Scilab Textbook Companion for Electric Machines by C. I. Hubert[1]

Created by
Roohum Jegan
Electrical Machines
Electrical Engineering
PHCET Mumbai University
College Teacher
None
Cross-Checked by
Reshma

April 11, 2016

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

Book Description

Title: Electric Machines

Author: C. I. Hubert

Publisher: Pearson, India

Edition: 2

Year: 2014

ISBN: 978-81-317-0802-6

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes 4

1 Magnetics Electromagnetic Forces Generated Voltage and Energy Conversion 6

2 Transformer Principles 17

3 Transformer Connections Operation and Specialty Transformers 45

4 Principles of Three Phase Induction Motors 65

5 Classification Performance Applications and Operation of Three Phase Induction Machines 77

6 Single Phase Induction Motors 121

7 Speciality Machines 132

8 Synchronous Motors 137

9 Synchronous Generators 150

10 Principles of Direct Current Machines 173

11 Direct Current Motor Characteristics and Applications 192

12 Direct Current Generator Characteristics and Operation 207

List of Scilab Codes

Exa 1.2	Computation of Current in the coil and Magnetic potential difference across R3 and Flux in R2	6
Exa 1.3	Computation of hysteresis loss if the apparatus is connected to a 60 Hz source	8
Exa 1.4	Computation of magnitude of the developed torque . .	9
Exa 1.5	Computation of length of conductor	11
Exa 1.6	Computation of Frequency and Pole flux	12
Exa 1.7	Computation of eddy current loss if the apparatus is connected to a 60 Hz source	13
Exa 1.8	Computation of Number of cycles per revolution and Number of electrical degrees per revolution and Frequency in hertz	14
Exa 1.9	Computation of Frequency of the generated emf and Speed of the rotor	15
Exa 2.1	Computation of peak value of sinusoidal flux in a transformer	17
Exa 2.2	Computation of Turns ratio and Number of turns in each winding and Magnetizing current	19
Exa 2.3	Computation of Exciting current and its quadrature components and Equalizing magnetic reactance and equivalent core loss resistance and Magnetizing current and repeat the calculations for the transformer in the step up mode	20
Exa 2.4	Computation of Secondary voltage and Load current and Input current to the primary and Input impedance looking into the primary terminals	22

Exa 2.5	Computation of Equivalent impedance of the transformer referred to the high side and Input impedance of the combined transformer and load and Actual input voltage at the high side and Input impedance if the load is disconnected and Exciting current	25
Exa 2.6	Computation of a Equivalent input impedance of the transformer and load combination and Primary current when 2400V is supplied to primary and Voltage across the load	28
Exa 2.8	Computation of Percent impedance and Rated high side current and Equivalent resistance and reactance referred to the high side and High side fault current	31
Exa 2.9	Computation of Transformer regulation and Secondary voltage when the load is disconnected and Input primary voltage	33
Exa 2.10	Computation of Transformer regulation and Secondary voltage when the load is disconnected and Input primary voltage	35
Exa 2.11	Computation of transformer regulation	37
Exa 2.12	Computation of Core loss and Core loss if operated at rated current and reduced power factor from 375V 50 HZ supply and Overall efficiency and Efficiency if the load is disconnected	38
Exa 2.13	Determine Efficiency at rated load and 80 percent power factor and 70 percent load and 80 percent power factor	40
Exa 2.14	Determine Magnetizing reactance and equivalent core loss resistance and Per unit resistance reactance and impedance of transformer windings and Voltage regulation	42
Exa 3.1	Computation of Load current and Incoming line current and Transformed current and Apparent power conducted and apparent power transformed	45
Exa 3.2	Computation of Rated primary and secondary currents and Apparent power rating	47
Exa 3.3	Computation of Buck boost transformer parameters and Repeating the same assuming utilization voltage as 246V	48

Exa 3.4 Determine Circulating current in the paralleled secondaries and Circulating current as a percent of the rated current of transformer A and Percent difference in secondary voltage 50
Exa 3.5 Determine Rated high side current of each transformer and Percent of the total bank current drawn by each transformer and Maximum load that can be handled by the bank without overloading by one of the transformer 53
Exa 3.6 Determine the percent of the total bank current drawn by each transformer 57
Exa 3.7 Computation of Bank ratio and Transformer ratio and Rated line and phase currents for the high side and Rated line and phase currents for the low side 60
Exa 3.8 Determine the maximum allowable power that the open delta bank handle without overheating 62
Exa 3.9 Determine the minimum power rating required for each transformer 63
Exa 4.1 Computation of synchronous speed of a six pole induction motor 65
Exa 4.2 Computation of Frequency and Induced voltage of six pole induction motor 66
Exa 4.3 Determine Synchronous speed and Slip and Rotor impedance and Rotor current and Rotor current by changing the shaft load and Speed 68
Exa 4.4 Determine Total three phase apparent power crossing the air gap and Active and reactive components and Rotor power factor 70
Exa 4.5 Computation of Shaft speed and Mechanical power developed and Developed torque 73
Exa 4.6 Determine Power input and Total losses and Air gap power and Shaft speed and Power factor and Combined windage friction and stray load loss and Shaft torque . 74
Exa 5.1 Computation of minimum value of Locked rotor torque and Breakdown torque and Pull up torque 77

Exa 5.2 Determine Slip and Line current and Apparent power active power reactive power and power factor of the motor and Equivalent rotor current and Stator copper loss and Rotor copper loss and Core loss and Air gap power and Mechanical power developed 79
Exa 5.3 Computation of Speed at which maximum torque is developed and Maximum torque that the machine can develop and Rated shaft torque 85
Exa 5.4 Computation of Amount of torque that must be removed from the motor shaft to maintain 1760 rpm and Expected minimum starting torque for the lower voltage and Percent change in developed torque 86
Exa 5.5 Computation of minimum value of Shaft speed and Rotor current referred to the stator 88
Exa 5.6 Determine new operating speed of a system and New shaft horsepower 89
Exa 5.7 Determine expected locked rotor line current 91
Exa 5.8 Determine expected minimum locked rotor torque and Repeat when voltage and frequency dropped to 230V and 58Hz 92
Exa 5.9 Determine shaft rpm and Slip 94
Exa 5.10 Determine range of rotor speed and Required rheostat resistance 96
Exa 5.11 Determine rotor frequency and Slip at which TDmax occurs and Rotor speed at half rated torque load and Required rheostat resistance and Rated torque 97
Exa 5.12 Determine the percent increase or decrease in rotor circuit resistance 100
Exa 5.13 Determine the expected in rush current 101
Exa 5.14 Determine percent voltage unbalance and Expected approximate temperature rise if operating at rated load in a 40 deg ambient and Expected insulation life and Required derating of motor to prevent shortening insulation life 102
Exa 5.15 Determine the machine parameters in ohms 105
Exa 5.16 Determine R1 R2 X1 X2 XM and the combined core friction and windage loss and Express the no load current as a percent of rated current 106

Exa 5.17 Determine the active power that the motor driven as an induction generator delivers to the system 109
Exa 5.18 Computation of Locked rotor torque and the expected average in rush current and Repeat assuming motor is started at reduced voltage with 65 percent tap and In rush line current line current 112
Exa 5.19 Computation of Locked rotor current per phase and minimum locked rotor torque when starting and Locked rotor current per phase when motor is delta connected and Code letter 115
Exa 5.20 Computation of Resistance of the resistors required to limit the locked rotor current to 3 times rated current and Stator voltage per phase at locked rotor and Expected minimum locked rotor torque when starting as a percent of rated torque 117
Exa 5.21 Computation of Inductance and voltage rating of each series connected inductor required to limit the starting current to approximately 2 times Irated 119
Exa 6.1 Determine Locked rotor current in each winding and Phase displacement angle between the two currents and Locked rotor torque in terms of the machine constant and External resistance required in series with the auxiliary winding and Locked rotor torque 121
Exa 6.2 Determine Capacitance required in series with the auxiliary winding and Locked rotor torque 125
Exa 6.3 Determine NEMA standard horsepower rating of machine and Required running capacitance and Additional capacitance required for starting 128
Exa 6.4 Computation of Motor line current and motor phase current and Motor line current and motor phase current if one line opens and Line and phase currents if the power factor is 82 percent 129
Exa 7.1 Determine Torque load on the shaft and Torque angle if the voltage drops to 224V 132
Exa 7.2 Determine Resolution and Number of steps required for the rotor and Shaft speed 133
Exa 7.3 Determine Synchronous speed and Rail speed 135

Exa 8.1 Determine Developed torque and Armature current and Excitation voltage and Power angle and Maximum torque 137
Exa 8.2 Determine minimum value of excitation and Repeat calculations using eq 8 16 and 8 21 and Power angle . . 140
Exa 8.3 Determine System active power and Power factor of the synchronous motor and System power factor and Percent change in synchronous field current and Power angle of the synchronous motor 142
Exa 8.4 Determine Developed torque and Developed torque in percent of rated torque 147
Exa 9.1 Determine Turbine torque supplied to the alternator and Excitation voltage and Active and reactive components of apparent power and Power factor and Excitation voltage 150
Exa 9.2 Determine Speed regulation and Governor drop 153
Exa 9.3 Determine Frequency of generator A and Frequency of generator B and Frequency of bus 154
Exa 9.4 Determine Operating frequency and Load carried by each machine 156
Exa 9.5 Determine Bus frequency and Load on each machine . 158
Exa 9.6 Determine System kilowatts and System frequency and kilowatt loads carried by each machine 159
Exa 9.7 Determine Active and reactive components of the bus load and Determine the reactive power supplied by each machine 161
Exa 9.8 Computation of per unit impedance of a generator . . 163
Exa 9.9 Determine Excitation voltage and Power angle and No load voltage and Voltage regulation and No load voltage if the field current is reduced to 80 percent 165
Exa 9.10 Determine Excitation voltage and Power angle and No load voltage and Voltage regulation and No load voltage if the field current is reduced to 80 percent 167
Exa 9.11 Determine Equivalent armature resistance and Synchronous reactance and Short circuit ratio 170
Exa 10.1 Computation of Induced emf and Frequency of the rectangular voltage wave in the armature winding 173

Exa 10.2 Computation of rheostat setting required to obtain an induced emf of 290 V 174
Exa 10.3 Computation of no load voltage 176
Exa 10.4 Computation of percentage reduction in field flux required to obtain a speed of 1650 rpm 177
Exa 10.5 Computation of no load speed 179
Exa 10.6 Computation of Induced emf 179
Exa 10.7 Computation of cemf 181
Exa 10.8 Computation of new armature current 182
Exa 10.9 Computation of Steady state armature current and Steady state speed 184
Exa 10.10 Computation of Mechanical power developed and Torque developed and Shaft torque 185
Exa 10.11 Determine Electrical losses and Rotational losses and Efficiency 187
Exa 10.12 Determine Rated torque and Armature current at locked rotor and External resistance required in the armature circuit and Determine the locked rotor torque 189
Exa 11.1 Computation of Armature current when operating at rated conditions and Resistance and power rating of an external resistance required 192
Exa 11.2 Computation of Shunt field current and Armature current and Developed torque and Armature current and External resistance required 194
Exa 11.3 Computation of Speed if the load is reduced to a value that causes the armature current to be 30 percent of the rated current 197
Exa 11.4 Computation of resistance using linear approximation and values are compared with results obtained in previous example 199
Exa 11.5 Computation using linear approximation to show the gross error that occurs when a linear assumption is applied to compound motors operating at overload conditions 200
Exa 11.6 Determine Torque developed when operating at rated speed and Developed torque required at half rated speed and Armature voltage required for half rated speed . . 202

Exa 11.7 Computation of the resistance of a dynamic braking resistor that will be capable of developing 500 lb ft of braking torque at a speed of 1000 rpm 204
Exa 12.1 Determine Field circuit resistance and Field rheostat setting and Armature voltage and Field rheostat setting that will cause critical resistance and Armature voltage 207
Exa 12.2 Computation of No load voltage and Voltage regulation and Resistance setting of rheostat necessary to obtain rated voltage 210
Exa 12.3 Computation of Induced emf at rated load and No load voltage and Voltage regulation 212
Exa 12.4 Computation of Required resistance of a noninductive diverter that will bypass 27 percent of the total armature current and Power rating of the diverter 214
Exa 12.5 Computation of New bus voltage and Current supplied by each generator 215
Exa 12.6 Determine the increment increase in load on each machine if an additional 400 A load is connected to the bus and Current carried by each machine 217

List of Figures

1.1 Computation of Current in the coil and Magnetic potential difference across R3 and Flux in R2 6
1.2 Computation of hysteresis loss if the apparatus is connected to a 60 Hz source 9
1.3 Computation of magnitude of the developed torque 9
1.4 Computation of length of conductor 10
1.5 Computation of Frequency and Pole flux 11
1.6 Computation of eddy current loss if the apparatus is connected to a 60 Hz source 13
1.7 Computation of Number of cycles per revolution and Number of electrical degrees per revolution and Frequency in hertz . 14
1.8 Computation of Frequency of the generated emf and Speed of the rotor 15

2.1 Computation of peak value of sinusoidal flux in a transformer 17
2.2 Computation of Turns ratio and Number of turns in each winding and Magnetizing current 18
2.3 Computation of Exciting current and its quadrature components and Equalizing magnetic reactance and equivalent core loss resistance and Magnetizing current and repeat the calculations for the transformer in the step up mode 20
2.4 Computation of Secondary voltage and Load current and Input current to the primary and Input impedance looking into the primary terminals 23
2.5 Computation of Equivalent impedance of the transformer referred to the high side and Input impedance of the combined transformer and load and Actual input voltage at the high side and Input impedance if the load is disconnected and Exciting current 25

2.6 Computation of a Equivalent input impedance of the transformer and load combination and Primary current when 2400V is supplied to primary and Voltage across the load 29
2.7 Computation of Percent impedance and Rated high side current and Equivalent resistance and reactance referred to the high side and High side fault current 31
2.8 Computation of Transformer regulation and Secondary voltage when the load is disconnected and Input primary voltage . . 34
2.9 Computation of Transformer regulation and Secondary voltage when the load is disconnected and Input primary voltage . . 35
2.10 Computation of transformer regulation 37
2.11 Computation of Core loss and Core loss if operated at rated current and reduced power factor from 375V 50 HZ supply and Overall efficiency and Efficiency if the load is disconnected 39
2.12 Determine Efficiency at rated load and 80 percent power factor and 70 percent load and 80 percent power factor 41
2.13 Determine Magnetizing reactance and equivalent core loss resistance and Per unit resistance reactance and impedance of transformer windings and Voltage regulation 42

3.1 Computation of Load current and Incoming line current and Transformed current and Apparent power conducted and apparent power transformed 45
3.2 Computation of Rated primary and secondary currents and Apparent power rating 47
3.3 Computation of Buck boost transformer parameters and Repeating the same assuming utilization voltage as 246V . . . 49
3.4 Determine Circulating current in the paralleled secondaries and Circulating current as a percent of the rated current of transformer A and Percent difference in secondary voltage . 50
3.5 Determine Rated high side current of each transformer and Percent of the total bank current drawn by each transformer and Maximum load that can be handled by the bank without overloading by one of the transformer 53
3.6 Determine the percent of the total bank current drawn by each transformer 57

3.7 Computation of Bank ratio and Transformer ratio and Rated line and phase currents for the high side and Rated line and phase currents for the low side 60
3.8 Determine the maximum allowable power that the open delta bank handle without overheating 62
3.9 Determine the minimum power rating required for each transformer 63

4.1 Computation of synchronous speed of a six pole induction motor 66
4.2 Computation of Frequency and Induced voltage of six pole induction motor 66
4.3 Determine Synchronous speed and Slip and Rotor impedance and Rotor current and Rotor current by changing the shaft load and Speed 68
4.4 Determine Total three phase apparent power crossing the air gap and Active and reactive components and Rotor power factor 71
4.5 Computation of Shaft speed and Mechanical power developed and Developed torque 72
4.6 Determine Power input and Total losses and Air gap power and Shaft speed and Power factor and Combined windage friction and stray load loss and Shaft torque 74

5.1 Computation of minimum value of Locked rotor torque and Breakdown torque and Pull up torque 78
5.2 Determine Slip and Line current and Apparent power active power reactive power and power factor of the motor and Equivalent rotor current and Stator copper loss and Rotor copper loss and Core loss and Air gap power and Mechanical power developed 79
5.3 Computation of Speed at which maximum torque is developed and Maximum torque that the machine can develop and Rated shaft torque 84
5.4 Computation of Amount of torque that must be removed from the motor shaft to maintain 1760 rpm and Expected minimum starting torque for the lower voltage and Percent change in developed torque 86

5.5 Computation of minimum value of Shaft speed and Rotor current referred to the stator 88
5.6 Determine new operating speed of a system and New shaft horsepower 90
5.7 Determine expected locked rotor line current 92
5.8 Determine expected minimum locked rotor torque and Repeat when voltage and frequency dropped to 230V and 58Hz . . 93
5.9 Determine shaft rpm and Slip 94
5.10 Determine range of rotor speed and Required rheostat resistance 96
5.11 Determine rotor frequency and Slip at which TDmax occurs and Rotor speed at half rated torque load and Required rheostat resistance and Rated torque 98
5.12 Determine the percent increase or decrease in rotor circuit resistance 100
5.13 Determine the expected in rush current 101
5.14 Determine percent voltage unbalance and Expected approximate temperature rise if operating at rated load in a 40 deg ambient and Expected insulation life and Required derating of motor to prevent shortening insulation life 103
5.15 Determine the machine parameters in ohms 105
5.16 Determine R1 R2 X1 X2 XM and the combined core friction and windage loss and Express the no load current as a percent of rated current 107
5.17 Determine the active power that the motor driven as an induction generator delivers to the system 109
5.18 Computation of Locked rotor torque and the expected average in rush current and Repeat assuming motor is started at reduced voltage with 65 percent tap and In rush line current line current 113
5.19 Computation of Locked rotor current per phase and minimum locked rotor torque when starting and Locked rotor current per phase when motor is delta connected and Code letter . 115
5.20 Computation of Resistance of the resistors required to limit the locked rotor current to 3 times rated current and Stator voltage per phase at locked rotor and Expected minimum locked rotor torque when starting as a percent of rated torque 117

5.21 Computation of Inductance and voltage rating of each series connected inductor required to limit the starting current to approximately 2 times Irated 119

6.1 Determine Locked rotor current in each winding and Phase displacement angle between the two currents and Locked rotor torque in terms of the machine constant and External resistance required in series with the auxiliary winding and Locked rotor torque 122
6.2 Determine Capacitance required in series with the auxiliary winding and Locked rotor torque 125
6.3 Determine NEMA standard horsepower rating of machine and Required running capacitance and Additional capacitance required for starting 128
6.4 Computation of Motor line current and motor phase current and Motor line current and motor phase current if one line opens and Line and phase currents if the power factor is 82 percent 130

7.1 Determine Torque load on the shaft and Torque angle if the voltage drops to 224V 133
7.2 Determine Resolution and Number of steps required for the rotor and Shaft speed 134
7.3 Determine Synchronous speed and Rail speed 135

8.1 Determine Developed torque and Armature current and Excitation voltage and Power angle and Maximum torque 138
8.2 Determine minimum value of excitation and Repeat calculations using eq 8 16 and 8 21 and Power angle 140
8.3 Determine System active power and Power factor of the synchronous motor and System power factor and Percent change in synchronous field current and Power angle of the synchronous motor 143
8.4 Determine Developed torque and Developed torque in percent of rated torque 147

9.1 Determine Turbine torque supplied to the alternator and Excitation voltage and Active and reactive components of apparent power and Power factor and Excitation voltage 151

9.2 Determine Speed regulation and Governor drop 154
9.3 Determine Frequency of generator A and Frequency of generator B and Frequency of bus 155
9.4 Determine Operating frequency and Load carried by each machine 156
9.5 Determine Bus frequency and Load on each machine 158
9.6 Determine System kilowatts and System frequency and kilowatt loads carried by each machine 160
9.7 Determine Active and reactive components of the bus load and Determine the reactive power supplied by each machine . . . 162
9.8 Computation of per unit impedance of a generator 163
9.9 Determine Excitation voltage and Power angle and No load voltage and Voltage regulation and No load voltage if the field current is reduced to 80 percent 165
9.10 Determine Excitation voltage and Power angle and No load voltage and Voltage regulation and No load voltage if the field current is reduced to 80 percent 168
9.11 Determine Equivalent armature resistance and Synchronous reactance and Short circuit ratio 171

10.1 Computation of Induced emf and Frequency of the rectangular voltage wave in the armature winding 174
10.2 Computation of rheostat setting required to obtain an induced emf of 290 V 175
10.3 Computation of no load voltage 176
10.4 Computation of percentage reduction in field flux required to obtain a speed of 1650 rpm 177
10.5 Computation of no load speed 178
10.6 Computation of Induced emf 180
10.7 Computation of cemf 181
10.8 Computation of new armature current 182
10.9 Computation of Steady state armature current and Steady state speed 183
10.10 Computation of Mechanical power developed and Torque developed and Shaft torque 185
10.11 Determine Electrical losses and Rotational losses and Efficiency 187

10.12 Determine Rated torque and Armature current at locked rotor and External resistance required in the armature circuit and Determine the locked rotor torque 189

11.1 Computation of Armature current when operating at rated conditions and Resistance and power rating of an external resistance required 193
11.2 Computation of Shunt field current and Armature current and Developed torque and Armature current and External resistance required 195
11.3 Computation of Speed if the load is reduced to a value that causes the armature current to be 30 percent of the rated current 197
11.4 Computation of resistance using linear approximation and values are compared with results obtained in previous example 199
11.5 Computation using linear approximation to show the gross error that occurs when a linear assumption is applied to compound motors operating at overload conditions 201
11.6 Determine Torque developed when operating at rated speed and Developed torque required at half rated speed and Armature voltage required for half rated speed 203
11.7 Computation of the resistance of a dynamic braking resistor that will be capable of developing 500 lb ft of braking torque at a speed of 1000 rpm 205

12.1 Determine Field circuit resistance and Field rheostat setting and Armature voltage and Field rheostat setting that will cause critical resistance and Armature voltage 208
12.2 Computation of No load voltage and Voltage regulation and Resistance setting of rheostat necessary to obtain rated voltage 210
12.3 Computation of Induced emf at rated load and No load voltage and Voltage regulation 212
12.4 Computation of Required resistance of a noninductive diverter that will bypass 27 percent of the total armature current and Power rating of the diverter 214
12.5 Computation of New bus voltage and Current supplied by each generator 216

12.6 Determine the increment increase in load on each machine if an additional 400 A load is connected to the bus and Current carried by each machine 218

Chapter 1

Magnetics Electromagnetic Forces Generated Voltage and Energy Conversion

Scilab code Exa 1.2 Computation of Current in the coil and Magnetic potential difference across R3 and Flux in R2

```
// Example 1.2
```

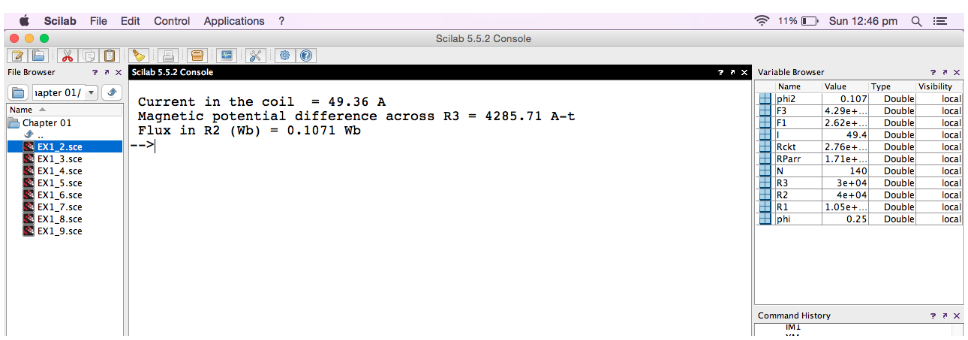

Figure 1.1: Computation of Current in the coil and Magnetic potential difference across R3 and Flux in R2

```
// Computation of (a) Current in the coil (b) Magnetic potential difference across R3
// (c) Flux in R2
//Page No. 13

clc;
clear all;
close;

// Given data
phi=0.250;                    // Flux in Wb
R1=10500;                     // First magnetic circuit parameter
R2=40000;                     // Second magnetic circuit parameter
R3=30000;                     // Third magnetic circuit parameter
N=140;                        // Number of turns of copper wire

// (a) Current in the coil
RParr=(R2*R3)/(R2+R3);          // Parallel resistance
Rckt=R1+RParr;                  // Circuit resistance
I=(phi*Rckt)/N;

// (b) Magnetic potential difference across R3
F1=phi*R1;                      // Magnetic drop across R1
F3=(I*N)-F1;                    // Flux across R3

//(c) flux in R2
phi2=F3/R2;


//Display result on command window
printf("\n Current in the coil  = %0.2f A ",I);
printf("\n Magnetic potential difference across R3 = %0.2f A-t  ",F3);
```

```
printf("\n Flux in R2 (Wb) = %0.4f Wb ",phi2);
```

Scilab code Exa 1.3 Computation of hysteresis loss if the apparatus is connected to a 60 Hz source

```
// Example 1.3
// Computation of hysteresis loss if the apparatus is connected to a 60 Hz source
//Page No. 16

clc;
clear all;
close;

// Given data
V=240;                     // Rated voltage
F1=25;                     // Rated frequency
Ph2=846;                   // hysteresis loss
F2=60;                     // Source Frequency
Bmax1=0.62                 // Flux density is 62 percent of its rated value 1
Bmax2=1.0                  // Flux density is 62 percent of its rated value 2
Sc=1.4                     // Steinmetz exponents

//  hysteresis loss if the apparatus is connected to a 60 Hz source
Ph1=Ph2*[(F2/F1)*(Bmax1/Bmax2)^Sc];
Ph1=Ph1/1000;

//Display result on command window
printf("\n Hysteresis loss if the apparatus is connected to a 60 Hz source = %0.2f kW",Ph1);
```

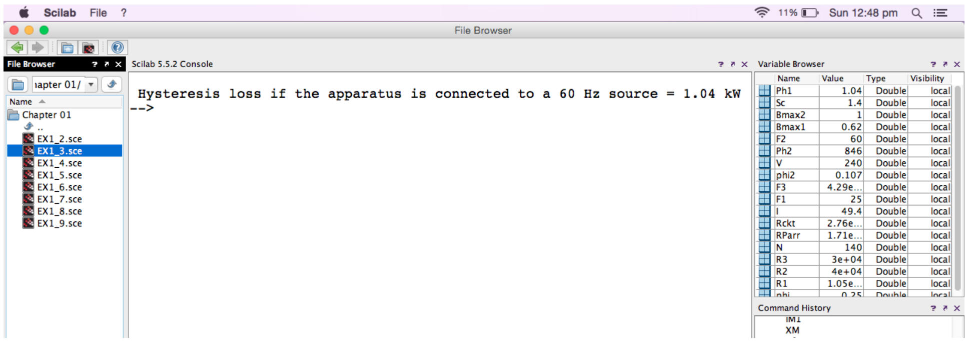

Figure 1.2: Computation of hysteresis loss if the apparatus is connected to a 60 Hz source

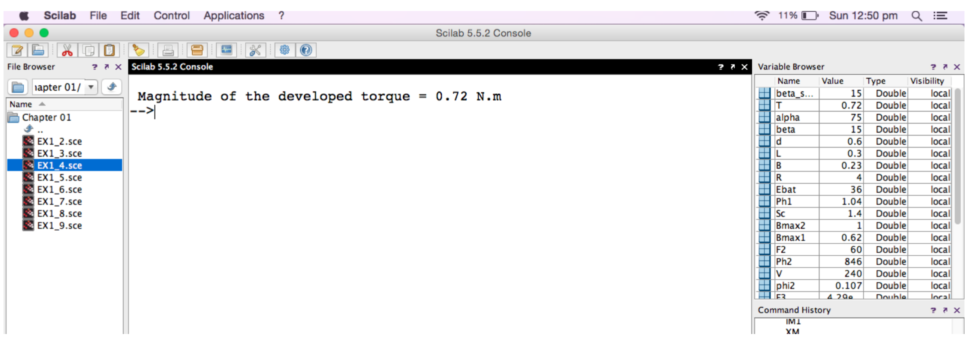

Figure 1.3: Computation of magnitude of the developed torque

Scilab code Exa 1.4 Computation of magnitude of the developed torque

```
// Example 1.4
// Computation of magnitude of the developed torque
// Page No. 21

clc;
```

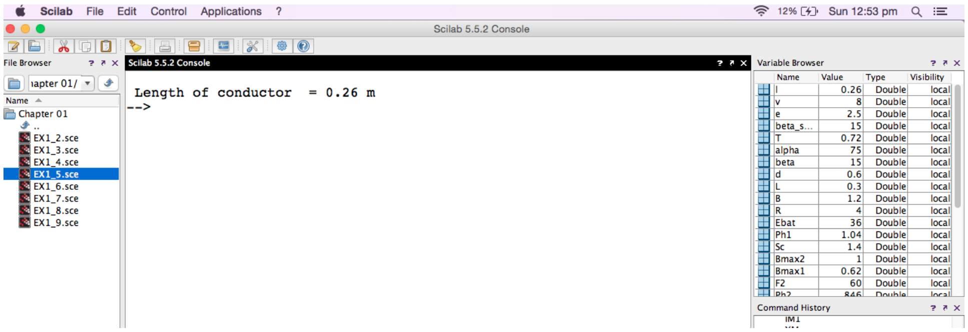

Figure 1.4: Computation of length of conductor

```
clear all;
close;

// Given data
Ebat=36;                    // Battery voltage
R=4;                        // Combined resistance of the coil
B=0.23;                     // Flux density
L=0.3;                      // Length of the coil
d=0.60;                    // Distance between centre of each conductor and centre
// of each shaft
beta_skew=15                          // Skew angle

//  Magnitude of the developed torque
alpha=90-beta_skew;
I=Ebat/R;
T=2*B*I*(L*sind(alpha))*d;   // Magnitude of the developed torque

//Display result on command window
printf("\n Magnitude of the developed torque = %0.2f N.m ",T);
```

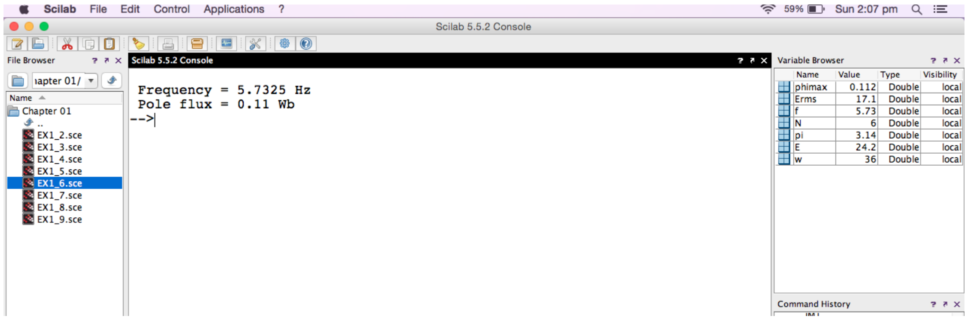

Figure 1.5: Computation of Frequency and Pole flux

Scilab code Exa 1.5 Computation of length of conductor

```
// Example 1.5
// Computation of length of conductor
// Page No. 25

clc;
clear all;
close;

// Given data
e=2.5;                    // Voltage generated
B=1.2;                   // Magnetic field
v=8.0;                    // Speed

//  Length of conductor (e=B*l*v)
l=e/(B*v);

//Display result on command window
printf("\n Length of conductor  = %0.2f m ",l);
```

Scilab code Exa 1.6 Computation of Frequency and Pole flux

```
// Example 1.6
// Computation of (a) Frequency (b) Pole flux
// Page No. 27

clc;
clear;
close;

// Given data
w=36;                    // Angular frequency
E=24.2;                  // Voltage
pi=3.14;
N=6;                     // Number of turns of rotor

// (a)  frequency
f=w/(2*pi);              // Relation between angular frequency and frequency

// (b) pole flux
Erms=E/sqrt(2);
phimax = Erms/(4.44*f*N);      // Relation to find pole flux


//Display result on command window
printf("\n Frequency = %0.4f Hz ",f);
printf("\n Pole flux = %0.2f Wb ",phimax);
```

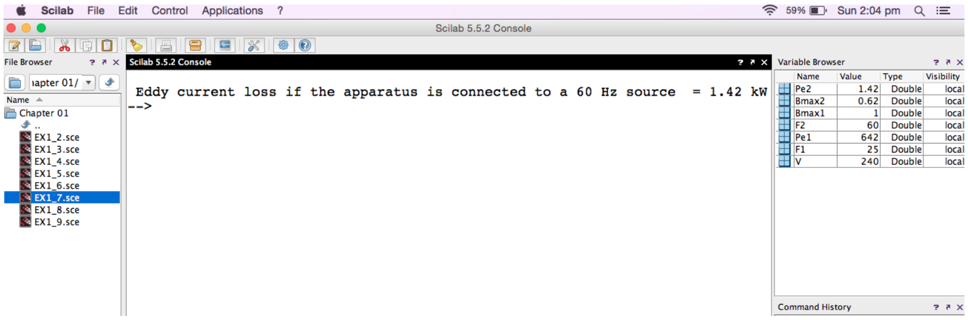

Figure 1.6: Computation of eddy current loss if the apparatus is connected to a 60 Hz source

Scilab code Exa 1.7 Computation of eddy current loss if the apparatus is connected to a 60 Hz source

```
// Example 1.7
// Computation of  eddy current loss if the apparatus is connected to a 60 Hz
//source
// Page No. 29

clc;
clear;
close;

// Given data
V=240;                    // Rated voltage
F1=25;                    // Rated frequency
Pe1=642;                  // Eddy current loss
F2=60;                    // Source Frequency
Bmax1=1.0                 // Flux density is 62 percent of its rated value
Bmax2=0.62                // Flux density is 62 percent of its rated value

//  Eddy current loss if the apparatus is connected to a 60 Hz source
```

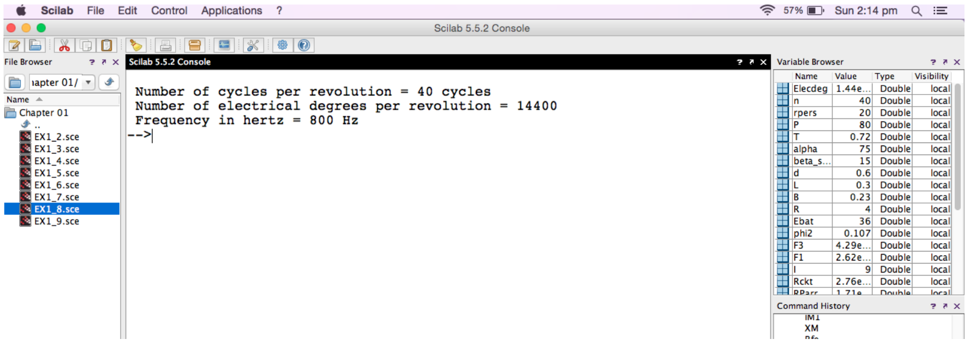

Figure 1.7: Computation of Number of cycles per revolution and Number of electrical degrees per revolution and Frequency in hertz

```
Pe2=Pe1*[(F2/F1)^2*(Bmax2/Bmax1)^2];
Pe2=Pe2/1000;

// Display result on command window
printf("\n Eddy current loss if the apparatus is
    connected to a 60 Hz source  = %0.2f kW ",Pe2);
```

Scilab code Exa 1.8 Computation of Number of cycles per revolution and Number of electrical degrees per revolution and Frequency in hertz

```
// Example 1.8
// Computation of (a) Number of cycles per
    revolution (b) Number of electrical
// degrees per revolution (c) Frequency in hertz
// Page No. 31

clc;
clear all;
close;

```

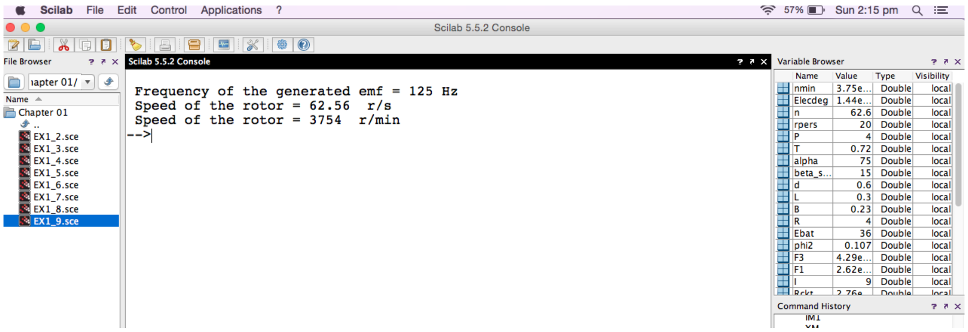

Figure 1.8: Computation of Frequency of the generated emf and Speed of the rotor

```
// Given data
P=80;                      // Number of poles
rpers=20;                  // Revolutions per second

// (a)  Number of cycles per revolution
n=P/2;

// (b) Number of electrical degrees per revolution
Elecdeg=360*P/2;

// (c) Frequency in hertz
f=P*rpers/2;

//Display result on command window
printf("\n Number of cycles per revolution = %0.0f
   cycles ",n);
printf("\n Number of electrical degrees per
   revolution = %0.0f  ",Elecdeg);
printf("\n Frequency in hertz = %0.0f Hz ",f);
```

Scilab code Exa 1.9 Computation of Frequency of the generated emf and Speed of the rotor

```
// Example 1.9
// Computation of (a) Frequency of the generated emf
    (b) Speed of the rotor
//Page No. 31

clc;
clear all;
close;

// Given data
Erms=100;                // Voltage generated in
   armature coil
N=15;                    // Number of turns in armature
   coil
phimax=0.012;            // Flux per pole
P=4;                     // Number of poles

// (a)  frequency of the generated emf
f=Erms/(4.44*N*phimax);

// (b) speed of the rotor
n=2*f/P;
nmin=n*60;

//Display result on command window
printf("\n Frequency of the generated emf = %0.0f Hz
    ",f);
printf("\n Speed of the rotor = %0.2f  r/s",n);
printf("\n Speed of the rotor = %0.0f  r/min",nmin);
```

Chapter 2

Transformer Principles

Scilab code Exa 2.1 Computation of peak value of sinusoidal flux in a transformer

```
// Example 2.1
// Computation of peak value of sinusoidal flux in a
   transformer
// Page No. 42

```

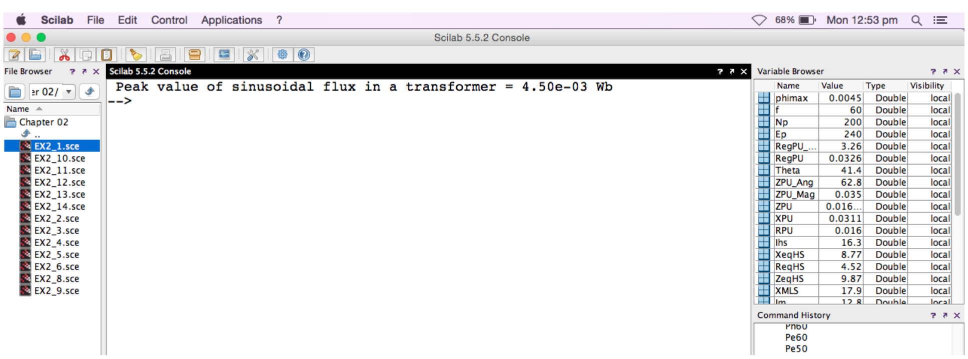

Figure 2.1: Computation of peak value of sinusoidal flux in a transformer

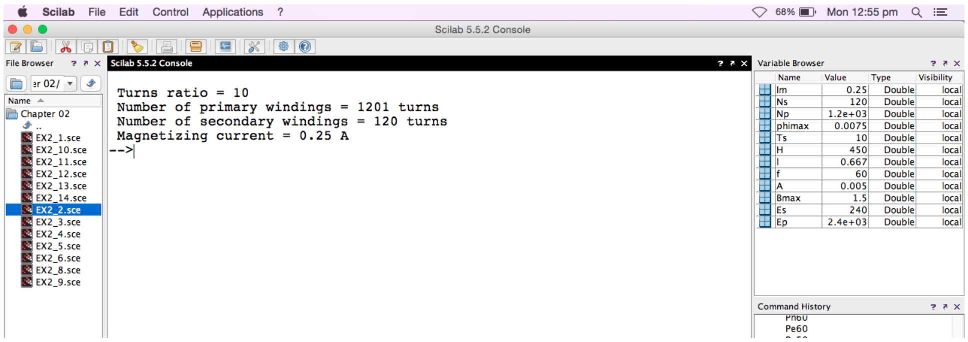

Figure 2.2: Computation of Turns ratio and Number of turns in each winding and Magnetizing current

```
clc;
clear all;
close;

// Given data
Ep=240;                    // Voltage in primary coil
Np=200;                    // Number of turns in primary coil of transformer
f=60;                      // Frequency of source

// Peak value of sinusoidal flux in a transformer
phimax=Ep/(4.44*Np*f);


//Display result on command window
//printf("\n Peak value of sinusoidal flux in a transformer = %0.4f WB ",phimax);


mprintf('Peak value of sinusoidal flux in a transformer = %3.2e Wb', phimax);
```

