# Scilab Textbook Companion for Theory Of Machines by R. S. Khurmi And J. K. Gupta[1]

Created by
Shreerang Deshpande
Bachelor of Engineering
Mechanical Engineering
B.V.Bhoomaraddi College of Engg. & Tech.
College Teacher
P.g.tewari
Cross-Checked by

August 14, 2013

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

# Book Description

**Title:** Theory Of Machines

**Author:** R. S. Khurmi And J. K. Gupta

**Publisher:** Eurasia Publishing House (Pvt.) Ltd., New Delhi

**Edition:** 14

**Year:** 2009

**ISBN:** 81-219-2524-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 2

# Kinematics of Motion

**Scilab code Exa 2.1** To Find the Acceleration and Distance

```
//To Find the Acceleration and Distance
clc
//Given:
u1=0,v1=72*1000/3600 //m/s
s1=500 //m
//Solution:
//Calculating the initial acceleration of the car
a1=(v1^2-u1^2)/(2*s1) //m/s^2
//Calculating time taken by the car to attain the
   speed
t1=(v1-u1)/a1 //seconds
//Parameters for the second case
u2=v1,v2=90*1000/3600 //m/s
t2=10 //seconds
//Calculating the acceleration for the second case
a2=(v2-u2)/t2 //m/s^2
//Calculating the distance moved by the car in the
   second case
s2=(u2*t2)+(a2/2*t2^2)
//Parameters for the third case
u3=v2,v3=0 //m/s
```

```
t3=5 //seconds
//Calculating the distance moved by the car
s3=(u3+v3)*t3/2 //m
//Results:
printf("\n\n The acceleration of the car, a = %.1f m
    /s^2. \n",a1)
printf(" The car takes t = %d s to attain the speed
    .\n",t1)
printf(" The acceleration of the car in the second
    case, a = %.1f m/s^2.",a2)
printf("\n The distance moved by the car, s = %d m.\
    n",s2)
printf(" The distance travelled by the car during
    braking, s = %.1f m.\n\n",s3)
```

**Scilab code Exa 2.3** To Find the Velocity

```
//To Find the Velocity
clc
//Given:
//Initial parameters
v0=100 //kmph
t0=0
//Parameters at the end of 40 seconds
v1=90/100*v0 //kmph
t1=40 //seconds
//Solution:
//The acceleration is given by, a=(-dv/dt)=k*v
//Integrating, we get ln(v)=-k*t+C
//Calculating the constant of integration
C=integrate('1/v','v',1,100)
//Calculating the constant of proportionality
k=(C-2.3*log10(90))/40
//Time after 120 seconds
t2=120 //seconds
```

```
//Calculating the velocity after 120 seconds
v120=10^((-k*t2+C)/2.29)
//Results:
printf("\n\n The velocity at the end of 120 seconds, v120 = %.1f kmph.\n\n",v120)
```

**Scilab code Exa 2.5** To Find the Maximum Cutting Speed

```
//To Find the Maximum Cutting Speed
clc
//Given:
s=500,s1=125,s2=250,s3=125 //mm
t=1 //second
//Solution:
//Matrices for the velocity vs. time graph
V=[0,750,750,0] //The velocity matrix
T=[0,1/3,2/3,1] //The time matrix
plot2d(T,V)
//Calculating the time of uniform acceleration
t1=rdivf('s1','v/2')
//Calculating the time of constant speed
t2=rdivf('s2','v')
//Calculating the time of uniform retardation
t3=rdivf('s3','v/2')
//Equating the time taken to complete the stroke to 1 second
v=(125/(1/2)+250/1+125/(1/2))/1 //mm/s
//Results:
printf("\n\n The maximum cutting speed, v = %d mm/s.\n\n",v)
```

**Scilab code Exa 2.6** To Find the Angular Acceleration

```
//To Find the Angular Acceleration
clc
//Given:
N0=0,N=2000 //rpm
t=20 //seconds
//Solution:
//Calculating the angular velocities
omega0=0, omega=2*%pi*N/60 //rad/s
//Calculating the angular acceleration
alpha=(omega-omega0)/t //rad/s^2
//Calculating the angular distance moved by the
   wheel during 2000 rpm
theta=(omega0+omega)*t/2 //rad
//Calculating the number of revolutions made by the
   wheel
n=theta/(2*%pi)
//Results:
printf("\n\n The angular acceleration of the wheel,
   alpha = %.3f rad/s^2.\n",alpha)
printf(" The wheel makes n = %.1f revolutions.\n\n",
   n)
```

**Scilab code Exa 2.7** To Find Velocity and Acceleration

```
//To Find Velocity and Acceleration
clc
//Given:
r=1.5 //m
N0=1200,N=1500 //rpm
t=5 //seconds
//Solution:
//Calculating the angular velocities
omega0=2*%pi*N0/60,omega=2*%pi*N/60 //rad/s
//Calculating the linear velocity at the beginning
v0=r*omega0 //m/s
```

```
//Calculating the linear velocity at the end of 5 seconds
v5=r*omega //m/s
//Calculating the angular acceleration
alpha=(omega-omega0)/t //ad/s^2
//Calculating the tangential acceleration after 5 seconds
TangentialAcceleration=alpha*(r/2) //m/s^2
//Calculating the radial acceleration after 5 seconds
RadialAcceleration=(omega^2)*(r/2) //m/s^2
//Results:
printf("\n\n The linear velocity at the beginning, v0 = %.1f m/s.\n",v0)
printf(" The linear velocity after 5 seconds, v5 = %.1f m/s.\n",v5)
printf(" The tangential acceleration after 5 seconds is %.1f m/s^2.\n",TangentialAcceleration)
printf(" The radial acceleration after 5 seconds is %d m/s^2.",RadialAcceleration)
```

# Chapter 3

# Kinetics of Motion

**Scilab code Exa 3.1** To find the angular acceleration and KE

```
//To find the angular acceleration and KE
clc
//Given:
k=1 //m
m=2500 //kg
T=1500 //N-m
//Solution:
//Calculating the mass moment of inertia of the flywheel
I=m*k^2 //kg-m^2
//Calculating the angular acceleration of the flywheel
alpha=T/I //rad/s^2
//The angular speed at start
omega1=0
t=10 //seconds
//Calculating the angular speed of the flywheel after t=10 seconds from start
omega2=omega1+alpha*t //rad/s
//Calculating the kinetic energy of the flywheel after 10 seconds from start
```

```
E=1/2*I*omega2^2/1000 //kJ
//Results:
printf("\n\n The angular acceleration of the
    flywheel, alpha = %.1f rad/s^2.\n",alpha)
printf(" The kinetic energy of the flywheel, E = %d
    kJ.\n\n",E)
```

**Scilab code Exa 3.2** To find the time torque and power

```
//To find the time, torque and power
clc
//Given:
mC=500,mD=250 //kg
s=100,r=0.5,k=0.35 //m
m=3 //kg/m
//Solution:
//Velocities of the cage
u1=0,v1=10,v2=10,u3=10,v3=0 //m/s
//Accelerations of the cage
a1=1.5,a3=-6 //m/s^2
s=100 //m
//Calculating the time taken by the cage to reach
    the top
t1=(v1-u1)/a1 //seconds
//Calculating the distance moved by the cage during
    time t1
s1=(v1+u1)/2*t1 //m
//Calculating the time taken for the cage from
    initial velocity u3=10 m/s to final velocity of
    v3=0
t3=(v3-u3)/a3 //seconds
//Calculating the distance moved by the cage during
    time t3
s3=(v3+u3)/2*t3 //m
//Calculating the distance travelled during constant
```

```
velocity of v2=10 m/s
s2=s-s1-s3 //m
//Calculating the time taken for the cage during constant velocity
t2=s2/v2 //seconds
//Calculating the time taken for the cage to reach the top
t=t1+t2+t3 //seconds
//Calculating the total mass of the rope for 100 metres
mR=m*s //kg
//Calculating the force to raise the cage and rope at uniform speed
F1=(mC+mR)*9.81 //N
//Calculating the torque to raise the cage and rope at uniform speed
T1=F1*r //N-m
//Calculating the force to accelerate the cage and rope
F2=(mC+mR)*a1 //N
//Calculating the torque to accelerate the cage and rope
T2=F2*r //N-m
//Calculating the mass moment of inertia of the drum
I=mD*k^2 //kg-m^2
//Calculating the angular acceleration of the drum
alpha=a1/r //rad/s^2
//Calculating the torque to accelerate the drum
T3=I*alpha //N-m
//Calculating the total torque which must be applied to the drum at starting
T=T1+T2+T3 //N-m
//Calculating the mass of 33.35 m rope
m1=m*33.35 //kg
//Calculating the reduction of torque
T4=(m1*9.81+m1*a1)*r //N-m
//Calculating the angular velocity of drum
omega=v2/(2*%pi*r) //rad/s
```

```
//Calculating the power
P=T4*omega/1000 //Power, kW
//Results:
printf("\n\n The time taken for the cage to reach the top, t = %.2f s.\n",t)
printf(" The total torque which must be applied to the drum during starting, T = %.1f N-m.\n",T)
printf(" The power required is %.3f kW.\n\n",P)
```

**Scilab code Exa 3.3** To find the reduction of speed

```
//To find the reduction of speed
clc
//Given:
P=4*1000 //W
I=140 //kg-m^2
N1=240 //rpm
//Solution:
//Calculating the angular acceleration at the commencement of operation
omega1=2*%pi*N1/60 //rad/s
//Calculating the energy supplied by the motor (E1) and the energy consumed in closing a revet in 1 second
E1=4000,E2=10000 //N-m
//Calculating the loss of kinetic energy of the flywheel during the operation
E=E2-E1 //N-m
//Calculating the kinetic energy of the flywheel at the commencement of operation
KEc=1/2*I*omega1^2 //Kinetic energy at the commencement, N-m
//Calculating the kinetic energy of the flywheel at the end of operation
KEe=KEc-E //Kinetic energy at the end, N-m
```

```
//Calculating the angular speed of the flywheel
   immediately after closing a revet
omega2=sqrt(KEe*2/I) //rad/s
//Calculating the reduction of speed
ReductionofSpeed=(omega1-omega2)*60/(2*%pi) //rpm
//Calculating the maximum rate at which the revets
   can be closed per minute
Rate=P*60/E2 //Maximum rate at which the revets can
   be closed per minute
//Results:
printf("\n\n The reduction of speed is %.1f rpm.\n",
   ReductionofSpeed)
printf(" The maximum rate at which rivets can be
   closed per minute is %d.\n\n",Rate)
```

**Scilab code Exa 3.4** To find the torque and power

```
//To find the torque and power
clc
//Given:
m=14*1000,m1=1.25*1000,m2=110 //kg
d=1,r=d/2,k1=450/1000,k2=125/1000 //m
F=1.2*1000 //N
eta=0.85
v=1.8 //m/s
a=0.1 //m/s^2
//Solution:
//Calculating the forces opposing the motion
P1=m*9.81*1/20+m*a+F //N
//Calculating the torque on the drum shaft to
   accelerate the load
T1=P1*r //N-m
//Calculating the mass moment of inertia of the drum
I1=m1*k1^2 //kg-m^2
//Calculating the angular acceleration of the drum
```

```
alpha1=a/r //rad/s
//Calculating the torque on the drum to accelerate
   the drum shaft
T2=I1*alpha1 //N-m
//Calculating the torque on the armature to
   accelerate drum and load
T3=(T1+T2)/(40*eta) //N-m
//Calculating the mass moment of inertia of the
   armature
I2=m2*k2^2 //kg-m^2
//Calculating the angular acceleration of the
   armature
alpha2=a/r*40 //rad/s^2
//Calculating the torque on the armature to
   accelerate armature shaft
T4=I2*alpha2 //N-m
//Calculating the torque on the motor shaft
T=T3+T4 //N-m
//Calculating the angular speed of the motor
omega=v/r*40 //rad/s
//Calculating the power developed by the motor
P=T*omega/1000 //Power developed by the motor, kW
//Results:
printf("\n\n The torque on the motor shaft, T = %.2f
    N-m.\n",T)
printf(" The power developed by the motor is %.2f kW
   .\n\n",P)
```

**Scilab code Exa 3.5** To find the KE and braking force

```
//To find the KE and braking force
clc
//Given:
m=12*1000,m1=2*1000,m2=2.5*1000 //kg
k1=0.4,d1=1.2,r1=d1/2,k2=0.6,d2=1.5,r2=d2/2,s=6 //m
```

```
v=9*1000/3600 //m/s
//Solution:
//Calculating the mass moment of inertia of the front roller
I1=m1*k1^2 //kg-m^2
//Calculating the mass moment of inertia of the rear axle together with its wheels
I2=m2*k2^2 //kg-m^2
//Calculating the angular speed of the front roller
omega1=v/r1 //rad/s
//Calculating the angular speed of rear wheels
omega2=v/r2 //rad/s
//Calculating the kinetic energy of rotation of the front roller
E1=1/2*I1*omega1^2 //N-m
//Calculating the kinetic energy of rotation of the rear axle with its wheels
E2=1/2*I2*omega2^2 //N-m
//Calculating the total kinetic energy of rotation of the wheels
E=E1+E2 //N-m
//Calculating the kinetic energy of translation of the road roller
E3=1/2*m*v^2 //N-m
//Calculating the total kinetic energy of the road roller
E4=E3+E //N-m
//Calculating the braking force to bring the roller to rest
F=E4/s //N
//Results:
printf("\n\n The total kinetic energy of rotation of the wheels, E = %d N-m.\n",E)
printf(" The total kinetic energy of the road roller, E4 = %d N-m.\n",E4)
printf(" The braking force required to bring the roller to rest, F = %.1f N.\n\n",F)
```

**Scilab code Exa 3.7** To Find the Speed and Energy Lost

```
//To Find the Speed and Energy Lost
clc
//Given:
r=500/1000,k=450/1000 //m
m1=500,m2=1250 //kg
u=0.75 //m/s
//Solution:
//Calculating the mass moment of inertia of drum
I2=m2*k^2 //kg-m^2
//Calculating the speed of truck
//Impulse, F=m1*v or, F-m1*v=0                    .....(i)
//Moment of impulse, F*r=I2*(omega2-omega2) or, F*r=I2*(u-v)/r or, F*r+I2*v/r=I2*u/r     .....(ii)
//Solving (i) and (ii)
A=[1 -m1; r I2/r]
B=[0; I2*u/r]
V=A \ B
v=V(2)
//Calculating the energy lost to the system
E=1/2*I2*(u^2-v^2)/r^2-1/2*m1*v^2 //Energy lost to the system, N-m
//Results:
printf("\n\n The speed of the truck when the motion becomes steady, v = %.3f m/s.\n",v)
printf(" The energy lost to the system is %d N-m.\n\n",E)
```

**Scilab code Exa 3.8** To find the velocity KE and compression

```
//To find the velocity, KE and compression
clc
//Given:
s=0.7*10^6 //N/m
m1=10*10^3,m2=15*10^3 //kg
v1=1.8,v2=0.6 //m/s
//Solution:
//Calculating the common velocity when moving together during impact
v=(m1*v1+m2*v2)/(m1+m2)
//Calculating the kinetic energy lost to the system
E=(1/2*m1*v1^2+1/2*m2*v2^2)-1/2*(m1+m2)*v^2
//Calculating the compression of each buffer spring
x=sqrt(E/(2*s))
//Calculating the velocity of each truck on separation
//Final KE after separation=KE at common velocity+Half of energy stored in springs.
//And initial and final momentum must be equal.
//Simplifying the two equations, we get,
//1/2*m1*v3^2+1/2*m2*v4^2=1/2*(m1+m2)*v^2+1/2*E .....(i)
//m1*v3+m2*v4=(m1+m2)*v
function y=f(x)
    v3=x(1)
    v4=x(2)
    y(1)=1/2*m1*v3^2+1/2*m2*v4^2-1/2*(m1+m2)*v^2-1/2*E
    y(2)=m1*v3+m2*v4-(m1+m2)*v
endfunction
z=fsolve([1,1],f)
v3=z(1)
v4=z(2)
//Results:
printf("\n\n The common velocity when moving together during impact, v = %.2f m/s.\n",v)
printf(" The kinetic energy lost to the system is %.2f kN-m.\n",E/1000)
```

```
printf(" The compression of each buffer spring, x = %d mm.\n",x*1000)
printf(" The velocity of separation for 10 tonnes truck, v3 = %.1f m/s.\n",v3)
printf(" The velocity of separation for 15 tonnes truck, v4 = %.1f m/s.\n",v4)
```

**Scilab code Exa 3.9** To find energy lost and resistance

```
//To find energy lost and resistance
clc
//Given:
m1=300,m2=500 //kg
s=1,x=150/1000 //m
//Solution:
//Calculating the velocity with which mass m1 hits the pile
u=0
v1=sqrt(2*9.81*s+u^2) //m/s
//Calculating the common velocity after impact
v2=0
v=(m1*v1+m2*v2)/(m1+m2) //m/s
//Calculating the kinetic energy before impact
KEb=m1*9.81*s //Kinetic energy before impact, N-m
//Calculating the kinetic energy after impact
KEa=1/2*(m1+m2)*v^2 //Kinetic energy after impact, N-m
//Calculating the energy lost in the blow
E=KEb-KEa //Energy lost in the blow, N-m
//Calculating the average resistance against the pile
R=KEa/x+m1*9.81+m2*9.81
//Results:
printf("\n\n The energy lost in the blow is %d N-m.\n",E)
```

```
printf(" The average resistance against the pile, R = %.3f kN.\n\n",R/1000)
```

**Scilab code Exa 3.10** To find the angular velocities

```
//To find the angular velocities
clc
//Given:
m1=0.7,m2=2.4 //kg
k1=270/1000,k2=185/1000,h1=0.25,DL=0.2,CM=0.275 //m
//Solution:
//Calculating the angular velocity of hammer just before impact
h=h1*(1-cos(20*%pi/180))
omega=sqrt(m1*9.81*h*2/(m1*k1^2)) //rad/s
//Calculating the relative linear velocity
RLV=0.8*omega*CM
//Calculating the values of angular velocities
//The two equations we get in terms of omegaA and omegaB are
//DL*omegaA-CM*omegaB=RLV                                        .....(i)
//m1*k1^2*(omega-omegaB)=.275/.2*m2*k2^2*omegaA, or
//2.21*omegaA+omegaB=2.01                                        .....(ii)
A=[DL -CM; 2.21 1]
B=[RLV; 2.01]
V=A \ B
//Results:
printf("\n\n The angular velocity of the anvil A, omegaA = %.2f rad/s.\n",V(1))
printf(" The angular velocity of the hammer B, omegaB = %.2f rad/s, i.e. %.2f rad/s, in the
```

```
reverse direction.\n\n",V(2),V(2)*-1)
```

**Scilab code Exa 3.11** To find the velocity impulse angle of swing and average force

```
//To find the velocity, impulse, angle of swing and
    average force
clc
//Given:
m=30 //kg
AG=1,GB=150/1000,k1=1.1,k2=350/1000 //m
theta=60*%pi/180 //rad
t=0.005 //s
a=AG,b=GB
//Solution:
//Calculating the mass moment of inertia of the
    pendulum about the point of suspension A
IA=m*k1^2 //kg-m^2
//Calculating the mass moment of inertia ofthe
    pendulum about centre of gravity G
IG=m*k2^2 //kg-m^2
//Calculating the angular velocity of the pendulum
h1=a-a*cos(theta)
omega=sqrt(2*m*9.81*h1/IA) //rad/s
//Calculating the striking velocity of the pendulum
v=omega*(a+b) //m/s
//Calculating the angular velocity of the pendulum
    just after the breakage of the specimen
omega1=sqrt(omega^2-2*54/IA)
//Calculating the linear velocity of G just before
    the breakage of specimen
vG=omega*AG //m/s
//Calculating the linear velocity of G just after
    the breakage of specimen
vGdash=omega1*AG //m/s
```

```
//Calculating the impulses at pivot A and knife edge B
//F1+F2=m*(vG-vGdash)                                    .....(i)
//b*F2-a*F1=IG*(omega-omega1)                        .....(ii)
A=[1 1; -a b]
B=[m*(vG-vGdash); IG*(omega-omega1)]
V=A \ B
F1=V(1),F2=V(2)
//Calculating the angle of swing of the pendulum after impact
theta1=acos(a-1/2*IA*omega1^2/(m*9.81))/a //radians
//Calculating the average force exerted at the pivot
Fp=F1/t //N
//Calculating the average force exerted at the knife edge
Fk=F2/t //N
//Results:
printf("\n\n The striking velocity of the pendulum, v = %.2f m/s.\n",v)
printf(" Impulse at the pivot A, F1 = %.1f N.\n",F1)
printf(" Impulse at the knife edge B, F2 = %.1f N.\n",F2)
printf(" Angle of swing of the pendulum after impact, theta = %.2f degree.\n",theta1*180/%pi)
printf(" Average force exerted at the pivot is %d N.\n",Fp)
printf(" Average force exerted at the knife edge is %d N.\n\n",Fk)
```

**Scilab code Exa 3.12** To find the speed time and KE lost

```
//To find the speed, time and KE lost
clc
```

```
//Given:
T=150 //N-m
m1=60,m2=20 //kg
k1=140/1000,k2=80/1000 //m
N1=750,N2=0 //rpm
//Sloution:
//Calculating the angular speeds
omega1=2*%pi*N1/60,omega2=0 //rad/s
//Calculating the mass moment of inertia of the rotor on motor
I1=m1*k1^2 //kg-m^2
//Calculating the mass moment of inertia of the parts attached to machine
I2=m2*k2^2 //kg-m^2
//Calculating the speed after engagement of the clutch and the time taken
//We know that impulsive torque = change in angular momentum
//T*t = I1*(omega1-omega), or I1*omega+T*t = I1*omega1                    .....(i)
//T*t = I2*(omega-omega2), or I2*omega-T*t = I2*omega2                    .....(ii)
A=[I1 T; I2 -T]
B=[I1*omega1; I2*omega2]
V=A \ B
omega=V(1) //rad/s
t=V(2) //s
//Calculating the kinetic energy lost during the operation
E=I1*I2*(omega1-omega2)^2/(2*(I1+I2)) //N-m
//Results:
printf("\n\n The speed after engagement, omega = %.1f rad/s.\n",omega)
printf(" The time taken, t = %.2f s.\n",t)
printf(" The kinetic energy lost during the operation, E = %d N-m.\n\n",E)
```

**Scilab code Exa 3.13** To find the acceleration

```
//To find the acceleration
clc
//Given:
M=75 //kg
r=0.3 //m
G=6
IA=100,IB=5 //kg-m^2
eta=90/100 //%
//Solution:
//Calculating the equivalent mass of the geared system
me=1/r^2*(IA+G^2*IB) //kg
//Calculating the total equivalent mass to be accelerated
Me=me+M //kg
//Calculating the acceleration when it is allowed to fall freely
F=M*9.81 //Accelerating force provided by the pull of gravity, N
a=F/Me //m/s^2
//Calculating the equivalent mass of the geared system when the efficiency is 90%
me1=1/r^2*(IA+G^2*IB/eta) //kg
//Calculating the total equivalent mass to be accelerated
Me1=me1+M //kg
//Calculating the acceleration when the efficiency is 90%
F1=M*9.81 //Accelerating force by the pull of gravity, N
a1=F1/Me1 //m/s^2
//Results:
```

```
printf("\n\n The acceleration of the mass M if it is
    allowed to fall freely , a = %.3f m/s^2.\n",a)
printf(" The acceleration of the mass M when the
    efficiency of the gearing system is 0.9, a = %.3f
    m/s^2.\n\n",a1)
```

**Scilab code Exa 3.18** To find the torque

```
//To find the torque
clc
//Given:
d=1.5,r=d/2,d1=1,kM=90/1000,kI=225/1000,kD=600/1000,
   kP=450/1000 //m
NM=900,N1=275,ND=50 //rpm
mM=200,mI=375,mD=2250,mP=200,m1=1150,m2=650 //kg
FI=150, FD=1125, FP=150 //N-m
F1=500,F2=350 //N
a=0.9 //m/s^2
//Solution:
//Calculating the speed of guide pulley
NP=ND*d/d1 //rpm
//Calculating the gear ratio for intermediate gear
   and motor
G1=N1/NM
//Calculating the gear ratio for drum and motor
G2=ND/NM
//Calculating the gear ratio for the guide pulley
   and motor
G3=NP/NM
//Calculating the mass moment of inertia of the
   motor
IM=mM*kM^2 //kg-m^2
//Calculating the mass moment of inertia of the
   intermediate gear
II=mI*kI^2 //kg-m^2
```

```
//Calculating the mass moment of inertia of the drum and shaft
ID=mD*kD^2 //kg-m^2
//Calculating the mass moment of inertia of the guide pulley
IP=mP*kP^2 //kg-m^2
//Calculating the angular acceleration of the drum
alphaD=a/r //rad/s^2
//Calculating the angular acceleration of the motor
alphaM=alphaD*NM/ND //rad/s^2
//Calculating the equivalent mass moment of inertia of the system
I=IM+G1^2*II+G2^2*ID+2*G3^2*IP //kg-m^2
//Calculating the torque at motor to accelerate the system
T1=I*alphaM //N-m
//Calculating the torque at motor to overcome friction at intermediate gear, drum and two guide pulleys
T2=G1*FI+G2*FD+2*G3*FP //N-m
//Calculating the tension in the rising rope between the pulley and drum
Q1=m1*9.81+m1*a+F1 //N
//Calculating the tension in the falling rope between the pulley and drum
Q2=m2*9.81-m2*a-F2 //N
//Calculating the torque at drum
TD=(Q1-Q2)*r //N-m
//Calculating the torque at motor to raise and lower cages and ropes and to overcome frictional resistance
T3=G2*TD //N-m
//Calculating the total motor torque required
T=T1+T2+T3 //N-m
//Results:
printf("\n\n The total motor torque required, T = %.1f N-m.\n\n",T)
```

**Scilab code Exa 3.19** To find velocities and loss of KE

```
//To find velocities and loss of KE
clc
//Given:
m1=50,m2=25 //kg
u1=3,u2=1.5 //m/s
//Solution:
//When the impact is inelastic
//Calculating the common velocity after impact
v=(m1*u1+m2*u2)/(m1+m2) //m/s
//Calculating the loss of kinetic energy during
    impact
EL=m1*m2/(2*(m1+m2))*(u1-u2)^2 //N-m
//When the impact is elastic
//Calculating the velocity of the first sphere
    immediately after impact
v1=2*v-u1 //m/s
//Calculating the velocity of the second sphere
    immediately after impact
v2=2*v-u2 //m/s
//Calculating the loss of kinetic energy
EL1=0
//When the coefficient of restitution, e=0.6
e=0.6
//Calculating the velocity of the first sphere
    immediately after impact
v12=(1+e)*v-e*u1 //m/s
//Calculating the velocity of the second sphere
    immediately after impact
v22=(1+e)*v-e*u2 //m/s
//Calculating the loss of kinetic energy
EL2=m1*m2/(2*(m1+m2))*(u1-u2)^2*(1-e^2) //N-m
//Results:
```

```
printf("\n\n The common velocity after impact when the impact is inelastic, v = %.1f m/s.\n",v)
printf(" The loss of kinetic energy during impact, EL = %.2f N-m.\n",EL)
printf(" The velocity of the first sphere immediately after impact when the impact is elastic, v1 = %d m/s.\n",v1)
printf(" The velocity of the second sphere immediately after impact, v2 = %.1f m/s.\n",v2)
printf(" The loss of kinetic energy, EL = %d.\n",EL1)
printf(" The velocity of the first sphere immediately after impact When the coefficient of restitution is 0.6, v1 = %.1f m/s.\n",v12)
printf(" The velocity of the second sphere immediately after impact, v2 = %.1f m/s.\n",v22)
printf(" The loss of kinetic energy during impactm EL = %d N-m.\n\n",EL2)
```

---

**Scilab code Exa 3.20** To find the speed and energy dissipated

```
//To find the speed and energy dissipated
clc
//Given:
m1=15*1000,m2=5*1000 //kg
u1=20*1000/3600,u2=12*1000/3600 //m/s
s=1000*10^3 //N/m
e=0.5
//Solution:
//Calculating the common speed
v=(m1*u1+m2*u2)/(m1+m2) //m/s
//Calculating the strain energy stored in one spring
SE=mulf('1/2*s','x^2') //Strain energy, N-m
//Calculating the strain energy stored in four buffer springs
```

```
SE4=mulf('4*1/2*s','x^2') //Strain energy, N-m
//Calculating the difference in kinetic energies
    before impact and during impact
d=m1*m2/(2*(m1+m2))*(u1-u2)^2 //Difference in
    kinetic energies, N-m
//Equating the difference between kinetic energies
    to the strain energy stored in the springs
x=sqrt(d*2/(4*s))*1000 //mm
//Calculating the speed of the loaded wagon
    immediately after impact ends
v11=2*v-u1 //m/s
//Calculating the speed of the empty wagon
    immediately after impact ends
v21=2*v-u2 //m/s
//Calculating the speeds of the wagons taking into
    account the coefficient of restitution, e=0.5
v12=(1+e)*v-e*u1 //m/s
v22=(1+e)*v-e*u2 //m/s
//Calculating the amount of energy dissipated during
    impact
EL=m1*m2/(2*(m1+m2))*(u1-u2)^2*(1-e^2) //N-m
//Results:
printf("\n\n The magnitude of common speed, v = %d m
    /s.\n",v)
printf(" The maximum deflection of each buffer
    spring during impact, x = %d mm.\n",x)
printf(" The speed of the loaded wagon immediately
    after the impact ends, v1 = %.2f m/s.\n",v11)
printf(" The speed of the empty wagon immediately
    after the impact ends, v2 = %.2f m/s.\n",v21)
printf(" When coefficient of restitution is taken
    into account, v1 = %.3f m/s.\n",v12)
printf(" When coefficient of restitution is taken
    into account, v2 = %.3f m/s.\n",v22)
printf(" The amount of energy dissipated during
    impact, EL = %d N-m.\n\n",EL)
```

**Scilab code Exa 3.21** To find strain energy twist and speed

```
//To find strain energy, twist and speed
clc
//Given:
IA=22.5,IB=67.5 //kg-m^2
q=225 //N-m/rad
NA=150,NB=0 //rpm
//Calculating the angular speed of the flywheel
omegaA=2*%pi*NA/60 //rad/s
//Calculating the angular speed of both the
    flywheels at the instant their speeds are equal
omega=IA*omegaA/(IA+IB) //rad/s
//Calculating the kinetic energy of the system at
    that instant
E2=1/2*(IA+IB)*omega^2 //N-m
//Calculating the kinetic energy of the flywheel A
E1=1/2*IA*omegaA^2 //N-m
//Calculating the strain energy stored in the spring
E=E1-E2 //Strain energy stored in the spring, N-m
//Calculating the maximum twist of the spring
theta=sqrt(E*2/q) //radians
thetad=theta*180/%pi //Maximum twist, degrees
//Calculating the speed of each flywheel when the
    spring regains its initial unstrained condition
N=60*omega/(2*%pi)
NA1=2*N-NA //rpm
NB1=2*N-NB //rpm
//Results:
printf("\n\n The strain energy stored in the spring
    is %d N-m.\n",E)
printf(" The maximum twist of the spring, theta = %
    .1f degrees.\n",thetad)
printf(" The speed of flywheel A when the spring
```

```
    regains its initial unstrained condition, NA1 =
    %d rpm, i.e. %d rpm in the opposite direction.\n"
    ,NA1,-NA1)
printf(" The speed of flywheel B when the spring
    regains its initial unstrained condition, NB1 =
    %d rpm.\n",NB1)
```

# Chapter 4

# Simple Harmonic Motion

**Scilab code Exa 4.1** To Find the Velocity and Acceleration

```
//To Find the Velocity and Acceleration
clc
//Given:
N=120 //rpm
r=1,x=0.75 //m
//Solution:
//Calculating Angular Velocity
omega=2*%pi*N/60 //rad/s
//Calculating Velocity of the Piston
v=omega*sqrt(r^2-x^2) //m/s
//Calculating Acceleration of the Piston
a=omega^2*x
//Results:
printf("\n\n The Velocity of the Piston, v = %.2f m/
    s.\n",v)
printf(" The Acceleration of the Piston, a = %.2f m/
    s^2.\n\n",a)
```

**Scilab code Exa 4.2** To Find the Angular Velocity Time and Acceleration

```
//To Find the Angular Velocity, Time and Acceleration
clc
//Given:
x1=.75,x2=2 //m
v1=11,v2=3 //m/s
//Solution:
//We have, 11=omega*sqrt(r^2-.75^2) and 3=omega*sqrt(r^2-2^2).
//These upon solving yield r^2-(121/omega^2)-0.5625=0 and r^2-(9/omega^2)-4=0.
//Take r^2=x and (1/omega^2)=y and the equation become x-121y=0.5625 and x-9y=4.
//Variables Matrix
A=[1 -121; 1 -9]
//Constants Matrix
B=[.5625; 4]
V=A \ B
//Calculating Amplitude of the Particle
r=sqrt(V(1)) //m
//Calculating Angular Velocity of the Particle
omega=sqrt(1/V(2)) //rad/s
//Calculating Periodic Time
tp=2*%pi/omega //seconds
//Calculating Maximum Acceleration
amax=omega^2*r //m/s^2
//Results:
printf("\n\n The Angular Velocity, omega = %.1f rad/s.\n",omega)
printf(" The Periodic Time, tp = %.1f s.\n",tp)
printf(" The Maximum Acceleration, amax = %.2f m/s^2.\n\n",amax)
```

**Scilab code Exa 4.3** To Find the Frequency and Velocity

```
//To Find the Frequency and Velocity
clc
//Given:
m=60 //kg
r=0.0125,x=0.005 //m
//Solution:
//Calculating the Extension of the Spring
delta=(.25/1.5)*60*10^-3 //m
//Calculating the Frequency of the System
n=1/(2*%pi)*sqrt(9.81/delta) //Hz
//Calculating the Angular Velocity of the Mass
omega=sqrt(9.81/delta) //rad/s
//Calculating the Linear Velocity of the Mass
v=omega*sqrt(r^2-x^2)
//Results:
printf("\n\n The Frequency of Natural Vibration, n = %.2f Hz.\n",n)
printf(" The Velocity of the Mass, v = %.2f m/s.\n\n",v)
```

**Scilab code Exa 4.4** To Find the Frequency of Oscillation

```
//To Find the Frequency of Oscillation
clc
//Given:
m=1,m1=2.5 //kg
s=1.8*10^3 //N/m
l=(300+300)*10^-3 //m
//Solution:
//Calculating the Mass Moment of Inertia of the System
IA=(m*l^2/3)+(m1*l^2) //kg-m^2
//Calculating the Ratio of Alpha to Theta
//delta=0.3*theta
//Restoring Force=s*delta=540*theta
```

```
//Restoring torque about A=540*theta*0.3=162*theta N
    -m ...(i)
//Torque about A= IA*alpha=1.02*alpha N-m
                    ...(ii)
//Equating (i) and (ii), 1.02*alpha=162*theta
alphabytheta=162/1.02
//Calculating the Frequency of Oscillation
n=1/(2*%pi)*sqrt(alphabytheta)
//Results:
printf("\n\n The Frequency of Oscillation, n = %.2f
    Hz.\n\n",n)
```

