // Max output offset voltage
printf("\n Max output offset voltage due to input offset current,Ib is = %.4f V dc \n",VoIb) // Result
```

Scilab code Exa 4.5.b Effect of input bias current

```
// Chapter4
// Page.No-127, Figure.No-4.19
// Example_4_4_b
// Effect of input bias current
// Given
clear;clc;
R1=1*10^3;Rf=100*10^3;
ROM=R1*Rf/(R1+Rf); // Parallel combination of R1 and Rf
printf("\n Parallel combination of R1 and Rf,i.e ROM is = %.f ohm \n",ROM)
```

Scilab code Exa 4.6 Max Output offset voltage

```
// Chapter4
// Page.No-130, Figure.No-4.21
// Example_4_6
// Max Output offset voltage
// Given
clear;clc;
Iio=200*10^-9; // Input offset current
Rf=100*10^3;
VoIio=Rf*Iio; // Max output offset voltage
printf("\n Max output offset voltage due to input
    offset current,Ib is = %.4f V dc \n",VoIio) //
    Result
```

Scilab code Exa 4.7 Total Output offset voltage

```
// Chapter4
// Page.No-132, Figure.No-4.22(a)
// Example_4_7
// Total Output offset voltage
// Given
clear;clc;
R1=1*10^3;Rf=10*10^3;
Vio=7.5*10^-3; // Max input offset voltage
Iio=50*10^-9; // Max input offset current
Ib=250*10^-9; // Max input bias current
// For figure 4.22(a)
VooT=(1+Rf/R1)*Vio+(Rf*Ib); // Since the current
    generated output offset voltage is due to input
    bias current Ib
printf("\n Max total output offset voltage due to
    input offset current,Ib is = %.4f V  \n",VooT) //
     Result

```

```
// For figure 4.22(b)
VooT=(1+Rf/R1)*Vio+(Rf*Iio); // Since the current generated output offset voltage is due to input offset current Ib
printf("\n Max total output offset voltage due to input offset current,Ib is = %.4f V  \n",VooT) // Result
```

Scilab code Exa 4.8.a Error voltage and output voltage

```
// Chapter4
// Page.No-136, Figure.No-4.24
// Example_4_8_a
// Error voltage and output voltage
// Given
clear;clc;
delta_Vio=(30*10^-6); // Change in input offset voltage
delta_T=1; // Unit change in temperature
delta_Iio=(300*10^-12); // Change in input offset current
Vs=15;
R1=1*10^3;Rf=100*10^3;Rl=10*10^3;
Vin=1*10^-3; // Input voltage
k=25; // Amplifier is nulled at 25 deg
T=35-k; // Change in temperature
Ev=(1+Rf/R1)*(delta_Vio/delta_T)*T + Rf*(delta_Iio/delta_T)*T; // Error voltage
printf("\n Error voltage is = %.4f V \n",Ev) // Result
Vo=-(Rf/R1)*Vin+Ev; // Output voltage
printf("\n Output voltage is = %.4f V \n",Vo) // Result
```

```
// (OR)
Vo=-(Rf/R1)*Vin-Ev; // Output voltage
printf("\n Output voltage is = %.4f V \n",Vo) // Result
```

Scilab code Exa 4.8.b Error voltage and output voltage

```
// Chapter4
// Page.No-136, Figure.No-4.24
// Example_4_8_b
// Error voltage and output voltage
// Given
clear;clc;
delta_Vio=(30*10^-6); // Change in input offset voltage
delta_T=1; // Unit change in temperature
delta_Iio=(300*10^-12); // Change in input offset current
Vs=15;
R1=1*10^3;Rf=100*10^3;Rl=10*10^3;
Vin=10*10^-3; // Input voltage
k=25; // Amplifier is nulled at 25 deg
T=35-k; // Change in temperature
Ev=(1+Rf/R1)*(delta_Vio/delta_T)*T + Rf*(delta_Iio/delta_T)*T; // Error voltage
printf("\n Error voltage is = %.4f V \n",Ev) // Result
Vo=-(Rf/R1)*Vin+Ev; // Output voltage
printf("\n Output voltage is = %.4f V \n",Vo) // Result
// (OR)
Vo=-(Rf/R1)*Vin-Ev; // Output voltage
printf("\n Output voltage is = %.4f V \n",Vo) //
```

```
   Result
```

Scilab code Exa 4.9.a Error voltage and output voltage

```
// Chapter4
// Page.No-136, Figure.No-4.24
// Example_4_9_a
// Error voltage and output voltage
// Given
clear;clc;
delta_Vio=(30*10^-6); // Change in input offset
   voltage
delta_T=1; // Unit change in temperature
delta_Iio=(300*10^-12); // Change in input offset
   current
Vs=15;
R1=1*10^3;Rf=100*10^3;Rl=10*10^3;
Vin=10*10^-3; // Input voltage
k=25; // Amplifier is nulled at 25 deg
T=55-k; // Change in temperature
Ev=(1+Rf/R1)*(delta_Vio/delta_T)*T + Rf*(delta_Iio/
   delta_T)*T; // Error voltage
printf("\n Error voltage is = %.4f V dc \n",Ev) //
   Result
Vo=-(Rf/R1)*Vin+Ev; // Output voltage
printf("\n Output voltage is = %.4f V \n",Vo) //
   Result
// (OR)
Vo=-(Rf/R1)*Vin-Ev; // Output voltage
printf("\n Output voltage is = %.4f V \n",Vo) //
   Result
```

Scilab code Exa 4.9.b Output waveform

```
// Chapter4
// Page.No-136, Figure.No-4.24
// Example_4_9_b
// Output waveform
// Given
clear;clc;
t =0:0.1:2*%pi;
y = -1000*sin(t)+91.8;
a = gca();
a.x_label.text = 'Time';
a.y_label.text = 'Voltage';
a.title.text = 'Output waveform';
plot2d(t,y);
t1=0:0.1:2*%pi;
y1=91.8*(t1>=0);
b=gca();
b.line_style=3;
plot2d(t1,y1);
```

Scilab code Exa 4.10.a Error voltage and output voltage

```
// Chapter4
// Page.No-141, Figure.No-4.26
// Example_4_10_a
// Error voltage and output voltage
// Given
clear;clc;
```

```
delta_Vio=(30*10^-6); // Change in input offset voltage
delta_T=1; // Unit change in temperature
delta_Iio=(300*10^-12); // Change in input offset current
Vs=15;
R1=1*10^3;Rf=100*10^3;
Vin=1*10^-3; // Input voltage
k=25; // Amplifier is nulled at 25 deg
T=35-k; // Change in temperature
Ev=(1+Rf/R1)*(delta_Vio/delta_T)*T + Rf*(delta_Iio/delta_T)*T; // Error voltage
printf("\n Error voltage is = %.4f V dc \n",Ev) // Result
Vo=(1+Rf/R1)*Vin+Ev; // Output voltage
printf("\n Output voltage is = %.4f V \n",Vo) // Result
// (OR)
Vo=(1+Rf/R1)*Vin-Ev; // Output voltage
printf("\n Output voltage is = %.4f V \n",Vo) // Result
```

Scilab code Exa 4.10.b Error voltage and output voltage

```
// Chapter4
// Page.No-141, Figure.No-4.26
// Example_4_10_b
// Error voltage and output voltage
// Given
clear;clc;
delta_Vio=(30*10^-6); // Change in input offset voltage
delta_T=1; // Unit change in temperature
```

```
delta_Iio=(300*10^-12); // Change in input offset current
Vs=15;
R1=1*10^3;Rf=100*10^3;Rl=10*10^3;
Vin=10*10^-3; // Input voltage
k=25; // Amplifier is nulled at 25 deg
T=35-k; // Change in temperature
Ev=(1+Rf/R1)*(delta_Vio/delta_T)*T + Rf*(delta_Iio/delta_T)*T; // Error voltage
printf("\n Error voltage is = %.4f V dc \n",Ev) // Result
Vo=(1+Rf/R1)*Vin+Ev; // Output voltage
printf("\n Output voltage is = %.4f V dc \n",Vo) // Result
// (OR)
Vo=(1+Rf/R1)*Vin-Ev; // Output voltage
printf("\n Output voltage is = %.4f V dc \n",Vo) // Result
```

Scilab code Exa 4.11.a Output offset voltage

```
// Chapter4
// Page.No-141, Figure.No-4.28
// Example_4_11_a
// Output offset voltage
// Given
clear;clc;
delta_Vio=15.85*10^-6; // Change in input offset voltage
delta_V=1; // Unit change in supply voltage
V=2; // Change in supply voltage
R1=1*10^3;Rf=100*10^3;
delta_Voo=(1+Rf/R1)*(delta_Vio/delta_V)*V; // Change
```

```
in output offset voltage
printf("\n Change in output offset voltage is = %.4f V \n",delta_Voo) // Result
```

Scilab code Exa 4.11.b Output offset voltage