Scilab code Exa 2.2 Computation of Turns ratio and Number of turns in each winding and Magnetizing current

```
// Example 2.2
// Computation of (a) Turns ratio (b) Number of
   turns in each winding
// (c) Magnetizing current
// Page No. 42

clc;
clear;
close;

Ep=2400;                  // Induced emf in primary
   winding
Es=240;                   // Induced emf in primary
   winding
Bmax=1.5;                 // Maximum flux density
A=50*10^-4;               // Cross section area
f=60;                     // Frequency
l=0.667;                  // Mean length of core
H=450;                    // Magnetic field intensity


// (a) Turns ratio
Ts=Ep/Es;

// (b) Number of turns in each winding
phimax=Bmax*A;
Np=Ep/(4.44*f*phimax);                  // Number of
   primary windings
Ns=Np/Ts;                               // Number of
   secondary windings

```

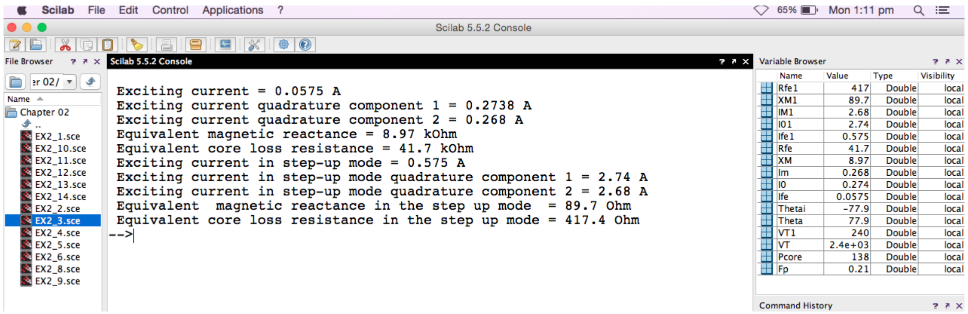

Figure 2.3: Computation of Exciting current and its quadrature components and Equalizing magnetic reactance and equivalent core loss resistance and Magnetizing current and repeat the calculations for the transformer in the step up mode

```
//(c) Magnetizing current
Im=H*l/Np;


//Display result on command window
printf("\n Turns ratio = %0.0f  ",Ts);
printf("\n Number of primary windings = %0.0f turns
   ",Np);
printf("\n Number of secondary windings = %0.0f
   turns ",Ns);
printf("\n Magnetizing current = %0.2f A ",Im);
```

Scilab code Exa 2.3 Computation of Exciting current and its quadrature components and Equalizing magnetic reactance and equivalent core loss resistance and Magnetizing current and repeat the calculations for the transformer in the step up mode

```
// Example 2.3
// Computation of (a) Exciting current and its
```

```
    quadrature components
// (b) Equalizing magnetic reactance and equivalent core loss resistance
// (c) Magnetizing current (d)repeat (a) and (b) for the transformer in the
// step up mode
//Page No. 44

clc;
clear;
close;

Fp=0.210;                  // Power factor
Pcore=138;                 // Active power
VT=2400;                   // Voltage applied to primary
VT1=240;                   // 240-V primary voltage -- Second case


// (a)Exciting current and its quadrature components
Theta=acosd(Fp);           // Angle
Thetai=-Theta;             // As phase angle of applied voltage is zero
Ife=Pcore/VT;              // Exciting current
I0=Ife/Fp;                 // Quadrature component
Im=tand(Thetai)*Ife;       // Quadrature component
Im=Im*-1;


// (b) Equalizing magnetic reactance and equivalent core loss resistance
XM=VT/Im;                    // Magnetic reactance
Rfe=VT/Ife;                  // Core-loss resistance
XM=XM/1000;
Rfe=Rfe/1000;
//(c) Magnetizing current
Ife1=Pcore/VT1;              // Exciting current
```

```
IO1=Ife1/cosd(Thetai);
IM1=tand(Thetai)*Ife1;      // Quadrature component
IM1=IM1*-1;

//(d) repeat (a) and (b) for the transformer in the step up mode
XM1=VT1/IM1;                 // Magnetizing reactance
Rfe1=VT1/Ife1;               // Core-loss resistance



//Display result on command window
printf("\n Exciting current = %0.4f A ",Ife);
printf("\n Exciting current quadrature component 1 = %0.4f A ",IO);
printf("\n Exciting current quadrature component 2 = %0.3f A ",Im);
printf("\n Equivalent magnetic reactance = %0.2f kOhm ",XM);
printf("\n Equivalent core loss resistance = %0.1f kOhm ",Rfe);
printf("\n Exciting current in step-up mode = %0.3f A ",Ife1);
printf("\n Exciting current in step-up mode quadrature component 1 = %0.2f A ",IO1);
printf("\n Exciting current in step-up mode quadrature component 2 = %0.2f A ",IM1);
printf("\n Equivalent  magnetic reactance in the step up mode  = %0.1f Ohm ",XM1);
printf("\n Equivalent core loss resistance in the step up mode = %0.1f Ohm ",Rfe1);
```

Scilab code Exa 2.4 Computation of Secondary voltage and Load current

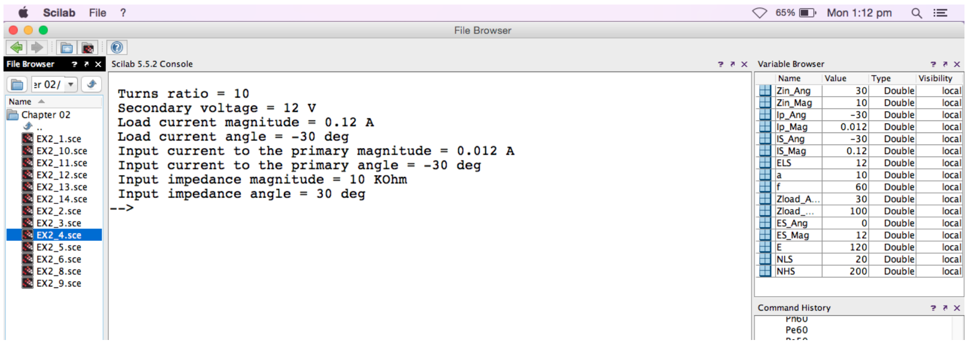

Figure 2.4: Computation of Secondary voltage and Load current and Input current to the primary and Input impedance looking into the primary terminals

and Input current to the primary and Input impedance looking into the primary terminals

```
// Example 2.4
// Computation of (a) Secondary voltage (b) Load current
// (c) Input current to the primary (d) Input impedance looking into the primary terminals
// Page No. 51

clc;
clear;
close;

NHS=200;                // Number of turns in primary
NLS=20;                 // Number of turns in secondary
E=120;                  // Primary voltage magnitude
ES_Mag=12;              // Secondary voltage magnitude
ES_Ang=0;               // Secondary voltage angle
Zload_Mag=100;          // Load magnitude
Zload_Ang=30;           // Load angle
f=60;                   // Frequency
```

```
// (a)  Secondary voltage
a=NHS/NLS;
ELS=E/a;

// (b) Load current
IS_Mag=ES_Mag/Zload_Mag;          // Load current
   magnitude
IS_Ang=ES_Ang - Zload_Ang;        // Load current
   angle

//(c) Input current to the primary
Ip_Mag=IS_Mag/a;                  // Input current to
    the primary magnitude
Ip_Ang=IS_Ang;                    // Input current to
    the primary angle

//(d) Input impedance looking into the primary
   terminals
Zin_Mag=a^2*Zload_Mag;            // Input impedance
   magnitude
Zin_Ang=Zload_Ang;                // Input impedance
   angle
Zin_Mag=Zin_Mag/1000;

// Display result on command window
printf("\n Turns ratio = %0.0f  ",a);
printf("\n Secondary voltage = %0.0f V", ELS);
printf("\n Load current magnitude = %0.2f A",IS_Mag)
   ;
printf("\n Load current angle = %0.0f deg",IS_Ang);
printf("\n Input current to the primary magnitude =
   %0.3f A",Ip_Mag);
printf("\n Input current to the primary angle = %0.0
   f deg",Ip_Ang);
printf("\n Input impedance magnitude = %0.0f KOhm",
   Zin_Mag);
printf("\n Input impedance angle = %0.0f deg",
```

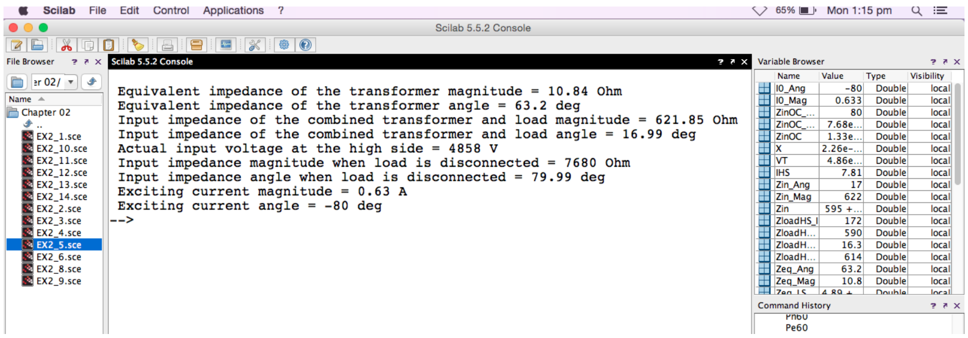

Figure 2.5: Computation of Equivalent impedance of the transformer referred to the high side and Input impedance of the combined transformer and load and Actual input voltage at the high side and Input impedance if the load is disconnected and Exciting current

```
    Zin_Ang);
```

Scilab code Exa 2.5 Computation of Equivalent impedance of the transformer referred to the high side and Input impedance of the combined transformer and load and Actual input voltage at the high side and Input impedance if the load is disconnected and Exciting current

```
// Example 2.5
// Computation of (a) Equivalent impedance of the
   transformer referred to the
// high side (b) Input impedance of the combined
   transformer and load (C) Actual
// input voltage at the high side (d) Input
   impedance if the load is disconnected
// (e) Exciting current for the conditions in (d)
// Page No. 60

clc;
clear;
```

```
close;

// Given data
S=75000;                    // Transformer ratings
VLS=240;                    // Low side voltage magnitude
PF=0.96;                    // Lagging power factor
VLS_Ang=0;                  // Low side voltage angle
VL=240;                     // Load voltage
VHS=4800;                   // High side voltage
RHS=2.488;                  // High side resistance
RLS=0.00600;                // Low side resistance
XHS=4.8384;                 // High side reactance
XLS=0.0121                  // Low side reactance
Rfe=44202;                  // High side resistance
Xm=7798.6;                  // High side reactance


// (a) Equivalent impedance of the transformer referred to the
// high side
ILS=S*1/2/VLS;                        // Delivering one-half rated load
Theta=acosd(PF);                      // Angle
ThetaI=0-Theta;
ZloadLS_Mag=VLS/ILS;                  // Low side impedance magnitude
ZloadLS_Ang=VLS_Ang-ThetaI;           // Low side impedance angle

a=VHS/VL;                             // Ratio of High side and low side voltages
Zeq_LS=RHS+a^2*RLS+%i*(XHS+a^2*XLS)

// Complex to Polar form...

Zeq_Mag=sqrt(real(Zeq_LS)^2+imag(Zeq_LS)^2);
              // Magnitude part
Zeq_Ang= atan(imag(Zeq_LS),real(Zeq_LS))*180/%pi;
```

```
// Angle part

// (b) Input impedance of the combined transformer and load
ZloadHS_Mag=a^2*ZloadLS_Mag;          // High side impedance magnitude
ZloadHS_Ang=ZloadLS_Ang;              // High side impedance angle

// Polar to Complex form

ZloadHS_R=ZloadHS_Mag*cos(-ZloadHS_Ang*%pi/180); // Real part of complex number
ZloadHS_I=ZloadHS_Mag*sin(ZloadHS_Ang*%pi/180);  // Imaginary part of complex number
Zin=ZloadHS_R+%i* ZloadHS_I+Zeq_LS;              // Input impedance
// Complex to Polar form...

Zin_Mag=sqrt(real(Zin)^2+imag(Zin)^2);           // Magnitude part
Zin_Ang= atan(imag(Zin),real(Zin))*180/%pi;      // Angle part

// (c) Actual input voltage at the high side
IHS=ILS/a;                    // High side current
VT=IHS*Zin_Mag;

// (d) Input impedance if the load is disconnected
X=(1/Rfe)+(1/Xm*%i);
ZinOC=1/X;                                       // Input impedance
ZinOC_Mag=sqrt(real(ZinOC)^2+imag(ZinOC)^2);     // Magnitude part
ZinOC_Ang= atan(imag(ZinOC),real(ZinOC))*180/%pi; // Angle part
ZinOC_Ang=ZinOC_Ang*-1;

```

```
// (e) Exciting current for the conditions in (d)
I0_Mag=VT/ZinOC_Mag;                  // Magnitude of current
I0_Ang=0-ZinOC_Ang;                   // Angle of current

// Display result on command window
printf("\n Equivalent impedance of the transformer magnitude = %0.2f Ohm ",Zeq_Mag);
printf("\n Equivalent impedance of the transformer angle = %0.1f deg ",Zeq_Ang);
printf("\n Input impedance of the combined transformer and load magnitude = %0.2f Ohm ",Zin_Mag);
printf("\n Input impedance of the combined transformer and load angle = %0.2f deg ",Zin_Ang);
printf("\n Actual input voltage at the high side = %0.0f V", VT);
printf(" \n Input impedance magnitude when load is disconnected = %0.0f Ohm",ZinOC_Mag);
printf(" \n Input impedance angle when load is disconnected = %0.2f deg",ZinOC_Ang);
printf(" \n Exciting current magnitude = %0.2f A",I0_Mag);
printf(" \n Exciting current angle = %0.0f deg",I0_Ang);
```

Scilab code Exa 2.6 Computation of a Equivalent input impedance of the transformer and load combination and Primary current when 2400V is supplied to primary and Voltage across the load

```
// Example 2.6
// Computation of (a) Equivalent input impedance of
```

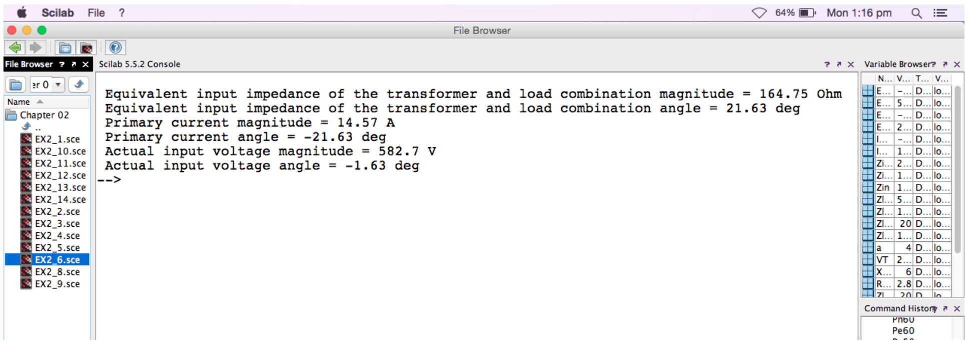

Figure 2.6: Computation of a Equivalent input impedance of the transformer and load combination and Primary current when 2400V is supplied to primary and Voltage across the load

```
    the transformer and load
// combination (b) Primary current when 2400V is
    supplied to primary
// (C) Voltage across the load
// Page No. 61

clc;
clear;
close;

// Given data
S=37500;                 // Transformer ratings
VHS=2400;                // High side voltage
VLS=600;                 // Low side voltage magnitude
ZloadLS_Mag=10;          // Low side load impedance
    magnitude
ZloadLS_Ang=20;          // Low side load impedance
    angle
Req=2.8;                 // Equivalent resistance
Xeq=6;                   // Equivalent reactance
VT=2400;                 // Primary voltage supplied

// (a) Equivalent input impedance of the transformer
     and load combination
```

```
a=VHS/VLS;                        // Ratio of High side and low side voltages
ZloadHS_Mag=a^2*ZloadLS_Mag;      // High side load impedance magnitude
ZloadHS_Ang=ZloadLS_Ang;          // High side load impedance angle
// Polar to Complex form
ZloadHS_R=ZloadHS_Mag*cos(-ZloadHS_Ang*%pi/180); // Real part of complex number
ZloadHS_I=ZloadHS_Mag*sin(ZloadHS_Ang*%pi/180);  // Imaginary part of complex number
Zin=Req+%i*Xeq+ZloadHS_R+%i*ZloadHS_I;
// Complex to Polar form...
Zin_Mag=sqrt(real(Zin)^2+imag(Zin)^2);        // Magnitude part
Zin_Ang = atan(imag(Zin),real(Zin))*180/%pi;  // Angle part
// (b) Primary current when 2400V is supplied to primary
IHS_Mag=VT/Zin_Mag;                     // Primary current magnitude
IHS_Ang=0-Zin_Ang;                      // Primary current angle
// (c) Voltage across the load
EHS_Mag= IHS_Mag*a^2*ZloadLS_Mag; // Magnitude of voltage across reflected load
EHS_Ang=IHS_Ang+ZloadLS_Ang;      // Angle of voltage across reflected load
ELS_Mag=EHS_Mag/a;                // Magnitude of actual voltage across real load
ELS_Ang=EHS_Ang;                  // Angle of actual voltage across real load
```

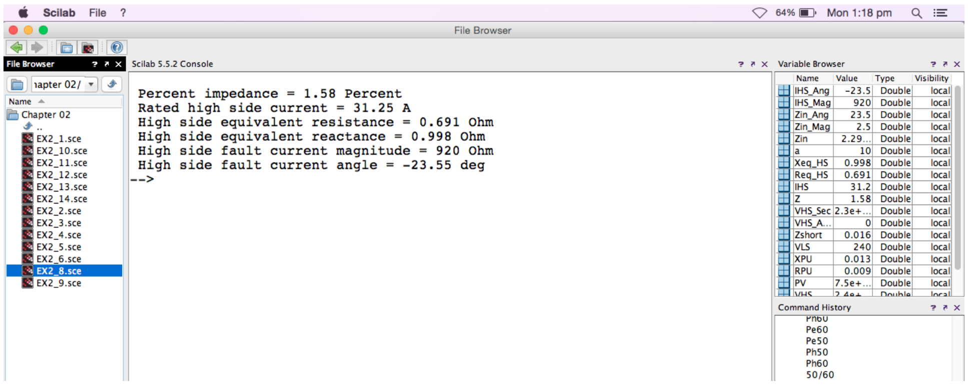

Figure 2.7: Computation of Percent impedance and Rated high side current and Equivalent resistance and reactance referred to the high side and High side fault current

```
//Display result on command window
printf("\n Equivalent input impedance of the transformer and load combination magnitude = %0.2f Ohm ",Zin_Mag);
printf("\n Equivalent input impedance of the transformer and load combination angle = %0.2f deg ",Zin_Ang);
printf("\n Primary current magnitude = %0.2f A ",IHS_Mag);
printf("\n Primary current angle = %0.2f deg ",IHS_Ang);
printf("\n Actual input voltage magnitude = %0.1f V", ELS_Mag);
printf(" \n Actual input voltage angle = %0.2f deg",ELS_Ang);
```

Scilab code Exa 2.8 Computation of Percent impedance and Rated high side current and Equivalent resistance and reactance referred to the high side and High side fault current

```
// Example 2.8
// Computation of (a) Percent impedance (b) Rated
    high side current
// (c) Equivalent resistance and reactance referred
    to the high side
// (d) High side fault current if an accidental
    short circuit of 0.016 Ohm
// occurs at secondary when 230V impressed across
    the primary
// Page No. 66

clc;
clear;
close;

// Given data
R=0.9;                    // Percent resistance
X=1.3;                    // Percent reactance
VHS=2400;                 // High side voltage
PV=75000;                 // Transformer power rating
RPU=0.009                 // Per unit resistance
XPU=0.013                 // Per unit reactance
VLS=240;                  // Low side voltage
Zshort=0.016;             // Short circuit resistance
VHS_Ang=0;                // High side voltage angle
VHS_Sec=2300;             // Secondary high side
    voltage

// (a) Percent impedance
Z=sqrt(R^2+X^2);

// (b) Rated high side current
IHS=PV/VHS;

//(c) Equivalent resistance referred to the high
    side
```

```
Req_HS=RPU*VHS/IHS;
// Equivalent reactance referred to the high side
Xeq_HS=XPU*VHS/IHS;

//(d) High side fault current
a=VHS/VLS;                                    // Ratio of High side and low side voltages
Zin=Req_HS+%i*Xeq_HS+a^2*Zshort;              // Input impedance
Zin_Mag=sqrt(real(Zin)^2+imag(Zin)^2);        // Magnitude part of input impedance
Zin_Ang= atan(imag(Zin),real(Zin))*180/%pi; // Angle part
IHS_Mag=VHS_Sec/Zin_Mag;                        // High side current magnitude
IHS_Ang=VHS_Ang-Zin_Ang;


//Display result on command window
printf("\n Percent impedance = %0.2f Percent ",Z);
printf("\n Rated high side current = %0.2f A", IHS);
printf(" \n High side equivalent resistance = %0.3f Ohm",Req_HS);
printf(" \n High side equivalent reactance = %0.3f Ohm",Xeq_HS);
printf(" \n High side fault current magnitude = %0.0f Ohm",IHS_Mag);
printf(" \n High side fault current angle = %0.2f deg",IHS_Ang);
```

Scilab code Exa 2.9 Computation of Transformer regulation and Secondary voltage when the load is disconnected and Input primary voltage

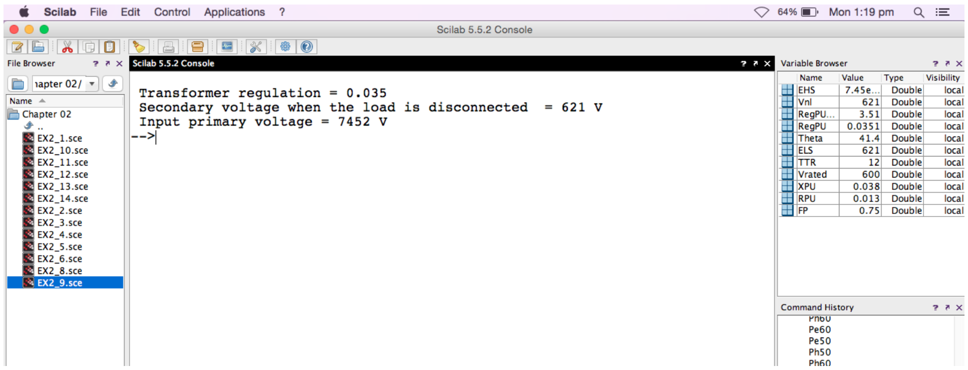

Figure 2.8: Computation of Transformer regulation and Secondary voltage when the load is disconnected and Input primary voltage

```
// Example 2.9
// Computation of (a) Transformer regulation (b) Secondary voltage when the
// load is disconnected (c) Input primary voltage
// Page No. 69

clc;
clear;
close;

// Given data
FP=0.75                    // Power-factor lagging
RPU=0.013;                 // Percent resistance
XPU=0.038;                 // Percent reactance
Vrated=600;                // Rated voltage of transformer
TTR=12;                    // Transformer turns ratio (7200/600)
ELS=621;                   // Low side voltage



// (a) Transformer regulation
Theta=acosd(FP);
```

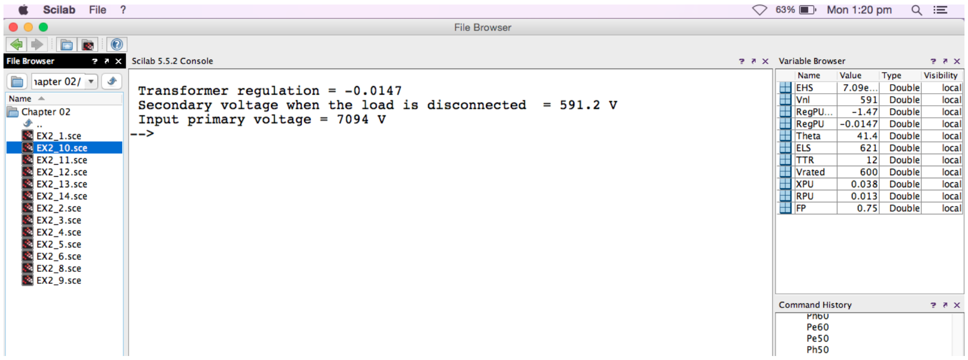

Figure 2.9: Computation of Transformer regulation and Secondary voltage when the load is disconnected and Input primary voltage

```
// Transformer regulation
RegPU=sqrt( ( (RPU+FP)^2)+  ((XPU+sind(Theta))^2))-1;
// Transformer regulation in percentage
RegPU_Per=RegPU*100;

// (b) Secondary voltage when the load is disconnected
Vnl=(RegPU*Vrated)+Vrated;

// (c) Input primary voltage
EHS=ELS*TTR;

// Display result on command window
printf("\n Transformer regulation = %0.3f ",RegPU);
printf("\n Secondary voltage when the load is disconnected  = %0.0f V", Vnl);
printf(" \n Input primary voltage = %0.0f V",EHS);
```

Scilab code Exa 2.10 Computation of Transformer regulation and Secondary voltage when the load is disconnected and Input primary voltage

```
// Example 2.10
// Computation of (a) Transformer regulation (b)
    Secondary voltage when the
// load is disconnected (c) Input primary voltage
// Page No. 70

clc;
clear;
close;

// Given data
FP=0.75                        // Power-factor leading
RPU=0.013;                     // Percent resistance
XPU=0.038;                     // Percent reactance
Vrated=600;                    // Rated voltage of
    transformer
TTR=12;                        // Transformer turns ratio
    (7200/600)
ELS=621;                       // Low side voltage



// (a) Transformer regulation
Theta=acosd(FP);
// Transformer regulation
RegPU=sqrt( ( (RPU+FP)^2)+  ((XPU-sind(Theta))^2))
    -1;
// Transformer regulation in percentage
RegPU_Per=RegPU*100;

// (b) Secondary voltage when the load is
    disconnected
Vnl=(RegPU*Vrated)+Vrated;

// (c) Input primary voltage
```

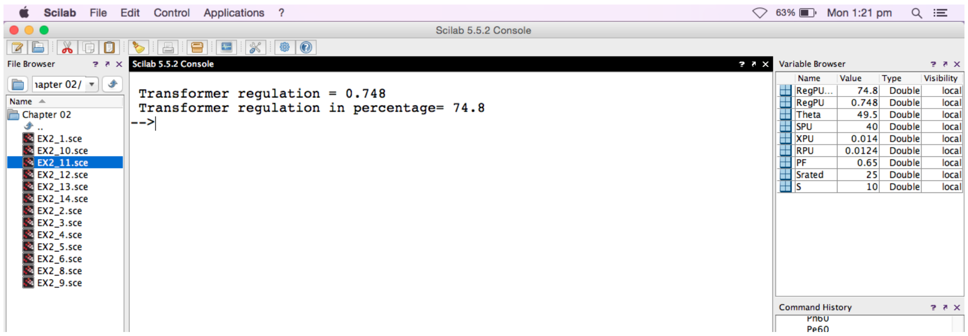

Figure 2.10: Computation of transformer regulation

```

EHS=Vnl*TTR;

// Display result on command window
printf("\n Transformer regulation = %0.4f ",RegPU);
printf("\n Secondary voltage when the load is
    disconnected = %0.1f V", Vnl);
printf(" \n Input primary voltage = %0.0f V",EHS);
```

Scilab code Exa 2.11 Computation of transformer regulation

```
// Example 2.11
// Computation of transformer regulation
// Page No. 71

clc;
clear;
close;

// Given data
S=10;                   // Transformer actual rating
    10KVA
```

```
Srated=25;                  // Rated 25KVA
PF=0.65;                    // Power factor lagging
RPU=0.0124;                 // Percent resistance drop
XPU=0.014;                  // Percent reactance drop

// Transformer regulation
SPU=S/Srated;
SPU=SPU*100;
Theta=acosd(PF);
// Transformer regulation
RegPU=sqrt( ( (RPU*SPU+PF)^2)+  ((XPU*SPU+sind(Theta
   ))^2))-1;
// Transformer regulation in percentage
RegPU_Per=RegPU*100;

// Display result on command window
printf("\n Transformer regulation = %0.3f ",RegPU);
printf("\n Transformer regulation in percentage= %0
   .1f ",RegPU_Per);

// Answer varies due to round off errors
```

Scilab code Exa 2.12 Computation of Core loss and Core loss if operated at rated current and reduced power factor from 375V 50 HZ supply and Overall efficiency and Efficiency if the load is disconnected

```
// Example 2.12
// Computation of (a) Core loss (b) Core loss if
   operated at rated current and
// 0.860 power factor from 375V, 50 HZ supply (c)
   Efficiency for condition in (b)
// (d) Efficiency if the load is disconnected
// Page No. 72
```

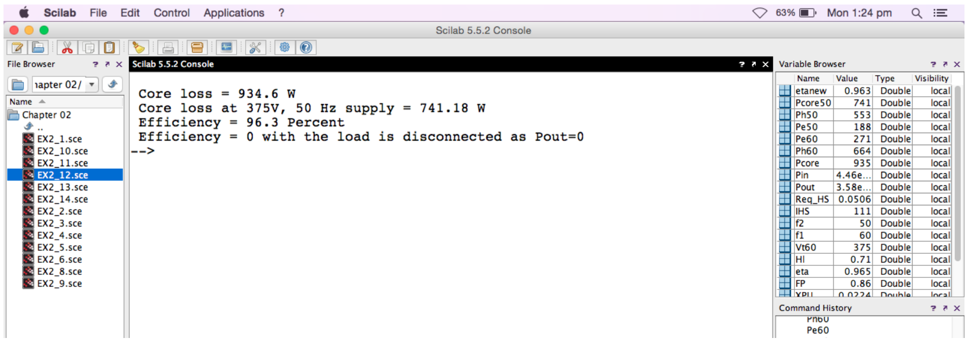

Figure 2.11: Computation of Core loss and Core loss if operated at rated current and reduced power factor from 375V 50 HZ supply and Overall efficiency and Efficiency if the load is disconnected

```

clc;
clear;
close;

// Given data
Srated=50000;                // Transformer power rating
VHS=450;                     // High side voltage
RPU=0.0125;                  // Percent resistance
XPU=0.0224;                  // Percent reactance
FP=0.86;                     // Power factor lagging
eta=0.965                    // Efficiency
Hl=0.71                      // Hysteresis loss
Vt60=375                     // Supply voltage
f1=60;                       // Transformer frequency
f2=50;                       // Supply frequency


// (a) Core loss
IHS=Srated/VHS;
// Using high-side values
Req_HS=RPU*VHS/IHS;              // Equivalent high-side resistance
```

```
Pout=Srated*FP;                    // Output power
Pin=Pout/eta;                      // Input power
Pcore=Pin-Pout-(IHS^2*Req_HS)      // Core loss

// (b) Core loss if operated at rated current and
    0.860 power factor from
// 375V, 50 HZ supply
Ph60=Hl*Pcore;                  // Hysteresis loss
Pe60=Pcore-Ph60;                // Eddy current loss
Pe50=Pe60*(Vt60/VHS)^2;         // Eddy current loss
Ph50=Ph60*(f2/f1)*(Vt60/VHS*f1/f2)^1.6;
Pcore50=Pe50+Ph50;              // Core loss

// (c) Efficiency
Pout=Vt60*IHS*FP;               // Output power
etanew=Pout/(Pout+Pcore50+IHS^2*Req_HS);

// (d) Efficiency with the load is disconnected

// Display result on command window
printf("\n Core loss = %0.1f W", Pcore);
printf("\n Core loss at 375V, 50 Hz supply = %0.2f W
    ",Pcore50);
printf("\n Efficiency = %0.1f Percent", etanew*100);
printf("\n Efficiency = 0 with the load is
    disconnected as Pout=0" )
```

Scilab code Exa 2.13 Determine Efficiency at rated load and 80 percent power factor and 70 percent load and 80 percent power factor

```
// Example 2.13
// Determine (a) Efficiency at rated load and 80%
    power factor
```

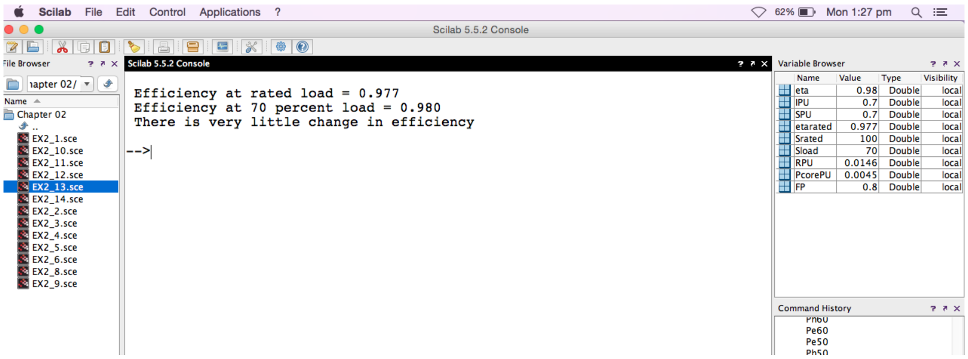

Figure 2.12: Determine Efficiency at rated load and 80 percent power factor and 70 percent load and 80 percent power factor

```
// (b) 70% load and 80% power factor
// Page No. 75

clc;
clear;
close;

// Given data
FP=0.80;                    // Power factor
PcorePU=0.0045;             // Percentage core loss
RPU=0.0146;                 // Percentage resistance
Sload=70;                   // 70% rated load
Srated=100;                 // 100% rated load

// (a) Efficiency at rated load and 80% power factor
etarated=FP/(FP+RPU+PcorePU);

// (b) Efficiency at 70% load and 80% power factor
SPU=Sload/Srated;
IPU=SPU;                                    // I_load
    is proportional to S_load
eta=(SPU*FP)/(SPU*FP+PcorePU+IPU^2*RPU)     //
    Efficiency

```

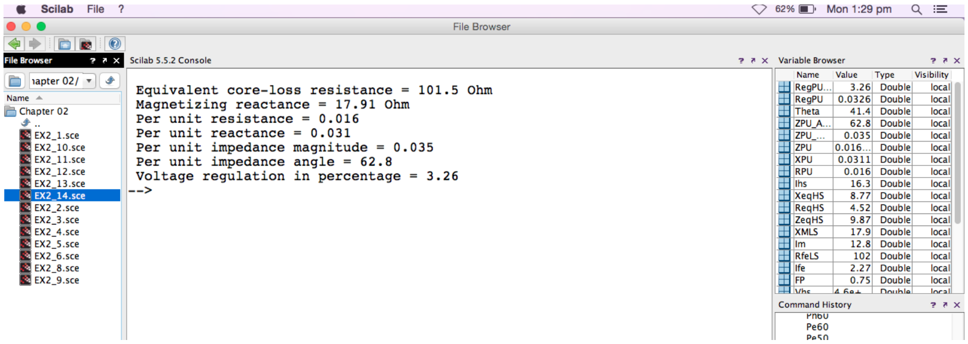

Figure 2.13: Determine Magnetizing reactance and equivalent core loss resistance and Per unit resistance reactance and impedance of transformer windings and Voltage regulation

```
// Display result on command window
printf("\n Efficiency at rated load = %0.3f ",
    etarated);
printf("\n Efficiency at 70 percent load = %0.3f ",
    eta);
disp('There is very little change in efficiency');
```

Scilab code Exa 2.14 Determine Magnetizing reactance and equivalent core loss resistance and Per unit resistance reactance and impedance of transformer windings and Voltage regulation

```
// Example 2.14
// Determine (a) Magnetizing reactance and
    equivalent core-loss resistance
// (b) Per unit resistance , reactance and impedance
    of transformer windings
// (c) Voltage regulation when operating at rated
    load and 0.75 power factor lagging
// Page No. 78
```

```
clc;
clear;
close;

// Given data
Poc=521;                          // Open circuit test power
Voc=230;                          // Open circuit voltage
Vo=230;                           // Output voltage
Ioc=13.04;                        // Open circuit current
Vsc=160.8;                        // Short circuit voltage
Isc=16.3;                         // Short circuit current
Psc=1200;                         // Short circuit power
S=75000;                          // Transformer rating
Vhs=4600;                         // High side voltage
FP=0.75;                          // Power factor lagging

// (a) Magnetizing reactance and equivalent core-loss resistance
Ife=Poc/Voc;                     // Current rating
RfeLS=Vo/Ife;                    // Core-loss resistance
Im=sqrt(Ioc^2-Ife^2);            // Magnetizing current
XMLS=Voc/Im;                     // Magnetizing reactance

// (b) Per unit resistance, reactance and impedance of transformer windings
ZeqHS=Vsc/Isc;                   // Equivalent impedance
ReqHS=Psc/Isc^2;                 // Equivalent resistance
XeqHS=sqrt(ZeqHS^2 - ReqHS^2); // Equivalent reactance
Ihs=S/Vhs;                        // High side current
RPU=Ihs*ReqHS/Vhs;                // Per unit resistance
XPU=Ihs*XeqHS/Vhs;                // Per unit reactance
ZPU=RPU+%i*XPU;                   // Per unit impedance
// Complex to Polar form...
ZPU_Mag=sqrt(real(ZPU)^2+imag(ZPU)^2);       // Magnitude part
```

```
ZPU_Ang=atan(imag(ZPU),real(ZPU))*180/%pi;  // Angle part

// (c) Voltage regulation when operating at rated load and 0.75 power factor lagging
// Transformer regulation
Theta=acosd(FP);
RegPU=sqrt( (RPU+FP)^2 +  (XPU+sind(Theta))^2 )-1;
// Transformer regulation in percentage
RegPU_Per=RegPU*100;

// Display result on command window
printf("\n Equivalent core-loss resistance = %0.1f Ohm",RfeLS);
printf("\n Magnetizing reactance = %0.2f Ohm", XMLS);
printf("\n Per unit resistance = %0.3f ", RPU);
printf("\n Per unit reactance = %0.3f ", XPU);
printf("\n Per unit impedance magnitude = %0.3f ", ZPU_Mag);
printf("\n Per unit impedance angle = %0.1f ", ZPU_Ang);
printf("\n Voltage regulation in percentage = %0.2f ", RegPU_Per);
```

Chapter 3

Transformer Connections Operation and Specialty Transformers

Scilab code Exa 3.1 Computation of Load current and Incoming line current and Transformed current and Apparent power conducted and apparent power transformed

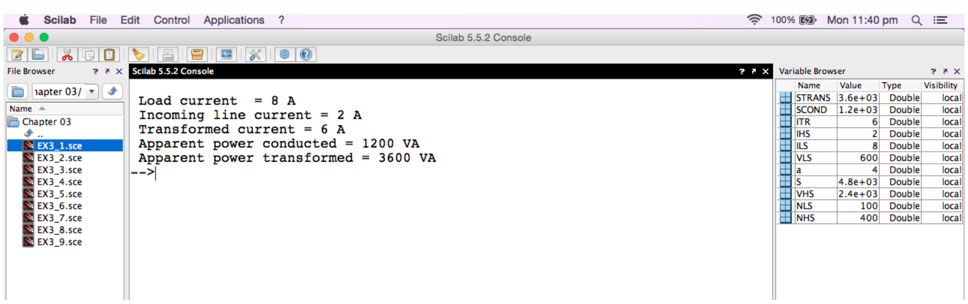

Figure 3.1: Computation of Load current and Incoming line current and Transformed current and Apparent power conducted and apparent power transformed

```
// Example 3.1
// Computation of (a) Load current (b) Incoming line
    current
// (c) Transformed current (d) Apparent power
   conducted and apparent power transformed
// Page No. 98

clc;
clear;
close;

// Given data
NHS=400;                // Number of turns in the
   high side
NLS=0.25*400;           // Number of turns in the low
    side
VHS=2400;               // Voltage at the high side
S=4800;                 // Supply voltage

// (a) Load current
a=NHS/NLS;              // Transformer turn ratio
VLS=VHS/a;              // Low side voltage
ILS=S/VLS;              // Load current

// (b) Incoming line current
IHS=ILS/a;

//(c)  Transformed current
ITR=ILS-IHS;

// (d) Apparent power conducted and apparent power
   transformed

SCOND=IHS*VLS;      // Apparent power conducted
STRANS=ITR*VLS;     // Apparent power transformed


// Display result on command window
```