**Scilab code Exa 4.5** To Find the Moment of Inertia

```
//To Find the Moment of Inertia
clc
//Given:
m=85 //kg
h=0.1 //m
//Solution:
//Calculating the Frequency of Oscillation
n=100/145 //Hz
//Calculating the Equivalent Length of Simple
    Pendulum
L=(1/(2*%pi)/.69*sqrt(9.81))^2
//Calculating the Radius of Gyration
kG=sqrt((L-h)*h)
//Calculating the Moment of Inertia of the Flywheel
    through the Centre of Gravity
I=m*kG^2 //kg-m^2
//Results:
printf("\n\n The Moment of Inertia of the Flywheel
    Through its c.g., I = %.1f kg-m^2.\n\n",I)
```

**Scilab code Exa 4.6** To Find the Moment of Inertia

```
//To Find the Moment of Inertia
clc
//Given:
m=60 //kg
d1=75,d2=102 //mm
//Solution:
//Calculating the Frequencies of Oscillation
n1=100/190,n2=100/165 //Hz
//Calculating the Equivalent Lengths of Simple
    Pendulum
L1=9.81/(2*%pi*n1)^2 //m
L2=9.81/(2*%pi*n2)^2 //m
//Calculating Distance of c.g. from the Small and
    Big End Centres (h1 and h2), and the Radius of
    Gyration
function y=f(x)
    h1=x(1)
    h2=x(2)
    kG=x(3)
    y(1)=L1*h1-h1^2-kG^2
    y(2)=L2*h2-h2^2-kG^2
    y(3)=h1+h2-1
endfunction
z=fsolve([1,1,1],f)
h1=z(1),h2=z(2),kG=z(3)
//Calculating the Mass Moment of Inertia of the Rod
I=m*kG^2 //kg-m^2
//Results:
printf("\n\n The Moment of Inertia of the Rod, I =
    %d kg-m^2.\n",I)
printf(" The C.G is at a Distance of h1 = %.3f m
    from the Small End Centre.\n\n",h1)
```

**Scilab code Exa 4.7** To Find the Time of Swing

```
//To Find the Time of Swing
clc
//Given:
l=1.2 //m
theta=3*%pi/180 //rad
//Solution:
//Calculating the Distance Between the Knife Edge
    and C.G. of the Rod
h=1.2/2-.05 //m
//Calculating the Radius of Gyration of the Rod
    About C.G.
kG=l/sqrt(12) //m
//Calculating the Time of Swing of the Rod
tp=2*%pi*sqrt((kG^2+h^2)/(9.81*h)) //seconds
//Calculating the Minimum Time of Swing
tpmin=2*%pi*sqrt((2*kG)/9.81) //seconds
//Calculating Angular Velocity
omega=2*%pi/tp //rad/s
//Calculating Maximum Angular Velocity
omegamax=omega*theta //rad/s
//Calculating Maximum Angular Acceleration
alphamax=omega^2*theta //rad/s^2
//Results:
printf("\n\n The Time of Swing of the Rod, tp = %.2f
     seconds.\n",tp)
printf(" The Minimum Time of Swing, tp(min) = %.2f
    seconds.\n",tpmin)
printf(" The Maximum Angular Velocity, omega(max) =
    %.4f rad/s.\n",omegamax)
printf(" The Maximum Angular Acceleration, alpha(max
    ) = %.3f rad/s^2.\n\n", alphamax)
```

**Scilab code Exa 4.8** To Find Centre of Percussion and Impulse

```
//To Find Centre of Percussion and Impulse
clc
//Given:
m=30 //kg
OG=1.05,h=OG,AG=0.15 //m
//Solution:
//Calculating the Frequency of Oscillation
n=20/43.5 //Hz
//Calculating the Equivalent Length of Simple
    Pendulum
L=9.81/(2*%pi*n)^2 //m
//Calculating the Distance of Centre of Percussion (
    C) from the Centre of Gravity (G)
CG=L-OG //m
//Calculating the Distance of Centre of Percussion (
    C) from the Knife Edge A
AC=AG-CG //m
//Calculating the Radius of Gyration of the Pendulum
     About O
kO=sqrt(L*h) //m
h1=h*(1-cos(60*%pi/180)) //m
//Calculating the Angular Velocity of the Pendulum
omega=sqrt(2*m*9.81*h1/(m*kO^2)) //rad/s
OA=OG+AG
//Calculating the Velocity of Striking
v=omega*(OA) //Velocity of Striking
//Calculating the Angular Velocity of the Pendulum
    Immediately After Impact
I=m*kO^2
LKE=55 //Loss of Kinetic Energy, N-m
omega1=sqrt(omega^2-LKE*2/I)
//Calculating the Impulses at Knife Edge A and at
```

```
Pivot O (P and Q)
CLM=m*h*(omega-omega1) //Change of Linear Momentum
CAM=m*(kO^2-h^2)*(omega-omega1) //Change of Angular Momentum
//P+Q=Change of Linear Momentum and, 0.15P-1.05Q=Change of Angular Momentum.
//i.e., P+Q=CLM and 0.15P-1.05Q=CAM
//Variables Matrix
A=[1 1; 0.15 -1.05]
B=[CLM; CAM]
V=A \ B
P=V(1)
Q=V(2)
//Calculating the Change in Axis Reaction When the Pendulum is Vertical
CAR=m*(omega^2-omega1^2)*h //Change in Axis Reaction, N
//Results:
printf("\n\n The Distance of Centre of Percussion, AC = %.3f m.\n",AC)
printf(" The Velocity of Striking = %.2f m/s.\n",v)
printf(" The Impulse at the Knife Edge, P = %.1f N-s.\n",P)
printf(" The Impulse at the Pivot, Q = %.2f N-s.\n",Q)
printf(" The Change in Axis Reaction When the Pendulum is Vertical = %d N.\n\n",CAR)
```

**Scilab code Exa 4.9** To Find the Radius of Gyration

```
//To Find the Radius of Gyration
clc
//Given:
m=1.5 //kg
l=1.25,x=120*10^-3,y=x //m
```

```
//Solution:
//Calculating the Frequency of Oscillation
n=20/40 //Hz
//Calculating the Radius of Gyration of the
    Connecting Rod
kG=1/(2*%pi*n)*sqrt(9.81*x*y/l) //m
//Calculating the Moment of Inertia of the
    Connecting Rod
I=m*kG^2 //kg-m^2
//Results:
printf("\n\n The Radius of Gyration, kG = %d mm.\n",
    kG*1000)
printf(" The Mass Moment of Inertia, I = %.3f kg-m
    ^2.\n\n",I)
```

**Scilab code Exa 4.10** To Find the Radius of Gyration

```
//To Find the Radius of Gyration
clc
//Given:
l=2.5,r=250*10^-3 //m
//Solution:
//Calculating the Frequency of Oscillation
n=50/170 //Hz
//Calculating the Radius of Gyration of the Wheel
kG=r/(2*%pi*n)*sqrt(9.81/l) //m
//Results:
printf("\n\n The Radius of Gyration, kG = %d mm.\n\n
    ",kG*10^3)
```

**Scilab code Exa 4.11** To Find the Mass Moment of Inertia

```
//To Find the Mass Moment of Inertia
```

```
clc
//Given:
m1=5.5,m2=1.5 //kg
l=1.25,r=125*10^-3 //m
//Solution:
//Calculating the Frequency of Oscillation
n=10/30 //Hz
//Calculating the Radius of Gyration About an Axis
    Through the c.g.
kG=r/(2*%pi*n)*sqrt(9.81/l) //m
//Calculating the Mass Moment of Inertia About an
    Axis Through its c.g.
m=m1+m2 //Total Mass, kg
I=m*kG^2 //kg-m^2
//Results:
printf("\n\n The Mass Moment of Inertia About an
    Axis Through its c.g., I = %.3f kg-m^2.\n\n",I)
```

# Chapter 5

# Simple Mechanisms

**Scilab code Exa 5.1** To find the time ratio

```
//To find the time ratio
clc
//Given:
AC=300,CB1=120 //mm
//Solution:
//Refer Fig. 5.28
//Calculating the sine of inclination of the slotted
    bar with the vertical
sineCAB1=CB1/AC
//Calculating the inclination of the slotted bar
    with the vertical
angleCAB1=asin(sineCAB1)*180/%pi //degrees
//Calculating the angle alpha
alpha=2*(90-angleCAB1) //degrees
//Calculating the ratio of time of cutting stroke to
    time of return stroke
r=(360-alpha)/alpha //Ratio of time of cutting
    stroke to time of return stroke
//Results:
printf("\n\n The ratio of the time of cutting stroke
    to the time of return stroke is %.2f.\n\n",r)
```

**Scilab code Exa 5.2** To find the time ratio

```
//To find the time ratio
clc
//Given:
AC=240,CB1=120,AP1=450 //mm
//Solution:
//Refer Fig. 5.29
//Calculating the sine of inclination of the slotted
    bar with the vertical
sineCAB1=CB1/AC
//Calculating the inclination of the slotted bar
    with the vertical
angleCAB1=asin(sineCAB1)*180/%pi //degrees
//Calculating the angle alpha
alpha=2*(90-angleCAB1) //degrees
//Calculating the time ratio of cutting stroke to
    the return stroke
r=(360-alpha)/alpha //Time ratio of cutting stroke
    to the return stroke
//Calculating the length of the stroke
R1R2=2*AP1*sin(%pi/2-alpha/2*%pi/180) //mm
//Results:
printf("\n\n The time ratio of cutting stroke to the
    return stroke is %d.\n",r)
printf(" The length of the stroke, R1R2 = P1P2 = %d
    mm.\n\n",R1R2)
```

**Scilab code Exa 5.3** To find the dimensions of AC and AP

```
//To find the dimensions of AC and AP
```

```
clc
//Given:
//Refer Fig. 5.30 and Fig. 5.31
BC=30,R1R2=120 //mm
r=1.7 //Time ratio of working stroke to the return
   stroke
//Solution:
//Calculating the angle alpha
alpha=360/(1.7+1) //degrees
//Calculating the length of the link AC
B1C=BC
AC=B1C/cosd(alpha/2) //mm
//Calculating the length of the link AP
AP1=R1R2/(2*cosd(alpha/2)) //mm
AP=AP1
//Results:
printf("\n\n The length of AC = %.1f mm.\n",AC)
printf(" The length of AP = %.1f mm.\n\n",AP)
```

**Scilab code Exa 5.4** To find the time ratio

```
//To find the time ratio
clc
//Given:
CD=50,CA=75,PA=150,PR=135 //mm
//Solution:
//Refer Fig. 5.32 and Fig. 5.33
//Calculating the cosine of angle beta
CA2=CA
cosbeta=CD/CA2
//Calculating the angle beta
beta=2*acos(cosbeta)*180/%pi //degrees
//Calculating the ratio of time of cutting stroke to
    time of return stroke
r=(360-beta)/beta //Ratio of time of cutting stroke
```

```
to time of return stroke
//Calculating the length of effective stroke
R1R2=87.5 //mm
//Results:
printf("\n\n The ratio of time of cutting stroke to time of return stroke is %.3f.\n",r)
printf(" The length of effective stroke, R1R2 = %.1f mm.\n\n",R1R2)
```

# Chapter 6

# Velocity in Mechanisms Instantaneous Centre Method

**Scilab code Exa 6.1** To find the angular velocity

```
//To find the angular velocity
clc
//Given:
NAB=100 //rpm
AB=300/1000,BC=360/1000,CD=BC //m
//Solution:
//Refer Fig. 6.9
//Calculating the angular speed of link AB
omegaAB=2*%pi*NAB/60 //rad/s
//Calculating the velocity of point B on link AB
vB=omegaAB*AB //m/s
//Calculating the angular velocity of link BC
//By measurement from instantaneous centre diagram, Fig. 6.10,
I13B=500/1000 //m
omegaBC=vB/I13B //rad/s
//Results:
printf("\n\n The angular velocity of the link BC, omegaBC = %.3f rad/s.\n\n",omegaBC)
```

**Scilab code Exa 6.2** To find velocity and angular velocity

```
//To find velocity and angular velocity
clc
//Given:
omegaOB=10 //rad/s
OB=100/1000 //m
//Solution:
//Refer Fig. 6.12
//Calculating the velocity of the crank OB
vOB=omegaOB*OB //m/s
vB=vOB
//By measurement from the instantaneous cemtre
    diagram, Fig. 6.13,
I13A=460/1000,I13B=560/1000 //m
//Calculating the velocity of slider A
vA=vB/I13B*I13A
//Calculating the angular velocity of the connecting
     rod AB
omegaAB=vB/I13B //rad/s
//Results:
printf("\n\n The velocity of slider A, vA = %.2f m/s
    .\n",vA)
printf(" The angular velocity of connecting rod AB,
    omegaAB = %.2f rad/s.\n\n",omegaAB)
```

**Scilab code Exa 6.3** To find velocity and angular velocity

```
//To find velocity and angular velocity
clc
//Given:
```

```
NOA=120 //rpm
OA=200/1000 //m/s
//Results:
//Refer Fig. 6.15
//Calculating the angular velocity of the crank OA
omegaOA=2*%pi*NOA/60 //rad/s
//Calculating the velocity of crank OA
vOA=omegaOA*OA //m/s
vA=vOA
//By measurement from the instantaneous cemtre diagram, Fig. 6.16,
I13A=840/1000,I13B=1070/1000,I14B=400/1000,I14C=200/1000,I15C=740/1000,I15D=500/1000 //m
//Calculating the velocity of point B
vB=vA/I13A*I13B //m/s
//Calculating the velocity of point C
vC=vB/I14B*I14C //m/s
//Calculating the velocity of point B
vD=vC/I15C*I15D //m/s
//Calculating the angular velocity of the link AB
omegaAB=vA/I13A //rad/s
//Calculating the angular velocity of the link BC
omegaBC=vB/I14B //rad/s
//Calculating the angular velocity of the link CD
omegaCD=vC/I15C //rad/s
//Results:
printf("\n\n The velocity of point B, vB = %.1f m/s.\n",vB)
printf(" The velocity of point C, vC = %.1f m/s.\n",vC)
printf(" The velocity of point D, vD = %.2f m/s.\n",vD)
printf(" The angular velocity of the link AB, omegaAB = %.2f rad/s.\n",omegaAB)
printf(" The angular velocity of the link BC, omegaBC = %d rad/s.\n",omegaBC)
printf(" The angular velocity of the link CD, omegaCD = %.2f rad/s.\n\n",omegaCD)
```

**Scilab code Exa 6.4** To find the velocity

```
//To find the velocity
clc
//Given:
omegaO1A=100 //rad/s
O1A=100/1000 //m
//Solution:
//Refer Fig. 6.18
//Calculating the linear velocity of crank O1A
vO1A=omegaO1A*O1A //m/s
vA=vO1A
//By measurement from the instantaneous cemtre
    diagram, Fig. 6.19,
I13A=910/1000,I13B=820/1000,I15B=130/1000,I15D
    =50/1000,I16D=200/1000,I16E=400/1000 //m
//Calculating the velocity of point B
vB=vA/I13A*I13B //m/s
//Calculating the velocity of point D
vD=vB/I15B*I15D //m/s
//Calculating the velocity of point E
vE=vD/I16D*I16E //m/s
//Results:
printf("\n\n The velocity of point B, vB = %.2f m/s
    .\n",vB)
printf(" The velocity of point D, vD = %.2f m/s.\n",
    vD)
printf(" The velocity of point E, vE = %.2f m/s.\n",
    vE)
```

**Scilab code Exa 6.5** To find the velocity

```
//To find the velocity
clc
//Given:
NO1A=400 //rpm
O1A=16/1000 //m
//Solution:
//Refer Fig. 6.21
//Calculating the angular velocity of the crank O1A
omegaO1A=2*%pi*NO1A/60 //rad/s
//Calculating the linear velocity of the crank O1A
vO1A=omegaO1A*O1A //m/s
vA=vO1A
//By measurement from the instantaneous cemtre diagram, Fig. 6.22,
I13A=41/1000,I13B=50/1000,I14B=23/1000,I14C=28/1000,I15C=65/1000,I15D=62/1000 //m
//Calculating the velocity of point B
vB=vA/I13A*I13B //m/s
//Calculating the velocity of point C
vC=vB/I14B*I14C //m/s
//Calculating the velocity of of the needle at D
vD=vC/I15C*I15D //m/s
//Results:
printf("\n\n The velocity of the needle at D, vD = %.2f m/s.\n\n",vD)
```

**Scilab code Exa 6.6** To find the velocity of ram

```
//To find velocity of ram
clc
//Given:
NOA=120 //rpm
//Solution:
//Refer Fig. 6.24
//Calculating the angular speed of crank OA
```

```
omegaOA=2*%pi*NOA/60 //rad/s
//By measurement from the instantaneous cemtre diagram, Fig. 6.25,
I12I26=65/1000 //m
//Calculating the velocity of the ram
vD=omegaOA*I12I26 //m/s
//Results:
printf("\n\n The velocity of ram D, vD = %.3f m/s.\n\n",vD)
```

# Chapter 7

# Velocity in Mechanisms Relative Velocity Method

**Scilab code Exa 7.1** To find the angular velocity

```
//To find the angular velocity
clc
//Given:
NBA=120 //rpm
AB=40/1000,CD=80/1000 //m
//Solution:
//Refer Fig. 7.7
//Calculating the angular velocity of the crank AB
omegaBA=2*%pi*NBA/60 //rad/s
//Calculating the velocity of B with respect to A
vBA=omegaBA*AB //m/s
vB=vBA
//By measurement from the velocity diagram, Fig.
   7.7(b),
vCD=0.385 //m/s
vC=vCD
//Calculating the angular velocity of link CD
omegaCD=vCD/CD //rad/s
//Results:
```

```
printf("\n\n The angular velocity of link CD, omegaCD = %.1f rad/s, clockwise about D.\n\n", omegaCD)
```

**Scilab code Exa 7.2** To find velocities angular velocities and position

```
//To find velocities, angular velocities and position
clc
//Given:
NBO=180 //rpm
OB=0.5,PB=2,dO=50/1000,dB=60/1000,dC=30/1000 //m
//Solution:
//Refer Fig. 7.8
//Calculating the angular velocity of the crank BO
omegaBO=2*%pi*NBO/60 //rad/s
//Calculating the linear velocity of B with respect to O
vBO=omegaBO*OB //m/s
vB=vBO
//By measurement from the velocity diagram, Fig. 7.8(b),
vP=8.15,vPB=6.8,vE=8.5,bg=5,bp=vPB,vG=8 //m/s
//Calculating the angular velocity of the connecting rod PB
omegaPB=vPB/PB //rad/s
//Calculating the velocity of rubbing at the pin of crank-shaft
vCS=dO/2*omegaBO //Velocity of rubbing at the pin of crank-shaft, m/s
//Calculating the velocity of rubbing at the pin of crank
vC=dB/2*(omegaBO+omegaPB) //Velocity of rubbing at the pin of crank, m/s
//Calculating the velocity of rubbing at the pin of
```

```
cross-head
vPCH=dC/2*omegaPB //Velocity of rubbing at the pin of cross-head, m/s
//Calculating the position of point G on the connecting rod
BG=bg/bp*PB //m
//Results:
printf("\n\n The velocity of piston P, vP = %.2f m/s.\n",vP)
printf(" The angular velocity of connecting rod, omegaPB = %.1f rad/s, anticlockwise.\n",omegaPB)
printf(" The velocity of point E on the connecting rod, vE = %.1f m/s.\n",vE)
printf(" The velocity of rubbing at the pin of crank-shaft is %.2f m/s.\n",vCS)
printf(" The velocity of rubbing at the pin of crank is %.4f m/s.\n",vC)
printf(" The velocity of rubbing at the pin of cross-head is %.3f m/s.\n",vPCH)
printf(" The position of point G on the connecting rod, BG = %.2f m.\n",BG)
printf(" The linear velocity of point G, vG = %d m/s.\n\n",vG)
```

**Scilab code Exa 7.3** To find the velocity

```
//To find the velocity
clc
//Given:
NAO=600 //rpm
OA=28/1000,BD=46/1000 //m
//Solution:
//Refer Fig. 7.10
//Calculating the angular velocity of crank AO
omegaAO=2*%pi*NAO/60 //rad/s
```

```
//Calculating the velocity of A with respect to O
vAO=omegaAO*OA //m/s
vA=vAO
//By measurement from the velocity diagram, Fig. 7.10(b),
vD=1.6,vDB=1.7 //m/s
//Calculating the angular velocity of D with respect to B
omegaBD=vDB/BD //rad/s
//Results:
printf("\n\n The velocity of the slider D, vD = %.1f m/s.\n",vD)
printf(" The angular velocity of the link BD, omegaBD = %.2f rad/s, clockwise sbout B.\n\n",omegaBD)
```

**Scilab code Exa 7.4** To find velocity angular velocity and rubbing speed

```
//To find velocity, angular velocity and rubbing speed
clc
//Given:
NBA=120 //rpm
AB=150/1000,DC=450/1000,BC=450/1000,dC=50/1000,rC=dC/2 //m
//Sloution:
//Refer Fig. 7.12
//Calculating the angular velocity of the crank AB
omegaBA=2*%pi*NBA/60 //rad/s
//Calculating the linear velocity of B with respect to A
vBA=omegaBA*AB //m/s
vB=vBA
//By measurement from the velocity diagram, Fig. 7.12(b),
```

```
vF=0.7,vCD=2.25,vCB=2.25 //m/s
//Calculating the angular velocity of DC
omegaDC=vCD/DC //rad/s
//Calculating the angular velocity of BC
omegaCB=vCB/BC //rad/s
//Calculating the rubbing speed at the pin C
vr=(omegaCB-omegaDC)*rC //The rubbing speed at the pin C,m/s
//Results:
printf("\n\n The velocity of block F, vF = %.1f m/s.\n",vF)
printf(" The angular velocity of DC, omegaDC = %d rad/s, anticlockwise about D.\n",omegaDC)
printf(" The rubbing speed at the pin C is %d m/s.\n\n",vr)
```

**Scilab code Exa 7.5** To find velocity and angular velocity

```
//To find velocity and angular velocity
clc
//Given:
NAO=120 //rpm
OA=100/1000,CE=350/1000 //m
//Solution:
//Refer Fig. 7.13
//Calculating the angular speed of the crank OA
omegaAO=2*%pi*NAO/60 //rad/s
//Calculating the velocity of A with respect to O
vAO=omegaAO*OA //m/s
vA=vAO
//By measurement from the velocity diagram, Fig. 7.14(b),
vF=0.53,od=1.08,vCE=0.44 //m/s
//Calculating the angular velocity of CE
omegaCE=vCE/CE //rad/s
```

```
//Results:
printf("\n\n The velocity of F, vF = %.2f m/s.\n",vF
    )
printf(" The velocity of sliding of CE in the
    trunnion is %.2f m/s.\n",od)
printf(" The angular velocity of CE, omegaCE = %.2f
    rad/s, clockwise about E.\n\n",omegaCE)
```

**Scilab code Exa 7.6** To find the absolute velocity

```
//To find the absolute velocity
clc
//Given:
NCO=120 //rpm
OC=125/1000 //m
//Solution:
//Refer Fig. 7.15
//Calculating the angular velocity of the crank CO
omegaCO=2*%pi*NCO/60 //rad/s
//Calculating the linear velocity of C with respect
    to O
vCO=omegaCO*OC //m/s
vC=vCO
//By measurement from the velocity diagram, Fig.
    7.16(b),
vCO=1.57,vE=0.7 //m/s
//Results:
printf("\n\n The absolute velocity of point E of the
     lever, vE = %.1f m/s.\n\n",vE)
```

**Scilab code Exa 7.7** To find linear and angular velocity

```
//To find linear and angular velocity
```

```
clc
//Given:
NBO1=40 //rpm
O1O2=800/1000,O1B=300/1000,O2D=1300/1000,DR=400/1000 //m
//Solution:
//Refer Fig. 7.18
//Calculating the angular speed of the crank BO
omegaBO1=2*%pi*NBO1/60 //rad/s
//Calculating the velocity of B with respect to O1
vBO1=omegaBO1*O1B //m/s
vB=vBO1
//By measurement from the velocity diagram, Fig. 7.18(b),
vR=1.44, vDO2=1.32 //m/s
vD=vDO2
//Calculating the angular velocity of the link O2D
omegaDO2=vDO2/O2D //rad/s
//Results:
printf("\n\n The velocity of the ram R, vR = %.2f m/s.\n",vR)
printf(" The angular velocity of the link O2D, omegaDO2 = %.3f rad/s, anticlockwise about O2.\n\n",omegaDO2)
```

**Scilab code Exa 7.8** To find speed and time ratio

```
//To find speed and time ratio
clc
//Given:
NAO1=60 //rpm
O1A=85,rQ=50 //mm
//Solution:
//Refer Fig. 7.20 and Fig. 7.21
//Calculating the angular velocity of AO1
```

```
omegaAO1=2*%pi*NAO1/60 //rad/s
//Calculating the velocity of A with respect to O1
vAO1=omegaAO1*O1A //mm/s
vA=vAO1
//By measurement from the velocity diagram, Fig.
    7.20(b),
vDO2=410 //mm/s
O2D=264 //mm
angleB1O2B2=60*%pi/180 //rad
funcprot(0) //To vary the Scilab function 'beta'
alpha=120,beta=240 //degrees
//Calculating the angular velocity of the quadant Q
omegaQ=vDO2/O2D //rad/s
//Calculating the linear speed of the rack
vR=omegaQ*rQ //mm/s
//Calculating the ratio of times of lowering and
    raising the rack
r=beta/alpha
//Calculating the length of stroke of the rack
L=rQ*angleB1O2B2 //mm
//Results:
printf("\n\n The linear speed of the rack, vR = %.1f
    mm/s.\n",vR)
printf(" The ratio of times of lowering and raising
    the rack is %d.\n",r)
printf(" The length of the stroke of the rack is %.2
    f mm.\n\n",L)
```

**Scilab code Exa 7.9** To find velocity and angular velocity

```
//To find velocity and angular velocity
clc
//Given:
NPO=120 //rpm
OQ=100/1000,OP=200/1000,RQ=150/1000,RS=500/1000 //m
```

```
//Solution:
//Refer Fig. 7.22
//Calculating the angular speed of the crank PO
omegaPO=2*%pi*NPO/60 //rad/s
//Calculating the velocity of P with respect to O
vPO=omegaPO*OP //m/s
vP=vPO
//By measurement from the velocity diagram, Fig.
    7.23(b),
vS=0.8,vSR=0.96,vTP=0.85 //m/s
//Calculating the angular velocity of link RS
omegaRS=vSR/RS //rad/s
//Results:
printf("\n\n The velocity of the slider S (cutting
    tool), vS = %.1f m/s.\n",vS)
printf(" The angular velocity of the link RS,
    omegaRS = %.2f rad/s, clockwise about R.\n",
    omegaRS)
printf(" The velocity of the sliding block T on the
    slotted lever QT, vTP = %.2f m/s.\n\n",vTP)
```

**Scilab code Exa 7.10** To find velocity and resisting torque

```
//To find velocity and resisting torque
clc
//Given:
NAD=100 //rpm
TA=50 //N-m
DA=300/1000,CB=360/1000,AB=CB,DC=600/1000 //m
eta=70/100 //%
//Solution:
//Refer Fig. 7.25
//Calculating the angular velocity of the crank AD
omegaAD=2*%pi*NAD/60 //rad/s
//Calculating the velocity of A with respect to D
```

```
vAD=omegaAD*DA //m/s
vA=vAD
//By measurement from the velocity diagram, Fig. 7.25(b),
vBC=2.25 //m/s
vB=vBC
//Calculating the angular velocity of the driven link CB
omegaBC=vBC/CB //rad/s
//Calculating the actual mechanical advantage
omegaA=omegaAD,omegaB=omegaBC
MAactual=eta*omegaA/omegaB
//Calculating the resisting torque
TB=eta*TA*omegaA/omegaB //N-m
//Results:
printf("\n\n The velocity of the point B, vB = %.2f m/s.\n",vB)
printf(" The angular velocity of the driven link CB, omegaBC = %.2f rad/s.\n",omegaBC)
printf(" The actual mechanical advantage, M.A.(actual) = %.2f.\n",MAactual)
printf(" The resisting torque, TB = %.1f N-m.\n\n",TB)
```

**Scilab code Exa 7.11** To find velocity ratio

```
//To find velocity ratio
clc
//Given:
WC=2.5*1000,WD=4*1000 //N
OA=175/1000,AB=180/1000,AD=500/1000,BC=325/1000 //m
//Solution:
//Refer Fig. 7.26
//Assuming the speed of crank OA to be 'N'
//Calculating the angular velocity of crank OA
```

```
omegaAO=mulf('2*%pi/60','N')
//Calculating the velocity of A with respect to O
vAO=mulf('omegaAO','OA')
vA=vAO
//Assume the vector oa (i.e. velocity of A) as 20 m/s
N=20/(2*%pi/60*OA) //mm
//By measurement from the velocity diagram, Fig. 7.27(b),
vC=35,vD=21 //mm
//Calculating the velocity ratio between C and the ram D
r=vC/vD //The velocity ratio between C and the ram D
//Calculating the efficiency of the machine
eta=(WD*vD)/(WC*vC)*100 //%
//Results:
printf("\n\n The velocity ratio between C and the ram D is %.2f.\n",r)
printf(" The efficiency of the machine, eta = %d %s.\n\n",eta,'%')
```

**Scilab code Exa 7.12** To find velocity angular velocity and torque

```
//To find velocity, angular velocity and torque
clc
//Given:
NAO=180 //rpm
OA=180/1000,CB=240/1000,AB=360/1000,BD=540/1000 //m
FD=2*1000 //N
DA=30/1000,DD=DA,rA=DA/2,rD=DD/2 //m
//Solution:
//Refer Fig. 7.28
//Calculating the angular velocity of the crank OA
omegaAO=2*%pi*NAO/60 //rad/s
//Calculating the velocity of A with respect to O
```

```
vAO=omegaAO*OA
vA=vAO
//By measurement fro the velocity diagram, Fig. 7.29(b)
vD=2.05,vBA=0.9,vBC=2.8,vDB=2.4 //m/s
//Calculating the angular velocity of the link AB
omegaAB=vBA/AB //rad/s
//Calculating the angular velocity of the link CB
omegaCB=vBC/CB //rad/s
//Calculating the angular velocity of the link BD
omegaBD=vDB/BD //rad/s
//Calculating the relative angular velocity at A
rvA=omegaCB-omegaAB+omegaBD //The relative angular velocity at A, rad/s
//Calculating the relative angular velocity at D
rvD=omegaBD //The relative angular velocity at D, rad/s
//Calculating the velocity of rubbing on the pin A
vrA=rvA*rA*1000 //The velocity of rubbing on the pin A, mm/s
//Calculating the velocity of rubbing on the pin D
vrD=rvD*rD*1000 //The velocity of rubbing on the pin D, mm/s
//Calculating the torque applied to crank OA
TA=FD*vD/omegaAO //N-m
//Results:
printf("\n\n The velocity of slider D, vD = %.2f m/s.\n",vD)
printf(" The angular velocity of the link AB, omegaAB = %.1f rad/s, anticlockwise about A.\n",omegaAB)
printf(" The angular velocity of the link CB, omegaCB = %.2f rad/s, anticlockwise about C.\n",omegaCB)
printf(" The angular velocity of the link BD, omegaBD = %.2f rad/s, clockwise about B.\n",omegaBD)
printf(" The velocity of rubbing on the pin A is %d
```

```
mm/s.\n",vrA)
printf(" The velocity of rubbing on the pin D is %d mm/s.\n",vrD)
printf(" The torque applied to the crank OA, TA = %.1f N-m.\n\n",TA)
```

**Scilab code Exa 7.13** To find the velocities

```
//To find the velocities
clc
//Given:
NBA=180 //rpm
AB=0.45,BD=1.5,BC=0.9,CE=BC //m
FD=500,FE=750 //N
//Solution:
//Refer Fig. 7.31
//Calculating the angular velocity of the crank AB
omegaBA=2*%pi*NBA/60 //rad/s
//Calculating the velocity of B with respect to A
vBA=omegaBA*AB //m/s
vB=vBA
//By measurement from the velocity diagram, Fig. 7.31(b),
vD=9.5,vE=1.7 //m/s
//Calculating the power input
Pi=FD*vD-FE*vE //N-m/s
//Calculating the turning moment at A
TA=Pi/omegaBA //N-m
//Results:
printf("\n\n The velocity of slider D, vD = %.1f m/s.\n",vD)
printf(" The velocity of slider E, vE = %.1f m/s.\n",vE)
printf(" The turning moment at A, TA = %.1f N-m.\n\n",TA)
```

# Chapter 8

# Acceleration in Mechanisms

**Scilab code Exa 8.1** To find linear and agular velocity and acceleration

```
//To find linear and agular velocity and
   acceleration
clc
//Given:
NBO=300 //rpm
OB=150/1000,BA=600/1000 //m
//Solution:
//Refer Fig. 8.4
//Calculating the angular velocity of BO
omegaBO=2*%pi*NBO/60 //rad/s
//Calculating the linear velocity of B with respect
   to O
vBO=omegaBO*OB //m/s
vB=vBO
//By measurement from the velocity diagram, Fig.
   8.4(b),
vAB=3.4,vD=4.1 //m/s
//Calculating the radial component of the
   acceleration of B with respect of O
arBO=vBO^2/OB //m/s^2
aB=arBO
```

```
//Calculating the radisla component of the accaleration of A with respect to B
arAB=vAB^2/BA //m/s^2
//By measurement from the acceleration diagram, Fig. 8.4(c),
aD=117,adashAB=103 //m/s^2
//Calculating the angular velocity of the connecting rod
omegaAB=vAB/BA //rad/s^2
//Calculating the angular acceleration of the connecting rod
alphaAB=adashAB/BA //rad/s^2
//Results:
printf("\n\n The linear velocity of the midpoint of the connecting rod, vD = %.1f m/s.\n",vD)
printf(" The linear acceleration of the midpoint of the connecting rod, aD = %d m/s^2.\n",aD)
printf(" The angular velocity of the connecting rod, omegaAB = %.2f rad/s, anticlockwise about B.\n",omegaAB)
printf(" The angular acceleration of the connecting rod, alphaAB = %.2f rad/s^2, clockwise about B.\n\n",alphaAB)
```

**Scilab code Exa 8.2** To find linear and angular velocity and acceleration

```
//To find linear and angular velocity and acceleration
clc
//Given:
omegaBC=75 //rad/s
alphaBC=1200 //rad/s^2
CB=100/1000,BA=300/1000 //m/
//Solution:
//Refer Fig. 8.5
```

```
//Calculating the linear velocity of B with respect to C
vBC=omegaBC*CB //m/s
//Calculating the tangential component of the acceleration of B with respect to C
alphatBC=alphaBC*CB //m/s^2
//By measurement from the velocity diagram, Fig. 8.6(b),
vG=6.8,vAB=4 //m/s
//Calculating the angular velocity of AB
omegaAB=vAB/BA //rad/s
//Calculating the radial component of the acceleration of B with respect to C
arBC=vBC^2/CB //m/s^2
//Calculating the radial component of the acceleration of A with respect to B
arAB=vAB^2/BA //m/s^2
//By measurement from the acceleration diagram, Fig. 8.6(c),
arBC=120,arAB=53.3,aG=414,atAB=546 //m/s^2
//Calculating the angular acceleration of AB
alphaAB=atAB/BA //rad/s^2
//Results:
printf("\n\n The velocity of G, vG = %.1f m/s.\n",vG)
printf(" The angular velocity of AB, omegaAB = %.1f rad/s, clockwise.\n",omegaAB)
printf(" The acceleration of G, aG = %d m/s^2.\n",aG)
printf(" The angular accaleration of AB, alphaAB = %d rad/s^2.\n\n",alphaAB)
```

