Scilab Textbook Companion for Electronic Measurements And Instrumentation by K. Lal Kishore[1]

Created by
Vishal Changlani
Electronics and telecommunication Engineering
Electronics Engineering
Mumbai University
College Teacher
None
Cross-Checked by
None

June 7, 2018

[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: Electronic Measurements And Instrumentation

Author: K. Lal Kishore

Publisher: Pearson Education India

Edition: 1

Year: 2012

ISBN: 9788131721995

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

Measurements and Instruments

Scilab code Exa 1.1 Determine the power dissipated and the probable error

```
// Exa 1.1

clc;
clear;

// Given

I = 10; // Current in Amp
dI = 0.1; // Probability of error in I (Amp)
R = 100; // Resistor value in Ohms
dR = 2; // Probability of error in R (Ohms)


// Solution

printf('The power dissipated P = I^2*R \n');

printf(' The probable error can be determined with
    the help of Erss(Root Sum Square) Formula, i.e
    Error = sqrt((dR*I^2)^2 + (2*I*R*dI)^2)  \n');

```

```
PE = sqrt((dR*I^2)^2 + (dI*2*I*R)^2); // Probable
   error
P = I^2 * R;

printf(' Error = %d W \n',round(PE));
printf(' Power dissipated P = %d kW \n',P*10^-3);
```

Scilab code Exa 4.1 Determine the percentage of non linearity in a saw tooth outpu

```
// Exa 4.1

clc;
clear;

// Given

// An oscilloscope

R = 400; // Resistance(k Ohms)
C = 0.025; // capacitance(micro Farad)
T = 0.4; // Time period of saw-tooth output waveform
   (msec)

// Solution

printf('  The percentage of non linearity i.e
   deviation in output can be given as t/(4*R*C)\n
   ');
PD = (T*10^-3)/(4*R*10^3*C*10^-6) ;

printf(' Therefore, by calculation, percent
   deviation = %d percent \n ',PD*100);
```

Scilab code Exa 1.2 Determine the resolution of the voltmeter in volts

```

// Exa 1.2

clc;
clear;

// Given

No_Div = 50; // No of divisions
V = 100; // Max voltage measured (V)

// Solution

printf(' Resolution is the smallest change in input
    that can be measured \n The meter can be read to
    1/2 division \n');
value = No_Div/V;
printf(' So, the resolution is 1/2 divisions and its
     value in volts is %.1f V \n', value);

printf(' Therefore, the resolution of instrument is
    0.5 V \n');
```

Scilab code Exa 1.3 Determine the resolution in volts for three and half digit DVM

```

// Exa 1.3

clc;
clear;

// Given

```

```
// A 3_1/2 digit DVM
V = 19.99; // Max voltage in Volts

// Solution

printf('The maximum number of counts that can be
   made with 9 3_1/2 digit DVM is  1999 \n The
   smallest change in input that can be measured is
   1 count \n');
// 1 count in volts corresponds to resolution :-
// 1999 counts = 19.99 V
// 1 count = ?
Resolution = 19.99/1999;

printf(' Resolution = %d mV \n',round(Resolution
   *10^3));
```

Scilab code Exa 1.4 To calculate the percentage error due to loading effect

```
// Exa 1.4

clc;
clear;

// Given

S = 10*10^3; // Sensitivity of voltmeter in Ohms/
   Volt
V = 75; // Reading in Volts
Vmax = 100; // Max voltage in Volts
I = 1.5*10^-3; // reading in Amp

// Solution

printf('Consider Fig.1.10, it shows Rm as meter of
```

```
    voltmeter drawing some current \n Thus, loading
    of the source happens i.e, loading effect \n');
Rapparent = V/I;
Rm = Vmax * S;
// Rapparent = parallel combination of Rm and Rx
// Therefore Rx can be given as
Rx = (1/Rapparent - 1/Rm)^-1;
printf(' True value of Rx = %.2f K Ohms \n',Rx);

Error = 100* (Rx-Rapparent)/Rx ; // Error in percent
printf(' The percentage error due to loading effect
    = %.1f percent \n',Error);

//The answers vary due to round off error
```

Scilab code Exa 1.5 To calculate value of shunt resistance

```
// Exa 1.5

clc;
clear;

// Given

Rm = 100; // Resistor value in Ohms
I = 10; // Current in Amp
Im = 1*10^-3; // Meter current in Amp

// Solution

Ish = I - Im;
Vm = Im * Rm; // Vm = Vsh
Vsh = Vm;
Rsh = Vsh/Ish;

```

```
printf('The value of shunt resistance Rsh = %.2f
   Ohms \n',Rsh);

//The answer provided in the textbook is wrong
```

Scilab code Exa 1.6 To find the value of Rsh

```
// Exa 1.6

clc;
clear;

// Given

Imax = 100*10^-6; // Initial range of Ammeter in Amp
Rm = 800; // Meter resistance in Ohms
I1max = 0.1; // Range to be extended in Amp
I2max = 10; // Range to be extended in Amp

// Solution

printf(' Referring Figs. 1.21 and 1.22 :- \n');

n = I1max/Imax;
Rsh = Rm/(n-1);
printf('  Ra + Rb + Rc = Rsh; \n ');
printf(' The value of Rsh by calculations = %.4f
   Ohms \n',Rsh);
printf('  Referring calculations done in textbook,\n
     we can get values of Ra,Rb and Rc as follows :-
    \n');
Rc = Imax*(Rsh+Rm)/I2max;
Rb = (Imax/I1max)*(Rsh+Rm) - Rc;
Ra = Rsh-(Rb+Rc);
printf('  Ra = %.3f Ohms, Rb = %.3f Ohms and Rc = %
```

```
    .3 f Ohms \n',Ra,Rb,Rc);

//The answer provided in the textbook is wrong for
    Rc and not at all given for Ra and Rb
```

Scilab code Exa 1.7 To determine the values of Ra alongwith Rb and Rc

```
// Formulae's from Example 1.6 are used here
// Exa 1.7

clc;
clear;

// Given

// Referring circuit given in Fig. 1.23
Rm = 1000; // Meter resistance in Ohms
Im = 100*10^-6; // Meter current in Amp
I1 = 1; // im Amp
I2 = 0.1; // in Amp
I3 = 10; // in Amp

// Solution

n =  I3/I2;
Rsh = Rm/(n-1);
Rc = (Im/I2)*(Rsh+Rm);
Rb = Rc - (Im/I1)*(Rsh+Rm);
Ra = Rsh-(Rb+Rc);

printf('The Values of Ra, Rb and Rc are %.2f Ohms, %
    .2f Ohms and %.2f Ohms respectively \n',Ra,Rb,Rc)
    ;

//The answer provided in the textbook is wrong for
```

Ra

Scilab code Exa 1.9 `To calculate the value of multiplier resistances`

```
// Exa 1.9

clc;
clear;

// Given

// Referring Fig.1.26
Ifs = 50*10^-6; // Full scale deflection current in Amp
Rm = 11; // Meter resistance in Ohms
R1 = 3; // Range in Volts
R2 = 10; // range in Volts
R3 = 30; // Range in Volts

// Solution

S = 1/Ifs; // Sensitivity in Ohms/V

printf('The values of multiplier resistances in the different ranges are :- \n');
printf(' For 3-V range :');
Rs1 = S*R1-Rm;
printf('%d k Ohms \n',Rs1/1000);
printf(' For 10-V range :');
Rs2 = S*R2-Rm;
printf('%d k Ohms \n',Rs2/1000);
printf(' For 30-V range :');
Rs3 = S*R3-Rm;
printf('%d k Ohms \n',Rs3/1000);

```

```
//The answer provided in the textbook is wrong for Rs1
```

Scilab code Exa 1.10 Determine the percentage errors of the reading due to ammeter

```
// Exa 1.10

clc;
clear;

// Given

// Referring Fig. 1.28 and 1.29
Rm = 78; // Meter resistance in Ohms
Ra = 1000; // in Ohms
Rb = 1000; // in Ohms
Rc = 1000; // in Ohms

// Solution

Rth = Rc + Ra*Rb/(Ra+Rb);

// Let Im/Ie be 'x' where Ie = expected value and Im be shown value
x = Rth/(Rth+Rm);

Error = (1-x)*100;

printf('The meter reading shown is %d percent of the expected value. Therefore, Error = %d percent \n',100-round(Error),round(Error));
```