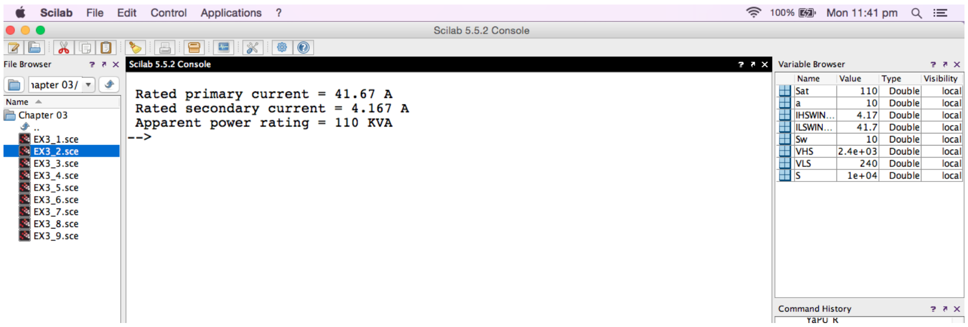

Figure 3.2: Computation of Rated primary and secondary currents and Apparent power rating

```
printf("\n Load current  = %0.0f A ",ILS);
printf("\n Incoming line current = %0.0f A ",IHS);
printf("\n Transformed current = %0.0f A ",ITR);
printf("\n Apparent power conducted = %0.0f VA ",
    SCOND);
printf("\n Apparent power transformed = %0.0f VA ",
    STRANS);
```

Scilab code Exa 3.2 Computation of Rated primary and secondary currents and Apparent power rating

```
// Example 3.2
// Computation of (a) Rated primary and secondary
    currents when connected as
// autotransformer (b) Apparent power rating when
    connected as an autotransformer
// Page No. 100

clc;
clear;
close;
```

```
// Given data
S=10000;                // Supply voltage
VLS=240;                // Voltage at the low side
VHS=2400;               // Voltage at the high side
Sw=10;                  // Power rating

// (a) Rated primary and secondary currents when
    connected as autotransformer

ILSWINDING=S/VLS;       // Rated primary current
IHSWINDING=S/VHS;       // Rated secondary current


// (b) Apparent power rating when connected as an
    autotransformer

a=VHS/VLS;                    // Magnetic drop across
   R1
Sat=(a+1)*Sw;


//Display result on command window

printf("\n Rated primary current = %0.2f A ",
    ILSWINDING);
printf("\n Rated secondary current = %0.3f A ",
    IHSWINDING);
printf("\n Apparent power rating = %0.0f KVA ",Sat);
```

Scilab code Exa 3.3 Computation of Buck boost transformer parameters and Repeating the same assuming utilization voltage as 246V

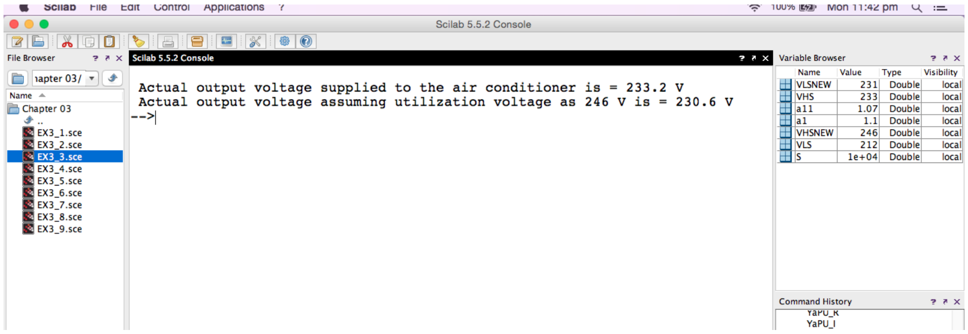

Figure 3.3: Computation of Buck boost transformer parameters and Repeating the same assuming utilization voltage as 246V

```
// Example 3.3
// Computation of (a) Buck boost transformer parameters
// (b) Repeating the same assuming utilization voltage as 246V
// Page No. 102

clc;
clear;
close;

// Given data
S=10000;                // Supply voltage
VLS=212;                // Voltage at the low side
VHSNEW=246;             // New voltage at the high side
a1=1.100;
a11=1.0667;

// (a) Buck boost transformer parameters
VHS=a1*VLS;

// (b) Repeating the same assuming utilization voltage as 246V

```

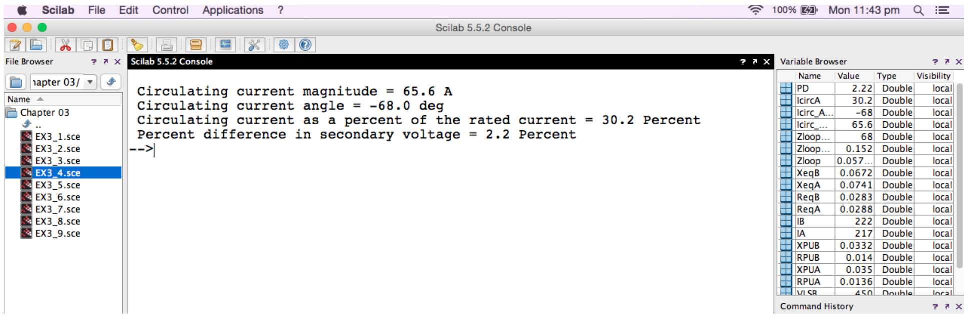

Figure 3.4: Determine Circulating current in the paralleled secondaries and Circulating current as a percent of the rated current of transformer A and Percent difference in secondary voltage

```
VLSNEW=VHSNEW/a11;

//Display result on command window

printf("\n Actual output voltage supplied to the air conditioner is = %0.1f V ",VHS);
printf("\n Actual output voltage assuming utilization voltage as 246 V is = %0.1f V ",VLSNEW);
```

Scilab code Exa 3.4 Determine Circulating current in the paralleled secondaries and Circulating current as a percent of the rated current of transformer A and Percent difference in secondary voltage

```
// Example 3.4
// Determine (a) Circulating current in the paralleled secondaries
// (b) Circulating current as a percent of the rated current of transformer A
```

```
// (c) Percent difference in secondary voltage that
    caused the circulating current
// Page No. 104

clc;
clear;
close;

// Given data
S=100000;                    // Transformer A and B rating
VLSA=460;                    // Voltage at the low side of
     transformer A
VLSB=450;                    // Voltage at the low side of
     transformer A
RPUA=0.0136;                 // Percent resistance of
    transformer A
XPUA=0.0350;                 // Percent reactance of
    transformer A
RPUB=0.0140;                 // Percent resistance of
    transformer B
XPUB=0.0332;                 // Percent reactance of
    transformer B



// (a) Circulating current in the paralleled
    secondaries
IA= S/VLSA;                     // Rated low side current
    for transformer A
IB= S/VLSB;                     // Rated low side current
    for transformer B
ReqA=RPUA*VLSA/IA;              // Equivalent resistance
    of transfomer A
ReqB=RPUB*VLSB/IB;              // Equivalent resistance
    of transfomer B
XeqA=XPUA*VLSA/IA;              // Equivalent reactance of
     transfomer A
XeqB=XPUB*VLSB/IB;              // Equivalent reactance of
```

```
    transfomer B

// Impedance of the closed loop formed by two
    secondaries is
Zloop=ReqA+%i*XeqA+ReqB+%i*XeqB;
// Complex to Polar form...
Zloop_Mag=sqrt(real(Zloop)^2+imag(Zloop)^2); //
    Magnitude part
Zloop_Ang=atan(imag(Zloop),real(Zloop))*180/%pi; //
    Angle part

Icirc_Mag=(VLSA-VLSB)/Zloop_Mag; // Circulating
    current magnitude
Icirc_Ang=0- Zloop_Ang;           // Circulating
    current angle

// (b) Circulating current as a percent of the rated
    current of transformer A
IcircA=Icirc_Mag*100/IA;

// (c) Percent difference in secondary voltage that
    caused the circulating current
PD=(VLSA-VLSB)*100/VLSB;

// Display result on command window

printf("\n Circulating current magnitude = %0.1f A "
    ,Icirc_Mag);
printf("\n Circulating current angle = %0.1f deg ",
    Icirc_Ang);
printf("\n Circulating current as a percent of the
    rated current = %0.1f Percent ",IcircA);
printf("\n Percent difference in secondary voltage =
    %0.1f Percent ",PD);
```

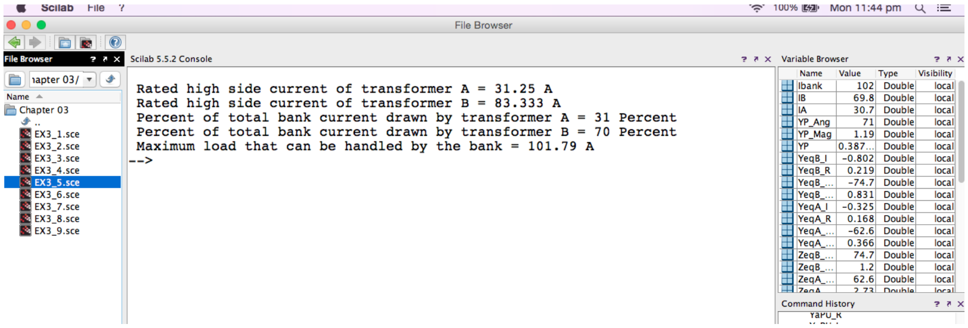

Figure 3.5: Determine Rated high side current of each transformer and Percent of the total bank current drawn by each transformer and Maximum load that can be handled by the bank without overloading by one of the transformer

Scilab code Exa 3.5 Determine Rated high side current of each transformer and Percent of the total bank current drawn by each transformer and Maximum load that can be handled by the bank without overloading by one of the transformer

```
// Example 3.5
// Determine (a) Rated high side current of each transformer (b) Percent of the
// total bank-current drawn by each transformer (c) Maximum load that can be
// handled by the bank without overloading by one of the transformer
// Page No. 107

clc;
clear;
close;

// Given data
SA=75000;                    // Transformer A rating
```

```
SB=200000;                 // Transformer B rating
VHSA=2400;                 // Voltage at the high side of transformer A
VHSB=2400;                 // Voltage at the high side of transformer B
RPUA=1.64;                 // Percent resistance of transformer A
XPUA=3.16;                 // Percent reactance of transformer A
RPUB=1.10;                 // Percent resistance of transformer B
XPUB=4.03;                 // Percent reactance of transformer B



// (a) Rated high side current of each transformer
IArated=SA/VHSA;           // High side rated current transformer A
IBrated=SB/VHSB;           // High side rated current transformer B

// (b) Percent of the total bank-current drawn by each transformer
ZAper=RPUA+%i*XPUA;        // Percent impadance for transformer A
 // Complex to Polar form...
ZAper_Mag=sqrt(real(ZAper)^2+imag(ZAper)^2);       // Magnitude part
ZAper_Ang=atan(imag(ZAper),real(ZAper))*180/%pi;   // Angle part

ZBper=RPUB+%i*XPUB;        // Percent impadance for transformer B
 // Complex to Polar form...
ZBper_Mag=sqrt(real(ZBper)^2+imag(ZBper)^2);       // Magnitude part
ZBper_Ang=atan(imag(ZBper),real(ZBper))*180/%pi;   //
```

```
     Angle part

ZAbase=VHSA/IArated;                  // Base
   impedance of transformer A
ZBbase=VHSB/IBrated;                  // Base
   impedance of transformer A

ZeqA_Mag=ZAbase*ZAper_Mag/100;        // Magnitude of
   equivalent impedance A
ZeqA_Ang=ZAper_Ang;                   // Angle of
   equivalent impedance A

ZeqB_Mag=ZBbase*ZBper_Mag/100;        // Magnitude of
   equivalent impedance B
ZeqB_Ang=ZBper_Ang;                   // Angle of
   equivalent impedance B

YeqA_Mag=1/ZeqA_Mag;                  // Magnitude of
   equivalent admittance A
YeqA_Ang=0-ZeqA_Ang;                  // Angle of
   equivalent admittance A

// Polar to Complex form
YeqA_R=YeqA_Mag*cos(-YeqA_Ang*%pi/180); // Real part
    of complex number
YeqA_I=YeqA_Mag*sin(YeqA_Ang*%pi/180); //Imaginary
   part of complex number

YeqB_Mag=1/ZeqB_Mag;                  // Magnitude of
   equivalent admittance B
YeqB_Ang=0-ZeqB_Ang;                  // Angle of
   equivalent admittance B

// Polar to Complex form

YeqB_R=YeqB_Mag*cos(-YeqB_Ang*%pi/180); // Real part
    of complex number
YeqB_I=YeqB_Mag*sin(YeqB_Ang*%pi/180); //Imaginary
```

```
part of complex number
YP=(YeqA_R - %i* YeqA_I)+(YeqB_R - %i* YeqB_I); // Parallel admittance

// Complex to Polar form...
YP_Mag=sqrt(real(YP)^2+imag(YP)^2);       // Magnitude part
YP_Ang=atan(imag(YP),real(YP))*180/%pi;  // Angle part

IA=YeqA_Mag/YP_Mag;                       // Transformer A load
IB=YeqB_Mag/YP_Mag;                       // Transformer A load
IA=IA*100;
IB=IB*100;

// (c) Maximum load that can be handled by the bank without overloading by
// one of the transformer
Ibank=IArated/0.307;

// Display result on command window

printf("\n Rated high side current of transformer A = %0.2f A ",IArated);
printf("\n Rated high side current of transformer B = %0.3f A ",IBrated);
printf("\n Percent of total bank current drawn by transformer A = %0.0f Percent ",IA);
printf("\n Percent of total bank current drawn by transformer B = %0.0f Percent ",IB);
printf("\n Maximum load that can be handled by the bank = %0.2f A ", Ibank);
```

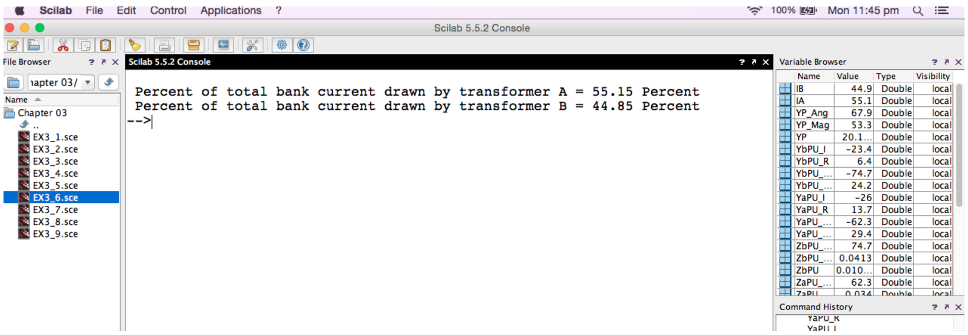

Figure 3.6: Determine the percent of the total bank current drawn by each transformer

Scilab code Exa 3.6 Determine the percent of the total bank current drawn by each transformer

```
// Example 3.6
// Determine the percent of the total bank-current drawn by each transformer
// Page No. 109

clc;
clear;
close;

// Given data
ZaPU_R=0.0158;            // Transformer A impedance real part
ZaPU_I=0.0301;            // Transformer A impedance imaginary part
ZbPU_R=0.0109;            // Transformer B impedance real part
ZbPU_I=0.0398;            // Transformer B impedance imaginary part
SB=200000;               // Transformer B rating
```

```
VHSA=2400;              // Voltage at the high side of transformer A
VHSB=2400;              // Voltage at the high side of transformer B
RPUA=1.64;              // Percent resistance of transformer A
XPUA=3.16;              // Percent reactance of transformer A
RPUB=1.10;              // Percent resistance of transformer B
XPUB=4.03;              // Percent reactance of transformer B



// Base impedance of transformer A
ZaPU=ZaPU_R+%i*ZaPU_I;
// Complex to Polar form...
ZaPU_Mag=sqrt(real(ZaPU)^2+imag(ZaPU)^2);       // Magnitude part
ZaPU_Ang=atan(imag(ZaPU),real(ZaPU))*180/%pi;   // Angle part

// Base impedance of transformer B
ZbPU=ZbPU_R+%i*ZbPU_I;
// Complex to Polar form...
ZbPU_Mag=sqrt(real(ZbPU)^2+imag(ZbPU)^2);       // Magnitude part
ZbPU_Ang=atan(imag(ZbPU),real(ZbPU))*180/%pi;   // Angle part

// Admittance of transformer A
YaPU_Mag=1/ZaPU_Mag;                    // Magnitude of equivalent admittance A
YaPU_Ang=0-ZaPU_Ang;                    // Angle of equivalent admittance A

// Polar to Complex form
```

```
YaPU_R=YaPU_Mag*cos(-YaPU_Ang*%pi/180); // Real part of complex number
YaPU_I=YaPU_Mag*sin(YaPU_Ang*%pi/180); //Imaginary part of complex number

// Admittance of transformer B
YbPU_Mag=1/ZbPU_Mag;                  // Magnitude of equivalent admittance B
YbPU_Ang=0-ZbPU_Ang;                  // Angle of equivalent admittance B
// Polar to Complex form

YbPU_R=YbPU_Mag*cos(-YbPU_Ang*%pi/180); // Real part of complex number
YbPU_I=YbPU_Mag*sin(YbPU_Ang*%pi/180); //Imaginary part of complex number

// Parallel admittance
YP=(YaPU_R-%i*YaPU_I)+(YbPU_R-%i*YbPU_I);
// Complex to Polar form...
YP_Mag=sqrt(real(YP)^2+imag(YP)^2);      // Magnitude part
YP_Ang=atan(imag(YP),real(YP))*180/%pi;  // Angle part

IA=YaPU_Mag/YP_Mag*100;                  // Percent current drawn by transformer A
IB=100-IA;

// Display the result on the command window
printf("\n Percent of total bank current drawn by transformer A = %0.2f Percent ",IA);
printf("\n Percent of total bank current drawn by transformer B = %0.2f Percent ",IB);
```

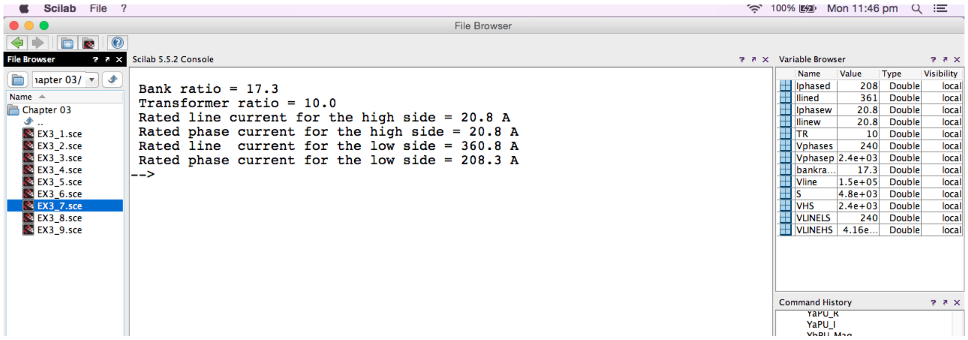

Figure 3.7: Computation of Bank ratio and Transformer ratio and Rated line and phase currents for the high side and Rated line and phase currents for the low side

Scilab code Exa 3.7 Computation of Bank ratio and Transformer ratio and Rated line and phase currents for the high side and Rated line and phase currents for the low side

```
// Example 3.7
// Computation of (a) Bank ratio (b) Transformer
   ratio (c) Rated line and phase
// currents for the high side (d) Rated line and
   phase currents for the low side
// Page No. 113

clc;
clear;
close;

// Given data
VLINEHS=4160;                  // Number of turns in
   the high side
VLINELS=240;                   // Number of turns in
```

```
the low side
VHS=2400;                       // Voltage at the high side
S=4800;                         // Supply voltage
Vline=150000;                   // Transformer rating

// (a) Bank ratio
bankratio=VLINEHS/VLINELS;

// (b) Transformer ratio
Vphasep= VLINEHS/  sqrt(3);    // For wye primary
Vphases=VLINELS                // For secondary
TR=Vphasep/Vphases;            // Transformer ratio

//(c) Rated line and phase currents for the high side
Ilinew=Vline/(sqrt(3)*VLINEHS);
Iphasew=Ilinew;

// (d) Rated line and phase currents for the low side
Ilined=Vline/(sqrt(3)*VLINELS);
Iphased=Ilined/sqrt(3);


// Display result on command window
printf("\n Bank ratio = %0.1f  ",bankratio);
printf("\n Transformer ratio = %0.1f  ",TR);
printf("\n Rated line current for the high side = %0.1f A ",Ilinew);
printf("\n Rated phase current for the high side = %0.1f A ",Iphasew);
printf("\n Rated line  current for the low side = %0.1f A ",Ilined);
printf("\n Rated phase current for the low side = %0.1f A ",Iphased);
```

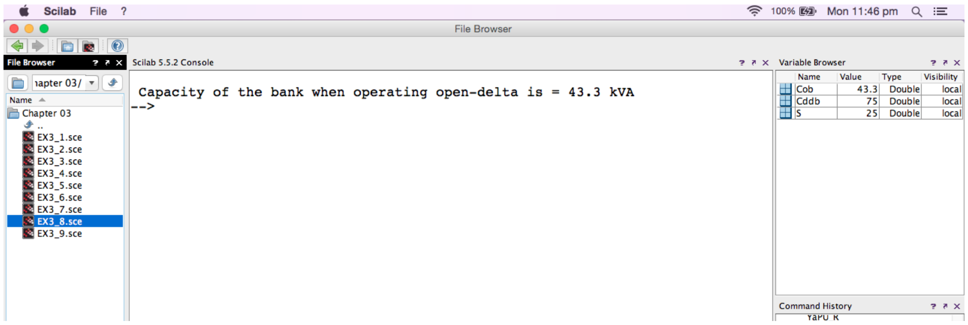

Figure 3.8: Determine the maximum allowable power that the open delta bank handle without overheating

Scilab code Exa 3.8 Determine the maximum allowable power that the open delta bank handle without overheating

```
// Example 3.8
// Determine the maximum allowable power that the
   open-delta bank handle
// without overheating
// Page No. 117

clc;
clear;
close;

// Given data
S=25;                          // Transformer rating

// Capacity of the delta-delta bank is
Cddb=S*3;
// Capacity of the bank when operating open-delta is
Cob=Cddb*0.577;
```

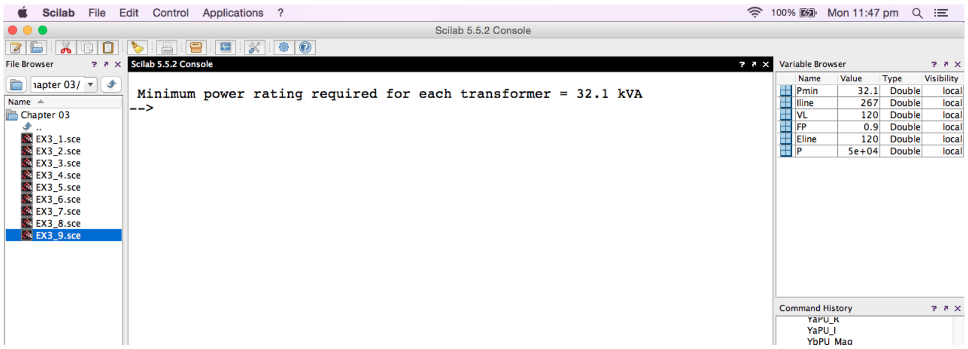

Figure 3.9: Determine the minimum power rating required for each transformer

```


// Display result on command window
printf("\n Capacity of the bank when operating open−
   delta is = %0.1f kVA ",Cob);
```

Scilab code Exa 3.9 Determine the minimum power rating required for each transformer

```
// Example 3.9
// Determine the minimum power rating required for
   each transformer
// Page No. 117

clc;
clear;
close;

// Given data
P=50000;                          // Transformer power
```

```
    rating
Eline=120;                          // Line voltage
FP=0.9                              // Power factor
    lagging
VL=120;

// Line current is
Iline=P/(sqrt(3)*Eline*FP);

// Minimum power rating required for each
    transformer
Pmin=VL*Iline/1000;


// Display result on command window
printf("\n Minimum power rating required for each
    transformer = %0.1f kVA ",Pmin);
```

Chapter 4

Principles of Three Phase Induction Motors

Scilab code Exa 4.1 Computation of synchronous speed of a six pole induction motor

```
// Example 4.1
// Computation of synchronous speed of a six pole
   induction motor
// Page No. 140

clc;
clear;
close;

// Given data
f=60;                    // Frequency
p=6;                     // Number of poles


fs=f*0.85;               // Frequency is 85% of its
   rated value
```

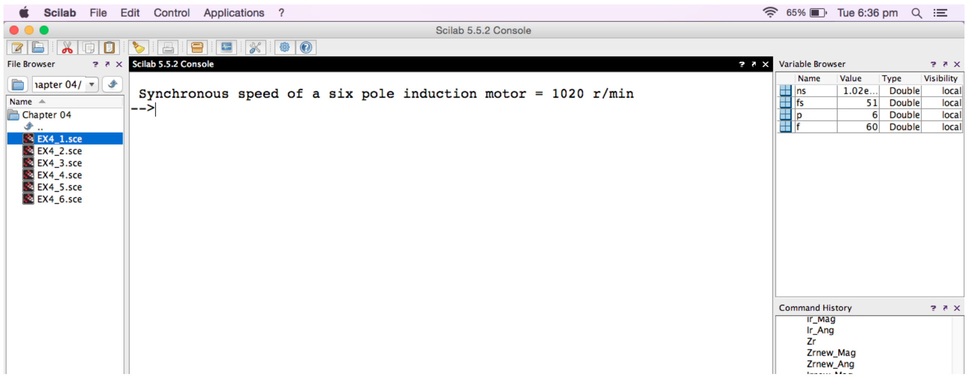

Figure 4.1: Computation of synchronous speed of a six pole induction motor

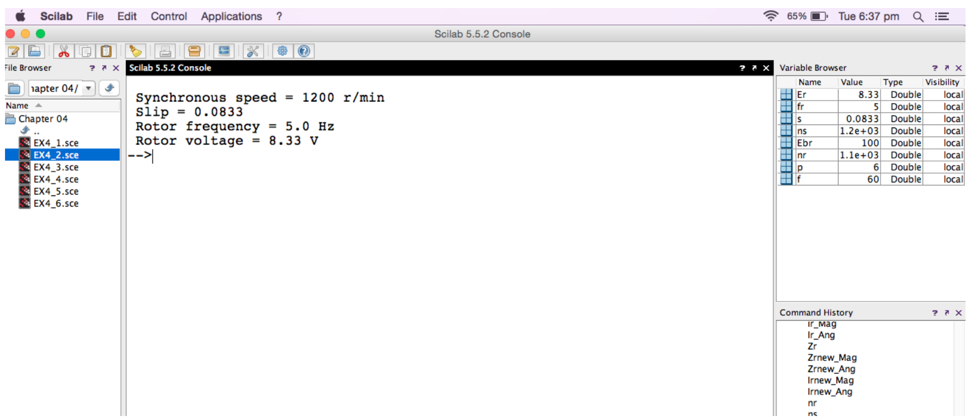

Figure 4.2: Computation of Frequency and Induced voltage of six pole induction motor

```
ns=120*fs/p;              // Synchronous speed

// Display result on command window
printf("\n Synchronous speed of a six pole induction
    motor = %0.0f r/min ",ns);
```

Scilab code Exa 4.2 Computation of Frequency and Induced voltage of six pole induction motor

```
// Example 4.2
// Computation of (a) Frequency (b) Induced voltage
   of six pole induction motor
// Page No. 143

clc;
clear;
close;

// Given data
f=60;                    // Frequency
p=6;                     // Number of poles
nr=1100;                 // Rotor speed
Ebr=100;                 // Blocked rotor voltage

// (a) Synchronous speed
ns=120*f/p;                  // Synchronous speed

// (b) Slip
s=(ns-nr)/ns;                  // Slip

// (c) Rotor frequency
fr=s*f;                  // Rotor frequency

// (d) Rotor voltage
Er=s*Ebr;                  // Rotor voltage


// Display result on command window
printf("\n Synchronous speed = %0.0f r/min ",ns);
printf("\n Slip = %0.4f  ",s);
printf("\n Rotor frequency = %0.1f Hz ",fr);
printf("\n Rotor voltage = %0.2f V ",Er);
```

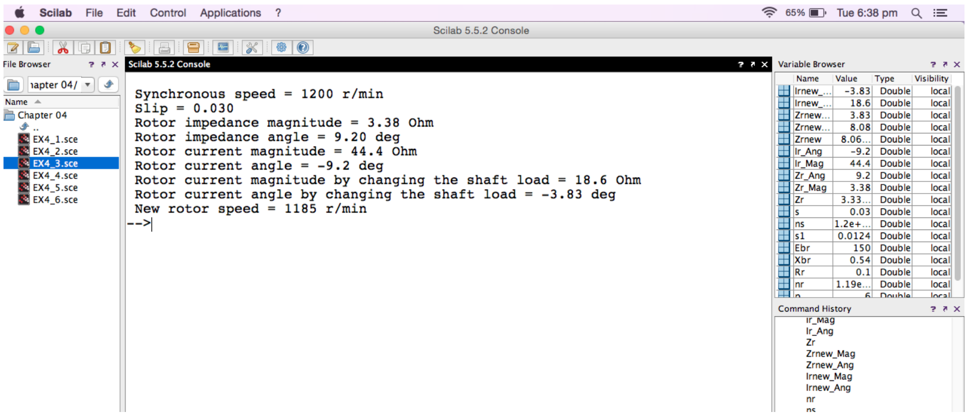

Figure 4.3: Determine Synchronous speed and Slip and Rotor impedance and Rotor current and Rotor current by changing the shaft load and Speed

Scilab code Exa 4.3 Determine Synchronous speed and Slip and Rotor impedance and Rotor current and Rotor current by changing the shaft load and Speed

```
// Example 4.3
// Determine (a) Synchronous speed (b) Slip (c) Rotor impedance (d) Rotor current
// (e) Rotor current if changing the shaft load resulted in 1.24 percenr slip
// (f) Speed for the condition in (e)
// Page No. 146

clc;
clear;
close;

// Given data
fs=60;                    // Frequency
```

```
p=6;                        // Number of poles
nr=1164;                    // Rotor speed
Rr=0.10;                    // Equivalent resistance
Xbr=0.54;                   // Equivalent reactance
Ebr=150;                    // Blocked rotor voltage per phase
s1=0.0124;                  // Percent slip

// (a) Synchronous speed
ns=120*fs/p;                    // Speed

// (b) Slip
s=(ns-nr)/ns;

// (c) Rotor impedance
Zr=(Rr/s)+%i*Xbr;
// Complex to Polar form...
Zr_Mag=sqrt(real(Zr)^2+imag(Zr)^2);          // Magnitude part
Zr_Ang=atan(imag(Zr),real(Zr))*180/%pi;      // Angle part

// (d) Rotor current
Ir_Mag=Ebr/Zr_Mag;                               // Magnitude
Ir_Ang=0-Zr_Ang;                                 // Angle

// (e) Rotor current if changing the shaft load resulted in 1.24 percent slip
Zrnew=Rr/s1+%i*Xbr;
// Complex to Polar form...
Zrnew_Mag=sqrt(real(Zrnew)^2+imag(Zrnew)^2);            // Magnitude part
Zrnew_Ang=atan(imag(Zrnew),real(Zrnew))*180/%pi;        // Angle part

Irnew_Mag=Ebr/Zrnew_Mag;                                 // Magnitude
```

```
Irnew_Ang=0-Zrnew_Ang;
                                        // Angle

// (f) Speed for the condition in (e)
nr=ns*(1-s1);

// Display result on command window
printf("\n Synchronous speed = %0.0f r/min ",ns);
printf("\n Slip = %0.3f ",s);
printf("\n Rotor impedance magnitude = %0.2f Ohm ",
    Zr_Mag);
printf("\n Rotor impedance angle = %0.2f deg ",
    Zr_Ang);
printf("\n Rotor current magnitude = %0.1f Ohm ",
    Ir_Mag);
printf("\n Rotor current angle = %0.1f deg ",Ir_Ang)
    ;
printf("\n Rotor current magnitude by changing the
    shaft load = %0.1f Ohm ",Irnew_Mag);
printf("\n Rotor current angle by changing the shaft
     load = %0.2f deg ",Irnew_Ang);
printf("\n New rotor speed = %0.0f r/min ",nr);
```

Scilab code Exa 4.4 Determine Total three phase apparent power crossing the air gap and Active and reactive components and Rotor power factor

```
// Example 4.4
// Determine (a) Total three phase apparent power
    crossing the air gap
// (b) Active and reactive components (c) Rotor
    power factor
// Page No. 149

```

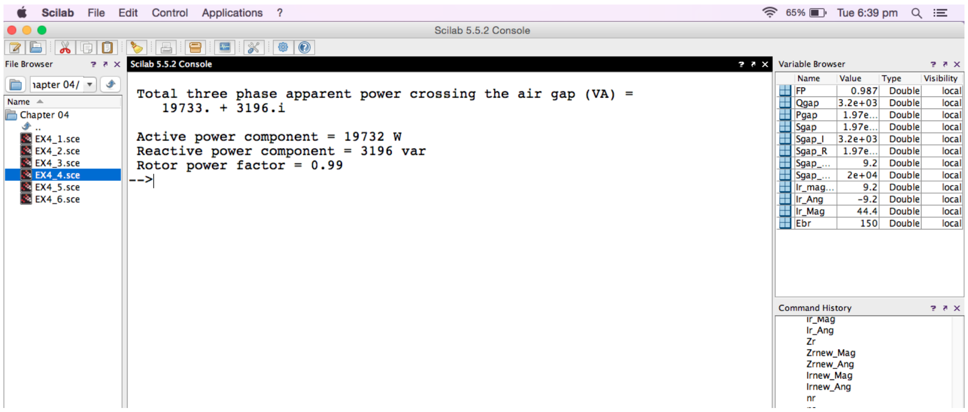

Figure 4.4: Determine Total three phase apparent power crossing the air gap and Active and reactive components and Rotor power factor

```
clc;
clear;
close;

// Given data
Ebr=150;                    // Blocked rotor voltage per phase
Ir_Mag=44.421;              // Rotor current magnitude
Ir_Ang=-9.2;                // Rotor current angle
Ir_magConj=9.2;


// (a) Total three phase apparent power crossing the air gap
Sgap_Mag=3*Ebr*Ir_Mag;        // Apparent power crossing the air gap magnitude
Sgap_Ang=Ir_magConj;          // Apparent power crossing the air gap angle

// Polar to Complex form
Sgap_R=Sgap_Mag*cos(-Sgap_Ang*%pi/180); // Real part of complex number
Sgap_I=Sgap_Mag*sin(Sgap_Ang*%pi/180); //Imaginary
```

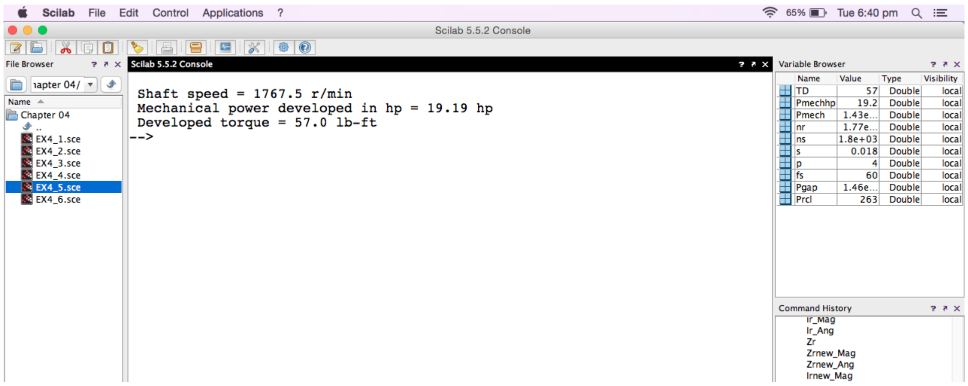

Figure 4.5: Computation of Shaft speed and Mechanical power developed and Developed torque

```
    part of complex number
Sgap=ceil(Sgap_R)+%i*ceil(Sgap_I);

// (b) Active and reactive components
Pgap=Sgap_R;                    // Active power component
Qgap=Sgap_I;                    // Reactive power
    component

// (c) Rotor power factor
FP=cosd(Ir_magConj);

// Display result on command window
printf("\n Total three phase apparent power crossing
     the air gap (VA) =");
disp(Sgap);
printf("\n Active power component = %0.0f W",Pgap);
printf("\n Reactive power component = %0.0f var ",
    Qgap);
printf("\n Rotor power factor = %0.2f ",FP);
```

Scilab code Exa 4.5 Computation of Shaft speed and Mechanical power developed and Developed torque

```
// Example 4.5
// Computation of (a) Shaft speed (b) Mechanical power developed
// (c) Developed torque
// Page No. 152

clc;
clear;
close;

// Given data
Prcl=263;                // Rotor copper loss
Pgap=14580;              // Power input to the rotor
fs=60;                   // Frequency
p=4;                     // Number of poles



// (a) Shaft speed
s=Prcl/Pgap;                // Slip
ns=120*fs/p;                // Speed of stator
nr=ns*(1-s);                // Speed of shaft

// (b) Mechanical power developed
Pmech=Pgap-Prcl;                     // Mechanical power developed
Pmechhp=Pmech/746;                   // Mechanical power developed in hp

//(c) Developed torque
TD=5252*Pmechhp/nr;


// Display result on command window
printf("\n Shaft speed = %0.1f r/min ",nr);
```

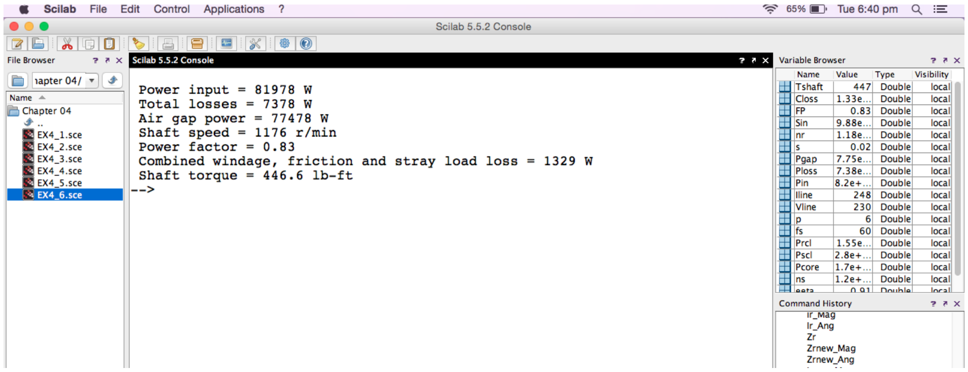

Figure 4.6: Determine Power input and Total losses and Air gap power and Shaft speed and Power factor and Combined windage friction and stray load loss and Shaft torque

```
printf("\n Mechanical power developed in hp = %0.2f hp ",Pmechhp);
printf("\n Developed torque = %0.1f lb-ft ",TD);
```

Scilab code Exa 4.6 Determine Power input and Total losses and Air gap power and Shaft speed and Power factor and Combined windage friction and stray load loss and Shaft torque

```
// Example 4.6
// Determine (a) Power input (b) Total losses (c) Air gap power (d) Shaft speed
// (e) Power factor (f) Combined windage, friction and stray load loss
// (g) Shaft torque
// Page No. 159

clc;
clear;
close;
```

```

// Given data
Pshaft=74600;                    // Shaft power
eeta=0.910;                      // Rated efficiency
ns=1200;                         // Speed of stator
Pcore=1697;                      // Power in core
Pscl=2803;                       // Stator copper loss
Prcl=1549;                       // Rotor copper loss
fs=60;                           // Synchronous frequency
p=6;                             // Number of poles
Vline=230;                       // Line voltage
Iline=248;                       // Line current

// (a) Power input
Pin=Pshaft/eeta;                 // Parallel resistance

// (b) Total losses
Ploss=Pin-Pshaft;

//(c) Air gap power
Pgap=Pin-Pcore-Pscl;

// (d) Shaft speed
s=Prcl/Pgap;        // Parallel resistance
ns=120*fs/p;
nr=ns*(1-s);

// (e) Power factor
Sin=sqrt(3)*Vline*Iline;
FP=Pin/Sin;

//(f) Combined windage, friction and stray load loss
Closs=Ploss-Pcore-Pscl-Prcl;

//(g) Shaft torque
Tshaft=5252*100/nr;