```
// Chapter4
// Page.No-141, Figure.No-4.28
// Example_4_11_b
// Output offset voltage
// Given
clear;clc;
delta_Vio=15.85*10^-6; // Change in input offset voltage
delta_V=1; // Unit change in supply voltage
V=2; // Change in supply voltage
Vin=10*10^-3;
R1=1*10^3;Rf=100*10^3;
delta_Voo=(1+Rf/R1)*(delta_Vio/delta_V)*V; // Output offset voltage
Vo=(-Rf/R1)*Vin+delta_Voo; // Total output offset voltage
printf("\n Total output offset voltage is = %.4f V \n",Vo) // Result
// (OR)
Vo=(-Rf/R1)*Vin-delta_Voo; // Total output offset voltage
printf("\n Total output offset voltage is = %.4f V \n",Vo) // Result
```

Scilab code Exa 4.12 Output ripple voltage

```
// Chapter4
// Page.No-141, Figure.No-4.28(b)
// Example_4_12
// Output ripple voltage
// Given
clear;clc;
delta_Vio=15.85*10^-6; // Change in input offset voltage
delta_V=1; // Unit change in supply voltage
V=10*10^-3; // Change in supply voltage
R1=1*10^3;Rf=100*10^3;
delta_Voo=(1+Rf/R1)*(delta_Vio/delta_V)*V; // Change in output offset voltage
printf("\n Change in output offset voltage is = %.6f V \n",delta_Voo) // Result
```

Scilab code Exa 4.13 Change in output offset voltage

```
// Chapter4
// Page.No-136, Figure.No-4.24
// Example_4_13
// Change in output offset voltage
// Given
clear;clc;
delta_Vio=5*10^-6; // Change in input offset voltage
delta_t=1; // Unit change in time
```

```
delta_Iio=2*10^-9; // Change in input offset current
t=4; // Time elapsed(weeks)
R1=1*10^3;Rf=100*10^3;Rl=10*10^3;
delta_Voot=(1+Rf/R1)*(delta_Vio/delta_t)*t+Rf*(
    delta_Iio/delta_t)*t; // Change in output offset
    voltage
printf("\n Change in output offset voltage is = %.4f
    V \n",delta_Voot) // Result
```

Scilab code Exa 4.14.a Output voltage

```
// Chapter4
// Page.No-153, Figure.No-4.32
// Example_4_14_a
// Output voltage
// Given
clear;clc;
R1=1*10^3;R2=1*10^3;Rf=10*10^3;R3=10*10^3;
vd=5*10^-3; // Differential voltage
vcm=2*10^-3; // Common-mode voltage
Ad=Rf/R1; // Closed-loop differential gain
vo=Ad*vd; // Output voltage
printf("\n Output voltage is = %.3f V \n",vo) //
    Result
```

Scilab code Exa 4.14.b Output common mode voltage

```
// Chapter4
// Page.No-153, Figure.No-4.32
```

```
// Example_4_14_b
// Output common-mode voltage
// Given
clear;clc;
R1=1*10^3;R2=1*10^3;Rf=10*10^3;R3=10*10^3;
vd=5*10^-3; // Differential voltage
vcm=2*10^-3; // Common-mode voltage
Ad=Rf/R1; // Closed-loop differential gain
CMRRdb=90
CMRR=10^(90/20); // Using CMRRdb=20*log10(CMRR), to convert the CMRR(dB) value into its equuivalent numerical value
printf("\n CMRR is = %.2f \n",CMRR) // Result
vocm=(Ad*vcm)/CMRR; // Output common-mode voltage
printf("\n Output common-mode voltage is = %.8f V \n",vocm) // Result
```

Chapter 5

Frequency Response of an Opamp

Scilab code Exa 5.1 Maximum gain

```
 // Chapter5
// Page.No-171, Figure.No-5.4
// Example_5_1
// Maximum gain
// Given
clear;clc;
fo=5; // Break freq of the op-amp in Hz
s=%s;
A=200000; // Gain of the op-amp at 0 Hz
H=syslin('c',(A*fo*2*%pi)/((fo*2*%pi)+s));
fmin=1;
fmax=100000;
bode(H,fmin,fmax);
Aol=40;
printf("\n Maximum gain is = %.f dB \n ",Aol); //
   From the graph
```

Scilab code Exa 5.2 Gain equation and break frequencies

```
//Chapter5
//Page.No-172, Figure.No-5.5
//Example_5_2
//Gain equation and break frequencies
//Given:
clear;clc;
phase=-157.5; // Phase shift at about 3 MHz
f=3*10^6;
disp("Gain equation is Aol(f)=A/((1+(f/fo1)*j)*(1+(f
    /fo2)*j)),where fo1-first break frequency and fo2
    -second break frequency") // From the figure
fo1=6;
printf("\n First break frequency fo1 is = %.f Hz \n"
    ,fo1) // From the graph
k=-atand(f/fo1)-phase;
fo2=f/tand(k);
printf("\n Second break frequency fo2 is = %.1f Hz \
    n",fo2) // Result
```

Scilab code Exa 5.3 Stability of voltage follower

```
 // Chapter5
// Page.No-83, Figure.No-3.7
// Example_5_3
// Stability of voltage follower
// Given
```

```
clear;clc;
fo=5; // Break freq of the op-amp in Hz
s=%s;
A=200000; // Gain of the op-amp at 0 Hz
H=syslin('c',(A*fo*2*%pi)/((fo*2*%pi)+s));
fmin=10;
fmax=1000000;
bode(H,fmin,fmax);
Aol=0;
printf("\n Magnitude at which voltage follower is
    stable is = %.f dB \n ",Aol); // From the graph
```

Chapter 6

General Linear Applications

Scilab code Exa 6.1 Bandwidth of the amplifier

```
// Chapter6
// Page.No-193, Figure.No-6.3(a)
// Example_6_1
// Bandwidth of the amplifier
// Given
clear;clc;
R1=100;Rf=1*10^3;Rin=50;Rl=10*10^3;
Ci=0.1*10^-6; // Capacitance b/w 2 stages being
    coupled
RiF=R1; // ac input resistance of the second stage
Ro=Rin; // ac output resistance of the 1st stage
UGB=10^6; // Unity gain bandwidth
fl=1/(2*%pi*Ci*(RiF+Ro)); // Low-freq cutoff
printf("\n Low-freq cutoff is = %.1f Hz \n",fl) //
    Result
K=Rf/(R1+Rf);
Af=-Rf/R1; // closed loop voltage gain
fh=UGB*K/abs(Af); // High-freq cutoff
printf("\n High-freq cutoff is = %.1f Hz \n",fh) //
    Result
BW=fh-fl; // Bandwidth
```

```
printf("\n Bandwidth is = %.1f Hz \n",BW) // Result
```

Scilab code Exa 6.2.a Bandwidth of the amplifier

```
// Chapter6
// Page.No-193, Figure.No-6.4(c)
// Example_6_2_a
// Bandwidth of the amplifier
// Given
clear;clc;
R1=100*10^3;R2=100*10^3;R3=100*10^3;Rf=1*10^6;Rin
    =50;
Ci=0.1*10^-6; // Capacitance b/w 2 stages being
    coupled
Ro=Rin; // ac output resistance of the 1st stage
Vcc=15;
UGB=10^6; // Unity gain bandwidth
Rif=R2*R3/(R2+R3);  // since Ri*(1+A*B)>>R2 or R3
fl=1/(2*%pi*Ci*(Rif+Ro)); // low-freq cutoff
printf("\n Low-freq cutoff is = %.1f Hz \n",fl) //
    Result
K=Rf/(R1+Rf);
Af=-Rf/R1; // closed loop voltage gain
fh=UGB*K/abs(Af); // High-freq cutoff
printf("\n High-freq cutoff is = %.1f Hz \n",fh) //
    Result
BW=fh-fl; // Bandwidth
printf("\n Bandwidth is = %.1f Hz \n",BW) // Result
```

Scilab code Exa 6.2.b Max output voltage swing

```
// Chapter6
// Page.No-193, Figure.No-6.4(c)
// Example_6_2_b
// Max output voltage swing
// Given
clear;clc;
R1=100*10^3;R2=100*10^3;R3=100*10^3;Rf=1*10^6;Rin
   =50;
Ci=0.1*10^-6; // Capacitance b/w 2 stages being
   coupled
Ro=Rin; // ac output resistance of the 1st stage
UGB=10^6; // Unity gain bandwidth
Vcc=15;
printf("\n The ideal maximum output voltage swing is
    = %.f V pp \n",Vcc)
```

Scilab code Exa 6.3 Components of peak amplifier