**Scilab code Exa 8.3** To find linear and angular acceleration

```
//To find linear and angular acceleration
```

```
clc
//Given:
vC=1,vCD=vC //m/s
aC=2.5 //m/s^2
AB=3,BC=1.5 //m
//Solution:
//Refer Fig. 8.8
//By measurement from the velocity diagram, Fig. 8.8(b),
vBA=0.72,vBC=0.72 //m/s
//Calculating the radial component of acceleration of B with respect to C
arBC=vBC^2/BC //m/s^2
//Calculating the radial component of acceleration of B with respect to A
arBA=vBA^2/AB //m/s^2
//By measurement from the acceleration diagram, Fig. 8.8(c),
aCD=2.5,aC=aCD,arBC=0.346,arBA=0.173, atBA=1.41,atBC=1.94,vectorbb=1.13,vectorab=0.9 //m/s^2
//Calculating the angular accaleration of AB
alphaAB=atBA/AB //rad/s^2
//Calculating the angular accaleration of BC
alphaBC=atBC/BC //rad/s^2
//Results:
printf("\n\n The magnitude of vertical component of the acceleration of the point B is %.2f m/s^2.\n",vectorbb)
printf(" The magnitude of horizontal component of the acceleration of the point B is %.1f m/s^2.\n",vectorab)
printf(" The angular acceleration of the link AB, alphaAB = %.2f rad/s^2.\n",alphaAB)
printf(" The angular acceleration of the link BC, alphaBC = %.1f rad/s^2.\n\n",alphaBC)
```

**Scilab code Exa 8.4** To find the angular velocity and acceleration

```
//To find the angular velocity and acceleration
clc
//Given:
omegaQP=10 //rad/s
PQ=62.5/1000,QR=175/1000,RS=112.5/1000,PS=200/1000 //m
//Solution:
//Refer Fig. 8.9
//Calculating the velocity of Q with respect to P
vQP=omegaQP*PQ //m/s
vQ=vQP
//By measurement from the velocity diagram, Fig. 8.9(b),
vRQ=0.333,vRS=0.426,vR=vRS //m/s
//Calculating the angular velocity of link QR
omegaQR=vRQ/QR //rad/s
//Calculating the angular velocity of link RS
omegaRS=vRS/RS //rad/s
//Calculating the radial component of the acceleration of Q with respect to P
arQP=vQP^2/PQ //m/s^2
aQP=arQP,aQ=aQP
//Calculating the radial component of the acceleration of R with respect to Q
arRQ=vRQ^2/QR //m/s^2
//Calculating the radial component of the acceleration of R with respect to S
arRS=vRS^2/RS //m/s^2
aRS=arRS,aR=aRS
//By measurement from the acceleration diagram, Fig. 8.9(c),
atRQ=4.1,atRS=5.3 //m/s^2
```

```
//Calculating the angular acceleration of link QR
alphaQR=atRQ/QR //rad/s^2
//Calculating the angular acceleration of link RS
alphaRS=atRS/RS //rad/s^2
//Results:
printf("\n\n The angular velocity of link QR,
    omegaQR = %.1f rad/s, anticlockwise.\n",omegaQR)
printf(" The angular velocity of link RS, omegaRS =
    %.2f rad/s, clockwise.\n",omegaRS)
printf(" The angular acceleration of link QR,
    alphaQR = %.2f rad/s^2, anticlockwise.\n",alphaQR
    )
printf(" The angular acceleration of link RS,
    alphaRS = %.1f rad/s^2, anticlockwise.\n\n",
    alphaRS)
```

**Scilab code Exa 8.5** To find angular velocities and accelerations

```
//To find angular velocities and accelerations
clc
//Given:
omegaAP1=10 //rad/s
alphaAP1=30 //rad/s^2
P1A=300/1000,P2B=360/1000,AB=P2B //m
//Solution:
//Refer Fig. 8.10
//Calculating the velocity of A with respect to P1
vAP1=omegaAP1/P1A //m/s
vA=vAP1
//By measurement from the velocity diagram, Fig.
    8.11(b),
vBP2=2.2,vBA=2.05 //m/s
//Calculating the angular velocity of P2B
omegaP2B=vBP2/P2B //rad/s
//Calculating the angular velocity of AB
```

```
omegaAB=vBA/AB //rad/s
//Calculating the tangential component of the
    acceleration of A with respect to P1
atAP1=alphaAP1*P1A //m/s^2
//Calculating the radial component of the
    acceleration of A with respect to P1
arAP1=vAP1^2/P1A //m/s^2
//Calculating the radial component of the
    acceleration of B with respect to A
arBA=vBA^2/AB //m/s^2
//Calculating the radial component of B with respect
     to P2
arBP2=vBP2^2/P2B //m/s^2
//By measurement from the acceleration diagram, Fig.
     8.11(c),
aBP2=29.6,aB=aBP2,atBA=13.6,atBP2=26.6 //m/s^2
//Calculating the angular acceleration of P2B
alphaP2B=atBP2/P2B //rad/s^2
//Calculating the angular acceleration of AB
alphaAB=atBA/AB //rad/s^2
//Results:
printf("\n\n The velocity of P2B, vBP2 = %.1f m/s.\n
    ",vBP2)
printf(" The angular velocity of P2B, omegaP2B = %.1
    f rad/s, clockwise.\n",omegaP2B)
printf(" The angular velocity of AB, omegaAB = %.1f
    rad/s, anticlockwise.\n",omegaAB)
printf(" The acceleration of the joint B, aB = %.1f
    m/s^2.\n",aB)
printf(" The angular acceleration of P2B, alphaP2B =
     %.1f rad/s^2, anticlockwise.\n",alphaP2B)
printf(" The angular acceleration of AB, alphaAB = %
    .1f rad/s^2, anticlockwise.\n\n",alphaAB)
```

**Scilab code Exa 8.6** To find velocities and accelerations

```
//To find velocities and accelerations
clc
//Given:
NAO=20 //rpm
OA=300/1000,AB=1200/1000,BC=450/1000,CD=BC //m
//Solution:
//Refer Fig. 8.13
//Calculating the angular velocity of crank AO
omegaAO=2*%pi*NAO/60 //rad/s
//Calculating the linear velocity of A with respect to O
vAO=omegaAO*OA //m/s
vA=vAO
//By measurement from the velocity diagram, Fig. 8.13(b),
vB=0.4,vD=0.24,vDC=0.37,vBA=0.54 //m/s
//Calculating the angular velocity of CD
omegaCD=vDC/CD //rad/s
//Calculating the radial component of the acceleration of A with respect to O
arAO=vAO^2/OA //m/s^2
//Calculating the radial component of the acceleration of B with respect to A
arBA=vBA^2/AB //m/s^2
//Calculating the radial component of the acceleration of D with respect to C
arDC=vDC^2/CD //m/s^2
//By measurement from the acceleration diagram, Fig. 8.13(c),
aD=0.16,atDC=1.28 //m/s^2
//Calculating the angular acceleration of CD
alphaCD=atDC/CD //rad/s^2
//Results:
printf("\n\n Velocity of sliding at B, vB = %.1f m/s.\n",vB)
printf(" Velocity of sliding at D, vD = %.2f m/s.\n",vD)
printf(" Angular velocity of CD, omegaCD = %.2f rad/
```

```
s.\n",omegaCD)
printf(" Linear acceleration of D, aD = %.2f m/s^2.\n",aD)
printf(" Angular acceleration of CD, alphaCD = %.2f rad/s^2, clockwise.\n\n",alphaCD)
```

**Scilab code Exa 8.7** To find linear and angular acceleration

```
//To find linear and angular acceleration
clc
//Given:
NAO=180 //rpm
OA=150/1000,AB=450/1000,PB=240/1000,CD=660/1000 //m
//solution:
//Refer Fig. 8.15
//Calculating the angular speed of crank AO
omegaAO=2*%pi*NAO/60 //rad/s
//Calculating the velocity of A with respect to O
vAO=omegaAO*OA //m/s
vA=vAO
//By measurement from the velocity diagram, Fig. 8.15(b),
vD=2.36,vDC=1.2,vBA=1.8,vBP=1.5 //m/s
//Calculating the radial component of the acceleration of B with respect to A
arAO=vBA^2/AB //m/s^2
//Calculating the radial component of the acceleration of B with respect to A
arBA=vBA^2/AB //m/s^2
//Calculating the radial component of the acceleration of B with respect to P
arBP=vBP^2/PB //m/s^2
//Calculating the radial component of D with respect to C
arDC=vDC^2/CD //m/s^2
```

```
//By measurement from the acceleration diagram, Fig. 8.15(c),
aD=69.6,atDC=17.4 //m/s^2
//Calculating the angular acceleration of CD
alphaCD=atDC/CD //rad/s^2
//Results:
printf("\n\n Acceleration of slider D, aD = %.1f m/s^2.\n",aD)
printf(" Angular acceleration of link CD, alphaCD = %.1f rad/s^2.\n\n",alphaCD)
```

**Scilab code Exa 8.8** To find linear and angular velocities and accelerations

```
//To find linear and angular velocities and accelerations
clc
//Given:
NAO=180 //rpm
OA=180/1000,CB=240/1000,AB=360/1000,BD=540/1000 //m
alphaAO=50 //rad/s^2
//Solution:
//Refer Fig. 8.17
//Calculating the angular speed of crank AO
omegaAO=2*%pi*NAO/60 //rad/s
//Calculating the velcoity of A with respect to O
vAO=omegaAO*OA //m/s
vA=vAO
//By measurement from the velocity diagram, Fig. 8.17(b),
vBA=0.9,vBC=2.4,vDB=2.4,vD=2.05 //m/s
//Calculating the angular velocity of BD
omegaBD=vDB/BD //rad/s
//Calculating the tangential component of the acceleration of A with respect to O
atAO=alphaAO*OA //m/s^2
```

```
//Calculating the radial component of the
    acceleration of A with respect to O
arAO=vAO^2/OA //m/s^2
//Calculating the radial component of the
    acceleration of B with respect to A
arBA=vBA^2/AB //m/s^2
//Calculating the radial component of the
    acceleration of B with respect to C
arBC=vBC^2/AB //m/s^2
//Calculating the radial component of the
    acceleration of D with respect to B
arDB=vDB^2/BD //m/s^2
//By measurement from the acceleration diagram, Fig.
     8.17(c),
aD=13.3,atDB=38.5 //m/s^2
//Calculating the angular acceleration of BD
alphaBD=atDB/BD //rad/s^2
//Results:
printf("\n\n Velocity of slider D, vD = %.2f m/s.\n"
    ,vD)
printf(" Angular velocity of BD, omegaBD = %.1f rad/
    s.\n",omegaBD)
printf(" Acceleration of slider D, aD = %.1f m/s^2.\
    n",aD)
printf(" Angular acceleration of BD, alphaBD = %.1f
    rad/s^2, clockwise.\n\n",alphaBD)
```

**Scilab code Exa 8.9** To find velocity and accelerations

```
//To find velocity and accelerations
clc
//Given:
omegaAO1=100 //rad/s
O1A=100/1000,AC=700/1000,BC=200/1000,BD=150/1000,O2D
    =200/1000,O2E=400/1000,O3C=200/1000 //m
```

```
//Solution:
//Refer Fig. 8.19
//Calculating the linear velocity of A with respect to O1
vAO1=omegaAO1/O1A //m/s
vA=vAO1
//By measurement from the velocity diagram, Fig. 8.19(b),
vCA=7,vCO3=10,vC=vCO3,vDB=10.2,vDO2=2.8,vD=vDO2,vE=5.8,vEO2=vE //m/s
//Calculating the radial component of the acceleration of A with respect to O1
arAO1=vAO1^2/O1A //m/s^2
aAO1=arAO1,aA=aAO1
//Calculating the radial component of the acceleration of C with respect to A
arCA=vCA^2/AC //m/s^2
//Calculating the radial component of the acceleration of C with respect to O3
arCO3=vCO3^2/O3C //m/s^2
//Calculating the radial component of the acceleration of D with respect to B
arDB=vDB^2/BD //m/s^2
//Calculating the radial component of the acceleration of D with respect to O2
arDO2=vDO2^2/O2D //m/s^2
//Calculating the radial component of the acceleration of E with respect to O2
arEO2=vEO2^2/O2E //m/s^2
//By measurement from the acceleration diagram, Fig. 8.19(c),
aE=1200,atDO2=610 //m/s^2
aEO2=aE
aB=440 //Acceleration of point B, m/s^2
//Calculating the angular acceleration of the bell crank lever
alpha=atDO2/O2D //The angular acceleration of the bell crank lever, rad/s^2
```

```
//Results:
printf("\n\n Velocity of the point E on the bell crank lever, vE = %.1f m/s.\n",vE)
printf(" Acceleration of point B = %d m/s^2.\n",aB)
printf(" Acceleration of point E, aE = %d m/s^2.\n",aE)
printf(" Angular acceleration of the bell crank lever = %d rad/s^2, anticlockwise.\n\n",alpha)
```

**Scilab code Exa 8.10** To find velocity torque and acceleration

```
//To find velocity, torque and acceleration
clc
//Given:
NAO=100 //rpm
OA=150/1000,AB=600/1000,BC=350/1000,CD=150/1000,DE=500/1000 //m
dA=50/1000,dB=dA,rA=dA/2,rB=dB/2 //m
pF=0.35 //N/mm^2
DF=250 //mm
//Solution:
//Refer Fig. 8.21
//Calculating the angular speed of the crank AO
omegaAO=2*%pi*NAO/60 //rad/s
//Calculating the velocity of A with respect to O
vAO=omegaAO*OA //m/s
vA=vAO
//By measurement from the velocity diagram, Fig. 8.21(b),
vBA=1.65,vBC=0.93,vB=vBC,vED=0.18,vEO=0.36,vE=vEO,vF=vE //m/s
//Calculating the velocity of D with respect to C
vDC=vBC*CD/BC //m/s
//Calculating the angular velocity of B with respect to A
```

```
omegaBA=vBA/AB //rad/s
//Calculating the angular velocity of B with respect to C
omegaBC=vBC/BC //rad/s
//Calculating the rubbing velocity of pin at A
vrA=(omegaAO-omegaBA)*rA //The rubbing velocity of pin at A, m/s
//Calculating the rubbing velocity of pin at B
vrB=(omegaBA+omegaBC)*rB //The rubbing velocity of pin at B, m/s
//Calculating the force at the pump piston at F
FF=pF*%pi/4*DF^2 //N
//Calculating the force required at the crankshaft A
FA=FF*vF/vA //N
//Calculating the torque required at the crankshaft
TA=FA*OA //N-m
//Calculating the radial component of the acceleration of A with respect to O
arAO=vAO^2/OA //m/s^2
//Calculating the radial component of the acceleration of B with respect to A
arBA=vBA^2/AB //m/s^2
//Calculating the radial component of the acceleration of B with respect to C
arBC=vBC^2/BC //m/s^2
//Calculating the radial component of the acceleration of E with respect to D
arED=vED^2/DE //m/s^2
//By measurement from the acceleration diagram, Fig. 8.21(c),
aBC=9.2,aB=aBC,aBA=9,aE=3.8 //m/s^2
//Calculating the acceleration of D
aD=aBC*CD/BC //m/s^2
//Results:
printf("\n\n The velocity of the cross-head E, vE = %.2f m/s.\n",vE)
printf(" The rubbing velocity of pin at A = %.3f m/s.\n",vrA)
```

```
printf(" The rubbing velocity of pin at B = %.3f m/s.\n",vrB)
printf(" The torque required at the crankshaft, TA = %d N-m.\n",TA)
printf(" The acceleration of the crosshead E, aE = %.1f m/s^2.\n\n",aE)
```

**Scilab code Exa 8.11** To find velocity and acceleration

```
//To find velocity and acceleration
clc
//Given:
NAO=150 //rpm
OA=150/1000,AB=550/1000,AC=450/1000,DC=500/1000,BE=350/1000 //m
//Solution:
//Refer Fig. 8.23
//Calculating the angular speed of the crank AO
omegaAO=2*%pi*NAO/60 //rad/s
//Calculating the linear velocity of A with respect to O
vAO=omegaAO*OA //m/s
vA=vAO
//By measurement from the velocity diagram, Fig. 8.23(b),
vCA=0.53,vCD=1.7,vC=vCD,vEB=1.93,vE=1.05 //m/s
//Calculating the radial component of the acceleration of A with respect to O
arAO=vAO^2/OA //m/s^2
aA=arAO
//Calculating the radial component of the acceleration of C with respect to A
arCA=vCA^2/AC //m/s^2
//Calculating the radial component of the acceleration of C with respect to D
```

```
arCD=vCD^2/DC //m/s^2
//Calculating the radial component of the
    acceleration of E with respect to B
arEB=vEB^2/BE //m/s^2
//By measurement from the acceleration diagram, Fig.
    8.23(c),
aE=3.1 //m/s^2
//Results:
printf("\n\n Velocity of the ram E, vE = %.2f m/s.\n
    ",vE)
printf(" Acceleration of the ram E, aE = %.1f m/s
    ^2.\n\n",aE)
```

**Scilab code Exa 8.12** To find the velocity and acceleration

```
//To find the velocity and acceleration
clc
//Given:
NDC=1140 //rpm
AB=80/1000,CD=40/1000,BE=150/1000,DE=BE,EP=200/1000
    //m
//Solution:
//Refer Fig. 8.25
//Calculating the angular speed of the link CD
omegaDC=2*%pi*NDC/60 //rad/s
//Calculating the velocity of D with respect to C
vDC=omegaDC*CD //m/s
vD=vDC
//Calculating the angular speed of the larger wheel
omegaBA=omegaDC*CD/AB //rad/s
//Calculating the velocity of B with respect to A
vBA=omegaBA*AB //m/s
vB=vBA
//By measurement from the velocity diagram, Fig.
    8.25(b),
```

```
vEB=8.1,vED=0.15,vPE=4.7,vP=0.35 //m/s
//Calculating the radial component of the acceleration of B with respect to A
arBA=vBA^2/AB //m/s^2
//Calculating the radial component of the acceleration of D with respect to C
arDC=vDC^2/CD //m/s^2
//Calculating the radial component of the acceleration of E with respect to B
arEB=vEB^2/BE //m/s^2
//Calculating the radial component of the acceleration of E with respect to D
arED=vED^2/DE //m/s^2
//Calculating the radial component of the acceleration of P with respect to E
arPE=vPE^2/EP //m/s^2
//By measurement from the acceleration diagram, Fig. 8.25(c),
aP=655 //m/s^2
//Results:
printf("\n\n Velocity of P, vP = %.2f m/s.\n",vP)
printf(" Acceleration of the piston P, aP = %d m/s^2.\n\n",aP)
```

**Scilab code Exa 8.13** To find velocity and acceleration

```
//To find velocity and acceleration
clc
//Given:
NBA=120 //rpm
AB=150/1000,OC=700/1000,CD=200/1000 //m
//Solution:
//Refer Fig. 8.29
//Calculating the angular speed of the crank AB
omegaAB=2*%pi*NBA/AB //rad/s
```

```
//Calculating the velocity of B with respect to A
vBA=omegaBA*AB //m/s
//By measurement from the velocity diagram, Fig. 8.29(b),
vD=2.15,vBBdash=1.05,vDC=0.45,vBdashO=1.55,vCO=2.15 //m/s
BdashO=0.52 //m
//Calculating the angular velocity of the link OC or OB'
omegaCO=vCO/OC //rad/s
omegaBdashO=omegaCO //rad/s
//Calculating the radial component of the acceleration of B with respect to A
arBA=omegaAB^2/AB //m/s^2
//Calculating the coriolis component of the acceleration of slider B with respect to the coincident point B'
acBBdash=2*omegaCO*vBBdash //m/s^2
//Calculating the radial component of the acceleration of D with respect to C
arDC=vDC^2/CD //m/s^2
//Calculating the radial component of the acceleration of B' with respect to O
arBdashO=vBdashO^2/BdashO //m/s^2
//By measurement fro the acceleration diagram, Fig. 8.29(c),
aD=8.4,atBdashO=6.4 //m/s^2
//Calculating the angular acceleration of the slotted lever
alpha=atBdashO/BdashO //The angular acceleration of the slotted lever, rad/s^2
//Results:
printf("\n\n Velocity of the ram D, vD = %.2f m/s.\n",vD)
printf(" Acceleration of the ram D, aD = %.1f m/s^2.\n",aD)
printf(" Angular acceleration of the slotted lever = %.1f rad/s^2, anticlockwise.\n\n",alpha)
```

**Scilab code Exa 8.14** To find the acceleration

```
//To find the acceleration
clc
//Given:
NBA=200 //rpm
AB=75/1000,PQ=375/1000,QR=500/1000 //m
//Solution:
//Refer Fig. 8.31
//Calculating the angular velocity of the crank AB
omegaBA=2*%pi*NBA/60 //rad/s
//Calculating the velocity of B with respect to A
vBA=omegaBA*AB //m/s
//By measurement from the velocity diagram, Fig.
    8.31(b),
vR=1.6,vBdashB=1.06,vBdashP=1.13,vRQ=0.4,vQP=1.7 //m
    /s
PBdash=248/1000 //m
//Calculating the angular velocity of the link PQ
omegaPQ=vQP/PQ //rad/s
//Calculating the radial component of the
    acceleration of B with respect to A
arBA=omegaBA^2*AB //m/s^2
//Calculating the coriolis component of the
    acceleration of B with respect to coincident
    point B'
acBBdash=2*omegaPQ*vBdashB //m/s^2
//Calculating the radial component of the
    acceleration of R with respect to Q
arRQ=vRQ^2/QR //m/s^2
//Calculating the radial component of the
    acceleration of B' with respect to P
arBdashP=vBdashP^2/PBdash //m/s^2
//By measurement from the acceleration diagram, Fig.
```

```
   8.31(d),
aR=22,aBBdash=18 //m/s^2
//Results:
printf("\n\n Velocity of the tool-box R, vR = %.1f m
   /s.\n",vR)
printf(" Acceleration of the tool-box R, aR = %d m/s
   ^2.\n",aR)
printf(" The acceleration of sliding of the block B
   along the slotted lever PQ, aBBdash = %d m/s^2.\n
   \n",aBBdash)
```

**Scilab code Exa 8.15** To find linear and angular acceleration

```
//To find linear and angular acceleration
clc
//Given:
NAO=30 //rpm
OA=150/1000,OC=100/1000,CD=125/1000,DR=500/1000 //m
//Solution:
//Refer Fig. 8.33
//Calculating the angular speed of the crank OA
omegaAO=2*%pi*NAO/60 //rad/s
//Calculating the velocity of A with respect to O
vAO=omegaAO*OA //m/s
vA=vAO
//By measurement from the velocity diagram, Fig.
   8.33(b),
vBC=0.46,vAB=0.15,vRD=0.12 //m/s
CB=240/1000 //m
//Calculating the angular velocity of the link BC
omegaBC=vBC/CB //rad/s
//Calculating the radial component of the
   acceleration of A with respect to O
arAO=vAO^2/OA //m/s^2
//Calculating the coriolis component of the
```

```
   acceleration of A with respect to coincident
   point B
acAB=2*omegaBC*vAB //m/s^2
//Calculating the radial component of the
   acceleration of B with respect to C
arBC=vBC^2/CB //m/s^2
//Calculating the radial component of the
   acceleration of R with respect to D
arRD=vRD^2/DR //m/s^2
//By measurement from the acceleration diagram, Fig.
    8.33(c),
aR=0.18,atBC=0.14 //m/s^2
//Calculating the angular acceleration of the
   slotted lever CA
alphaCA=atBC/CB //rad/s^2
alphaBC=alphaCA
//Results:
printf("\n\n Acceleration of the sliding block R, aR
    = %.2f m/s^2.\n",aR)
printf(" Angular acceleration of the slotted lever
   CA, alphaCA = %.3f rad/s^2, anticlockwise.\n\n",
   alphaCA)
```

**Scilab code Exa 8.16** To find linear and angular acceleration

```
//To find linear and angular acceleration
clc
//Given:
AB=125/1000 //m
NCO=300 //rpm
//Solution:
//Refer Fig. 8.35
//By measurement from the space diagram, Fig. 8.35(a
   ),
OC=85/1000 //m
```

```
//Calculating the angular velocity of the link CO
omegaCO=2*%pi*NCO/60 //rad/s
//Calculating the velocity of C with respect to O
vCO=omegaCO*OC //m/s
vC=vCO
//By measurement from the velocity diagram, Fig.
    8.35(b),
vBC=0.85,vBA=2.85,vB=vBA //m/s
//Calculating the radial component of of the
    acceleration of C with respect to O
arCO=vCO^2/OC //m/s^2
//Calculating the coriolis component of of
    acceleration of the piston B with respect to the
    cylinder or the coincident point C
acBC=2*omegaCO*vBC //m/s^2
//Calculating the radial component of of the
    acceleration of B with respect to A
arBA=vBA^2/AB //m/s^2
//By measurement from the acceleration diagram, Fig.
    8.35(d),
aBC=73.2,atBA=37.6 //m/s^2
//Calculating the angular acceleration of the
    connecting rod AB
alphaAB=atBA/AB //rad/s^2
//Results:
printf("\n\n Acceleration of the piston inside the
    cylinder, aBC = %.1f m/s^2.\n",aBC)
printf(" Angular acceleration of the connecting rod
    AB, alphaAB = %d rad/s^2, clockwise.\n\n",alphaAB
    )
```

**Scilab code Exa 8.17** To find velocities and acceleration

```
//To find velocities and acceleration
clc
```

```
//Given:
NAO=100 //rpm
OA=50/1000,AB=350/1000,DE=250/1000,EF=DE,CB=125/1000
    //m
//Solution:
//Refer Fig. 8.37
//Calculating the angular velocity of the crank AO
omegaAO=2*%pi*NAO/60 //rad/s
//Calculating the velocity of A with respect to O
vAO=omegaAO*OA //m/s
vA=vAO
//By measurement from the velocity diagram, Fig.
   8.37(b),
vBA=0.4,vBC=0.485,vB=vBC,vSD=0.265,vQS=0.4,vED=0.73,
   vFE=0.6,vF=0.27 //m/s
DS=85/1000 //m
//Calculating the angular velocity of the link DE
omegaDE=vED/DE //rad/s
//Calculating the velocity of sliding of the link DE
    in the swivel block
vS=vQS //m/s
//Calculating the radial component of the
   acceleration of A with respect to O
arAO=vAO^2/OA //m/s^2
//Calculating the radial component of the
   acceleration of B with respect to A
arBA=vBA^2/AB //m/s^2
//Calculating the radial component of the
   acceleration of B with respect to C
arBC=vBC^2/CB //m/s^2
//Calculating the radial component of the
   acceleration of S with respect to D
arSD=vSD^2/DS //m/s^2
//Calculating the coriolis component of the
   acceleration of Q with respect to S
acQS=2*omegaDE*vQS //m/s^2
//Calculating the radial component of the
   acceleration of F with respect to E
```

```
arFE=vFE^2/EF //m/s^2
//By measurement from the acceleration diagram, Fig.
    8.37(d),
arQS=1.55 //m/s^2
//Results:
printf("\n\n Velocity of the slider block F, vF = %
    .2f m/s.\n",vF)
printf(" Angular velocity of the link DE, omegaDE =
    %.2f rad/s, anticlockwise.\n",omegaDE)
printf(" Velocity of sliding of the link DE in the
    swivel block, vS = %.1f m/s.\n",vS)
printf(" Acceleration of sliding of the link DE in
    the trunnion, arQS = %.2f m/s^2.\n\n",arQS)
```

# Chapter 9

# Mechanisms with Lower Pairs

**Scilab code Exa 9.1** To find inclination of track arm

```
//To find inclination of track arm
clc
//Given:
c=1.2,b=2.7 //m
//Solution:
//Calculating the inclination of the track arm to
   the longitudinal axis
alpha=atan(c/(2*b))*180/%pi //degrees
//Results:
printf("\n\n Inclination of the track arm to the
   longitudinal axis, alpha = %.1f degrees.\n\n",
   alpha)
```

**Scilab code Exa 9.3** To find the angle turned

```
//To find the angle turned
clc
//Given:
```

```
alpha=18*%pi/180 //radians
//Solution:
//Maximum velocity is possible when
theta1=0,theta2=180 //degrees
//Calculating the angle turned by the driving shaft when the velocity ratio is unity
theta3=acos(sqrt((1-cos(alpha))/(sin(alpha)^2)))*180/%pi //degrees
theta4=180-theta3 //degrees
//Results:
printf("\n\n Angle turned by the driving shaft when the velocity ratio is maximum, theta = %d degrees or %d degrees.\n",theta1,theta2)
printf(" Angle turned by the driving shaft when the velocity ratio is unity, theta = %.1f degrees or %.1f degrees.\n\n",theta3,theta4)
```

**Scilab code Exa 9.4** To find the greatest permissible angle

```
//To find the greatest permissible angle
clc
//Given:
N=500 //rpm
//Solution:
//Calculating the angular velocity of the driving shaft
omega=2*%pi*N/60 //rad/s
//Calculating the total fluctuation of speed of the driven shaft
q=12/100*omega //rad/s
//Calculating the greatest permissible angle between the centre lines of the shafts
alpha=acos((-(q/omega)+sqrt(0.12^2+4))/2)*180/%pi //degrees
//Results:
```

```
printf("\n\n Greatest permissible angle between the centre lines of the shafts, alpha = %.2f degrees.\n\n",alpha)
```

**Scilab code Exa 9.5** To find speeds and permissible angle

```
//To find speeds and permissible angle
clc
//Given:
N=1200,q=100 //rpm
//Solution:
//Calculating the greatest permissible angle between the centre lines of the shafts
alpha=acos((-(100/1200)+sqrt(0.083^2+4))/2)*180/%pi //degrees
//Calculating the maximum speed of the driven shaft
N1max=N/cosd(alpha) //rpm
//Calculating the minimum speed of the driven shaft
N1min=N*cosd(alpha) //rpm
//Results:
printf("\n\n Greatest permissible angle between the centre lines of the shafts, alpha = %.1f degrees.\n",alpha)
printf(" Maximum speed of the driven shaft, N1(max) = %d rpm.\n",N1max)
printf(" Minimum speed of the driven shaft, N1(min) = %d rpm.\n\n",N1min)
```

**Scilab code Exa 9.7** To find speeds of shafts

```
//To find speeds of shafts
clc
//Given:
```

```
alpha=20 //degrees
NA=500 //rpm
//Solution:
//Calculating the maximum speed of the intermediate shaft
NBmax=NA/cosd(alpha) //rpm
//Calculating the minimum speed of the intermediate shaft
NBmin=NA*cosd(alpha) //rpm
//Calculating the maximum speed of the driven shaft
NCmax=NBmax/cosd(alpha) //rpm
//Calculating the minimum speed of the driven shaft
NCmin=NBmin*cosd(alpha) //rpm
//Results:
printf("\n\n Maximum speed of the intermediate shaft, NB(max) = %.1f rad/s.\n",NBmax)
printf(" Minimum speed of the intermediate shaft, NB(min) = %.2f rad/s.\n",NBmin)
printf(" Maximum speed of the driven shaft, NC(max) = %.2f rad/s.\n",NCmax)
printf(" Minimum speed of the driven shaft, NC(min) = %.1f rad/s.\n",NCmin)
```

# Chapter 10

# Friction

**Scilab code Exa 10.1** To find weight and coefficient of friction

```
//To find weight and coefficient of friction
clc
//Given:
theta=30 //degrees
P1=180 //Pulling force, N
P2=220 //Pushing force, N
//Solution:
//Resolving the forces horizontally for the pull of 180N
F1=P1*cosd(theta) //N
//Resolving the forces for the push of 220 N
F2=P2*cosd(theta) //N
//Calculating the coefficient of friction
//For the pull of 180N, F1=mu*W-90*mu, or F1/mu-W=-90 .....(i)
//For the push of 220N, F2=W*mu+110*mu, or F2/mu-W=110 .....(ii)
A=[F1 -1; F2 -1]
B=[-90; 110]
V=A \ B
mu=1/V(1)
```

```
W=V(2)
//Results:
printf("\n\n The weight of the body, W = %d N.\n",W)
printf(" The coefficient of friction, mu = %.4f.\n\n
    ",mu)
```

**Scilab code Exa 10.2** To find weight and coefficient of friction

```
//To find weight and coefficient of friction
clc
//Given:
P1=1500,P2=1720 //N
alpha1=12,alpha2=15 //degrees
//Solution:
//Refer Fig. 10.10
//Effort applied parallel to the plane, P1=W*(sind(
    alpha1)+mu*cosd(alpha1)), or P1/W-mu*cosd(alpha1)
    =sind(alpha1)      .....(i)
//Effort applied parallel to the plane, P2=W*(sind(
    alpha2)+mu*cosd(alpha2)), or P2/W-mu*cosd(alpha2)
    =sind(alpha2)      .....(ii)
A=[P1 -cosd(alpha1); P2 -cosd(alpha2)]
B=[sind(alpha1); sind(alpha2)]
V=A \ B
W=1/V(1)
mu=V(2)
//Results:
printf("\n\n Coefficient of friction, mu = %.3f.\n",
    mu)
printf(" Weight of the body, W = %d N.\n\n",W)
```

**Scilab code Exa 10.3** To estimate the power

```
//To estimate the power
clc
//Given:
W=75*1000 //W
v=300 //mm/min
p=6,d0=40 //mm
mu=0.1
//Solution:
//Calculating the mean diameter of the screw
d=(d0-p/2)/1000 //m
//Calculating the helix angle
alpha=atan(p/(%pi*d*1000)) //radians
//Calculating the force required at the
   circumference of the screw
phi=atan(mu) //Limiting angle of friction, radians
P=W*tan(alpha+phi) //N
//Calculating the torque required to overcome the
   friction
T=P*d/2 //N-m
//Calculating the speed of the screw
N=v/p //rpm
//Calculating the angular speed
omega=2*%pi*N/60 //rad/s
//Calculating the power of the motor
Power=T*omega/1000 //Power of the motor, kW
//Results:
printf("\n\n Power of the motor required = %.3f kW.\
   n\n",Power)
```

**Scilab code Exa 10.4** To find work done

```
//To find work done
clc
//Given:
p=12,d=40 //mm
```

```
mu=0.16
W=2500 //N
//Solutiom:
//Work done in drawing the wagons together agianst a steady load of 2500 N:
//Calculating the helix angle
alpha=atan(p/(%pi*d)) //radians
//Calculating the effort required at the circumference of the screw
phi=atan(mu) //Limiting angle of friction, radians
P=W*tan(alpha+phi) //N
//Calculating the torque required to overcome friction between the screw and nut
T=P*d/(2*1000) //N-m
//Calculating the number of turns required
N=240/(2*p)
//Calculating the work done
W1=T*2*%pi*N //Work done, N-m
//Work done in drawing the wagons together when the load increases from 2500 N to 6000 N:
W2=W1*(6000-2500)/2500 //Work done, N-m
//Results:
printf("\n\n Work done in drawing the wagons together agianst a steady load of 2500 N = %.1f N-m.\n",W1)
printf(" Work done in drawing the wagons together when the load increases from 2500 N to 6000 N = %.1f N-m.\n\n",W2)
```