```

```
// Display result on command window
printf("\n Power input = %0.0f W",Pin);
printf("\n Total losses = %0.0f W",Ploss);
printf("\n Air gap power = %0.0f W ",Pgap);
printf("\n Shaft speed = %0.0f r/min ",nr);
printf("\n Power factor = %0.2f ",FP);
printf("\n Combined windage, friction and stray load
    loss = %0.0f W ",Closs);
printf("\n Shaft torque = %0.1f lb-ft ",Tshaft);
```

Chapter 5

Classification Performance Applications and Operation of Three Phase Induction Machines

Scilab code Exa 5.1 Computation of minimum value of Locked rotor torque and Breakdown torque and Pull up torque

```
// Example 5.1
// Computation of minimum value of (a) Locked rotor
   torque (b) Breakdown torque
// (c) Pull up torque
// Page No. 173

clc;
clear;
close;

// Given data
f=60;                  // Frequency in Hz
```

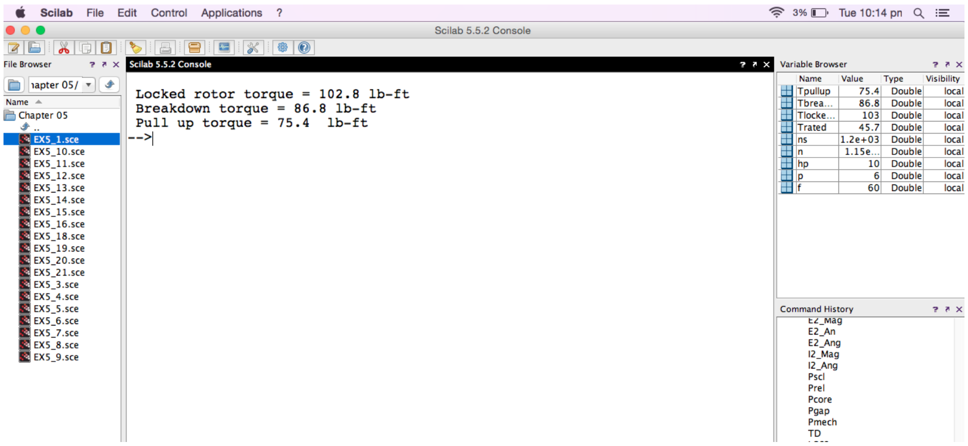

Figure 5.1: Computation of minimum value of Locked rotor torque and Breakdown torque and Pull up torque

```
p=6;                    // Number of poles
hp=10;                  // Horsepower
n=1150;                 // Rated speed of machine
ns=120*f/p;


// (a) Locked rotor torque
Trated=hp*5252/n;       // Rated torque
Tlockedrotor=2.25*Trated;

// (b) Breakdown torque
Tbreakdown=1.90*Trated;

// (c) Pull up torque
Tpullup=1.65*Trated;


// Display result on command window
printf("\n Locked rotor torque = %0.1f lb-ft ",
    Tlockedrotor);
printf("\n Breakdown torque = %0.1f lb-ft ",
    Tbreakdown);
```

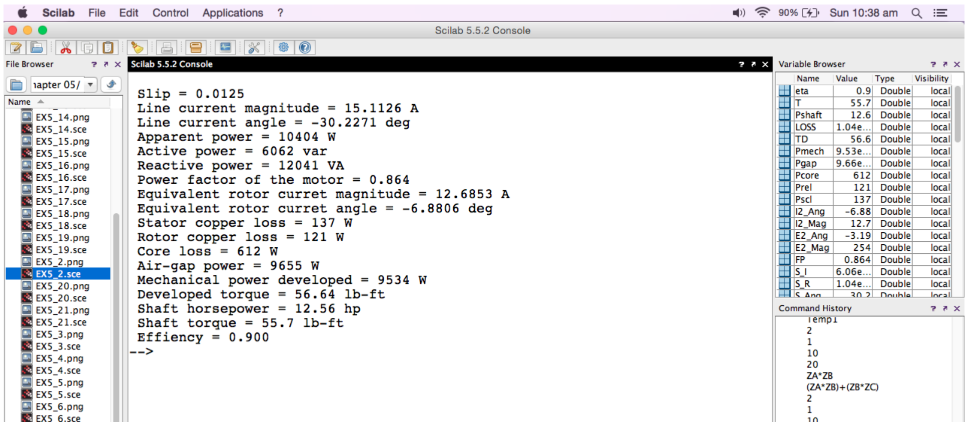

Figure 5.2: Determine Slip and Line current and Apparent power active power reactive power and power factor of the motor and Equivalent rotor current and Stator copper loss and Rotor copper loss and Core loss and Air gap power and Mechanical power developed

```
printf("\n Pull up torque = %0.1f  lb-ft",Tpullup);
```

Scilab code Exa 5.2 Determine Slip and Line current and Apparent power active power reactive power and power factor of the motor and Equivalent rotor current and Stator copper loss and Rotor copper loss and Core loss and Air gap power and Mechanical power developed

```
// Example 5.2
// Determine (a) Slip (b) Line current (c) Apparent
    power, active power,
// reactive power and power factor of the motor (d)
    Equivalent rotor curret
// (e) Stator copper loss (f) Rotor copper loss (g)
    Core loss (h) Air-gap
// power (i) Mechanical power developed (j)
    Developed torque (k) Shaft
```

```
// horsepower (l) Shaft torque (m) Effiency
// Page No. 180

clc;
clear;
close;

// Given data
f=60;                       // Frequency
P=6;                        // Number of poles
nr=1185;
R1=0.200;                   // Motor resistance
R2=0.250;
X1=1.20;                    // Motor reactance
X2=1.29;
Rfe=317;                    // Field resistance
XM=42;                      // Motor reactance
V=460;                      // Voltage rating
PFPS=166;                   // Stray loss

// (a) Slip
ns=(120*f)/P;
s=(ns-nr)/ns;                // Speed difference

// (b) Line current
Z2=(R2/s)+%i*X2;
// Complex to Polar form...
Z2_Mag=sqrt(real(Z2)^2+imag(Z2)^2);        //
    Magnitude part
Z2_Ang = atan(imag(Z2),real(Z2))*180/%pi; // Angle
    part

Z0_Num_Mag=Rfe*XM;    // Z0 numerator
Z0_Num_Ang=0+90;

Z0_Den_R=Rfe;        // Z0 denominator
Z0_Den_I=XM;
Z0_Den=Z0_Den_R+%i*Z0_Den_I;
```

```
// Complex to Polar form...
Z0_Den_Mag=sqrt(real(Z0_Den)^2+imag(Z0_Den)^2);            // Magnitude part
Z0_Den_Ang = atan(imag(Z0_Den),real(Z0_Den))*180/%pi; // Angle part

Z0_Mag=Z0_Num_Mag/Z0_Den_Mag;                    // Magnitude of Z0
Z0_Ang=Z0_Num_Ang-Z0_Den_Ang;                    // Angle of Z0

// Polar to Complex form
Z0_R=Z0_Mag*cos(-Z0_Ang*%pi/180);          // Real part of complex number
Z0_I=Z0_Mag*sin(Z0_Ang*%pi/180);           // Imaginary part of complex number

// ZP computation
ZP_Num_Mag=Z2_Mag*Z0_Mag;                       // ZP numerator magnitude
ZP_Num_Ang=Z2_Ang+Z0_Ang;                       // ZP numerator angle

ZP_Den_R=real(Z2)+Z0_R;                         // Real part of ZP denominator
ZP_Den_I=imag(Z2)+Z0_I;
ZP_Den=ZP_Den_R+%i*ZP_Den_I;                    // ZP in complex form

// Complex to Polar form...
ZP_Den_Mag=sqrt(real(ZP_Den)^2+imag(ZP_Den)^2);            // Magnitude part
ZP_Den_Ang = atan(imag(ZP_Den),real(ZP_Den))*180/%pi; // Angle part

ZP_Mag=ZP_Num_Mag/ZP_Den_Mag;                   // Final vlaue of ZP in polar form
ZP_Ang=ZP_Num_Ang-ZP_Den_Ang;
```

```
// Polar to Complex form
ZP_R=ZP_Mag*cos(-ZP_Ang*%pi/180);        // Real part
    of complex number
ZP_I=ZP_Mag*sin(ZP_Ang*%pi/180);         // Imaginary
    part of complex number

// Zin computation
ZP=ZP_R+%i*ZP_I;                         // Parallel
    impedance
Z1=R1+%i*X1;
Zin=Z1+ZP;                               // Input
    impedance
// Complex to Polar form...
Zin_Mag=sqrt(real(Zin)^2+imag(Zin)^2);         //
    Magnitude part
Zin_Ang = atan(imag(Zin),real(Zin))*180/%pi; //
    Angle part

// I1 computation
I1_Mag=(V/sqrt(3))/Zin_Mag;             // I1
    magnitude
I1_Ang=0-Zin_Ang;                       // I1 angle

// (c) Apparent power, active power, reactive power
    and power factor of the motor
S_Mag=3*(V/sqrt(3))*I1_Mag;            // S magnitude
S_Ang=0-(-Zin_Ang);                    // S angle

// Polar to Complex form
S_R=S_Mag*cos(-S_Ang*%pi/180);         // Real part of
     complex number
S_I=S_Mag*sin(S_Ang*%pi/180);          // Imaginary
    part of complex number

FP=cosd(S_Ang);                        // Power factor

// (d) Equivalent rotor curret
E2_Mag=I1_Mag*ZP_Mag;                  // E2 magnitude
```

```
E2_Ang=I1_Ang+ZP_Ang;                          // E2 angle

I2_Mag=E2_Mag/Z2_Mag;                          // I2 magnitude
I2_Ang=E2_Ang-Z2_Ang;                           // I2 angle

// (e) Stator copper loss
Pscl=3*I1_Mag^2*R1;

// (f) Rotor copper loss
Prel=3*I2_Mag^2*R2;

// (g) Core loss
Pcore=3*(E2_Mag^2/Rfe);

// (h) Air-gap power
Pgap=Prel/s;

// (i) Mechanical power developed
Pmech=Prel*(1-s)/s;

// (j) Developed torque
TD=(21.12*I2_Mag^2*R2)/(s*ns);

// (k) Shaft horsepower
LOSS=Pscl+Prel+Pcore+PFPS;
Pshaft=(S_R-LOSS)/746;

// (l) Shaft torque
T=5252*Pshaft/nr;

// (m) Effiency
eta=Pshaft/S_R*746;

// Display result on command window
printf("\n Slip = %0.4f ",s);
printf("\n Line current magnitude = %0.4f A",I1_Mag)
    ;
printf("\n Line current angle = %0.4f deg",I1_Ang);
```

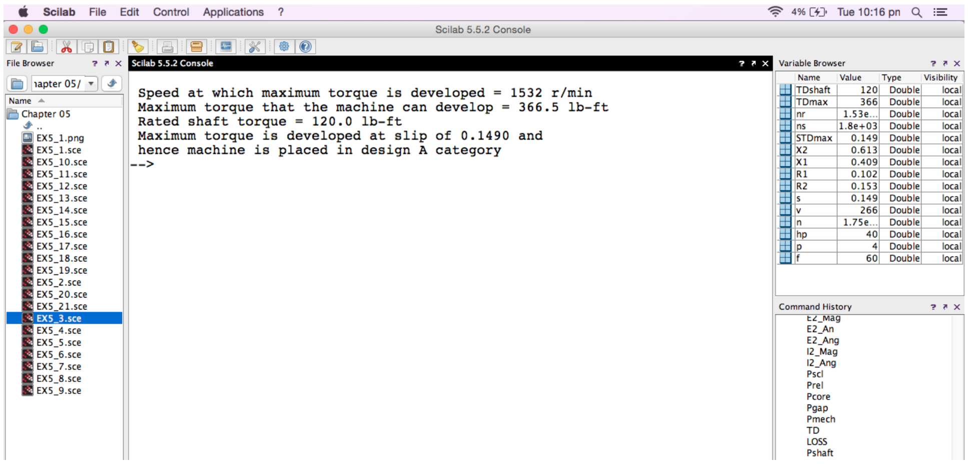

Figure 5.3: Computation of Speed at which maximum torque is developed and Maximum torque that the machine can develop and Rated shaft torque

```
printf("\n Apparent power = %0.0f W",S_R);
printf("\n Active power = %0.0f var",S_I);
printf("\n Reactive power = %0.0f VA",S_Mag);
printf("\n Power factor of the motor = %0.3f ",FP);
printf("\n Equivalent rotor curret magnitude = %0.4f
    A",I2_Mag);
printf("\n Equivalent rotor curret angle = %0.4f deg
    ",I2_Ang);
printf("\n Stator copper loss = %0.0f W",Pscl);
printf("\n Rotor copper loss = %0.0f W",Prel);
printf("\n Core loss = %0.0f W",Pcore);
printf("\n Air-gap power = %0.0f W",Pgap);
printf("\n Mechanical power developed = %0.0f W",
    Pmech);
printf("\n Developed torque = %0.2f lb-ft",TD);
printf("\n Shaft horsepower = %0.2f hp",Pshaft);
printf("\n Shaft torque = %0.1f lb-ft",T);
printf("\n Effiency = %0.3f",eta);
```

Scilab code Exa 5.3 Computation of Speed at which maximum torque is developed and Maximum torque that the machine can develop and Rated shaft torque

```
// Example 5.3
// Computation of (a) Speed at which maximum torque
   is developed (b) Maximum
// torque that the machine can develop (c) Rated
   shaft torque (d) Which NEMA
// design fits this motor?
// Page No. 184

clc;
clear;
close;

// Given data
f=60;                       // Frequency in Hz
p=4;                        // Number of poles
hp=40;                      // Horsepower
n=1751;                     // Rated speed of machine
v=460/sqrt(3);            // Voltage
s=0.1490;                   // Slip
R2=0.153;                   // Rotor resistance
R1=0.102;
X1=0.409;                   // Rotor reactance
X2=0.613;

// (a) Speed at which maximum torque is developed
STDmax=R2/(sqrt(R1^2+(X1+X2)^2));
ns=120*f/p;                                  //stator spped
nr=ns*(1-s);

// (b) Maximum torque that the machine can develop
```

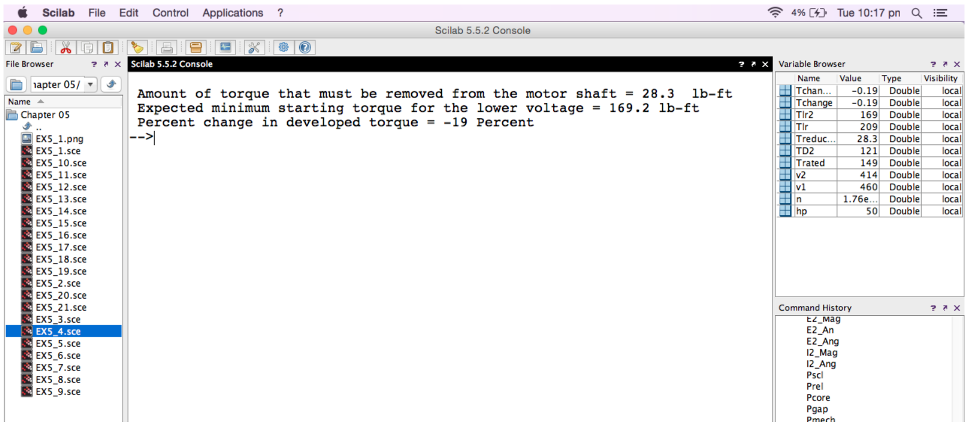

Figure 5.4: Computation of Amount of torque that must be removed from the motor shaft to maintain 1760 rpm and Expected minimum starting torque for the lower voltage and Percent change in developed torque

```
TDmax=(21.12*v^2)/(2*ns*(sqrt(R1^2+(X1+X2)^2)+R1));

// (c) Rated shaft torque
TDshaft=hp*5252/n;

// Display result on command window
printf("\n Speed at which maximum torque is
    developed = %0.0f r/min ",nr);
printf("\n Maximum torque that the machine can
    develop = %0.1f lb-ft ",TDmax);
printf("\n Rated shaft torque = %0.1f lb-ft ",
    TDshaft);
printf("\n Maximum torque is developed at slip of
    0.1490 and \n hence machine is placed in design A
     category ");
```

Scilab code Exa 5.4 Computation of Amount of torque that must be removed from the motor shaft to maintain 1760 rpm and Expected minimum starting torque for the lower voltage and Percent change in developed torque

```
// Example 5.4
// Computation of (a) Amount of torque that must be
   removed from the motor
// shaft to maintain 1760r/min (b) Expected minimum
   startimg torque for the
// lower voltage (c) Percent change in developed
   torque caused by 10% drop in
// system voltage.
// Page No. 185

clc;
clear;
close;

// Given data

hp=50;                  // Horsepower
n=1760;                 // Rated speed of machine
v1=460;


// (a) Amount of torque that must be removed from
   the motor shaft to maintain
// 1760r/min
v2=v1*0.90;
Trated=hp*5252/n;        //Rated torque
TD2=Trated*(v2/v1)^2;
Treduction=Trated-TD2;

// (b) Expected minimum startimg torque for the
   lower voltage

Tlr=1.40*Trated;
Tlr2=Tlr*(v2/v1)^2;
```

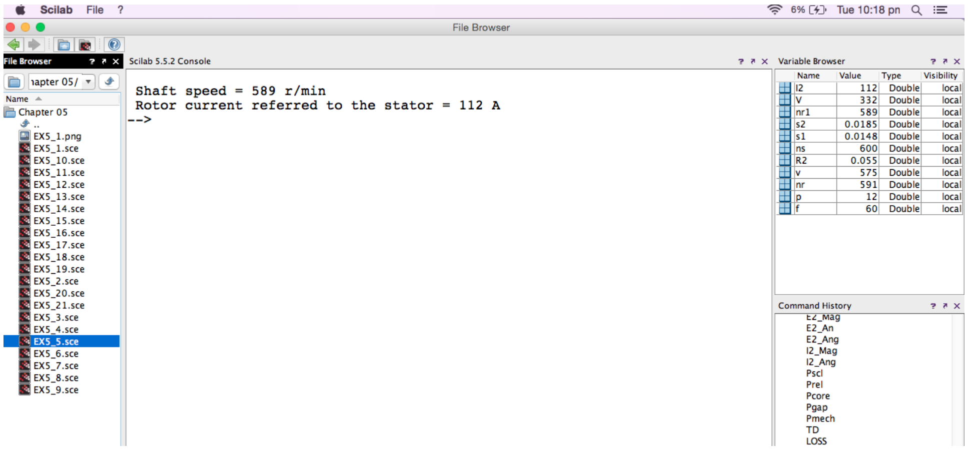

Figure 5.5: Computation of minimum value of Shaft speed and Rotor current referred to the stator

```

// (c) Percent change in developed torque caused by 10% drop in system voltage

Tchange=(TD2-Trated)/Trated;
Tchanger=(Tlr2-Tlr)/Tlr;

// Display result on command window
printf("\n Amount of torque that must be removed from the motor shaft = %0.1f lb-ft",Treduction);
printf("\n Expected minimum starting torque for the lower voltage = %0.1f lb-ft ",Tlr2);
printf("\n Percent change in developed torque = %0.0f Percent ",Tchanger*100);
```

Scilab code Exa 5.5 Computation of minimum value of Shaft speed and Rotor current referred to the stator

```
// Example 5.5
// Computation of minimum value of (a) Shaft speed (
    b) Rotor current referred
// to the stator
// Page No. 187

clc;
clear;
close;

// Given data
f=60;                    // Frequency in Hz
p=12;                    // Number of poles
nr=591.1;                // Rated speed of machine
v=575;                   // Voltage rating of the machine
R2=0.055;

// (a) Shaft speed
ns=120*f/p;             // Speed (r/min)
s1=(ns-nr)/ns;          // Slip 1
s2=1.25*s1;             // Slip 2
nr1=ns*(1-s2);

// (b) Rotor current referred to the stator
V=v/sqrt(3);
I2=V*s2/R2;

// Display result on command window
printf("\n Shaft speed = %0.0f r/min ",nr1);
printf("\n Rotor current referred to the stator = %0
    .0f A ",I2);
```

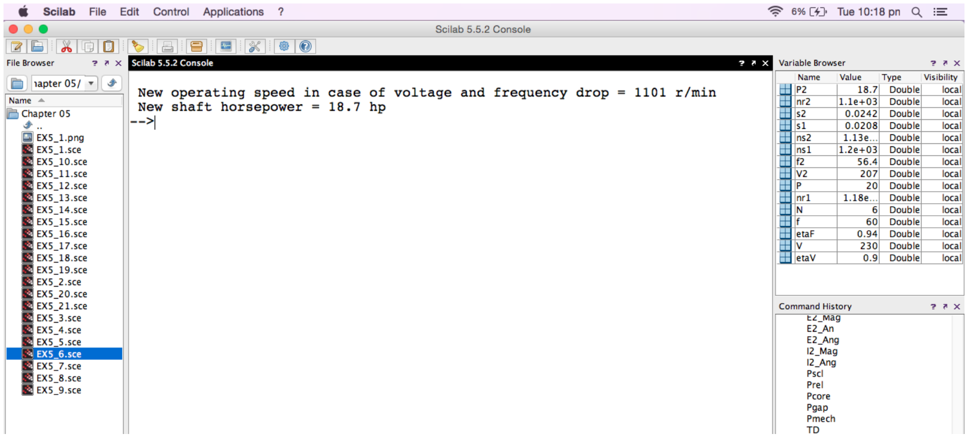

Figure 5.6: Determine new operating speed of a system and New shaft horsepower

Scilab code Exa 5.6 Determine new operating speed of a system and New shaft horsepower

```
// Example 5.6
// Determine (a) New operating speed if a system disturbance causes a 10% drop
// in voltage and 6% drop in frequency (b) New shaft horsepower.
// Page No. 190

clc;
clear;
close;

// Given data
etaV=0.90;                // Efficiency related to voltage
V=230;                    // Voltage
etaF=0.94;                // Efficiency related to voltage
f=60;                     // Frequency
N=6;                      // Number of poles
```

```
nr1=1175;                    // Speed of motor
P=20;                        // Horsepower of motor

// (a)  New operating speed if a system disturbance causes a 10% drop in
// voltage and 6% drop in frequency
V2=etaV*V;                  // New voltage after 10% drop
f2=etaF*f;                  // New frequency after 6% drop
ns1=120*f/N;
ns2=120*0.94*f/N;
s1=(ns1-nr1)/ns1;           // Speed difference

s2=s1*((V/V2)^2)*(f2/f);
nr2=ns2*(1-s2);             // New speed

// (b) New shaft horsepower
P2=P*(nr2/nr1);             // With a constant torque load T2=T1

// Display result on command window
printf("\n New operating speed in case of voltage and frequency drop = %0.0f r/min ",nr2);
printf("\n New shaft horsepower = %0.1f hp ",P2);
```

Scilab code Exa 5.7 Determine expected locked rotor line current

```
// Example 5.7
// Determine expected locked-rotor line current
// Page No. 192

clc;
```

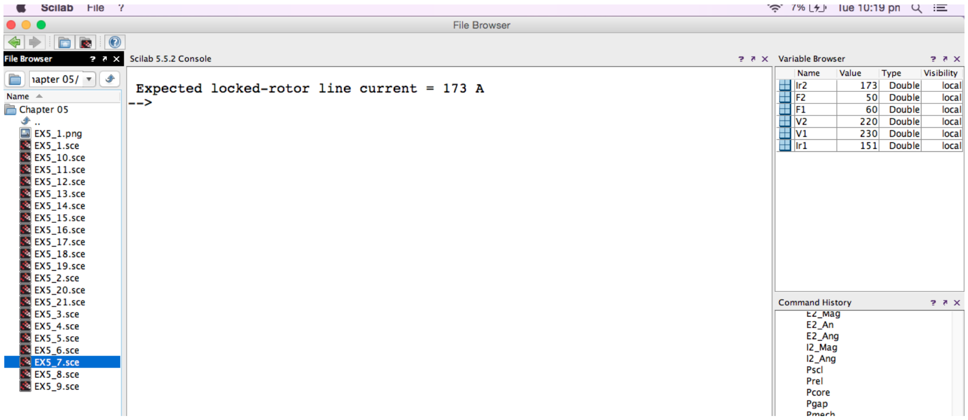

Figure 5.7: Determine expected locked rotor line current

```
clear;
close;

// Given data
Ir1=151;                    // Rated current
V1=230;                     // Rated voltage
V2=220;                     // Motor starting voltage
F1=60;                      // Rated frequency
F2=50;                      // Motor starting
   frequency

// Expected locked-rotor line current
Ir2=Ir1*((V2/F2)/(V1/F1));

// Display result on command window
printf("\n Expected locked-rotor line current = %0.0
   f A ",Ir2);
```

Scilab code Exa 5.8 Determine expected minimum locked rotor torque

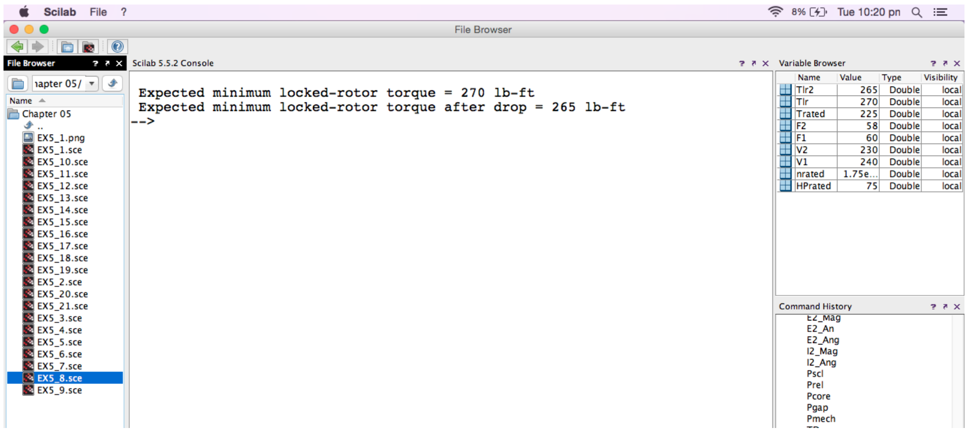

Figure 5.8: Determine expected minimum locked rotor torque and Repeat when voltage and frequency dropped to 230V and 58Hz

and Repeat when voltage and frequency dropped to 230V and 58Hz

```
// Example 5.8
// Determine (a) Expected minimum locked-rotor torque (b) Repeat (a) when
// voltage and frequency dropped to 230V and 58Hz
// Page No. 193

clc;
clear;
close;

// Given data
HPrated=75;                // Rated horsepower
nrated=1750;               // Rated speed
V1=240;                    // Rated voltage
V2=230;                    // Voltage after drop
F1=60;                     // Rated frequency
F2=58;                     // Frequency after drop

// (a) Expected minimum locked-rotor torque
Trated=5252*HPrated/nrated;     // Rated torque
```

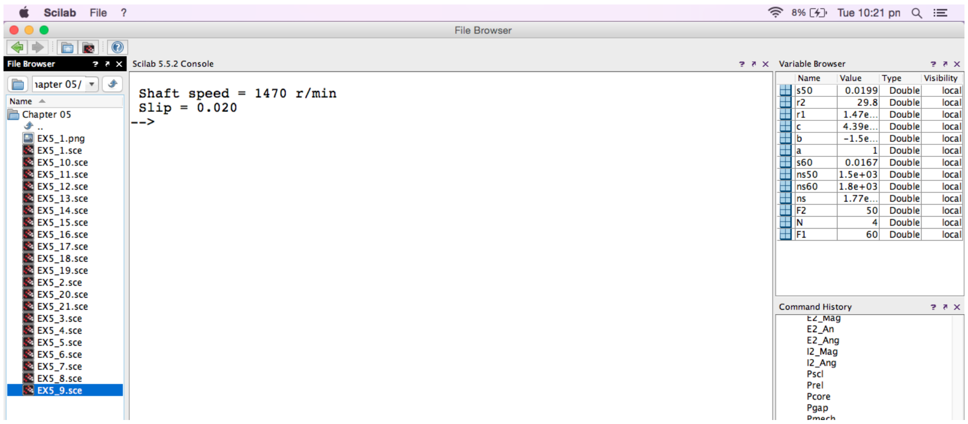

Figure 5.9: Determine shaft rpm and Slip

```
Tlr=Trated*1.2;                    // Minimum locked-rotor torque is 120% rated

// (b) Expected minimum locked-rotor torque when voltage and frequency dropped
// to 230V and 58Hz
Tlr2=Tlr*((V2/F2)^2)*((F1/V1)^2);

// Display result on command window
printf("\n Expected minimum locked-rotor torque = %0.0f lb-ft",Tlr);
printf("\n Expected minimum locked-rotor torque after drop = %0.0f lb-ft",Tlr2);
```

Scilab code Exa 5.9 Determine shaft rpm and Slip

```
// Example 5.9
// Determine (a) Shaft r/min (b) Slip
// Page No. 194
```

```

clc;
clear;
close;

// Given data
F1=60;              // Rated frequency
N=4;                // Number of poles
F2=50;              // New frequency
ns=1770;            // Rated speed

// (a) Shaft r/min
ns60=120*F1/N;       // Speed at rated ferquency
ns50=120*F2/N;       // Speed at 50 Hz frequency
s60=(ns60-ns)/ns60;  // Slip at 60 Hz frequency

// Using eq. (5.16) and by solving..s50=29.251/nr50
// Using eq. (4.3) and solving for nr50 we get the quadratic equation..
// Using various values of quadratic equations, we have
a=1;
b=-1500;
c=43876.5;
r1=(-b+sqrt(b^2-4*a*c))/(2*a); // Root 1

r2=(-b-sqrt(b^2-4*a*c))/(2*a); // Root 2
// Answer 'r2' is not valid

// (b) Slip
s50=(ns50-r1)/ns50;

// Display result on command window
printf("\n Shaft speed = %0.0f r/min",r1);
printf("\n Slip = %0.3f ",s50);
```

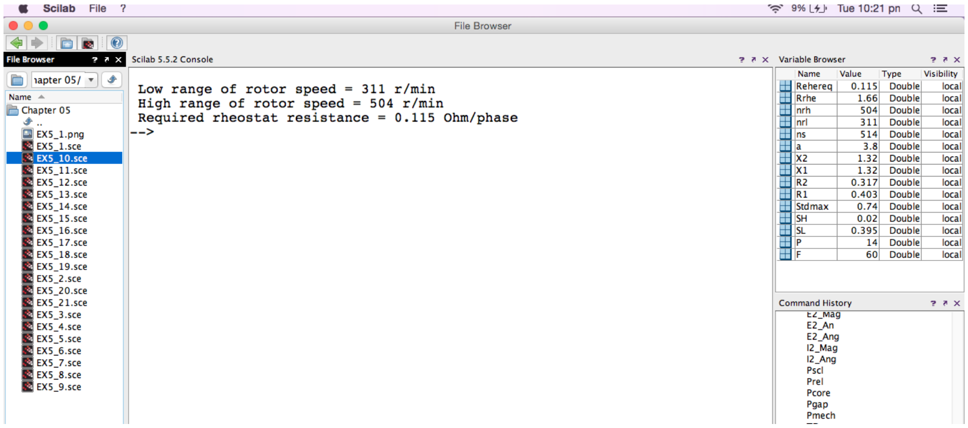

Figure 5.10: Determine range of rotor speed and Required rheostat resistance

Scilab code Exa 5.10 Determine range of rotor speed and Required rheostat resistance

```
// Example 5.10
// Determine (a) Range of rotor speed (b) Required
    rheostat resistance
// Page No. 198

clc;
clear;
close;

// Given data
F=60;                    // Frequency of motor
P=14;                    // Number of poles
SL=0.395;                // Low speed point
SH=0.02;                 // High speed point
Stdmax=0.74;             // Value at which TD is
```

```
   maximum (from curve B)
R1=0.403;                    // Motor resistance
R2=0.317;
X1=1.32;                     // Motor reactance
X2=1.32;
a=3.8;                       // Ratio of stator turns/
   phase to rotor turns/phase

// (a) Range of rotor speed
ns=120*F/P;                 // Speed
nrl=ns*(1-SL);              // Rotor low speed
nrh=ns*(1-SH);              // Rotor high speed

// (b) Required rheostat resistance
Rrhe=Stdmax*(sqrt(R1^2+(X1+X2)^2))-R2;
Rehereq=Rrhe/a^2;

// Display result on command window
printf("\n Low range of rotor speed = %0.0f r/min",
   nrl);
printf("\n High range of rotor speed = %0.0f r/min",
   nrh);
printf("\n Required rheostat resistance = %0.3f Ohm/
   phase",Rehereq);
```

Scilab code Exa 5.11 Determine rotor frequency and Slip at which TDmax occurs and Rotor speed at half rated torque load and Required rheostat resistance and Rated torque

```
// Example 5.11
// Determine (a) Rotor frequency (b) Slip at which
   TDmax occurs (c) Rotor speed
// at 1/2 rated torque load (d) Required rheostat
```

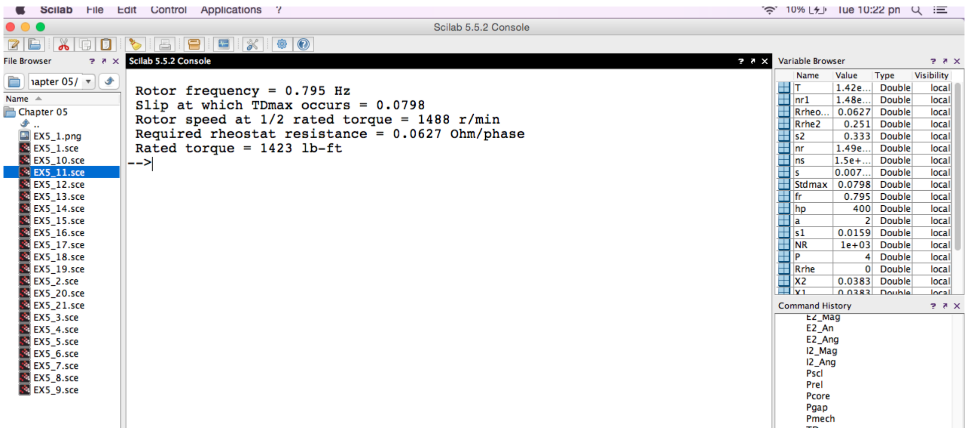

Figure 5.11: Determine rotor frequency and Slip at which TDmax occurs and Rotor speed at half rated torque load and Required rheostat resistance and Rated torque

```
   resistance (e) Rated torque
// Page No. 201

clc;
clear;
close;

// Given data
S=0.0159;                  // Slip
Fbr=50;                    // Rated frequency
R1=0.00536;                // Motor resistance
R2=0.00613;
X1=0.0383;                 // Motor reactance
X2=0.0383;
Rrhe=0;                    // Initial rheostat
   resistance
P=4;                       // Number of poles
NR=1000;                   // Rated speed
s1=0.0159;                 // Slip of rheostat
a=2;                       // Stator to rotor turns
   ratio
```

```
hp=400;                     // Motor horsepower

// (a) Rotor frequency
fr=S*Fbr;

// (b) Slip at which TDmax occurs
Stdmax=(R2+Rrhe)/(sqrt(R1^2+(X1+X2)^2));

// (c) Rotor speed at 1/2 rated torque load
s=S*(0.5)*(R2/R2);   // Rotor speed at 1/2 rated torque
ns=120*Fbr/P;
nr=ns*(1-s);              // Rotor speed

// (d) Required rheostat resistance
s2=(ns-NR)/ns;
Rrhe2=((s2/s1)*(1/0.5)*(R2+Rrhe))-R2;   // rheostat resistance
Rrheostat=Rrhe2/a^2;

// (e) Rated torque
nr1=ns*(1-s1);            // Rated speed
T=hp*5252/nr1;



// Display result on command window
printf("\n Rotor frequency = %0.3f Hz",fr);
printf("\n Slip at which TDmax occurs = %0.4f ",Stdmax);
printf("\n Rotor speed at 1/2 rated torque = %0.0f r/min",nr);
printf("\n Required rheostat resistance = %0.4f Ohm/phase",Rrheostat);
printf("\n Rated torque = %0.0f lb-ft",T);
```

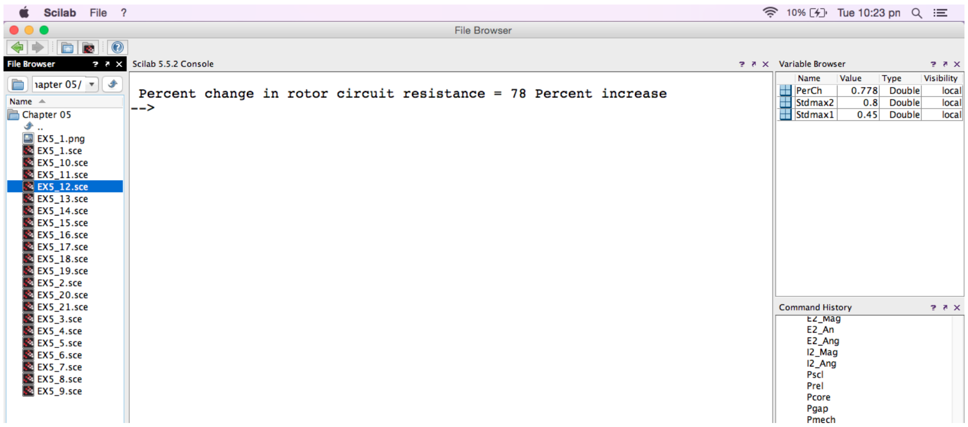

Figure 5.12: Determine the percent increase or decrease in rotor circuit resistance

Scilab code Exa 5.12 Determine the percent increase or decrease in rotor circuit resistance

```
// Example 5.12
// Determine the percent increase or decrease in
   rotor circuit resistance
// Page No. 202

clc;
clear;
close;

// Given data

Stdmax1=0.45;          // Maximum torque condition 1
Stdmax2=0.80;          // Maximum torque condition 2

```

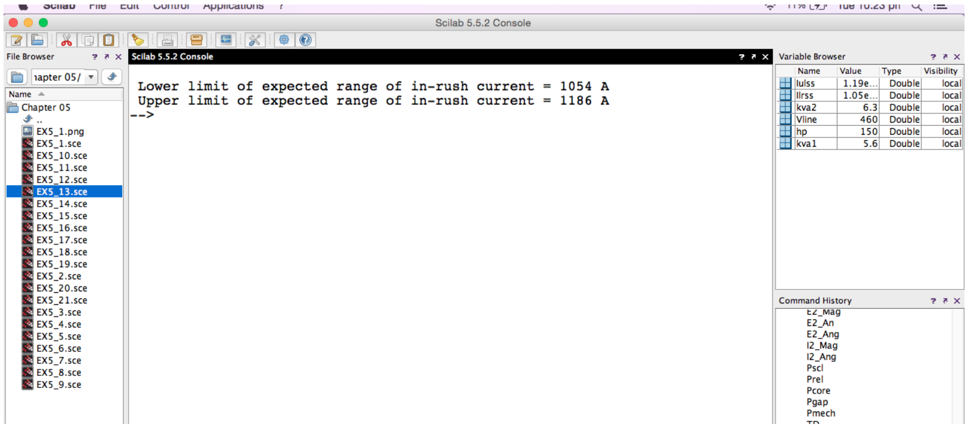

Figure 5.13: Determine the expected in rush current

```
// Percent increase or decrease in rotor circuit
   resistance

PerCh=1/(Stdmax1/Stdmax2);
PerCh=PerCh-1;

// Display result on command window
printf("\n Percent change in rotor circuit
   resistance = %0.0f Percent increase",PerCh*100);
```