```
// Chapter6
// Page.No-193, Figure.No-6.5(a)
// Example_6_3
// Components of peak amplifier
// Given
clear;clc;
fp=16*10^3; // Peak frequency
Af=10; // Gain at peak frequency
C=0.01*10^-6; // Assume
L=1/(((2*%pi*fp)^2)*10^-8); // Simplifying fp=1/(2*
   pi*sqrt(L*C))
printf("\n Inductance is = %.4f H \n",L)
L=10*10^-3; // Approximate
```

```
R=30; // Assume the value of internal resistance of the inductor
Xl=2*%pi*fp*L; // Inductive reactance
Qcoil=Xl/R; // Figure of merit of the coil
printf("\n Figure of merit of the coil is = %.1f \n",Qcoil)
Rp=(Qcoil)^2*R; // Parallel resistance of the tank circuit
printf("\n Parallel resistance of the tank circuit is = %.1f ohm \n",Rp)
R1=100; // Assume the value of internal resisrance of the coil
Rf=-Rp/(1-(Rp/(Af*R1))); // Simplifying Af=(Rf||Rp)/R1
printf("\n Feedback resistance is = %.1f ohm \n",Rf)
```

Scilab code Exa 6.4 Output voltage

```
// Chapter6
// Page.No-200, Figure.No-6.6
// Example_6_4
// Output voltage
// Given
clear;clc;
Va=1;Vb=2;Vc=3; // Input voltages in volts
Ra=3*10^3;Rb=3*10^3;Rc=3*10^3;Rf=1*10^3;
Vo=-((Rf/Ra)*Va+(Rf/Rb)*Vb+(Rf/Rc)*Vc); // Output voltage
printf("\n Output voltage is = %.f V \n",Vo)
```

Scilab code Exa 6.5 Output voltage

```
// Chapter6
// Page.No-203, Figure.No-6.7
// Example_6_5
// Output voltage
// Given
clear;clc;
Va=2;Vb=-3;Vc=4; // Input voltages in volts
R1=1*10^3;Rf=2*10^3;
V1=(Va+Vb+Vc)/3; // Voltage at non-inverting terminal
printf("\n Voltage at non-inverting terminal is = %.f V \n",V1)
Vo=(1+Rf/R1)*V1; // Output voltage
printf("\n Output voltage is = %.f V \n",Vo)
```

Scilab code Exa 6.6 Output voltage

```
// Chapter6
// Page.No-205, Figure.No-6.9
// Example_6_6
// Output voltage
// Given
clear;clc;
Va=2;Vb=3;Vc=4;Vd=5; // Input voltages in volts
R=1*10^3;
Vo=-Va-Vb+Vc+Vd; // Output voltage
```

```
printf("\n Output voltage is = %.f V \n",Vo)
```

Scilab code Exa 6.7 Output voltage

```
// Chapter6
// Page.No-209, Figure.No-6.12
// Example_6_7
// Output voltage
// Given
clear;clc;
R1=1*10^3;Rf=4.7*10^3;Ra=100*10^3;Rb=100*10^3;Rc
    =100*10^3;
Vdc=5;
Rt=100*10^3; // Resistance of a thermistor
temp_coeff=1*10^3;

// Output voltage at 0 degree
delta_R=-temp_coeff*(0-25); // Change in resistance
R=Ra; // Ra=Rb=Rc=R
Vo=((Rf*delta_R)/(R1*4*R))*Vdc;
printf("\n Output voltage at 0 degree is = %.2f V \n
    ",Vo)

// Output voltage at 100 degree
delta_R=-temp_coeff*(100-25); // Change in
    resistance
Vo=((Rf*delta_R)/(R1*4*R))*Vdc;
printf("\n Output voltage at 100 degree is = %.2f V
    \n",Vo)
```

Scilab code Exa 6.8 Change in resistance in straingage

```
// Chapter6
// Page.No-209, Figure.No-6.12
// Example_6_7
// Change in resistance in straingage
// Given
clear;clc;
A=-100; // Gain of the differential instrumentation
    amplifier
Ra=100;Rb=100;Rc=100;
Vdc=10;Vo=1;
R=Ra; // Ra=Rb=Rc=R
delta_R=(Vo*R)/(Vdc*abs(A)); // Change in resistance
printf("\n Change in resistance is = %.1f ohm \n",
    delta_R)
```

Scilab code Exa 6.9 Gain of the amplifier

```
// Chapter6
// Page.No-216, Figure.No-6.14(a)
// Example_6_9
// Gain of the amplifier
// Given
clear;clc;
Vo=3.7;Vin=100*10^-3;
R1=100; // Assume
Rf=0.5*((Vo*R1)/Vin-1); // Feedback resisrance
```

```
printf("\n Feedback resisrance is = %.1f ohm \n",Rf)
A=(1+2*Rf/R1); // Gain of the differential amplifier
printf("\n Gain of the differential amplifier is = %
    .1f \n",A)
```

Scilab code Exa 6.10 Range of input voltage

```
// Chapter6
// Page.No-220, Figure.No-6.17
// Example_6_10
// Range of input voltage
// Given
clear;clc;
R1min=1*10^3;R1max=6.8*10^3;
io=1*10^-3; // Meter current for full-wave
    rectification
vin_min=1.1*R1min*io; // Minimum input voltage
printf("\n Minimum input voltage is = %.1f V \n",
    vin_min)
vin_max=1.1*R1max*io; // Maximum input voltage
printf("\n Maximum input voltage is = %.2f V \n",
    vin_max)
```

Scilab code Exa 6.11 Current and voltage drop

```
// Chapter6
// Page.No-222, Figure.No-6.18
// Example_6_11
// Current and voltage drop
```

```
// Given
clear;clc;
Vin=0.5;Vo=1.2;
R1=100;
Io=Vin/R1; // Current through diode
printf("\n Current through diode is = %.4f A \n",Io)
Vd=Vo-Vin; // Voltage drop across diode
printf("\n Voltage drop across diode is = %.1f V \n"
   ,Vd)
```

Scilab code Exa 6.12.a Load current

```
// Chapter6
// Page.No-222, Figure.No-6.19
// Example_6_12_a
// Load current
// Given
clear;clc;
Vin=5;V1=1;
R=10*10^3;
Il=Vin/R; // Load current
printf("\n Load current is = %.5f A \n",Il)
```

Scilab code Exa 6.12.b Output voltage

```
// Chapter6
// Page.No-222, Figure.No-6.19
// Example_6_12_b
// Output voltage
```

```
// Given
clear;clc;
Vin=5;V1=1;
R=10*10^3;
Vo=2*V1; // Output voltage
printf("\n Output voltage is = %.f V \n",Vo)
```

Scilab code Exa 6.13 Range of output voltage

```
// Chapter6
// Page.No-222, Figure.No-6.20
// Example_6_13
// Range of output voltage
// Given
clear;clc;
R1=1*10^3;Rf=2.7*10^3;
Vref=2;
Io=0; // Since all the binary inputs D0 to D7 are logic zero
Vo_min=Io*Rf; // Minimum output voltage
printf("\n Minimum output voltage is = %.f V \n",Vo_min)
Io=(Vref/R1)*(1/2+1/4+1/8+1/16+1/32+1/64+1/128+1/256);
Vo_max=Io*Rf; // Maximum output voltage
printf("\n Maximum output voltage is = %.2f V \n",Vo_max)
```

Scilab code Exa 6.14 Change in output voltage

```
// Chapter6
// Page.No-228, Figure.No-6.21
// Example_6_14
// Change in output voltage
// Given
clear;clc;
Rf=3*10^3;
Vdc=5;
Rt=100*10^3; // Resistance at darkness
Vomin=-(Vdc/Rt)*Rf; // Min output voltage at darkness
printf("\n Min output voltage at darkness is = %.2f V \n",Vomin)
Rt=1.5*10^3; // Resistance at Illumination
Vomax=-(Vdc/Rt)*Rf; // Max output voltage at Illumination
printf("\n Max output voltage at Illumination is = %.f V \n",Vomax)
```

Scilab code Exa 6.15 Output voltage of an integrator

```
// Chapter6
// Page.No-230, Figure.No-6.23
// Example_6_15
// Output voltage of an integrator
// Given
clear;clc;
Vin=2; // Input voltage in volt
Vo0=0;
Vo1=-integrate('2','t',0,1);
disp(Vo1)
Vo2=-integrate('2','t',1,2)+Vo1;
disp(Vo2)
```

```
Vo3=-integrate('2','t',2,3)+Vo2;
disp(Vo3)
Vo4=-integrate('2','t',3,4)+Vo3;
disp(Vo4)
Vo=[Vo0 Vo1 Vo2 Vo3 Vo4];
t=[0 1 2 3 4];
plot(t,Vo);
title('Output Voltage');
xlabel('t');
ylabel('Vo');
```

Scilab code Exa 6.16.a Design of differentiator

