

Scilab Textbook Companion for
A First Course on Electrical Drives
by S. K. Pillai¹

Created by
Byragoni Pranavi
BE
Electrical Engineering
Jawaharlal Nehru Tehnological University Hyderabad
College Teacher
None
Cross-Checked by
Lavitha

July 31, 2019

¹Funded 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: A First Course on Electrical Drives

Author: S. K. Pillai

Publisher: Wiley Eastern, India

Edition: 2

Year: 1989

ISBN: 81-224-0166-X

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
4 Characteristics of AC motor	5
5 Starting	8
6 Electric Braking	11
7 Rating and Heating Motors	17

List of Scilab Codes

Exa 4.1	Roots of the equation	5
Exa 4.3	Steady operation	5
Exa 4.5	Value of resistance	6
Exa 4.6	Value of Delta	7
Exa 5.1	Value of Torque	8
Exa 5.3	Torque	8
Exa 5.4	Permissible start	9
Exa 6.1	Speed of shunt machine	11
Exa 6.2	Resistance	11
Exa 6.3	Braking Torque and current	12
Exa 6.4	Braking Torque	13
Exa 6.5	Initial braking torque	14
Exa 6.6	number of starts and stops	14
Exa 6.7	Minimum time to bring rotor to rest	15
Exa 6.8	Number of revolutions made	16
Exa 7.1	Final tempearture	17
Exa 7.3	power	17
Exa 7.4	Power	18
Exa 7.5	Weight of fly wheel	18

Chapter 4

Characteristics of AC motor

Scilab code Exa 4.1 Roots of the equation

```
1 //Example 4.1, Page 72
2 clc;
3 p1=poly([1, -3, 1], 'x', 'c')//Polynomial equation
4 disp('Part a')
5 disp('roots of the equation when slip at max torque'
      )
6 x=roots(p1)
7 disp(x)
8
9
10 p2=poly([0.146, -1.719, 1], 'x', 'c')//polynomial
    equation in scilab function
11 y=roots(p2)
12 disp('Part b')
13 disp('roots of the equation when slip at max load')
14 disp(y)
```

Scilab code Exa 4.3 Steady operation

```

1 //Example 4.1, Page 76
2 clc;
3 p1=poly([.0016, -0.0266, 0.04], 'x', 'c')//
   Polynomial equation
4 disp('Part a')
5 disp('roots of the equation that slip will run is')
6 x=roots(p1)
7 disp(x)

```

Scilab code Exa 4.5 Value of resistance

```

1 //Example 4.5, page 81
2 clc
3 pole=24
4 Ns=245//in rpm
5 N=(120*50)/pole//synchronous speed in rpm
6 f=(N-Ns)/N
7 p=110//in kw
8 T=(p*1000*60)/(2*pi*Ns)
9 v1=440/sqrt(3)//in v
10 ws=(2*pi*250)/60
11 s=0.02
12 R=0.03125//in ohm
13 x=sqrt(((3*R*v1**2)/(T*ws*s))-(R/s)**2)//by
   rearranging formula
14 printf("\n Stator resistance per phase is %f ohm ",x
   )
15 //calculating original resistance
16
17 //Example 4.1, Page 72
18 clc;
19 p1=poly([72.78, -3235, 3190], 'x', 'c')//Polynomial
   equation
20 disp('Part a')
21 disp('The value of original resistance is')

```

```
22 x=roots(p1)
23 disp(x)
24
25 //Taking r=0.99108
26 r=(0.99108-R)/1.25**2
27
28 printf("\n The value of resistance to be added is
        %f ohm ",r)
```

Scilab code Exa 4.6 Value of Delta

```
1 //Example 4.6, Page no 92
2 clc
3 disp("Part ii")
4 new_sin_delta=sind(45)/.95
5 delta=asind(new_sin_delta)
6 printf("\n The value of delta is %f degree ",delta)
```

Chapter 5

Starting

Scilab code Exa 5.1 Value of Torque

```
1 //example 5.1, Page 98
2 clc;
3 IL=(100*1000)/(400*.9*.8*sqrt(3))//Full load current
4 FS=(1000-950)/1000//Full load slip
5 //disp(" hhhhhhhhhhh")
6 //Writing loop
7 x=[1 0.577350269 .7 0.333333333]
8 for i=1:1:4
9 T=1.8*(x(i))^2
10 printf("\n The value of Tst/Tf when x=%f is %f ",x(i)
    ),T)
11 end
```

Scilab code Exa 5.3 Torque

```
1 //Example 5.3, page 104
2 clc
3 s=.05//slip
```

```

4 x=1//ie x1+x2
5 s_max=((1-s**2)/(2*log(1/s)))**(1/2)//max slip
6 //disp(s_max)
7 R2_opt=x*s_max
8 ws=(2*pi*1000)/60//angular frequency
9 v1=400/sqrt(3)//voltage
10 j=10//angular V
11 Tmax=(3*v1**2)/(ws*2*x)
12 Tmin=((j*ws)/(2*Tmax))*(((1-s**2)/(2*R2_opt))+((
    R2_opt*log(1/s))))
13 printf("\n The value of T_min is %f sec ",Tmin)