Scilab code Exa 5.13 Determine the expected in rush current

```
// Example 5.13
// Determine the expected in-rush current
// Page No. 208

clc;
clear;
close;
```

```
// Given data
kva1=5.6;                // KVA/hp lower limit from
   table 5.9
hp=150;                  // Motor horsepower
Vline=460;               // Line voltage
kva2=6.3;                // KVA/hp upper limit from
   table 5.9

// Expected in-rush current

// Lower limit of expected range of in-rush current
   is
Ilrss=(kva1*hp*1000)/(sqrt(3)*Vline);

// Upper limit of expected range of in-rush current
   is
Iulss=(kva2*hp*1000)/(sqrt(3)*Vline);

// Display result on command window
printf("\n Lower limit of expected range of in-rush
   current = %0.0f A",Ilrss);
printf("\n Upper limit of expected range of in-rush
   current = %0.0f A",Iulss);
```

Scilab code Exa 5.14 Determine percent voltage unbalance and Expected approximate temperature rise if operating at rated load in a 40 deg ambient and Expected insulation life and Required derating of motor to prevent shortening insulation life

```
// Example 5.14
// Determine (a) Percent voltage unbalance (b)
   Expected approximate temp. rise
```

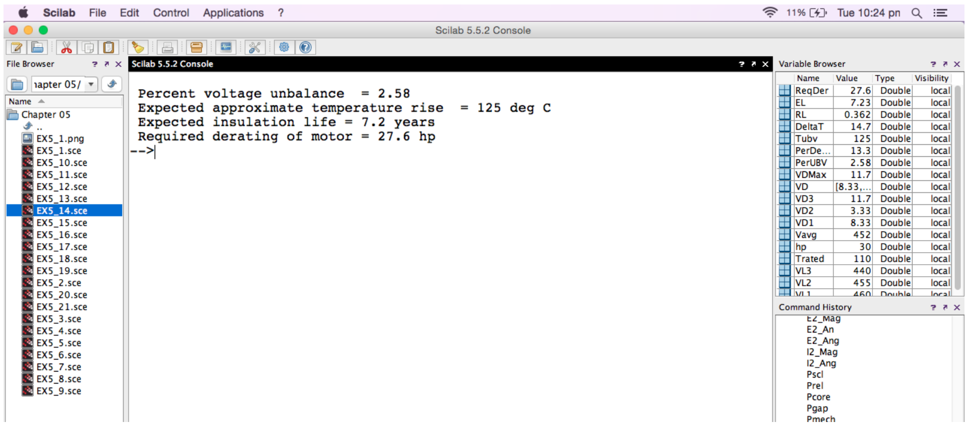

Figure 5.14: Determine percent voltage unbalance and Expected approximate temperature rise if operating at rated load in a 40 deg ambient and Expected insulation life and Required derating of motor to prevent shortening insulation life

```
// if operating at rated load in a 40 deg ambient (c) Expected insulation life
// (d) Required derating of motor to prevent shortening isulation life.
// Page No. 211

clc;
clear;
close;

// Given data
VL1=460;                    // Line voltage 1
VL2=455;                    // Line voltage 2
VL3=440;                    // Line voltage 3
Trated=110;                 // Rated temp. (from table 5.8)
hp=30;                      // Motor horsepower

// (a) Percent voltage unbalance
Vavg=(VL1+VL2+VL3)/3;           // Average line voltage
```

```

VD1=abs(VL1-Vavg);             // Voltage deviation
   from the average
VD2=abs(VL2-Vavg);
VD3=abs(VL3-Vavg);
VD=[VD1 VD2 VD3];
VDMax=max(VD);                 // Choose maximum value
    of voltage deviation
PerUBV=(VDMax/Vavg)*100;

// (b) Expected approximate temp. rise if operating
   at rated load in a 40 deg
PerDeltaT=2*PerUBV^2;          // Percent change in
   temp.
Tubv=Trated*(1+(PerDeltaT/100));

// (c) Expected insulation life
DeltaT=Tubv-Trated;           // Percent increase in
   motor temp.
RL=1/(2^(DeltaT/10));         // Relative life on
   insulation
EL=RL*20;

// (d) Required derating of motor to prevent
   shortening isulation life
ReqDer=hp*0.92;

// Display result on command window
printf("\n Percent voltage unbalance  = %0.2f ",
   PerUBV);
printf("\n Expected approximate temperature rise  =
   %0.0f deg C",Tubv);
printf("\n Expected insulation life = %0.1f years",
   EL);
printf("\n Required derating of motor = %0.1f hp",
   ReqDer);
```

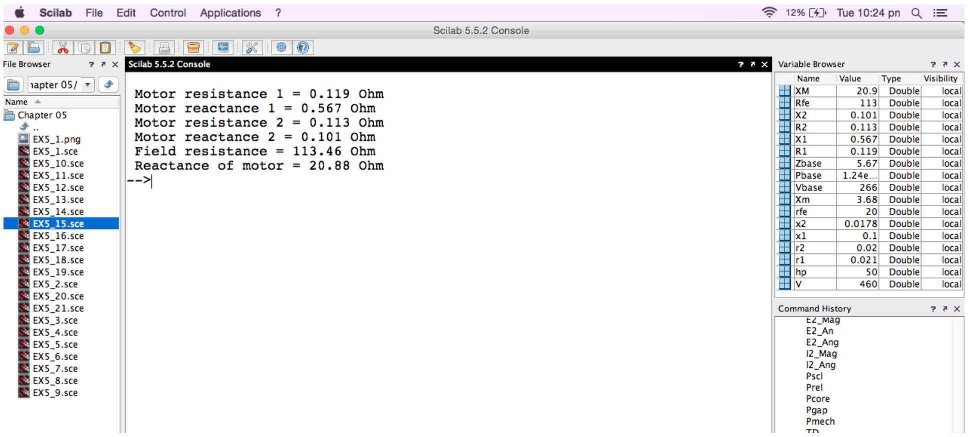

Figure 5.15: Determine the machine parameters in ohms

Scilab code Exa 5.15 Determine the machine parameters in ohms

```
// Example 5.15
// Determine the machine parameters in ohms
// Page No. 213

clc;
clear;
close;

// Given data
V=460;               // Motor voltage
hp=50;               // Motor horsepower
r1=0.021;            // Resistance
r2=0.020;
x1=0.100;            // Reactance
x2=0.0178;
rfe=20;
```

```
Xm=3.68;              // Motor reactance

// Machine parameters in ohms
Vbase=V/sqrt(3);          // Base voltage
Pbase=hp*746/3;           // Base power
Zbase=Vbase^2/Pbase;      // Base impedance

R1=r1*Zbase;
X1=x1*Zbase;
R2=r2*Zbase;
X2=x2*Zbase;
Rfe=rfe*Zbase;
XM=Xm*Zbase;

// Display result on command window
printf("\n Motor resistance 1 = %0.3f Ohm",R1);
printf("\n Motor reactance 1 = %0.3f Ohm",X1);
printf("\n Motor resistance 2 = %0.3f Ohm",R2);
printf("\n Motor reactance 2 = %0.3f Ohm",X2);
printf("\n Field resistance = %0.2f Ohm",Rfe);
printf("\n Reactance of motor = %0.2f Ohm",XM);
```

Scilab code Exa 5.16 Determine R1 R2 X1 X2 XM and the combined core friction and windage loss and Express the no load current as a percent of rated current

```
// Example 5.16
// Determine (a) R1, R2, X1, X2, XM and the combined
    core, friction and windage
// loss (b) Express the no-load current as a percent
    of rated current
// Page No. 218

```

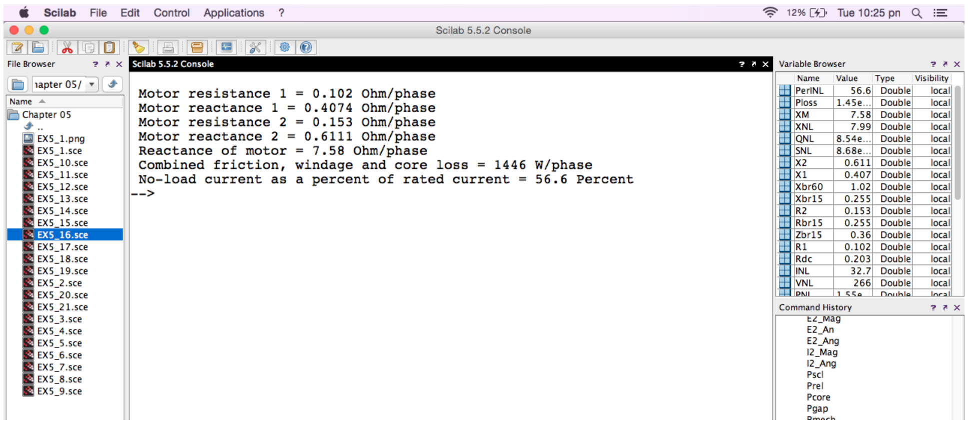

Figure 5.16: Determine R1 R2 X1 X2 XM and the combined core friction and windage loss and Express the no load current as a percent of rated current

```
clc;
clear;
close;

// Given data
P3ph=2573.4;                    // 3-ph power of induction motor
Vline=36.2;                     // Line voltage
Iline=58;                       // Line current
P3phnl=4664.4;                  // No load power
Vlinenl=460;                    // No load line volatge
Ilinenl=32.7;                   // No load line current
Vdc=12;                         // DC voltage
Idc=59;                         // DC current
F1=60;                          // Rated frequency
F2=15;                          // Test frequency
Irated=57.8;                    // Rated current

// (a) R1, R2, X1, X2, XM and the combined core, friction and windage loss
Pbr15=P3ph/3;                   // Power/phase
Vbr15=Vline/sqrt(3);            // Voltage/phase
Ibr15=Iline;
```

```
PNL=P3phnl/3;                    // No load power/phase
VNL=Vlinenl/sqrt(3);             // No load voltage/phase
INL=Ilinenl;                     // No load current/phase

// Determination of R1
Rdc=Vdc/Idc;
R1=Rdc/2;

// Determination of R2
Zbr15=Vbr15/Ibr15;              // Impedance
Rbr15=Pbr15/Ibr15^2;
R2=Rbr15-R1;

// Determination of X1 and X2
Xbr15=sqrt(Zbr15^2-Rbr15^2);
Xbr60=Xbr15*(F1/F2);
X1=0.4*Xbr60;                   // From Table 5.10
X2=0.6*Xbr60;

// Determination of XM
SNL=VNL*INL;
QNL=sqrt(SNL^2-PNL^2);
XNL=QNL/INL^2;
XM=XNL-X1;

// Determination of combined friction , windage and core loss
Ploss=PNL-(INL^2*R1);

// (b) No-load current as a percent of rated current
PerINL=INL*100/Irated;

// Display result on command window
printf("\n Motor resistance 1 = %0.3f Ohm/phase",R1);
printf("\n Motor reactance 1 = %0.4f Ohm/phase",X1);
```

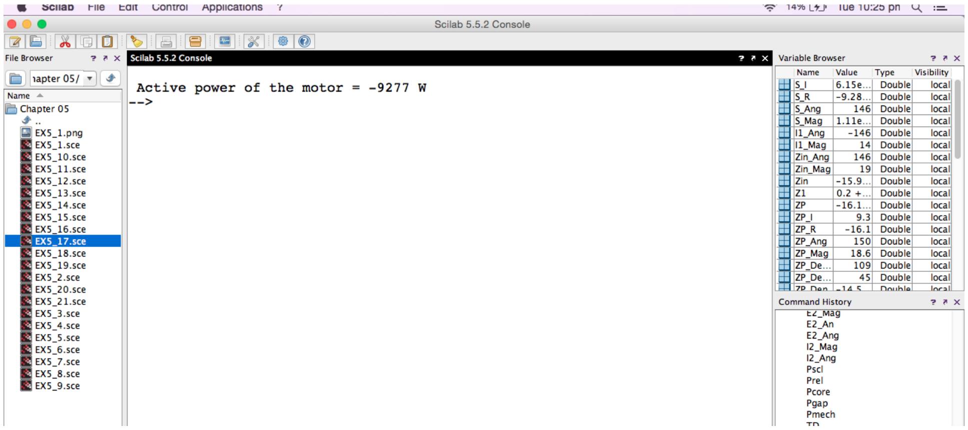

Figure 5.17: Determine the active power that the motor driven as an induction generator delivers to the system

```
printf("\n Motor resistance 2 = %0.3f Ohm/phase",R2);
printf("\n Motor reactance 2 = %0.4f Ohm/phase",X2);
printf("\n Reactance of motor = %0.2f Ohm/phase",XM);
printf("\n Combined friction , windage and core loss = %0.0f W/phase",Ploss);
printf("\n No-load current as a percent of rated current = %0.1f Percent",PerINL);
```

Scilab code Exa 5.17 Determine the active power that the motor driven as an induction generator delivers to the system

```
// Example 5.17
// Determine the active power that the motor, driven as an induction generator
// delivers to the system.
// Page No. 223
```

```
clc;
clear;
close;

// Given data
ns=1200;                    // Speed
nr=1215;
R1=0.200;                   // Motor resistance
R2=0.250;
X1=1.20;                    // Motor reactance
X2=1.29;
Rfe=317;                    // Field resistance
XM=42;                      // Motor reactance
V=460;                      // Voltage rating

// Active power of the motor computation
s=(ns-nr)/ns;               // Speed difference
Z2=(R2/s)+%i*X2;

// Complex to Polar form...
Z2_Mag=sqrt(real(Z2)^2+imag(Z2)^2);        // Magnitude part
Z2_Ang = atan(imag(Z2),real(Z2))*180/%pi; // Angle part

Z0_Num_Mag=Rfe*XM;    // Z0 numerator
Z0_Num_Ang=0+90;

Z0_Den_R=Rfe;        // Z0 denominator
Z0_Den_I=XM;
Z0_Den=Z0_Den_R+%i*Z0_Den_I;
// Complex to Polar form...
Z0_Den_Mag=sqrt(real(Z0_Den)^2+imag(Z0_Den)^2);          // Magnitude part
Z0_Den_Ang = atan(imag(Z0_Den),real(Z0_Den))*180/%pi; // Angle part

```

```
Z0_Mag=Z0_Num_Mag/Z0_Den_Mag;               // Magnitude of Z0
Z0_Ang=Z0_Num_Ang-Z0_Den_Ang;               // Angle of Z0

// Polar to Complex form
Z0_R=Z0_Mag*cos(-Z0_Ang*%pi/180);           // Real part of complex number
Z0_I=Z0_Mag*sin(Z0_Ang*%pi/180);            // Imaginary part of complex number

// ZP computation
ZP_Num_Mag=Z2_Mag*Z0_Mag;                   // ZP numerator magnitude
ZP_Num_Ang=Z2_Ang+Z0_Ang;                   // ZP numerator angle

ZP_Den_R=real(Z2)+Z0_R;                     // Real part of ZP denominator
ZP_Den_I=imag(Z2)+Z0_I;
ZP_Den=ZP_Den_R+%i*ZP_Den_I;                // ZP in complex form

// Complex to Polar form...
ZP_Den_Mag=sqrt(real(ZP_Den)^2+imag(ZP_Den)^2);       // Magnitude part
ZP_Den_Ang = atan(imag(ZP_Den),real(ZP_Den))*180/%pi; // Angle part

ZP_Mag=ZP_Num_Mag/ZP_Den_Mag;               // Final vlaue of ZP in polar form
ZP_Ang=ZP_Num_Ang-ZP_Den_Ang;
// Polar to Complex form
ZP_R=ZP_Mag*cos(-ZP_Ang*%pi/180);           // Real part of complex number
ZP_I=ZP_Mag*sin(ZP_Ang*%pi/180);            // Imaginary part of complex number

```

```
// Zin computation
ZP=ZP_R+%i*ZP_I;                                    // Parallel
    impedance
Z1=R1+%i*X1;
Zin=Z1+ZP;                                          // Input
    impedance
// Complex to Polar form...
Zin_Mag=sqrt(real(Zin)^2+imag(Zin)^2);          //
    Magnitude part
Zin_Ang = atan(imag(Zin),real(Zin))*180/%pi; //
    Angle part

// I1 computation
I1_Mag=(V/sqrt(3))/Zin_Mag;                      // I1
    magnitude
I1_Ang=0-Zin_Ang;                                // I1 angle

// S computation
S_Mag=3*(V/sqrt(3))*I1_Mag;                     // S magnitude
S_Ang=0-(-Zin_Ang);                             // S angle

// Polar to Complex form
S_R=S_Mag*cos(-S_Ang*%pi/180);                  // Real part of
     complex number
S_I=S_Mag*sin(S_Ang*%pi/180);                   // Imaginary
    part of complex number

// Display result on command window
printf("\n Active power of the motor = %0.0f W",S_R)
    ;
```

Scilab code Exa 5.18 Computation of Locked rotor torque and the expected average in rush current and Repeat assuming motor is started at

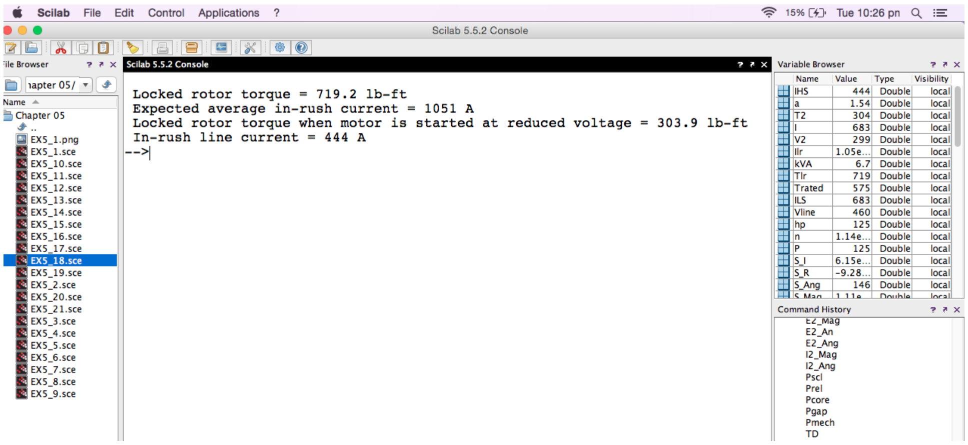

Figure 5.18: Computation of Locked rotor torque and the expected average in rush current and Repeat assuming motor is started at reduced voltage with 65 percent tap and In rush line current line current

reduced voltage with 65 percent tap and In rush line current line current

```
// Example 5.18
// Computation of (a) Locked rotor torque and the expected average in rush
// current (b) Repeat part (a) assuming motor is started at reduced voltage
// with 65% tap (c) In rush line current line current when starting at reduced
// voltage
// Page No. 231

clc;
clear all;
close;

// Given data
P=125;                        // Rated Voltage
n=1141;                       // Speed of machine
hp=125;                       // Horsepower rating of device
```

```
Vline=460;                  // Line voltage
ns=1200;                    // Stator speed
s=0.125;                    // Slip
ILS=683;                    // Current at low side

// (a) Locked rotor torque and the expected average
   in rush current
Trated=P*5252/(n);                    // Rated torque
Tlr=1.25*Trated;                      // Locked rotor
   torque
kVA=(6.3+7.1)/2;
Ilr=(kVA*1000*hp)/(Vline*sqrt(3)); // In-rush
   current

// (b) Locked rotor torque and the expected average
   in rush current when motor
// is started at reduced voltage
V2=0.65*Vline;                       // Voltage
   impressed across the stator
I=Ilr*0.65;                          // Average in-rush
    current
T2=Tlr*(V2/Vline)^2;                 // Locked rotor
   toreque
nr=ns*(1-s);

// (c) In rush line current line current when
   starting at reduced voltage
a=1/0.65;                            // Bank ratio of
   autotransformer
IHS=ILS/a;

// Display result on command window
printf("\n Locked rotor torque = %0.1f lb-ft ",Tlr);
printf("\n Expected average in-rush current = %0.0f
   A ",Ilr);
printf("\n Locked rotor torque when motor is started
    at reduced voltage = %0.1f lb-ft ",T2);
printf("\n In-rush line current = %0.0f A",IHS);
```

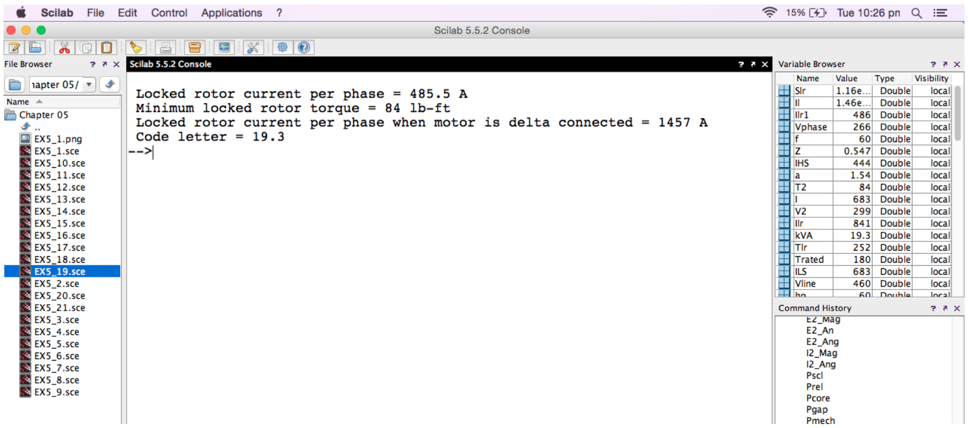

Figure 5.19: Computation of Locked rotor current per phase and minimum locked rotor torque when starting and Locked rotor current per phase when motor is delta connected and Code letter

---

Scilab code Exa 5.19 Computation of Locked rotor current per phase and minimum locked rotor torque when starting and Locked rotor current per phase when motor is delta connected and Code letter

```
// Example 5.19
// Computation of (a) Locked rotor current per phase
    and minimum locked rotor
// torque when starting (b) Locked rotor current per
    phase when motor is delta
// connected (c) Code letter
// Page No.233

clc;
clear all;
close;

```

```
// Given data
V=460;                  // Rated Voltage
Z=0.547;                // Locked rotor impedance
n=1750;                 // Speed of machine
hp=60;                  // Horsepower rating of
   device
f=60;                   // Frequency of motor


// (a) Locked rotor current per phase and minimum
   locked rotor torque
Vphase=V/sqrt(3);         // Voltage/phase
Ilr1=Vphase/Z;             // Locked rotor current/
   phase
Trated=hp*5252/(n);       // Rated torque
Tlr=1.4*Trated;           // Locked rotor torque
T2=Tlr*(Vphase/V)^2;

// (b) Locked rotor current per phase when motor is
   delta connected
Ilr=V/Z;                  // Locked rotor current/
   phase
Il=Ilr*sqrt(3);           // Line current

// (c) Code letter
Slr=sqrt(3)*V*Il/1000;    // Code letter at rated
   voltage
kVA=Slr/f;

// Display result on command window

printf("\n Locked rotor current per phase = %0.1f A"
   ,Ilr1);
printf("\n Minimum locked rotor torque = %0.0f lb-ft
   ",T2);
printf("\n Locked rotor current per phase when motor
    is delta connected = %0.0f A ",Il);
printf("\n Code letter = %0.1f",kVA);
```

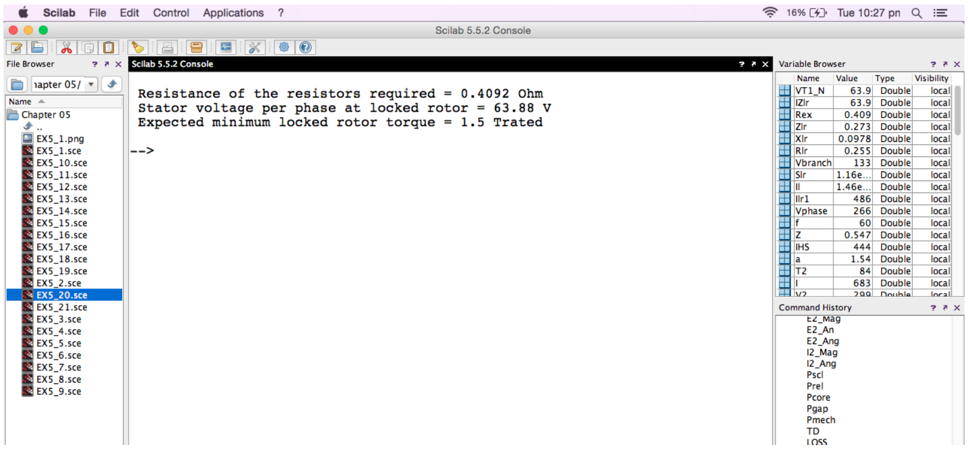

Figure 5.20: Computation of Resistance of the resistors required to limit the locked rotor current to 3 times rated current and Stator voltage per phase at locked rotor and Expected minimum locked rotor torque when starting as a percent of rated torque

---

Scilab code Exa 5.20 Computation of Resistance of the resistors required to limit the locked rotor current to 3 times rated current and Stator voltage per phase at locked rotor and Expected minimum locked rotor torque when starting as a percent of rated torque

```
// Example 5.20
// Computation of (a) Resistance of the resistors required to limit the locked
// rotor current to 3 times rated current (b) Stator voltage per phase at
// locked rotor (c) Expected minimum locked rotor torque when starting as a
// percent of rated torque
// Page No. 235
```

```
clc;
clear all;
close;

// Given data
Ilr=3*78;                    // Locked rotor current
Vbranch=132.79;              // Branch voltage
Rlr=0.2549;                  //Locked rotor resistance
Xlr=0.0978;                  //Locked rotor impedance
f=60;                        //Frequency of motor
Zlr=0.273;

// (a) Resistance of the resistors required to limit
    the locked rotor current
// to 3 times rated current
Rex=sqrt((Vbranch^2/Ilr^2)-(Rlr^2))-Xlr;

// (b) Stator voltage per phase at locked rotor
IZlr=Ilr*Zlr;
VT1_N=IZlr;

// (c) Expected minimum locked rotor torque when
    starting as a percent of
// rated torque
// From table 5.1 --> Minimum locked rotor torque =
    150% rated torque

// Display result on command window

printf("\n Resistance of the resistors required = %0
    .4f Ohm ",Rex);
printf("\n Stator voltage per phase at locked rotor
    = %0.2f V ",VT1_N);
disp('Expected minimum locked rotor torque = 1.5
    Trated');
```

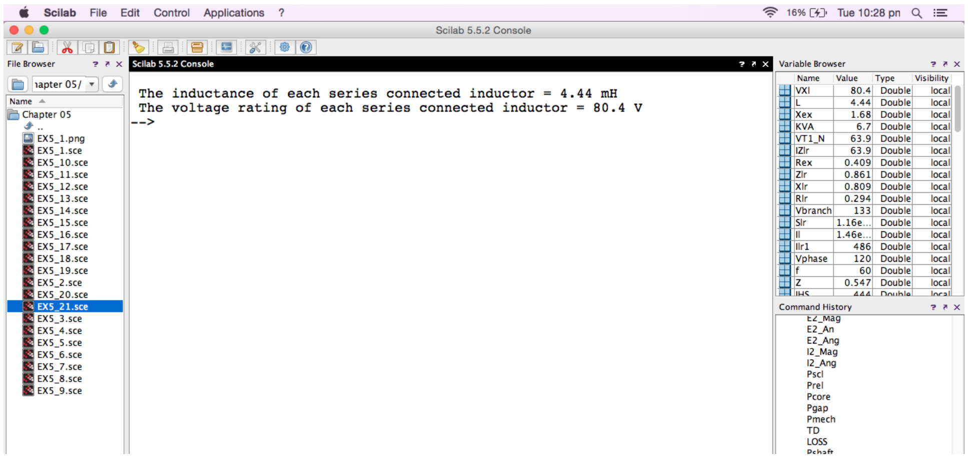

Figure 5.21: Computation of Inductance and voltage rating of each series connected inductor required to limit the starting current to approximately 2 times Irated

Scilab code Exa 5.21 Computation of Inductance and voltage rating of each series connected inductor required to limit the starting current to approximately 2 times Irated

```
// Example 5.21
// Computation of Inductance and voltage rating of
    each series connected
// inductor required to limit the starting current
    to approximately 2*Irated.
// Page No. 236

clc;
clear all;
close;

// Given data
```

```
KVA=6.7;                          // Average locked rotor
   KVA/hp
hp=7.5;                           // Motor horsepower
Vline=208;                        // Line voltage
I=48;                             // Total current
Rlr=0.294;                        // Locked rotor
   resistance
Xlr=0.809;                        // Locked rotor
   impedance
f=60;                             // Frequency of motor

// Corresponding approximate load current
Ilr=KVA*1000*hp/(sqrt(3)*Vline);
Vphase=Vline/sqrt(3);                   // Voltage/phase

// Applying ohm's law to one phase
Zlr=Vphase/Ilr;                     // Impedance
Xex=sqrt((Vphase^2/I^2)-(Rlr^2))-Xlr;
L=Xex/(2*%pi*f);
L=L*10^03;
VXl=I*Xex;

// Display result on command window
printf("\n The inductance of each series connected
   inductor = %0.2f mH ",L);
printf("\n The voltage rating of each series
   connected inductor = %0.1f V ",VXl);
```

Chapter 6

Single Phase Induction Motors

Scilab code Exa 6.1 Determine Locked rotor current in each winding and Phase displacement angle between the two currents and Locked rotor torque in terms of the machine constant and External resistance required in series with the auxiliary winding and Locked rotor torque

```
// Example 6.1
// Determine (a) Locked rotor current in each
   winding (b) Phase displacement
// angle between the two currents (c) Locked rotor
   torque in terms of the
// machine constant (d) External resistance required
    in series with the auxillary
// winding in order to obtain a 30 degree phase
   displacement between the currents
// in the two windings (e) Locked rotor torque for
   the conditions in (d)
// (f) Percent increase in locked rotor torque due
   to the addition of external
// resistance
// Page No. 257

```

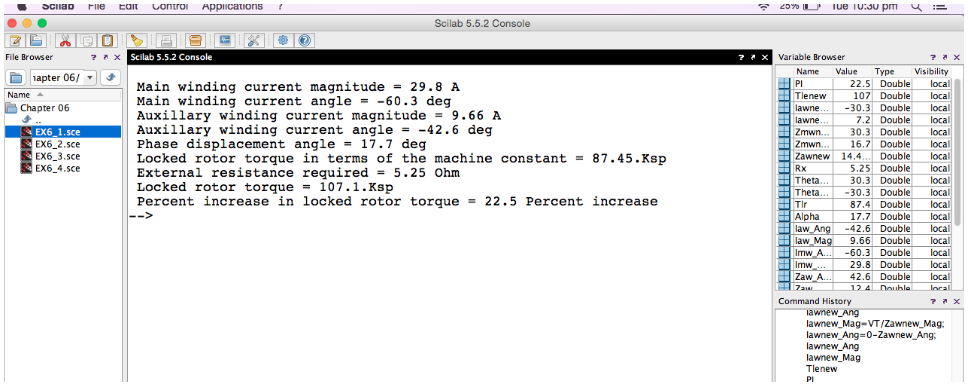

Figure 6.1: Determine Locked rotor current in each winding and Phase displacement angle between the two currents and Locked rotor torque in terms of the machine constant and External resistance required in series with the auxiliary winding and Locked rotor torque

```
clc;
clear;
close;

// Given data
Zmw=2.00+%i*3.50                        // Main winding impedance
Zaw=9.15+%i*8.40                        // Auxillary winding impedance
VT=120;                                 // Transformer voltage
Xaw=8.40;                               // Auxillary winding reactance
Raw=9.15;                               // Auxillary winding resistance

// (a) Locked rotor current in each winding
// Main winding impedance in polar form
// Complex to Polar form...
Zmw_Mag=sqrt(real(Zmw)^2+imag(Zmw)^2);       // Magnitude part
```

```
Zmw_Ang=atan(imag(Zmw),real(Zmw))*180/%pi;  // Angle
    part

// Auxillary winding impedance in polar form
// Complex to Polar form...
Zaw_Mag=sqrt(real(Zaw)^2+imag(Zaw)^2);      //
   Magnitude part
Zaw_Ang=atan(imag(Zaw),real(Zaw))*180/%pi;  // Angle
    part

// Main winding current
Imw_Mag=VT/Zmw_Mag;                    // Main winding
   current magnitude
Imw_Ang=0-Zmw_Ang;                     // Main winding
   current angle

// Auxillary winding current
Iaw_Mag=VT/Zaw_Mag;                    // Auxillary
   winding current magnitude
Iaw_Ang=0-Zaw_Ang;                     // Auxillary
   winding current angle

// (b) Phase displacement angle between the two
   currents
Alpha=abs(Imw_Ang-Iaw_Ang);

// (c) Locked rotor torque in terms of the machine
   constant
Tlr=Imw_Mag*Iaw_Mag*sind(Alpha);

// (d) External resistance required in seris with
   the auxillary winding in
// order to obtain a 30 degree phase displacement
   between the currents in the
// two windings
Theta_awi=Imw_Ang+30;       // Required phase angle
Theta_awz=-Theta_awi;
Rx=(Xaw/tand(Theta_awz))-Raw;
```

```

// (e) Locked rotor torque for the conditions in (d)
Zawnew=Raw+Rx+%i*Xaw;                    // Auxillary winding impedance
// Complex to Polar form...
Zmwnew_Mag=sqrt(real(Zawnew)^2+imag(Zawnew)^2);         // Magnitude part
Zmwnew_Ang=atan(imag(Zawnew),real(Zawnew))*180/%pi;      // Angle part

Iawnew_Mag=VT/Zmwnew_Mag;             // Auxillary winding current magnitude
Iawnew_Ang=0-Zmwnew_Ang;             // Auxillary winding current magnitude
Tlenew=Imw_Mag*Iawnew_Mag*sind(30);

// (f) Percent increase in locked rotor torque due to the addition of external
// resistance
PI=(Tlenew-Tlr)/Tlr*100;


// Display result on command window
printf("\n Main winding current magnitude = %0.1f A ",Imw_Mag);
printf("\n Main winding current angle = %0.1f deg ",Imw_Ang);
printf("\n Auxillary winding current magnitude = %0.2f A ",Iaw_Mag);
printf("\n Auxillary winding current angle = %0.1f deg ",Iaw_Ang);
printf("\n Phase displacement angle = %0.1f deg ",Alpha);
printf("\n Locked rotor torque in terms of the machine constant = %0.2f.Ksp ",Tlr);
printf("\n External resistance required = %0.2f Ohm ",Rx);
printf("\n Locked rotor torque = %0.1f.Ksp ",Tlenew)
```

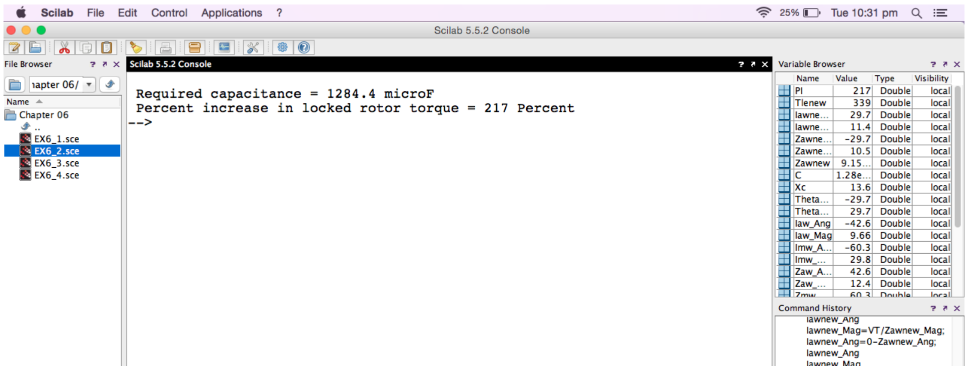

Figure 6.2: Determine Capacitance required in series with the auxiliary winding and Locked rotor torque

```
    ;
printf("\n Percent increase in locked rotor torque =
    %0.1f Percent increase ",PI);
```

Scilab code Exa 6.2 Determine Capacitance required in series with the auxiliary winding and Locked rotor torque

```
// Example 6.2
// Determine (a) Capacitance required in series with
    the auxillary winding
// in order to obtain a 90 degree phase displacement
    between the current in
// the main winding and the current in the auxillary
    winding at locked rotor
// (b) Locked rotor torque in terms of the machine
   constant
// Page No. 265

clc;
clear;
```

```
close;

// Given data
Zmw=2.00+%i*3.50                    // Main winding impedance
Zaw=9.15+%i*8.40                    // Auxillary winding impedance
VT=120;                             // Transformer voltage
Xaw=8.40;                           // Auxillary winding reactance
Raw=9.15;                           // Auxillary winding resistance
f=60;                               // Frequency
Tlr=107.1;                          // Original torque

// (a) Capacitance required in series with the auxillary winding
// Main winding impedance in polar form
// Complex to Polar form...
Zmw_Mag=sqrt(real(Zmw)^2+imag(Zmw)^2);        // Magnitude part
Zmw_Ang=atan(imag(Zmw),real(Zmw))*180/%pi;   // Angle part

// Auxillary winding impedance in polar form
// Complex to Polar form...
Zaw_Mag=sqrt(real(Zaw)^2+imag(Zaw)^2);        // Magnitude part
Zaw_Ang=atan(imag(Zaw),real(Zaw))*180/%pi;   // Angle part

// Main winding current
Imw_Mag=VT/Zmw_Mag;                     // Main winding current magnitude
Imw_Ang=0-Zmw_Ang;                      // Main winding current angle

```

```
// Auxillary winding current
Iaw_Mag=VT/Zaw_Mag;                    // Auxillary winding current magnitude
Iaw_Ang=0-Zaw_Ang;                     // Auxillary winding current angle

Theta_awi=90-60.26;                   // Required phase angle
Theta_awz=-Theta_awi;

Xc=Xaw-Raw*tand(Theta_awz);           // Capacitive reactance

C=1/2*%pi*f*Xc;                        // Required capacitance


// (b) Locked rotor torque in terms of the machine constant
Zawnew=Raw+%i*Xaw-%i*Xc;               // Auxillary winding impedance
// Complex to Polar form...
Zawnew_Mag=sqrt(real(Zawnew)^2+imag(Zawnew)^2);        // Magnitude part
Zawnew_Ang=atan(imag(Zawnew),real(Zawnew))*180/%pi;    // Angle part

Iawnew_Mag=VT/Zawnew_Mag;              // Auxillary winding current magnitude
Iawnew_Ang=0-Zawnew_Ang;               // Auxillary winding current magnitude

Tlenew=Imw_Mag*Iawnew_Mag*sind(90);

// Percent change increase in locked rotor torque
PI=(Tlenew-Tlr)/Tlr*100;


```

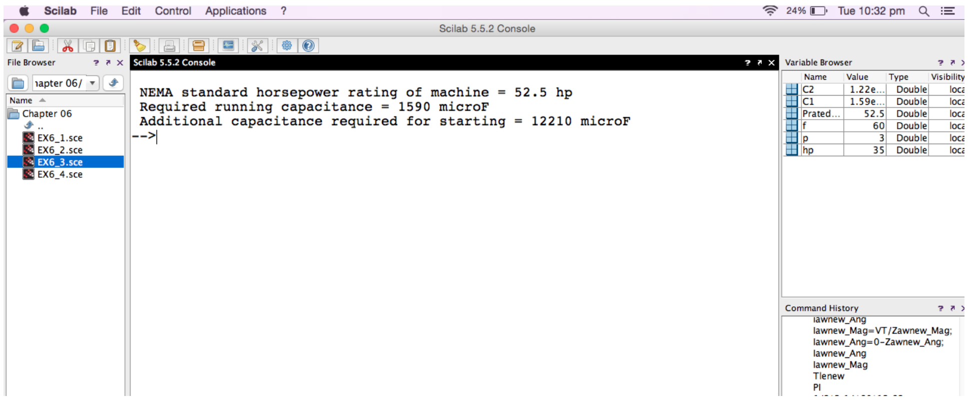

Figure 6.3: Determine NEMA standard horsepower rating of machine and Required running capacitance and Additional capacitance required for starting

```
// Display result on command window
printf("\n Required capacitance = %0.1f microF ",C);
printf("\n Percent increase in locked rotor torque =
    %0.0f Percent",PI);

//Note: Capacitor computation is wrong in the book
```

Scilab code Exa 6.3 Determine NEMA standard horsepower rating of machine and Required running capacitance and Additional capacitance required for starting

```
// Example 6.3
// Determine (a) NEMA standard horsepower rating of
    machine (b) Required
// running capacitance (c) Additional capacitance
    required for starting
// Page No. 271

