Scilab Textbook Companion for
A First Course on Electrical Drives
by S. K. Pillai[1]

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

February 28, 2015

[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: 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

Chapter 4

Characteristics of AC motor

Scilab code Exa 4.1 Roots of the equation

```
//Example 4.1, Page 72
clc;
p1=poly([1, -3, 1], 'x', 'c')//Polynomial equation
disp('Part a')
disp('roots of the equation when slip at max torque'
  )
x=roots(p1)
disp(x)


p2=poly([0.146, -1.719, 1], 'x', 'c')//polynomial
  equation in scilab function
y=roots(p2)
disp('Part b')
disp('roots of the equation when slip at max load')
disp(y)
```

Scilab code Exa 4.3 Steady operation

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

Scilab code Exa 4.5 Value of resistance

```
//Example 4.5, page 81
clc
pole=24
Ns=245//in rpm
N=(120*50)/pole//synchronous speed in rpm
f=(N-Ns)/N
p=110//in kw
T=(p*1000*60)/(2*%pi*Ns)
v1=440/sqrt(3)//in v
ws=(2*%pi*250)/60
s=0.02
R=0.03125//in ohm
x=sqrt(((3*R*v1**2)/(T*ws*s))-(R/s)**2)//by
    rearranging formula
printf("\n Stator resistance per phase is %f ohm ",x
    )
//calculating original resistance

//Example 4.1, Page 72
clc;
p1=poly([72.78, -3235, 3190], 'x', 'c')//Polynomial
    equation
disp('Part a')
disp('The value of original resistance is')
```

```
x=roots(p1)
disp(x)

//Taking r=0.99108
r=(0.99108-R)/1.25**2

printf("\n The value of resistance to be added is
    %f ohm ",r)
```

Scilab code Exa 4.6 Value of Delta

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

Chapter 5

Starting

Scilab code Exa 5.1 Value of Torque

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

Scilab code Exa 5.3 Torque

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

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

Scilab code Exa 5.4 Permissible start

```
//Example 5.4, Page no 110
clc
p=400*20*.88*sqrt(3)//input power in watt
l=12193.6-10000//in watt, full load loss
e=l*60//energy lost per minute
ws=(2*%pi*1000)/60//angular frequency
j=.5
R=1//resistance in ohm ir R1/R2
El=(0.5*j*ws**2)*(1+R)
//disp(El)
N=e/El
//disp(N)
printf("\n The number of starts that can be made is %d ",N)
ws=(2*%pi*1500)/60//angular frequency
j=.5
R=1//resistance in ohm ir R1/R2
El=(0.5*j*ws**2)*(1+R)
//disp(El)
N=e/El
//disp(N)
```

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

Chapter 6

Electric Braking

Scilab code Exa 6.1 Speed of shunt machine

```
//Example 6.1, page 121
clc
T=172//in N-m
w=(2*%pi*960)/60
E=215//in V
Ia=(T*w)/E
Ra=.062//in ohm
v=220//in v
Eg=v+(Ia*Ra)
//disp(Eg)
N=960//in Rpm
S=(N*Eg)/E
//disp(S)
printf("\n The speed of shunt machine is %f rpm ",S)
```

Scilab code Exa 6.2 Resistance

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

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

Scilab code Exa 6.3 Braking Torque and current

```
//Example 6.3, Page 123
clc
per=.88
v=220//in v
p=20//in kw
I=(p*1000)/(per*v)
T=(p*1000*60)/(2*%pi*1200)
//Part a
E_motor=v-(I*.1)
v_arm=v+E_motor
Ir=2*I//Rated current
R=v_arm/Ir
disp(R)
R_Extra=R-.1
printf("\n Extra resistance added to motor armature
    is %f ohm ",R_Extra)
//Part b
T_Full_load=T*2
printf("\n Full load torque is %f N-m ",T_Full_load)
//Part c
E=(E_motor*400)/1200
```

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

Scilab code Exa 6.4 Braking Torque

```
//Example 6.4, page 131
clc
R1=.15//in ohm
Rs=.45//in ohm
x1=.6//in ohm
xz=1.8//in ohm
sf=.05
Turn=1/sqrt(3)
R_rotor=Rs*Turn//in ohm
X_rotor=xz*Turn^2//in ohm

//Part 1
//BY FIGURE
E1=sqrt((3^2+.6^2)/(3.15^2+1.2^2))*440/sqrt(3)
//disp(E1)
s=1-sf
I2=E1/sqrt(x1**2+2**2)
//disp(I2)
R2=2*60//ohm
w=2*%pi*600
T=(R2*3*(I2^2))/(s*w)
//disp(T)
printf("Initial braking torque of rheostatic is %f N
   -m",T)

//Part 2
s1=2-sf
```

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

Scilab code Exa 6.5 Initial braking torque

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

Scilab code Exa 6.6 number of starts and stops

```
//Example 6.6, Page 138
clc
```

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

Scilab code Exa 6.7 Minimum time to bring rotor to rest

```
//Example 6.7, page 138
clc;
v1=400/sqrt(3)//in v
ws=(2*%pi*1000)/60//angular f
x=1//resistance in ohm
T_max=(3*v1^2)/(ws*2*x)
//disp(T_max)
j=10//in kg-m2
s1=.05
s_maxT=0.2
a=(1.95^2-1)/(2*s_maxT)
temp=a+(.2*log(1.95))
r=((10*ws)/(2*T_max))*(temp)
//disp(r)
Extra_R=r-(.2)
a=(1.95^2-1)/(2*1.45)
temp=a+(1.45*log(1.95))
```

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

Scilab code Exa 6.8 Number of revolutions made

```
//Example 6.8, page 141
clc
//part a
w=(2*%pi*50)/3//angular f, rad/sec
k=6000/w
kw=6000//n-m, initial brakin torque
Tf=300//n-m, fictional torque
j=540//kg-m2
tr=(j/k)*log((kw+Tf)/Tf)
//disp(tr)
s=%e^((-k*tr)/j)
//disp(s)
temp=((j/k)*(kw+Tf)*(1-s))-((Tf*tr))
Nr=(1/(2*%pi*k))*temp
//disp(Nr)
printf("Time taken for rheostatic braking is %f s",
    Nr)
//part b
beta=3600/j
motor_rest_time=w/6.67
//disp(motor_rest_time)
rev=(1000/60)*.5*(motor_rest_time)
printf("Number of revolutions made is %f",rev)
```

Chapter 7

Rating and Heating Motors

Scilab code Exa 7.1 Final tempearture

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

Scilab code Exa 7.3 power

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

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

Scilab code Exa 7.4 Power

```
//example 7.4, page 165
clc
pr=100//in kw
N=18//in min
tou=90//in min
R=30//in min
tou_1=120//in min
a=N/tou
b=R/tou_1
px=pr*sqrt((1-exp(-(a+b)))/(1-exp(-(a))))
printf("The power is %f kw",px)
```

Scilab code Exa 7.5 Weight of fly wheel

```
//Example , page 162//169
clc
Tr=(50*1000*60)/(2*%pi*960)
//disp(Tr)
Tmax=2*Tr
Tmin=300//in nm
Tlh=1500+Tmin
t=10//in sec
ws=(2*%pi*1000)/60//angular f
wr=(2*%pi*960)/60//angular f
temp=log((Tlh-Tmin)/(Tlh-Tmax))
j=(Tr/(ws-wr))*(t/temp)
```

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