**Scilab code Exa 10.5** To find the torque required

```
//To find the torque required
clc
//Given:
D=150/1000 //m
```

```
ps=2*10^6 //N/m^2
d0=50,p=6 //mm
mu=0.12
//Solution:
//Calculating the load on the valve
W=ps*%pi/4*D^2 //N
//Calculating the mean diameter of the screw
d=(d0-p/2)/1000 //m
//Calculating the helix angle
alpha=atan(p/(%pi*d*1000))
//Calculating the force required to turn the handle
phi=atan(mu) //Limiting angle of friction, radians
P=W*tan(alpha+phi) //N
//Calculating the torque required to turn the handle
T=P*d/2 //N-m
//Results:
printf("\n\n The torque required to turn the handle,
    T = %.1f N-m.\n\n",T)
```

**Scilab code Exa 10.6** To find force required

```
//To find force required
clc
//Given:
dc=22.5,p=5,D=50,R=D/2,l=500 //mm
mu=0.1,mu1=0.16
W=10*1000 //N
//Solution:
//Calculating the mean diameter of the screw
d=dc+p/2 //mm
//Calculating the helix angle
alpha=p/(%pi*d) //radians
//Calculating the force required at the
    circumference of the screw
phi=atan(mu) //Limiting angle of friction, radians
```

```
P=W*tan(alpha+phi) //N
//Calculating the total torque required
T=P*d/2+mu1*W*R //N-mm
//Calculating the force required at the end of a
    spanner
P1=T/l //N
//Results:
printf("\n\n Force required at the end of a spanner,
     P1 = %.2f N.\n\n",P1)
```

**Scilab code Exa 10.7** To find diameter of hand wheel

```
//To find diameter of hand wheel
clc
//Given:
d=50,p=12.5,D=60,R=D/2 //mm
W=10*1000,P1=100 //N
mu=0.15,mu1=0.18
//Solution:
//Calculating the helix angle
alpha=atan(p/(%pi*d)) //radians
//Calculating the tangential force required at the
    circumference of the screw
phi=atan(mu) //Limiting angle of friction, radians
P=W*tan(alpha+phi) //N
//Calculating the total torque required to turn the
    hand wheel
T=P*d/2+mu1*W*R //N-mm
//Calculating the diameter of the hand wheel
D1=T/(2*P1*1000)*2 //m
//Results:
printf("\n\n Diameter of the hand wheel, D1 = %.3f m
    .\n\n",D1)
```

**Scilab code Exa 10.8** To find the power required

```
//To find the power required
clc
//Given:
d0=55,D2=60,R2=D2/2,D1=90,R1=D1/2 //mm
p=10/1000 //m
W=400 //N
mu=0.15
v=6 //Cutting speed, m/min
//Solution:
//Calculating the mean diameter of the screw
d=d0-p/2 //mm
//Calculating the helix angle
alpha=p/(%pi*d) //radians
//Calculating the force required at the
    circumference of the screw
phi=atan(mu) //Limiting angle of friction, radians
P=W*tan(alpha+phi) //N
//Calculating the mean radius of the flat surface
R=(R1+R2)/2 //mm
//Calculating the torque required
T=(P*d/2+mu1*W*R)/1000 //N-m
//Calculating the speed of the screw
N=v/p //rpm
//Calculating the angular speed
omega=2*%pi*N/60 //rad/s
//Calculating the power required to operate the nut
Power=T*omega/1000 //Power required to operate the
    nut, kW
//Results:
printf("\n\n Power required to operate the nut = %.3
    f kW.\n\n",Power)
```

**Scilab code Exa 10.9** To find the force applied

```
//To find the force applied
clc
//Given:
d=50/1000,l=0.7 //m
p=10 //mm
mu=0.15
W=20*1000 //N
//Solution:
//Calculating the helix angle
alpha=atan(p/(%pi*d*1000)) //radians
//Force required to raise the load:
//Calculating the force required at the circumference of the screw
phi=atan(mu) //Limiting angle of friction, radians
P1=W*tan(alpha+phi) //N
//Calculating the force required at the end of the lever
P11=P1*d/(2*l) //N
//Calculating the force required at the circumference of the screw
P2=W*(phi-alpha) //N
//Foce rewuired to lower the load:
//Calculating the force required at the end of the lever
P21=P2*d/(2*l) //N
//Results:
printf("\n\n Force required at the end of the lever to raise the load, P1 = %d N.\n",P11)
printf(" Force required at the end of the lever to lower the load, P1 = %d N.\n\n",P21)
```

**Scilab code Exa 10.10** To find ratio of torques and efficiency

```
//To find ratio of torques and efficiency
clc
//Given:
d=50,p=12.5 //mm
mu=0.13
W=25*1000 //N
//Solution:
//Calculating the helix angle
alpha=atan(p/(%pi*d)) //radians
//Calculating the force required on the screw to raise the load
phi=atan(mu) //Limiting angle of friction, radians
P1=W*(alpha+phi) //N
//Calculating the torque required on the screw to raise the load
T1=P1*d/2 //N-mm
//Calculating the force required on the screw to lower the load
P2=W*tan(phi-alpha) //N
//Calculating the torque required to lower the load
T2=P2*d/2 //N
//Calculating the ratio of the torques required
r=T1/T2 //Ratio of the torques required, N-mm
//Calculating the efficiency of the machine
eta=tan(alpha)/tan(alpha+phi)*100 //%
//Results:
printf("\n\n Torque required on the screw to raise the load, T1 = %d N-mm.\n",T1)
printf(" Ratio of the torque required to raise the load to the torque required to lower the load = %.1f.\n",r)
printf(" Efficiency of the machine, eta = %.1f %c.\n
```

```
\n",eta,"%")
```

**Scilab code Exa 10.11** To find work done and efficiency

```
//To find work done and efficiency
clc
//Given:
p=10,d=50,D2=60,R2=D2/2,D1=10,R1=D1/2 //mm
W=20*1000 //N
mu=0.08,mu1=mu
//Solution:
//Calculating the helix angle
alpha=atan(p/(%pi*d)) //radians
//Calculating the force required at the circumference of the screw to lift the load
phi=atan(mu) //Limiting angle of friction, radians
P=W*tan(alpha+phi) //N
//Calculating the torque required to overcome friction at the screw
T=P*d/(2*1000) //N-m
//Calculating the number of rotations made by the screw
N=170/p
//When the load rotates with the screw:
//Calculating the work done in lifting the load
W1=T*2*%pi*N //Work done in lifting the load, N-m
//Calculating the efficiency of the screw jack
eta1=tan(alpha)/tan(alpha+phi)*100 //%
//When the load does not rotate with the screw:
//Calculating the mean radius of the bearing surface
R=(R1+R2)/2 //mm
//Calculating the torque required to overcome friction at the screw and the collar
T=(P*d/2+mu1*W*R)/1000 //N-m
//Calculating the work done by the torque in lifting
```

```
 the load
W2=T*2*%pi*N //Work done by the torque in lifting the load , N-m
//Calculating the torque required to lift the load , neglecting frition
T0=(W*tan(alpha)*d/2)/1000 //N-m
//Calculating the efficiency of the screw jack
eta2=T0/T*100 //%
//Results:
printf("\n\n When the load rotates with the screw , work done in lifting the load = %d N-m.\n",W1)
printf(" Efficiency of the screw jack , eta = %.1f %c.\n",eta1,"%")
printf(" When the load does not rotate with the screw , work done in lifting the load = %d N-m.\n",W2)
printf(" Efficiency of the screw jack , eta = %.1f %c.\n\n",eta2,"%")
```

**Scilab code Exa 10.12** To find length of lever

```
//To find length of lever
clc
//Given:
W=10*1000,P1=100 //N
p=12,d=50 //mm
mu=0.15
//Solution:
//Calculating the helix angle
alpha=atan(p/(%pi*d)) //radians
//Calculating the effort required at the circumference of the screw to raise the load
phi=atan(mu) //Limiting angle of friction , radians
P=W*tan(alpha+phi) //N
//Calculating the torque required to overcome
```

```
    friction
T=P*d/2 //N-mm
//Calculating the length of the lever
l=T/P1 //mm
//Calculating the mechanical advantage
MA=W/P1
//Calculating the efficiency of the screw jack
eta=tan(alpha)/tan(alpha+phi)*100 //%
//Results:
printf("\n\n The length of the lever to be used, l =
    %.1f mm.\n",l)
printf(" Mechanical advantage obtained, M.A. = %d.\n
    ",MA)
if eta<50 then
    printf(" The screw is a self locking screw.\n\n"
        );
else
    printf(" The screw is not a self locking screw."
        );
end
```

**Scilab code Exa 10.13** To find the torque required

```
//To find the torque required
clc
//Given:
d=22,p=3 //mm
funcprot(0)
beta=60/2 //degrees
W=40*1000 //N
mu=0.15
//Solution:
//Calculating the helix angle
alpha=atan(p/(%pi*d)) //radians
//Calculating the virtual coefficient of friction
```

```
mu1=mu/cosd(beta)
//Calculating the force required at the
   circumference of the screw
phi1=atan(mu1) //Virtual limiting angle of friction ,
    radians
P=W*tan(alpha+phi1)
//Calculating the torque on one rod
T=P*d/(2*1000) //N-m
//Calculating the torque required on the nut
T1=2*T //N-m
//Results:
printf("\n\n The torque required on the nut , T1 = %
   .2f N-m.\n\n",T1)
```

**Scilab code Exa 10.14** To find the forcr

```
//To find the forcr
clc
//Given:
d=25,p=5,R=25 //mm
funcprot(0)
beta=27.5 //degrees
mu=0.1,mu2=0.16
l=0.5 //m
W=10*1000 //N
//Solution:
//Calculating the virtual coefficient of friction
mu1=mu/cosd(beta)
//Calculating the helix angle
alpha=atan(p/(%pi*d)) //radians
//Calculating the force on the screw
phi1=atan(mu1) //Virtual limiting angle of frcition ,
    radians
P=W*tan(alpha+phi1) //N
//Calculating the total torque transmitted
```

```
T=(P*d/2+mu2*W*R)/1000 //N-m
//Calculating the force required at the end of a
   spanner
P1=T/l //N
//Results:
printf("\n\n Force required at the end of a spanner,
    P1 = %.1f N.\n\n",P1)
```

**Scilab code Exa 10.15** To find power transmitted

```
//To find power transmitted
clc
//Given:
d=60,r=d/2 //mm
W=2000 //N
mu=0.03
N=1440 //rpm
//Solution:
//Calculating the angular speed of the shaft
omega=2*%pi*N/60 //rad/s
//Calculating the torque transmitted
T=mu*W*(r/1000) //N-m
//Calculating the power transmitted
P=T*omega //W
//Results:
printf("\n\n The power transmitted, P = %.1f W.\n\n"
   ,P)
```

**Scilab code Exa 10.16** To estimate power lost in friction

```
//To estimate power lost in friction
clc
//Given:
```

```
D=150/1000,R=D/2 //m
N=100 //rpm
W=20*1000 //N
mu=0.05
//Solution:
//Calculating the angular speed of the shaft
omega=2*%pi*N/60 //rad/s
//Calculating the total frictional torque for
    uniform pressure distribution
T=2/3*mu*W*R //N-m
//Calculating the power lost in friction
P=T*omega //W
//Results:
printf("\n\n Power lost in friction, P = %.1f W.\n\n
    ",P)
```

---

**Scilab code Exa 10.17** To find power absorbed in friction

```
//To find power absorbed in friction
clc
//Given:
W=20*1000 //N
alpha=120/2 //degrees
Pn=0.3 //N/mm^2
N=200 //rpm
mu=0.1
//Solution:
//Calculating the angular speed of the shaft
omega=2*%pi*N/60 //rad/s
//Calculating the inner radius of the bearing
    surface
r2=sqrt(W/(3*%pi*Pn)) //mm
//Calculating the outer radius of the bearing
    surface
r1=2*r2 //mm
```

```
//Calculating the total frictional torque assuming uniform pressure
T=2/3*mu*W*(1/sind(alpha))*(r1^3-r2^3)/(r1^2-r2^2)/1000 //N-m
//Calculating the power absorbed in friction
P=T*omega/1000 //kW
//Results:
printf("\n\n Power absorbed in friction, P = %.3f kW.\n\n",P)
```

**Scilab code Exa 10.18** To find power lost in friction

```
//To find power lost in friction
clc
//Given:
D=200/1000,R=D/2 //m
W=30*1000 //N
alpha=120/2 //degrees
mu=0.025
N=140 //rpm
//Solution:
//Calculating the angular speed of the shaft
omega=2*%pi*N/60 //rad/s
//Power lost in friction assuming uniform pressure:
//Calculating the total frictional torque
T=2/3*mu*W*R*(1/sind(alpha)) //N-m
//Calculating the power lost in friction
P1=T*omega //Power lost in friction, W
//Power lost in friction assuming uniform wear:
//Calculating the total frictional torque
T=1/2*mu*W*R*(1/sind(alpha)) //N-m
//Calculating the power lost in friction
P2=T*omega //Power lost in friction, W
//Resluts:
printf("\n\n Power lost in friction assuming uniform
```

```
    pressure , P = %d W.\n",P1)
printf(" Power lost in friction assuming uniform
   wear , P = %.1f W.\n\n",P2)
```

**Scilab code Exa 10.19** To find power absorbed in friction

```
//To find power absorbed in friction
clc
//Given:
n=6
d1=600,r1=d1/2,d2=300,r2=d2/2 //mm
W=100*1000 //N
mu=0.12
N=90 //rpm
//Solution:
//Calculating the angular speed of the engine
omega=2*%pi*N/60 //rad/s
//Power absorbed in friction assuming uniform
   pressure:
//Calculating the total frictional torque
   transmitted
T=2/3*mu*W*(r1^3-r2^3)/(r1^2-r2^2)/1000 //N-m
//Calculating the power absorbed in friction
P1=T*omega/1000 //Power absorbed in friction
   assuming uniform pressure, kW
//Power absorbed in friction assuming uniform wear:
//Calculating the total frictional torque
   transmitted
T=1/2*mu*W*(r1+r2)/1000 //N-m
//Calculating the power absorbed in friction
P2=T*omega/1000 //Power absorbed in friction
   assuming uniform wear, kW
//Results:
printf("\n\n Power absorbed in friction assuming
   uniform pressure, P = %.1f kW.\n",P1)
```

```
printf(" Power absorbed in friction assuming uniform
    wear, P = %.2f kW.\n\n",P2)
```

**Scilab code Exa 10.20** To find power absorbed

```
//To find power absorbed
clc
//Given:
d1=400,r1=d1/2,d2=250,r2=d2/2 //mm
p=0.35 //N/mm^2
mu=0.05
N=105 //rpm
W=150*1000 //N
//Solution:
//Calculating the angular speed of the shaft
omega=2*%pi*N/60 //rad/s
//Calculating the total frictional torque
   transmitted for uniform pressure
T=2/3*mu*W*(r1^3-r2^3)/(r1^2-r2^2)/1000 //N-m
//Calculating the power absorbed
P=T*omega/1000 //kW
//Calculating the number of collars required
n=W/(p*%pi*(r1^2-r2^2))
//Results:
printf("\n\n Power absorbed, P = %.2f kW.\n",P)
printf(" Number of collars required, n = %d.\n\n",n
   +1)
```

**Scilab code Exa 10.21** To find diameter and number of collars

```
//To find diameter and number of collars
clc
//Given:
```

```
d2=300/1000,r2=d2/2 //m
W=200*1000 //N
N=75 //rpm
mu=0.05
p=0.3 //N/mm^2
P=16*1000 //W
//Solution:
//Calculating the angular velocity of the shaft
omega=2*%pi*N/60 //rad/s
//Calculating the total frictional torque
    transmitted
T=P/omega //N-m
//Calculating the external diameter of the collar
//We have, T=2/3*mu*W*(r1^3-r2^3)/(r1^2-r2^2), or
    (2*mu*W)*r1^2-(3*T-2*mu*W*r2)*r1+(2*mu*W*r2^2-3*T
    *r2)=0
A=2*mu*W, B=-(3*T-2*mu*W*r2), C=2*mu*W*r2^2-3*T*r2
r1=(-B+sqrt(B^2-4*A*C))/(2*A)*1000 //mm
d1=2*r1 //mm
//Calculating the number of collars
n=W/(p*%pi*(r1^2-(r2*1000)^2))
//Results:
printf("\n\n External diameter of the collar, d1 =
    %d mm.\n",d1)
printf(" Number of collars, n = %d.\n\n",n+1)
```

**Scilab code Exa 10.22** To find the pressure

```
//To find the pressure
clc
//Given:
W=4*1000 //N
r2=50,r1=100 //mm
//Solution:
//Calculating the maximum pressure
```

```
pmax=W/(2*%pi*r2*(r1-r2)) //N/mm^2
//Calculating the minimum pressure
pmin=W/(2*%pi*r1*(r1-r2)) //N/mm^2
//Calculating the average pressure
pav=W/(%pi*(r1^2-r2^2)) //N/mm^2
//Results:
printf("\n\n Maximum pressure, pmax = %.4f N/mm^2.\n
   ",pmax)
printf(" Minimum pressure, pmin = %.4f N/mm^2.\n",
   pmin)
printf(" Average pressure, pav = %.2f N/mm^2.\n\n",
   pav)
```

---

**Scilab code Exa 10.23** To find power transmitted

```
//To find power transmitted
clc
//Given:
d1=300, r1=d1/2, d2=200, r2=d2/2 //mm
p=0.1 //N/mm^2
mu=0.3
N=2500 //rpm
n=2
//Solution:
//Calculating the radial speed of the clutch
omega=2*%pi*N/60 //rad/s
//Calculating the intensity of pressure
C=p*r2 //N/mm
//Calculating the axial thrust
W=2*%pi*C*(r1-r2) //N
//Calculating the mean radius of the friction
   surfaces for uniform wear
R=(r1+r2)/(2*1000) //m
//Calculating the torque transmitted
T=n*mu*W*R //N-m
```

```
//Calculating the power transmitted by a clutch
P=T*omega/1000 //kW
//Results:
printf("\n\n Power transmitted by a clutch, P = %.3f kW.\n\n",P)
```

**Scilab code Exa 10.24** To find radii and axial thrust

```
//To find radii and axial thrust
clc
//Given:
n=2, mu=0.255
P=25*1000 //W
N=3000 //rpm
r=1.25 //Ratio of radii, r1/r2
p=0.1 //N/mm^2
//Solution:
//Calculating the angular speed of the clutch
omega = 2*%pi*N/60 //rad/s
//Calculating the torque transmitted
T=P/omega*1000 //N-mm
//Calculating the inner radius
r2=(T/(n*mu*2*%pi*0.1*(1.25-1)*(1.25+1)/2))^(1/3) //mm
//Calculating the outer radius
r1=r*r2 //mm
//Calculating the axial thrust to be provided by springs
C=0.1*r2 //Intensity of pressure, N/mm
W=2*%pi*C*(r1-r2) //N
//Results:
printf("\n\n Outer radius of the frictional surface, r1 = %d mm.\n",r1)
printf(" Inner radius of the frictional surface, r2 = %d mm.\n",r2)
```

```
printf(" Axial thrust to be provided by springs , W = %d N.\n\n",W)
```

**Scilab code Exa 10.25** To find dimensions of clutch plate

```
//To find dimensions of clutch plate
clc
//Given:
P=7.5*1000 //W
N=900 //rpm
p=0.07 //N/mm^2
mu=0.25
n=2
//Solution:
//Calculating the angular speed of the clutch
omega=2*%pi*N/60 //rad/s
//Calculating the torque transmitted
T=P/omega*1000 //N-mm
//Calculating the mean radius of the friction lining
R=(T/(%pi/2*n*mu*p))^(1/3) //mm
//Calculating the face width of the friction lining
w=R/4 //mm
//Calculating the outer and inner radii of the clutch plate
//We have, w = r1-r2, or r1-r2 = w .....(i)
//Also, R = (r1+r2)/2, or r1+r2 = 2*R .....(ii)
A=[1 -1; 1 1]
B=[w; 2*R]
V=A \ B
r1=V(1)
r2=V(2)
//Results:
printf("\n\n Mean radius of the friction lining , R =
```

```
    %d mm.\n",R)
printf(" Face width of the friction lining, w = %.2f
    mm.\n",w)
printf(" Outer radius of the clutch plate, r1 = %.3f
    mm.\n",r1)
printf(" Inner radius of the clutch plate, r2 = %.3f
    mm.\n\n",r2)
```

**Scilab code Exa 10.26** To find dimensions of clutch plate

```
//To find dimensions of clutch plate
clc
//Given:
P=100 //kW
N=2400 //rpm
T=500*1000 //N-mm
p=0.07 //N/mm^2
mu=0.3
Ns=8 //Number of springs
k=40 //Stiffness, N/mm
n=2
//Solution:
//Calculating the inner radius of the friction plate
r2=(T/(n*mu*2*%pi*p*(1.25-1)*(1.25+1)/2))^(1/3) //mm
//Calculating the outer radius of the friction plate
r1=1.25*r2 //mm
//Calculating the total stiffness of the springs
s=k*Ns //N/mm
//Calculating the intensity of pressure
C=p*r2 //N/mm
//Calculating the axial force required to engage the
    clutch
W=2*%pi*C*(r1-r2) //N
//Calculating the initial compression in the springs
IC=W/s //Initial compression in the springs, mm
```

```
//Results:
printf("\n\n Outer radius of the friction plate, r1 = %.1f mm.\n",r1)
printf(" Inner radius of the friction plate, r2 = %d mm.\n",r2)
printf(" Initial compression in the springs = %.1f mm.\n\n",IC)
```

**Scilab code Exa 10.27** To find speed time and KE lost

```
//To find speed, time and KE lost
clc
//Given:
d1=220, r1=d1/2, d2=160, r2=d2/2 //mm
W=570 //N
m1=800, m2=1300 //kg
k1=200/1000, k2=180/1000 //m
mu=0.35
N1=1250 //rpm
n=2
//Solution:
//Calculating the initial angular speed of the motor shaft
omega1=2*%pi*N1/60 //rad/s
//Calculating the moment of inertia for the motor armature and shaft
I1=m1*k1^2 //kg-m^2
//Calculating the moment of inertia for the rotor
I2=m2*k2^2 //kg-m^2
//Calculating the final speed of the motor and rotor
omega2=0
omega3=(I1*omega1+I2*omega2)/(I1+I2) //rad/s
//Calculating the mean radius of the friction plate
R=(r1+r2)/(2*1000) //m
//Calculating the frictional torque
```

```
T=n*mu*W*R //N-m
//Calculating the angular acceleration of the rotor
alpha2=T/I2 //rad/s^2
//Calculating the time to reach the speed of omega3
omegaF=omega3, omegaI=omega2
t=(omegaF-omegaI)/alpha2 //seconds
//Calculating the angular kinetic energy before impact
E1=1/2*I1*omega1^2+1/2*I2*omega2^2 //N-m
//Calculating the angular kinetic energy after impact
E2=1/2*(I1+I2)*omega3^2 //N-m
//Calculating the kinetic energy lost during the period of slipping
E=E1-E2 //N-m
//Calculating the torque on armature shaft
T1=-60-T //N-m
//Calculating the torque on rotor shaft
T2=T //N-m
//Calculating the time of slipping assuming constant resisting torque:
//Considering armature shaft, omega3 = omega1+alpha1*t1, or omega3-(T1/I1)*t1 = omega1          .....(i)
//Considering rotor shaft, omega3 = alpha2*t1, or omega3-(T2/I2)*t1 = 0          .....(ii)
A=[1 -T1/I1; 1 -T2/I2]
B=[omega1; 0]
V=A \ B
t11=V(2) //Time of slipping assuming constant resisting torque, seconds
//Calculating the time of slipping assuming constant driving torque:
//Calculating the torque on armature shaft
T1=60-T //N-m
t12=(omega2-omega1)/(T1/I1-T2/I2) //Time of slipping assuming constant driving torque, seconds
```

```
//Results:
printf("\n\n Final speed of the motor and rotor, omega3 = %.2f rad/s.\n",omega3)
printf(" Time to reach the speed of %.2f rad/s, t = %.1f s.\n",omega3,t)
printf(" Kinetic energy lost during the period of slipping = %d N-m.\n",E)
printf(" Time of slipping assuming constant resisting torque, t1 = %.1f s.\n",t11)
printf(" Time of slipping assuming constant driving torque, t1 = %d s.\n\n",t12)
```

**Scilab code Exa 10.28** To find the power transmitted

```
//To find the power transmitted
clc
//Given:
n=4, mu=0.3
p=0.127 //N/mm^2
N=500 //rpm
r1=125, r2=75 //mm
//Solution:
//Calculating the angular speed of the clutch
omega=2*%pi*N/60 //rad/s
//Calculating the maximum intensity of pressure
C=p*r2 //N/mm
//Calculating the axial force required to engage the clutch
W=2*%pi*C*(r1-r2) //N
//Calculating the mean radius of the friction surfaces
R=(r1+r2)/(2*1000) //m
//Calculating the torque transmitted
T=n*mu*W*R //N-m
//Calculating the power transmitted
```

```
P=T*omega/1000 //kW
//Results:
printf("\n\n Power transmitted, P = %.1f kW.\n\n",P)
```

**Scilab code Exa 10.29** To find maximum intensity of pressure

```
//To find maximum intensity of pressure
clc
//Given:
n1=3, n2=2, mu=0.3
d1=240, r1=d1/2, d2=120, r2=d2/2 //mm
P=25*1000 //W
N=1575 //rpm
//Solution:
//Calculating the angular speed of the shaft
omega=2*%pi*N/60 //rad/s
//Calculating the torque transmitted
T=P/omega //N-m
//Calculating the number of pairs of friction
    surfaces
n=n1+n2-1
//Calculating the mean radius of friction surfaces
    for uniform wear
R=(r1+r2)/(2*1000) //m
//Calculating the axial force on each friction
    surface
W=T/(n*mu*R) //N
//Calculating the maximum axial intensity of
    pressure
p=W/(2*%pi*r2*(r1-r2)) //N/mm^2
//Results:
printf("\n\n Maximum axial intensity of pressure, p
    = %.3f N/mm^2.\n\n",p)
```

**Scilab code Exa 10.30** To find maximum power transmitted

```
//To find maximum power transmitted
clc
//Given:
n1=3, n2=2, n=4, mu=0.3
d1=240, r1=d1/2, d2=120, r2=d2/2 //mm
P=25*1000 //W
N=1575 //rpm
//Solution:
//Calculating the angular speed of the shaft
omega=2*%pi*N/60 //rad/s
//Calculating the torque transmitted
T=P/omega //N-m
//Calculating the mean radius of the contact surface
    , for uniform pressure
R=2/3*(r1^3-r2^3)/(r1^2-r2^2)/1000 //m
//Calculating the total spring load
W1=T/(n*mu*R) //N
//Calculating the maximum power transmitted:
//Given:
ns=6 //Number of springs
c=8 //Contact surfaces of the spring
w=1.25 //Wear on each contact surface, mm
k=13*1000 //Stiffness of each spring, N/m
//Calculating the total wear
Tw=c*w/1000 //Total wear, m
//Calculating the reduction in spring force
Rs=Tw*k*ns //N
//Calculating the new axial load
W2=W1-Rs //N
//Calculating the mean radius of the contact
    surfaces for uniform wear
R=(r1+r2)/(2*1000) //m
```

```
//Calculating the torque transmitted
T=n*mu*W2*R //N-m
//Calculating the maximum power transmitted
P=T*omega/1000 //kw
//Results:
printf("\n\n Total spring load, W = %d N.\n",W1)
printf(" Maximum power that can be transmitted, P = %.2f kW.\n\n",P)
```

**Scilab code Exa 10.31** To find dimensions and axial load

```
//To find dimensions and axial load
clc
//Given:
P=90*1000 //W
N=1500 //rpm
alpha=20 //degrees
mu=0.2
D=375, R=D/2 //mm
pn=0.25 //N/mm^2
//SOlution:
//Calculating the angular speed of the clutch
omega=2*%pi*N/60 //rad/s
//Calculating the torque transmitted
T=P/omega*1000 //N-mm
//Calculating the width of the bearing surface
b=T/(2*%pi*mu*pn*R^2) //mm
//Calculating the external and internal radii of the bearing surface
//We know that, r1+r2 = 2*R, and r1-r2 = b*sind(alpha)
A=[1 1; 1 -1]
B=[2*R; b*sind(alpha)]
V=A \ B
r1=V(1) //mm
```

```
r2=V(2) //mm
//Calculating the intensity of pressure
C=pn*r2 //N/mm
//Calculating the axial load required
W=2*%pi*C*(r1-r2) //N
//Results:
printf("\n\n Width of the bearing surface, b = %.1f
    mm.\n",b)
printf(" External radius of the bearing surface, r1
    = %.1f mm.\n",r1)
printf(" Internal radius of the bearing surface, r2
    = %.1f mm.\n",r2)
printf(" Axial load required, W = %d N.\n\n",W)
```

**Scilab code Exa 10.32** To find axial force and face width

```
//To find axial force and face width
clc
//Given:
P=45*1000 //W
N=1000 //rpm
alpha=12.5 //degrees
D=500/1000, R=D/2 //m
mu=0.2
pn=0.1 //N/mm^2
//Solution:
//Calculating the angular speed of the shaft
omega=2*%pi*N/60 //rad/s
//Calculating the torque developed by the clutch
T=P/omega //N-m
//Calculating the normal load acting on the friction
    surface
Wn=T/(mu*R) //N
//Calculating the axial spring force necessary to
    engage the clutch
```

```
We=Wn*(sind(alpha)+mu*cosd(alpha)) //N
//Calculating the face width required
b=Wn/(pn*2*%pi*R*1000) //mm
//Results:
printf("\n\n Axial force necessary to engage the clutch , We = %d N.\n",We)
printf(" Face width required , b = %.1f mm.\n\n",b)
```

**Scilab code Exa 10.33** To find dimensions of contact surfaces

```
//To find dimensions of contact surfaces
clc
//Given:
alpha=30/2 //degrees
pn=0.35 //N/mm^2
P=22.5*1000 //W
N=2000 //rpm
mu=0.15
//Solution:
//Calculating the angular speed of the clutch
omega=2*%pi*N/60 //rad/s
//Calculating the torque transmitted by the clutch
T=P/omega*1000 //N-mm
//Calculating the mean radius of the contact surface
R=(T/(2*%pi*mu*pn/3))^(1/3) //mm
//Calculating the face width of the contact surface
b=R/3
//Calculating the outer and inner radii of the contact surface
//Refer Fig. 10.27
//We have, r1-r2 = b*sind(alpha), and r1+r2 = 2*R
A=[1 -1; 1 1]
B=[b*sind(alpha); 2*R]
V=A \ B
r1=V(1) //mm
```

```
r2=V(2) //mm
//Results:
printf("\n\n Mean radius of the contact surface, R = %d mm.\n",R)
printf(" Outer radius of the contact surface, r1 = %.2f mm.\n",r1)
printf(" Inner radius of the contact surface, r2 = %.2f mm.\n\n",r2)
```

---

**Scilab code Exa 10.34** To find time required and energy lost

```
//To find time required and energy lost
clc
//Given:
D=75/1000, R=D/2 //m
alpha=15 //degrees
mu=0.3
W=180 //N
NF=1000 //rpm
m=13.5 //kg
k=150/1000 //m
//Solution:
//Calculating the angular speed of the flywheel
omegaF=2*%pi*NF/60 //rad/s
//Calculating the torque required to produce slipping
T=mu*W*R*(1/sind(alpha)) //N-m
//Calculating the mass moment of inertia of the flywheel
IF=m*k^2 //kg-m^2
//Calculating the angular acceleration of the flywheel
alphaF=T/IF //rad/s^2
//Calculating the time required for the flywheel to attain full speed
```

```
tF=omegaF/alphaF //seconds
//Calculating the angle turned through by the motor and flywheel in time tF
theta=1/2*omegaF*tF //rad
//Calculating the energy lost in slipping of the clutch
E=T*theta //Energy lost in slipping of the clutch, N-m
//Results:
printf("\n\n Torque required to produce slipping, T = %.1f N-m.\n",T)
printf(" Time required for the flywheel to attain full speed, tF = %.1f s.\n",tF)
printf(" Energy lost in slipping of the clutch = %d N-m.\n\n",E)
```

**Scilab code Exa 10.35** To find mass and size of shoes

```
//To find mass and size of shoes
clc
//Given:
P=15*1000 //W
N=900 //rpm
n=4, mu=0.25
R=150/1000, r=120/1000 //m
theta=60 //degrees
p=0.1 //N/mm^2
//Solution:
//Calculating the angular speed of the clutch
omega=2*%pi*N/60 //rad/s
//Calculating the speed at which the engagement begins
omega1=3/4*omega //rad/s
//Calculating the torque transmitted at the running speed
```

```
T=P/omega //N-m
//Calculating the mass of the shoes
m=T/(n*mu*(omega^2*r-omega1^2*r)*R) //kg
//Calculating the contact length of shoes
l=(theta*%pi/180)*R*1000 //mm
//Calculating the centrifugal force acting on each
    shoe
Pc=m*omega^2*r //N
//Calculating the inward force on each shoe exerted
    by the spring
Ps=m*omega1^2*r //N
//Calculating the width of the shoes
b=(Pc-Ps)/(l*p) //mm
//Results:
printf("\n\n Mass of the shoes, m = %.2f kg.\n",m)
printf(" Width of the shoes, b = %.1f mm.\n\n",b)
```

**Scilab code Exa 10.36** To fnd power transmitted

```
//To fnd power transmitted
clc
//Given:
n=4, mu=0.3
c=5, r=160 //mm
S=500 //N
D=400/1000, R=D/2 //m
m=8 //kg
s=50 //N/mm
N=500 //rpm
//Solution:
//Calculating the angular speed of the clutch
omega=2*%pi*N/60 //rad/s
//Calculating the operating radius
r1=(r+c)/1000 //m
//Calculating the centrifugal force on each shoe
```

```
Pc=m*omega^2*r1 //N
//Calculating the inward force exerted by the spring
Ps=S+c*s //N
//Calculating the frictional force acting
    tangentially on each shoe
F=mu*(Pc-Ps) //N
//Calculating the total frictional torque
    transmitted by the clutch
T=n*F*R //N-m
//Calculating the power transmitted
P=T*omega/1000 //kW
//Results:
printf("\n\n Power transmitted, P = %.1f kW.\n\n",P)
```

# Chapter 11

# Belt Rope and Chain Drives

**Scilab code Exa 11.1** To find speed of shaft

```
//To find speed of shaft
clc
//Given:
N1=150 //rpm
d1=750, d2=450, d3=900, d4=150 //mm
//Solution:
//Calculating the speed of the dynamo shaft when
    there is no slip
N41=N1*(d1*d3)/(d2*d4) //rpm
//Calculating the speed of the dynamo shaft whne
    there is a slip of 2% at each drive
s1=2, s2=2 //%
N42=N1*(d1*d3)/(d2*d4)*(1-s1/100)*(1-s2/100) //rpm
//Results:
printf("\n\n Speed of the dynamo shaft when there is
     no slip, N4 = %d rpm.\n\n",N41)
printf(" Speed of the dynamo shaft when there is a
    slip of 2%c at each drive, N4 = %d rpm.\n\n","%",
    N42)
```

**Scilab code Exa 11.2** To find speed lost

```
//To find speed lost
clc
//Given:
d1=1, d2=2.25 //m
N1=200 //rpm
sigma1=1.4*10^6, sigma2=0.5*10^6, E=100*10^6 //N/m^2
//Solution:
//Calculating the speed of the driven pulley
N21=N1*(d1/d2) //rpm
//Calculating the speed of the shaft considering
    creep
N22=N1*(d1/d2)*(E+sqrt(sigma2))/(E+sqrt(sigma1)) //
    rpm
//Calculating the speed lost by the driven pulley
    due to creep
Nl=N21-N22 //Speed lost by the driven pulley due to
    creep, rpm
//Results:
printf("\n\n Speed lost by the driven pulley due to
    creep = %.4f rpm.\n\n",Nl)
```