```

```
clc;
clear;
close;

// Given data
hp=35;                    // Power in hp
p=3;                      // Number of phase
f=60;                     // Frequency


// (a) NEMA standard horsepower rating of machine

Prated3ph=hp*p/2;

// (b)Required running capacitance

C1=26.5*f;

// (c) Additional capacitance required for starting.

C2=230*f-C1;

// Display result on command window
printf("\n NEMA standard horsepower rating of
   machine = %0.1f hp ",Prated3ph);
printf("\n Required running capacitance = %0.0f
   microF ",C1);
printf("\n Additional capacitance required for
   starting = %0.0f microF ",C2);
```

Scilab code Exa 6.4 Computation of Motor line current and motor phase current and Motor line current and motor phase current if one line opens and Line and phase currents if the power factor is 82 percent

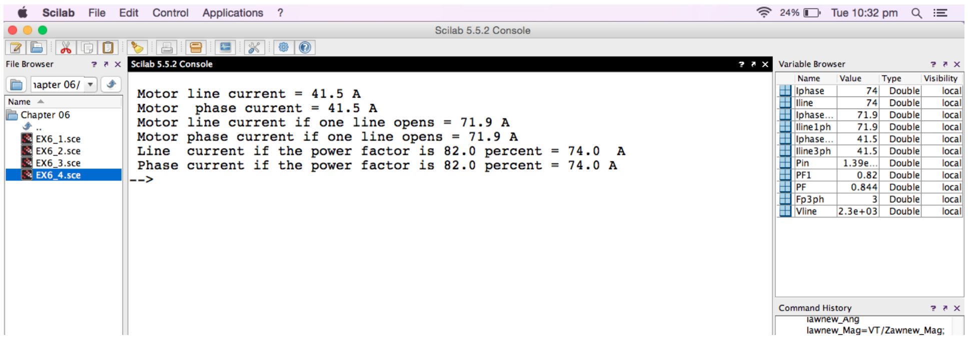

Figure 6.4: Computation of Motor line current and motor phase current and Motor line current and motor phase current if one line opens and Line and phase currents if the power factor is 82 percent

```
// Example 6.4
// Computation of (a) Motor line current and motor phase current (b) Motor line
// current and motor phase current if one line opens (c) Line and phase
// currents if the power factor when single phasing is 82.0 percent.
// Page No. 274

clc;
clear;
close;

// Given data
Vline=2300;                         // Line voltage
Fp3ph=3;                            // Frequency of three phase
PF=0.844;                           // Power factor
PF1=0.820;                          // 82.2 percent power factor
Pin=350*746/(0.936*2);              // Input power


// (a) Motor line current and motor phase current
```

```

Iline3ph=Pin/(sqrt(3)*Vline*PF);
Iphase3ph=Iline3ph;

//(b) Motor line current and motor phase current if
    one line opens

Iline1ph=(sqrt(3)*Iline3ph*PF)/PF;
Iphase1ph=Iline1ph;

// (c) Line and phase currents if the power factoe
    when single phasing is 82.0 percent.

Iline=(Iline1ph*PF)/PF1;
Iphase=Iline;

// Display result on command window
printf("\n Motor line current = %0.1f A ",Iline3ph);
printf("\n Motor  phase current = %0.1f A ",
    Iphase3ph);
printf("\n Motor line current if one line opens = %0
    .1f A ",Iline1ph);
printf("\n Motor phase current if one line opens =
    %0.1f A ",Iphase1ph);
printf("\n Line  current if the power factor is 82.0
      percent = %0.1f  A",Iline);
printf("\n Phase current if the power factor is 82.0
      percent = %0.1f A ",Iphase);
```

Chapter 7

Speciality Machines

Scilab code Exa 7.1 Determine Torque load on the shaft and Torque angle if the voltage drops to 224V

```
// Example 7.1
// Determine (a) Torque load on the shaft (b) Torque
    angle if the voltage
// drops to 224V (c) Will the rotor pull out of
    synchronism?
// Page No. 282

clc;
clear;
close;

// Given data
f=60;                    // Frequency
P=4;                     // Number of poles
Pshaft=10;               // Shaft power in hp
V1=240;                  // Rated voltage
V2=224;                  // New voltage
phirel1=30;              // Torque angle
```

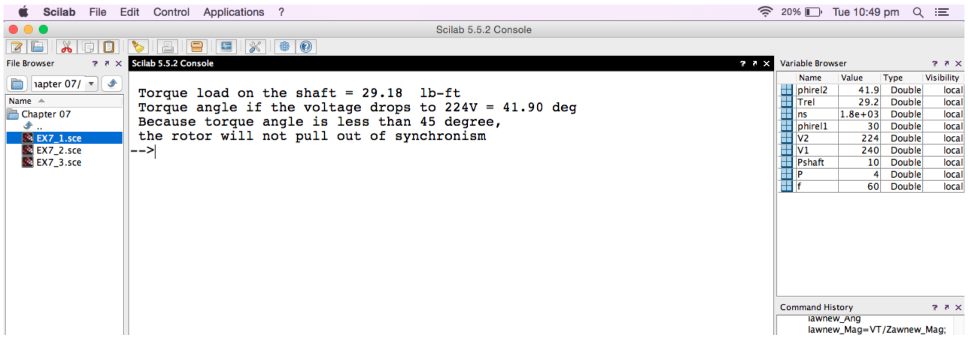

Figure 7.1: Determine Torque load on the shaft and Torque angle if the voltage drops to 224V

```
// (a) Torque load on the shaft
ns=120*f/P;                    // speed of machine
Trel=Pshaft*5252/ns;


// (b) Torque angle if the voltage drops to 224V
phirel2=asind((V1^2/V2^2)*sind(2*phirel1))/2

// Display result on command window
printf("\n Torque load on the shaft = %0.2f  lb-ft "
    ,Trel);
printf("\n Torque angle if the voltage drops to 224V
    = %0.2f deg ",phirel2);
printf("\n Because torque angle is less than 45
    degree, \n the rotor will not pull out of
    synchronism ")
```

Scilab code Exa 7.2 Determine Resolution and Number of steps required for the rotor and Shaft speed

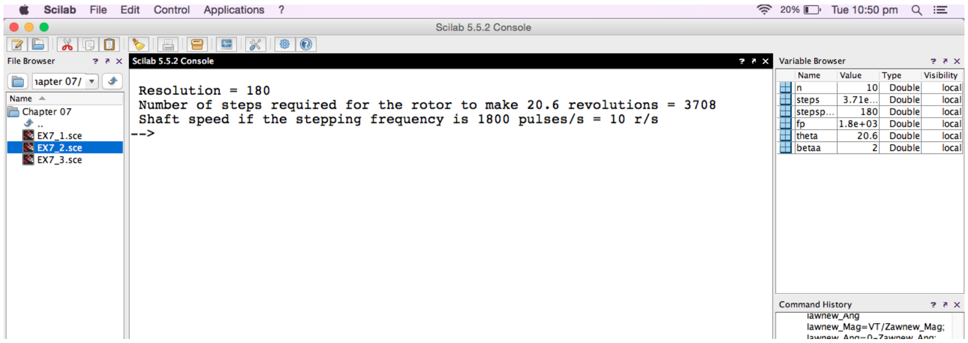

Figure 7.2: Determine Resolution and Number of steps required for the rotor and Shaft speed

```
// Example 7.2
// Determine (a) Resolution (b) Number of steps
   required for the rotor to make
// 20.6 revolutions (c) Shaft speed if the stepping
   frequency is 1800 pulses/s
// Page No. 287

clc;
clear;
close;

// Given data
betaa=2;                 // Step angle
theta=20.6;              // Number of revolutions
fp=1800;                 // Stepping frequency


// (a) Resolution
stepsperrev=360/betaa;       // Speed of machine


// (b) Number of steps required for the rotor to
   make 20.6 revolutions
steps=theta*360/betaa;

```

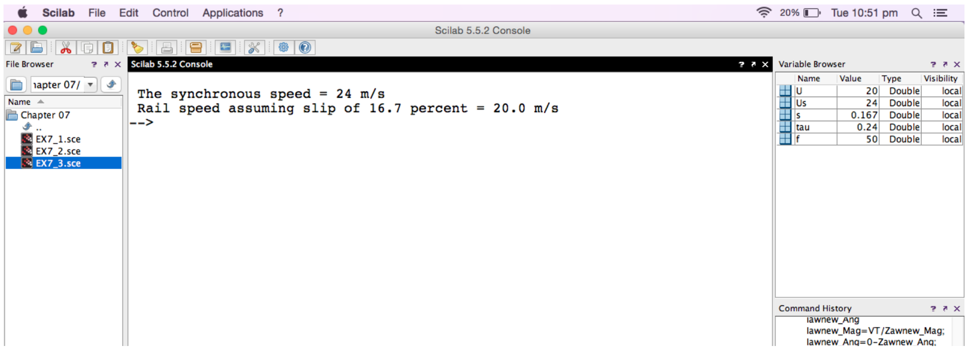

Figure 7.3: Determine Synchronous speed and Rail speed

```

// (c) Shaft speed if the stepping frequency is 1800 pulses/s.
n=betaa*fp/360;


// Display result on command window
printf("\n Resolution = %0.0f ",stepsperrev);
printf("\n Number of steps required for the rotor to make 20.6 revolutions = %0.0f ",steps);
printf("\n Shaft speed if the stepping frequency is 1800 pulses/s = %0.0f r/s ",n);
```

Scilab code Exa 7.3 Determine Synchronous speed and Rail speed

```
// Example 7.3
// Determine (a) Synchronous speed (b) Rail speed assuming slip of 16.7%
// Page No. 299

clc;
```

```
clear;
close;

// Given data
f=50;                     // Frequency of machine
tau=0.24;                 // Pole pitch
s=0.167;                  // Slip

// (a) The synchronous speed
Us=2*tau*f;

// (b) Rail speed assuming slip of 16.7 percent
U=Us*(1-s);


// Display result on command window
printf("\n The synchronous speed = %0.0f m/s ",Us);
printf("\n Rail speed assuming slip of 16.7 percent
   = %0.1f m/s ",U);
```

Chapter 8

Synchronous Motors

Scilab code Exa 8.1 Determine Developed torque and Armature current and Excitation voltage and Power angle and Maximum torque

```
// Example 8.1
// Determine (a) Developed torque (b) Armature
    current (c) Excitation voltage
// (d) Power angle (e) Maximum torque
// Page No. 317

clc;
clear;
close;

// Given data
f=60;                          // Operating frequency
P=4;                           // Number of poles
Pmech=100;                     // Mechanical power
eta=0.96;                      // Efficiency
FP=0.80;                       // Power factor leading
V=460;                         // Motor voltage
Xs_Mag=2.72;                   // Synchronous reactnace
```

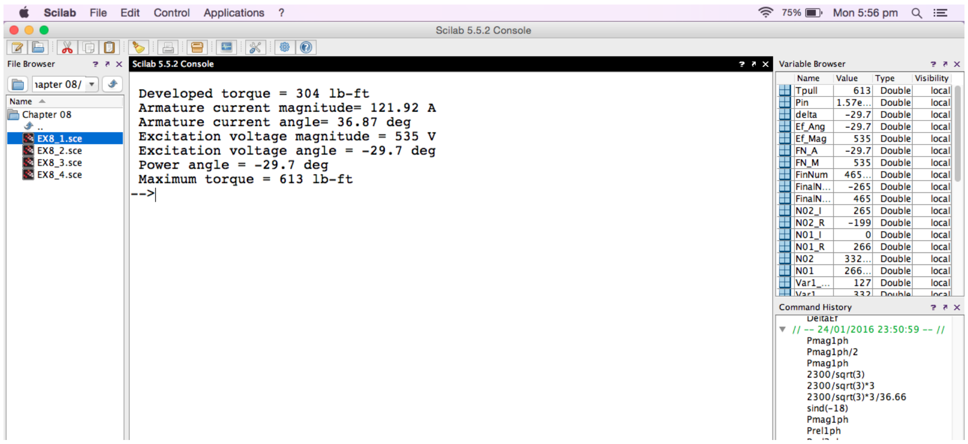

Figure 8.1: Determine Developed torque and Armature current and Excitation voltage and Power angle and Maximum torque

```
    magnitude
Xs_Ang=90;                          // Synchronous reactnace
    magnitude
deltaPull=-90;                       // Pullout power angle
// (a) Developed torque
ns=120*f/P;                         // Synchronous speed
Td=5252*Pmech/(ns*eta);


// (b) Armature current
S=Pmech*746/(eta*FP);
Theta=-acosd(FP);                  // Power factor angle (
   negative as FP is leading)
V1phi=V/sqrt(3);                   // Single line voltage
S1phi_Mag=S/3;                     // Magnitude
S1phi_Ang=Theta;                   // Angle
VT_Mag=V1phi;
VT_Ang=0;
Ia_Mag=S1phi_Mag/VT_Mag;           // Armature current
   magnitude
Ia_Ang=S1phi_Ang-VT_Ang;           // Armature current angle
Ia_Ang=-Ia_Ang;                    // Complex conjugate of
```

```
    Ia

// (c) Excitation voltage
Var1_Mag=Ia_Mag*Xs_Mag;
Var1_Ang=Ia_Ang+Xs_Ang;

/////////
N01=VT_Mag+%i*VT_Ang;
N02=Var1_Mag+%i*Var1_Ang;
// Polar to Complex form

N01_R=VT_Mag*cos(-VT_Ang*%pi/180); // Real part of
    complex number 1
N01_I=VT_Mag*sin(VT_Ang*%pi/180); //Imaginary part
    of complex number 1

N02_R=Var1_Mag*cos(-Var1_Ang*%pi/180); // Real part
    of complex number 2
N02_I=Var1_Mag*sin(Var1_Ang*%pi/180); //Imaginary
    part of complex number 2

FinalNo_R=N01_R-N02_R;
FinalNo_I=N01_I-N02_I;
FinNum=FinalNo_R+%i*FinalNo_I;
// Complex to Polar form...

FN_M=sqrt(real(FinNum)^2+imag(FinNum)^2); //
    Magnitude part
FN_A = atan(imag(FinNum),real(FinNum))*180/%pi;//
    Angle part
//////
Ef_Mag=FN_M;
Ef_Ang=FN_A;
// (d) Power angle
delta=Ef_Ang;

// (e) Maximum torque
Pin=3*(-VT_Mag*Ef_Mag/Xs_Mag)*sind(deltaPull);    //
```

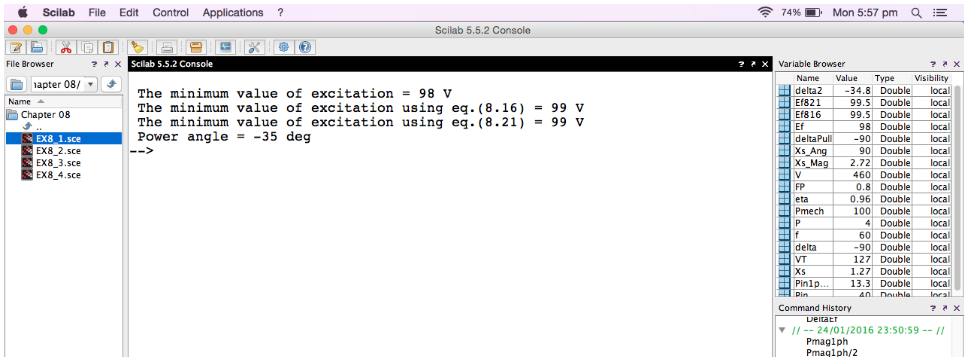

Figure 8.2: Determine minimum value of excitation and Repeat calculations using eq 8 16 and 8 21 and Power angle

```
    Active power input
Tpull=5252*Pin/(746*ns);



// Display result on command window
printf("\n Developed torque = %0.0f lb-ft ",Td);
printf("\n Armature current magnitude= %0.2f A ",
    Ia_Mag);
printf("\n Armature current angle= %0.2f deg ",
    Ia_Ang);
printf("\n Excitation voltage magnitude = %0.0f V ",
    Ef_Mag);
printf("\n Excitation voltage angle = %0.1f deg ",
    Ef_Ang);
printf("\n Power angle = %0.1f deg ",delta);
printf("\n Maximum torque = %0.0f lb-ft ",Tpull);
```

Scilab code Exa 8.2 Determine minimum value of excitation and Repeat calculations using eq 8 16 and 8 21 and Power angle

```
// Example 8.2
// Determine (a) The minimum value of excitation
   that will maintain
// synchronism (b) Repeat (a) using eq.(8.16) (c)
   Repeat (a) using eq.(8.21)
// (d) Power angle if the field excitation voltage
   is increased to 175% of the
// stability limit determined in (c)
// Page No. 322

clc;
clear;
close;

// Given data
Pin=40;                     // Input power
Pin1phase=40/3;             // Single phase power
Xs=1.27;                    // Synchronous reactnace
VT=220/sqrt(3);             // Voltage
delta=-90;                  // Power angle

f=60;                       // Operating frequency
P=4;                        // Number of poles
Pmech=100;                  // Mechanical power
eta=0.96;                   // Efficiency
FP=0.80;                    // Power factor leading
V=460;                      // Motor voltage
Xs_Mag=2.72;                // Synchronous reactnace
    magnitude
Xs_Ang=90;                  // Synchronous reactnace
    magnitude
deltaPull=-90;               // Pullout power angle

// (a) The minimum value of excitation that will
   maintain synchronism
Ef=98;                      // From the graph (
   Figure 8.13)

```

```
// (b) The minimum value of excitation using eq.(8.16)
Ef816=-Pin*Xs*746/(3*VT*sind(delta));


// (c) The minimum value of excitation using eq.(8.21)
Ef821=Xs*Pin1phase*746/(VT);

// (d) Power angle if the field excitation voltage is increased to 175%
delta2=Ef816*sind(delta)/(1.75*Ef816);
delta2=asind(delta2);

// Display result on command window
printf("\n The minimum value of excitation = %0.0f V ",Ef);
printf("\n The minimum value of excitation using eq.(8.16) = %0.0f V ",Ef816);
printf("\n The minimum value of excitation using eq.(8.21) = %0.0f V ",Ef821);
printf("\n Power angle = %0.0f deg ",delta2);
```

Scilab code Exa 8.3 Determine System active power and Power factor of the synchronous motor and System power factor and Percent change in synchronous field current and Power angle of the synchronous motor

```
// Example 8.3
// Determine (a) System active power (b) Power factor of the synchronous motor
// (c) System power factor (d) Percent change in synchronous field current
// required to adjust the system power factor to
```

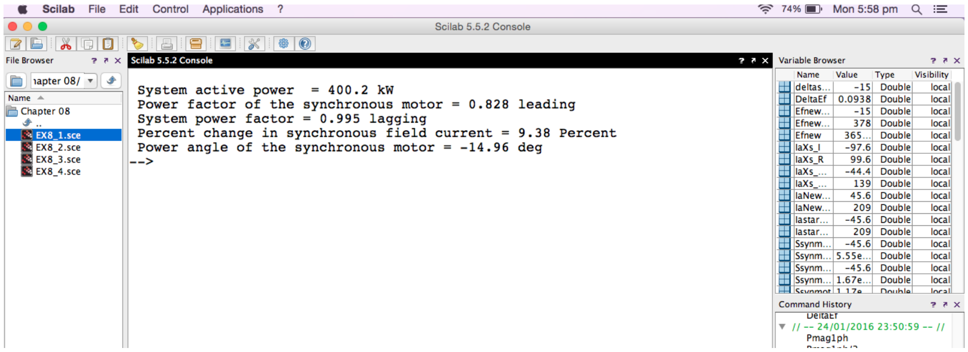

Figure 8.3: Determine System active power and Power factor of the synchronous motor and System power factor and Percent change in synchronous field current and Power angle of the synchronous motor

```
    unity (e) Power angle of the
// synchronous motor for the conditions in (d)
// Page No. 324

clc;
clear;
close;

// Given data

Php=400;                    // Power in hp
eta=0.958;                  // Efficiency
Pheater=50000;              // Resistance heater
    power
Vs=300;                     // Synchronous motor
    voltage
eta2=0.96;                  // Synchronous motor
    efficiency
Xs=0.667;                   // Synchronous reactnace
VT=460;                     // 3-Phase supply
    voltage
delta=-16.4;                // Power angle

```

```
// (a) System active power
Pindmot=Php*0.75*746/(eta);    // Motor operating at three quarter rated load
Psynmot=Vs*0.5*746/(eta2);     // Synchronous motor power
Psys=Pindmot+Pheater+Psynmot;
Psysk=Psys/1000;

// (b) Power factor of the synchronous motor
Pin=Psynmot;                     // Power input
Vtph=VT/sqrt(3);                 // Voltage per phase
Ef=-(Pin*Xs)/(3*Vtph*sind(delta));
// Complex to Polar form...

Ef_Mag=Ef;            // Magnitude part
Ef_Ang=delta;         // Angle part
Vtph_Mag=Vtph;
Vtph_Ang=0;
////////////
N01=Ef_Mag+%i*Ef_Ang;       // Ef in polar form
N02=Vtph_Mag+%i*Vtph_Ang;   // Vt in polar for

N01_R=Ef_Mag*cos(-Ef_Ang*%pi/180); // Real part of complex number Ef
N01_I=Ef_Mag*sin(Ef_Ang*%pi/180); //Imaginary part of complex number Ef

N02_R=Vtph_Mag*cos(-Vtph_Ang*%pi/180); // Real part of complex number Vt
N02_I=Vtph_Mag*sin(Vtph_Ang*%pi/180); //Imaginary part of complex number Vt

FinalNo_R=N01_R-N02_R;
FinalNo_I=N01_I-N02_I;
FinNum=FinalNo_R+%i*FinalNo_I;
// Complex to Polar form...

FN_M=sqrt(real(FinNum)^2+imag(FinNum)^2); //
```

```
    Magnitude part
FN_A = atan(imag(FinNum),real(FinNum))*180/%pi;//
    Angle part

Ia_Mag=FN_M/Xs;          // Magnitude of Ia
Ia_Ang=FN_A-(-90);       // Angle of Ia
Theta=0-Ia_Ang;
FP=cosd(Theta);          // Power factor


// (c) System power factor
ThetaIndMot=acosd(0.891);    // Induction motor
    power factor
Thetaheat=acosd(1);          // Heater power factor
ThetaSyncMot=-34.06;         // Synchronous motor
    power factor
Qindmot=tand(27)*Pindmot;
Qsynmot=tand(ThetaSyncMot)*Psynmot;
Qsys=Qindmot+Qsynmot;
Ssys=Psys+%i*Qsys;          // System variable in
    complex form

// Complex to Polar form...

Ssys_Mag=sqrt(real(Ssys)^2+imag(Ssys)^2);          //
     Magnitude part
Ssys_Ang = atan(imag(Ssys),real(Ssys))*180/%pi;    //
     Angle part

FPsys=cosd(Ssys_Ang);                              //
     System power factor

// (d) Percent change in synchronous field current
    required to adjust the
// system power factor to unity

Ssynmot=Psynmot-(%i*(-Qsynmot+Qsys));    //
    Synchronous motor system
```

```
// Complex to Polar form...

Ssynmot_Mag=sqrt(real(Ssynmot)^2+imag(Ssynmot)^2);      // Magnitude part
Ssynmot_Ang=atan(imag(Ssynmot),real(Ssynmot))*180/%pi; // Angle part

Ssynmot1ph_Mag=Ssynmot_Mag/3;              // For single phase magnitude
Ssynmot1ph_Ang=Ssynmot_Ang;                // For single phase angle

Iastar_Mag=Ssynmot1ph_Mag/Vtph;            // Current magnitude
Iastar_Ang=Ssynmot1ph_Ang-0;               // Current angle

IaNew_Mag=Iastar_Mag;
IaNew_Ang=-Iastar_Ang;

IaXs_Mag=IaNew_Mag*Xs;
IaXs_Ang=IaNew_Ang-90;

// Convert these number into complex and then perform addition
// Polar to Complex form

// Y=29.416<-62.3043 //Polar form number
IaXs_R=IaXs_Mag*cos(-IaXs_Ang*%pi/180);  // Real part of complex number
IaXs_I=IaXs_Mag*sin(IaXs_Ang*%pi/180);   // Imaginary part of complex number
Efnew=Vtph+IaXs_R+%i*IaXs_I;
// Complex to Polar form...

Efnew_Mag=sqrt(real(Efnew)^2+imag(Efnew)^2);      // Magnitude part
```

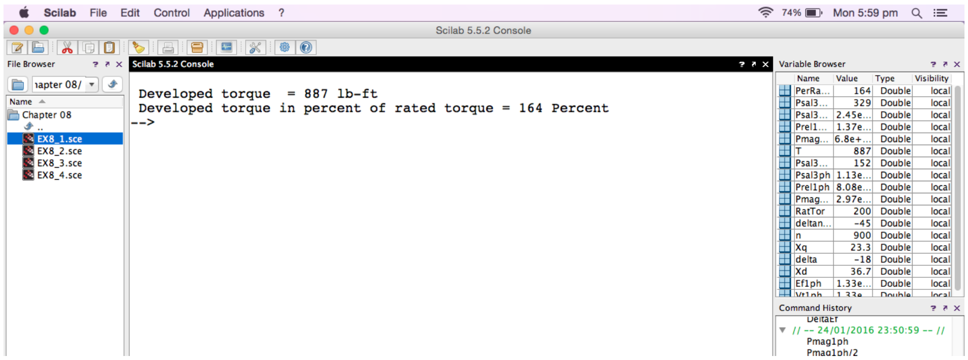

Figure 8.4: Determine Developed torque and Developed torque in percent of rated torque

```
Efnew_Ang=atan(imag(Efnew),real(Efnew))*180/%pi; //
    Angle part

DeltaEf=(Efnew_Mag-Ef)/Ef;

// (e) Power angle of the synchronous motor
deltasynmot=Efnew_Ang;

// Display result on command window
printf("\n System active power  = %0.1f kW ",Psysk);
printf("\n Power factor of the synchronous motor =
    %0.3f leading ",FP);
printf("\n System power factor = %0.3f lagging ",
    FPsys);
printf("\n Percent change in synchronous field
    current = %0.2f Percent ",DeltaEf*100);
printf("\n Power angle of the synchronous motor = %0
    .2f deg ",deltasynmot);
```

Scilab code Exa 8.4 Determine Developed torque and Developed torque in percent of rated torque

```
// Example 8.4
// Determine (a) Developed torque if the field current is adjusted so that the
// excitation voltage is equal to two times the applied stator voltage , and the
// power angle is −18 degrees (b) Developed torque in percent of rated torque ,
// if the load is increased until maximum reluctance torque occurs .
// Page No. 328

clc;
clear;
close;

// Given data
Vt1ph=2300/sqrt(3);            // Applied voltage/phase
Ef1ph=2300/sqrt(3);            // Excitation voltage/phase
Xd=36.66;                      // Direct axis reactance/phase
delta=-18;                     // Power angle
Xq=23.33;                      // Quadrature−axis reactance/phase
n=900;                         // Speed of motor
deltanew=-45;
RatTor=200;                    // Rated torque of motor
// (a)  Developed torque
Pmag1ph=-((Vt1ph*2*Ef1ph)/Xd)*sind(delta);   // Power
Prel1ph=-Vt1ph^2*( (Xd-Xq) / (2*Xd*Xq)) *sind(2*delta); // Reluctance power
Psal3ph=3*(Pmag1ph+Prel1ph);   // Salient power of motor
Psal3phHP=Psal3ph/746;
T=(5252*Psal3phHP)/n;          // Developed torque
```

```
// (b) Developed torque in percent of rated torque
// The reluctance torque has its maximum value at delta= -45 degrees
Pmag1phnew=-((Vt1ph*2*Ef1ph)/Xd)*sind(deltanew); // Power
Prel1phnew=-Vt1ph^2*( (Xd-Xq) / (2*Xd*Xq)) *sind(2*deltanew); // Reluctance power
Psal3phnew=3*(Pmag1phnew+Prel1phnew);  // Salient power of motor
Psal3phHPnew=Psal3phnew/746;
PerRatTorq=Psal3phHPnew*100/RatTor;

// Display result on command window
printf("\n Developed torque  = %0.0f lb-ft ",T);
printf("\n Developed torque in percent of rated torque = %0.0f Percent ",PerRatTorq);
```

Chapter 9

Synchronous Generators

Scilab code Exa 9.1 Determine Turbine torque supplied to the alternator and Excitation voltage and Active and reactive components of apparent power and Power factor and Excitation voltage

```
// Example 9.1
// Determine (a) Turbine torque supplied to the
    alternator (b) Excitation
// voltage (c) Active and reactive components of
    apparent power (d) Power
// factor (e) Neglecting saturation effects ,
    excitation voltage if the field
// current is reduced to 85% of its voltage in (a) (
    f) Turbine speed.
// Page No. 342

clc;
clear;
close;

// Given data
hp=112000;                          // Power input
```

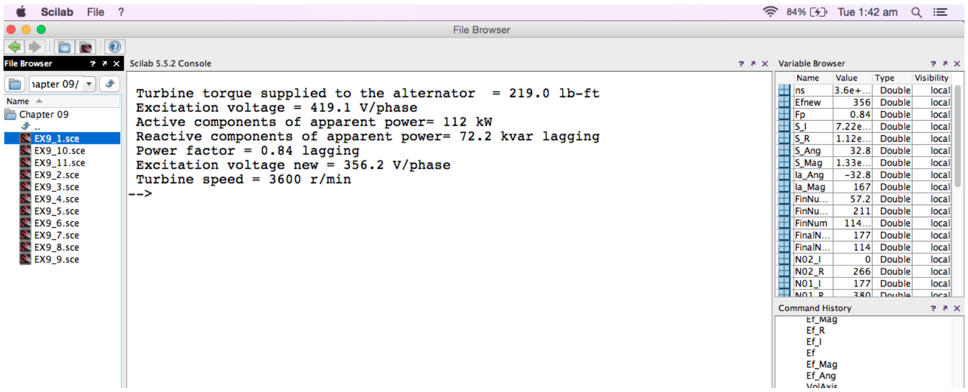

Figure 9.1: Determine Turbine torque supplied to the alternator and Excitation voltage and Active and reactive components of apparent power and Power factor and Excitation voltage

```
n=746*3600;                      // Speed
VT=460;                          // 3-Phase supply voltage
Pout=112000;                     // Power
Xs=1.26;                         // Synchronous reactnace
delta=25;                        // Power angle
eta=0.85;                        // Percent reduction factor
P=2;                             // Number of poles
f=60;                            // Frequnecy

// (a) Turbine torque supplied to the alternator
T=(hp*5252)/n;

// (b) Excitation voltage
Vt=VT/sqrt(3);                       // Voltage/phase
Ef=(Pout*Xs)/(3*Vt*sind(delta));

// (c) Active and reactive components of apparent power
// Vt=Ef-Ia*j*Xs
// Solving for Vt-Ef
```

```
Vt_Mag=Vt;
Vt_Ang=0;
Ef_Mag=Ef;
Ef_Ang=delta;
//
N01=Ef_Mag+%i*Ef_Ang;       // Ef in polar form
N02=Vt_Mag+%i*Vt_Ang;       // Vt in polar for

N01_R=Ef_Mag*cos(-Ef_Ang*%pi/180); // Real part of complex number Ef
N01_I=Ef_Mag*sin(Ef_Ang*%pi/180); //Imaginary part of complex number Ef

N02_R=Vt_Mag*cos(-Vt_Ang*%pi/180); // Real part of complex number Vt
N02_I=Vt_Mag*sin(Vt_Ang*%pi/180); //Imaginary part of complex number Vt

FinalNo_R=N01_R-N02_R;
FinalNo_I=N01_I-N02_I;
FinNum=FinalNo_R+%i*FinalNo_I;

// Now FinNum/Xs in polar form
FinNum_Mag=sqrt(real(FinNum)^2+imag(FinNum)^2);           // Magnitude part
FinNum_Ang = atan(imag(FinNum),real(FinNum))*180/%pi;   // Angle part
Ia_Mag=FinNum_Mag/Xs;
Ia_Ang=FinNum_Ang-90;

// Computation of S=3*Vt*Ia*
S_Mag=3*Vt_Mag*Ia_Mag;
S_Ang=Vt_Ang+-Ia_Ang;

// Polar to complex form
S_R=S_Mag*cos(-S_Ang*%pi/180);   // Real part of complex number S
S_I=S_Mag*sin(S_Ang*%pi/180);    // Imaginary part of
```

```
    complex number S

// (d) Power factor
Fp=cosd(Ia_Ang);

// (e) Excitation voltage
Efnew=eta*Ef_Mag;

// (f) Turbine speed
ns=120*f/P;

// Display result on command window
printf("\n Turbine torque supplied to the alternator
     = %0.1f lb-ft ",T);
printf("\n Excitation voltage = %0.1f V/phase ",Ef);
printf("\n Active components of apparent power= %0.0
   f kW ",S_R/1000);
printf("\n Reactive components of apparent power= %0
   .1f kvar lagging ",S_I/1000);
printf("\n Power factor = %0.2f lagging ",Fp);
printf("\n Excitation voltage new = %0.1f V/phase ",
   Efnew);
printf("\n Turbine speed = %0.0f r/min ",ns);
```

Scilab code Exa 9.2 Determine Speed regulation and Governor drop

```
// Example 9.2
// Determine (a) Speed regulation (b) Governor drop
// Page 351

clc;
clear;
close;
```

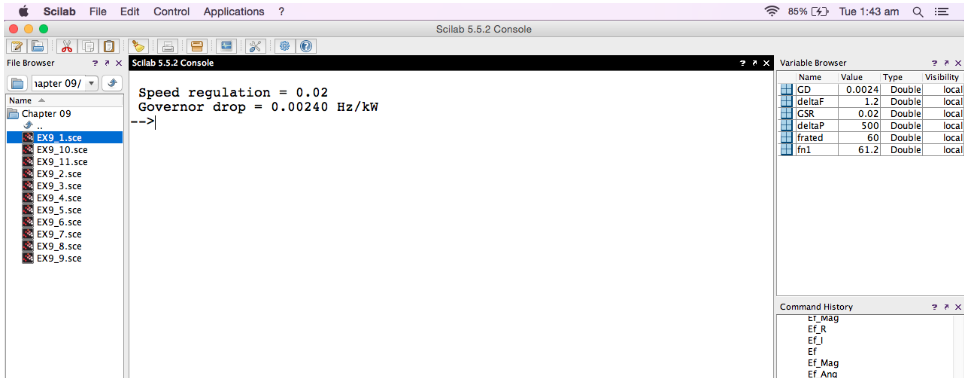

Figure 9.2: Determine Speed regulation and Governor drop

```

// Given data
fn1=61.2;                        // No-load frequency
frated=60;                       // Rated requency
deltaP=500;                      // Governor rated power
// (a) Speed regulation
GSR=(fn1-frated)/frated;

// (b) Governor drop
deltaF=(fn1-frated);             // Frequency difference
GD=deltaF/deltaP;

// Display result on command window
printf("\n Speed regulation = %0.2f  ",GSR);
printf("\n Governor drop = %0.5f Hz/kW ",GD);
```

Scilab code Exa 9.3 Determine Frequency of generator A and Frequency of generator B and Frequency of bus

```
// Example 9.3
```

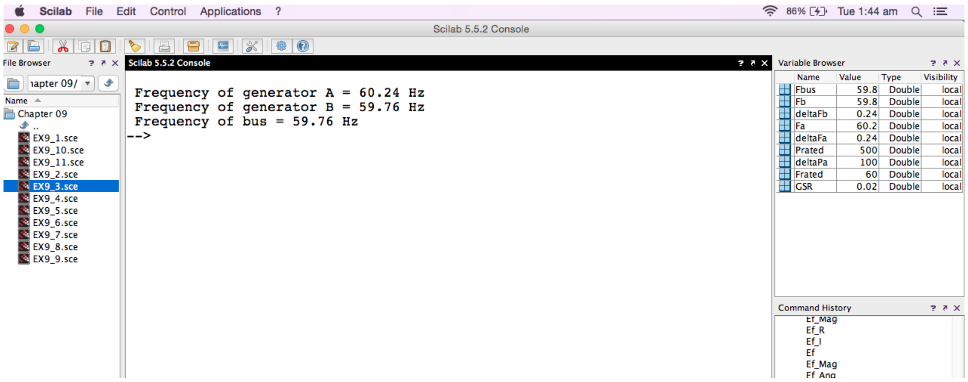

Figure 9.3: Determine Frequency of generator A and Frequency of generator B and Frequency of bus

```
// Determine (a) Frequency of generator A (b) Frequency of generator B
// (c) Frequency of bus
// Page 358

clc;
clear;
close;

// Given data
GSR=0.020;                     // Governor speed regulation
Frated=60;                     // Rated frequency
deltaPa=100;                   // Change in load (200-100 =100 KW)
Prated=500;                    // Rated power of both generators


// (a) Frequency of generator A
deltaFa=(GSR*Frated*deltaPa)/Prated; // Change in frequency due to change in load
Fa=Frated+deltaFa;                   // Frequency of generator A
```

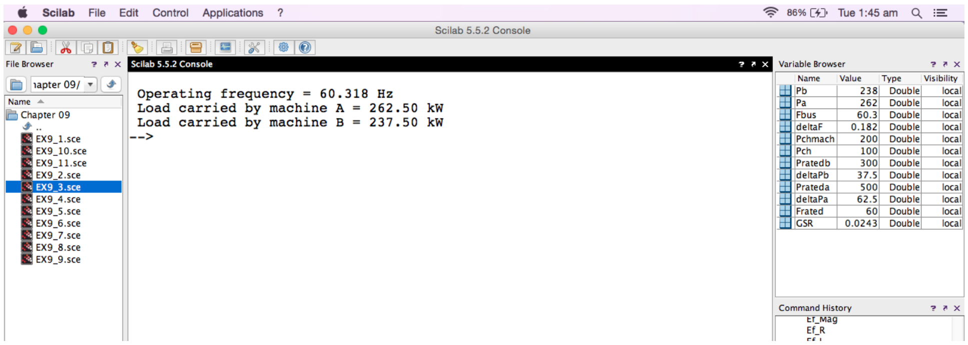

Figure 9.4: Determine Operating frequency and Load carried by each machine

```

// (b) Frequency of generator B
deltaFb=0.24;                                    // Since both machines are identical
Fb=Frated-deltaFb;

// (c) Frequency of bus
Fbus=Fb;                                         // Bus frequency is frequency of generator B

// Display result on command window
printf("\n Frequency of generator A = %0.2f Hz ",Fa);
printf("\n Frequency of generator B = %0.2f Hz ",Fb);
printf("\n Frequency of bus = %0.2f Hz ",Fbus);
```

Scilab code Exa 9.4 Determine Operating frequency and Load carried by each machine

```
// Example 9.4
// Determine (a) Operating frequency (b) Load carried by each machine
// Page 359

clc;
clear;
close;

// Given data
GSR=0.0243;                    // Governor speed regulation
Frated=60;                     // Rated frequency
deltaPa=500;                   // Change in load for alternator A
Prateda=500;                   // Rated power of alternator A
deltaPb=400;                   // Change in load for alternator B
Pratedb=300;                   // Rated power of alternator B
Pch=100;                       // Change is power (500−400=100 KW))
Pchmach=200;                   // Power difference (500−300=200 KW)

// (a) Operating frequency
// From the curve in figure 9.17
// GSR*Frated/Prated=deltaP/deltaP

deltaF=(deltaPa-deltaPb)/548.697;    // Change in frequency
Fbus=60.5-deltaF;


// (b) Load carried by each machine
deltaPa=(deltaF*Prateda)/(GSR*Frated); // Change in power for machine A
```

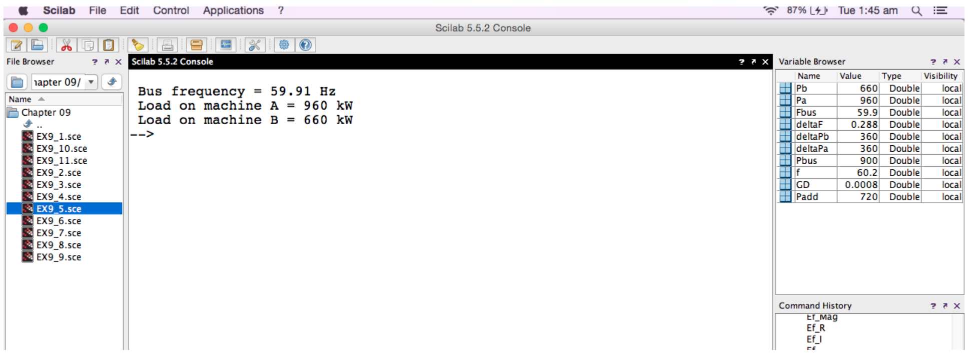

Figure 9.5: Determine Bus frequency and Load on each machine

```
deltaPb=Pch-deltaPa;                          // Change in power for machine B
Pa=Pchmach+deltaPa;
Pb=Pchmach+deltaPb;

// Display result on command window
printf("\n Operating frequency = %0.3f Hz ",Fbus);
printf("\n Load carried by machine A = %0.2f kW",Pa);
printf("\n Load carried by machine B = %0.2f kW",Pb);
```