Scilab code Exa 1.11 Determine the resistor values for full scale along with 20 pe

```
// Exa 1.11

clc;
clear;

// Given

// An Ohm meter
Ifs = 0.001; // Current in Amp
V =5; // Supply voltage in Volts
Rm = 200; // Meter resistance in Ohms

// Solution

// Ifs = V/(Rm+Rp);

Rp =V/Ifs - Rm;

printf('Value of Rx with 20 percent full-scale
    deflection :- \n');
P1 = 0.2;
Rx1 = (Rp+Rm)/P1 - (Rp+Rm);
printf('The resistor value for 20 percent of full
    scale deflection = %d k Ohms \n',Rx1/1000);
printf(' Value of Rx with 40 percent full-scale
    deflection :- \n');
P2 = 0.4;
Rx2 = (Rp+Rm)/P2 - (Rp+Rm);
printf('The resistor value for 40 percent of full
    scale deflection = %.1f k Ohms \n',Rx2/1000);
printf(' Value of Rx with 50 percent full-scale
    deflection :- \n');
P3 = 0.5;
Rx3 = (Rp+Rm)/P3 - (Rp+Rm);
printf('The resistor value for 50 percent of full
    scale deflection = %d k Ohms \n',Rx3/1000);
```

Scilab code Exa 1.12 To calculate the resistance of a shunt required to make a mil

```
// Exa 1.12

clc;
clear;

// Given

// Referring Fig. 1.49
Ifs = 10; // Full scale deflection current in Amp
Im = 0.015; // Meter resistance in Amp
R = 5; // Resistance in Ohms

// Solution

Isg = Ifs-Im;
Rsg = Im*R/Isg;

printf('The value of shunt resistance for the milli
    Ammeter = %.2f * 10^-3  Ohms \n',Rsg*1000);
```

Scilab code Exa 1.13 To find the current range of the ammeter

```
// Exa 1.13

clc;
clear;

// Given

Rm = 1; // Meter resistance in ohms
```

```
Vmax = 250; // Max voltage(V)
R1 = 4.999; // Series resistance in Ohms
R2 = 1/499; //Shunt resistance in Ohms
V = 500; // Volage to be measured(V)
// Solution

Imeter = V/(R1+Rm);
printf('Meter current Imeter = %.2f A \n ',Imeter);
printf('The current through the meter should not
    exceed %.1f A \n Hence, Voltage drop = %.2f V \n'
    ,Imeter,Imeter*Rm);
Vdrop = Imeter*Rm;
// When Shunt resistance is connected accross meter
CurrentRange = Vdrop/R2 + Imeter;
Rs =(V-Imeter)/0.05;
printf(' Series resistance Rs = %d Ohms \n',Rs);

//The answer provided in the textbook are incorrect
```

Scilab code Exa 1.14 To determine the values of Ra along with Rb and Rc

```
// Formulae's from Example 1.6 are used here

// Exa 1.14

clc;
clear;

// Given

// Referring Fig. 1.50
Im = 100*10^-6; // Meter resistance in Amp
Rm = 1000; // Meter resistance in Ohms
I1 = 1; // in Amp
I2 = 0.1; // in Amp
```

```
I3 = 0.01; // in Amp

// Solution

n = I3/Im;
Rsh = Rm/(n-1);
printf('  Ra + Rb + Rc = Rsh; \n ');
printf(' The value of Rsh by calculations = %.4f Ohms \n',Rsh);

Rc = Im*(Rsh+Rm)/I1;
Rb = (Im/I2)*(Rsh+Rm) - Rc;
Ra = Rsh-(Rb+Rc);
printf('  Ra = %.3f Ohms, Rb = %.3f Ohms and Rc = %.3f Ohms \n',Ra,Rb,Rc);

//The answer provided in the textbook is wrong for Ra calculation
```

Scilab code Exa 1.15 To design a range switch for the DC volt section of a balance

```
// Exa 1.15

clc;
clear;

// Given

// Refer fig. 1.73
Rtotal = 11*10^6; // Total resistance in Ohms
Vg = 1; // Voltage at gate of balancing circuit
Vmax = 1000; // Max voltage in Volts
// Solution

printf('When the selector switch is at 1000 V
```

```
position , and the mass input is 1000 V, the drop across R7 should be 1 value \n');
R7 = (Vg/Vmax)*Rtotal;
printf('R7 = %d k Ohms \n',R7/1000);
printf('Similarly , when the selector switch is at the 300 V position , the drop across (R6+R7) should be 1 V \n');
R6 = (Vg/300)*Rtotal-R7;
printf('R6 = %.2f k Ohms \n',R6/1000);
R5 = (Vg/100)*Rtotal-(R7+R6);
printf('R5 = %.2f k Ohms \n',R5/1000);
R4 = (Vg/30)*Rtotal-(R7+R6+R5);
printf('R4 = %.2f k Ohms \n',R4/1000);
R3 = (Vg/10)*Rtotal-(R7+R6+R5+R4);
printf('R3 = %.2f k Ohms \n',R3/1000);
R2 = (Vg/3)*Rtotal-(R7+R6+R5+R4+R3);
printf('R2 = %.2f k Ohms \n',R2/1000);
R1 = Rtotal - (R2+R3+R4+R5+R6+R7);
printf('R1 = %.2f k Ohms \n',R1/1000);
```

Scilab code Exa 1.16 To calculate the emf of the unknown source along with resolut

```
// Exa 1.16

clc;
clear;

// Given

// Refer Fig.1.74
Rmv = 200; // Voltmeter resistance in Ohms
Vt = 3; // Terminal voltage(V)
S = 1; // Sensitivity in mm/microV
Rmi = 100; // Galvanometer resistance in Ohms
Deflection = 250; // in mm
```

```
S1 = 5; // sesitivity of second galvanometer in mm/micro Amp
Ri = 1000; // Internal resistance of 2nd galvanometer
// Solution

Ig = Deflection/S;
printf(' Current through the galvanometer = %d micro Amp \n',Ig);
Rtotal = Rmv+Rmi;
Vrtotal = Rtotal*Ig*10^-6; // in Volts
Ek = Vt - Vrtotal;
printf('The emf of the unknown source = %.3f V \n',Ek);
printf('1 mm correspponds to 1 micro Amp. Therefore, Resolution = %d micro V/mm \n',Rtotal);

printf('For galvanometer A, 1 mm deflection for 300 mV. So, Sa = 1/300 mm/microV \n');
Sa = 1/300; // Sensitivity of galvanometer A in mm/micro V
printf('For galvanometer B, - \n');
Rbtotal = Ri+Rmv;
// A 5mm deflcetion is caused by 1200 micro V
printf('Sb = 5/1200  mm/microV \n'); // mm/microV
printf('Galvanometer B provides the amplified sensitivity i.e, Since, Sb>Sa \n');
```

Scilab code Exa 1.17 To find the value of multiplier resistance for 10 Vrms AC ran

```
// Exa 1.17

clc;
clear;

```

```
// Given

// Refer Fig. 1.77
Ifs = 10^-6; // Full scale deflection current in Amp
Rm = 300; // Meter resistance in Ohms
Erms = 10; // in Volts
Idc = 1*10^-3; // in Amp
// Solution
S = 1/Ifs;
Rs = 0.45* Erms/Idc - Rm;
printf(' The value of multiplier resistance Rs = %.1
    f k Ohms \n',Rs/1000);
```

Scilab code Exa 1.18 `Determine the value of Rs`

```
// Exa 1.18

clc;
clear;

// Given

Ifs = 10^-3; // Full scale deflection current in Amp
Rm = 500; // Meter resistance in Ohms
Range = 10; // Em = 10*Vrms
// Solution

Sdc = 1/Ifs; // Dc Sensitivity in Ohms/Volt
Sac = 0.9*Sdc; // Ac Sensitivity in Ohms/Volt

Rs = Sac * Range - Rm;
printf('The value of multiplier resistance Rs = %d
    Ohms \n',Rs);
```

Scilab code Exa 1.19 Design a thermocouple voltmeter for three ranges

```
// Exa 1.19

clc;
clear;

// Given

// Design of thermocouple voltmeter
// Three ranges
V1 = 5; // Volts
V2 = 10; // Volts
V3 = 25; // Volts
Ifs = 50*10^-3; // Amp
Rm = 200; // Ohms
Imax = 5*10^-3; // Amps
Rheater = 200; // Ohms

// Solution

printf(' To get a FSD for 5,10 and 25V, current
    through the heater must be limited to 5mA \n');

printf('For a 5V range \n');
Rs1 = V1/Imax - Rm;
printf('Series resistance Rs = %d Ohms \n',Rs1);
printf('For a 10V range \n');
Rs2 = V2/Imax - Rm;
printf('Series resistance Rs = %d Ohms \n',Rs2);
printf('For a 25V range \n');
Rs3 = V3/Imax - Rm;
printf('Series resistance Rs = %d Ohms \n',Rs3);
```

```
//The answer provided in the textbook for Rs3 is
   wrong
```