```
// Chapter6
// Page.No−238
// Example_6_16_a
// Design of differentiator
// Given
clear;clc;
C1=0.1*10^-6; // Assume
fa=1*10^3; // Freq at which gain is 0 dB
Rf=1/(2*%pi*fa*C1); // Using fa=1/(2*%pi*Rf*C1)
printf("\n Feedback resistance is = %.1f ohm \n",Rf)
Rf=1.5*10^3; // Approximation
fb=20*10^3; // Gain limiting freq
R1=1/(2*%pi*fb*C1);
printf("\n Resistance,R1 is = %.1f ohm \n",R1)
R1=82; // Approximation
Cf=R1*C1/Rf;
printf("\n Capacitance,Cf is = %.10f farad \n",Cf)
Cf=0.005*10^-6; // Approximation
```

Scilab code Exa 6.16.b Output waveform of differentiator

```
// Chapter6
// Page.No-238
// Example_6_16_b
// Output waveform of differentiator
// Given
clear;clc;
step=0.01;
t=0:step:2*%pi;
dy=diff(sin(t))/step; //approximate differentiation
    of sine function
Vo=-1.5*10^3*0.1*10^-6*2*%pi*10^3*dy;
plot(Vo);
title('Output Voltage');
xlabel('t');
ylabel('Vo');
```

Chapter 7

Active Filters and Oscillators

Scilab code Exa 7.1 Design of low pass filter

```
// Chapter7
// Page.No-256
// Example_7_1
// Design of low pass filter
// Given
clear;clc;
fh=1*10^3; // Cut-off frequency
C=0.01*10^-6; // Assumption
R=1/(2*%pi*fh*C);
printf("\n Resistance R is = %.1f ohm \n",R) //
   Result
printf("\n Use 20 kohm POT as R \n")
R1=10*10^3; // Assumption
printf("\n Resistance R1 is = %.1f ohm \n",R1)
Rf=R1; // Sice passband gain is 2,R1 and Rf must be
   equal
printf("\n Resistance Rf is = %.1f ohm \n",Rf)
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 7.2 Design of low pass filter

```
// Chapter7
// Page.No-256
// Example_7_2
// Design of low pass filter
// Given
clear;clc;
fc0=1*10^3; // Original cut-off frequency
fc1=1.6*10^3; // New cut-off frequency
R=15.9*10^3; // Original resistance value
k=fc0/fc1;
Rnew=R*k;
printf("\n New Resistance Rnew is = %.1f ohm \n",
   Rnew) // Result
```

Scilab code Exa 7.3 Frequency response of low pass filter

```
// Chapter7
// Page.No-257
// Example_7_3
// Frequency response of low pass filter
// Given
clear;clc;
Af=2; // Passband gain of the filter
```

```
fh=1000; // Cut-off frequency
f1=10; // Input freq in Hz
av1=Af/sqrt(1+(f1/fh)^2);
printf("\n Gain magnitude av1 at f1 is = %.2f \n",av1) // Result
f2=100; // Freq in Hz
av2=Af/sqrt(1+(f2/fh)^2);
printf("\n Gain magnitude av2 at f2 is = %.2f \n",av2) // Result
f3=200; // Freq in Hz
av3=Af/sqrt(1+(f3/fh)^2);
printf("\n Gain magnitude av3 at f3 is = %.2f \n",av3) // Result
f4=700; // Freq in Hz
av4=Af/sqrt(1+(f4/fh)^2);
printf("\n Gain magnitude av4 at f4 is = %.2f \n",av4) // Result
f5=1000; // Freq in Hz
av5=Af/sqrt(1+(f5/fh)^2);
printf("\n Gain magnitude av5 at f5 is = %.2f \n",av5) // Result
f6=3000; // Freq in Hz
av6=Af/sqrt(1+(f6/fh)^2);
printf("\n Gain magnitude av6 at f6 is = %.2f \n",av6) // Result
f7=7000; // Freq in Hz
av7=Af/sqrt(1+(f7/fh)^2);
printf("\n Gain magnitude av7 at f7 is = %.2f \n",av7) // Result
f8=10000; // Freq in Hz
av8=Af/sqrt(1+(f8/fh)^2);
printf("\n Gain magnitude av8 at f8 is = %.2f \n",av8) // Result
f9=30000; // Freq in Hz
av9=Af/sqrt(1+(f9/fh)^2);
printf("\n Gain magnitude av9 at f9 is = %.2f \n",av9) // Result
f10=100000; // Freq in Hz
```

```
av10=Af/sqrt(1+(f10/fh)^2);
printf("\n Gain magnitude av10 at f10 is = %.2f \n",
    av10) // Result
x=[f1 f2 f3 f4 f5 f6 f7 f8 f9 f10];
y=[av1 av2 av3 av4 av5 av6 av7 av8 av9 av10];
gainplot(x,y);
title('Frequency Response');
xlabel('Frequency(Hz)');
ylabel('Voltage gain(dB)');
```

Scilab code Exa 7.4.a Design of second order low pass filter

```
// Chapter7
// Page.No-256
// Example_7_4_a
// Design of second order low pass filter
// Given
clear;clc;
fh=1*10^3; // Cut-off frequency
C2=0.0047*10^-6; // Assumption
C3=C2;
R2=1/(2*%pi*fh*C2);
printf("\n Resistance R2 is = %.1f ohm \n",R2) //
    Result
R2=33*10^3; // Approximation
R3=R2;
printf("\n Resistance R3 is = %.1f ohm \n",R3) //
    Result
R1=27*10^3; // Assumption
Rf=0.586*R1;
printf("\n Resistance Rf is = %.1f ohm \n",Rf) //
    Result
printf("\n Use 20 kohm POT as Rf \n")
```

Scilab code Exa 7.4.b Frequency response of second order highpass filter

```
// Chapter7
// Page.No-260
// Example_7_4_b
// Frequency response of second order highpass filter
// Given
clear;clc;
Af=1.586; // Passband gain of the filter
fh=1000; // Cut-off frequency
f1=10; // Input freq in Hz
av1=Af/sqrt(1+(f1/fh)^4);
printf("\n Gain magnitude av1 at f1 is = %.2f \n",av1) // Result
f2=100; // Freq in Hz
av2=Af/sqrt(1+(f2/fh)^4);
printf("\n Gain magnitude av2 at f2 is = %.2f \n",av2) // Result
f3=200; // Freq in Hz
av3=Af/sqrt(1+(f3/fh)^4);
printf("\n Gain magnitude av3 at f3 is = %.2f \n",av3) // Result
f4=700; // Freq in Hz
av4=Af/sqrt(1+(f4/fh)^4);
printf("\n Gain magnitude av4 at f4 is = %.2f \n",av4) // Result
f5=1000; // Freq in Hz
av5=Af/sqrt(1+(f5/fh)^4);
printf("\n Gain magnitude av5 at f5 is = %.2f \n",av5) // Result
f6=3000; // Freq in Hz
```

```
av6=Af/sqrt(1+(f6/fh)^4);
printf("\n Gain magnitude av6 at f6 is = %.2f \n",av6) // Result
f7=7000; // Freq in Hz
av7=Af/sqrt(1+(f7/fh)^4);
printf("\n Gain magnitude av7 at f7 is = %.2f \n",av7) // Result
f8=10000; // Freq in Hz
av8=Af/sqrt(1+(f8/fh)^4);
printf("\n Gain magnitude av8 at f8 is = %.2f \n",av8) // Result
f9=30000; // Freq in Hz
av9=Af/sqrt(1+(f9/fh)^4);
printf("\n Gain magnitude av9 at f9 is = %.5f \n",av9) // Result
f10=100000; // Freq in Hz
av10=Af/sqrt(1+(f10/fh)^4);
printf("\n Gain magnitude av10 at f10 is = %.6f \n",av10) // Result
x=[f1 f2 f3 f4 f5 f6 f7 f8 f9 f10];
y=[av1 av2 av3 av4 av5 av6 av7 av8 av9 av10];
gainplot(x,y);
title('Frequency Response');
xlabel('Frequency(Hz)');
ylabel('Voltage gain(dB)');
```

Scilab code Exa 7.5.a Design of highpass filter

```
// Chapter7
// Page.No-263
// Example_7_5_a
// Design of highpass filter
// Given
```

```
clear;clc;
fh=1*10^3; // Cut-off frequency
Af=2; // Passband gain of the filter
C=0.01*10^-6; // Assumption
R=1/(2*%pi*fh*C);
printf("\n Resistance R is = %.1f ohm \n",R) //
    Result
printf("\n Use 20 kohm POT as R \n")
R1=10*10^3; // Assumption
printf("\n Resistance R1 is = %.1f ohm \n",R1)
Rf=R1; // Sice passband gain is 2,R1 and Rf must be
    equal
printf("\n Resistance Rf is = %.1f ohm \n",Rf)
```