Scilab code Exa 9.5 Determine Bus frequency and Load on each machine

```
// Example 9.5
// Determine (a) Bus frequency (b) Load on each machine
// Page 360

clc;
clear;
```

```
close;

// Given data
Padd=720;                          // Additional load
   connected
GD=0.0008;                         // Governor droop
f=60.2;                            // Frequency of machine
Pbus=900;                          // Bus load

// (a) Bus frequency
deltaPa=Padd/2;
deltaPb=deltaPa;                 // Since both machines
   have identical governor drops
deltaF=GD*deltaPa;               // Change in frequency
Fbus=f-deltaF;

// (b) Load on each machine
Pa=(2/3)*Pbus+deltaPa;           // Load on machine A
Pb=(1/3)*Pbus+deltaPb;           // Load on machine B

// Display result on command window
printf("\n Bus frequency = %0.2f Hz ",Fbus);
printf("\n Load on machine A = %0.0f kW",Pa);
printf("\n Load on machine B = %0.0f kW",Pb);
```

Scilab code Exa 9.6 Determine System kilowatts and System frequency and kilowatt loads carried by each machine

```
// Example 9.6
// Determine (a) System kilowatts (b) System
   frequency (c) kilowatt loads
// carried by each machine
// Page 361
```

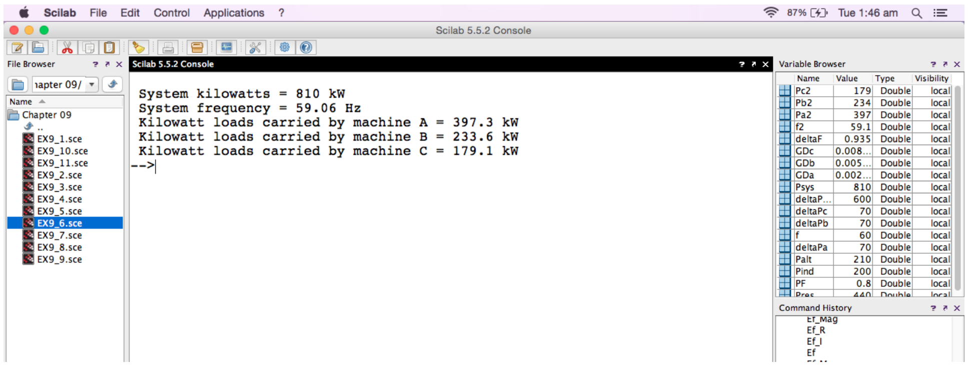

Figure 9.6: Determine System kilowatts and System frequency and kilowatt loads carried by each machine

```

clc;
clear;
close;

// Given data
Pres=440;                    // Resistive load
PF=0.8;                      // Power factor
Pind=200;                    // Induction motor power
Palt=210;                    // Alternator bus load
deltaPa=70;                  // Change in load for machine A
f=60;                        // Frequency
deltaPb=70;                  // Change in load for machine B
deltaPc=70;                  // Change in load for machine C

// (a) System kilowatts
deltaPbus=Pres+PF*Pind;     // Increase in bus load
Psys=Palt+deltaPbus;

// (b) System frequency
```

```
GDa=(60.2-f)/deltaPa;        // Governor droop for
    machine A
GDb=(60.4-f)/deltaPb;        // Governor droop for
    machine B
GDc=(60.6-f)/deltaPc;        // Governor droop for
    machine C
// From the figure 9.18(b)
deltaF=600/(350+175+116.6667) ;
f2=f-deltaF;

// (c) Kilowatt loads carried by each machine
Pa2=deltaPa+350*deltaF;
Pb2=deltaPb+175*deltaF;
Pc2=deltaPc+116.6667*deltaF;

// Display result on command window
printf("\n System kilowatts = %0.0f kW ",Psys);
printf("\n System frequency = %0.2f Hz",f2);
printf("\n Kilowatt loads carried by machine A = %0
    .1f kW",Pa2);
printf("\n Kilowatt loads carried by machine B = %0
    .1f kW",Pb2);
printf("\n Kilowatt loads carried by machine C = %0
    .1f kW",Pc2);
```

Scilab code Exa 9.7 Determine Active and reactive components of the bus load and Determine the reactive power supplied by each machine

```
// Example 9.7
// Determine (a) Active and reactive components of
    the bus load (b) If the
// power factor of generator A is 0.94 lagging,
    determine the reactive power
```

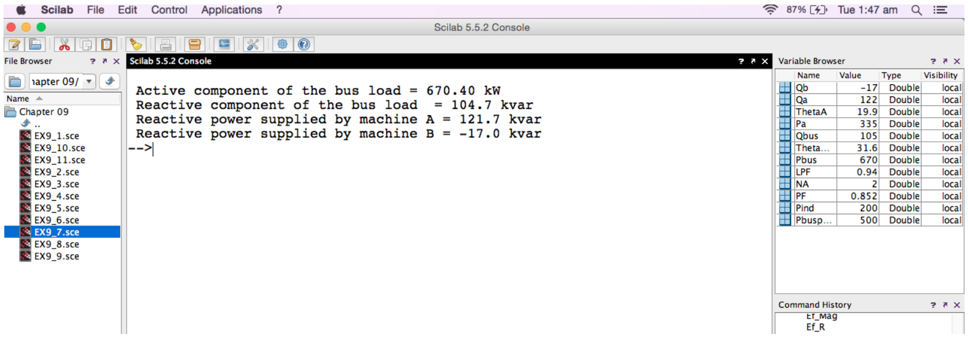

Figure 9.7: Determine Active and reactive components of the bus load and Determine the reactive power supplied by each machine

```
// supplied by each machine.
// Page 366

clc;
clear;
close;

// Given data
Pbuspower=500;                    // Power supplied
Pind=200;                         // Induction motor power
PF=0.852;                         // Percent power factor
NA=2;                             // Number of alternators
LPF=0.94;                         // Lagging power factor

// (a) Active and reactive components of the bus load
Pbus=Pbuspower+Pind*PF;           // Active component of the bus load
ThetaMot=acosd(PF);               // Power angle of motor
Qbus=Pind*sind(ThetaMot);         // Reactive component the bus load

// (b) Reactive power supplied by each machine
Pa=Pbus/NA;                       // Alternator A power
ThetaA=acosd(LPF);                // Alternator A angle
```

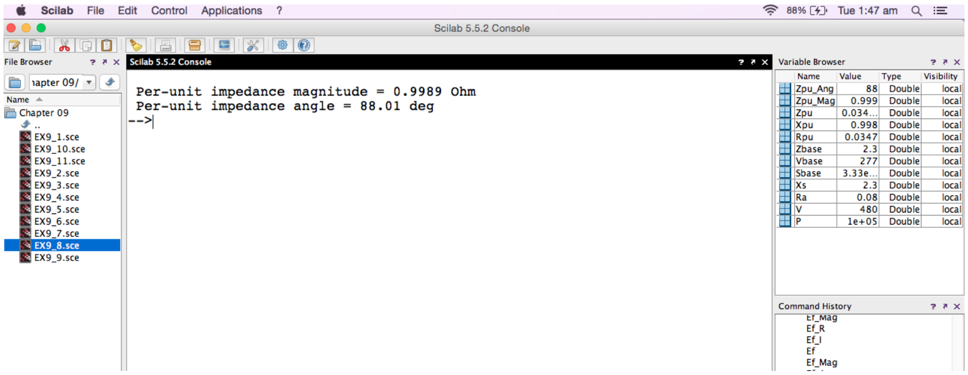

Figure 9.8: Computation of per unit impedance of a generator

```
Qa=tand(ThetaA)*Pa;              // Reactive power supplied by machine A
Qb=Qbus-Qa;                      // Reactive power supplied by machine B


// Display result on command window
printf("\n Active component of the bus load = %0.2f kW ",Pbus);
printf("\n Reactive component of the bus load  = %0.1f kvar",Qbus);
printf("\n Reactive power supplied by machine A = %0.1f kvar",Qa);
printf("\n Reactive power supplied by machine B = %0.1f kvar",Qb);
```

Scilab code Exa 9.8 Computation of per unit impedance of a generator

```
// Example 9.8
// Computation of per-unit impedance of a generator
// Page 368
```

```

clc;
clear;
close;

// Given data
P=100000;                    // Power of synchronous generator
V=480;                       // Voltage of synchronous generator
Ra=0.0800;                   // Resistive component
Xs=2.3;                      // Reactive component

// Computation of per-unit impedance of a generator
Sbase=P/3;                   // Rated apparent power per phase
Vbase=V/sqrt(3);             // Rated voltage per phase
Zbase=Vbase^2/Sbase;         // Rated impedance
Rpu=Ra/Zbase;                // Per unit resistance
Xpu=Xs/Zbase;                // Per unit reactance

Zpu=Rpu+%i*Xpu;              // Per unit impedance

// Complex to Polar form...
Zpu_Mag=sqrt(real(Zpu)^2+imag(Zpu)^2);        // Magnitude part
Zpu_Ang = atan(imag(Zpu),real(Zpu))*180/%pi; // Angle part

// Display result on command window
printf("\n Per-unit impedance magnitude = %0.4f Ohm ",Zpu_Mag);
printf("\n Per-unit impedance angle = %0.2f deg ",Zpu_Ang);
```

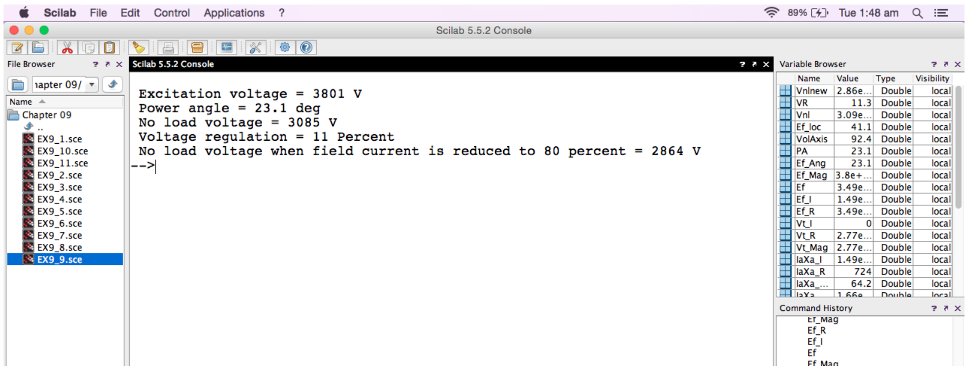

Figure 9.9: Determine Excitation voltage and Power angle and No load voltage and Voltage regulation and No load voltage if the field current is reduced to 80 percent

Scilab code Exa 9.9 Determine Excitation voltage and Power angle and No load voltage and Voltage regulation and No load voltage if the field current is reduced to 80 percent

```
// Example 9.9
// Determine (a) Excitation voltage (b) Power angle
   (c) No load voltage,
// assuming the field current is not changed (d)
   Voltage regulation (e) No load
// voltage if the field current is reduced to 80% of
    its value at rated load.
// Page 369

clc;
clear;
close;

// Given data
```

```
V=4800;                          // Voltage of synchronous generator
PF=0.900;                        // Lagging power factor
S_Mag=1000000/3;
Xa_Mag=13.80;                    // Synchronous reactance
Xa_Ang=90;
Vt_Ang=0;

// (a) Excitation voltage
Vt=V/sqrt(3);
Theta=acosd(PF);                     // Angle
Ia_Magstar=S_Mag/Vt;                 // Magnitude of curent
Ia_Angstar=Theta-0;                  // Angle of current
Ia_Mag=Ia_Magstar;
Ia_Ang=-Ia_Angstar;

// Ef=Vt+Ia*j*Xa
// First compute Ia*Xa
IaXa_Mag=Ia_Mag*Xa_Mag;
IaXa_Ang=Ia_Ang+Xa_Ang;
// Polar to Complex form for IaXa
IaXa_R=IaXa_Mag*cos(-IaXa_Ang*%pi/180); // Real part of complex number
IaXa_I=IaXa_Mag*sin(IaXa_Ang*%pi/180);  // Imaginary part of complex number
// Vt term in polar form
Vt_Mag=Vt;
Vt_Ang=Vt_Ang;
// Polar to Complex form for Vt
Vt_R=Vt_Mag*cos(-Vt_Ang*%pi/180);      // Real part of complex number
Vt_I=Vt_Mag*sin(Vt_Ang*%pi/180);       // Imaginary part of complex number
// Ef in complex form
Ef_R=IaXa_R+Vt_R;
Ef_I=IaXa_I+Vt_I;
Ef=Ef_R+%i*Ef_I;
```

```
// Complex to Polar form for Ef
Ef_Mag=sqrt(real(Ef)^2+imag(Ef)^2);         // Magnitude part
Ef_Ang= atan(imag(Ef),real(Ef))*180/%pi;    // Angle part

// (b) Power angle
PA=Ef_Ang;

// (c) No load voltage, assuming the field current is not changed
// From figure 9.23 (b)
VolAxis=Vt_Mag/30;          // The scale at the given voltage axis
Ef_loc=Ef_Mag/VolAxis;      // Location of Ef voltage
Vnl=33.4*VolAxis;           // No load voltage

// (d) Voltage regulation
VR=(Vnl-Vt)/Vt*100;

// (e) No load voltage if the field current is reduced to 80%
Vnlnew=31*VolAxis;

// Display result on command window
printf("\n Excitation voltage = %0.0f V ",Ef_Mag);
printf("\n Power angle = %0.1f deg ",PA);
printf("\n No load voltage = %0.0f V ",Vnl);
printf("\n Voltage regulation = %0.0f Percent ",VR);
printf("\n No load voltage when field current is reduced to 80 percent = %0.0f V ",Vnlnew);
```

Scilab code Exa 9.10 Determine Excitation voltage and Power angle and

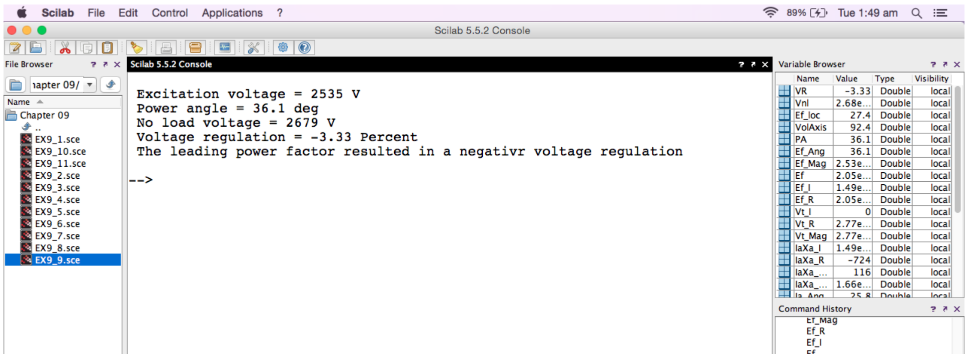

Figure 9.10: Determine Excitation voltage and Power angle and No load voltage and Voltage regulation and No load voltage if the field current is reduced to 80 percent

No load voltage and Voltage regulation and No load voltage if the field current is reduced to 80 percent

```
// Example 9.10
// Repeat the example 9.9 assuming 90 % leading
   power factor
// Determine (a) Excitation voltage (b) Power angle
   (c) No load voltage ,
// assuming the field current is not changed (d)
   Voltage regulation (e) No load
// voltage if the field current is reduced to 80% of
    its value at rated load.
// Page 372

clc;
clear;
close;

// Given data
V=4800;                         // Voltage of
   synchronous generator
PF=0.900;                       // Lagging power factor
S_Mag=1000000/3;
```

```
Xa_Mag=13.80;                    // Synchronous reactance
Xa_Ang=90;
Vt_Ang=0;

// (a) Excitation voltage
Vt=V/sqrt(3);
Theta=acosd(PF);                        // Angle
Ia_Magstar=S_Mag/Vt;                    // Magnitude of
   curent
Ia_Angstar=Theta-0;                     // Angle of current
Ia_Mag=Ia_Magstar;
Ia_Ang=Ia_Angstar;

// Ef=Vt+Ia*j*Xa
// First compute Ia*Xa
IaXa_Mag=Ia_Mag*Xa_Mag;
IaXa_Ang=Ia_Ang+Xa_Ang;
// Polar to Complex form for IaXa
IaXa_R=IaXa_Mag*cos(-IaXa_Ang*%pi/180); // Real part
    of complex number
IaXa_I=IaXa_Mag*sin(IaXa_Ang*%pi/180);  // Imaginary
    part of complex number
// Vt term in polar form
Vt_Mag=Vt;
Vt_Ang=Vt_Ang;
// Polar to Complex form for Vt
Vt_R=Vt_Mag*cos(-Vt_Ang*%pi/180);       // Real part
   of complex number
Vt_I=Vt_Mag*sin(Vt_Ang*%pi/180);        // Imaginary
   part of complex number
// Ef in complex form
Ef_R=IaXa_R+Vt_R;
Ef_I=IaXa_I+Vt_I;
Ef=Ef_R+%i*Ef_I;
// Complex to Polar form for Ef
Ef_Mag=sqrt(real(Ef)^2+imag(Ef)^2);          //
   Magnitude part
Ef_Ang= atan(imag(Ef),real(Ef))*180/%pi;   // Angle
```

```
part

// (b) Power angle
PA=Ef_Ang;

// (c) No load voltage, assuming the field current is not changed
// From figure 9.23 (b)
VolAxis=Vt_Mag/30;         // The scale at the given voltage axis
Ef_loc=Ef_Mag/VolAxis;     // Location of Ef voltage
Vnl=29*VolAxis;          // No load voltage

// (d) Voltage regulation
VR=(Vnl-Vt)/Vt*100;


// Display result on command window
printf("\n Excitation voltage = %0.0f V ",Ef_Mag);
printf("\n Power angle = %0.1f deg ",PA);
printf("\n No load voltage = %0.0f V ",Vnl);
printf("\n Voltage regulation = %0.2f Percent ",VR);
disp('The leading power factor resulted in a negativr voltage regulation')
```

Scilab code Exa 9.11 Determine Equivalent armature resistance and Synchronous reactance and Short circuit ratio

```
// Example 9.11
// Determine (a) Equivalent armature resistance (b) Synchronous reactance
// (c) Short-circuit ratio
// Page 377
```

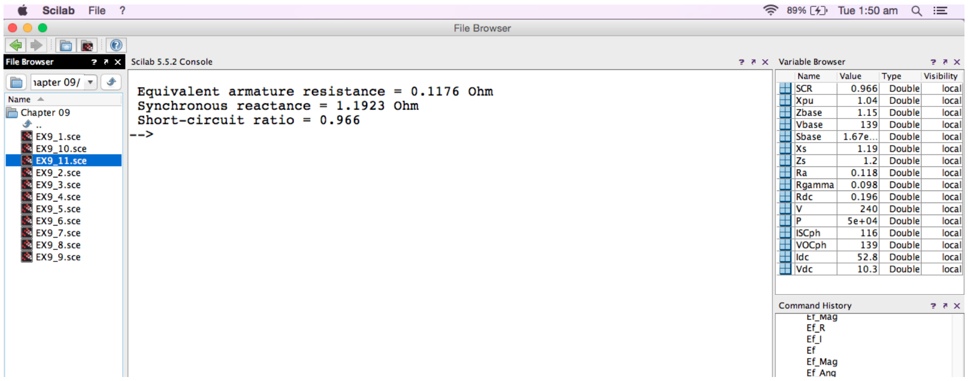

Figure 9.11: Determine Equivalent armature resistance and Synchronous reactance and Short circuit ratio

```

clc;
clear;
close;

// Given data
Vdc=10.35;                    // DC voltage
Idc=52.80;                    // DC current
VOCph=240/sqrt(3);            // Open-circuit phase voltage
ISCph=115.65;                 // Short-circuit phase current
P=50000;
V=240;                        // Supply voltage

// (a) Equivalent armature resistance
Rdc=Vdc/Idc;                  // DC resistance
Rgamma=Rdc/2;
Ra=1.2*Rgamma;                // Armature resistance

// (b) Synchronous reactance
Zs= VOCph/ISCph;             // Synchronous impedance/phase
Xs=sqrt(Zs^2-Ra^2);
```

```
// (c) Short-circuit ratio
Sbase=P/3;                    // Power/phase
Vbase=V/sqrt(3);              // Voltage/phase
Zbase=Vbase^2/Sbase;
Xpu=Xs/Zbase;                 // Per unit synchronous
    reactance
SCR=1/Xpu;                    // Short-circuit ratio


// Display result on command window
printf("\n Equivalent armature resistance = %0.4f
    Ohm ",Ra);
printf("\n Synchronous reactance = %0.4f Ohm ",Xs);
printf("\n Short-circuit ratio = %0.3f ",SCR);
```

Chapter 10

Principles of Direct Current Machines

Scilab code Exa 10.1 Computation of Induced emf and Frequency of the rectangular voltage wave in the armature winding

```
// Example 10.1
// Computation of (a) Induced emf (b) Frequency of
   the rectangular voltage
// wave in the armature winding
// Page No. 394

clc;
clear;
close;

// Given data
E1=136.8;                 // Generated emf
P=6;                      // Number of poles
n=1180;                   // Operating speed of
   machine

```

Figure 10.1: Computation of Induced emf and Frequency of the rectangular voltage wave in the armature winding

```
// (a) Induced emf

E2=E1*0.75*2;

// (b) Frequency of the rectangular voltage wave in
    the armature winding

f=P*n*0.75/120;

//Display result on command window
printf("\n Induced emf = %0.1f V ",E2);
printf("\n Frequency of the rectangular voltage wave
    = %0.2f Hz ",f);
```

Scilab code Exa 10.2 Computation of rheostat setting required to obtain an induced emf of 290 V

```
// Example 10.2
```

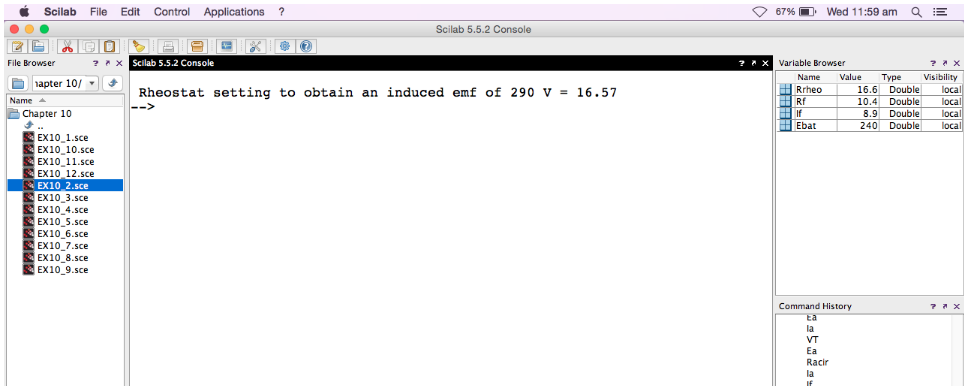

Figure 10.2: Computation of rheostat setting required to obtain an induced emf of 290 V

```
// Computation of rheostat setting required to
   obtain an induced emf of 290 V
// Page No. 399

clc;
clear;
close;

// Given data
Ebat=240;                    // Induced emf
If=8.9;                      // Field current
Rf=10.4;                     // Field resistance

// Rheostat setting required to obtain an induced
   emf of 290 V

Rrheo=(Ebat/If)-Rf;

// Display result on command window
printf("\n Rheostat setting to obtain an induced emf
    of 290 V = %0.2f  ",Rrheo);
```

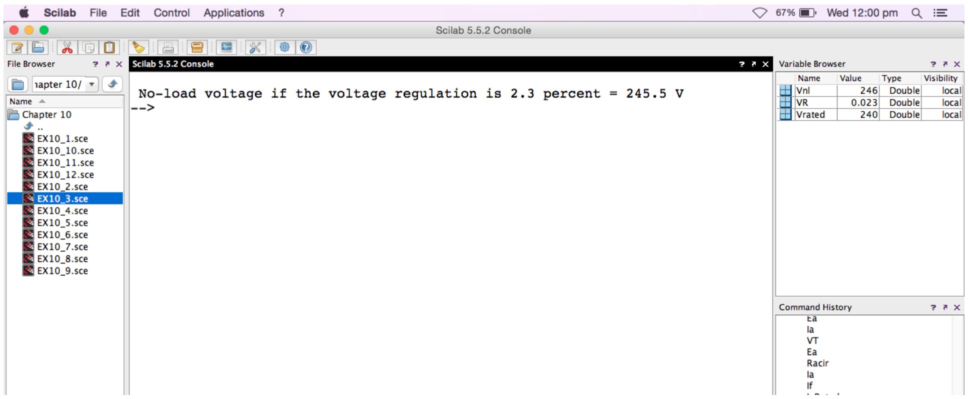

Figure 10.3: Computation of no load voltage

Scilab code Exa 10.3 Computation of no load voltage

```
// Example 10.3
// Computation of no-load voltage if the voltage
    regulation is 2.3 percent
// Page No. 401

clc;
clear;
close;

// Given data
Vrated=240;                    // Rated voltage
VR=0.023;                      // Voltage regulation


// No-load voltage if the voltage regulation is 2.3
    percent

Vnl=Vrated*(1+VR);

```

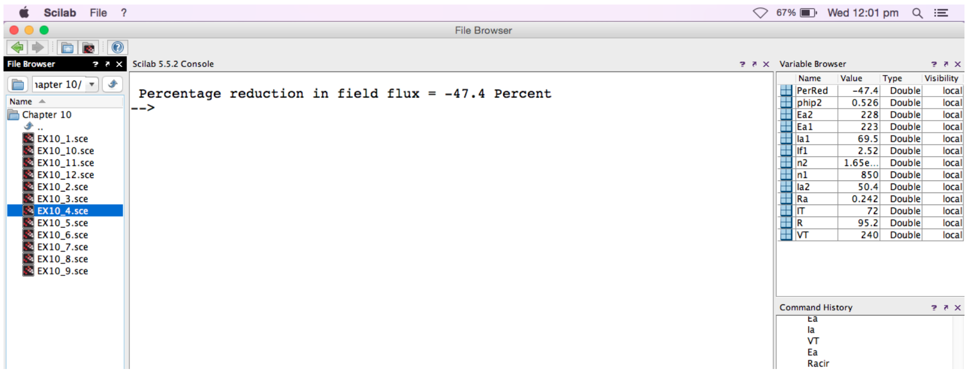

Figure 10.4: Computation of percentage reduction in field flux required to obtain a speed of 1650 rpm

```
// Display result on command window
printf("\n No-load voltage if the voltage regulation
    is 2.3 percent = %0.1f V ",Vnl);
```

Scilab code Exa 10.4 Computation of percentage reduction in field flux required to obtain a speed of 1650 rpm

```
// Example 10.4
// Computation of percentage reduction in field flux
    required to obtain a
// speed of 1650 r/min while drawing an armature
   current of 50.4 A.
// Page No. 405

clc;
clear;
close;

// Given data
VT=240;                    // Induced emf
```

Figure 10.5: Computation of no load speed

```
R=95.2;                     // Shunt field resistance
IT=72;                      // Total current
Ra=0.242;                   // Armature resistance
Ia2=50.4;                   // Armature current
n1=850;                     // Rated speed of shunt motor
n2=1650;                    // Speed of armature winding


// Percentage reduction in field flux

If1=VT/R;                   // Field current
Ia1=IT-If1;                 // Armature current
Ea1=VT-Ia1*Ra;              // Armature emf
Ea2=VT-Ia2*Ra;
phip2=(n1/n2)*(Ea2/Ea1);
PerRed=(phip2-1)*100;



// Display result on command window
printf("\n Percentage reduction in field flux = %0.1
    f Percent ",PerRed);
```

Scilab code Exa 10.5 Computation of no load speed

```
// Example 10.5
// Computation of no-load speed
// Page No. 408

clc;
clear;
close;

// Given data
nrated=1750;                    // Rated speed
SR=4;                           // Speed regulation

// No-load speed

Snl=nrated*(1+SR/100);

// Display result on command window
printf("\n No-load speed = %0.0f r/min ",Snl);
```

Scilab code Exa 10.6 Computation of Induced emf

```
// Example 10.6
// Computation of Induced emf
// Page No. 418

clc;
clear;
close;
```

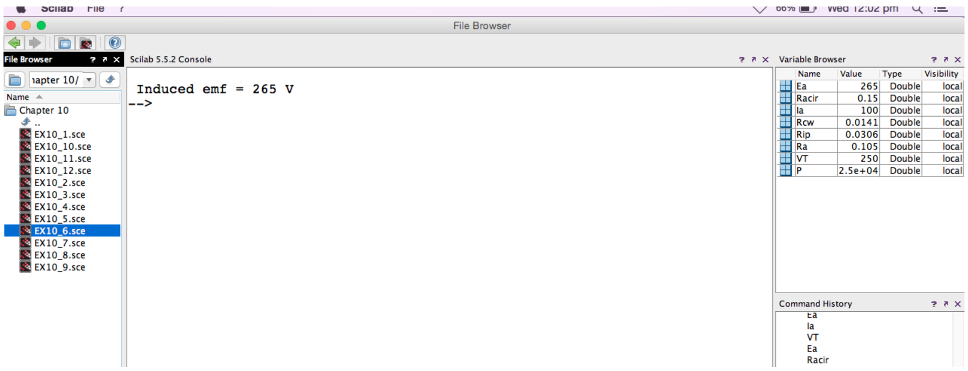

Figure 10.6: Computation of Induced emf

```

// Given data
P=25000;                // Power of the generator
VT=250;                 // Rated voltade of the machine
Ra=0.1053;              // Armature resistance
Rip=0.0306;             // Resistance of interpolar winding
Rcw=0.0141;             // Resistance of compensating windings


// Induced emf
Ia=P/VT;                // Armature current
Racir=Ra+Rip+Rcw;       // Resistance of armature circuit
Ea=VT+Ia*Racir;         // Induced emf


// Display result on command window
printf("\n Induced emf = %0.0f V ",Ea);
```

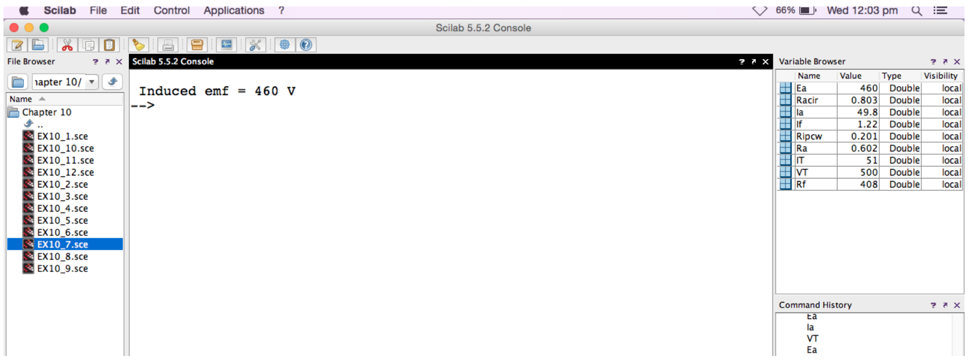

Figure 10.7: Computation of cemf

Scilab code Exa 10.7 Computation of cemf

```
// Example 10.7
// Computation of cemf
// Page No. 418

clc;
clear;
close;

// Given data
Rf=408.5;                  // Field resistance
VT=500;                    // Rated voltade of the machine
IT=51.0;                   // Total current
Ra=0.602;                  // Armature resistance
Ripcw=0.201;               // Resistance of interpolar winding and compensating windings

// Induced emf
If=VT/Rf;                // Current
Ia=IT-If;                // Armature current
```

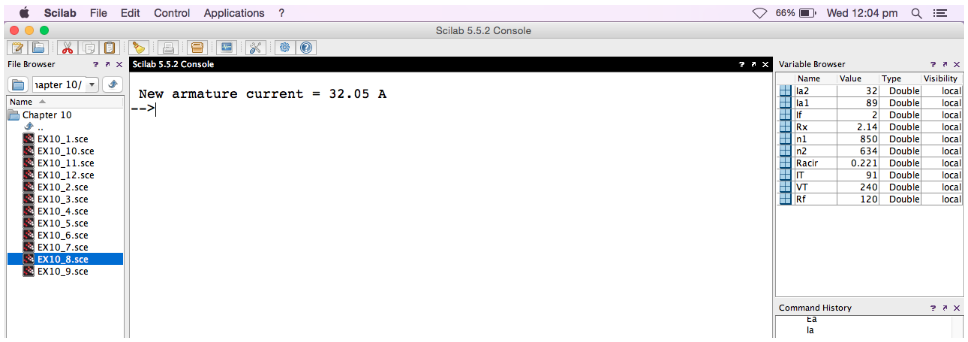

Figure 10.8: Computation of new armature current

```
Racir=Ra+Ripcw;           // Resistance of armature circuit
Ea=VT-Ia*Racir;


// Display result on command window
printf("\n Induced emf = %0.0f V ",Ea);
```

Scilab code Exa 10.8 Computation of new armature current

```
// Example 10.8
// Computation of new armature current
// Page No. 420

clc;
clear;
close;

// Given data
Rf=120;                   // Resistance of inserted resistor
```

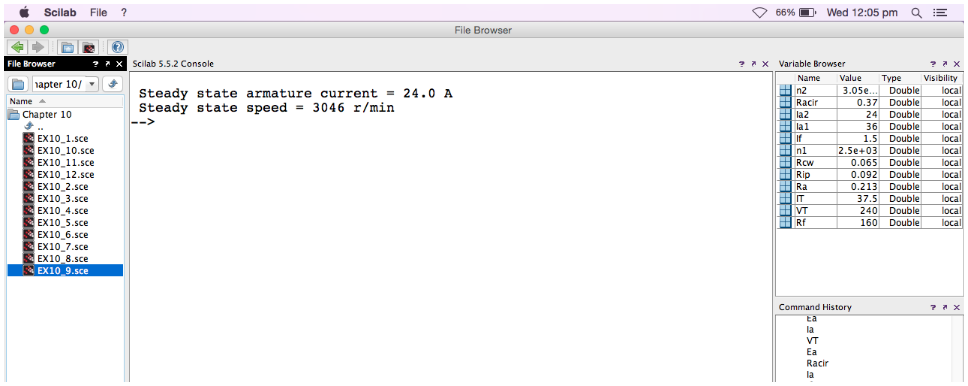

Figure 10.9: Computation of Steady state armature current and Steady state speed

```
VT=240;                       // Rated voltade of the
   machine
IT=91;                        // Total current
Racir=0.221;                  // Armature sircuit
   resistance
n2=634;                       // New speed after resistor
   was inserted
n1=850;                       // Rated speed OF THE MACHINE
Rx=2.14;                      // Resistance inserted in
   series witH armature

// New armature current

If=VT/Rf;                  // Resistor current
Ia1=IT-If;                 // Armature current
Ia2=(VT-(n2/n1)*(VT-Ia1*Racir))/(Racir+Rx);


// Display result on command window
printf("\n New armature current = %0.2f A ",Ia2);
```

Scilab code Exa 10.9 Computation of Steady state armature current and Steady state speed

```
// Example 10.9
// Computation of (a) Steady state armature current if a rheostat in the
// shunt field circuit reduces flux in air gap to 75 % of its rated value
// (b) Steady state speed for the conditions in (a)
// Page No. 421

clc;
clear;
close;

// Given data
Rf=160;                   // Field resistance
VT=240;                   // Rated voltade of the machine
IT=37.5;                  // Total current
Ra=0.213;                 // Armature resistance
Rip=0.092;                // Resistance of interpolar winding
Rcw=0.065;                // Resistance of compensating windings
n1=2500;                  // Rated speed of the machine


// (a) At rated conditions

If=VT/Rf;                   // Field current
Ia1=IT-If;                  // Armature current
Ia2=Ia1*0.50*1/0.75;

```

Figure 10.10: Computation of Mechanical power developed and Torque developed and Shaft torque

```
// (b) steady state speed for the above mentioned
    conditions

Racir=Ra+Rip+Rcw;

n2=n1*(VT-(Ia2*(1+Racir)))/0.75*(1/(VT-(Ia1*Racir)))
    ;


// Display result on command window

printf("\n Steady state armature current = %0.1f A "
    ,Ia2);
printf("\n Steady state speed = %0.0f r/min ",n2);
```

Scilab code Exa 10.10 Computation of Mechanical power developed and Torque developed and Shaft torque

```
// Example 10.10
```

```
// Computation of (a) Mechanical power developed (b)
    Torque developed
// (c) Shaft torque
// Page No.427

clc;
clear;
close;

// Given data
T=40;                       // Hp rating of motor
Rf=95.3;                    // Field resistance
VT=240;                     // Rated voltade of the
   machine
IT=140;                     // Total current
Racir=0.0873;               // Armature circuit
   resistance
n=2500;                     // Rated speed of the machine


// (a) The mechanical power developed

If=VT/Rf;                    // Field winding current
Ia1=IT-If;                   // Armature current
Ea=VT-Ia1*Racir;             // Armature emf
Pmech=Ea*Ia1;                // Mechanical power
Pmechhp=Ea*Ia1/746;

// (b) Torque developed

TD=7.04*Ea*Ia1/n;

// (c) Shaft torque

Tshaft=T*5252/n;

// Display result on command window
printf("\n Mechanical power developed= %0.0f W ",
```

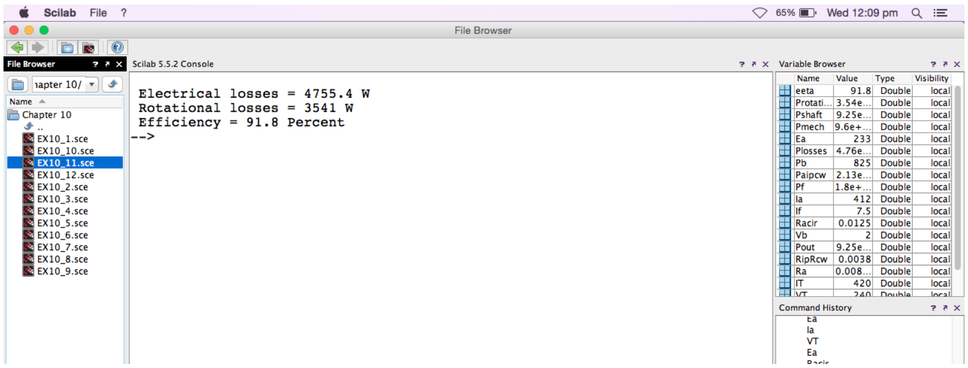

Figure 10.11: Determine Electrical losses and Rotational losses and Efficiency

```
    Pmech);
printf("\n Mechanical power developed= %0.1f hp ",
    Pmechhp);
printf("\n Torque developed = %0.1f lb-ft ",TD);
printf("\n Shaft torque = %0.1f lb-ft ",Tshaft);
```

Scilab code Exa 10.11 Determine Electrical losses and Rotational losses and Efficiency

```
// Example 10.11
// Determine (a) Electrical losses (b) Rotational
    losses (c) Efficiency
// Page No. 430

clc;
clear;
close;

// Given data
T=124;                   // Hp rating of motor
Rf=32.0;                 // Field resistance
```

```
VT=240;                    // Rated voltade of the machine
IT=420;                    // Total current
Ra=0.00872;                // Armature resistance
RipRcw=0.0038;             // Resistance of interpolar winding and compensating windings
Pout=92504;
Vb=2.0;                    // Rated speed of the machine
Racir=Ra+RipRcw;

// (a) Electrical losses

If=VT/Rf;                   // Field current
Ia=IT-If;                   // Armature current
Pf=If^2*Rf;                 // Field power
Paipcw=Ia^2*(Ra+RipRcw);
Pb=Vb*Ia;                   // Brush loss power
Plosses=Pf+Paipcw+Pb;       // Total power loss

// (b) Rotational losses

Ea=VT-(Ia*Racir)-Vb;           // Armature emf
Pmech=Ea*Ia;                // Mechanical power
Pshaft=T*746;               // Shaft power
Protational=Pmech-Pshaft;

// (c) Ffficiency

eeta=Pout/(VT*IT)*100;

// Display result on command window

printf("\n Electrical losses = %0.1f W ",Plosses);
printf("\n Rotational losses = %0.0f W ",Protational);
printf("\n Efficiency = %0.1f Percent ",eeta);
```