Scilab code Exa 1.20 Design a standard DC voltage source of 12 V

```
// Exa 1.20

clc;
clear;

// Given

// Refer Fig. 1.86
Vdc = 12; // Volts
Vac = 20; // Volts
Vz = 12; // Volts
Iz = 10*10^-3; // in Amps

// Solution

R =(Vac-Vdc)/Iz;
printf('The value of resistance R = %d Ohms \n',R);
P = Vdc*Iz;
printf(' Power rating of the Zener = %d mW \n',P
   *1000);
printf(' Factor 2 is the safety factor \n The power
   rating of the resistor taking a safety factor of
   2 is P = ');
psafety = 2*(Vdc - Vz)/R;
printf(' %.2f W \n',psafety);
printf(' 1/16 W resistor curve serves the purpose \n
   ');

//The answer provided in the textbook for power
   rating is wrong
```

Scilab code Exa 1.21 Design a volt box such that addition of R1 and R2 should be 1

```
// Exa 1.21

clc;
clear;

// Given

// A Volt box design
Vs = 100; // Input voltage (V)
V2 = 5; // Output voltage (V)
Rs = 10*10^6; // Desired sum of resistance(R1+R2)
   Ohms


// Solution

// By voltage divider formula, we get
//  R2/(R1+R2) = V2/Vs ;
// i.e, By simplifying
R2 = Rs*V2/Vs;

R1 = Rs - R2;
printf(' The desired values of R1 and R2 to satisfy
   Volt box requirements are %.1f M ohms and %.2f M
   ohms respectively \n ',R1/10^6,R2/10^6);

//The answer provided in the textbook is wrong
```

Scilab code Exa 1.22 To calculate the percentage error in the meter indication

```
// Exa 1.22

clc;
clear;

// Given

// A sine wave AC input

// Solution

printf('For a square wave, the form factor is 1.0,
   that is, the average and rms value are same \n');
printf(' For a sine wave, the form factor is 1.1,
   that is, the rms is 1.11 times the average \n');
printf(' Since the meter is calibrated for a sine
   wave, for a 1.0 V rms value of square wave, it
   indicates 1.11 V \n');
FFsq = 1.0;
FFsi = 1.11;
Perror = 100*(FFsi-FFsq)/FFsq;
printf(' The percentage error in the meter
   indication = %d percent \n',Perror);
```

Scilab code Exa 1.23 To find the period of integration of dual slope integrating t

```
// Exa 1.23

clc;
clear;

// Given

// Dual slope integrating-type DVM
C = 0.1*10^-6; // Capacitor in Farads
```

```
R = 10*10^3; // Resistance in  Ohms
Vr = 2; // Reference Voltage(Volts)
Vmax = 10; // maximum output of circuit(Volts)

// Solution

Tc = C*R; // Integrator Time Constant
Vo = Vr/Tc ; // Integrator output in Volt/sec

Ti = Vmax/Vo;  //in sec

printf(' The period of integration of dual slope
   integrating-type DVM = %d m sec \n',Ti*1000);

//The answer provided in the textbook is wrong
```

Scilab code Exa 1.24 To determine the limits of capacitance that it is guranteed

```
// Exa 1.24

clc;
clear;

// Given

Am = 20; // Capacitance in Farads
dr= 5 ; // Percentage variation in capacitor value

// Solution

// A = Am (  ) Am  dr/100 ; // A is guranteed value
    of capacitor

A_upperlimit = Am*(1+ dr/100) ;
A_lowerlimit = Am*(1- dr/100) ;
```

```

printf(' The guranteed  limits of capacitance range
   from %.1f F to %.1f F \n',A_lowerlimit,
   A_upperlimit);
```

Scilab code Exa 1.25 To find the limiting error

```
// Exa 1.25

clc;
clear;

// Given

// A 0-250 range milliAmmeter
Er = 2; // Percentage accuracy of Ammeter in terms
   of FSR
I = 150; // Measurement of Ammeter in mA
Ifsr = 250; // Full scale reading of milliAmmeter (
   mA)

// Solution

dV = Er/100 * Ifsr; // Error in FSR reading
Lr = 100*dV/I;
printf('The limiting error = %.2f percent \n',Lr);
```

Scilab code Exa 1.26 To calculate the limiting error

```
// Exa 1.26

clc;
clear;
```

```
// Given

R = 50; // Resistance value (Ohms)
dR = 0.2; // variation in Resistance value (Ohms)
I = 4; // Current value measured (Amp)
dI = 0.02; // variation in current measurements (Amp
    )

// Solution

Per_Limiting_Error_Resis = dR/R * 100;
Per_Limiting_Error_Curr = dI/I * 100;

P = I^2 * R;
dP = Per_Limiting_Error_Curr*2 +
    Per_Limiting_Error_Resis;

printf('The limiting error in resistance measurement
     =    %.2f percent \n',Per_Limiting_Error_Resis);
printf(' The limiting error in current measurement =
        %.2f percent \n',Per_Limiting_Error_Curr);
printf(' The limiting error in power measurement =
    %.2f percent \n',dP);
```

Scilab code Exa 1.27 To determine the resolution of a moving coil ammeter

```
// Exa 1.27

clc;
clear;

// Given

// A moving coil Ammeter
```

```
FSR = 10; // Full scale reading in Amp
No_of_div = 100;

// Solution

one_scale_div = FSR/No_of_div ; // in Amp
Resolution = 1/2 * one_scale_div ; // Since, the instrument can read upto half of the full-scale division(Amp)
printf('Resolution = %d mA \n', Resolution*1000);
```

Scilab code Exa 1.28 To determine the limiting error

```
// Exa 1.28

clc;
clear;

// Given

// Limiting error for series and parallel combination of capacitors
c1 = 99; // Capacitor value in Mf
dc1 = 1; // Variation in capacitor value in Mf
c2 = 49; // Capacitor value in Mf
dc2 = 1; // Variation in capacitor value in Mf

// Solution

// C1 = c1(   ) dc1;
// C2 = c2(   ) dc2;
printf('For parallel combination, we have y = C1+C2 \n');
dY_parallel = dc1 + dc2;
printf(' Limiting error for parallel combination = (
```

```
    ) %d Mf \n',dY_parallel);
printf(' For series combination, we have 1/y = 1/C1
    +1/C2 \n');
Yseries = c1*c2/(c1+c2);
dYseries = (-dc1+c1)*(-dc2+c2)/(c1+c2-dc1-dc2);
dY = Yseries - dYseries;
printf(' Limiting error for series combination = (
    ) %.3f Mf \n',dY);

//The answer provided in the textbook is wrong
```

Scilab code Exa 1.29 To determine the relative error for the series and parallel c

```
// Exa 1.29

clc;
clear;

// Given

// 3 resistances in series and parallel combination

r1 = 200; // First resistance in (Ohms)
dr1 = 5 ; // Percentage variation for first
    resistance
r2 = 100; // Second resistance in (Ohms)
dr2 = 5 ; // Percentage variation for second
    resistance
r3 = 50; // Third resistance in (Ohms)
dr3 = 5 ; // Percentage variation for third
    resistance

// Solution

printf('Lets say Rse be the series combination of
```

```
resistances \n');// series
Rse = r1+r2+r3;
Relative_Error_series = r1/Rse * dr1 + r2/Rse * dr2 + r3/Rse * dr3; // in percentage
Error_series_Ohms = Rse * Relative_Error_series/100;
printf(' The Relative error for series combination(Rse) is %d percent which is equivalent to %.2f Ohms \n',Relative_Error_series,Error_series_Ohms);
printf(' Lets say Rpa be the parallel combination of resistances \n');// parallel
Rpa = r1*r2*r3/(r2*r3+r1*r2+r1*r3);  // lets say (x/y1+y2+y3)
Error_x = dr1+dr2+dr3;
Error_y1 = dr1+dr2;
Error_y2 = dr2+dr3;
Error_y3 = dr3+dr1;
Error_Y = r1/Rse * Error_y1 + r2/Rse * Error_y2 + r3/Rse * Error_y3;
Relative_Error_parallel = Error_x+ Error_Y;  // in percentage
Error_parallel_Ohms = Rpa * Relative_Error_parallel/100 ;
printf(' The Relative error for parallel combination(Rpa) is %d percent which is equivalent to %.4f Ohms \n',Relative_Error_parallel,Error_parallel_Ohms);
```

Scilab code Exa 1.30 To find the arithmatic mean along with average and standard d

```
// Exa 1.30

clc;
clear;