Scilab code Exa 7.5.b Frequency response of highpass filter

```
// Chapter7
// Page.No-263
// Example_7_5_b
// Frequency response of highpass filter
// Given
clear;clc;
Af=2; // Passband gain of the filter
fl=1000; // Cut-off frequency
f1=100; // Input freq in Hz
av1=(Af*(f1/fl))/sqrt(1+(f1/fl)^2);
printf("\n Gain magnitude av1 at f1 is = %.2f \n",
    av1) // Result
f2=200; // Freq in Hz
av2=(Af*(f2/fl))/sqrt(1+(f2/fl)^2);
printf("\n Gain magnitude av2 at f2 is = %.2f \n",
    av2) // Result
f3=400; // Freq in Hz
```

```
av3=(Af*(f3/fl))/sqrt(1+(f3/fl)^2);
printf("\n Gain magnitude av3 at f3 is = %.2f \n",
    av3) // Result
f4=700; // Freq in Hz
av4=(Af*(f4/fl))/sqrt(1+(f4/fl)^2);
printf("\n Gain magnitude av4 at f4 is = %.2f \n",
    av4) // Result
f5=1000; // Freq in Hz
av5=(Af*(f5/fl))/sqrt(1+(f5/fl)^2);
printf("\n Gain magnitude av5 at f5 is = %.2f \n",
    av5) // Result
f6=3000; // Freq in Hz
av6=(Af*(f6/fl))/sqrt(1+(f6/fl)^2);
printf("\n Gain magnitude av6 at f6 is = %.2f \n",
    av6) // Result
f7=7000; // Freq in Hz
av7=(Af*(f7/fl))/sqrt(1+(f7/fl)^2);
printf("\n Gain magnitude av7 at f7 is = %.2f \n",
    av7) // Result
f8=10000; // Freq in Hz
av8=(Af*(f8/fl))/sqrt(1+(f8/fl)^2);
printf("\n Gain magnitude av8 at f8 is = %.2f \n",
    av8) // Result
f9=30000; // Freq in Hz
av9=(Af*(f9/fl))/sqrt(1+(f9/fl)^2);
printf("\n Gain magnitude av9 at f9 is = %.2f \n",
    av9) // Result
f10=100000; // Freq in Hz
av10=(Af*(f10/fl))/sqrt(1+(f10/fl)^2);
printf("\n Gain magnitude av10 at f10 is = %.2f \n",
    av10) // Result
x=[f1 f2 f3 f4 f5 f6 f7 f8 f9 f10];
y=[av1 av2 av3 av4 av5 av6 av7 av8 av9 av10];
gainplot(x,y);
title('Frequency Response');
xlabel('Frequency(Hz)');
ylabel('Voltage gain(dB)');
```

Scilab code Exa 7.6.a Determination of low cutoff frequency

```
// Chapter7
// Page.No-264
// Example_7_6
// Determination of low cutoff frequency
// Given
clear;clc;
R2=33*10^3;
R3=R2;
C2=0.0047*10^-6;
C3=C2;
fl=1/(2*%pi*sqrt(R2*R3*C2*C3));
printf("\n Low cutoff freq fl is = %.1f Hz \n",fl)
    // Result
```

Scilab code Exa 7.6.b Frequency response of second order highpass filter

```
// Chapter7
// Page.No-264
// Example_7_6_b
// Frequency response of second order highpass
    filter
// Given
clear;clc;
Af=1.586; // Passband gain of the filter
fl=1000; // Cut-off frequency
f1=100; // Input freq in Hz
```

```
av1=Af/sqrt(1+(fl/f1)^4);
printf("\n Gain magnitude av1 at f1 is = %.5f \n",av1) // Result
f2=200; // Freq in Hz
av2=Af/sqrt(1+(fl/f2)^4);
printf("\n Gain magnitude av2 at f2 is = %.4f \n",av2) // Result
f3=700; // Freq in Hz
av3=Af/sqrt(1+(fl/f3)^4);
printf("\n Gain magnitude av3 at f3 is = %.4f \n",av3) // Result
f4=1000; // Freq in Hz
av4=Af/sqrt(1+(fl/f4)^4);
printf("\n Gain magnitude av4 at f4 is = %.4f \n",av4) // Result
f5=3000; // Freq in Hz
av5=Af/sqrt(1+(fl/f5)^4);
printf("\n Gain magnitude av5 at f5 is = %.4f \n",av5) // Result
f6=7000; // Freq in Hz
av6=Af/sqrt(1+(fl/f6)^4);
printf("\n Gain magnitude av6 at f6 is = %.4f \n",av6) // Result
f7=10000; // Freq in Hz
av7=Af/sqrt(1+(fl/f7)^4);
printf("\n Gain magnitude av7 at f7 is = %.4f \n",av7) // Result
f8=30000; // Freq in Hz
av8=Af/sqrt(1+(fl/f8)^4);
printf("\n Gain magnitude av8 at f8 is = %.4f \n",av8) // Result
f9=100000; // Freq in Hz
av9=Af/sqrt(1+(fl/f9)^4);
printf("\n Gain magnitude av9 at f9 is = %.4f \n",av9) // Result
x=[f1 f2 f3 f4 f5 f6 f7 f8 f9];
y=[av1 av2 av3 av4 av5 av6 av7 av8 av9];
gainplot(x,y);
```

```
title('Frequency Response');
xlabel('Frequency(Hz)');
ylabel('Voltage gain(dB)');
```

Scilab code Exa 7.7.a Design of wide bandpass filter

```
// Chapter7
// Page.No-270
// Example_7_7_a
// Design of wide bandpass filter
// Given
clear;clc;
fl=200; // Low cutoff freq in Hz
fh=1*10^3; // High cutoff freq in Hz
C1=0.01*10^-6; // Assumption
R1=1/(2*%pi*fh*C1);
printf("\n Resistance R1 is = %.1f ohm \n", R1) //
    Result
C=0.05*10^-6;
R=1/(2*%pi*fl*C);
printf("\n Resistance R is = %.1f ohm \n", R) //
    Result
printf("\n Bandpass Gain Af is = 4 \n") // Since
    gain of high pass and lowpass is set to 2
R1=10*10^3; // Assumption
printf("\n Resistance R1 is = %.1f ohm \n",R1)
Rf=R1; // Sice passband gain is 2,R1 and Rf must be
    equal
printf("\n Resistance Rf is = %.1f ohm \n",Rf)
```

Scilab code Exa 7.7.b Frequency response of bandpass filter

```
// Chapter7
// Page.No-270
// Example_7_7_b
// Frequency response of bandpass filter
// Given
clear;clc;
Aft=4; // Passband gain of the filter
fl=200; // Lower Cut-off frequency
fh=1000; // Higher Cut-off frequency
f1=10; // Input freq in Hz
av1=(Aft*(f1/fl))/sqrt((1+(f1/fl)^2)*(1+(f1/fh)^2));
printf("\n Gain magnitude av1 at f1 is = %.4f \n",av1) // Result
f2=30; // Freq in Hz
av2=(Aft*(f2/fl))/sqrt((1+(f2/fl)^2)*(1+(f2/fh)^2));
printf("\n Gain magnitude av2 at f2 is = %.4f \n",av2) // Result
f3=100; // Freq in Hz
av3=(Aft*(f3/fl))/sqrt((1+(f3/fl)^2)*(1+(f3/fh)^2));
printf("\n Gain magnitude av3 at f3 is = %.4f \n",av3) // Result
f4=200; // Freq in Hz
av4=(Aft*(f4/fl))/sqrt((1+(f4/fl)^2)*(1+(f4/fh)^2));
printf("\n Gain magnitude av4 at f4 is = %.4f \n",av4) // Result
f5=447.2; // Freq in Hz
av5=(Aft*(f5/fl))/sqrt((1+(f5/fl)^2)*(1+(f5/fh)^2));
printf("\n Gain magnitude av5 at f5 is = %.4f \n",av5) // Result
f6=700; // Freq in Hz
```

```
av6=(Aft*(f6/fl))/sqrt((1+(f6/fl)^2)*(1+(f6/fh)^2));
printf("\n Gain magnitude av6 at f6 is = %.4f \n", av6) // Result
f7=1000; // Freq in Hz
av7=(Aft*(f7/fl))/sqrt((1+(f7/fl)^2)*(1+(f7/fh)^2));
printf("\n Gain magnitude av7 at f7 is = %.4f \n", av7) // Result
f8=2000; // Freq in Hz
av8=(Aft*(f8/fl))/sqrt((1+(f8/fl)^2)*(1+(f8/fh)^2));
printf("\n Gain magnitude av8 at f8 is = %.4f \n", av8) // Result
f9=7000; // Freq in Hz
av9=(Aft*(f9/fl))/sqrt((1+(f9/fl)^2)*(1+(f9/fh)^2));
printf("\n Gain magnitude av9 at f9 is = %.4f \n", av9) // Result
f10=10000; // Freq in Hz
av10=(Aft*(f10/fl))/sqrt((1+(f10/fl)^2)*(1+(f10/fh)^2));
printf("\n Gain magnitude av10 at f10 is = %.4f \n", av10) // Result
x=[f1 f2 f3 f4 f5 f6 f7 f8 f9 f10];
y=[av1 av2 av3 av4 av5 av6 av7 av8 av9 av10];
gainplot(x,y);
title('Frequency Response');
xlabel('Frequency(Hz)');
ylabel('Voltage gain(dB)');
```