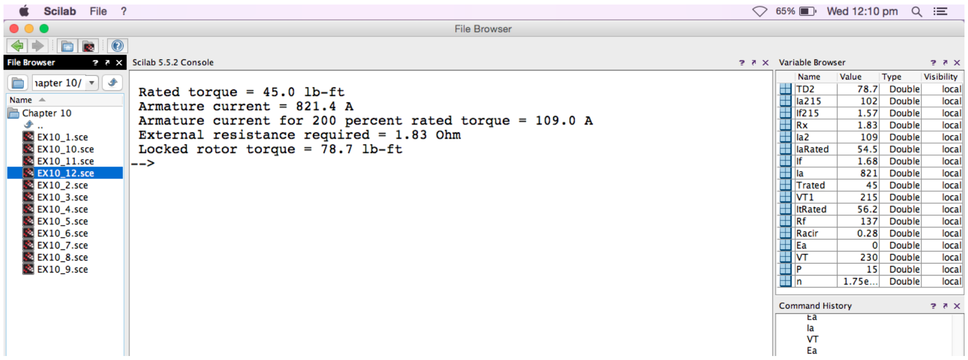

Figure 10.12: Determine Rated torque and Armature current at locked rotor and External resistance required in the armature circuit and Determine the locked rotor torque

Scilab code Exa 10.12 Determine Rated torque and Armature current at locked rotor and External resistance required in the armature circuit and Determine the locked rotor torque

```
// Example 10.12
// Determine (a) Rated torque (b) Armature current at locked rotor if no
// starting resistance is used (c) External resistance required in the armature
// circuit that would limit the current and develop 200 percent rated torque
// when starting (d) Assuming the system voltage drops to 215V, determine the
// locked rotor torque using the external resistor in (c)
// Page No. 433

clc;
```

```
clear;
close;

// Given data
n=1750;                         // Rotor speed
P=15;                           // Hp rating of motor
VT=230;                         // Rated voltade of the machine
Ea=0;
Racir=0.280;                    // Armature circuit loss
Rf=137;                         // Field resistance
ItRated=56.2;                   // Total current drawn
VT1=215;                        // Rated voltage after drop

// (a) Rated torque
Trated=P*5252/n;

// (b) Armature current
Ia=(VT-Ea)/Racir;

// (c) External resistance required
If=VT/Rf;                            // Field current
IaRated=ItRated-If;                  // Rated armature current

Ia2=IaRated*2;                       // Armature current for 200% rated torque

Rx=((VT-Ea)/Ia2)-Racir;              // External resistance required

// (d) Locked rotor torque
If215=VT1/Rf;                        // Field current at 215V
Ia215=(VT1-Ea)/(Racir+Rx);           // Armature current at 215V
TD2=Trated*( (If215*Ia215) / (If*IaRated) );
```

```
// Display result on command window

printf("\n Rated torque = %0.1f lb-ft ",Trated);
printf("\n Armature current = %0.1f A ",Ia);
printf("\n Armature current for 200 percent rated
    torque = %0.1f A ",Ia2);
printf("\n External resistance required = %0.2f Ohm
    ",Rx);
printf("\n Locked rotor torque = %0.1f lb-ft ",TD2);
```

Chapter 11

Direct Current Motor Characteristics and Applications

Scilab code Exa 11.1 Computation of Armature current when operating at rated conditions and Resistance and power rating of an external resistance required

```
// Example 11.1
// Computation of (a) The armature current when
   operating at rated conditions
// (b) The resistance and power rating of an
   external resistance required in
// series with the shunt field circuit to operate at
    125 percent rated speed
// Page No. 448

clc;
clear;
close;

```

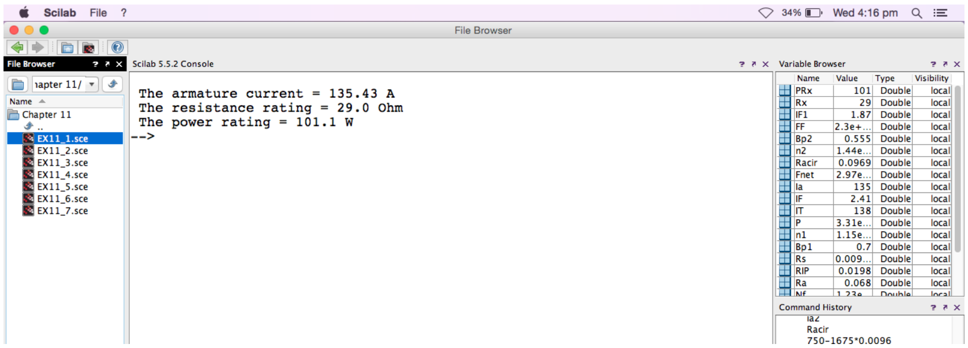

Figure 11.1: Computation of Armature current when operating at rated conditions and Resistance and power rating of an external resistance required

```
// Given data
HP=40;                    // hp rating of the device
Perratedload=0.902;       // Percentage rated load
VT=240;                   // Voltage value of motor
RF=99.5;                  // Resistance of shunt motor
Nf=1231;                  // Turns per pole of the
    shunt motor
Ra=0.0680;                // Armature resistance
RIP=0.0198;               // Interpole winding
    resistance
Rs=0.00911;               // Resistance of series
    field winding
Bp1=0.70;                 // Flux density for a net
    mmf
n1=1150;                  // Speed of shunt motor

// (a) The armature current when operating at rated
    conditions
P=HP*746/Perratedload;
IT=P/VT;                  // Total current
IF=VT/RF;                 // Field current
Ia=IT-IF;

// (b) The resistance and power rating of an
```

```
external resistance required in
// series with the shunt field circuit to operate at 125 percent rated speed

Fnet=Nf*IF;                        // Corresponding mmf from magnetization curve
Racir=Ra+RIP+Rs;
n2=n1*1.25;                        // 125 percent rated speed
// Shaft load is adjusted to  value that limits the armature current to 115%
// of rated current
Bp2=Bp1*(n1/n2)*((VT-Ia*Racir*1.15)/(VT-Ia*Racir))
FF=2.3*1000;
IF1=FF/Nf;
Rx=(VT/IF1)-RF;
PRx=(IF1^2)*Rx;

// Display result on command window
printf("\n The armature current = %0.2f A ",Ia);
printf("\n The resistance rating = %0.1f Ohm ",Rx);
printf("\n The power rating = %0.1f W ",PRx);

//Note: Answer varies due to round-off errors
```

Scilab code Exa 11.2 Computation of Shunt field current and Armature current and Developed torque and Armature current and External resistance required

```
// Example 11.2
// Computation of (a) Shunt field current (b) Armature current (c) Developed
// torque (d) Armature current if a resistor
```

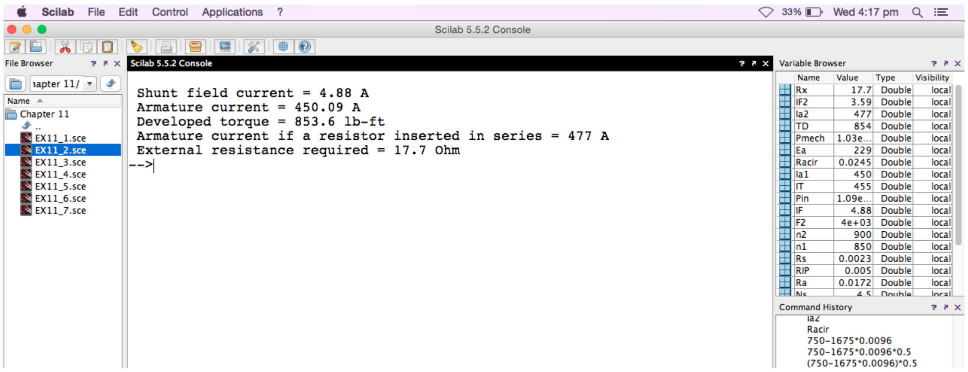

Figure 11.2: Computation of Shunt field current and Armature current and Developed torque and Armature current and External resistance required

```
    inserted in series with the shunt
// field circuit caused the speed to increase to 900
     r/min (e) External
// resistance required in series with the shunt
    field circuit to operate
// at 900 r/min
// Page No. 450

clc;
clear;
close;

// Given data
HP=125;
perratedload=0.854;              // Percentage rated
    load
VT=240;                          // Voltage value of
    motor
RF=49.2;                         // Resistance of shunt
    motor
Nf=577;                          // Turns per pole of
    the shunt motor
Ns=4.5;
Ra=0.0172;                       // Armature resistance
```

```
RIP=0.005;                    // Interpole winding resistance
Rs=0.0023;                    // Resistance of series field winding
n1=850;                       // Speed of shunt motor
n2=900;
F2=4000;

// (a) Shunt field current

IF=VT/RF;                     // Field current

// (b) Armature current
Pin=HP*746/perratedload;     // Input power
IT=Pin/VT;                   // Total current
Ia1=IT-IF;

// (c) Developed torque

Racir=Ra+RIP+Rs;
Ea=VT-Ia1*Racir;             // Armature emf
Pmech=Ea*Ia1;                // Mechanical power
TD=Pmech*5252/n1/746;             // Torque developed

// (d) Armature current if a resistor inserted in series with the shunt field
// circuit caused the speed to increase to 900 r/min

Ia2=Ia1*n2/n1;

//(e) External resistance required in series with the shunt field circuit to
// operate at 900 r/min
IF2=(F2-0.90*Ns*Ia2)/Nf;
Rx=(VT/IF2)-RF;


// Display result on command window
```

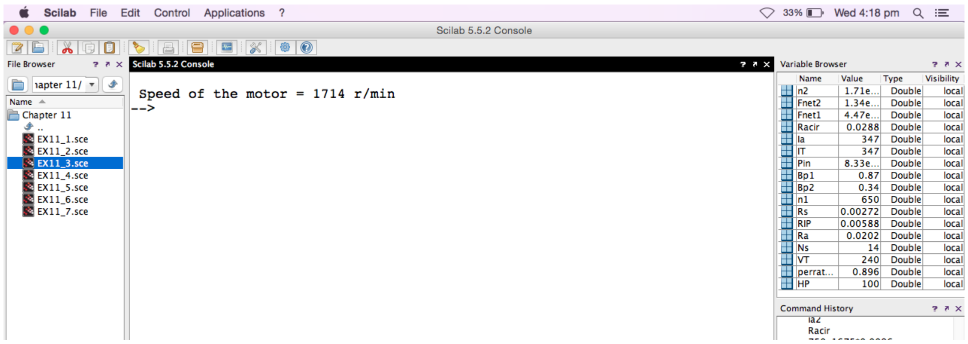

Figure 11.3: Computation of Speed if the load is reduced to a value that causes the armature current to be 30 percent of the rated current

```
printf("\n Shunt field current = %0.2f A ",IF);
printf("\n Armature current = %0.2f A ",Ia1);
printf("\n Developed torque = %0.1f lb-ft ",TD);
printf("\n Armature current if a resistor inserted in series = %0.0f A ",Ia2);
printf("\n External resistance required = %0.1f Ohm ",Rx);
```

Scilab code Exa 11.3 Computation of Speed if the load is reduced to a value that causes the armature current to be 30 percent of the rated current

```
// Example 11.3
// Computation of Speed if the load is reduced to a value that causes the
// armature current to be 30 percent of the rated current
// Page No.453

clc;
clear;
```

```
close;

// Given data
HP=100;
perratedload=0.896;      // Percentage rated load
VT=240;                  // Voltage value of motor
Ns=14;                   // Number of turns/pole in series field
Ra=0.0202;               // Armature resistance
RIP=0.00588;             // Interpole winding resistance
Rs=0.00272;              // Resistance of series field winding
n1=650;                  // Speed of shunt motor
Bp2=0.34;                // Air gap flux density from magnetization curve
Bp1=0.87;                // Air gap flux density from magnetization curve

// Computation of Speed if the load is reduced to a value that causes the
// armature current to be 30 percent of the rated current

Pin=HP*746/perratedload;        // Input power
IT=Pin/VT;                      // Total current
Ia=IT;                          // Armature current

Racir=Ra+RIP+Rs;                // Resistance of armature circuit
Fnet1=Ns*Ia*(1-0.080);          // Net mmf
Fnet2=0.30*Fnet1;               // Net mmf from magnetization curve
n2=n1/((VT-(Ia*Racir))/Bp1 * Bp2/(VT-(0.30*Ia*Racir)));

// Display result on command window
printf("\n Speed of the motor = %0.0f r/min ",n2);
```

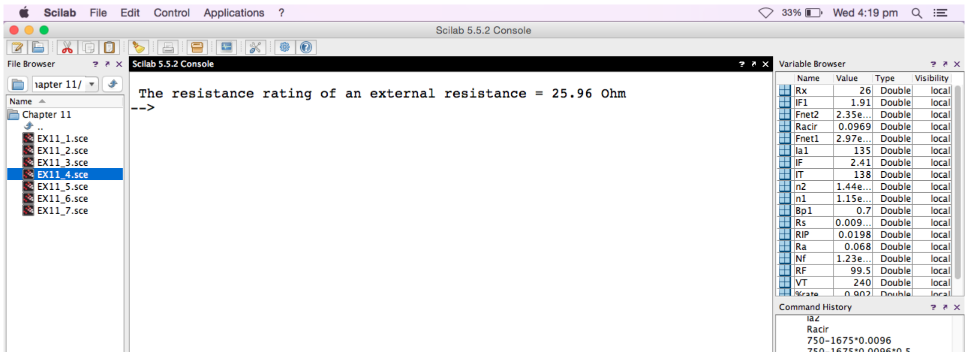

Figure 11.4: Computation of resistance using linear approximation and values are compared with results obtained in previous example

---

Scilab code Exa 11.4 Computation of resistance using linear approximation and values are compared with results obtained in previous example

```
// Example 11.4
// Computation of resistance using linear
    approximation and values are
// compared with results obtained in example 11.1
// Page No. 456
clc;
clear;
close;

// Given data
HP=40;                    // hp rating of the device
%ratedload=0.902;         // Percentage rated load
VT=240;                   // Voltage value of motor
RF=99.5;                  // Resistance of shunt
    motor
```

```
Nf=1231;                    // Turns per pole of the
    shunt motor
Ra=0.0680;                  // Armature resistance
RIP=0.0198;                 // Interpole winding
    resistance
Rs=0.00911;                 // Resistance of series
    field winding
Bp1=0.70;                   // Flux density for a net
    mmf
n1=1150;                    // Speed of shunt motor
n2=1.25*n1;
IT=137.84;
// Computation of resistance using linear
    approximation and values are
// compared with results obtained in example 11.1

IF=VT/RF;                   // Field current
Ia1=IT-IF;                  // Armature current
Fnet1=Nf*IF;                // Net mmf
Racir=Ra+RIP+Rs;            // Armature circuit
    resistance
Fnet2=Fnet1*(n1/n2)*((VT-Ia1*Racir*1.15)/(VT-Ia1*
    Racir));
IF1=Fnet2/Nf;              // Field current
Rx=(VT/IF1)-RF;            // External resistance
    required


// Display result on command window
printf("\n The resistance rating of an external
    resistance = %0.2f Ohm ",Rx);
```

Scilab code Exa 11.5 Computation using linear approximation to show

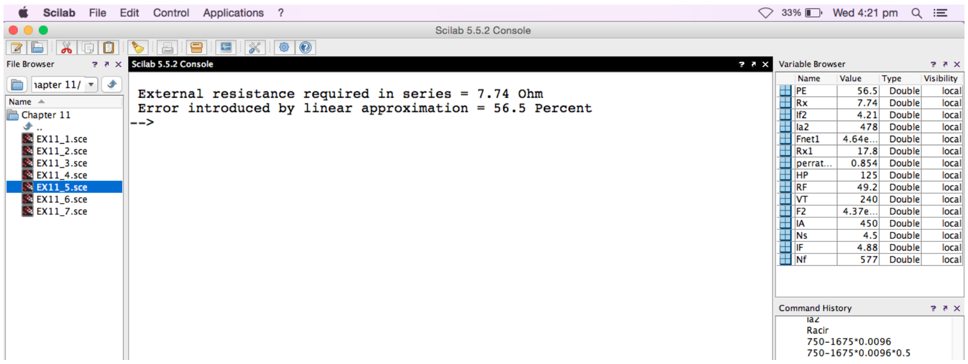

Figure 11.5: Computation using linear approximation to show the gross error that occurs when a linear assumption is applied to compound motors operating at overload conditions

the gross error that occurs when a linear assumption is applied to compound motors operating at overload conditions

```
// Example 11.5
// Computation  using linear approximation to show the gross error that occurs
// when a linear assumption is applied to compound motors operating at overload
// conditions
// Page No. 456

clc;
clear;
close;

// Given data
Nf=577;                    // Turns per pole of the shunt motor
IF=4.88;                   // Field current
Ns=4.5;
IA=450.09;                 // Armature current
F2=4367.8;                 // mmf
VT=240;                    // Voltage value of motor
```

```
RF=49.2;                     // Resistance of shunt motor
HP=125;
perratedload=0.854;          // Percentage rated load
Rx1=17.8;                    // Value of resistance in Example 11.2


Fnet1=(Nf*IF)+ (0.90 * Ns*IA);
Ia2=Fnet1*IA/F2;             // Armature current

If2=(F2 - Ns*Ia2*0.90)/Nf;
Rx=(VT/If2)-RF;             // External resistance required

// Error introduced by linear approximation
PE=(17.8-Rx)/17.8*100;

// Display result on command window
printf("\n External resistance required in series = %0.2f Ohm ",Rx);
printf("\n Error introduced by linear approximation = %0.1f Percent ",PE);
```

Scilab code Exa 11.6 Determine Torque developed when operating at rated speed and Developed torque required at half rated speed and Armature voltage required for half rated speed

```
// Example 11.6
// Determine (a) Torque developed when operating at rated speed (b) Developed
// torque required at half rated speed (c) Armature voltage required for half
```

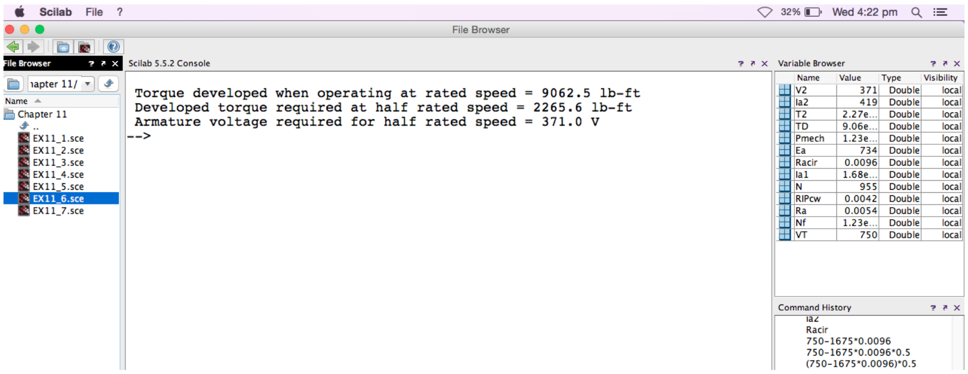

Figure 11.6: Determine Torque developed when operating at rated speed and Developed torque required at half rated speed and Armature voltage required for half rated speed

```
// rated speed
// Page No. 460

clc;
clear;
close;

// Given data

VT=750;                      // Voltage value of motor
Nf=1231;                     // Turns per pole of the
    shunt motor
Ra=0.00540;                  // Armature resistance
RIPcw=0.00420;               // Interpole winding
    resistance
N=955;                       // Speed of shunt motor
Ia1=1675;                    // Armature current

// (a) Torque developed when operating at rated
    speed

Racir=Ra+RIPcw;
Ea=VT-Ia1*Racir;
```

```
Pmech=Ea*Ia1;
TD=Pmech*5252/N/746;

// (b) Developed torque required at half rated speed

T2=TD*(0.5*N/N)^2;

// (c) Armature voltage required for half rated
   speed

Ia2=T2*Ia1/TD;
V2=(0.5*N/N)*(VT-Ia1*Racir) + Ia2*Racir ;

// Shaft load is adjusted to  value that limits the
   armature current to 115 % of rated current

//Display result on command window
printf("\n Torque developed when operating at rated
   speed = %0.1f lb-ft ",TD);
printf("\n Developed torque required at half rated
   speed = %0.1f lb-ft  ",T2);
printf("\n Armature voltage required for half rated
   speed = %0.1f V ",V2);
```

Scilab code Exa 11.7 Computation of the resistance of a dynamic braking resistor that will be capable of developing 500 lb ft of braking torque at a speed of 1000 rpm

```
// Example 11.7
// Computation of the resistance of a dynamic
   braking resistor that will be
// capable of developing 500 lb-ft of braking torque
    at a speed of 1000 r/min.
```

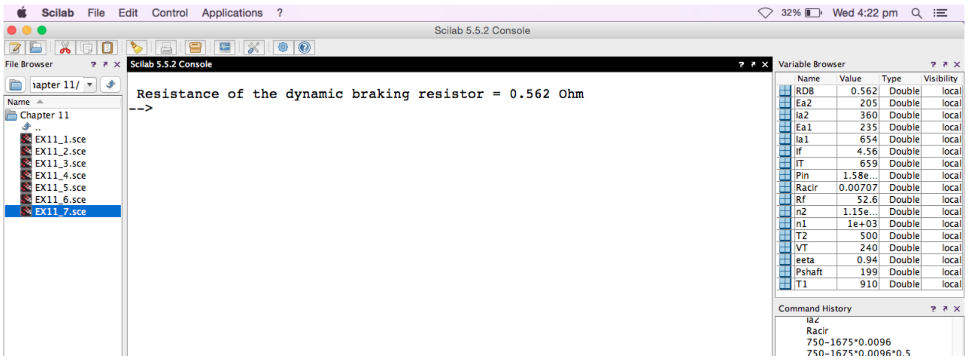

Figure 11.7: Computation of the resistance of a dynamic braking resistor that will be capable of developing 500 lb ft of braking torque at a speed of 1000 rpm

```
// Page No. 464

clc;
clear;
close;

// Given data
T1=910;                         // Torque load
Pshaft=199.257*746;             // Power of shaft
eeta=0.940;                     // Efficiency
VT=240;                         // Rated voltage
T2=500;                         // Braking torque
n1=1000;                        // Windage and friction speed
n2=1150;                        // Speed of motor
Rf=52.6;                        // Field resistance
Racir=0.00707;                  // Combined armature,compensating winding and
                                // interpolar resistance

// Resistance of a dynamic braking resistor
Pshaft=T1*n2/5252;            // Shaft power
```

```
Pin=Pshaft*746/eeta;                // Input power
IT=Pin/VT;                          // Total current
If=VT/Rf;                           // Field current
Ia1=IT-If;                          // Armature current
Ea1=VT-Ia1*Racir;                   // Armature emf

Ia2=Ia1*T2/T1;                      // Armature current
Ea2=Ea1*n1/n2;
RDB=(Ea2-Ia2*Racir)/Ia2;            // Resistance

//Display result on command window
printf("\n Resistance of the dynamic braking
   resistor = %0.3f Ohm ",RDB);
```

Chapter 12

Direct Current Generator Characteristics and Operation

Scilab code Exa 12.1 Determine Field circuit resistance and Field rheostat setting and Armature voltage and Field rheostat setting that will cause critical resistance and Armature voltage

```
// Example 12.1
// Determine (a) Field circuit resistance (b) Field
   rheostat setting that will
// provide no load voltage of 140V (c) Armature
   voltage if the rheostat is set
// to 14.23 ohm (d) Field rheostat setting that will
    cause critical resistance
// (e) Armature voltage at 80 percent rated speed (f
   ) Rheostat setting required
// to obtain no load armature voltage of 140V if
   shunt field is separately
// excited from a 120V DC source
// Page No. 479

clc;
```

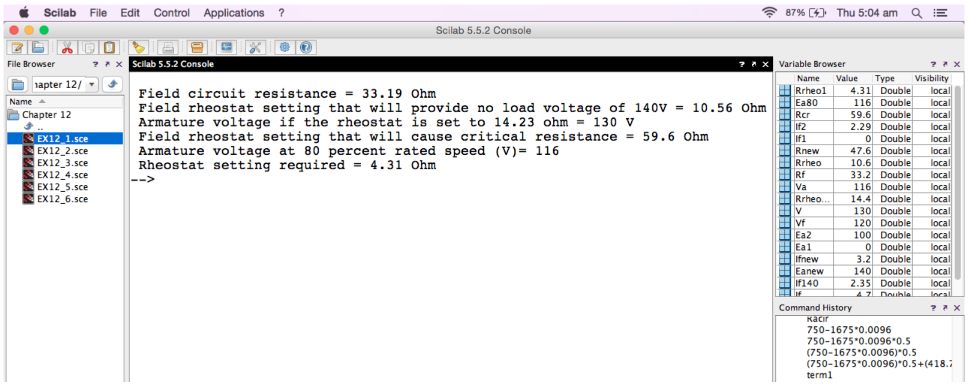

Figure 12.1: Determine Field circuit resistance and Field rheostat setting and Armature voltage and Field rheostat setting that will cause critical resistance and Armature voltage

```
clear;
close;

// Given data
Ea=156;                  // No load voltage
If=4.7;                  // Shunt field current
If140=2.35;              // New field current at Ea=140
    V
Eanew=140;               // No load voltage
Ifnew=3.2;               // Field current corresponding
      to no load voltage
Ea1=0;                   // First arbitrary voltage
Ea2=100;                 // Second arbitrary voltage
Vf=120;
V=130;                   // Intersection of I1 and I2
Rrheonew=14.42;          // Rheostat set to new
    settings
Va=116;                  // Intersection of field
    resistance line with the low
                         // speed magnetization curve


```

```
//(a) Field circuit resistance
Rf=Ea/If;       // Field circuit resistance

// (b) Field rheostat setting that will provide no load voltage of 140V
Rrheo=(Eanew/Ifnew)-Rf;

//(c) Armature voltage if the rheostat is set to 14.23 ohm
Rnew=Rf+Rrheonew;        // New field resistance
If1=Ea1/(Rf+Rrheo);     // Field current corresponding to first arbitrary voltage
If2=Ea2/(Rf+Rrheo);     // Field current corresponding to second arbitrary voltage

// (d) Field rheostat setting that will cause critical resistance
Rcr=Eanew/If140;          // Critical resistance

// (e) Armature voltage at 80 percent rated speed
// Ea80=0.80*Ea;
Ea80=116;

//(f) Rheostat setting required to obtain no load armature voltage of 140V if
// shunt field is separately excited from a 120V DC source
Rrheo1=(Vf/Ifnew)-Rf;

// Display result on command window
printf("\n Field circuit resistance = %0.2f Ohm",Rf);
printf("\n Field rheostat setting that will provide no load voltage of 140V = %0.2f Ohm ",Rrheo);
printf("\n Armature voltage if the rheostat is set to 14.23 ohm = %0.0f V ",V);
printf("\n Field rheostat setting that will cause
```

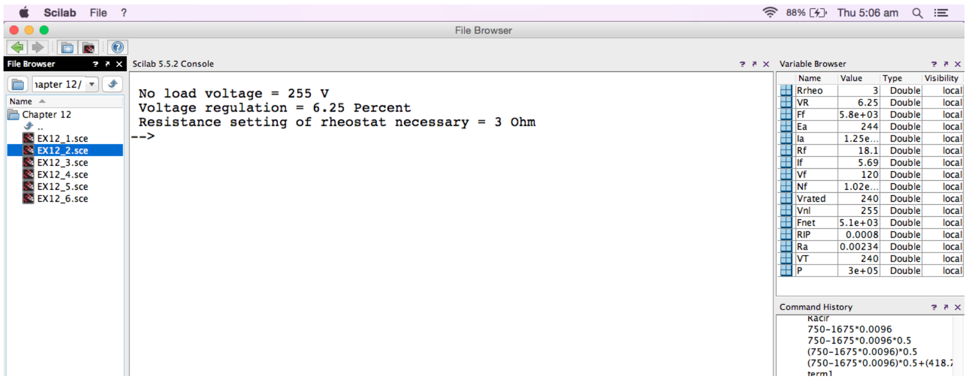

Figure 12.2: Computation of No load voltage and Voltage regulation and Resistance setting of rheostat necessary to obtain rated voltage

```
    critical resistance = %0.1f Ohm ",Rcr);
printf("\n Armature voltage at 80 percent rated
    speed (V)= %0.0f  ",Ea80);
printf("\n Rheostat setting required = %0.2f Ohm ",
    Rrheo1);
```

Scilab code Exa 12.2 Computation of No load voltage and Voltage regulation and Resistance setting of rheostat necessary to obtain rated voltage

```
// Example 12.2
// Computation of (a) No load voltage (b) Voltage
    regulation
// (c) Resistance setting of rheostat necessary to
    obtain rated voltage
// at rated conditions
// Page No. 487

clc;
clear;
```

```
close;

// Given data
P=300000;                  // Shunt generator power
    rating
VT=240;                    // Shunt generator voltage
    rating
Ra=0.00234;                // Armature winding
    resistance
RIP=0.00080;               // Resistance of interpole
    winding
Fnet=5100;                 // Net mmf
Vnl=255;                   // No load voltage
Vrated=240;                // Rated voltage
Nf=1020;                   // Turns per pole
Vf=120;                    // Source that separately
    excites the generator
If=5.69;
Rf=18.1;

// (a) No load voltage
Ia=P/VT;                  // Armature current
Ea=VT+Ia*(Ra+RIP);        // Armature emf
Ff=Fnet/(1-0.121);


// (b) Voltage regulation
VR=(Vnl-Vrated)*100/Vrated;

// (c) Resistance setting of rheostat necessary to
    obtain rated voltage at rated conditions
If=Ff/Nf;
Rrheo=(Vf/If)-Rf; // Rheostat setting


// Display result on command window
printf("\n No load voltage = %0.0f V ",Vnl);
printf("\n Voltage regulation = %0.2f Percent ",VR);
```

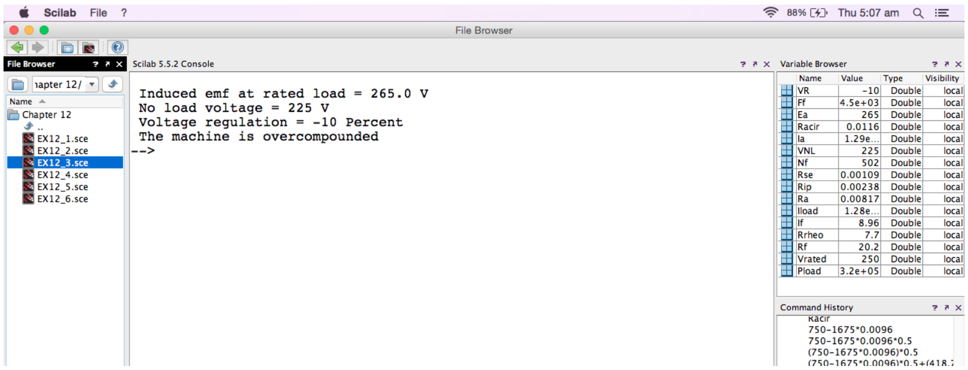

Figure 12.3: Computation of Induced emf at rated load and No load voltage and Voltage regulation

```
printf("\n Resistance setting of rheostat necessary
    = %0.0f Ohm",Rrheo);
```

Scilab code Exa 12.3 Computation of Induced emf at rated load and No load voltage and Voltage regulation

```
// Example 12.3
// Computation of (a) Induced emf at rated load (b)
    No load voltage
// (c) Voltage regulation (d) What is the type of
    compounding?
// Page No. 492

clc;
clear;
close;

// Given data
Pload=320000;                 // Shunt generator power
     rating
```

```
Vrated=250;                  // Shunt generator voltage rating
Rf=20.2;                     // Shunt resistance
Rrheo=7.70;                  // Shunt field rheostat value
If=8.96;                     // Field current
Iload=1280;                  // Load current
Ra=0.00817;                  // Armature resistance
Rip=0.00238;                 // Resistance of interpole winding
Rse=0.00109;                 // Resistance of series winding
Nf=502;                      // Turns per pole
VNL=225;                     // No load voltage

// (a) Induced emf at rated load
Iload=Pload/Vrated;           // Load current
If=Vrated/(Rf+Rrheo);         // Field current
Ia=If+Iload;                  // Armature current
Racir=Ra+Rip+Rse;
Ea=Vrated+Ia*Racir;

// (b) No load voltage
Ff=Nf*If;

// (c) Voltage regulation
VR=(VNL-Vrated)*100/Vrated;


// Display result on command window
printf("\n Induced emf at rated load = %0.1f V ",Ea);
printf("\n No load voltage = %0.0f V ",VNL);
printf("\n Voltage regulation = %0.0f Percent ",VR);
printf("\n The machine is overcompounded ");
```

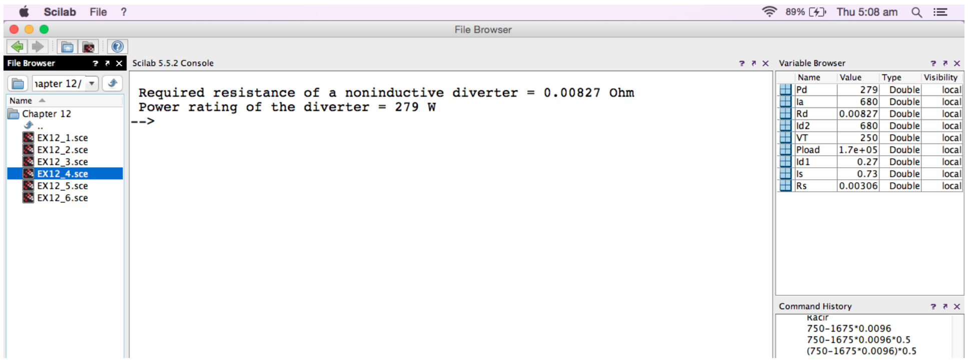

Figure 12.4: Computation of Required resistance of a noninductive diverter that will bypass 27 percent of the total armature current and Power rating of the diverter

Scilab code Exa 12.4 Computation of Required resistance of a noninductive diverter that will bypass 27 percent of the total armature current and Power rating of the diverter

```
// Example 12.4
// Computation of (a) Required resistance of a noninductive diverter that will
// bypass 27 percent of the total armature current(b) Power rating of the
// diverter
// Page No. 494

clc;
clear;
close;

// Given data
Rs=0.00306;                  // Shunt generator
```

```
resistance rating
Is=0.73;                    // Shunt generator current
    rating
Id1=0.27;                   // Armature winding
    resistance
Pload=170000;               // Load of power
VT=250;                     // Shunt generator voltage
    rating
Id2=680;                    // No load voltage
Rd=0.27;                    // Resistance drop

// (a) Required resistance of a noninductive
    diverter that will bypass
// 27 percent of the total armature current
Rd=Rs*Is/Id1;


// (b) Power rating of the diverter
Ia=Pload/VT;
Pd=((Id1*Id2)^2)*Rd;



//Display result on command window
printf("\n Required resistance of a noninductive
    diverter = %0.5f Ohm ",Rd);
printf("\n Power rating of the diverter = %0.0f W ",
    Pd);
```

Scilab code Exa 12.5 Computation of New bus voltage and Current supplied by each generator

```
// Example 12.5
```

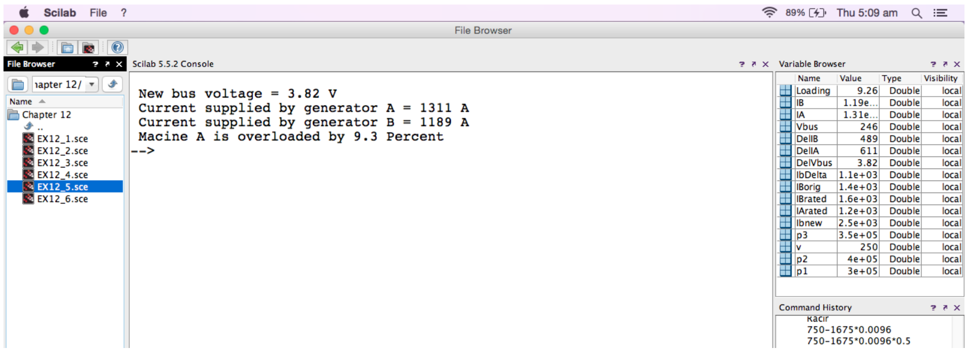

Figure 12.5: Computation of New bus voltage and Current supplied by each generator

```
// Computation of (a) New bus voltage (b) Current supplied by each generator
//Page No. 500

clc;
clear;
close;

// Given data
p1=300000;                // Rated power in generator A
p2=400000;                // Rated power in generator B
v=250;                    // Rated voltage in machine
p3=350000;                // Rated power in generator C
Ibnew=2500;

// (a) New bus voltage

IArated=p1/v;                // Rated current in generator A
IBrated=p2/v;                // Rated current in generator B
```

```
IBorig=p3/v;                    // Original bus current
IbDelta=Ibnew-IBorig;           // Current difference
DelVbus=IbDelta/(160+128);      // Voltage difference


// (b) Current supplied by each generator
DelIA=160*DelVbus;              // Generator A current difference
DelIB=128*DelVbus;              // Generator A current difference
Vbus=v-DelVbus;                 // Voltage across the bus
IA=700+DelIA;                   // Current in generator A
IB=700+DelIB;                   // Current in generator B

Loading= (IA-IArated)*100/IArated;


// Display result on command window
printf("\n New bus voltage = %0.2f V ",DelVbus);
printf("\n Current supplied by generator A = %0.0f A ",IA);
printf("\n Current supplied by generator B = %0.0f A ",IB);
printf("\n Macine A is overloaded by %0.1f Percent ",Loading);
```

Scilab code Exa 12.6 Determine the increment increase in load on each machine if an additional 400 A load is connected to the bus and Current carried by each machine

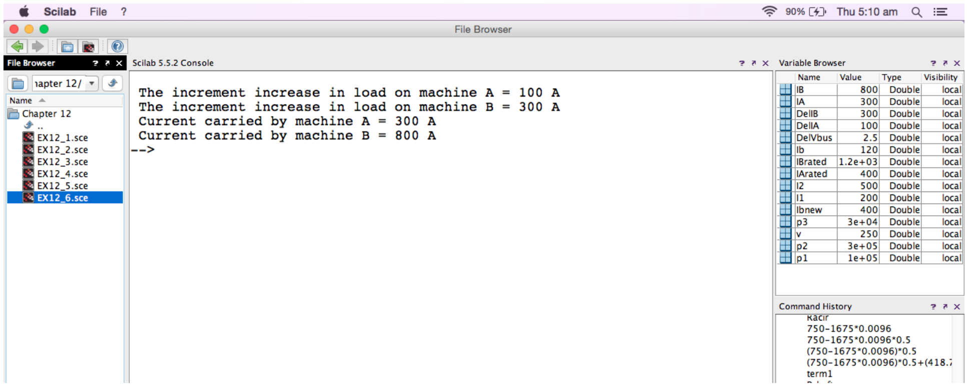

Figure 12.6: Determine the increment increase in load on each machine if an additional 400 A load is connected to the bus and Current carried by each machine

```
// Example 12.6
// Determine (a) The increment increase in load on each machine if an
// additional 400 A load is connected to the bus (b) Current carried
// by each machine
// Page No. 502

clc;
clear;
close;

// Given data
p1=100000;                  // Rated power in generator A
p2=300000;                  // Rated power in generator B
v=250;                      // Rated voltage in machine
p3=30000;                   // Rated power in generator C
Ibnew=400;                  // New bus current
I1=200;
```

```
I2=500;

// (a) The increment increase in load on each
    machine if an additional 400 A
// load is connected to the bus

IArated=p1/v;                           // Rated current
    in generator A
IBrated=p2/v;                           // Rated current
    in generator B
Ib=p3/v;                                // Original bus
    current
DelVbus=Ibnew/(40+120);                 // Change in bus
    current
DelIA=40*DelVbus;
DelIB=120*DelVbus;


// (b) Current carried by each machine

IA=I1+DelIA;                      // Current in generator
    A
IB=I2+DelIB;                      // Current in generator
    B


// Display result on command window
printf("\n The increment increase in load on machine
     A = %0.0f A ",DelIA);
printf("\n The increment increase in load on machine
     B = %0.0f A ",DelIB);
printf("\n Current carried by machine A = %0.0f A ",
    IA);
printf("\n Current carried by machine B = %0.0f A ",
    IB);
```