```

```
// Given

// Various Inductance Measurements
L1 = 1.003; // First reading in mH
L2 = 0.998; // second reading in mH
L3 = 1.001; // third reading in mH
L4 = 0.991; // fourth reading in mH
L5 = 1.009; // Fifth reading in mH
L6 = 0.996; // sixth reading in mH
L7 = 1.005; // seventh reading in mH
L8 = 0.997; // eight reading in mH
L9 = 1.008; // nineth reading in mH
L10 = 0.994; // tenth reading in mH
n = 10; // total no of readings

// Solution

AM = (L1+L2+L3+L4+L5+L6+L7+L8+L9+L10)/n;
printf('The arithmatic mean = %.4f mH \n',AM);

// Deviation for each reading will be -
d1 = L1 - AM; // deviation for 1st reading
d2 = L2 - AM; // deviation for 2nd reading
d3 = L3 - AM; // deviation for 3rd reading
d4 = L4 - AM; // deviation for 4th reading
d5 = L5 - AM; // deviation for 5th reading
d6 = L6 - AM; // deviation for 6th reading
d7 = L7 - AM; // deviation for 7th reading
d8 = L8 - AM; // deviation for 8th reading
d9 = L9 - AM; // deviation for 9th reading
d10 = L10 - AM; // deviation for 10th reading

Avg_deviation = (d1+d2+d3+d4+d5+d6+d7+d8+d9+d10)/n;
printf(' The average deviation = %d mH \n',
    Avg_deviation);

SD = sqrt((d1^2+d2^2+d3^2+d4^2+d5^2+d6^2+d7^2+d8^2+
    d9^2+d10^2)/(n-1));
```

```
printf(' The standard deviation = %.3f mH \n',SD);

//The answer provided in the textbook is wrong
```

Scilab code Exa 1.31 To find the arithmatic mean along with average and standard d

```
// Exa 1.31

clc;
clear;

// Given

// Various Current Measurements

I1 = 41.7; // First reading in A
I2 = 42; // second reading in A
I3 = 41.8; // third reading in A
I4 = 42; // fourth reading in A
I5 = 42.1; // Fifth reading in A
I6 = 41.9; // sixth reading in A
I7 = 42; // seventh reading in A
I8 = 41.9; // eight reading in A
I9 = 42.5; // nineth reading in A
I10 = 41.8; // tenth reading in A
n=10; // Total no of observations
I = [41.7;42;41.8;42;42.1;41.9;42;41.9;42.5;41.8];

// Solution

AM = (I1+I2+I3+I4+I5+I6+I7+I8+I9+I10)/n;
printf('The arithmatic mean = %.4f A \n',AM);

// Deviation for each reading will be -
d1 = I1 - AM; // deviation for 1st reading
```

```
d2 = I2 - AM; // deviation for 2nd reading
d3 = I3 - AM; // deviation for 3rd reading
d4 = I4 - AM; // deviation for 4th reading
d5 = I5 - AM; // deviation for 5th reading
d6 = I6 - AM; // deviation for 6th reading
d7 = I7 - AM; // deviation for 7th reading
d8 = I8 - AM; // deviation for 8th reading
d9 = I9 - AM; // deviation for 9th reading
d10 = I10 - AM; // deviation for 10th reading

SD = sqrt((d1^2+d2^2+d3^2+d4^2+d5^2+d6^2+d7^2+d8^2+d9^2+d10^2)/(n-1));
printf(' The standard deviation = %.3f A \n',SD);

Y = 0.6745*SD;
printf(' Probable error of one reading = %.3f A \n',Y);
Vm = Y/sqrt(n-1);
printf(' Probable error of mean = %.3f A \n',Vm);


printf(' Range = %.1f A \n',max(I)-min(I));

// The answers vary due to round off error
```

Scilab code Exa 1.32 To estimate the approximaate torque produced by the string

```
// Exa 1.32

clc;
clear;

// Given

E = 1.2 * 10^4 ; // Phosphor Young's Modulus (kg per
```

```
    mm^2)
l = 400; // Length of strip (mm)
w = 0.5; // Width of strip (mm)
t = 0.08; // Thickness of strip (mm)
Theta = 90; // In degrees


// Solution

T = (E*w*t^2)/(12*l);

printf('By using the torque formula having E as
   youngs modulus, we get T = %.3f kg-mm \n',T);

//The answer provided in the textbook is wrong
```

Scilab code Exa 1.33 To determine the deflection in degrees

```
// Exa 1.33

clc;
clear;

//Given

W = 0.005; // Weight in Kg
l = 2.4*10^-2; // Distance in m
Td = 1.05*10^-4; // Deflection torque in kg-m

// Solution

Theta = asind(Td/(W*l));
printf('Deflection torque is given by, Td = W*l*sin(
   theta)\n Therefore theta = %.1f degrees \n',
   Theta);
```

Scilab code Exa 1.34 To find the deflection in degrees for a current of 1A while t

```
// Exa 1.34

clc;
clear;

// Given

I1 = 10; // Current which produces deflection of 90 degrees
Theta1 = 90; // In degrees
I2 = 5; // Current for which theta is to be calculated

// Solution

//The deflection which produces a current of 1A when instrument is spring controlled
// Tc      theta
// theta      I^2

theta2 = (I2/I1)^2 * Theta1 ;
printf('The deflection which produces a current of 1A when instrument is spring controlled  is equal to = %.1f degrees \n',theta2);
//The deflection which produces a current of 1A when instrument is gravity controlled
// Tc      sin(theta)
// theta      I^2

theta2_gravity = asind((I2/I1)^2 *sind(Theta1)) ;
printf(' The deflection which produces a current of 1A when instrument is gravity controlled = %.2f
```

```
degrees \n',theta2_gravity);

// The value of I given as 1A in problem statement
   is incorrect to satisfy the problem answer(
   correct value is 5A)
```

Scilab code Exa 1.35 To find the effective resistance

```
// Exa 1.35

clc;
clear;

// Given

f = 450; // Resonating frequency in kHz
C = 250; // Capacitor value at resonating frequency
   (pf)
Q = 105; // Q-meter reading at resonance
Rsh = 0.75; // Value of shunt resistance in ohms

// Solution

L = 1/((2*%pi*f*10^3)^2*C*10^-12);  // in H
w=2*%pi*f*10^3;
R = (w*L)/Q - Rsh;
x= round(w*L/Q);
printf(' The value of resistance ,R = %.2f Ohms \n',
   double(w*L/Q)-Rsh);

// The answer vary due to round off error
```

Scilab code Exa 1.36 To determine the Cd and Q of the coil

```
// Exa 1.36

clc;
clear;

// Given

f1 = 1*10^6; // first resonant frequency in Hz
C1 = 480*10^-12; // Capacitor value at f1 in Farad
f2 = 2*10^6; // second resonant frequency in Hz
C2 = 120*10^-12; // Capacitor value at f2 in Farad
R = 10; // Resistance in Ohms

// Solution

Cd = (C1-4*C2)/3;  // Distributive capacitor
Q = 1/((2*%pi*f1)*R*(C1+Cd));

printf('The value of Cd and Q of the coil are %.1f
    pf and %.2f respectively',Cd*10^12,Q);

//The answer provided in the textbook cannot be
    confirmed(The formulae for Cd mentions L2
    variable whose value is not given in problem
    statement)
```

Chapter 2

Waveform Generators

Scilab code Exa 2.1 To determine the frequency of oscillations in a Wien bridge os

```
// Exa 2.1

clc;
clear;

// Given

// A wien bridge oscillator under consideration
R = 55*10^3; // Resistance in Ohms
// R = R1 = R2  ... given
C = 800*10^-12; // Capacitor in Farad
// C = C2 = C1  .. given

// Solution

f = 1/(2*%pi*R*C) ;

printf(' The frequency of oscillations = %.1f Hz \n'
    ,f);
```

Scilab code Exa 2.2 To design R and C elements of Wien bridge Oscillators

```
// Exa 2.2

clc;
clear;

// Given

// A wien bridge oscillator under consideration

fo= 10^6 ; // frequency of oscillations in Hz

// Solution

printf(' Let R = 3 k Ohms \n');

R = 3000; // Ohm's
// since , fo = 1/(2*%pi*R*C); therefore ,
C = 1/(2*%pi*fo*R);

printf(' Substituting that , the value of capacitor
    = %d pf \n',C*10^12);
```

Scilab code Exa 2.3 To find frequency of oscillations for an ordinary phase shift

```
// Exa 2.3

clc;
clear;

// Given
```

```
// A phase shift oscillator

R = 800*10^3; // in Ohm's
// R = R1 = R2 = R3   .. given
C = 100*10^-12; // in Farad
// C = C1 = C2 = C3   .. farad

// Solution

fo = 1/(2*%pi*R*C*sqrt(6));

printf(' The frequency of oscillations = %d Hz \n',
    fo);
```

Scilab code Exa 2.4 To calculate frequency of transistor Colpitts oscillator

```
// Exa 2.4

clc;
clear;

// Given

// Transistor Colpitts oscillator
L = 100*10^-3; // Inductance(H)
C1 = 0.005*10^-6; // Capacitor(F)
C2 = 0.01*10^-6; // Capacitor(F)

// Solution

C = C1*C2/(C1+C2);
printf(' By calculation , C = %.2f pf \n',C*10^12);
fo = 1/(2*%pi*sqrt(L*C));