Scilab code Exa 7.7.c Calculation of quality factor

```
// Chapter7
// Page.No-270
// Example_7_7_c
// Calculation of quality factor
```

```
// Given
clear;clc;
fh=1*10^3; // Higher cut-off frequency
fl=200; // Lower cut-off frequency
fc=sqrt(fl*fh); // Center frequency
printf("\n Center frequency fc is = %.1f Hz \n",fc) // Result
Q=fc/(fh-fl); // Quality factor
printf("\n Quality factor Q is = %.2f  \n",Q) // Result
```

Scilab code Exa 7.8.a Design of narrow bandpass filter

```
// Chapter7
// Page.No-272
// Example_7_8_a
// Design of narrow bandpass filter
// Given
clear;clc;
fc=1*10^3; // Center frequency
Q=3; // Quality factor
Af=10; // Passband gain
C1=0.01*10^-6; // Assumption
C2=C1;
R1=Q/(2*%pi*fc*C1*Af);
R2=Q/(2*%pi*fc*C1*(2*Q^2-Af));
R3=Q/(%pi*fc*C1);
printf("\n Resistance R1 is = %.1f ohm \n", R1) // Result
printf("\n Resistance R2 is = %.1f ohm \n", R2) // Result
printf("\n Resistance R3 is = %.1f ohm \n", R3) // Result
```

Scilab code Exa 7.8.b Design of narrow bandpass filter

```
// Chapter7
// Page.No-272
// Example_7_8_a
// Design of narrow bandpass filter
// Given
clear;clc;
fc0=1*10^3; // Original center frequency
fc1=1.5*10^3; // New center frequency
R2=5.97*10^3; // Original resistance
R2new=R2*(fc0/fc1)^2;
printf("\n Resistance R1 is = %.1f ohm \n", R2new)
    // Result
```

Scilab code Exa 7.9 Design of wide bandreject filter

```
// Chapter7
// Page.No-274
// Example_7_9
// Design of wide bandreject filter
// Given
clear;clc;
fh=200; // Low cutoff freq in Hz
fl=1*10^3; // High cutoff freq in Hz
C2=0.01*10^-6; // Assumption
R2=1/(2*%pi*fl*C2);
```

```
printf("\n Resistance R2 of highpass section is = %.1f ohm \n", R2) // Result
C=0.05*10^-6;
R=1/(2*%pi*fh*C);
printf("\n Resistance R of lowpass section is = %.1f ohm \n", R) // Result
printf("\n Bandpass Gain Af is = 4 \n") // Since gain of high pass and lowpass is set to 2
R1=10*10^3; // Assumption
printf("\n Resistance R1 is = %.1f ohm \n",R1)
Rf=R1; // Since passband gain is 2,R1 and Rf must be equal
printf("\n Resistance Rf is = %.1f ohm \n",Rf)
```

Scilab code Exa 7.10 Design of notch filter

```
// Chapter7
// Page.No-277
// Example_7_10
// Design of notch filter
// Given
clear;clc;
fn=60; // Notch-out frequency in Hz
C=0.068*10^-6; // Assumption
R=1/(2*%pi*fn*C);
printf("\n Resistance R is = %.1f ohm \n",R) // Result
```

Scilab code Exa 7.11 Phase angle

```
// Chapter7
// Page.No-279
// Example_7_11
// Phase angle
// Given
clear;clc;
f=1*10^3; // Input frequency in Hz
C=0.01*10^-6;
R=15.9*10^3;
phi=-2*atand(2*%pi*f*C*R); // Phase angle
printf("\n Phase angle phi is = %.f deg \n",phi) //
   Result
```

Scilab code Exa 7.12 Design of phase shift oscillator

```
// Chapter7
// Page.No-282
// Example_7_1
// Design of phase shift oscillator
// Given
clear;clc;
fo=200; // Frequency of oscillation
C=0.1*10^-6; // Assumption
R=0.065/(fo*C);
printf("\n Resistance R is = %.1f ohm \n",R) //
   Result
printf("\n Use Resistance R as 3.3 kohm \n")
R=3.3*10^3;
R1=10*R; // To prevent loading of amplifier
Rf=29*R1;
printf("\n Resistance Rf is = %.1f ohm \n",Rf) //
   Result
```

Scilab code Exa 7.13 Design of wein bridge oscillator

```
// Chapter7
// Page.No-282
// Example_7_13
// Design of wein bridge oscillator
// Given
clear;clc;
fo=965; // Frequency of oscillation
C=0.05*10^-6; // Assumption
R=0.159/(fo*C);
printf("\n Resistance R is = %.1f ohm \n",R) //
   Result
R1=12*10^3; // Assumption
Rf=2*R1;
printf("\n Resistance Rf is = %.1f ohm \n",Rf) //
   Result
```

Scilab code Exa 7.14 Design of quadrature oscillator

```
// Chapter7
// Page.No-285
// Example_7_14
// Design of quadrature oscillator
// Given
clear;clc;
fo=159; // Frequency of oscillation
C=0.01*10^-6; // Assumption
```

```
R=0.159/(fo*C);
printf("\n Resistance values R1,R2,R3 is = %.1f ohm
   \n",R) // R1=R2=R3=R
printf("\n Capacitance values C1,C2,C3 is = %.8f
   farad \n",C) // C1=C2=C3=C
```

Scilab code Exa 7.15 Design of squarewave oscillator

```
// Chapter7
// Page.No-287
// Example_7_15
// Design of squarewave oscillator
// Given
clear;clc;
fo=1*10^3;; // Frequency of oscillation
R1=10*10^3; // Assumption
C=0.05*10^-6; // Assumption
R2=1.16*R1;
printf("\n Resistance R2 is = %.1f ohm \n",R2) //
   Result
R=1/(2*fo*C);
printf("\n Resistance R is = %.1f ohm \n",R) //
   Result
```

Scilab code Exa 7.16 Design of triangular wave generator

```
// Chapter7
// Page.No-291
// Example_7_16
```

```
// Design of triangular wave generator
// Given
clear;clc;
fo=2*10^3; // Frequency of oscillation
vo=7; // Output voltage
Vsat=14; // Saturation voltage for opamp 1458
R3=40*10^3; // Assumption
R2=(vo*R3)/(2*Vsat);
printf("\n Resistance R2 is = %.1f ohm \n",R2) // Result
k=R3/(4*fo*R2); // Using fo=R3/(4*R1*C1*R2),k=R1*C1;
C1=0.05*10^-6; // Assumption
R1=k/C1;
printf("\n Resistance R1 is = %.1f ohm \n",R1) // Result
```

Scilab code Exa 7.17.a Nominal frequency

```
// Chapter7
// Page.No-296
// Example_7_17_a
// Nominal frequency
// Given
clear;clc;
R2=1.5*10^3;
R1=10*10^3;
R3=10*10^3;
C1=0.001*10^-6;
V=12; // Supply voltage
Vc=R3*V/(R2+R3); // Using voltage divider rule
printf("\n Terminal voltage Vc is = %.2f V \n",Vc) // Result
fo=2*(V-Vc)/(V*R1*C1);
```

```
printf("\n Approximate Nominal freq fo is = %.1f Hz
   \n",fo) // Result
```

Scilab code Exa 7.17.b Modulation in output frequency

```
// Chapter7
// Page.No-296
// Example_7_17_b
// Modulation in output frequency
// Given
clear;clc;
R2=1.5*10^3;
R1=10*10^3;
R3=10*10^3;
C1=0.001*10^-6;
V=12; // Supply voltage
Vc1=9.5;
Vc2=11.5;
fo1=2*(V-Vc1)/(V*R1*C1);
printf("\n Approximate Nominal freq fo1 is = %.1f Hz
    \n",fo1) // Result
fo2=2*(V-Vc2)/(V*R1*C1);
printf("\n Approximate Nominal freq fo2 is = %.1f Hz
    \n",fo2) // Result
delta_fo=fo1-fo2; // Change in output freq
printf("\n Change in output freq delta_fo is = %.1f
   Hz \n",delta_fo) // Result
```