---

**Scilab code Exa 11.3** To find radii of stepped pulleys

```
//To find radii of stepped pulleys
clc
//Given:
N1=160, N3=N1, N5=N3, N2=60, N4=80, N6=100 //rpm
x=720, r1=40 //mm
//Solution:
//For a crossed belt:
```

```
//Calcluating the radii of pulleys 2, 3, 4, 5, and 6
r2=r1*(N1/N2) //mm
//For pulleys 3 and 4, r4 = r3*(N3/N4), or r3*(N3/N4)-r4 = 0
//For a crossed belt drive, r3+r4 = r1+r2
A=[N3/N4 -1; 1 1]
B=[0; r1+r2]
V=A \ B
r3=V(1) //mm
r4=V(2) //mm
//For pulleys 5 and 6, r6 = r5*(N5/N6), or r5*(N5/N6)-r6 = 0
//For a crossed belt drive, r5+r6 = r1+r2
A=[N5/N6 -1; 1 1]
B=[0; r1+r2]
V=A \ B
r5=V(1) //mm
r6=V(2) //mm
//Results:
printf("\n\n For a crossed belt,\n r2 = %.1fmm;\n", r2)
printf(" r3 = %.1f mm;\n",r3)
printf(" r4 = %.1f mm;\n",r4)
printf(" r5 = %.1f mm;\n",r5)
printf(" r6 = %.1f mm.\n\n",r6)
//For an open belt:
//Calcluating the radii of pulleys 2, 3, 4, 5, and 6
r2=r1*(N1/N2) //mm
//Calculating the length of belt for an open belt drive
L=%pi*(r1+r2)+(r2-r1)^2/x+2*x //mm
//For pulleys 3 and 4, r4 = r3*(N3/N4), or r3*(N3/N4)-r4 = 0
//Since L is constant, for pulleys 3 and 4, %pi*(r3+r4)+(r4-r3)^2/x+2*x-L = 0
funcprot(0)
function y=f(a)
    r3=a(1)
```

```
    r4=a(2)
    y(1)=r3*(N3/N4)-r4
    y(2)=%pi*(r3+r4)+(r4-r3)^2/x+2*x-L
endfunction
z=fsolve([1,1],f)
r3=z(1) //mm
r4=z(2) //mm
//For pulleys 5 and 6, r6 = r5*(N5/N6), or r5*(N5/N6)-r6 = 0
//Since L is constant, for pulleys 5 and 6, %pi*(r5+r6)+(r6-r5)^2/x+2*x-L = 0
function y=f(a)
    r5=a(1)
    r6=a(2)
    y(1)=r5*(N5/N6)-r6
    y(2)=%pi*(r5+r6)+(r6-r5)^2/x+2*x-L
endfunction
z=fsolve([1,1],f)
r5=z(1) //mm
r6=z(2) //mm
//Results:
printf(" For an open belt,\n r2 = %.1fmm,\n",r2)
printf(" r3 = %.1f mm;\n",r3)
printf(" r4 = %.1f mm;\n",r4)
printf(" r5 = %d mm;\n",r5)
printf(" r6 = %d mm.\n\n",r6)
```

**Scilab code Exa 11.4** To find the power transmitted

```
//To find the power transmitted
clc
//Given:
d=600/1000 //m
N=200 //rpm
mu=0.25
```

```
theta=160*%pi/180 //radians
T1=2500 //N
//Solution:
//Calcluating the velocity of the belt
v=%pi*d*N/60 //m/s
//Calcluating the tension in the slack side of the belt
T2=T1/exp(mu*theta) //N
//Calcluating the power transmitted by the belt
P=(T1-T2)*v/1000 //kW
//Results:
printf("\n\n Power transmitted by the belt, P = %.2f kW.\n\n",P)
```

**Scilab code Exa 11.5** To find force and power

```
//To find force and power
clc
//Given:
W=9*1000, T1=W //N
d=300/1000 //m
N=20 //rpm
mu=0.25
//Solution:
//Force required by the man:
//Calculating the angle of contact
theta=2.5*2*%pi //rad
//Calculating the force required by the man
T2=T1/exp(mu*theta) //N
//Power to raise the casting:
//Calculating the velocity of the rope
v=%pi*d*N/60 //m/s
//Calculating the power to raise the casting
P=(T1-T2)*v/1000 //kW
//Results:
```

```
printf("\n\n Force required by the man, T2 = %.2f N
    .\n\n",T2)
printf(" Power to raise the casting, P = %.3f kW.\n\
    n",P)
```

**Scilab code Exa 11.6** To find length and power transmitted

```
//To find length and power transmitted
clc
//Given:
d1=450/1000, r1=d1/2, d2=200/1000, r2=d2/2, x=1.95
    //m
N1=200 //rpm
T1=1*1000 //N
mu=0.25
//Solution:
//Calculating the speed of the belt
v=%pi*d1*N1/60 //m/s
//Length of the belt:
//Calculating the length of the crossed belt
L=%pi*(r1+r2)+2*x+(r1+r2)^2/x //m
//Angle of contact between the belt and each pulley:
//Calculating the angle alpha
alpha=asin((r1+r2)/x)*180/%pi //degrees
//Calculating the angle of contact between the belt
    and each pulley
theta=(180+2*alpha)*%pi/180 //radians
//Power transmitted:
//Calculating the tension in the slack side of the
    belt
T2=T1/exp(mu*theta) //N
//Calculating the power transmitted
P=(T1-T2)*v/1000 //kW
//Results:
printf("\n\n Length of the belt, L = %.3f m.\n\n",L)
```

```
printf(" Angle of contact between the belt and each pulley, theta = %.3f rad.\n\n",theta)
printf(" Power transmitted, P = %.2f kW.\n\n",P)
```

**Scilab code Exa 11.7** To find stress in the belt

```
//To find stress in the belt
clc
//Given:
N1=200, N2=300 //rpm
P=6*1000 //W
b=100, t=10 //mm
x=4, d2=0.5 //m
mu=0.3
//Solution:
//Stress in the belt for an open belt drive:
//Calculating the diameter of the larger pulley
d1=d2*(N2/N1) //m
//Calculating the velocity of the belt
v=%pi*d2*N2/60 //m/s
//Calculating the angle alpha for an open belt drive
alphao=asin((r1-r2)/x)*180/%pi //degrees
//Calculating the angle of contact on the smaller pulley
thetao=(180-2*alphao)*%pi/180 //radians
//Calculating the tensions in the belt
//Ratio of the tensions in the belt, T1/T2 = exp(mu*thetao), or T1-T2*exp(mu*thetao) = 0
//Power transmitted, P = (T1-T2)*v, or T1-T2 = P/v
A=[1 -exp(mu*thetao); 1 -1]
B=[0; P/v]
V=A \ B
T1o=V(1) //N
T2o=V(2) //N
//Calculating the stress in the belt
```

```
sigmao=T1o/(b*t) //MPa
//Stress in the belt for a cross belt drive:
//Calculating the angle alpha for a cross belt drive
alphac=asin((d1+d2)/(2*x))*180/%pi //degrees
//Calculating the angle of contact
thetac=(180+2*alphac)*%pi/180 //radians
//Calculating the tensions in the belt
//Ratio of the tensions in the belt, T1/T2 = exp(mu*
    thetac), or T1-T2*exp(mu*thetac) = 0
//Power transmitted, P = (T1-T2)*v, or T1-T2 = P/v
A=[1 -exp(mu*thetac); 1 -1]
B=[0; P/v]
V=A \ B
T1c=V(1) //N
T2c=V(2) //N
//Calculating the stress in the belt
sigmac=T1c/(b*t) //MPa
//Results:
printf("\n\n Stress in the belt for an open belt
    drive, sigma = %.3f MPa.\n\n",sigmao)
printf(" Stress in the belt for a cross belt drive,
    sigma = %.3f MPa.\n\n",sigmac)
```

**Scilab code Exa 11.8** To find width of the belt

```
//To find width of the belt
clc
//Given:
P=7.5*1000 //W
d=1.2, t=10/1000 //m
N=250 //rpm
theta=165*%pi/180 //radians
mu=0.3
sigma=1.5*10^6 //N/m^2
rho=1*10^3 //kg/m^3
```

```
//Solution:
//Calculating the velocity of the belt
v=%pi*d*N/60 //m/s
//Calculating the tensions in the belt
//Power transmitted, P = (T1-T2)*v, or T1-T2 = P/v
//Ratio of tensions in the belt, log(T1/T2) = mu*
    theta, or T1-T2*exp(mu*theta) = 0
A=[1 -1; 1 -exp(mu*theta)]
B=[P/v; 0]
V=A \ B
T1=V(1) //N
T2=V(2) //N
//Calculating the width of the belt
b=T1/(sigma*t-t*1*rho*v^2)*1000 //mm
//Results:
printf("\n\n Width of the belt, b = %.1f mm.\n\n",b)
```

**Scilab code Exa 11.9** To find width of the belt

```
//To find width of the belt
clc
//Given:
t=9.75/1000, d1=300/1000, x=3 //m
P=15*1000 //W
N1=900, N2=300 //rpm
rho=1000 //kg/m^3
sigma=2.5*10^6 //N/m^2
mu=0.3
//Solution:
//Calculating the diameter of the driven pulley
d2=d1*(N1/N2) //m
//Calculating the velocity of the belt
v=%pi*d1*N1/60 //m/s
//Calculating the angle alpha for an open belt drive
alpha=asin((d2-d1)/(2*x))*180/%pi //degrees
```

```
//Calculating the angle of lap
theta=(180-2*alpha)*%pi/180 //radians
//Calculating the tensions in the belt
//Ratio of tensions , log(T1/T2) = mu*theta , or T1-T2
    *exp(mu*theta) = 0
//Power transmitted , P = (T1-T2)*v, or T1-T2 = P/v
A=[1 -exp(mu*theta); 1 -1]
B=[0; P/v]
V=A \ B
T1=V(1) //N
T2=V(2) //N
//Calculating the width of the belt
b=T1/(sigma*t-t*1*rho*v^2)*1000 //mm
//Results:
printf("\n\n Width of the belt , b = %d mm.\n\n",b)
```

**Scilab code Exa 11.10** To find greatest power transmitted

```
//To find greatest power transmitted
clc
//Given:
theta=120*%pi/180 //radians
b=100/1000, t=6/1000 //m
rho=1000 //kg/m^3
mu=0.3
sigma=2*10^6 //N/m^2
//Solution:
//Speed of the belt for greatest power:
//Calculating the maximum tension in the belt
T=sigma*b*t //N
//Calculating the mass of the belt per metre length
l=1 //m
m=b*t*l*rho //kg/m
//Calculating the speed of the belt for greatest
    power
```

```
v=sqrt(T/(3*m)) //m/s
//Greatest power which the belt can transmit
//Calculating the centrifugal tension for maximum
    power to be transmitted
TC=T/3 //N
//Calculating the tension in the tight side of the
    belt
T1=T-TC //N
//Calculating the tension in the slack side of the
    belt
T2=T1/exp(mu*theta) //N
//Calculating the greatest power which the belt can
    transmit
P=(T1-T2)*v/1000 //kW
//Results:
printf("\n\n Speed of the belt for greatest power, v
    = %.2f m/s.\n\n",v)
printf(" Greatest power which the belt can transmit,
    P = %.2f kW.\n\n",P)
```

**Scilab code Exa 11.11** To find torque power and efficiency

```
//To find torque, power and efficiency
clc
//Given:
d1=1.2, r1=d1/2, d2=0.5, r2=d2/2, x=4 //m
m=0.9 //kg/m
T=2000 //N
mu=0.3
N1=200, N2=450 //rpm
//Solution:
//Calculating the velocity of the belt
v=%pi*d1*N1/60 //m/s
//Calculating the centrifugal tension
TC=m*v^2 //N
```

```
//Calculating the tension in the tight side of the
   belt
T1=T-TC //N
//Calculating the angle alpha for an open belt drive
alpha=asin((r1-r2)/x)*180/%pi //degrees
//Calculating the angle of lap on the smaller pulley
theta=(180-2*alpha)*%pi/180 //radians
//Calculating the tension in the slack side of the
   belt
T2=T1/exp(mu*theta) //N
//Calculating the torque on the shaft of larger
   pulley
TL=(T1-T2)*r1 //N-m
//Calculating the torque on the shaft of smaller
   pulley
TS=(T1-T2)*r2 //N-m
//Calculating the power transmitted
P=(T1-T2)*v/1000 //kW
//Power lost in friction:
//Calculating the input power
P1=TL*2*%pi*N1/(60*1000) //kW
//Calculating the output power
P2=TS*2*%pi*N2/(60*1000) //kW
//Calculating the power lost in friction
Pf=P1-P2 //Power lost in friction, kW
//Calculating the efficiency of the drive
eta=P2/P1*100 //%
//Results:
printf("\n\n Torque on the shaft of larger pulley,
  TL = %.1f N-m.\n\n",TL)
printf(" Torque on the shaft of smaller pulley, TS =
   %d N-m.\n\n",TS)
printf(" Power transmitted, P = %.2f kW.\n\n",P)
printf(" Power lost in friction = %.2f kW.\n\n",Pf)
printf(" Efficiency of the drive, eta = %.1f %c.\n\n
  ",eta,"%")
```

**Scilab code Exa 11.12** To find power transmitted

```
//To find power transmitted
clc
//Given:
T0=2000 //N
mu0=0.3
theta=150*%pi/180 //radians
r2=200/1000, d2=2*r2 //m
N2=500//rpm
//Solution:
//Calculating the velocity of the belt
v=%pi*d2*N2/60 //m/s
//Calculating the tensions in the belt
//Initial tension, T0 = (T1+T2)/2, or T1+T2 = 2*T0
//Ratio of the tensions in the belt, log(T1/T2) = mu*theta, or T1-T2*exp(mu*theta) = 0
A=[1 1; 1 -exp(mu*theta)]
B=[2*T0; 0]
V=A \ B
T1=V(1) //N
T2=V(2) //N
//Calculating the power transmitted
P=(T1-T2)*v/1000 //kW
//Results:
printf("\n\n Power transmitted, P = %.1f kW.\n\n",P)
```

---

**Scilab code Exa 11.13** To find power trnasmitted

```
//To find power trnasmitted
clc
//Given:
```

```
x=4.8, d1=1.5, d2=1 //m
T0=3*1000 //N
m=1.5 //kg/m
mu=0.3
N2=400 //rpm
//Solution:
//Calculating the velocity of the belt
v=%pi*d2*N2/60 //m/s
//Calculating the centrifugal tension
TC=m*v^2 //N
//Calculating the angle alpha
alpha=asin((d1-d2)/(2*x))*180/%pi //degrees
//Calculating the angle of lap for the smaller pulley
theta=(180-2*alpha)*%pi/180 //radians
//Calculating the tensions in the belt
//Initial tension, T0 = (T1+T2+2*TC)/2, or T1+T2 = 2*(T0-TC)
//Ratio of tensions in the belt, log(T1/T2) = mu*theta, or T1-T2*exp(mu*theta) = 0
A=[1 1; 1 -exp(mu*theta)]
B=[2*(T0-TC); 0]
V=A \ B
T1=V(1) //N
T2=V(2) //N
//Calculating the power transmitted
P=(T1-T2)*v/1000 //kW
//Results:
printf("\n\n Power transmitted, P = %.1f kW.\n\n",P)
```

**Scilab code Exa 11.14** To find diameter power and tension

```
//To find diameter, power and tension
clc
//Given:
```

```
x=1.2, d2=400/1000, t=5/1000, b=80/1000 //m
N1=350, N2=140 //rpm
mu=0.3
sigma=1.4*10^6 //N/m^2
//Solution:
//Calculating the diameter of the driving pulley
d1=d2*(N2/N1) //m
//Maximum power transmitted by the belting:
//Refer Fig. 11.18
//Calculating the angle alpha
alpha=asin((d2-d1)/(2*x))*180/%pi //degrees
//Calculating the angle of contact of the belt on the driving pulley
theta=(180-2*alpha)*%pi/180 //radians
//Calculating the maximum tension to which the belt can be subjected
T1=sigma*b*t //N
//Calculating the tension in the slack side of the belt
T2=T1/exp(mu*theta) //N
//Calculating the velocity of the belt
v=%pi*d1*N1/60 //m/s
//Calculating the power transmitted
P=(T1-T2)*v/1000 //kW
//Calculating the required initial belt tension
T0=(T1+T2)/2 //N
//Results:
printf("\n\n Diameter of the driving pulley, d1 = %.2f m.\n\n",d1)
printf(" Maximum power transmitted by the belting, P = %.3f kW.\n\n",P)
printf(" Required initial belt tension, T0 = %.1f N.\n\n",T0)
```

**Scilab code Exa 11.15** To find width tension and length

```
//To find width, tension and length
clc
//Given:
d2=240/1000, d1=600/1000, x=3 //m
P=4*1000 //W
N2=300 //rpm
mu=0.3
T1s=10 //Safe working tension, N/mm width
//Solution:
//Minimum width of the belt:
//Calculating the velocity of the belt
v=%pi*d2*N2/60 //m/s
//Calculating the angle alpha for an open belt drive
alpha=asin((d1-d2)/(2*x))*180/%pi //degrees
//Calculating the angle of lap on the smaller pulley
theta=(180-2*alpha)*%pi/180 //radians
//Calculating the tensions in the belt
//Power transmitted, P = (T1-T2)*v, or T1-T2 = P/v
//Ratio of tensions, log(T1/T2) = mu*theta, or T1-T2*exp(mu*theta) = 0
A=[1 -1; 1 -exp(mu*theta)]
B=[P/v; 0]
V=A \ B
T1=V(1) //N
T2=V(2) //N
//Calculating the minimum width of the belt
b=T1/T1s //mm
//Calculating the initial belt tension
T0=(T1+T2)/2 //N
//Calculating the length of the belt required
L=%pi/2*(d1+d2)+2*x+(d1-d2)^2/(4*x) //m
//Results:
printf("\n\n Minimum width of the belt, b = %.1f mm.\n\n",b)
printf(" Initial belt tension, T0 = %.1f N.\n\n",T0)
printf(" Length of the belt required, L = %.2f m.\n\n",L)
```

**Scilab code Exa 11.16** To find power transmitted

```
//To find power transmitted
clc
//Given:
d1=400/1000 ,d2=250/1000, x=2, mu=0.4 //m
T=1200 //N
v=10 //m/s
//Solution:
//Power transmitted:
//Calculating the angle alpha for an open belt drive
alpha=asin((d1-d2)/(2*x))*180/%pi //degrees
//Calculating the angle of contact
theta=(180-2*alpha)*%pi/180 //radians
//Calculating the tension in the tight side of the
    belt
T1=T //Neglecting centrifugal tension, N
//Calculating the tension in the slack side of the
    belt
T2=T1/exp(mu*theta) //N
//Calculating the power transmitted
P=(T1-T2)*v/1000 //kW
//Results:
printf("\n\n Power transmitted, P = %.2f kW.\n\n",P)
//Power transmitted when initial tension is
    increased by 10%:
//Calculating the initial tension
T0=(T1+T2)/2 //N
//Calculating the increased initial tension
T0dash=T0+10/100*T0 //N
//Calculating the corresponding tensions in the belt
//We have, T0dash = (T1+T2)/2, or T1+T2 = 2*T0dash
//Ratio of the tensions, log(T1/T2) = mu*theta, or
    T1-T2*exp(mu*theta) = 0
```

```
A=[1 1; 1 -exp(mu*theta)]
B=[2*T0dash; 0]
V=A \ B
T1=V(1) //N
T2=V(2) //N
//Calculating the power transmitted
P1=(T1-T2)*v/1000 //kW
//Power transmitted when coefficient of friction is increased by 10%:
//Calculating the increased coefficient of friction
mudash=mu+10/100*mu
//Calculating the corresponding tensions in the belt
//Ratio of the tensions, log(T1/T2) = mudash*theta, or T1-T2*exp(mudash*theta) = 0
//Initial tension, T0 = (T1+T2)/2, or T1+T2 = 2*T0
A=[1 -exp(mudash*theta); 1 1]
B=[0; 2*T0]
V=A \ B
T1=V(1) //N
T2=V(2) //N
//Calculating the power transmitted
P2=(T1-T2)*v/1000 //kW
//Results:
if P1>P2 then
    printf(" Since the power transmitted by increasing the initial tension is more, therefore in order to increase the power transmitted we shall adopt the method of increasing the initial tension.\n\n")
else
    printf(" Since the power transmitted by increasing the coefficient of friction is more, therefore in order to increase the power transmitted we shall adopt the method of increasing the coefficient of friction.\n\n")
end
//Percentage increase in power:\
```

```
//Calculating the percentage increase in power when
    the initial tension is increased
I1=(P1-P)/P*100 //Percentage increase in power when
    the initial tension is increased, %
//Calculating the percentage increase in power when
    coefficient of friction is increased
I2=(P2-P)/P*100 //Percentage increase in power when
    coefficient of friction is increased, %
//Results:
printf(" Percentage increase in power when the
    initial tension is increased = %.2f %c.\n\n",I1,"
   %")
printf(" Percentage increase in power when
    coefficient of friction is increased = %.1f %c.\n
    \n",I2,"%")
```

**Scilab code Exa 11.17** To find power and shaft speed

```
//To find power and shaft speed
clc
//Given:
funcprot(0)
beta=30/2 //degrees
alpha=750*10^-6 //mm^2
mu=0.12
rho=1.2*1000 //kg/m^3
sigma=7*10^6 //N/m^2
d=300/1000 //m
N=1500 //rpm
//Solution:
//Power transmitted:
//Calculating the velocity of the belt
v=%pi*d*N/60 //m/s
//Calculating the mass of the belt per metre length
l=1 //m
```

```
m=alpha*l*rho //kg/m
//Calculating the centrifugal tension
TC=m*v^2 //N
//Calculating the maximum tension in the belt
T=sigma*alpha //N
//Calculating the tension in the tight side of the
    belt
T1=T-TC //N
//Calculating the tension in the slack side of the
    belt
theta=%pi //Angle of contact, radians
T2=T1/exp(mu*theta*(1/sind(beta))) //N
//Calculating the power transmitted
P=(T1-T2)*v*2/1000 //kW
//Shaft speed:
//Calculating the belt speed for maximum power
    transmitted
v1=sqrt(T/(3*m)) //m/s
//Calculating the shaft speed for maximum power
    transmitted
N1=v1*60/(%pi*d) //rpm
//Results:
printf("\n\n Power transmitted, P = %.3f kW.\n\n",P)
printf(" Shaft speed at which the power transmitted
    would be maximum, N1 = %d rpm.\n\n",N1)
```

**Scilab code Exa 11.18** To find maximum power transmitted

```
//To find maximum power transmitted
clc
//Given:
funcprot(0)
beta=30/2 //degrees
t=20/1000, b=20/1000 //m
m=0.35 //kg/m
```

```
sigma=1.4*10^6 //N/m^2
theta=140*%pi/180 //radians
mu=0.15
//Solution:
//Calculating the maximum tension in the belt
T=sigma*b*t //N
//Calculating the velocity of the belt for maximum
    power to be transmitted
v=sqrt(T/(3*m)) //m/s
//Calculating the centrifugal tension
TC=T/3 //N
//Calculating the tension in the tight side of the
    belt
T1=T-TC //N
//Calculating the tension in the slack side of the
    belt
T2=T1/exp(mu*theta*(1/sind(beta))) //N
//Calculating the maximum power transmitted
P=(T1-T2)*v/1000 //kW
//Results:
printf("\n\n Maximum power transmitted, P = %.2f kW
    .\n\n",P)
```

**Scilab code Exa 11.19** To find number of V belts

```
//To find number of V-belts
clc
//Given:
P=90 //kW
N2=250, N1=750 //rpm
d2=1, x=1.75 //m
v=1600/60 //m/s
a=375*10^-6 //m^2
rho=1000 //kg/m^3
sigma=2.5*10^6 //N/m^2
```

```
beta=35/2 //degrees
mu=0.25
//Solution:
//Calculating the diameter of the pulley on the motor shaft
d1=d2*(N2/N1) //m
//Calculating the mass of the belt per metre length
l=1 //m
m=a*l*rho //kg/m
//Calculating the centrifugal tension
TC=m*v^2 //N
//Calculating the maximum tension in the belt
T=sigma*a //N
//Calculating the tension in the tight side of the belt
T1=T-TC //N
//Refer Fig. 11.21
//Calculating the angle alpha
alpha=asin((d2-d1)/(2*x))*180/%pi //degrees
//Calculating the angle of lap on smaller pulley
theta=(180-2*alpha)*%pi/180 //radians
//Calculating the tension in the slack side of the belt
T2=T1/exp(mu*theta*(1/sind(beta))) //N
//Number of V-belts:
//Calculating the power transmitted per belt
P1=(T1-T2)*v/1000 //Power transmitted per belt, kW
//Calculating the number of V-belts
n=P/P1 //Number of V-belts
//Calculating the length each of belt for an open belt drive
L=%pi/2*(d2+d1)+2*x+(d2-d1)^2/(4*x) //m
//Results:
printf("\n\n Number of V-belts = %d.\n\n",n+1)
printf(" Length of each belt, L = %.3f m.\n\n",L)
```

**Scilab code Exa 11.20** To find number of ropes required

```
//To find number of ropes required
clc
//Given:
P=600 //kW
d=4 //m
N=90 //rpm
theta=160*%pi/180 //radians
funcprot(0)
beta=45/2 //degrees
mu=0.28
m=1.5 //kg/m
T=2400 //N
//Solution:
//Calculating the velocity of the rope
v=%pi*d*N/60 //m/s
//Calculating the centrifugal tension
TC=m*v^2 //N
//Calculating the tension in the tight side of the
    rope
T1=T-TC //N
//Calculating the tension in the slack side of the
    belt
T2=T1/exp(mu*theta*(1/sind(beta))) //N
//Calculating the power transmitted per rope
P1=(T1-T2)*v/1000 //Power transmitted per rope, kW
//Calculating the number of ropes
n=P/P1 //Number of ropes
//Results:
printf("\n\n Number of ropes required = %d.\n\n",n
    +1)
```

**Scilab code Exa 11.21** To find speed of pulley

```
//To find speed of pulley
clc
//Given:
d=3.6 //m
n=15 //Number of grooves
funcprot(0)
beta=45/2 //degrees
theta=170*%pi/180 //radians
mu=0.28
T=960 //N
m=1.5 //kg/m
//Solution:
//Speed of the pulley:
//Calculating the velocity of the rope
v=sqrt(T/(3*m)) //m/s
//Calculating the speed of the pulley
N=v*60/(%pi*d) //rpm
//Power transmitted
//Calculating the centrifugal tension for maximum
   power
TC=T/3 //N
//Calculating the tension in the tight side of the
   rope
T1=T-TC //N
//Calculating the tension in the slack side of the
   rope
T2=T1/exp(mu*theta*(1/sind(beta))) //N
//Calculating the power transmitted per rope
P1=(T1-T2)*v/1000 //Power transmitted per rope, kW
//Calculating the total power transmitted
P=P1*n //Total power transmitted, kW
//Results:
```

```
printf("\n\n Speed of the pulley for maximum power, N = %.1f rpm.\n\n",N)
printf(" Power transmitted = %.2f kW.\n\n",P)
```

**Scilab code Exa 11.22** To find initial tension and diameter

```
//To find initial tension and diameter
clc
//Given:
PT=24 //kW
d=400/1000 //m
N=110 //rpm
funcprot(0)
beta=45/2 //degrees
theta=160*%pi/180 //radians
mu=0.28
n=10
//Solution:
//Initial tension:
//Calculating the power transmitted per rope
P=PT/n*1000 //W
//Calculating the velocity of the rope
v=%pi*d*N/60 //m/s
//Calculating the tensions in the rope
//Power transmitted, P = (T1-T2)*v, or T1-T2 = P/v
//Ratio of tensions, log(T1/T2) = mu*theta*(1/sind(beta)), or T1-T2*exp(mu*theta*(1/sind(beta))) = 0
A=[1 -1; 1 -exp(mu*theta*(1/sind(beta)))]
B=[P/v; 0]
V=A \ B
T1=V(1) //N
T2=V(2) //N
//Calculating the initial tension in each rope
T0=(T1+T2)/2 //N
//Diameter of each rope:
```

```
//Calculating the girth of rope
C=sqrt(T1/(122*10^3-53*v^2))*1000 //mm
//Calculating the diameter of each rope
d1=C/%pi //mm
//Results:
printf("\n\n Initial tension, T0 = %.2f N.\n\n",T0)
printf(" Diameter of each rope, d1 = %.2f mm.\n\n",d1)
```

**Scilab code Exa 11.23** To find pitch and length of chain

```
//To find pitch and length of chain
clc
//Given:
N1=240, N2=120 //rpm
T1=20
d2=600/1000, r2=d2/2, x=800/1000 //m
//SOlution:
//Calculating the number of teeth on the drive sprocket
T2=T1*(N1/N2)
//Calculating the pitch of the chain
p=r2*2*sind(180/T2)*1000 //mm
//Length of the chain:
m=x*1000/p
//Calculating the multiplying factor
K=(T1+T2)/2+2*m+(1/sind(180/T1)-1/sind(180/T2))^2/(4*m)
//Calculating the length of the chain
L=p*K/1000 //m
//Results:
printf("\n\n Number of teeth on the driven sprocket, T2 = %d.\n\n",T2)
printf(" Pitch of the chain, p = %.1f mm.\n\n",p)
printf(" Length of the chain, L = %.4f m.\n\n",L)
```

# Chapter 12

# Toothed Gearing

**Scilab code Exa 12.1** To find total load

```
//To find total load
clc
//Given:
P=120*1000 //W
d=250/1000, r=d/2 //m
N=650 //rpm
phi=20 //degrees
//Solution:
//Calculating the angular speed of the gear
omega=2*%pi*N/60 //rad/s
//Calculating the torque transmitted
T=P/omega //N-m
//Calculating the tangential load on the pinion
FT=T/r //N
//Calculating the total load due to power
   transmitted
F=FT/(cosd(phi)*1000) //kN
//Results:
printf("\n\n Total load due to power transmitted, F
   = %.3f kN.\n\n",F)
```

**Scilab code Exa 12.2** To find addendum

```
//To find addendum
clc
//Given:
T=40, t=T
phi=20 //degrees
m=6 //mm
//Solution:
//Calculating the circular pitch
pc=%pi*m //mm
//Calculating the length of arc of contact
Lac=1.75*pc //Length of arc of contact, mm
//Calculating the length of path of contact
Lpc=Lac*cosd(phi) //Length of path of contact, mm
//Calculating the pitch circle radii of each wheel
R=m*T/2 //mm
r=R //mm
//Calculating the radius of the addendum circle of
    each wheel
RA=sqrt(R^2*(cosd(phi))^2+(Lpc/2+R*sind(phi))^2) //
    mm
//Calculating the addendum of the wheel
Ad=RA-R //Addendum of the wheel, mm
//Results:
printf("\n\n Addendum of the wheel = %.2f mm.\n\n",
    Ad)
```

**Scilab code Exa 12.3** To find length of path of contact

```
//To find length of path of contact
clc
```

```
//Given:
t=30, T=80
phi=20 //degrees
m=12 //mm
Addendum=10 //mm
//Solution:
//Length of path of contact:
//Calculating the pitch circle radius of pinion
r=m*t/2 //mm
//Calculating the pitch circle radius of gear
R=m*T/2 //mm
//Calculating the radius of addendum circle of pinion
rA=r+Addendum //mm
//Calculating the radius of addendum circle of gear
RA=R+Addendum //mm
//Calculating the length of path of approach
//Refer Fig. 12.11
KP=sqrt(RA^2-R^2*(cosd(phi))^2)-R*sind(phi) //mm
//Calculating the length of path of recess
PL=sqrt(rA^2-r^2*(cosd(phi))^2)-r*sind(phi) //mm
//Calculating the length of path of contact
KL=KP+PL //mm
//Calculating the length of arc of contact
Lac=KL/cosd(phi) //Length of arc of contact, mm
//Contact ratio:
//Calculating the circular pitch
Pc=%pi*m //mm
//Calculating the contact ratio
CR=Lac/pc //Contact ratio
//Results:
printf("\n\n Length of path of contact, KL = %.1f mm.\n\n",KL)
printf(" Length of arc of contact = %.2f mm.\n\n",Lac)
printf(" Contact ratio = %d.\n\n",CR)
```

**Scilab code Exa 12.4** To find angle and maximum velocity

```
//To find angle and maximum velocity
clc
//Given:
phi=20 //degrees
t=20, G=2
m=5 //mm
v=1.2 //m/s
addendum=1*m //mm
//Solution:
//Angle turned through by pinion when one pair of teeth is in mesh:
//Calculating the pitch circle radius of pinion
r=m*t/2 //mm
//Calculating the pitch circle radius of wheel
R=m*G*t/2 //mm
//Calculating the radius of addendum circle of pinion
rA=r+addendum //mm
//Calculating the radius of addendum circle of wheel
RA=R+addendum //mm
//Calculating the length of path of approach
KP=sqrt(RA^2-R^2*(cosd(phi))^2)-R*sind(phi) //mm
//Calculating the length of path of recess
PL=sqrt(rA^2-r^2*(cosd(phi))^2)-r*sind(phi) //mm
//Calculating the length of path of contact
KL=KP+PL //mm
//Calculating the length of arc of contact
Lac=KL/cosd(phi) //mm
//Calculating the angle turned by the pinion
angle=Lac*360/(2*%pi*r) //Angle turned by the pinion, degrees
//Maximum velocity of sliding:
```

```
//Calculating the angular speed of pinion
omega1=v*1000/r //rad/s
//Calculating the angular speed of wheel
omega2=v*1000/R //rad/s
//Calculating the maximum velocity of sliding
vS=(omega1+omega2)*KP //mm/s
//Results:
printf("\n\n Angle turned through by pinion when one
    pair of teeth is in mesh = %.2f degrees.\n\n",
  angle)
printf(" Maximum velocity of sliding, vS = %.1f mm/s
  .\n\n",vS)
```

**Scilab code Exa 12.5** To find velocity and angle turned

```
//To find velocity and angle turned
clc
//Given:
T=40, t=20
N1=2000 //rpm
phi=20 //degrees
addendum=5, m=5 //mm
//Solution:
//Calculating the angular velocity of the smaller
  gear
omega1=2*%pi*N1/60 //rad/s
//Calculating the angular velocity of the larger
  gear
omega2=omega1*t/T //rad/s
//Calculating the pitch circle radius of the smaller
    gear
r=m*t/2 //mm
//Calculating the pitch circle radius of the larger
  gear
R=m*T/2 //mm
```

```
//Calculating the radius of aaddendum circle of
   smaller gear
rA=r+addendum //mm
//Calculating the radius of addendum circle of
   larger gear
RA=R+addendum //mm
//Calculating the length of path of approach
KP=sqrt(RA^2-R^2*(cosd(phi))^2)-R*sind(phi) //mm
//Calculating the length of path of recess
PL=sqrt(rA^2-r^2*(cosd(phi))^2)-r*sind(phi) //mm
//Calculating the velocity of sliding at the point
   of engagement
vSK=(omega1+omega2)*KP //mm/s
//Calculating the velocity of sliding at the point
   of disengagement
vSL=(omega1+omega2)*PL //mm/s
//Angle through which the pinion turns:
//Calculating the length of path of contact
KL=KP+PL //mm
//Calculating the length of arc of contact
Lac=KL/cosd(phi) //Length of arc of contact, mm
//Calculating the circumference of pinion
C=2*%pi*r //Circumference of pinion, mm
//Calculating the angle through which the pinion
   turns
angle=Lac*360/C //Angle through which the pinion
   turns, degrees
//Results:
printf("\n\n Velocity of sliding at the point of
   engagement, vSK = %d mm/s.\n\n",vSK)
printf(" Since the velocity of sliding is
   proportional to the distance of the contact point
    from the pitch point, therefore the velocity of
   sliding at the pitch point is zero.\n\n")
printf(" Velocity of sliding at the point of
   disengagement, vsL = %d mm/s.\n\n",vSL)
printf(" Angle through which the pinion turns = %.2f
    degrees.\n\n",angle)
```