```

```
printf('  The frequency of oscillator = %.1f kHz \n'
    ,fo*10^-3);

// The answer provided in the textbook is wrong
```

Scilab code Exa 2.5 To determine the oscillation frequency of a Hartley oscillator

```
// Exa 2.5

clc;
clear;

// Given

// A Hartley oscillator under consideration
L1 = 100*10^-3; // Inductance(H)
L2 = 1*10^-3; // Inductance(H)
M = 50*10^-3; // Inductance(H)
C = 100*10^-12; // Capacitor(F)

// Solution

L = L1+L2+2*M;
printf(' By calculation , L = %d H \n',L*10^3);

f = 1/(2*%pi*sqrt(L*C));

printf('  The frequency of oscillation = %d kHz \n',
    f*10^-3);

// The answer provided in the textbook is wrong
```

Scilab code Exa 2.6 To calculate the value of the feedback ratio

```
// Exa 2.6

clc;
clear;

// Given

// An Amilifier under consideration
Av = 40; // Voltage gain
Vi = 0.1; // Input voltage without feedback(V)
Vi_fb = 2.4; // Input voltage with feedback(V)

// Solution

A = Av*Vi_fb/Vi;

// Av = A/(1-B*A) ; therefore ,
B = (1-A/Av)/A;

printf(' The value of feedback ratio = %.6f \n ',B);
```

Chapter 3

Signal Analysers

Scilab code Exa 3.2 To determine the dynamic range of a spectrum analyser

```
// Exa 3.2

clc;
clear;

// Given

Noise = -90; // Minimum detectable signal (dbm)
Ip = 300 ; // power level of third-order product(dbm
    )

// Solution

printf(' The expression for the dynamic range of the
     spectrum analyser = 2/3*(Ip-MDS) \n  So, by
    calculations :-\n');

DR = 2/3*(Ip-Noise);
printf('  Dynamic range %.1f dB \n',DR);

// The answer provided in the textbook is wrong
```

Scilab code Exa 3.3 To calculate the minimum detectable signal of a spectrum analy

```
// Exa 3.3

clc;
clear;

// Given

NF = 30; // Noise figure in dB
BW = 1; // Bandwidth of 3 dB filter in kHz

// Solution

printf(' The noise level of the spectrum analyser is
    related to the noise figure and the IF bandwidth
    by the following equation - \n  MDS = -114 dbm +
    10*log(BW/1MHz) + NF \n  so, by calculation :- '
   );

MDS = -114 + 10*log10(BW*10^3/10^6)+NF;

printf(' MDS = %d  dBm \n ', MDS);

// The answer provided in the textbook is wrong
```

Chapter 4

Oscilloscopes

Scilab code Exa 4.1 Determine the percentage of non linearity in a saw tooth outpu

```
// Exa 4.1

clc;
clear;

// Given

// An oscilloscope

R = 400; // Resistance(k Ohms)
C = 0.025; // capacitance(micro Farad)
T = 0.4; // Time period of saw-tooth output waveform
   (msec)

// Solution

printf(' The percentage of non linearity i.e
   deviation in output can be given as t/(4*R*C)\n
   ');
PD = (T*10^-3)/(4*R*10^3*C*10^-6) ;

```

```
printf(' Therefore, by calculation, percent deviation = %d percent \n ',PD*100);
```

Scilab code Exa 4.2 To determine the rise and decay time of the saw tooth time bas

```
// Exa 4.2

clc;
clear;

// Given

f = 83.3 ; // frequency of  sinusoidal voltage in KHz

// Solution
// part a

printf('Being sunchronised, the frequency of the saw-tooth wave will be a submultiple of the signal. \n');

printf(' Frequency of saw-tooth curve = %.2f kHz \n',f/10);
F = f/10;
printf(' Period of the saw-tooth curve = %.1f microsec \n',(1/F)*10^3);

// since, Sine wave y = A sin theta
// but y/A = 0.5(since, end of trace was at position half the amplitide away from x-axis)
theta = asind(1/2) ;
printf(' The 10th wave is in short of a complete since wave by %d degrees \n',theta);
printf(' Therefore, No of full waves of sine form
```

```
seen on the screen are 9 11/12 waveforms \n');

// Rise time +decay time = period of wave = 120 microsec
T = 120 ; // period in microsec
 Rise_by_decay = (119/12) / (10- 119/12);
DecayTime = Rise_by_decay/T;
printf(' Decay time = %.1f microsec \n',round(DecayTime));
printf(' Rise time = %.1f microsec \n',T-DecayTime);

// part b

printf(' Since, increase time base frequency = 10/4 times the final value \n');

L = (10/4)* theta ;
printf(' Length of trace blanked in degrees due to flyback time = %d degrees \n ',L);
T_new = T*4/10;
printf('Period of new time base = %d microsec \n',T_new);
printf(' Rise time as per new time base = %d microsec \n',T_new-1);
```

Scilab code Exa 4.3 To calculate the velocity of the electron beam of an oscillosc

```
// Exa 4.3

clc;
clear;

// Given

Va = 2500; // Applied voltage(Volts)
```

```
e = 1.602*10^-19; // Charge of electron(C)
m = 9.107*10^-31; // Mass of electron(Kg)

// Solution

// For Electron beam in the oscilloscope , its
   velocity is given as−
V = sqrt(2*e*Va/m);

printf(' The velocity of electron beam of an
   oscilloscope = %.3f * 10^6 m/sec \n',V/10^6);
```

Scilab code Exa 4.4 To determine the unknown applied voltage

```
// Exa 4.4

clc;
clear;

// Given

Def_sensitivity = 0.05; // Deflection sensitivity in
   mm/V
Spot_deflection = 5; // in mm

// Solution

AppliedVoltage = Spot_deflection/Def_sensitivity ;

printf(' The applied voltage = %d V \n',
   AppliedVoltage);
```

Scilab code Exa 4.5 To determine the deflection sensitivity and the deflection fac

```
// Exa 4.5

clc;
clear;

// Given

// A CRT under consideration
l = 20; // length of x-deflection plates in mm
d = 5; // distance between x-deflection plates in mm
s = 250; // distance between screen and center of plate in mm
Va = 3000; // applied accelerating voltage in volts

// Solution

Def_sensitivity = l*s/(2*d*Va) ;
printf(' The deflection sensitivity = %.5f mm/V \n',Def_sensitivity);
printf('  The deflection factor = %.1f V/mm \n',1/Def_sensitivity);
```

Scilab code Exa 4.6 To find the Vrms of the sinusoidal voltage applied

```
// Exa 4.6

clc;
clear;

// Given

l = 25; // length of x-deflection plates in mm
d = 1; // distance between x-deflection plates in mm
s = 200; // distance between screen and centre of plate in mm
```

```
Va = 3000; // applied accelerating voltage in volts
Lt = 100; // length of trace in mm

// Solution

// Deflection produced = y/Vd = s*l/(2*d*Va)

y = 1/2 *(Lt);
// Therefore,
Vd = 2*d*Va*y/(l*s) ;

Vrms = Vd/sqrt(2) ;

printf(' The Vrms of the applied sinusoidal voltage = %.1f V \n',Vd);

Def_sensitivity = l*s/(2*d*Va) ;
printf('  The deflection sensitivity = %.5f mm/V \n',Def_sensitivity);

// The answer provided in the textbook is wrong
```

Scilab code Exa 4.7 To find the phase angle in each trace

```
// Exa 4.7

clc;
clear;

// Given

// Two sinusoidal voltage signals are applied to vertical and horizontal plates of CRO

// Solution
```

```
printf('Theta = asin(dvo/DV');
// Referring fig(a)
Theta_a = asind(0) ; // dvo = 0
printf(' Theta for trace shown in fig(a) = %d
    degrees\n',Theta_a);
// Referring fig(b)
Theta_b = asind(3/6) ; // dvo = 3 and DV =6
printf(' Theta for trace shown in fig(b) = %d
    degrees\n',Theta_b);
// Referring fig(c)
Theta_c = asind(1/1) ; // dvo = DV = 1
printf(' Theta for trace shown in fig(c) = %d
    degrees\n',Theta_c);
```

Scilab code Exa 4.8 To find the ratio of frequencies of vertical and horizontal si

```
// Exa 4.8

clc;
clear;

// Given

// Referring closed Lissajous pattern as shown in
    fig.
wx = 2; // no of positive x-peak
wy = 3; // no of positive y-peak

// Solution

fy_fx = wy/wx ;
printf(' Ratio of frequencies between vertical and
    horizontal signals = %.1f \n',fy_fx);
```