```

Scilab code Exa 5.4 Permissible start

```

1 //Example 5.4, Page no 110
2 clc
3 p=400*20*.88*sqrt(3)//input power in watt
4 l=12193.6-10000//in watt, full load loss
5 e=1*60//energy lost per minute
6 ws=(2*pi*1000)/60//angular frequency
7 j=.5
8 R=1//resistance in ohm ir R1/R2
9 E1=(0.5*j*ws**2)*(1+R)
10 //disp(E1)
11 N=e/E1
12 //disp(N)
13 printf("\n The number of starts that can be made is
    %d ",N)
14 ws=(2*pi*1500)/60//angular frequency
15 j=.5
16 R=1//resistance in ohm ir R1/R2
17 E1=(0.5*j*ws**2)*(1+R)
18 //disp(E1)
19 N=e/E1
20 //disp(N)

```

```
21 printf("\n The number of permissible starts is %d ",  
        N)
```

Chapter 6

Electric Braking

Scilab code Exa 6.1 Speed of shunt machine

```
1 //Example 6.1, page 121
2 clc
3 T=172//in N-m
4 w=(2*%pi*960)/60
5 E=215//in V
6 Ia=(T*w)/E
7 Ra=.062//in ohm
8 v=220//in v
9 Eg=v+(Ia*Ra)
10 //disp(Eg)
11 N=960//in Rpm
12 S=(N*Eg)/E
13 //disp(S)
14 printf("\\n The speed of shunt machine is %f rpm ",S)
```

Scilab code Exa 6.2 Resistance

```
1 //Example 6.2, page 122
```

```

2  clc;
3  N=480//in rpm
4  T=318.3//in N-m
5  P=(2*%pi*N*T)/60
6  //From graph
7  E=333.3//in V
8  Ia=48//Amp
9  R=E/Ia
10 //disp(R)
11 printf("\n The total resistance of circuit is %f ohm
        ",R)

```

Scilab code Exa 6.3 Braking Torque and current

```

1  //Example 6.3, Page 123
2  clc
3  per=.88
4  v=220//in v
5  p=20//in kw
6  I=(p*1000)/(per*v)
7  T=(p*1000*60)/(2*%pi*1200)
8  //Part a
9  E_motor=v-(I*.1)
10 v_arm=v+E_motor
11 Ir=2*I//Rated current
12 R=v_arm/Ir
13 disp(R)
14 R_Extra=R-.1
15 printf("\n Extra resistance added to motor armature
        is %f ohm ",R_Extra)
16 //Part b
17 T_Full_load=T*2
18 printf("\n Full load torque is %f N-m ",T_Full_load)
19 //Part c
20 E=(E_motor*400)/1200

```

```

21 //disp(E)
22 I_braking=(v+E)/R
23 T_braking=(T/103.3)*I_braking
24 printf("\n Braking Torque is %f N-m ",T_braking)
25 printf("\n Braking current is %f A",I_braking)

```

Scilab code Exa 6.4 Braking Torque

```

1 //Example 6.4 , page 131
2 clc
3 R1=.15//in ohm
4 Rs=.45//in ohm
5 x1=.6//in ohm
6 xz=1.8//in ohm
7 sf=.05
8 Turn=1/sqrt(3)
9 R_rotor=Rs*Turn//in ohm
10 X_rotor=xz*Turn^2//in ohm
11
12 //Part 1
13 //BY FIGURE
14 E1=sqrt((3^2+.6^2)/(3.15^2+1.2^2))*440/sqrt(3)
15 //disp(E1)
16 s=1-sf
17 I2=E1/sqrt(x1**2+2**2)
18 //disp(I2)
19 R2=2*60//ohm
20 w=2*pi*600
21 T=(R2*3*(I2^2))/(s*w)
22 //disp(T)
23 printf("Initial braking torque of rheostatic is %f N
      -m",T)
24
25 //Part 2
26 s1=2-sf

```

```

27 a=.15+(1.9/1.95)**2
28 b=1.2**2
29 I2=(440/sqrt(3))*(1/sqrt(a+b))
30 //disp(I2)
31 T=(60*1.9*3*(I2^2))/(1.95*w)
32 //disp(T)
33 printf("\n Initial braking torque during reverse is
        %f N-m",T)

```

Scilab code Exa 6.5 Initial braking torque

```

1 //Example 6.5 , page no 136
2 clc
3 Kva=3000/3//kva per phase
4 v=2300/sqrt(3)//voltage per phase
5 //disp(v)
6 i=(1000*1000)/1330//current per phase
7 //disp(i)
8 s=i*.2
9 x=sqrt((v+s)^2+(s^2))
10 temp=((x/i)^2)-2**2//temp=(.2+R)**2
11 temp1=sqrt(temp)-(.2)
12 //disp(temp1)
13 //Answer difference is because of round off value of
    x
14 r=1.97//in ohms
15 T_br=(3*i*i*r*60)/(2*pi*200)
16 printf("Initial braking torque is %f N-m",T_br)