Chapter 8

Comparators and Converters

Scilab code Exa 8.1 Threshold voltage

```
// Chapter8
// Page.No-320, Figure.No-8.4(a)
// Example_8_1
// Threshold voltage
// Given
clear;clc;
R1=100;R2=56*10^3;
vin=1; // Input voltage in volt
pos_Vsat=14; // Positive saturation voltage in volt
neg_Vsat=-14; // Negative saturation voltage in volt
Vut=(R1/(R1+R2))*(pos_Vsat); // Upper threshold voltage
printf("\n Upper threshold voltage is = %.4f V \n",Vut) // Result
Vlt=(R1/(R1+R2))*(neg_Vsat); // Lower threshold voltage
printf("\n Lower threshold voltage is = %.4f V \n",Vlt) // Result
t=0:0.1:2*%pi;
vut=0.5*sin(t);
subplot(2,1,1);
```

```
plot(t,vut);
title('Input Voltage');
xlabel('t');
ylabel('Vin');
c=0;
for i=0:0.1:2*%pi
   c=c+1;
end
for i=1:c;
if vut(i)>0.025
  v(i)=-14;
else if vut(i)<-0.025
  v(i)=14;
end
end
end
subplot(2,1,2);
plot(t,v);
title('Output Waveform');
xlabel('t');
ylabel('Vo');
```

Scilab code Exa 8.2 Output voltage swing

```
// Chapter8
// Page.No-326, Figure.No-8.7(a)
// Example_8_2
// Output voltage swing
// Given
clear;clc;
vin=5*10^-3;
R=100;
Vd1=-0.7; // Output voltage during positive half-
```

```
    cycle of the input
Vd2=5.1; // Output voltage during negative half-
    cycle of the input
printf("\n Output voltage during positive half-cycle
     of the input is = %.1f V \n",Vd1) // Since zener
     diode is forward biased
printf("\n Output voltage during negative half-cycle
     of the input is = %.1f V \n",Vd2) // Since zener
     diode is reverse-biased
t=0:0.1:2*%pi;
vut=0.5*sin(t);
subplot(2,1,1);
plot(t,vut);
title('Input Voltage');
xlabel('t');
ylabel('Vin');
c=1;
for t=0:0.1:2*%pi
if t<%pi
  v(c)=-0.7;
  else
  v(c)=5.1;
end
c=c+1;
subplot(2,1,2);
plot(v);
end
title('Output Waveform');
xlabel('t');
ylabel('Vo');
```

Scilab code Exa 8.3 Output frequencies

```
// Chapter8
// Page.No-320, Figure.No-8.12
// Example_8_3
// Output frequencies
// Given
clear;clc;
Vin=2; // Input voltage
Fo1=2*10^3; // Output freq Fo when Vin=2V
Fo2=1*10^3; // Output freq Fo/2 when Vin=2V
printf("\n Output freq Fo is = %.f Hz \n",Fo1) // Result
printf("\n Output freq Fo/2 is = %.f Hz \n",Fo2) // Result
count=1;
for i=1:50;     //for 5 cycles
if count<4;
    v(i)=5;
  else
    v(i)=0;
end
if count<10
    count=count+1;
else
    count=1;
end
end
subplot(2,1,1);
plot(v);
title('Output Waveform');
xlabel('t(microsec)');
ylabel('Pulse freq output,Fo(V)');
for i=1:50;     //for 5 cycles
if count<10;
    v(i)=5;
  else
    v(i)=0;
end
if count<20
```

```
    count=count+1;
else
    count=1;
end
end
subplot(2,1,2);
plot(v);
title('Output Waveform');
xlabel('t(microsec)');
ylabel('Pulse freq output,Fo/2(V)');
```

Scilab code Exa 8.4 Output voltage

```
// Chapter8
// Page.No-335, Figure.No-8.14(a)
// Example_8_4
// Output voltage
// Given
clear;clc;
Vo=2.8; // At Finmax of 10kHz
Vo1=Vo/10; // Output voltage at Fin=1kHz
printf("\n Output voltage is = %.2f V \n",Vo1) //
   Result
```

Scilab code Exa 8.5 Output voltage

```
// Chapter8
// Page.No-335, Figure.No-8.25(a)
// Example_8_5
```

```
// Output voltage
// Given
clear;clc;
vin=100*10^-3;
t=0:0.1:2*%pi;
i=1;
for t=0:0.1:2*%pi;
  if t<=%pi
    v(i)=vin*sin(t);
    else
      v(i)=0;
    end
    i=i+1;
end
t=0:0.1:2*%pi;
plot(t,v)
```

Chapter 9

Specialized IC Applications

Scilab code Exa 9.1 Second order inverting butterworth lowpass filter

```
// Chapter9
// Page.No-387
// Example_9_1
// Second order inverting butterworth lowpass filter
// Given
clear;clc;
dc_gain=5;
f1=2*10^3; // Cutoff freq in Hz
Q=10; // Figure of merit
R2=(316*10^3)/10; // Resistance R2
printf("\n Resistance R2 is = %.1f ohm \n",R2) //
   Result
R3=(100*10^3)/((3.16*Q)-1);
printf("\n Resistance R3 is = %.1f ohm \n",R3) //
   Result
printf("\n Resistance R1 is Open \n") // Result
R4=(5.03*10^7)/f1;
printf("\n Resistance R4 is = %.1f ohm \n",R4) //
   Result
R5=R4;
printf("\n Resistance R5 is = %.1f ohm \n",R5) //
```

```
Result
R6=1.8*10^3; // Assumption
R7=dc_gain*R6;
printf("\n Resistance R7 is = %.1f ohm \n",R7) // Result and its a potentiometer
R8=(R6*R7)/(R6+R7);
printf("\n Resistance R8 is = %.3f ohm \n",R8) // Result
```

Scilab code Exa 9.2 Second order inverting butterworth bandpass filter

```
// Chapter9
// Page.No-388
// Example_9_2
// Second order inverting butterworth bandpass filter
// Given
clear;clc;
f1=5*10^3; // Center freq in Hz
Q=10; // Figure of merit
R2=100*10^3; // Constant for band-pass filter
printf("\n Resistance R2 is = %.1f ohm \n",R2) // Result
R3=(100*10^3)/((3.48*Q)-1);
printf("\n Resistance R3 is = %.1f ohm \n",R3) // Result
printf("\n Resistance R1 is Open \n") // Result
R4=(5.03*10^7)/f1;
printf("\n Resistance R4 is = %.1f ohm \n",R4) // Approximately 10kohm
R5=R4;
printf("\n Resistance R5 is = %.1f ohm \n",R5) // Approximately 10kohm and its a potentiometer
```

Scilab code Exa 9.3 Design of notch filter

```
// Chapter9
// Page.No-390
// Example_9_3
// Design of notch filter
// Given
clear;clc;
f1=5*10^3; // notch freq in Hz
Q=10; // Figure of merit
R2=100*10^3; // Constant for band-pass filter
printf("\n Resistance R2 is = %.1f ohm \n",R2) // Result
R3=(100*10^3)/((3.48*Q)-1);
printf("\n Resistance R3 is = %.1f ohm \n",R3) // Result
printf("\n Resistance R1 is Open \n") // Result
R4=(5.03*10^7)/f1;
printf("\n Resistance R4 is = %.1f ohm \n",R4) // Approximately 10kohm
R5=R4;
printf("\n Resistance R5 is = %.1f ohm \n",R5) // Approximately 10kohm and its a potentiometer
R6=10*10^3; // Assumption
printf("\n Resistance R6 is = %.1f ohm \n",R6) // Result
R7=R6;
printf("\n Resistance R7 is = %.1f ohm \n",R7) // Result
R8=R6;
printf("\n Resistance R8 is = %.1f ohm \n",R8) // Result
```

```
R9=(R6*R7*R8)/(R6*R7+R6*R8+R7*R8); // Since R6||R7||R8
printf("\n Resistance R9 is = %.1f ohm \n",R9) // Result
```

Scilab code Exa 9.4 Second order butterworth lowpass filter

```
// Chapter9
// Page.No-398
// Example_9_4
// Second order butterworth lowpass filter
// Given
clear;clc;
f1=500; // Cut-off freq in Hz
Holp=-2; // Passband gain
R1=10*10^3; // Assumption
R2=-R1*Holp; // Using Holp=-R2/R1;
printf("\n Resistance R2 is = %.1f ohm \n",R2) // Result
Q=0.707; // Figure of merit Q is fixed for second order butterworth low-pass filter
R3=Q*R2; // Using Q=R3/R2
printf("\n Resistance R3 is = %.1f ohm \n",R3) // Approximately 15kohm
```

Scilab code Exa 9.5 Value of capacitor

```
// Chapter9
// Page.No-402, Figure.No-9.16(a)
```

```
// Example_9_5
// Value of capacitor
// Given
clear;clc;
Ra=10*10^3; // Resistance in ohm
tp=10*10^-3; // Output pulse width
C=tp/(1.1*Ra);
printf("\n Capacitance C is = %.9f farad \n",C) //
    Approximately 1uF
```

Scilab code Exa 9.6 Value of resistor

```
// Chapter9
// Page.No-402, Figure.No-9.16(a)
// Example_9_6
// Value of resistor
// Given
clear;clc;
f=2*10^3; // Freq of input trigger signal in Hz
C=0.01*10^-6;
tp=1.2/f;
Ra=tp/(1.1*C);
printf("\n Resistance Ra is = %.1f ohm \n",Ra) //
    Result
```