Scilab code Exa 4.9 To find the frequency of the vertical signal for a Lissajous p

```
// Exa 4.9

clc;
clear;

// Given

// Referring Lissajous pattern shown in figure
wx = 1 ; // Sum of x-peak pattern
wy = 2.5; // sum of y-peak pattern
fx = 3; // frequency of horizontal signal

X = wy/wx ; // X is ratio of fy/fx

// Therefore, fy = 2.5*fx

printf(' Frequency of vertical signal = %.1f kHz \n
    ',X*fx);

// The answer provided in the textbook is wrong
```

Chapter 5

Special Types of CROs

Scilab code Exa 5.3 To determine the RC value of each section

```
// Exa 5.3

clc;
clear;

// Given

fs = 10000; // frequency of modulated signal(Hz)
fm = 200*10^3; // modulation frequency(Hz)
Ri = 10; // Input resistance(ohms)
e2_by_e1 = 1.3; // limit for lowest frequency(in %)

// Solution

F_lower = fm - fs ;

printf(' For a double-section filter , \n  e2/e1 = 1/sqrt(1+(w*Rf*Cf)^2) \n');
// Therefore,

function y=f(x)
```

```
    y =(1/(sqrt(2*%pi*F_lower*x)^2+1))-e2_by_e1/100;
endfunction
[x,v,info] = fsolve(0,f);
printf('  The product of Rf*Cf = %.4f   sec \n ',x);
printf(' Let Rf = 10^5 Ohms, so that attenuation is
    10:1. Therefore , Cf = ');
Cf = x*10^-7;
printf(' %.3f pf \n ',Cf*10^12);

// The answer provided in the textbook is wrong
```

Scilab code Exa 5.4 Calculate the phase shift between V2 and V1

```
// Exa 5.4

clc;
clear;

// Given

// The Lissajous pattern
Y2 = 2.5; // slope of the major axis(in div)
Y1 = 1.2; // slope of the vertical axis(in div)

// Solution

printf(' The phase shift V2 and V1 can be given as
    sin(Theta) = Y1/Y2 \n  -where V1 and V2 are
    voltages applied to X and Y axis respectively \n
    ');

Theta = asind(Y1/Y2) ;
printf(' Since, the ellipse is lying in the I and
    the III quadrant, \n  The angle is theta or 360-
    theta , i.e, %.2f or %.2f \n',Theta,360-Theta);
```

Scilab code Exa 5.5 Determine the power loss for a capacitor

```
// Exa 5.5

clc;
clear;

// Given

S = 0.6*10; // sensitivity of oscillograph in V per cm
A = 2; // Area of oscilloscope area in cm^2
dx = 4; // x-axis deflection in cm
dy = 3; // y-axis deflection in cm

// solution

printf('Ref fig. 5.5(a and b) -If Ic leads Vc by 90 degree, C will be a lossless ideal capacitor, and it will have infinite resistance R. Therefore, Ic is leading Vc by <90 degree. Theta is loss of the capacitor \n ');
printf('Power factor = cos(theta) = 1  when theta = o degree) \n');

pf = 1;

Vcondenser = (1/sqrt(2)) * S * dx*200; // since one-two thousandth od C voltage is applied to the x-plates
Icondenser = (1/sqrt(2)) * S * 1/100000 ; // since Y-plates are impressed with voltage 100000 times the magniture of  condenser I.

```

```
Pcondenser = Vcondenser * Icondenser;

printf(' If p.f =1, the ellipse could have a major
   axis of %d cm and a minimum axis of %d cm \n',2*
   dx,2*dy);

printf(' Total area = %.2f cm^2 \n',%pi/4 * 2*dx*2*
   dy);

printf(' power loss of the capacitor = %.4f W \n',
   Pcondenser*A/(12/%pi));

// The answer provided in the textbook is wrong
```

Scilab code Exa 5.6 Determine the frequency of the vertical input

```
// Exa 5.6

clc;
clear;

// Given

// A stationary Lissajous pattern
Vy = 6 ; // max value on vertical axis
Vx = 5; // max value on horizontal axis
fx = 1500; // horizontal input frequency(Hz)

// Solution

// fy/fx = No of pts the target meets per bottom(x-
   axis)/No of pts the target meets per bottom(y-
   axis)

fy = (Vy/Vx)*fx;
```

```
printf('The frequency of vertical axis = %d Hz \n',
   fy);
```

Scilab code Exa 5.7 Determine the possible phase angles of V2 with repect to V1

```
// Exa 5.7

clc;
clear;

// Given

b2 =2.5 ; // Max no of divisions on y-axis
b1 = 1.25; // point of intersection on y-axis(div)

// Solution

printf(' Let theta be the phase angle of V2 w.r.t V1
    where V1 and V2 are the voltages applied to x
   and y axis respectively \n');
// Sin theta = b1/b2;
Theta = asind(b1/b2);

printf('  Therefore, the phase angle of V2 w.r.t V1
   = %d degrees \n. But another possible value is
   (360-theta) i.e. %d degrees \n',Theta,360-Theta);
```

Scilab code Exa 1.8 Estimate the boost required for electron gun accelerating volt

```
// Exa 5.1

clc;
```

```
clear;

// Given

E1 = 1/100; // exposure set for grid line impression (sec)
E2 = 10; // second exposure duration(sec)
R = 10^-4; // persistence of CRO screen(sec)
I1 = 1; // Trace intensity for exposure 1(candle power)
I2_normal = 4 ; // trace intensity for normal settings(candle power)

// Solution

printf(' The emission of light that would be received by photographic paper in both exposures must be the same \n  Also, the product of time and light is to be the same. \n');
I_req = I1*E1/R;
printf('  Hence, the image intensity required = %d \n' ,I_req );
I_boost = I_req/I2_normal;
printf('  Therefore, the intensity boost required = %d times \n' , I_boost);

printf('  The light emitted is proportional to the kinetic energy of the electron while it strikes the screen, which is equal to sqrt(V) , where V is the velocity while striking \n');
 printf('  Therefore, the accelerating voltage of the accelerating anode should br increased by %d times \n',sqrt(I_boost));
```

Scilab code Exa 5.8 Calculate the entering velocity of electrons and deflection pr

```
// Exa 5.8

clc;
clear;

// Given

Va = 2000; // Anode voltage(Volts)
Vd = 100; // Deflecting plates volage(Volts)
a =1.5*10^-2; // axial length in m
Sd= 30*10^-2; // screen distance in m
Ld = 5*10^-2; // deflecting plates length in m

// Solution

// Let,
 x = 1.76*10^11 ; // e/m ratio in c/kg
L = Sd + Ld/2 ;
D = (Ld*L*Vd)/(2*a*Va) ; // Deflection produced(m)
Vo = sqrt(2*x*Va); // velocity of electrons in m/kg


printf(' The deflection produced on screen = %.3f cm
    \n',D*100);
printf('  The velocity of the electrons when they
    enter the field of the deflecting plates = %.4f *
     10^7 m/kg \n', Vo/10^7);

// The answer provided in the textbook is wrong
```

Scilab code Exa 5.9 To find the error in the meter indication

```
// Exa 5.9

clc;
```

```
clear;

// Given

// A saw−tooth waveform is applied to an average
   diode voltmeter(Refer Fig. 5.24)

printf('For a saw−tooth waveform, the rms value = Vm
   /T  \n −where Vm js max voltage value and T being
    the time period \n');

printf(' Average value Va, 0.433 Vrms \n ');

printf('Similarly, Iav = 0.433∗Vrms/R \n');

printf(' Error = 100 ∗ (0.433∗Vrms − (0.433/R)∗0.45
   / (0.45∗(Vrms/R))) = −3.8 percent \n');
printf(' The meter reading is 3.8 percent less than
   the expected value \n');
```

Scilab code Exa 5.10 To calculate the velocity of the electron beam in the oscillo

```
// Exa 5.10

clc;
clear;

// Given

Va = 2000; // voltage applied to anodes(V)
l = 50*10^-3; // length of horizontal plates(m)
m = 9.1*10^-31; // mass of electron in kg
e = 1.6*10^-19; // velocity of electron in m/s
// Max transit time is T/4

```

```
// Solution

V = sqrt(2*Va*e/m);
Fc = V/(4*l);
printf(' The velocity of electron = %.3f * 10^8 m/s
   \n',V*10^-8);
printf('  The cutoff frequency = %.3f MHz \n',Fc
   /10^6);
```