**Scilab code Exa 12.6** To find teeth angle and ratio

```
//To find teeth , angle and ratio
clc
//Given:
phi=20 //degrees
m=6, addendum=1*m //mm
t=17, T=49
//Solution:
//Number of pairs of teeth in contact:
//Calculating the pitch circle radius of pinion
r=m*t/2 //mm
//Calculating the pitch circle radius of gear
R=m*T/2 //mm
//Calculating the radius of addendum circle of pinion
rA=r+addendum //mm
//Calculating the radius of addendum circle of gear
RA=R+addendum //mm
//Calculating the length of path of approach
//Refer Fig. 12.11
KP=sqrt(RA^2-R^2*(cosd(phi))^2)-R*sind(phi) //mm
//Calculating the length of path of recess
PL=sqrt(rA^2-r^2*(cosd(phi))^2)-r*sind(phi) //mm
//Calculating the length of path of contact
KL=KP+PL //mm
//Calculating the length of arc of contact
Lac=KL/cosd(phi) //Length of arc of contact , mm
//Calculating the circular pitch
pc=%pi*m //mm
//Calculating the number of pairs of teeth in contact
n=Lac/pc //Number of pairs of teeth in contact
//Angle turned by the pinion and gear wheel when one
```

```
pair of teeth is in contact:
//Calculating the angle turned through by the pinion
anglep=Lac*360/(2*%pi*r) //Angle turned through by the pinion, degrees
//Calculating the angle turned through by the wheel
angleg=Lac*360/(2*%pi*R) //Angle turned through by the gear wheel, degrees
//Ratio of sliding to rolling motion:
//At the instant when the tip of a tooth on the larger wheel is just making contact with its mating teeth
r1=((1+t/T)*KP)/r //Ratio of sliding velocity to rolling velocity
//At the instant when the tip of a tooth on a larger wheel is just leaving contact with its mating teeth
r2=((1+t/T)*PL)/r //Ratio of sliding velocity to rolling velocity
//Results:
printf("\n\n Number of pairs of teeth in contact = %d.\n\n",n+1)
printf(" Angle turned through by the pinion = %.1f degrees.\n\n",anglep)
printf(" Angle turned through by the gear wheel = %d degrees.\n\n",angleg)
printf(" At the instant when the tip of a tooth on the larger wheel is just making contact with its mating teeth, ratio of sliding velocity to rolling velocity = %.2f.\n\n",r1)
printf(" At the instant when the tip of a tooth on a larger wheel is just leaving contact with its mating teeth, ratio of sliding velocity to rolling velocity = %.3f.\n\n",r2)
printf(" Since at the pitch point, the sliding velocity is zero,, therefore the ratio of sliding velocity to rolling velocity is zero.\n\n")
```

**Scilab code Exa 12.7** To find length of path of contact

```
//To find length of path of contact
clc
//Given:
t=18, T=72
phi=20 //degrees
m=4 //mm
addendump=8.5 //Addendum on pinion, mm
addendumg=3.5 //Addendum on gear, mm
//SOlution:
//Refer Fig. 12.12
//Calculating the pitch circle radius of the pinion
r=m*t/2 //mm
//Calculating the pitch circle radius of the gear
R=m*T/2 //mm
//Calculating the radius of addendum circle of the
    pinion
rA=r+addendump //mm
//Calculating the radius of addendum circle of the
    gear
RA=R-addendumg //mm
//Calculating the radius of the base circle of the
    pinion
O1M=r*cosd(phi) //mm
//Calculating the radius of the base circle of the
    gear
O2N=R*cosd(phi) //mm
//Calculating the length of path of approach
KP=R*sind(phi)-sqrt(RA^2-O2N^2) //mm
//Calculating the length of path of recess
PL=sqrt(rA^2-O1M^2)-r*sind(phi) //mm
//Calculating the length of the path of contact
KL=KP+PL //mm
```

```
//Results:
printf("\n\n Length of the path of contact, KL = %.2
    f mm.\n\n",KL)
```

**Scilab code Exa 12.8** To find path and arc of contact

```
//To find path and arc of contact
clc
//Given:
t=20, T=40
m=10 //mm
phi=20 //degrees
//Solution:
//Addendum height for each gear wheel:
//Calculating the pitch circle radius of the smaller
     gear wheel
r=m*t/2 //mm
//Calculating the pitch circle radius of the larger
    wheel
R=m*T/2 //mm
//Calculating the radius of addendum circle for the
    larger gear wheel
RA=sqrt((r*sind(phi)/2+R*sind(phi))^2+R^2*(cosd(phi)
    )^2) //mm
//Calculating the addendum height for larger gear
    wheel
addendumg=RA-R //mm
//Calculating the radius of addendum circle for the
    smaller gear wheel
rA=sqrt((R*sind(phi)/2+r*sind(phi))^2+r^2*(cosd(phi)
    )^2) //mm
//Calculating the addendum height for smaller gear
    wheel
addendump=rA-r //mm
//Calculating the length of the path of contact
```

```
Lpc=(r+R)*sind(phi)/2 //Length of the path of
   contact , mm
//Calculating the length of the arc of contact
Lac=Lpc/cosd(phi) //Length of the arc of contact , mm
//Contact ratio:
//Calculating the circular pitch
pc=%pi*m //mm
//Calculating the contact ratio
CR=Lpc/pc //Contact ratio
//Results:
printf("\n\n Addendum height for larger gear wheel =
    %.1f mm.\n\n",addendumg)
printf(" Addendum height for smaller gear wheel = %
   .1f mm.\n\n",addendump)
printf(" Length of the path of contact = %.1f mm.\n\
   n",Lpc)
printf(" Length of the arc of contact = %.1f mm.\n\n
   ",Lac)
printf(" Contact ratio = %d.\n\n",CR+1)
```

**Scilab code Exa 12.9** To find number of teeth

```
//To find number of teeth
clc
//Given:
G=3
phi=20 //degrees
Aw=1 //module
//Solution:
//Calculating the minimum number of teeth for a gear
    ratio of 3:1
t1=(2*Aw)/(G*(sqrt(1+1/G*(1/G+2)*(sind(phi))^2)-1))
//Calculating the minimum number of teeth for equal
   wheel
t2=(2*Aw)/(sqrt(1+3*(sind(phi))^2)-1)
```

```
//Results:
printf("\n\n Minimum number of teeth for a gear
   ratio of 3:1, t = %d.\n\n",t1+1)
printf(" Minimum number of teeth for equal wheel, t
   = %d.\n\n",t2+1)
```

**Scilab code Exa 12.10** To find number of teeth

```
//To find number of teeth
clc
//Given:
G=4
phi=14.5 //degrees
//Solution:
//Least number of teeth on each wheel:
//Calculating the least number of teeth on the
   pinion
t=2*%pi/tand(phi)
//Calculating the least number of teeth on the gear
T=G*t
//Results:
printf("\n\n Least number of teeth on the pinion, t
   = %d.\n\n",t+1)
printf(" Least number of teeth on the gear, T = %d.\
   n\n",T+1)
```

**Scilab code Exa 12.11** To find addenda and velocity

```
//To find addenda and velocity
clc
//Given:
phi=16 //degrees
m=6 //mm
```

```
t=16, G=1.75, T=G*t
N1=240 //rpm
//Solution:
//Calculating the angular speed of the pinion
omega1=2*%pi*N1/60 //rad/s
//Addenda on pinion and gear wheel:
//Calculating the addendum on pinion
addendump=m*t/2*(sqrt(1+T/t*(T/t+2)*(sind(phi))^2)
    -1) //Addendum on pinion, mm
//Calculating the addendum on wheel
addendumg=m*T/2*(sqrt(1+t/T*(t/T+2)*(sind(phi))^2)
    -1) //Addendum on wheel, mm
//Length of path of contact:
//Calculating the pitch circle radius of wheel
R=m*T/2 //mm
//Calculating the pitch circle radius of pinion
r=m*t/2 //mm
//Calculating the addendum circle radius of wheel
RA=R+addendump //mm
//Calculating the addendum circle radius of pinion
rA=r+addendumg //mm
//Calculating the length of path of approach
KP=sqrt(RA^2-R^2*(cosd(phi))^2)-R*sind(phi) //mm
//Calculating the length of path of recess
PL=sqrt(rA^2-r^2*(cosd(phi))^2)-r*sind(phi) //mm
//Calculating the length of path of contact
KL=KP+PL //mm
//Maximum velocity of sliding of teeth on either
    side of pitch point:
//Calculating the angular speed of gear wheel
omega2=omega1/G //rad/s
//Calculating the maximum velocity of sliding of
    teeth on the left side of pitch point
vmax1=(omega1+omega2)*KP //Maximum velocity of
    sliding of teeth on the left side of pitch point,
     mm/s
//Calculating the maximum velocity of sliding of
    teeth on the right side of pitch point
```

```
vmaxr=(omega1+omega2)*PL //Maximum velocity of
    sliding of teeth on the right side of pitch point
    , mm/s
//Results:
printf("\n\n Addendum on pinion = %.2f mm.\n\n",
    addendump)
printf(" Addendum on wheel = %.2f mm.\n\n",addendumg
    )
printf(" Length of path of contact, KL = %.2f mm.\n\
    n",KL)
printf(" Maximum velocity of sliding of teeth on the
      left side of pitch point = %d mm/s.\n\n",vmaxl)
printf(" Maximum velocity of sliding of teeth on the
      right side of pitch point = %d mm/s.\n\n",vmaxr)
```

**Scilab code Exa 12.12** To find sliding velocities and contact ratio

```
//To find sliding velocities and contact ratio
clc
//Given:
phi=20 //degrees
t=30, T=50, m=4
N1=1000 //rpm
//Solution:
//Calculating the angular speed of thr pinion
omega1=2*%pi*N1/60 //rad/s
//Sliding velocities at engagement and at
    disengagement of a pair of teeth:
//Calculating the addendum of the smaller gear
addendump=m*t/2*(sqrt(1+T/t*(T/t+2)*(sind(phi))^2)
    -1) //Addendum of the smaller gear, mm
//Calculating the addendum of the larger gear
addendumg=m*T/2*(sqrt(1+t/T*(t/T+2)*(sind(phi))^2)
    -1) //Addendum of the larger gear, mm
//Calculating the pitch circle radius of the smaller
```

```
gear
r=m*t/2 //mm
//Calculating the radius of addendum circle of the smaller gear
rA=r+addendump //mm
//Calculating the pitch circle radius of the larer gear
R=m*T/2 //mm
//Calculating the radius of addendum circle of the larger gear
RA=R+addendumg //mm
//Calculating the path of approach
KP=sqrt(RA^2-R^2*(cosd(phi))^2)-R*sind(phi) //mm
//Calculating the path of recess
PL=sqrt(rA^2-r^2*(cosd(phi))^2)-r*sind(phi) //mm
//Calculating the angular speed of the larger gear
omega2=omega1*t/T //rad/s
//Calculating the sliding velocity at engagement of a pair of teeth
v1=(omega1+omega2)*KP //Sliding velocity at engagement of a pair of teeth, mm/s
//Calculating the sliding velocity at disengagement of a pair of teeth
v2=(omega1+omega2)*PL //Sliding velocity at disengagement of a pair of teeth, mm/s
//Contact ratio:
//Calculating the length of the arc of contact
Lac=(KP+PL)/cosd(phi) //mm
//Calculating the circular pitch
pc=%pi*m //Circular pitch, mm
//Calculating the contact ratio
CR=Lac/pc //Contact ratio
//Results:
printf("\n\n Sliding velocity at engagement of a pair of teeth = %.3f m/s.\n\n",v1/1000)
printf(" Sliding velocity at disengagement of a pair of teeth = %.3f m/s.\n\n",v2/1000)
printf(" Contact ratio = %d.\n\n",CR+1)
```

**Scilab code Exa 12.13** To find teeth and velocity

```
//To find teeth and velocity
clc
//Given:
G=3
m=6, AP=1*m, AW=AP //mm
phi=20 //degrees
N1=90 //rpm
//Solution:
//Calculating the angular speed of the pinion
omega1=2*%pi*N1/60 //rad/s
//Calculating the number of teeth on the pinion to
    avoid interference on it
t=2*AP/(sqrt(1+G*(G+2)*(sind(phi))^2)-1)
//Calculating the corresponding number of teeth on
    the wheel
T=G*t
//Length of path and arc of contact:
//Calculating the pitch circle radius of pinion
r=m*t/2 //mm
//Calculating the radius of addendum circle of
    pinion
rA=r+AP //mm
//Calculating the pitch circle radius of wheel
R=m*T/2 //mm
//Calculating the radius of addendum circle of wheel
RA=R+AW //mm
//Calculating the path of approach
KP=sqrt(RA^2-R^2*(cosd(phi))^2)-R*sind(phi) //mm
//Calculating the path of recess
PL=sqrt(rA^2-r^2*(cosd(phi))^2)-r*sind(phi) //mm
//Calculating the length of path of contact
KL=KP+PL //mm
```

```
//Calculating the length of arc of contact
Lac=KL/cosd(phi) //Length of arc of contact, mm
//Number of pairs of teeth in contact:
//Calculating the circular pitch
pc=%pi*m //mm
//Calculating the number of pairs of teeth in contact
n=Lac/pc //Number of pairs of teeth in contact
//Maximum velocity of sliding:
//Calculating the angular speed of wheel
omega2=omega1*t/T //rad/s
//Calculating the maximum velocity of sliding
vs=(omega1+omega2)*KP //mm/s
//Results:
printf("\n\n Number of teeth on the pinion to avoid interference, t = %d.\n\n",t+1)
printf(" Corresponding number of teeth on the wheel, T = %d.\n\n",T+1)
printf(" Length of path of contact, KL = %.2f mm.\n\n",KL)
printf(" Length of arc of contact = %.2f mm.\n\n",Lac)
printf(" Number of pairs of teeth in contact = %d.\n\n",n+1)
printf(" Maximum velocity of sliding, vs = %.2f mm/s.\n\n",vs)
```

---

**Scilab code Exa 12.14** To find pressure angle and teeth

```
//To find pressure angle and teeth
clc
//Given:
T=20
d=125, r=d/2, OP=r, LH=6.25 //mm
//Calculating the least pressure angle to avoid
```

```
interference
phi=asin(sqrt(LH/r))*180/%pi //degrees
//Length of arc of contact:
//Calculating the length of path of contact
KL=sqrt((OP+LH)^2-(OP*cosd(phi))^2) //mm
//Calculating the length of arc of contact
Lac=KL/cosd(phi) //Length of arc of contact, mm
//Minimum number of teeth:
//Calculating the circular pitch
pc=%pi*d/T //mm
//Calculating the number of pairs of teeth in contact
n=Lac/pc //Number of pairs of teeth in contact
//Calculating the minimum number of teeth in contact
nmin=n //Mimimum number of teeth in contact
//Results:
printf("\n\n Least pressure angle to avoid interference, phi = %.3f degrees.\n\n",phi)
printf(" Length of arc of contact = %.2f mm.\n\n",Lac)
printf(" Minimum number of teeth in contact = %d or %d pair.\n\n",nmin+1,(nmin+1)/2)
```

**Scilab code Exa 12.15** To find axial thrust

```
//To find axial thrust
clc
//Given:
L=175/1000, d2=100/1000, r2=d2/2 //m
theta=70 //degrees
G=1.5, T2=80
Tf=75 //Torque on faster wheel, N-m
funcprot(0)
//Solution:
//Spiral angles for each wheel:
```

```
//Calculating the number of teeth on slower wheel
T1=T2*G
//Calculating the pitch circle diameter of the slower wheel
d1=(L*2)-d2 //m
//Calculating the spiral angles
//We have, d2/d1 = (T2*cos(alpha1))/(T1*cos(alpha2)), or T2*d1*cos(alpha1)-T1*d2*cos(alpha2) = 0 .....(i)
//Also, alpha1+alpha2 = theta, or alpha1+alpha2-theta = 0 .....(ii)
function y=f(x)
    alpha1=x(1)
    alpha2=x(2)
    y(1)=T2*d1*cos(alpha1)-T1*d2*cos(alpha2)
    y(2)=alpha1+alpha2-theta*%pi/180
endfunction
z=fsolve([1,1],f)
alpha1=z(1)*180/%pi //Spiral angle for slower wheel, degrees
alpha2=z(2)*180/%pi //Spiral angle for faster wheel, degrees
//Axial thrust on each shaft:
//Calculating the tangential force at faster wheel
F2=Tf/r2 //N
//Calculating the normal reaction at the point of contact
RN=F2/cosd(alpha2) //N
//Calculating the axial thrust on the shaft of slower wheel
Fa1=RN*sind(alpha1) //N
//Calculating the axial thrust on the shaft of faster wheel
Fa2=RN*sind(alpha2) //N
//Results:
printf("\n\n Spiral angle for slower wheel, alpha1 =
```

```
   %.2f degrees.\n\n",alpha1)
printf(" Spiral angle for faster wheel, alpha2 = %.2
   f degrees.\n\n",alpha2)
printf(" Axial thrust on the shaft of slower wheel,
   Fa1= %d N.\n\n",Fa1+1)
printf(" Axial thrust on the shaft of faster wheel,
   Fa2 = %d N.\n\n",Fa2+1)
```

**Scilab code Exa 12.16** To find teeth distance and efficiency

```
//To find teeth, distance and efficiency
clc
//Given:
L=400/1000 //m
G=3
theta=50, phi=6 //degrees
pN=18 //mm
//Solution:
//Number of teeth on each wheel:
//Calculating the spiral angles of the driving and
   driven wheels
alpha1=theta/2 //degrees
alpha2=alpha1 //degrees
//Calculating the number of teeth on driver wheel
T1=L*1000*2*%pi/(pN*(1/cosd(alpha1)+G/cosd(alpha2)))
//Calculating the number of teeth on driven wheel
T2=G*T1
//Calculating the exact centre distance
L1=pN*T1/(2*%pi)*(1/cosd(alpha1)+G/cosd(alpha2)) //
   mm
//Calculating the efficiency of the drive
eta=(cosd(alpha2+phi)*cosd(alpha1))/(cosd(alpha1-phi
   )*cosd(alpha2))*100 //%
//Results:
printf("\n\n Number of teeth on driver wheel, T1 =
```

```
    %d.\n\n",T1+1)
printf(" Number of teeth on driven wheel, T2 = %d.\n
    \n", T2+1)
printf(" Exact centre distance, L1 = %.1f mm.\n\n",
    L1)
printf(" Efficiency of the drive, eta = %.1f %c.\n\n
    ",eta,"%")
```

**Scilab code Exa 12.17** To find angle teeth and efficiency

```
//To find angle, teeth and efficiency
clc
//Given:
pN=12.5, L=134 //mm
theta=80, phi=6 //degrees
G=1.25
//Solution:
funcprot(0)
//Spiral angle of each wheel:
//Calculating the spiral angles of wheels 1 and 2
//We have, d2/d1 = (T2*cos(alpha1))/(T1*cos(alpha2))
    , or cos(alpha1)-G*cos(alpha2) = 0
    .....(i)
//Also, alpha1+alpha2 = theta, or alpha1+alpha2-
    theta = 0
                                                  .....(
    ii)
function y=f(x)
    alpha1=x(1)
    alpha2=x(2)
    y(1)=cos(alpha1)-G*cos(alpha2)
    y(2)=alpha1+alpha2-theta*%pi/180
endfunction
z=fsolve([1,1],f)
alpha1=z(1)*180/%pi //Spiral angle for slower wheel,
```

```
degrees
alpha2=z(2)*180/%pi //Spiral angle for faster wheel, degrees
//Number of teeth on each wheel:
//Calculating the diameters of the wheels
d1=L, d2 = d1 //mm
//Calculating the number of teeth on wheel 1
T1=d1*%pi*cosd(alpha1)/pN
//Calculating the number of teeth on wheel 2
T2=T1/G
//Calculating the efficiency of the drive
eta=(cosd(alpha2+phi)*cosd(alpha1))/(cosd(alpha1-phi)*cosd(alpha2))*100 //%
//Calculating the maximum efficiency
etamax=(cosd(theta+phi)+1)/(cosd(theta-phi)+1)*100 //%
//Results:
printf("\n\n Spiral angle for slower wheel, alpha1 = %.2f degrees.\n\n",alpha1)
printf(" Spiral angle for faster wheel, alpha2 = %.2f degrees.\n\n",alpha2)
printf(" Number of teeth on wheel 1, T1 = %d.\n\n",T1+1)
printf(" Number of teeth on wheel 2, T2 = %d.\n\n",T2+1)
printf(" Efficiency of the drive, eta = %d %c.\n\n",eta+1,"%")
printf(" Maximum efficiency, etamax = %.1f %c.\n\n",etamax,"%")
```

# Chapter 13

# Gear Trains

**Scilab code Exa 13.1** To find speed of gear F

```
//To find speed of gear F
clc
//Given:
NA=975 //rpm
TA=20, TB=50, TC=25, TD=75, TE=26, TF=65
//Solution:
//Calculating the speed of gear F
NF=NA*(TA*TC*TE)/(TB*TD*TF) //rpm
//Results:
printf("\n\n Speed of gear F, NF = %d rpm.\n\n",NF)
```

**Scilab code Exa 13.2** To design the gears

```
//To design the gears
clc
//Given:
x=600, pc=25 //mm
N1=360, N2=120 //rpm
```

```
//Solution:
//Calculating the pitch circle diameters of each gear
//Speed ratio, N1/N2 = d2/d1, or N1*d1-N2*d2 = 0                    .....(i)
//Centre distance between the shafts, x = 1/2*(d1+d2), or d1+d2 = 600*2          .....(ii)
A=[N1 -N2; 1 1]
B=[0; 600*2]
V=A \ B
d1=V(1) //mm
d2=V(2) //mm
//Calculating the number of teeth on the first gear
T1=round(%pi*d1/pc)
//Calculating the number of teeth on the second gear
T2=int(%pi*d2/pc+1)
//Calculating the pitch circle diameter of the first gear
d1dash=T1*pc/%pi //mm
//Calculating the pitch circle diameter of the second gear
d2dash=T2*pc/%pi //mm
//Calculating the exact distance between the two shafts
xdash=(d1dash+d2dash)/2 //mm
//Results:
printf("\n\n The number of teeth on the first and second gear must be %d and %d and their pitch circle diameters must be %.2f mm and %.1f mm respectively.\n\n",T1,T2,d1dash,d2dash)
printf(" The exact distance between the two shafts must be %.2f mm.\n\n",xdash)
```

**Scilab code Exa 13.3** To find the number of teeth

```
//To find the number of teeth
clc
//Given:
rAD=12 //Speed ratio , NA/ND
mA=3.125 , mB=mA , mC=2.5 , mD=mC , x=200 //mm
//Solution:
//Calculating the speed ratio between the gears A
    and B, and C and D
rAB=sqrt(rAD) //Speed ratio between the gears A and
    B
rCD=sqrt(rAB) //Speed ratio between the gears C and
    D
//Calculating the ratio of teeth on gear B to gear A
rtBA=rAB //Ratio of teeth on gear B to gear A
//Calculating the ratio of teeth on gear D to gear C
rtDC=rCD //Ratio of teeth on gear D to gear C
//Calculating the number of teeth on the gears A and
     B
//Distance between the shafts , x = mA*TA/2+mB*TB/2,
    or (mA/2)*TA+(mB/2)*TB = x          .....(i)
//Ratio of teeth on gear B to gear A, TB/TA = sqrt
    (12), or sqrt(12)*TA-TB = 0          .....(ii)
A=[mA/2 mB/2; sqrt(12) -1]
B=[x; 0]
V=A \ B
TA=int(V(1))
TB=round(V(2))
//Calculating the number of teeth on the gears C and
     D
//Distance between the shafts , x = mC*TC/2+mD*TD/2,
    or (mC/2)*TC+(mD/2)*TD = x          .....(iii)
//Ratio of teeth on gear D to gear C, TD/TC = sqrt
    (12), or sqrt(12)*TC-TD = 0          .....(iv)
A=[mC/2 mD/2; sqrt(12) -1]
B=[x; 0]
V=A \ B
TC=round(V(1))
TD=int(V(2))
```

```
//Results:
printf("\n\n Number of teeth on gear A, TA = %d.\n\n
    ",TA)
printf(" Number of teeth on gear B, TB = %d.\n\n",TB
    )
printf(" Number of teeth on gear C, TC = %d.\n\n",TC
    )
printf(" Number of teeth on gear D, TD = %d.\n\n",TD
    )
```

**Scilab code Exa 13.4** To find speed of gear B

```
//To find speed of gear B
clc
//Given:
TA=36, TB=45
NC=150 //rpm, anticlockwise
//Solution:
//Refer Fig. 13.7
//Algebraic method:
//Calculating the speed of gear B when gear A is
    fixed
NA=0, NC=150 //rpm
NB1=(-TA/TB)*(NA-NC)+NC //rpm
//Calculating the speed of gear B when gear A makes
    300 rpm clockwise
NA=-300 //rpm
NB2=(-TA/TB)*(NA-NC)+NC //rpm
//Results:
printf("\n\n Speed of gear B when gear A is fixed,
    NB = %d rpm.\n\n",NB1)
printf(" Speed of gear B when gear A makes 300 rpm
    clockwise, NB = %d rpm.\n\n",NB2)
```

**Scilab code Exa 13.5** To find speed of gear C

```
//To find speed of gear C
clc
//Given:
TB=75, TC=30, TD=90
NA=100 //rpm, clockwise
//Solution:
//Refer Table 13.3
//Calculating the number of teeth on gear E
TE=TC+TD-TB
//Calculating the speed of gear C
y=-100
x=y*(TB/TE)
NC=y-x*(TD/TC) //rpm
//Results:
printf("\n\n Speed of gear C, NC = %d rpm,
  anticlockwise.\n\n",NC)
```

---

**Scilab code Exa 13.6** To find speed of gears B and C

```
//To find speed of gears B and C
clc
//Given:
TA=72, TC=32
NEF=18 //Speed of arm EF, rpm
//Solution:
//Refer Table 13.5
//Speed of gear C:
y=18 //rpm
x=y*(TA/TC)
NC=x+y //Speed of gear C, rpm
```

```
//Speed of gear B:
//Calculating the number of teeth on gear B
TB=(TA-TC)/2
//Calculating the speed of gear B
NB=y-x*(TC/TB) //Speed of gear B, rpm
//Solution:
printf("\n\n Speed of gear C = %.1f rpm.\n\n",NC)
printf(" Speed of gear B = %.1f rpm in the opposite direction of arm.\n\n",-NB)
```

**Scilab code Exa 13.7** To find revolutions of arm

```
//To find revolutions of arm
clc
//Given:
TA=40, TD=90
//Solution:
//Calculating the number of teeth on gears B and C
//From geometry of the Fig. 13.11, dA+2*dB=dD.
//Since the number of teeth are proportional to their pitch circle diameters,
TB=(TD-TA)/2
TC=TB
//Refer Table 13.6
//Speed of arm when A makes 1 revolution clockwise and D makes half revolution anticlockwise:
//Calculating the values of x and y
//From the fourth row of the table, -x-y = -1, or x+y = 1          .....(i)
//The gear D makes half revolution anticlockwise, i.e., x*(TA/TD)-y = 1/2          .....(ii)
A=[1 1; TA/TD -1]
B=[1; 1/2]
V=A \ B
x=V(1)
```

```
y=V(2)
//Calculating the speed of arm
varm=-y //Speed of arm, revolutions
//Results:
printf("\n\n Speed of arm when A makes 1 revolution
   clockwise and D makes half revolution
   anticlockwise = %.2f revolution anticlockwise.\n\
   n",varm)
//Speed of arm when A makes 1 revolution clockwise
   and D is stationary:
//Calculating the values of x and y
//From the fourth row of the table, −x−y = −1, or x+
   y = 1                                   .....(iii)
//The gear D is stationary, i.e., x*(TA/TD)−y = 0
                                           .....(iv)
A=[1 1; TA/TD -1]
B=[1; 0]
V=A \ B
x=V(1)
y=V(2)
//Calculating the speed of arm
varm=-y //Speed of arm, revolutions
//Results:
printf(" Speed of arm when A makes 1 revolution
   clockwise and D is stationary = %.3f revolution
   clockwise.\n\n",-varm)
```

**Scilab code Exa 13.8** To find teeth and speed

```
//To find teeth and speed
clc
//Given:
TC=28, TD=26, TE=18, TF=TE
//Solution:
//The sketch is as in Fig. 13.12
```

```
//Number of teeth on wheels A and B:
//From geometry, dA = dC+2*dE, and dB = dD+2*dF
//Since the number of teeth are proportional to their pitch circle diameters,
TA=TC+2*TE
TB=TD+2*TF
//Speed of wheel B when arm G makes 100 rpm clockwise and wheel A is fixed:
//Since the arm G makes 100 rpm clockwise, therefore from the fourth row of Table 13.7,
y=-100
x=-y
//Calculating the speed of wheel B
NB1=y+x*(TA/TC)*(TD/TB) //Speed of wheel B when arm G makes 100 rpm clockwise and wheel A is fixed, rpm
//Speed of wheel B when arm G makes 100 rpm clockwise and wheel A makes 10 rpm counter clockwise:
//Since the arm G makes 100 rpm clockwise, therefore from the fourth row of Table 13.7,
y=-100
x=10-y
//Calculating the speed of wheel B
NB2=y+x*(TA/TC)*(TD/TB) //Speed of wheel B when arm G makes 100 rpm clockwise and wheel A makes 10 rpm counter clockwise, rpm
//Solution:
printf("\n\n Number of teeth on wheel A, TA = %d.\n\n",TA)
printf(" Number of teeth on wheel B, TB = %d.\n\n",TB)
printf(" Speed of wheel B when arm G makes 100 rpm clockwise and wheel A is fixed = %.1f rpm, clockwise.\n\n",-NB1)
printf(" Speed of wheel B when arm G makes 100 rpm clockwise and wheel A makes 10 rpm counter clockwise = %.1f rpm, counter clockwise.\n\n",NB2
```

```
)
```

**Scilab code Exa 13.9** To find number of teeth

```
//To find number of teeth
clc
//Given:
dD=224, m=4 //mm
//Solution:
//Refer Table 13.8
//Calculating the values of x and y
y=+1
x=+5-y
//Calculating the number of teeth on gear D
TD=dD/m
//Calculating the number of teeth on gear B
TB=y/x*TD
//Calculating the number of teeth on gear C
TC=(TD-TB)/2
//Results:
printf("\n\n Number of teeth on gear D, TD = %d.\n\n",TD)
printf(" Number of teeth on gear B, TB = %d.\n\n",TB)
printf(" Number of teeth on gear C, TC = %d.\n\n",TC)
```

**Scilab code Exa 13.10** To find angular velocities

```
//To find angular velocities
clc
//Given:
TC=50, TD=20, TE=35
```

```
NA=110 //rpm
//Solution:
//Calculating the number of teeth on internal gear G
TG=TC+TD+TE
//Speed of shaft B:
//Calculating the values of x and y
//From the fourth row of Table 13.9, y-x*(TC/TD)*(TE
    /TG) = 0          .....(i)
//Also, x+y = 110, or y+x = 110
                                                .....(ii)
A=[1 -(TC/TD)*(TE/TG); 1 1]
B=[0; 110]
V=A \ B
x=V(2)
y=V(1)
//Calculating the speed of shaft B
NB=round(+y) //Speed of shaft B, rpm
//Results:
printf("\n\n Number of teeth on internal gear G, TG
    = %d.\n\n",TG)
printf(" Speed of shaft B = %d rpm, anticlockwise.\n
    \n",NB)
```

**Scilab code Exa 13.11** To find angular velocities

```
//To find angular velocities
clc
//Given:
TA=12, TB=30, TC=14
NA=1, ND=5 //rps
//Solution:
//Number of teeth on wheels D and E:
//Calculating the number of teeth on wheel E
TE=TA+2*TB
//Calculating the number of teeth on wheel E
```

```
TD=TE-(TB-TC)
//Magnitude and direction of angular velocities of arm OP and wheel E:
//Calculating the values of x and y
//From the fourth row of Table 13.10, -x-y = -1, or x+y = 1                .....(i)
//Also, x*(TA/TB)*(TC/TD)-y = 5                .....(ii)
A=[1 1; (TA/TB)*(TC/TD) -1]
B=[1; 5]
V=A \ B
x=V(1)
y=V(2)
//Calculating the angular velocity of arm OP
omegaOP=-y*2*%pi //Angular velocity of arm OP, rad/s
//Calculating the angular velocity of wheel E
omegaE=(x*TA/TE-y)*2*%pi //Angular velocity of wheel E, rad/s
//Results:
printf("\n\n Number of teeth on wheel E, TE = %d.\n\n",TE)
printf(" Number of teeth on wheel D, TD = %d.\n\n",TD)
printf(" Angular velocity of arm OP = %.3f rad/s, counter clockwise.\n\n",omegaOP)
printf(" Angular velocity of wheel E = %.2f rad/s, counter clockwise.\n\n",omegaE)
```

**Scilab code Exa 13.12** To find speed of shaft

```
//To find speed of shaft
clc
//Given:
TB=80, TC=82, TD=28
NA=500 //rpm
```

```
//Solution:
//Calculating the number of teeth on wheel E
TE=TB+TD-TC
//Calculating the values of x and y
y=800
x=-y*(TE/TB)*(TC/TD)
//Calculating the speed of shaft F
NF=x+y //Speed of shaft F, rpm
//Results:
printf("\n\n Speed of shaft F = %d rpm, anticlockwise.\n\n",NF)
```

**Scilab code Exa 13.14** To find number of teeth and speed

```
//To find number of teeth and speed
clc
//Given:
NA=300 //rpm
TD=40, TE=30, TF=50, TG=80, TH=40, TK=20, TL=30
//Solution:
//Refer Fig. 13.18 and Table 13.13
//Calculating the speed of wheel E
NE=NA*(TD/TE) //rpm
//Calculating the number of teeth on wheel C
TC=TH+TK+TL
//Speed and direction of rotation of shaft B:
//Calculating the values of x and y
//We have, -x-y = -400, or x+y = 400 .....(i)
//Also, x*(TH/TK)*(TL/TC)-y = 0 .....(ii)
A=[1 1; (TH/TK)*(TL/TC) -1]
B=[400; 0]
V=A \ B
x=V(1)
```

```
y=V(2)
//Calculating the speed of wheel F
NF=-y //rpm
//Calculating the speed of shaft B
NB=-NF*(TF/TG) //Speed of shaft B, rpm
//Results:
printf("\n\n Number of teeth on wheel C, TC = %d.\n\n",TC)
printf(" Speed of shaft B = %d rpm, anticlockwise.\n\n",NB)
```