Scilab code Exa 9.7 Value of tc td and f0

```
// Chapter9
// Page.No-402, Figure.No-9.21(a)
```

```
// Example_9_7
// Value of tc,td and f0
// Given
clear;clc;
Ra=2.2*10^3; // Resistance in ohm
Rb=3.9*10^3;  // Resistance in ohm
C=0.1*10^-6; // capacitance in farad
tc=0.69*(Ra+Rb)*C; // Charging time of the capacitor
printf("\n Charging time of the capacitor is = %.6f
    sec \n",tc) // Result
td=0.69*Rb*C; // Discharging time of the capacitor
printf("\n Discharging time of the capacitor is = %
    .6f sec \n",td) // Result
T=tc+td;
fo=1/T // Freq of oscillation
printf("\n Freq of oscillation is = %.1f Hz \n",fo)
    // Result
```

Scilab code Exa 9.8 Freq of free running ramp generator

```
// Chapter9
// Page.No-412, Figure.No-9.24(a)
// Example_9_8
// Freq of free running ramp generator
// Given
clear;clc;
R=10*10^3; // Resistance in ohm
Vcc=5 // Supply voltage in volt
Vbe=0.7 // Base to emitter voltage in volt
C=0.05*10^-6; // Capacitance in farad
Ic=(Vcc-Vbe)/R; // Collector current in ampere
fo=(3*Ic)/(Vcc*C);
printf("\n Freq of free running ramp generator is =
```

```
%.1f Hz \n",fo) // Result
```

Scilab code Exa 9.9 Value of fout fl fc

```
// Chapter9
// Page.No-423, Figure.No-9.33(a)
// Example_9_9
// Value of fout,fl,fc
// Given
clear;clc;
R1=12*10^3; // Resistance in ohm
V_plus=10 // Supply voltage in volt
V_minus=-10 // Supply voltage in volt
C1=0.01*10^-6; // Capacitance in farad
C2=10*10^-6; // Capacitance in farad
fout=1.2/(4*R1*C1);
printf("\n Free running frequency of VCO is = %.1f
   Hz \n",fout) // Result
V=V_plus-V_minus;
fl=(8*fout)/V;
printf("\n Lock range frequency of VCO is = %.1f Hz
   \n",fl) // Result
fc=sqrt(fl/(2*%pi*3.6*10^3*C2));
printf("\n Capture range frequency of VCO is = %.2f
   Hz \n",fc) // Result
```

Scilab code Exa 9.10 Design of current source

```
// Chapter9
```

```
// Page.No-440
// Example_9_10
// Design of current source
// Given
clear;clc;
Vr=5; // Voltage in volt
Il=0.25; // Load current in ampere
Rl=48; // Load resistance in ohm
dropout_volt=2; // Constant for IC7805C
R=Vr/Il; // Approximate result sice Iq is negligible in the eq. Il=(Vr/Il)+Iq where Iq is quiescent current
printf("\n Resistance R is = %.f ohm \n",R) // Result
Vl=Rl*Il;
Vo=Vr+Vl;
printf("\n Output voltage Vo is = %.f V \n",Vo) // Result
Vin=Vo+dropout_volt;
printf("\n Min input voltage Vin is = %.f V \n",Vin) // Result
```

Scilab code Exa 9.11 Design of voltage regulator

```
// Chapter9
// Page.No-444
// Example_9_11
// Design of voltage regulator
// Given
clear;clc;
Vo_min=5; // Min output voltage in volt
Vo_max=12; // Max output voltage in volt
Vref=1.25; // Reference voltage in volt
```

```
Iadj=100*10^-6; // Adjustment pin current in ampere
R1=240; // Assumption
R2_min=R1*(Vo_min-Vref)/(Vref+Iadj*R1); // Using Vo_min=Vref*(1+R2/R1)+Iadj*R2
printf("\n Resistance R2_min is = %.1f ohm \n", R2_min) // Result
R2_max=R1*(Vo_max-Vref)/(Vref+Iadj*R1); // Using Vo_max=Vref*(1+R2/R1)+Iadj*R2
printf("\n Resistance R2_max is = %.1f ohm \n", R2_max) // Result
printf("\n Therefore resistance should be varied from R2_min to R2_max values.To do this we take R2 as 3kohm potentiometer \n")
C2=1*10^-6; // Added to the circuit to improve transient response
C3=1*10^-6; // Added to the circuit to obtain high ripple rejection ratios
```

Scilab code Exa 9.12 Design of stepdown switching regulator

```
// Chapter9
// Page.No-453
// Example_9_12
// Design of stepdown switching regulator
// Given
clear;clc;
Iomax=500*10^-3; // Max output current in ampere
Vo=5; // Output voltage in volt
Vd=1.25; // Voltage drop across the power diode in volt
Vin=12; // Input voltage in volt
Vs=1.1; // Output saturation voltage in volt
Vripple=50*10^-3; // Output ripple voltage in volt
```

```
Vref=1.245; // Reference voltage in volt
Vr2=1.2; // Voltage across resistance R2 in volt
Ipk=2*Iomax; // Sense current in ampere
printf("\n Sense current,Ipk is = %.f A \n",Ipk) // Result
Rsc=0.33/Ipk; // Sense resistance in ohm
printf("\n Sense resistance,Rsc is = %.2f ohm \n",Rsc) // Result
K=(Vo+Vd)/(Vin-Vs-Vo); // K= ton/toff
printf("\n Constant K = %.2f  \n",K) // Result
printf("\n i.e, ton is K times of toff \n")
f=20*10^3; // Assuming operating freq in Hz
T=1/f;
toff=T/2.06; // Using ton+toff=T and substituting for ton
printf("\n OFF time period,toff is = %.8f sec \n",toff) // Result
ton=1.06*toff;
printf("\n ON time period,ton is = %.8f sec \n",ton) // Result
Ct=45*10^-5*toff; // Oscillator timing capacitance in farad
printf("\n Oscillator timing capacitance,Ct is = %.10f F \n",Ct) // Result
L=((Vo+Vd)/Ipk)*toff; // Inductance in henry
printf("\n Inductance,L is = %.8f H \n",L) // Result
Co=Ipk*((ton+toff)/(8*Vripple)); // Output capacitance in farad
printf("\n Output capacitance,Co is = %.7f F \n",Co) // Result
I2=0.1*10^-3; // Assuming the current through R2
R2=Vref/I2; // Resistance R2 in ohm
printf("\n Resistance R2 is = %.1f ohm \n",R2) // Result
R2=12*10^3; // Taking approximate value
R1=(R2*(Vo-Vr2))/Vr2; // Using Vr2=(R1*Vo)/R1+R2, voltage divider rule
printf("\n Resistance R1 is = %.1f ohm \n",R1) //
```

```
Result
efficiency=((Vin-Vs+Vd)/Vin)*(Vo/(Vo+Vd))*100;
printf("\n efficiency is = %.1f \n",efficiency) //
   Result
```

Scilab code Exa 9.13 Design of stepdown switching regulator

```
// Chapter9
// Page.No-458
// Example_9_13
// Design of stepdown switching regulator
// Given
clear;clc;
Iomax=3; // Max output current in ampere
Vo=5; // Output voltage in volt
Vd=1.25; // Voltage drop across the power diode in
   volt
Vin=12; // Input voltage in volt
Vs=1.1; // Output saturation voltage in volt
Vripple=50*10^-3; // Output ripple voltage in volt
Vref=1.245; // Reference voltage in volt
Vr2=1.2; // Voltage across resistance R2 in volt
Ipk=2*Iomax; // Sense current in ampere
printf("\n Sense current,Ipk is = %.f A \n",Ipk) //
   Result
Rsc=0.33/Ipk; // Sense resistance in ohm
printf("\n Sense resistance,Rsc is = %.3f ohm \n",
   Rsc) // Result
K=(Vo+Vd)/(Vin-Vs-Vo); // K= ton/toff
printf("\n Constant K = %.2f  \n",K) // Result
printf("\n i.e, ton is K times of toff \n")
f=20*10^3; // Assuming operating freq in Hz
T=1/f;
```

```
toff=T/2.06; // Using ton+toff=T and substituting for ton
printf("\n OFF time period, toff is = %.8f sec \n",toff) // Result
ton=1.06*toff;
printf("\n ON time period, ton is = %.8f sec \n",ton) // Result
Ct=45*10^-5*toff; // Oscillator timing capacitance in farad
printf("\n Oscillator timing capacitance, Ct is = %.10f F \n",Ct) // Result
L=((Vo+Vd)/Ipk)*toff; // Inductance in henry
printf("\n Inductance, L is = %.8f H \n",L) // Result
Co=Ipk*((ton+toff)/(8*Vripple)); // Output capacitance in farad
printf("\n Output capacitance, Co is = %.7f F \n",Co) // Result
```