Chapter 6

DC and AC Bridges

Scilab code Exa 6.1 To calculate the galvanometer deflection for imbalance in R4

```
// Exa 6.1

clc;
clear;

// Given data

// Refering bridge shown in fig. 6.8
R1 = 1000; // Ohms
R2 = 4000; // Ohms
R3 = 100; // Ohms
R4 = 400; // Ohms
Rg = 100; // Ohms
Si = 100; // Sensitivity in mm/microAmp
V = 3; // Voltage applied
R4_imbalance = 1; // resistance added in R4 to create imbalance

// Solution

printf('The bridge is originally in balance.
```

```
Therefore, R1/R3 = R2/R4 \n');
printf('Let there be imbalance in the bridge circuit because of increase in value of R4 value by 1 Ohm \n');
printf('Therefore, R4 = 400+X Ohms \n');
printf('Thevenins Resistance  Rth = (100*1000)/(100+1000) + (4000*(400+X))/(4400+X) \n'); // Rth = R1*R3/(R1+R3) + R2*R4/(R2+R4)
printf('Neglecting X \n');
// Therefore
Rth = R1*R3/(R1+R3) + R2*R4/(R2+R4);
printf('Rth becomes %d ohms \n',round(Rth));
printf('Eth = [R3/(R1+R3) + R4/(R2+R4)]*E; \n');
// Applying binomial expansion and neglecting X2 term, X is small
// Therefore
X = R4_imbalance;

Eth =  V*10*X/48400;
printf('Applying binomial expansion, Eth = %.2f  V \n',round(Eth*10^6));
Ig = Eth/(Rth+Rg); // Galvanometer current
D = Ig*Si; // Deflection in mm
printf('Galvanometer Current Ig = %.3f  A \n', Ig*10^6);
printf('Galvanometer deflection D = %.2f mm \n',D*10^6);

// The answer provided in the textbook is wrong
```

Scilab code Exa 6.2 To calculate the deflection of the galvanometer caused by the

```
// Exa 6.2

clc;
```

```
clear;

// Given

//Fig. 6.9 shows wheatstone bridge
R1 = 1000; // Ohms
R2 = 100; // Ohms
R3 = 400; // Ohms
Rx = 41; // Ohms(Unknown resistance)
V = 1.5; // Voltage supplied
Rg = 50; // Galvanometer resistance (ohms)
Si = 2; // current sensitivity in mm/microAmp


// Solution

Rth = (R1*R3/(R1+R3)) + R2*Rx/(R2+Rx);
Eth = V*(R3/(R1+R3) - Rx/(R2+Rx));
Ig = Eth/(Rth+Rg);
d = Ig*Si;
printf('The thevenins equivalent resistance = %.1f Ohms \n',round(Rth));
printf(' The thevenins equivalent voltage = %.1f mV \n',abs(Eth*10^3));
printf(' The current through the galvanometer = %.2f micro Amp \n',abs(Ig*10^6));
printf(' The deflection produced by the galvanometer caused by the imbalance in the circuit = %.2f mm \n',abs(d*10^6));
```

Scilab code Exa 6.3 To find the frequency for which the bridge is in balance and t

```
// Exa 6.3

clc;
```

```
clear;

// Given

//Fig 6.41 shows an AC bridge
R1 = 800; // Ohms
C1 = 0.4; // microFarad
R2 = 500; // Ohms
C2 = 1.0; // microFarad
R3 = 1200; // Ohms


// Solution

// Z = R + j X;
// Z1 = 800 + j/(w*C1)
// Y2 = 1/R2 - j*(w*C2)
//Z3 = 1200

printf('At balance, Z1/Z4 = Z2/Z3 \n');

printf(' Rearranging the equation, Z4 = Z1*Z3*Y2 \n');
printf(' Equating the real and imaginary parts on both sides, \n');
Z4 = R1*R3*1/R2;
w = sqrt(C1*C2);
printf(' The value of R in arm DA to produce a balance = %d ohms \n',Z4);
printf(' The value of frequency at balance = %.4f Hz \n',w/(2*%pi));

// The answers given in textbook for R and f are incorrect
```

Chapter 7

Transducers

Scilab code Exa 7.1 To determine the sensitivity of potentiometer

```
// Exa 7.1

clc;
clear;

// Given

NonLinearity = 1 ; // in percentage
P = 5; //Power rating in Watts
StepSize = 50; // in Ohms
Rmin = 10 ; // in Ohms
Rmax = 10000 ; // in Ohms

// Solution

printf('Max Error in linearity - Non-linearity = 1
    percent \n');
printf(' Therefore, Rp/Rm should be less than 0.1 \n
    ');
// If Rp/Rm < 0.1
// per_Error = 15 * (Rp/Rm)
```

```
// Therefore
Rp = (1/15)*Rmax;
printf(' If Rp/Rm < 0.1 \n Therefore we can choose a
    potentiometer with a total resistance Rp = %.2f
   Ohms at the maximum. Any value of Rp less than %
   .2f Ohms would be all right as far as the non-
   linearity is concerned \n',Rp,Rp);

printf(' However, lower the value of Rp lower will
   be the sensitivity. Therefore we choose 650 Ohms
   potentiometer from the family, which will have
   maximum sensitivity and at the same time have non
   -linearity less than 10 percent \n');
Rp_selected = 650; // Ohms

Max_Ecx = sqrt(P*Rp_selected);
s = Max_Ecx/360; //Sensitivity

printf(' The senstivity of potentiometer = %.2f V/
   degree \n',s);
```

Scilab code Exa 7.2 To determine the linear displacement

```
// Exa 7.2

clc;
clear;

// Given

l = 50; // length of potentiometer in mm
R = 5000; // Total resistance of potentiometer in
   Ohms
Rt = 1850; // Resistance of potentiometer in Ohms

```

```
// Solution

R_length = R/l ; // Resistance per unit length
R_normal = R/(l*10^-3*0.5);
printf(' Resistance of normal position = %d Ohms \n',R_normal);
R_change = R_normal - Rt;
printf(' Change in resistance = %d Ohms \n',R_change);
Displacement = R_change/R_length ;
printf(' The linear displacement when the resistance of the potentiometer is 1850 ohms = %.2f mm \n',Displacement);

// The answer provided in the textbook is wrong
```

Scilab code Exa 7.3 `To determine the potentiometer resolution`

```
// Exa 7.3

clc;
clear;

// Given

N = 50; // No of turns of potentiometer per mm
Number_of_Resolution = 4; // No of resolutions of potentiometer

// Solution

Resolution = 1/N;
printf(' Resolution of potentiometer = %.3f mm \n',Resolution);
printf(' 4 resolutions of potentiometer with one
```

```
    rotation = %.1f mm \n',10^3*Resolution/
    Number_of_Resolution);
```

Scilab code Exa 7.4 Determine the poissions ratio for thin circular wire

```
// Exa 7.4

clc;
clear;

// Given

G = 3.8; //  Gauge factor

// Solution

P = (G-1)/2;
printf(' Poissons ratio of thin circular/wire of
    soft iron = %.1f \n',P);
```

Scilab code Exa 7.5 Determine the gauge factor of the device

```
// Exa 7.5

clc;
clear;

// Given

L = 0.1 ; // Initial length of wire in m
R = 120; // Initial resistance of wire in ohms
delta_L =  0.1*10^-3;// change in length of wire in
    m
```

```
delta_R = 0.21; // change in resistance of wire in
    ohms

// Solution

e = delta_L/L;
G = (delta_R/R)/e;

printf(' The gauge factor of device = %.2f \n',G);
```

Scilab code Exa 7.6 To calculate the percentage change in resistance of strain gau

```
// Exa 7.6

clc;
clear;

// Given

S = 1400; // Stress in Kgf/cm^2
E = 2.1*10^6; // Youngs Modulus in Kgf/cm^2
G = 2; // Gauge factor

// Solution

e = S/E;
change_in_R = G*e;

printf(' Percentage change in resistance of strain
    gauge = %.3f \n',change_in_R*100);

// The answer provided in the textbook vary due to
    round off
```

Scilab code Exa 7.7 To find the poissions ratio for strain gauge

```
// Exa 7.7

clc;
clear;

// Given

Gf = 2 ; // Gauge factor of strain gauge
S = 1000; // Stress in kg/cm^2
E = 2*10^6; // Youngs Modulus in kg/cm^2

// Solution

e = S/E; // strain

dR_R = e*Gf; // change in resistance
 // Gf = 1+2u;
// Therefore
u = (Gf-1)/2; // poissons ratio

printf('The percentage change in resistance of
    strain gauge = %.1f \n',dR_R*100);
printf(' Poissons ratio  = %.2f \n',u);
```

Scilab code Exa 7.8 To determine the change in output voltage

```
// Exa 7.8

clc;
clear;
```

```
R = 200; // strain gauge resistance in Ohms
G = 2.5; // Gauge factor
RL = 400; // load resistance in Ohms
V = 24; // input voltage in volts
S = 140; // applied stress in mgf/m^2
Y = 200; // Modulus of elasticity in GN/m^2

// Solution

V_normal = V*(R/(R+RL));

printf('Voltage across strain gauge = %d V \n', V_normal);
e = (S*10^-3)/Y;
// Strain e = dell_L/L
//dell_R/R = G* dell_L/L;
// so,
dell_R = R*G*e;


//strain gauge under strained condition
V_strained = (R+dell_R) * V/(R+dell_R+RL);
printf(' Voltage across strain gauge under strained condition = %.4f ohms \n',V_strained);

dif = V_normal - V_strained;
printf(' Change in output voltage = %.2f mV \n',abs(dif*10^3));
```