```

Scilab code Exa 6.6 number of starts and stops

```

1 //Example 6.6 , Page 138
2 clc

```

```

3 pf=.85
4 N1=1500//in rpm
5 N=1440//in rpm
6 P=pf*230*10*sqrt(3)
7 p_stator_loss=86.16//in w
8 p_rotor=P-p_stator_loss
9 rotor_copper_loss=((N1-N)/N1)*p_rotor
10 printf("The rotor copper loss is %f watt",
        rotor_copper_loss)
11 inertia=.0486//in kg-m2
12 E=2*.96*inertia*((2*pi*50)/2)**2
13 disp(E)
14 stops_starts=7920/E
15 printf("Total number of starts and stops is %d",
        stops_starts)

```

Scilab code Exa 6.7 Minimum time to bring rotor to rest

```

1 //Example 6.7, page 138
2 clc;
3 v1=400/sqrt(3)//in v
4 ws=(2*pi*1000)/60//angular f
5 x=1//resistance in ohm
6 T_max=(3*v1^2)/(ws*2*x)
7 //disp(T_max)
8 j=10//in kg-m2
9 s1=.05
10 s_maxT=0.2
11 a=(1.95^2-1)/(2*s_maxT)
12 temp=a+(.2*log(1.95))
13 r=((10*ws)/(2*T_max))*(temp)
14 //disp(r)
15 Extra_R=r-(.2)
16 a=(1.95^2-1)/(2*1.45)
17 temp=a+(1.45*log(1.95))

```



```

18 t=((10*ws)/(2*T_max))*(temp)
19 //disp(t)
20 printf("Minimum time to bring rotor to rest is %f s"
        ,t)

```

Scilab code Exa 6.8 Number of revolutions made

```

1 //Example 6.8, page 141
2 clc
3 //part a
4 w=(2*pi*50)/3//angular f, rad/sec
5 k=6000/w
6 kw=6000//n-m, initial brakin torque
7 Tf=300//n-m, fictional torque
8 j=540//kg-m2
9 tr=(j/k)*log((kw+Tf)/Tf)
10 //disp(tr)
11 s=%e^((-k*tr)/j)
12 //disp(s)
13 temp=((j/k)*(kw+Tf)*(1-s))-((Tf*tr))
14 Nr=(1/(2*pi*k))*temp
15 //disp(Nr)
16 printf("Time taken for rheostatic braking is %f s",
        Nr)
17 //part b
18 beta=3600/j
19 motor_rest_time=w/6.67
20 //disp(motor_rest_time)
21 rev=(1000/60)*.5*(motor_rest_time)
22 printf("Number of revolutions made is %f",rev)

```

Chapter 7

Rating and Heating Motors

Scilab code Exa 7.1 Final tepearture

```
1 //Example 7.1, page 147
2 clc
3 G=500 //in kg
4 h=700 // in j/kg/c
5 s_lambda=%pi*.7*1*12.5
6 h=(G*h)/s_lambda
7 //disp(h)
8 L=((10*1000)/.9)-10000
9 w=1111
10 T=w/s_lambda
11 printf("Final tepearture rise is %f k",T)
```

Scilab code Exa 7.3 power

```
1 //Example 7.3, page 162
2 clc
3 alpha=0.9
4 N=.5 //in hr
```

```

5 tou=1.5//in hr
6 pr=25//in kw
7 px=pr*sqrt(((1+alpha)/(1-exp(-N/tou)))-.9)
8 printf("Power id %f kw",px)

```

Scilab code Exa 7.4 Power

```

1 //example 7.4, page 165
2 clc
3 pr=100//in kw
4 N=18//in min
5 tou=90//in min
6 R=30//in min
7 tou_1=120//in min
8 a=N/tou
9 b=R/tou_1
10 px=pr*sqrt((1-exp(-(a+b)))/(1-exp(-(a))))
11 printf("The power is %f kw",px)

```

Scilab code Exa 7.5 Weight of fly wheel

```

1 //Example , page 162//169
2 clc
3 Tr=(50*1000*60)/(2*pi*960)
4 //disp(Tr)
5 Tmax=2*Tr
6 Tmin=300//in nm
7 Tlh=1500+Tmin
8 t=10//in sec
9 ws=(2*pi*1000)/60//angular f
10 wr=(2*pi*960)/60//angular f
11 temp=log((Tlh-Tmin)/(Tlh-Tmax))
12 j=(Tr/(ws-wr))*(t/temp)

```

```
13 //disp(j)
14 r=.9//in m
15 wt=j/r^2
16 //disp(wt)
17 printf("Weight of fly wheel is %f kg",wt)
18 //Part b
19 Tmax=994.72
20 Tmin=700
21 TL1=300
22 temp=log((Tmax-TL1)/(Tmin-TL1))
23 t=(j*temp*(ws-wr))/Tr
24 printf("\n The estimated time is %f s",t)
```