**Scilab code Exa 13.15** To find velocity ratio

```
//To find velocity ratio
clc
//Given:
T1=80, T8=160, T4=100, T3=120, T6=20, T7=66
//Solution:
//Refer Fig. 13.19 and Table 13.14
//Calculating the number of teeth on wheel 2
T2=(T3-T1)/2
//Calculating the values of x and y
//Assuming that wheel 1 makes 1 rps anticlockwise, x+y = 1                .....(i)
//Also, y-x*(T1/T3) = 0, or x*(T1/T3)-y = 0                .....(ii)
A=[1 1; 1 T1/T3]
B=[1; 0]
V=A \ B
x=V(1)
y=V(2)
//Calculating the speed of casing C
NC=y //Speed of casing C, rps
//Calculating the speed of wheel 2
N2=y-x*(T1/T2) //Speed of wheel 2, rps
```

```
//Calculating the number of teeth on wheel 5
T5=(T4-T6)/2
//Calculating the values of x1 and y1
y1=-2
x1=(y1-0.4)*(T4/T6)
//Calculating the speed of wheel 6
N6=x1+y1 //Speed of wheel 6, rps
//Calculating the values of x2 and y2
y2=0.4
x2=-(14+y2)*(T7/T8)
//Calculating the speed of wheel 8
N8=x2+y2 //Speed of wheel 8, rps
//Calculating the velocity ratio of the output shaft
    B to the input shaft A
vr=N8/1 //Velocity ratio
//Results:
printf("\n\n Velocity ratio of the output shaft B to
    the input shaft A = %.2f.\n\n",vr)
```

**Scilab code Exa 13.16** To find speed of shaft

```
//To find speed of shaft
clc
//Given:
TA=40, TB=30, TC=50
NX=100, NA=NX //rpm
Narm=100 //Speed of arm, rpm
//Solution:
//Refer Fig. 13.22 and Table 13.18
//Calculating the values of x and y
y=+100
x=-100-y
//Calculating the speed of the driven shaft
NY=y-x*(TA/TB) //rpm
//Results:
```

```
printf("\n\n Speed of the driven shaft , NY = %.1f
   rpm, anticlockwise.\n\n",NY)
```

**Scilab code Exa 13.17** To find speed of output shaft

```
//To find speed of output shaft
clc
//Given:
TB=20, TC=80, TD=80, TE=30, TF=32
NB=1000 //rpm
//Solution:
//Refer Fig. 13.23 and Table 13.19
//Speed of the output shaft when gear C is fixed:
//Calculating the values of x and y
//From the fourth row of the table, y-x*(TB/TC) = 0
                    .....(i)
//Also, x+y = +1000, or y+x = 1000
                                        .....(ii)
A=[1 -TB/TC; 1 1]
B=[0; 1000]
V=A \ B
x=V(2)
y=V(1)
//Calculating the speed of output shaft
NF1=y-x*(TB/TD)*(TE/TF) //Speed of the output shaft
   when gear C is fixed, rpm
//Speed of the output shaft when gear C is rotated
   at 10 rpm counter clockwise:
//Calculating the values of x and y
//From the fourth row of te table, y-x*(TB/TC) = +10
                    .....(iii)
//Also, x+y = +1000, or y+x = 1000
                                        .....(iv)
A=[1 -TB/TC; 1 1]
B=[10; 1000]
```

```
V=A \ B
x=V(2)
y=V(1)
//Calculating the speed of output shaft
NF2=y-x*(TB/TD)*(TE/TF) //Speed of the output shaft when gear C is rotated at 10 rpm counter clockwise, rpm
//Results:
printf("\n\n Speed of the output shaft when gear C is fixed = %.1f rpm, counter clockwise.\n\n",NF1)
printf(" Speed of the output shaft when gear C is rotated at 10 rpm counter clockwise = %.1f rpm, counter clockwise.\n\n",NF2)
```

**Scilab code Exa 13.18** To find speed of road wheel

```
//To find speed of road wheel
clc
//Given:
TA=10, TB=60
NA=1000, NQ=210, ND=NQ //rpm
//Solution:
//Refer Fig. 13.24 and Table 13.20
//Calculating the speed of crown gear B
NB=NA*(TA/TB) //rpm
//Calculating the values of x and y
y=200
x=y-210
//Calculating the speed of road wheel attached to axle P
NC=x+y //Speed of road wheel attached to axle P, rpm
//Results:
printf("\n\n Speed of road wheel attached to axle P = %d rpm.\n\n",NC)
```

**Scilab code Exa 13.19** To find torque exerted

```
//To find torque exerted
clc
//Given:
TA=15, TB=20, TC=15
NA=1000 //rpm
Tm=100 //Torque developed by motor, N-m
//Solution:
//Refer Fig. 13.26 and Table 13.21
//Calculating the number of teeth on gears E and D
TE=TA+2*TB
TD=TE-(TB-TC)
//Speed of the machine shaft:
//From the fourth row of the table, x+y = 1000, or y
    +x = 1000          .....(i)
//Also, y-x*(TA/TE) = 0

    .....(ii)
A=[1 1; 1 -TA/TE]
B=[1000; 0]
V=A \ B
y=V(1)
x=V(2)
//Calculating the speed of machine shaft
ND=y-x*(TA/TB)*(TC/TD) //rpm
//Calculating the torque exerted on the machine
    shaft
Ts=Tm*NA/ND //Torque exerted on the machine shaft, N
    -m
//Results:
printf("\n\n Speed of machine shaft, ND = %.2f rpm,
    anticlockwise.\n\n",ND)
printf(" Torque exerted on the machine shaft = %d N-
```

```
m.\n\n",Ts)
```

**Scilab code Exa 13.20** To find teeth and torque

```
//To find teeth and torque
clc
//Given:
Ts=100 //Torque on the sun wheel, N-m
r=5 //Ratio of speeds of gear S to C, NS/NC
//Refer Fig. 13.27 and Table 13.22
//Number of teeth on different wheels:
//Calculating the values of x and y
y=1
x=5-y
//Calculating the number of teeth on wheel E
TS=16
TE=4*TS
//Calculating the number of teeth on wheel P
TP=(TE-TS)/2
//Torque necessary to keep the internal gear
    stationary:
Tc=Ts*r //Torque on CN-m
//Caluclating the torque necessary to keep the
    internal gear stationary
Ti=Tc-Ts //Torque necessary to keep the internal
    gear stationary, N-m
//Results:
printf("\n\n Number of teeth on different wheels, TE
     = %d.\n\n",TE)
printf(" Torque necessary to keep the internal gear
    stationary = %d N-m.\n\n",Ti)
```

**Scilab code Exa 13.21** To find speed direction and torque

```
//To find speed, direction and torque
clc
//Given:
TA=14, TC=100
r=98/41 //TE/TD
PA=1.85*1000 //W
NA=1200 //rpm
//Solution:
//Refer Fig. 13.28 and Table 13.23
//Calculating the number of teeth on wheel B
TB=(TC-TA)/2
//Calculating the values of x and y
//From the fourth row of the table, -y+x*(TA/TC) = 0, or x*(TA/TC)-y = 0     .....(i)
//Also, -x-y = 1200, or x+y = -1200                                          .....(ii)
A=[TA/TC -1; 1 1]
B=[0; -1200]
V=A \ B
x=V(1)
y=V(2)
//Calculating the speed of gear E
NE=round(-y+x*(TA/TB)*(1/r)) //rpm
//Fixing torque required at C:
//Calculating the torque on A
Ta=PA*60/(2*%pi*NA) //Torque on A, N-m
//Calculating the torque on E
Te=PA*60/(2*%pi*NE) //Torque on E
//Calculating the fixing torque required at C
Tc=Te-Ta //Fixing torque at C, N-m
//Results:
printf("\n\n Speed and direction of rotation of gear E, NE = %d rpm, anticlockwise.\n\n",NE)
printf(" Fixing torque required at C = %.1f N-m.\n\n",Tc)
```

**Scilab code Exa 13.22** To find holding torque

```
//To find holding torque
clc
//Given:
TB=15, TA=60, TC=20
omegaY=740, omegaA=omegaY //rad/s
P=130*1000 //W
//Solution:
//Refer Fig. 13.29 and Table 13.24
//Calculating the number of teeth on wheel D
TD=TA-(TC+TB)
//Calculating the values of x and y
//From the fourth row of the table, y-x*(TD/TC)*(TB/
    TA) = 740                .....(i)
//Also, x+y = 0, or y+x = 0

     .....(ii)
A=[1 -(TD/TC)*(TB/TA); 1 1]
B=[740; 0]
V=A \ B
x=V(2)
y=V(1)
//Calculating the speed of shaft X
omegaX=y //rad/s
//Holding torque on wheel D:
//Calculating the torque on A
Ta=P/omegaA //Torque on A, N-m
//Calculating the torque on X
Tx=P/omegaX //Torque on X, N-m
//Calculating the holding torque on wheel D
Td=Tx-Ta //Holding torque on wheel D, N-m
//Results:
printf("\n\n Speed of shaft X, omegaX = %.1f rad/s.\
```

```
    n\n",omegaX)
printf(" Holding torque on wheel D = %.1f N-m.\n\n",
    Td)
```

**Scilab code Exa 13.23** To find speed direction and torque

```
//To find speed, direction and torque
clc
//Given:
TP=144, TQ=120, TR=120, TX=36, TY=24, TZ=30
NI=1500 //rpm
P=7.5*1000 //W
eta=0.8
//Solution:
//Refer Fig. 13.30 and Table 13.25
//Calculating the values of x and y
//From the fourth row of the table, x+y = -1500
          .....(i)
//Also, y-x*(TZ/TR) = 0, or -x*(TZ/TR)+y = 0
              .....(ii)
A=[1 1; -TZ/TR 1]
B=[-1500; 0]
V=A \ B
x=V(1)
y=V(2)
//Calculating the values of x1 and y1
//We have, y1-x1*(TY/TQ) = y
                                  .....(iii)
//Also, x1+y1 = x+y, or y1+x1 = x+y
                         .....(iv)
A=[1 -TY/TQ; 1 1]
B=[y; x+y]
V=A \ B
x1=V(2)
y1=V(1)
```

```
//Speed and direction of the driven shaft O and the
    wheel P:
//Calculating the speed of shaft O
NO=y1 //rpm
//Calculating the speed of wheel P
NP=y1+x1*(TY/TQ)*(TX/TP) //rpm
//Torque tending to rotate the fixed wheel R:
//Calculating the torque on shaft I
T1=P*60/(2*%pi*NI) //N-m
//Calculating the torque on shaft O
T2=eta*P*60/(2*%pi*(-NO)) //N-m
//Calculating the torque tending to rotate the fixed
     wheel R
T=T2-T1 //Torque tending to rotate the fixed wheel R
    , N-m
//Results:
printf("\n\n Speed of the driven shaft O, NO = %d
    rpm, clockwise.\n\n",-NO)
printf(" Speed of the wheel P, NP = %d rpm,
    clockwise.\n\n",-NP)
printf(" Torque tending to rotate the fixed wheel R
    = %.2f N-m.\n\n",T)
```

**Scilab code Exa 13.24** To find torque and forces

```
//To find torque and forces
clc
//Given:
TA=34, TB=120, TC=150, TD=38, TE=50
PX=7.5*1000 //W
NX=500 //rpm
m=3.5 //mm
//Solution:
//Refer Fig. 13.31 and Table 13.27
//Output torque of shaft Y:
```

```
//Calculating the values of x and y
//From the fourth row of the table, x+y = 500, or y+x = 500          .....(i)
//Also, y-x*(TA/TC) = 0                                                  .....(ii)
A=[1 1; 1 -TA/TC]
B=[500; 0]
V=A \ B
y=V(1) //rpm
x=V(2) //rpm
//Calculating the speed of output shaft Y
NY=y-x*(TA/TB)*(TD/TE) //rpm
//Calculating the speed of wheel E
NE=NY //rpm
//Calculating the input power assuming 100 per cent efficiency
PY=PX //W
//Calculating the output torque of shaft Y
Ty=PY*60/(2*%pi*NY*1000) //Output torque on shaft Y, kN-m
//Tangential force between wheels D and E:
//Calculating the pitch circle radius of wheel E
rE=m*TE/(2*1000) //m
//Calculating the tangential force between wheels D and E
FtDE=Ty/rE //Tangential force between wheels D and E, kN
//Tangential force between wheels B and C:
//Calculating the input torque on shaft X
Tx=PX*60/(2*%pi*NX) //Input torque on shaft X, N-m
//Calculating the fixing torque on the fixed wheel C
Tf=Ty-Tx/1000 //Fixing torque on the fixed wheel C, kN-m
//Calculating the pitch circle radius of wheel C
rC=m*TC/(2*1000) //m
//Calculating the tangential forces between wheels B and C
```

```
FtBC=Tf/rC //kN
//Results:
printf("\n\n Output torque of shaft Y = %.3f kN-m.\n\n",Ty)
printf(" Tangential force between wheels D and E = %.1f kN.\n\n",FtDE)
printf(" Tangential force between wheels B and C = %d kN.\n\n",FtBC)
```

# Chapter 14

# Gyroscopic Couple and Precessional Motion

**Scilab code Exa 14.1** To find speed of precession

```
//To find speed of precession
clc
//Given:
d=300/1000, r=d/2, l=600/1000 //m
m=5 //kg
N=300 //rpm
//Solution:
//Calculating the angular speed of the disc
omega=2*%pi*N/60 //rad/s
//Calculating the mass moment of inertia of the disc, about an axis through its centre of gravity and perpendicular to the plane of the disc
I=m*r^2/2 //kg-m^2
//Calculating the couple due to mass of disc
C=m*9.81*l //N-m
//Calculating the speed of precession
omegaP=C/(I*omega) //rad/s
//Results:
printf("\n\n Speed of precession, omegaP = %.1f rad/
```

```
s.\n\n",omegaP)
```

---

**Scilab code Exa 14.2** To find the resultant reaction

```
//To find the resultant reaction
clc
//Given:
d=150/1000, r=d/2, x=100/1000 //m
m=5 //kg
N=1000, NP=60 //rpm
//Solution:
//Calculating the angular speed of the disc
omega=2*%pi*N/60 //rad/s
//Calculating the speed of precession of the axle
omegaP=2*%pi*NP/60 //rad/s
//Calculating the mass moment of inertia of the disc, about an axis through its centre of gravity and perpendicular to the plane of disc
I=m*r^2/2 //kg-m^2
//Calculating the gyroscopic couple acting on the disc
C=I*omega*omegaP //N-m
//Calculating the force at each bearing due to the gyroscopic couple
F=C/x //N
//Calculating the reactions at the bearings A and B
RA=m/2*9.81 //N
RB=RA //N
//Resultant reaction at each bearing:
//Calculating the resultant reaction at the bearing A
RA1=F+RA //N
//Calculating the resultant reaction at the bearing B
RB1=F-RB //N
```

```
//Results:
printf("\n\n Resultant reaction at the bearing A,
   RA1 = %.1f N, upwards.\n\n",RA1)
printf(" Resultant reaction at the bearing B, RB1 =
   %.1f N, downwards.\n\n", RB1)
```

**Scilab code Exa 14.3** To find gyroscopic couple

```
//To find gyroscopic couple
clc
//Given:
R=50, k=0.3 //m
v=200*1000/3600 //m/s
m=400 //kg
N=2400 //rpm
//Solution:
//Calculating the angular speed of the engine
omega=2*%pi*N/60 //rad/s
//Calculating the mass moment of inertia of the
   engine and the propeller
I=m*k^2 //kg-m^2
//Calculating the angular velocity of precession
omegaP=v/R //rad/s
//Calculating the gyroscopic couple acting on the
   aircraft
C=I*omega*omegaP/1000 //kN-m
//Results:
printf("\n\n Gyroscopic couple acting on the
   aircraft, C = %.3f kN-m.\n\n",C)
printf(" The effect of the gyroscopic couple is to
   lift the nose upwards and tail downwards.\n\n")
```

**Scilab code Exa 14.4** To find gyroscopic couple

```
//To find gyroscopic couple
clc
//Given:
m=8*1000 //kg
k=0.6, R=75 //m
N=1800 //rpm
v=100*1000/3600 //m/s
//Solution:
//Calculating the angular speed of the rotor
omega=2*%pi*N/60 //rad/s
//Calculating the mass moment of inertia of the rotor
I=m*k^2 //kg-m^2
//Calculating the angular velocity of precession
omegaP=v/R //rad/s
//Calculating the gyroscopic couple
C=I*omega*omegaP/1000 //kN-m
//Results:
printf("\n\n Gyroscopic couple, C = %.3f kN-m.\n\n", C)
```

**Scilab code Exa 14.5** To find gyroscopic couple and direction

```
//To find gyroscopic couple and direction
clc
//Given:
N=1500 //rpm
m=750 //kg
omegaP=1 //rad/s
k=250/1000 //m
//Solution:
//Calculating the angular speed of the rotor
omega=2*%pi*N/60 //rad/s
//Calculating the mass moment of inertia of the rotor
```

```
I=m*k^2 //kg-m^2
//Calculating the gyroscopic couple transmitted to
    the hull
C=I*omega*omegaP/1000 //kN-m
//Results:
printf("\n\n Gyroscopic couple transmitted to the
    hull, C = %.3f kN-m.\n\n",C)
printf(" When the pitching is upward, the relative
    gyroscopic couple acts in the clockwise direction
    .\n\n")
```

**Scilab code Exa 14.6** To find gyroscopic couple and effect

```
//To find gyroscopic couple and effect
clc
//Given:
m=3500 //kg
k=0.45 //m
N=3000 //rpm
//Solution:
//Calculating the angular speed of the rotor
omega=2*%pi*N/60 //rad/s
//When the ship is steering to the left:
R=100 //m
v=36*1000/3600 //m/s
//Calculating the mass moment of inertia of the
    rotor
I=m*k^2 //kg-m^2
//Calculating the angular velocity of precession
omegaP=v/R //rad/s
//Calculating the gyroscopic couple
C=I*omega*omegaP/1000 //kN-m
//Results:
printf("\n\n Gyroscopic couple when the ship is
    steering to the left, C = %.2f kN-m.\n\n",C)
```

```
printf(" When the rotor rotates clockwise and the
    ship takes a left turn, the effect of the
    reactive gyroscopic couple is to raise the bow
    and lower the stern.\n\n")
//When the ship is pitching with the bow falling:
tp=40 //s
//Calculating the amplitude of swing
phi=12/2*%pi/180 //rad
//Calculating the angular velocity of the simple
    harmonic motion
omega1=2*%pi/tp //rad/s
//Calculating the maximum angular velocity of
    precession
omegaP=phi*omega1 //rad/s
//Calculating the gyroscopic couple
C=I*omega*omegaP/1000 //kN-m
//Results:
printf(" Gyroscopic couple when the ship is pitching
     with the bow falling, C = %.3f kN-m.\n\n",C)
printf(" When the bow is falling, the effect of the
    reactive gyroscopic couple is to move the ship
    towards port side.\n\n")
```

**Scilab code Exa 14.7** To find gyroscopic couple and acceleration

```
//To find gyroscopic couple and acceleration
clc
//Given:
m=20*1000 //kg
k=0.6 //m
N=2000 //rpm
phi=6*%pi/180 //rad
tp=30 //s
//Solution:
//Calculating the angular speed of the rotor
```

```
omega=2*%pi*N/60 //rad/s
//Maximum gyroscopic couple:
//Calculating the mass moment of inertia of the
    rotor
I=m*k^2 //kg-m^2
//Calculating the angular velocity of the simple
    harmonic motion
omega1=2*%pi/tp //rad/s
//Calculating the maximum angular velocity of
    precession
omegaPmax=phi*omega1 //rad/s
//Calculating the maximum gyroscopic couple
Cmax=I*omega*omegaPmax/1000 //kN-m
//Calculating the maximum angular acceleration
    during pitching
alphamax=phi*omega1^2 //Maximum angular acceleration
     during pitching, rad/s^2
//Results:
printf("\n\n Maximum gyroscopic couple, Cmax = %.3f
    kN-m.\n\n",Cmax)
printf(" Maximum angular acceleration during
    pitching = %.4f rad/s^2.\n\n",alphamax)
printf(" When the rotation of the rotor is clockwise
     when looking from the left and when the bow is
    rising, then the reactive gyroscopic couple acts
    in the direction which tends to turn the bow
    towrds right.\n\n")
```

**Scilab code Exa 14.8** To find the gyroscopic effects

```
//To find the gyroscopic effects
clc
//Given:
m=5*1000 //kg
N=1000 //rpm
```

```
k=0.5 //m
//Solution:
//Calculating the angular speed of the rotor
omega=2*%pi*N/60 //rad/s
//When the ship steers to the left:
v=30*1000/3600 //m/s
R=60 //m
//Calculating the angular velocity of precession
omegaP=v/R //rad/s
//Calculating the mass moment of inertia of the rotor
I=m*k^2 //kg-m^2
//Calculating the gyroscopic couple
C=I*omega*omegaP/1000 //kN-m
//Results:
printf("\n\n When the rotor rotates in a clockwise direction when viewed from from the stern and the ship steers to the left , the effect of reactive gyroscopic couple is to raise the bow and lower the stern.\n\n")
//When the ship pitches with the bow descending:
phi=6*%pi/180 //rad
tp=20 //s
//Calculating the angular velocity of simple harmonic motion
omega1=2*%pi/tp //rad/s
//Calculating the maximum velocity of precession
omegaPmax=phi*omega1 //rad/s
//Calculating the maximum gyroscopic couple
Cmax=I*omega*omegaPmax //N-m
//Results:
printf(" Since the ship is pitching with the low bow descending, therefore the effect of this maximum gyroscopic couple is to turn the ship towards port side.\n\n")
//When the ship rolls:
omegaP=0.03 //rad/s
//Calculating the gyroscopic couple
```

```
C=I*omega*omegaP //N-m
//Results:
printf(" In case of rolling of a ship, the axis of
   precession is always parallel to the axis of spin
    for all positions, therefore there is no effect
   of gyroscopic couple.\n\n")
//Calculating the maximum angular acceleration
   during pitching
alphamax=phi*omega1^2 //rad/s^2
//Results:
printf(" Maximum angular acceleration during
   pitching, alphamax = %.2f rad/s^2.\n\n",alphamax)
```

**Scilab code Exa 14.9** To find maximum acceleration during pitching

```
//To find maximum acceleration during pitching
clc
//Given:
m=2000 //kg
N=3000 //rpm
k=0.5, R=100 //m
v=16.1*1855/3600 //m/s
//Solution:
//Calculating the angular speed of the rotor
omega=2*%pi*N/60 //rad/s
//Gyroscopic couple:
//Calculating the mass moment of inertia of the
   rotor
I=m*k^2 //kg-m^2
//Calculating the angular velocity of precession
omegaP=v/R //rad/s
//Calculating the gyroscopic couple
C=I*omega*omegaP/1000 //kN-m
//Torque during pitching:
tp=50 //s
```

```
phi=12/2*%pi/180 //rad
//Calculating the angular velocity of simple harmonic motion
omega1=2*%pi/tp //rad/s
//Calculating the maximum angular velocity of precession
omegaPmax=phi*omega1 //rad/s
//Calculating the maximum gyroscopic couple during pitching
Cmax=I*omega*omegaPmax //N-m
//Calculating the maximum acceleration during pitching
alphamax=phi*omega1^2 //rad/s^2
//Results:
printf("\n\n When the rotor rotates clockwise when looking from a stern the ship steers to the right, the effect of the reactive gyroscopic couple is to raise the stern and lower the bow.\n\n")
printf(" Torque during pitching, Cmax = %d N-m.\n\n",Cmax)
printf(" Maximum acceleration during pitching, alphamax = %.5f rad/s^2.\n\n",alphamax)
```

**Scilab code Exa 14.10** To find centrifugal and gyroscopic effects

```
//To find centrifugal and gyroscopic effects
clc
//Given:
m=2500 //kg
x=1.5, R=30, dW=0.75, rW=dW/2, h=0.9 //m
v=24*1000/3600 //m/s
G=5
IW=18, IE=12 //kg-m^2
//Solution:
//Calculating the road reaction on each wheel
```

```
r=m*9.81/4 //Road reaction on each wheel, N
//Calculating the angular velocity o the wheels
omegaW=v/rW //rad/s
//Calculating the angular velocity of precession
omegaP=v/R //rad/s
//Calculating the gyroscopic couple due to one pair of wheels and axle
CW=round(2*IW*omegaW*omegaP) //N-m
//Calculating the gyroscopic couple due to the rotating parts of the motor and gears
CE=round(2*IE*G*omegaW*omegaP) //N-m
//Calculating the net gyroscopic couple
C=CW-CE //N-m
//Calculating the reaction due to gyroscopic couple at each of the outer or inner wheels
P=2*(-C)/(2*x) //N
//Calculating the centrifugal force
FC=m*v^2/R //N
//Calculating the overturning couple
CO=FC*h //N-m
//Calculating the reaction due to overturning couple at each of the outer and inner wheels
Q=2*CO/(2*x) //N
//Calculating the vertical force exerted on each outer wheel
PO=m*9.81/4-P/2+Q/2 //N
//Calculating the vertical force exerted on each inner wheel
PI=m*9.81/4+P/2-Q/2 //N
//Results:
printf("\n\n Vertical force exerted on each outer wheel, PO = %.2f N.\n\n",PO)
printf(" Vertical force exerted on each inner wheel, PI = %.2f N.\n\n",PI)
```

**Scilab code Exa 14.12** To find load on each wheel

```
//To find load on each wheel
clc
//Given:
m=2000, mE=75 //kg
b=2.5, x=1.5, h=500/1000, L=1, dW=0.8, rW=dW/2, kE=100/1000, R=60 //m
IW=0.8 //kg-m^2
G=4
v=60*1000/3600 //m/s
//Solution:
//Refer Fig. 14.12
//Calculating the weight on the rear wheels
W2=(m*9.81*1)/b //N
//Calculating the weight on the front wheels
W1=m*9.81-W2 //N
//Calculating the weight on each of the front wheels
Wf=W1/2 //Weight on each of the front wheels, N
//Calculating the weight on each of the rear wheels
Wr=W2/2 //Weight on each of the rear wheels, N
//Calculating the angular velocity of wheels
omegaW=v/rW //rad/s
//Calculating the angular velocity of precession
omegaP=v/R //rad/s
//Calculating the gyroscopic couple due to four wheels
CW=4*IW*omegaW*omegaP //N-m
//Calculating the magnitude of reaction due to gyroscopic couple due to four wheels at each of the inner or outer wheel
P=2*(CW/(2*x)) //N
//Calculating the mass moment of inertia of rotating parts of the engine
IE=mE*(kE)^2 //kg-m^2
//Calculating the gyroscopic couple due to rotating parts of the engine
CE=IE*(kE)^2*G*omegaW*omegaP //N-m
```

```
//Calculating the magnitude of reaction due to gyroscopic couple due to rotating parts of the engine at each of the inner or outer wheel
F=2*(CE/(2*b)) //N
//Calculating the centrifugal force
FC=m*v^2/R //N
//Calculating the centrifugal couple tending to overturn the car
CO=FC*h //N-m
//Calculating the magnitude of reaction due to overturning couple at each of the inner or outer wheel
Q=2*(CO/(2*x)) //N
//Calculating the load on front wheel 1
Fw1=W1/2-P/2-F/2-Q/2 //Load on front wheel 1, N
//Calculating the load on front wheel 2
Fw2=W1/2+P/2-F/2+Q/2 //Load on front wheel 2, N
//Calculating the load on rear wheel 3
Rw3=W2/2-P/2+F/2-Q/2 //Load on rear wheel 3, N
//Calculating the load on rear wheel 4
Rw4=W2/2+P/2+F/2+Q/2 //Load on rear wheel 4, N
//Results:
printf("\n\n Load on front wheel 1 = %.2f N.\n\n",
  Fw1)
printf(" Load on front wheel 2 = %.2f N.\n\n",Fw2)
printf(" Load on rear wheel 3 = %.2f N.\n\n",Rw3)
printf(" Load on rear wheel 4 = %.2f N.\n\n",Rw4)
```

**Scilab code Exa 14.13** To find pressure on each rail

```
//To find pressure on each rail
clc
//Given:
m=2000, mI=200 //kg
x=1.6, R=30, dW=0.7, rW=dW/2, k=0.3, h=1 //m
```

```
v=54*1000/3600 //m/s
theta=8 //degrees
//Solution:
//Refer Fig. 14.13
//Calculating the reactions at the wheels:
//Taking moments about B
RA=(m*9.81*cosd(theta)+m*v^2/R*sind(theta))*1/2+(m*9.81*sind(theta)-m*v^2/R*cosd(theta))*h/x //N
//Resolving the forces perpendicular to the track
RB=(m*9.81*cosd(theta)+m*v^2/R*sind(theta))-RA //N
//Calculating the angular velocity of wheels
omegaW=v/rW //rad/s
//Calculating the angular velocity of precession
omegaP=v/R //rad/s
//Calculating the gyroscopic couple
C=mI*k^2*omegaW*cosd(theta)*omegaP //N-m
//Calculating the force at each pair of wheels due to the gyroscopic couple
P=C/x //N
//Calculating the pressure on the inner rail
PI=RA-P //N
//Calculating the pressure o the outer rail
PO=RB+P //N
//Results:
printf("\n\n Pressure on the inner rail, PI = %.2f N.\n\n",PI)
printf(" Pressure on the outer rail, PO = %.2f N.\n\n",PO)
```

**Scilab code Exa 14.14** To find gyroscopic couple and reaction

```
//To find gyroscopic couple and reaction
clc
//Given:
I=180 //kg-m^2
```

```
D=1.8, R=D/2, x=1.5 //m
v=95*1000/3600 //m/s
t=0.1 //s
//Solution:
//Gyroscopic couple set up:
//Calculating the angular velocity of the locomotive
omega=v/R //rad/s
//Calculating the amplitude
A=1/2*6 //mm
//Calculating the maximum velocity while falling
vmax=2*%pi/t*A/1000 //m/s
//Calculating the maximum angular velocity of tilt
    of the axle or angular velocity of precession
omegaPmax=vmax/x //rad/s
//Calculating the gyroscopic couple set up
C=I*omega*omegaPmax //N-m
//Calculating the reaction between the wheel and
    rail due to the gyroscopic couple
P==C/x //N
//Results:
printf("\n\n Gyroscopic couple set up, C = %.1f N-m
    .\n\n",C)
printf(" Reaction between the wheel and rail due to
    the gyroscopic couple, P = %d N.\n\n",P)
```

**Scilab code Exa 14.15** To find angle of inclination

```
//To find angle of inclination
clc
//Given:
m=250 //kg
IE=0.3, IW=1 //kg-m^2
G=5
h=0.6, rW=300/1000, R=50 //m
v=90*1000/3600 //m/s
```

```
//Solution:
//Calculating the angle of inclination with respect
   to the vertical of a two wheeler
//Equating total overturning couple to balancing
   couple,
theta=atand((1/(m*9.81*h))*((v^2/(R*rW)*(2*IW+G*IE))
   +(m*v^2/R*h))) //degrees
//Results:
printf("\n\n Angle of inclination with respect to
   the vertical of a two wheeler, theta = %.2f
   degrees.\n\n",theta)
```

**Scilab code Exa 14.16** To find inclination of gyrowheel

```
//To find inclination of gyrowheel
clc
//Given:
m1=0.5, m2=0.3 //kg
k=20/1000, OG=10/1000, h=OG, R=50 //m
N=3000 //rpm
v=15 //m/s
//Solution:
//Refer Fig. 14.15 and Fig. 14.16
//Calculating the angular speed of the wheel
omega=2*%pi*N/60 //rad/s
//Calculating the mass moment of inertia of the
   gyrowheel
I=m1*k^2 //kg-m^2
//Calculating the angular velocity of precession
omegaP=v/R //rad/s
//When the vehicle moves in the direction of arrow X
    taking a left turn along the curve:
//Calculating the angle of inclination of the
   gyrowheel from the vertical
//Equating the overturning couple to the balancing
```

```
couple for equilibrium condition,
theta1=atand((1/(m2*9.81*h))*(I*omega*omegaP-m2*v^2/R*h)) //degrees
//When the vehicle reverses at the same speed in the direction of arrow Y along the same path:
//Calculating the angle of inclination of the gyrowheel from the vertical
//Equating the overturning couple to the balancing couple for equilibrium condition,
theta2=atand((1/(m2*9.81*h))*(I*omega*omegaP+m2*v^2/R*h)) //degrees
//Results:
printf("\n\n Angle of inclination of the gyrowheel from the vertical when the vehicle moves in the direction of arrow X taking a left turn along the curve, theta = %.2f degrees.\n\n",theta1)
printf(" Angle of inclination of the gyrowheel from the vertical when the vehicle reverses at the same speed in the direction of arrow Y along the same path, theta = %.2f degrees.\n\n",theta2)
```

**Scilab code Exa 14.17** To find the gyroscopic couple

```
//To find the gyroscopic couple
clc
//Given:
d=0.6, r=d/2 //m
m=30 //kg
theta=1 //degree
N=1200 //rpm
//Solution:
//Calculating the angular speed of the shaft
omega=2*%pi*N/60 //rad/s
//Calculating the gyroscopic couple acting on the bearings
```

```
C=round(m/8*omega^2*r^2*sind(2*theta)) //N-m
//Results:
printf("\n\n Gyroscopic couple acting on the
    bearings, C = %d N-m.\n\n",C)
```

# Chapter 15

# Inertia Forces in Reciprocating Parts

**Scilab code Exa 15.1** To find linear and angular velocity and acceleration

```
//To find linear and angular velocity and
   acceleration
clc
//Given:
OC=200/1000, PC=700/1000 //m
omega=120 //rad/s
//Solution:
//Refer Fig. 15.5
OM=127/1000, CM=173/1000, QN=93/1000, NO=200/1000 //
   m
//Velocity and acceleration of the piston:
//Calculating the velocity of the piston P
vP=omega*OM //m/s
//Calculating the acceleration of the piston P
aP=omega^2*NO //m/s^2
//Velocity and acceleration of the mid-point of the
   connecting rod:
//By measurement,
OD1=140/1000, OD2=193/1000 //m
```

```
//Calculating the velocity of D
vD=omega*OD1 //m/s
//Calculating the acceleration of D
aD=omega^2*OD2 //m/s^2
//Angular velocity and angular acceleration of the
    connecting rod:
//Calculating the velocity of the connecting rod PC
vPC=omega*CM //m/s
//Calculating the angular velocity of the connecting
     rod PC
omegaPC=vPC/PC //rad/s
//Calculating the tangential component of the
    acceleration of P with respect to C
atPC=omega^2*QN //m/s^2
//Calculating the angular acceleration of the
    connecting rod PC
alphaPC=atPC/PC //ra/s^2
//Results:
printf("\n\n Velocity of the piston P, vP = %.2f m/s
    .\n\n",vP)
printf(" Acceleration of the piston P, aP = %d m/s
    ^2.\n\n", aP)
printf(" Velocity of D, vD = %.1f m/s.\n\n",vD)
printf(" Acceleration of D, aD = %.1f m/s^2.\n\n",aD
    )
printf(" Angular velocity of the connecting rod PC,
    omegaPC = %.2f rad/s.\n\n",omegaPC)
printf(" Angular acceleration of the connecting rod
    PC, alphaPC = %.2f rad/s^2.\n\n",alphaPC)
```