Scilab code Exa 7.9 To determine the resistance at 75 degree celsius along with va

```
// Exa 7.9

clc;
```

```
clear;

// Given

// A platinum resitance thermometer
R1 = 120; // resistance in ohms at 25   c
T1 = 25; // temperature in   c
T2 = 75; // temperature in   c
Alpha_T = 0.00392; // temperature coefficient of
    resistance at 25  c
R3 = 180; // resistance in ohms at unknown temp T3

// Solution

R2 = R1*(1+Alpha_T*(T2-T1)); // resistance at 75   c
printf(' The resistance at 75   c = %.2f ohms \n',R2
    );

// now, to get T3 corresponding to R3= 180 ohms

// R3 = R2*(1+Alpha_T*(T3-T1));
// Rearranging above equation to get T3 as
T3 = (R3/R1 -1)/Alpha_T + T1;

printf(' The temperature corresponding to resistance
     180 ohms = %.2f   c \n',T3);
```

Scilab code Exa 7.10 To determine the limiting value of resistance k

```
// Exa 7.10

clc;
clear;

// Given
```

```
// A copper resistance thermometer

R1 = 15; // resistance in ohms at 20  c
T1 = 20; // temperature in  c
T2 = 175; // max temperature in  c
Alpha_T = 0.00425; // temperature coefficient of resistance at 25 c

// Solution

R2 = R1*(1+Alpha_T*(T2-T1)); // resistance at 175 c
printf(' The limiting value of resistance  = %.2f ohms \n',R2);
```

Scilab code Exa 7.11 `To determine the resistance of the thermistor at 40 degree ce`

```
// Exa 7.11

clc;
clear;

// Given

// A thermistor
R1 = 120; // resistance in ohms at 25  c
T1 = 25; // temperature in  c
T2 = 40; // temperature in  c
Alpha_T = -0.05; // temperature coefficient of resistance over range 25-50 c

// Solution

R2 = R1*(1+Alpha_T*(T2-T1)); // resistance at 175
```

```
c
printf(' The resistance of thermistor at 40  c = %d
  ohms \n',R2);
```

Scilab code Exa 7.12 Determine the inductance of the coil

```
// Exa 7.12

clc;
clear;

// Given

// A variable inductive transducer
L1 = 2.5; // inductance in mH
N1 = 50; // No of effective turns at L1
N2 = 52; // No of effective turns at L2

// Solution

printf(' Since L directly proportional to N^2 \n');
printf('  L1/N1^2 = L2/N2^2 \n ');
printf(' Therefore , L2 i.e, \n ');

L2 = L1* (N2/N1)^2;
printf(' The inductance of coil when the effective
   turns of the coil are 52 = %.2f mH \n',L2);
```

Scilab code Exa 7.13 Determine the inductance of the coil for a variable reluctanc

```
// Exa 7.13

clc;
```

```
clear;

// Given

// A variable reluctance-type inductive transducer
L1 = 5; // Inductance of transducer in mH
lg1 = 1.5; // Length of iron piece in mm
d = 0.025; // Distance by which irno piece is moved
    towards electro magnet (mm)

// Solution

air_gap = lg1-d;
printf(' Length of air gap = %.3f mm \n',air_gap);
New_Inductance = L1 + lg1/air_gap;

printf('  The coil inductance becomes = %.2f mH \n',
    New_Inductance);

// The answer provided in the textbook is wrong
```

Scilab code Exa 7.14 Determine the sensitivity of LVDT

```
// Exa 7.14

clc;
clear;

// Given

// An LVDT
vo = 2.6; // Output voltage(volts) of LVDT
d = 0.4; // displacement in mm

// Solution
```

```

printf(' The sensitivity s = RMS value of output
    voltage/Displacement \n');

S = vo/d; // sensitivity

printf('  Therefore, s = %.1f V/mm \n',S);
```

Scilab code Exa 7.15 Determine linearity at given load

```
// Exa 7.15

clc;
clear;

// Given

// An LVDT
Vo = 1.25; // Output voltage
Dmax  = 0.0025;// max. deviation of linearity
L = 0.75; // weight of load in kgf

// Solution

Linearity = (Dmax/Vo)*100;
printf(' The linearity at a given load 0.65/kgf = %
    .1f percent \n',Linearity);
```

Scilab code Exa 7.16 Determine output voltage for a displacement of 8 mm from cent

```
// Exa 7.16

clc;
```

```
clear;

// Given

// An LVDT
vo = 5; // secondary voltage(volts) of LVDT
d = 12.5; // displacement in mm
d0 = 8; // displacement from central position in mm

// Solution

printf(' The sensitivity s = RMS value of output
    voltage/Displacement \n');

S = vo/d; // sensitivity

printf('  Therefore, s = %.1f V/mm \n',S);

printf('  Output voltage for a displacement of 8mm
    from its central position =  %.1f V \n',S*d0);
```

Scilab code Exa 7.17 Determine the sensitivity of LVDT and the pressure

```
// Exa 7.17

clc;
clear;

// Given

// An LVDT to measure deflection of bellows
S1 = 40; // sensitivity in V/mm
d = 0.125; // displacement in mm
P1 = 0.8*10^6; // pressure in N/m^2
Vo2 = 3.5 ; // Output of LVDT for pressure P2
```

```
// Solution

// output voltage for the pressure p1
Vo1 = S1*d; // in volts

L_senstivity = Vo1/P1;

// For P2 calculations when V = 3.5
P2 = Vo2/L_senstivity;

printf('The sensitivity of LVDT and pressure when
    the output voltage of LVDT is 3.5 V \n are  %.2f
    * 10^-6 V/N/m^2 and %.1f * 10^5 N/m^2
    respectively \n',L_senstivity*10^6,P2*10^-5);
```

Scilab code Exa 7.18 Determine the displacement for a capactive transducer

```
// Exa 17.18

clc;
clear;

// Given

// Capacitive Transducer
d = 0.05; // plate separation in mm
C = 5*10^-12; // Capacitence in farad
dell_C = 0.75*10^-12; // change in capacitence in
    farad

// Solution

// C = e*A/d;
eA = C*d;
```

```
//Now,
dell_x = eA/dell_C;
printf('The displacment that caused a change in
   capacitence is %.3f mm \n',dell_x);
```

Scilab code Exa 7.19 To determine the value of the capacitance

```

// Exa 17.19

clc;
clear;

// Given

// A Capacitive Transducer
d = 2.5; // plate separation in mm
A = 600; // Area (in mm^2)
P = 8*10^5; // Pressure applied in N/m^2
x = 0.5; // deflection produced in mm
C = 400*10^-12; // Capacitance in farad

// Solution

// Since, C = e*A/d
 e =C*d/A;

printf('Since we have to find capacitance when no
   pressure is applied. At that time plate
   separation = %d mm \n', d-x);

d1 = d-x; // plate separation(mm) after pressure
   applied
C1 = e*A/d1;
```

```

printf(' The value of capacitance, C with d = 2mm = %d micro farad \n', C1*10^12);
```

Scilab code Exa 7.20 Calculate the sensitivity and separation between the plates

```
// Exa 7.20

clc;
clear;

// Given

// A Capacitence Transducer
A = 5*10^-4; // Area in m^2
C = 9.5*10^-12; // Capacitence in farad
er = 81; //  Relative dielectric constant
e0 = 8.854*10^-12; // Absolute dielectric constant in F/m

// Solution

// C = e0*er*A/d;
// Therefore
d = e0*er*A/C;
printf('The plate separation d = %.2f mm \n',d*10^3);
S = e0*er*A/d^2;

printf(' Sensitivity s = %.3f * 10^-8 F/m \n',S*10^8);
```

Scilab code Exa 7.21 Determine the sensitivity of the arrangement

```
// Exa 7.21

clc;
clear;

// Given

// A 5-plate transducer
n = 5; // no of plates
l = 20*10^-3; // length of plate in m
b = 20*10^-3; // breadth of plate in m
d = 0.25*10^-3; // separation between plates in m

// Solution

A = l*b; // Area in mm^2
er = 1; //  Relative dielectric constant
e0 = 8.854*10^-12; // Absolute dielectric constant
    in F/m

S = (n-1)*e0*er*A/d^2;

printf('Sensitivity of the arrangement = %.3f *
    10^-9 F/m \n',S*10^9);

// The answer provided in the textbook is wrong
```