**Scilab code Exa 15.2** To find linear and angular velocity and acceleration

```
//To find  linear and angular velocity and
    acceleration
clc
```

```
//Given:
OC=150/1000, PC=600/1000, CD=150/1000 //m
N=450 //rpm
//Solution:
//Refer Fig. 15.6
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//By measurement,
OM=145/1000, CM=78/1000, QN=130/1000, NO=56/1000 //m
//Velocity and acceleration of alider:
//Calculating the velocity of the slider P
vP=omega*OM //m/s
//Calculating the acceleration of the slider P
aP=omega^2*NO //m/s^2
//Velocity and acceleration of point D on the connecting rod:
//Calculating the length od CD1
CD1=CD/PC*CM //m
//By measurement,
OD1=145/1000, OD2=120/1000 //m
//Calculating the velocity of point D
vD=omega*OD1 //m/s
//Calculating the acceleration of point D
aD=omega^2*OD2 //m/s^2
//Angular velocity and angular acceleration of the connecting rod:
//Calculating the velocity of the connecting rod PC
vPC=omega*CM //m/s
//Calculating the angular velocity of the connecting rod
omegaPC=vPC/PC //rad/s
//Calculating the tangential component of the acceleration of P with respect to C
atPC=omega^2*QN //m/s^2
//Calculating the angular acceleration of the connecting rod PC
alphaPC=atPC/PC //rad/s^2
//Results:
```

```
printf("\n\n Velocity of the slider P, vP = %.3f m/s.\n\n",vP)
printf(" Acceleration of the slider P, aP = %.1f m/s^2.\n\n",aP)
printf(" Velocity of point D, vD = %.3f m/s.\n\n",vD)
printf(" Acceleration of point D, aD = %.2f m/s^2.\n\n",aD)
printf(" Angular velocity of the connecting rod, omegaPC = %.3f rad/s.\n\n",omegaPC)
printf(" Angular acceleration of the connecting rod PC, alphaPC = %.2f rad/s^2.\n\n",alphaPC)
```

**Scilab code Exa 15.3** To find crank angle and velocity

```
//To find crank angle and velocity
clc
//Given:
r=300/1000, l=1 //m
N=200 //rpm
//Solution:
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Crank angle at which the maximum velocity occurs:
//Calculating the ratio of length of connecting rod to crank radius
n=l/r
//Velocity of the piston, vP = omega*r*(sind(theta)+sind(2*theta)/(2*n))      .....(i)
//For maximum velocity, d(vP)/d(theta) = 0                        .....(ii)
//Substituting (i) in (ii), we get, 2(cosd(theta))^2+n*cosd(theta)-1 = 0
a=2, b=n, c=-1
costheta=(-b+sqrt(b^2-4*a*c))/(2*a)
```

```
//Calculating the crank angle from the inner dead centre at which the maximum velocity occurs
theta=round(acosd(costheta)) //degrees
//Calculating the maximum velocity of the piston:
vPmax=omega*r*(sind(theta)+sind(2*theta)/(2*n)) //m/s
//Results:
printf("\n\n Crank angle from the inner dead centre at which the maximum velocity occurs, theta = %d degrees.\n\n",theta)
printf(" Maximum velocity of the piston, vP(max) = %.2f m/s.\n\n",vPmax)
```

**Scilab code Exa 15.4** To find velocity and acceleration

```
//To find velocity and acceleration
clc
//Given:
r=0.3, l=1.5 //m
N=180 //rpm
theta=40 //degrees
//Solution:
//Calculating the angular speed of the piston
omega=2*%pi*N/60 //rad/s
//Velocity of the piston:
//Calculating the ratio of lengths of the connecting rod and crank
n=l/r
//Calculating the velocity of the piston
vP=omega*r*(sind(theta)+sind(2*theta)/(2*n)) //m/s
//Calculating the acceleration of the piston
aP=omega^2*r*(cosd(theta)+cosd(2*theta)/n) //m/s^2
//Position of the crank for zero acceleration of the piston:
ap1=0
```

```
//Calculating the position of the crank from the
   inner dead centre for zero acceleration of the
   piston
//We have, ap1 = omega^2*r*(cosd(theta1)+cosd(2*
   theta1)/n), or 2*(cosd(theta1))^2+n*cosd(theta1)
   -1 = 0
a=2, b=n, c=-1
costheta1=(-b+sqrt(b^2-4*a*c))/(2*a)
//Calculating the crank angle from the inner dead
   centre for zero acceleration of the piston
theta1=acosd(costheta1) //degrees
//Results:
printf("\n\n Velocity of the piston, vP = %.2f m/s.\
   n\n", vP)
printf(" Acceleration of the piston, aP = %.2f m/s
   ^2.\n\n",aP)
printf(" Position of the crank for zero acceleration
    of the piston, theta1 = %.2f degrees or %.2f
   degrees.\n\n",theta1,360-theta1)
```

**Scilab code Exa 15.5** To find linear and angualr velocity and acceleration

```
//To find linear and angualr velocity and
   acceleration
clc
//Given:
r=150/1000, l=600/1000 //m
theta=60 //degrees
N=450 //rpm
//Solution:
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Velocity and acceleration of the slider:
//Calculating the ratio of length of connecting rod
   and crank
```

```
n=l/r
//Calculating the velocity of the slider
vP=omega*r*(sind(theta)+sind(2*theta)/(2*n)) //m/s
//Calculating the acceleration of the slider
aP=omega^2*r*(cosd(theta)+cosd(2*theta)/n) //m/s^2
//Angular velocity and angular acceleration of the connecting rod:
//Calculating the angular velocity of the connecting rod
omegaPC=omega*cosd(theta)/n //rad/s
//Calculating the angular acceleration of the connecting rod
alphaPC=round(omega^2*sind(theta)/n) //rad/s^2
//Results:
printf("\n\n Velocity of the slider, vP = %.1f m/s.\n\n",vP)
printf(" Acceleration of the slider, aP = %.2f m/s^2.\n\n",aP)
printf(" Angular velocity of the connecting rod, omegaPC = %.1f rad/s.\n\n",omegaPC)
printf(" Angular acceleration of the connecting rod, alphaPC = %d rad/s^2.\n\n",alphaPC)
```

**Scilab code Exa 15.6** To find inertia force

```
//To find inertia force
clc
//Given:
D=175/1000, L=200/1000, r=L/2, l=400/1000 //m
N=500 //rpm
mR=180 //kg
//Solution:
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Analytical method:
```

```
//Calculating the ratio of lengths of connecting rod and crank
n=l/r
//Calculating the inertia force
FI=mR*omega^2*r*(cosd(theta)+cosd(2*theta)/n)/1000 //kN
//Results:
printf("\n\n Inertia force, FI = %.2f kN.\n\n",FI)
```

**Scilab code Exa 15.7** To find pressure thrust force and moment

```
//To find pressure, thrust, force and moment
clc
//Given:
r=300/1000, l=1.2, D=0.5 //m
mR=250 //kg
theta=60 //degrees
dp=0.35 //p1-p2, N/mm^2
N=250 //rpm
//Solution:
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Calculating the net load on the piston
FL=(dp)*%pi/4*(D*1000)^2 //N
//Calculating the ratio of length of connecting rod and crank
n=l/r
//Calculating the accelerating or inertia force on reciprocating parts
FI=mR*omega^2*r*(cosd(theta)+cosd(2*theta)/n) //N
//Calculating the piston effort
FP=(FL-FI)/1000 //kN
//Pressure on slide bars:
//Calculating the angle of inclination of the connecting rod to the line of stroke
```

```
phi=asind(sind(theta)/n) //degrees
//Calculating the pressure on the slide bars
FN=FP*tand(phi) //kN
//Calculating the thrust in the connecting rod
FQ=FP/cosd(phi) //kN
//Calculating the tangential force on the crank pin
FT=FQ*sind(theta+phi) //kN
//Calculating the turning moment on the crank shaft
T=FT*r //kN-m
//Results:
printf("\n\n Pressure on the slide bars, FN = %.2f
    kN.\n\n",FN)
printf(" Thrust in the connecting rod, FQ = %.2f kN
    .\n\n",FQ)
printf(" Tangential force on the crank-pin, FT = %.2
    f kN.\n\n",FT)
printf(" Turning moment on the crank shaft, T = %.3f
     kN-m.\n\n",T)
```

**Scilab code Exa 15.8** To find turning moment

```
//To find turning moment
clc
//Given:
D=300/1000, L=450/1000, r=L/2, d=50/1000, l=1.2 //m
N=200 //rpm
mR=225 //kg
theta=125 //degrees
p1=30*1000, p2=1.5*1000 //N/m^2
//Solution:
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Calculating the area of the piston
A1=%pi/4*D^2 //m^2
//Calculating the area of the piston rod
```

```
a=%pi/4*d^2 //m^2
//Calculating the force on the piston due to steam pressure
FL=round(p1*A1-p2*(A1-a)) //N
//Calculating the ratio of lengths of connecting rod and crank
n=l/r
//Calculating the inertia force on the reciprocating parts
FI=mR*omega^2*r*(cosd(theta)+cosd(2*theta)/n) //N
//Calculating the net force on the piston or piston effort
FP=FL-FI+mR*9.81 //N
//Calculating the angle of inclination of the connecting rod to the line of stroke
phi=asind(sind(theta)/n) //degrees
//Calculating the effective turning moment on the crank shaft
T=FP*sind(theta+phi)/cosd(phi)*r //N-m
//Results:
printf("\n\n Effective turning moment of the crank shaft, T = %.1f N-m.\n\n",T)
```

**Scilab code Exa 15.9** To find load thrust reaction and speed

```
//To find load, thrust, reaction and speed
clc
//Given:
N=1800 //rpm
r=50/1000, l=200/1000, D=80/1000, x=10/1000 //m
mR=1 //kg
p=0.7 //N/mm^2
//Solution:
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
```

```
//Net load on the gudgeon pin:
//Calculating the load on the piston
FL=round(%pi/4*(D*1000)^2*p) //N
//Refer Fig. 15.10
//By measurement,
theta=33 //degrees
//Calculating the ratio of lengths of connecting rod and crank
n=l/r
//Calculating the inertia force on the reciprocating parts
FI=mR*omega^2*r*(cosd(theta)+cosd(2*theta)/n) //N
//Calculating the net load on the gudgeon pin
FP=FL-FI //N
//Thrust in the connecting rod:
//Calculating the angle of inclination of the connecting rod to the line of stroke
phi=asind(sind(theta)/n) //degrees
//Calculating the thrust in the connecting rod
FQ=FP/cosd(phi) //N
//Calculating the reaction between the piston and cylinder
FN=FP*tand(phi) //N
//Engine speed at which the abov values will become zero:
//Calculating the speed at which FI=FL
omega1=sqrt((%pi/4*(D*1000)^2*p)/(mR*r*(cosd(theta)+cosd(2*theta)/n))) //rad/s
//Calculating the corresponding speed in rpm
N1=omega1*60/(2*%pi) //rpm
//Results:
printf("\n\n Net load on the gudgeon pin, FP = %d N.\n\n",FP)
printf(" Thrust in the connecting rod, FQ = %.1f N.\n\n",FQ)
printf(" Reaction between the piston and cylinder, FN = %d N.\n\n",FN)
printf(" Engine speed at which the above values will
```

```
become zero, N1 = %d rpm.\n\n",N1)
```

**Scilab code Exa 15.10** To find reaction thrust and turning moment

```
//To find reaction, thrust and turning moment
clc
//Given:
aP=36 //m/s^2
theta=30 //degrees
p=0.5 //N/mm^2
RF=600 //N
D=300/1000, r=300/1000 //m
mR=180 //kg
n=4.5
//Solution:
//Reaction on the guide bars:
//Calculating the load on the piston
FL=round(p*%pi/4*(D*1000)^2) //N
//Calculating the inertia force due to reciprocating
    parts
FI=mR*aP //N
//Calculating the piston effort
FP=(FL-FI-RF)/1000 //kN
//Calculating the angle of inclination of the
   connecting rod to the line of stroke
phi=asind(sind(theta)/n) //degrees
//Calculating the reaction on the guide bars
FN=FP*tand(phi) //kN
//Calculating the thrust on the crank shaft bearing
FB=(FP*cosd(phi+theta))/cosd(phi) //kN
//Calculating the turning moment on the crank shaft
T=(FP*sind(theta+phi))/cosd(phi)*r //kN-m
//Results:
printf("\n\n Reaction on the guide bars, FN = %.2f
   kN.\n\n",FN)
```

```
printf(" Thrust on the crank shaft bearing , FB = %.1
    f kN.\n\n",FB)
printf(" Turning moment on the crank shaft , T = %.2f
     kN-m.\n\n",T)
```

**Scilab code Exa 15.11** To find force load thrust and speed

```
//To find force , load , thrust and speed
clc
//Given:
D=100/1000, L=120/1000, r=L/2, l=250/1000 //m
mR=1.1 //kg
N=2000 //rpm
theta=20 //degrees
p=700 //kN/m^2
//Solution:
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Net force on the piston:
//Calculating the force due to gas pressure
FL=p*%pi/4*D^2 //kN
//Calculating the ratio of lengths of the connecting
     rod and crank
n=l/r
//Calculating the inertia force on the piston
FI=round(mR*omega^2*r*(cosd(theta)+cosd(2*theta)/n))
     //N
//Calculating the net force on the piston
FP=(FL*1000)-FI+mR*9.81 //N
//Resultant force on the gudgeon pin:
//Calculating the angle of inclination of the
    connecting rod to the line of stroke
phi=asind(sind(theta)/n) //degrees
//Calculating the resultant load on the gudgeon pin
FQ=round(FP/cosd(phi)) //N
```

```
//Calculating the thrust on the cylinder walls
FN=FP*tand(phi) //N
//Speed, above which, the gudgeon pin load would be reversed in direction:
//Calculating the minimum speed for FP to be negative
omega1=sqrt((FL*1000+mR*9.81)/(mR*r*(cosd(theta)+cosd(2*theta)/n))) //rad/s
//Calculating the corresponding speed in rpm
N1=omega1*60/(2*%pi) //rpm
//Results:
printf("\n\n Net force on the piston, FP = %.1f N.\n\n",FP)
printf(" Resultant load on the gudgeon pin, FQ = %d N.\n\n",FQ)
printf(" Thrust on the cylinder walls, FN = %.1f N.\n\n",FN)
printf(" Speed, above which, the gudgeon pin load would be reversed in direction, N1 > %d rpm.\n\n",N1)
```

**Scilab code Exa 15.12** To find turning moment thrust and acceleration

```
//To find turning moment, thrust and acceleration
clc
//Given:
N=120 //rpm
D=250/1000, L=400/1000, r=L/2, l=0.6, d=50/1000 //m
mR=60 //kg
theta=45 //degrees
p1=550*1000, p2=70*1000 //N/m^2
//Solution:
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Turning moment on the crankshaft:
```

```
//Calculating the area of the piston on the cover
   end side
A1=%pi/4*D^2 //m^2
//Calculating the area of the piston rod
a=%pi/4*d^2 //m^2
//Calculating the net load on the piston
FL=p1*A1-p2*(A1-a) //N
//Calculating the ratio of lengths of the connecting
    rod and crank
n=l/r
//Calculating the inertia force on the reciprocating
    parts
FI=mR*omega^2*r*(cosd(theta)+cosd(2*theta)/n) //N
//Calculating the net force on the piston or piston
   effort
FP=(FL-FI)/1000 //kN
//Calculating the angle of inclination of the
   connecting rod to the line of stroke
phi=asind(sind(theta)/n) //degrees
//Calculating the turning moment on the crank shaft
T=(FP*sind(theta+phi))/cosd(phi)*r*1000 //N-m
//Calculating the thrust on the bearings
FB=(FP*cosd(theta+phi))/cosd(phi) //kN
//Acceleration of the flywheel:
P=20*1000 //W
m=60 //kg
k=0.6 //m
//Calculating the mass moment of inertia of the
   flywheel
I=m*k^2 //kg-m^2
//Calculating the resisting torque
TR=P*60/(2*%pi*N) //N-m
//Calculating the acceleration of the flywheel
alpha=(T-TR)/I //rad/s^2
//Results:
printf("\n\n Turning moment on the crank shaft, T =
   %d N-m.\n\n",T)
printf(" Thrust on the bearings, FB = %.2f kN.\n\n",
```

```
    FB)
printf(" Acceleration of the flywheel, alpha = %.1f
    rad/s^2.\n\n",alpha)
```

**Scilab code Exa 15.13** To find effort thrust and turning moment

```
//To find effort, thrust and turning moment
clc
//Given:
D=300/1000, L=500/1000, r=L/2 //m
n=4.5
N=180 //rpm
mR=280 //kg
theta=45 //degrees
p1=0.1 //N/mm^2
CR=14 //Compression ration, V1/V2
//Solution:
//Refer Fig. 15.12
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Calculating the pressure corresponding to point 2
p2=p1*(CR)^1.35 //N/mm^2
//Calculating the swept volume
VS=%pi/4*D^2*L //m^3
//Calculating the clearance volume
VC=VS/(CR-1) //m^3
//Calculating the volume corresponding to point 3
V3=VC+(1/10*VS) //m^3
//Calculating the displacement of the piston
    corresponding to crank displacement of 45 degrees
x=r*((1-cosd(theta))+(sind(theta))^2/(2*n)) //m
//Calculating the volume corresponding to point 4'
V4dash=VC+(%pi/4*D^2*x) //m^2
//Calculating the pressure corresponding to point 4'
p3=p2
```

```
p4dash=p3*(V3/V4dash)^1.35 //N/mm^2
//Calculating the difference of pressures on two sides of the piston
p=(p4dash-p1)*10^6 //N/m^2
//Calculating the net load on the piston
FL=p*%pi/4*D^2 //N
//Calculating the inertia force on the reciprocating parts
FI=mR*omega^2*r*(cosd(theta)+cosd(2*theta)/n) //N
//Calculating the net force on the piston or piston effort
FP=FL-FI+mR*9.81 //N
//Crank-pin effort:
//Calculating the angle of inclination of the connecting rod to the line of stroke
phi=asind(sind(theta)/n) //degrees
//Calculating the crank-pin effort
FT=(FP*sind(theta+phi))/(cosd(phi)*1000) //kN
//Calculating the thrust on the bearings
FB=(FP*cosd(theta+phi))/(cosd(phi)*1000) //kN
//Calculating the turning moment on the crankshaft
T=FT*r //kN-m
//Results:
printf("\n\n Crank-pin effort, FT = %.3f kN.\n\n",FT)
printf(" Thrust on the bearings, FB = %.3f kN.\n\n",FB)
printf(" Turning moment on the crankshaft, T = %.2f kN-m.\n\n",T)
```

---

**Scilab code Exa 15.14** To find turning moment

```
//To find turning moment
clc
//Given:
```

```
D=240/1000, L=360/1000, r=L/2, l=0.6 //m
N=300 //rpm
mR=160 //kg
pA=(8+1.03)*10^5, pE=(-0.75+1.03)*10^5 //N/m^2
FR=500 //N
theta=75 //degrees
//Solution:
//Refer Fig. 15.13
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Calculating the stroke volume
VS=%pi/4*D^2*L //m^3
//Calculating the volume of steam at cut-off
VB=VS/3 //m^3
//Calculating the ratio of lengths of the connecting
    rod and crank
n=l/r
//Calculating the displacement of the piston when
   the crank position is 75 degrees from the top
   dead centre
x=r*((1-cosd(theta))+(sind(theta))^2/(2*n)) //m^3
//Calculating the volume corresponding to point C'
VCdash=VS*x/L //m^3
//Calculating the pressure corresponding to point C'
pB=pA
pCdash=round((pB*VB)/VCdash) //N/m^2
//Calculating the difference of pressures on the two
    sides of the piston
p=round(pCdash-pE) //N/m^2
//Calculating the net load on the piston
FL=round(%pi/4*D^2*p) //N
//Calculating the inertia force on the reciprocating
    parts
FI=round(mR*omega^2*r*(cosd(theta)+(cosd(2*theta)/n)
   )) //N
//Calculating the piston effort
FP=FL-FI+mR*9.81-FR //N
//Turning moment on the crankshaft:
```

```
//Calculating the angle of inclination of the connecting rod to the line of stroke
phi=asind(sind(theta)/n) //degrees
//Calculating the turning moment on the crankshaft
T=(FP*sind(theta+phi))/cosd(phi)*r //N-m
//Results:
printf("\n\n Turning moment on the crankshaft, T = %d N-m.\n\n",T)
```

**Scilab code Exa 15.15** To find equivalent system

```
//To find equivalent system
clc
//Given:
l=300, l1=200 //mm
m=15 //kg
I=7000 //kg-mm^2
//Solution:
//Refer Fig. 15.16 and Fig. 15.17
//Calculating the radius of gyration of the connecting rod about an axis passing through its centre of gravity
kG=sqrt(I/m) //mm
//Calculating the distance of other mass from the centre of gravity
l2=(kG)^2/l1 //mm
//Calculating the magnitude of mass placed at the small end centre
m1=(l2*m)/(l1+l2) //kg
//Calculating the magnitude of the mass placed at a distance l2 from the centre of gravity
m2=(l1*m)/(l1+l2) //kg
//Results:
printf("\n\n Mass placed at the small end centre, m1 = %.2f kg.\n\n",m1)
```

```
printf(" Mass placed at a distance %.2f mm from the
    centre of gravity , m2 = %.2f kg.\n\n",l2,m2)
```

**Scilab code Exa 15.16** To find equivalent system

```
//To find equivalent system
clc
//Given:
h=650/1000, l1=(650-25)/1000 //m
m=37.5 //kg
tp=1.87 //seconds
//Solution:
//Refer Fig. 15.18 and Fig. 15.19
//Calculating the radius of gyration of the
    connecting rod about an axis passing through its
    centre of gravity
kG=sqrt((tp/(2*%pi))^2*(9.81*h)-h^2) //m
//Calculating the distance of mass m2 from the
    centre of gravity
l2=(kG)^2/l1 //m
//Calculating the magnitude of mass placed at the
    small end centre
m1=(l2*m)/(l1+l2) //kg
//Calculating the magnitude of mass placed at a
    distance l2 from centre of gravity
m2=(l1*m)/(l1+l2) //kg
//Results:
printf("\n\n Mass placed at the small end centre A,
    m1 = %d kg.\n\n",m1)
printf(" Mass placed at a distance %.3f m from G, m2
    = %.1f kg.\n\n",l2,m2)
```

**Scilab code Exa 15.17** To find radius and MI

```
//To find radius and MI
clc
//Given:
m=55 //kg
l=850/1000, d1=75/1000, d2=100/1000 //m
tp1=1.83, tp2=1.68 //seconds
//Solution:
//Refer Fig. 15.20
//Calculating the length of equivalent simple pendulum when suspended from the top of small end bearing
L1=9.81*(tp1/(2*%pi))^2 //m
//Calculating the length of equivalent simple pendulum when suspended from the top of big end bearing
L2=9.81*(tp2/(2*%pi))^2 //m
//Radius of gyration of the rod about an axis passing through the centre of gravity and perpendicular to the plane of oscillation:
//Calculating the distances of centre of gravity from the top of small end and big end bearings
//We have, h1*(L1-h1) = h2*(L2-h2), or h1^2-h2^2+h2*L2-h1*L1 = 0                    .....(i)
//Also, h1+h2 = d1/2+l+d2/2, or h1+h2-d1/2-l-d2/2 = 0                    .....(ii)
function y=f(x)
    h1=x(1)
    h2=x(2)
    y(1)=h1^2-h2^2+h2*L2-h1*L1
    y(2)=h1+h2-d1/2-l-d2/2
endfunction
z=fsolve([1,1],f)
h1=z(1), h2=z(2) //m
//Calculating the required radius of gyration of the
```

```
rod
kG=sqrt(h1*(L1-h1)) //m
//Calculating the moment of inertia of the rod
I=m*(kG)^2 //kg-m^2
//Dynamically equivalent system for the rod:
//Calculating the distance of the mass situated at the centre of small end bearing from the centre of gravity
l1=h1-d1/2 //m
//Calculating the distance of the second mass from the centre of gravity towards big end bearing
l2=(kG)^2/l1 //m
//Calculating the magnitude of the mass situated at the centre of small end bearing
m1=(l2*m)/(l1+l2) //kg
//Calculating the magnitude of the second mass
m2=(l1*m)/(l1+l2) //kg
//Results:
printf("\n\n Radius of gyration of the rod about an axis passing through the centre of gravity and perpendicular to the plane of oscillation, kG = %.3f m.\n\n",kG)
printf(" Moment of inertia of the rod, I = %.2f kg-m^2.\n\n",I)
printf(" Magnitude of the mass situated at the centre of small end bearing, m1 = %.2f kg.\n\n",m1)
printf(" Magnitude of the second mass, m2 = %.2f kg.\n\n",m2)
```

**Scilab code Exa 15.18** To find correcting couple

```
//To find correcting couple
clc
//Given:
```

```
m=2 //kg
l=250/1000, l1=100/1000, kG=110/1000 //m
alpha=23000 //rad/s^2
//Solution:
//Equivalent dynamical system:
//Calculating the distance of the second mass from the centre of gravity
l2=(kG)^2/l1 //m
//Calculating the magnitude of the mass placed at the gudgeon pin
m1=(l2*m)/(l1+l2) //kg
//Calculating the magnitude of the mass placed at a distance l2 from centre of gravity
m2=(l1*m)/(l1+l2) //kg
//Correction couple:
//Calculating the magnitude of l3
l3=l-l1 //m
//Calculating the new radius of gyration
k1=sqrt(l1*l3) //m^2
//Calculating the correction couple
Tdash=m*(k1^2-kG^2)*alpha //N-m
//Results:
printf("\n\n Mass placed at the gudgeon pin, m1 = %.1f kg.\n\n",m1)
printf(" Mass placed at a distance %.3f m from the centre of gravity, m2 = %.1f kg.\n\n",l2,m2)
printf(" Correction couple, Tdash = %.1f N-m.\n\n",Tdash)
```

**Scilab code Exa 15.19** To find acceleration and inertia force

```
//To find acceleration and inertia force
clc
//Given:
r=125, OC=r, l=500, PC=l, PG=275, kG=150 //mm
```

```
mC=60 //kg
N=600 //rpm
theta=45 //degrees
//Solution:
//Refer Fig. 15.24
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Acceleration of the piston:
//By measurement,
NO=90/1000 //m
//Calculating the acceleration of the piston
aP=omega^2*NO //m/s^2
//The magnitude, position and direction of inertia
    force due to the mass of the connecting rod:
//By measurement,
gO=103/1000 //m
//Calculating the magnitude of the inertia force of
    the connecting rod
FC=mC*omega^2*gO/1000 //kN
//Results:
printf("\n\n Acceleration of the piston, aP = %.1f m
    /s^2.\n\n",aP)
printf(" The magnitude of inertia force due to the
    mass of the connecting rod, FC = %.1f kN.\n\n",FC
    )
```

**Scilab code Exa 15.20** To find torque exerted

```
//To find torque exerted
clc
//Given:
D=240/1000, L=600/1000, r=L/2, l=1.5, GC=500/1000,
    kG=650/1000 //m
mR=300, mC=250 //kg
N=125 //rpm
```

```
theta=30 //degrees
//Solution:
//Refer Fig. 15.25
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Analytical method:
//Calculating the distance of centre of gravity of the connecting rod from P
l1=l-GC //m
//Calculating the ratio of lengths of the connecting rod and crank
n=l/r
//Calculating the inertia force due to total mass of the reciprocating parts at P
FI=(mR+(l-l1)/l*mC)*omega^2*r*(cosd(theta)+cosd(2*theta)/n) //N
//Calculating the corresponding torque due to FI
TI=FI*r*(sind(theta)+sind(2*theta)/(2*sqrt(n^2-(sind(theta))^2))) //N-m
//Calculating the equivalent length of a simple pendulum when swung about an axis through P
L=((kG)^2+(l1)^2)/l1 //m
//Calculating the correcting torque
TC=mC*l1*(l-L) //N-m
//Calculating the torque due to the weight of the connecting rod at C
TW=mC*9.81*(l1/n)*cosd(theta) //N-m
//Calculating the total torque exerted on the crankshaft
Tt=TI+TC+TW //Total torque exerted on the crankshaft, N-m
//Results:
printf("\n\n Total torque exerted on the crankshaft = %.1f N-m.\n\n",Tt)
```

**Scilab code Exa 15.21** To find acceleration and inertia torque

```
//To find acceleration and inertia torque
clc
//Given:
N=1200 //rpm
L=110/1000, r=L/2, l=250/1000, PC=l, CG=75/1000 //m
mC=1.25 //kg
theta=40 //degrees
//Solution:
//Refer Fig. 15.26
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Radius of gyration of the connecting rod about an axis through its mass centre:
//Calculating the distance of the centre of gravity from the point of suspension
l1=l-CG //m
PG=l1
//Calculating the frequency of oscillation
n=21/20 //Hz
//Calculating the radius of gyration of the connecting rod about an axis through its mass centre
kG=round(sqrt((9.81*l1/(2*%pi*n)^2)-l1^2)*1000) //mm
//Acceleration of the piston:
//Calculating the ratio of lengths of the connecting rod and crank
n=l/r
//Calculating the acceleration of the piston
aP=omega^2*r*(cosd(theta)+cosd(2*theta)/n) //m/s^2
//Calculating the angular acceleration of the connecting rod
alphaPC=(-omega^2*sind(theta))/n //rad/s^2
//Inertia torque exerted on the crankshaft:
//Calculating the mass of the connecting rod at P
m1=(l-l1)/l*mC //kg
//Calculating the vertical inertia force
```

```
FI=round(m1*aP) //N
//By measurement,
OM=0.0425, NC=0.035 //m
//Calculating the corresponding torque due to FI
TI=FI*OM //N-m
//Calculating the equivalent length of a simple pendulum when swung about an axis passing through P
L=((kG/1000)^2+(l1)^2)/l1 //m
//Calculating the correction couple
Tdash=mC*l1*(l-L)*alphaPC //N-m
//Calculating the corresponding torque on the crankshaft
TC=-Tdash*cosd(theta)/n //N-m
//Calculating the torque due to mass at P
TP=m1*9.81*OM //N-m
//Calculating the equivalent mass of the connecting rod at C
m2=mC*(l1/l) //kg
//Calculating the torque due to mass at C
TW=m2*9.81*NC //N-m
//Calculating the inertia force exerted on the crankshaft
Ti=TI+TC-TP-TW //Inertia torque exerted on the crankshaft, N-m
//Results:
printf("\n\n Radius of gyration of the connecting rod about an axis through its mass centre, kG = %d mm.\n\n",kG)
printf(" Acceleration of the piston, aP = %.1f m/s^2.\n\n",aP)
printf(" Angular acceleration of the connecting rod, alphaPC = %.1f rad/s^2.\n\n",alphaPC)
printf(" Inertia torque exerted on the crankshaft = %.3f N-m.\n\n",Ti)
```

**Scilab code Exa 15.22** To find resultant force

```
//To find resultant force
clc
//Given:
l=225/1000, PC=l, L=150/1000, r=L/2, D=112.5/1000, PG=150/1000, kG=87.5/1000 //m
mC=1.6, mR=2.4 //kg
theta=40 //degrees
p=1.8*10^6 //N/m^2
N=2000 //rpm
//Solution:
//Refer Fig. 15.27
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//By measurement,
NO=0.0625, gO=0.0685, IC=0.29, IP=0.24, IY=0.148, IX=0.08 //m
//Calculating the force due to gas pressure
FL=%pi/4*D^2*p //N
//Calculating the inertia force due to mass of the reciprocating parts
FI=mR*omega^2*NO //N
//Calculating the net force on the piston
FP=FL-FI //N
//Calculating the inertia force due to mass of the connecting rod
FC=mC*omega^2*gO //N
//Calculating the force acting perpendicular to the crank OC
FT=((FP*IP)-((mC*9.81*IY)+(FC*IX)))/IC //N
//By measurement,
FN=3550, FR=7550, FQ=13750 //N
//Results:
```

```
printf("\n\n Resultant force on the crank pin, FQ =
    %d N.\n\n",FQ)
```

# Chapter 16

# Turning Moment Diagrams and Flywheel

**Scilab code Exa 16.1** To find maximum and minimum speeds

```
//To find maximum and minimum speeds
clc
//Given:
m=6.5*1000 //kg
k=1.8 //m
deltaE=56*1000 //N-m
N=120 //rpm
//Solution:
//Calculating the maximum and minimum speeds
//We know that fluctuation of energy, deltaE = %pi
    ^2/900*m*k^2*N*(N1-N2), or N1-N2 = (deltaE/(%pi
    ^2/900*m*k^2*N))     .....(i)
//Also mean speed, N = (N1+N2)/2, or N1+N2 = 2*N

    .....(ii)
A=[1 -1; 1 1]
B=[deltaE/(%pi^2/900*m*k^2*N); 2*N]
V=A \ B
N1=round(V(1)) //rpm
```

```
N2=round(V(2)) //rpm
//Results:
printf("\n\n Maximum speed, N1 = %d rpm.\n\n",N1)
printf(" Minimum speed, N2 = %d rpm.\n\n",N2)
```

**Scilab code Exa 16.2** To find angular acceleration and KE

```
//To find angular acceleration and KE
clc
//Given:
k=1 //m
m=2500 //kg
T=1500 //N-m
//Solution:
//Angular acceleration of the flywheel:
//Calculating the mass moment of inertia of the flywheel
I=m*k^2 //kg-m^2
//Calculating the angular acceleration of the flywheel
alpha=T/I //rad/s^2
//Kinetic energy of the flywheel:
omega1=0 //Angular speed at rest
//Calculating the angular speed after 10 seconds
omega2=omega1+alpha*10 //rad/s
//Calculating the kinetic energy of the flywheel
KE=1/2*I*(omega2)^2/1000 //Kinetic energy of the flywheel, kN-m
//Results:
printf("\n\n Angular acceleration of the flywheel, alpha = %.1f rad/s^2.\n\n",alpha)
printf(" Kinetic energy of the flywheel = %d kN-m.\n\n",KE)
```

**Scilab code Exa 16.3** To find weight of flywheel

```
//To find weight of flywheel
clc
//Given:
P=300*1000 //W
N=90 //rpm
CE=0.1
k=2 //m
//Solution:
//Calculating the mean angular speed
omega=2*%pi*N/60 //rad/s
//Calculating the coefficient of fluctuation of
    speed
CS=1/100
//Calculating the work done per cycle
WD=P*60/N //Work done per cycle, N-m
//Calculating the maximum fluctuation of energy
deltaE=WD*CE //N-m
//Calculating the mass of the flywheel
m=deltaE/(k^2*omega^2*CS) //kg
//Results:
printf("\n\n Mass of the flywheel, m = %d kg.\n\n",m
    )
```

---

**Scilab code Exa 16.4** To find coefficient of fluctuation of speed

```
//To find coefficient of fluctuation of speed
clc
//Given:
m=36 //kg
k=150/1000 //m
```

```
N=1800 //rpm
//Solution:
//Refer Fig. 16.6
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Calculating the value of 1 mm^2 on the turning
   moment diagram
c=5*%pi/180 //Value of 1 mm^2 on turning miment
   diagram, N-m
//Calculating the maximum fluctuation of energy
//From the turning moment diagram, maximum energy =
   E+295, and minimum energy = E-690
deltaE=(285-(-690))*c //N-m
//Calculating the coefficient of fluctuation of
   energy
CS=deltaE/(m*k^2*omega^2)*100 //%
//Results:
printf("\n\n Coefficient of fluctuation of speed, CS
    = %.1f %c.\n\n",CS,"%")
```

**Scilab code Exa 16.5** To find mass of the flywheel

```
//To find mass of the flywheel
clc
//Given:
N=600 //rpm
R=0.5 //m
//Solution:
//Refer Fig. 16.7
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Calculating the coefficient of fluctuation of
   speed
CS=3/100
//Calculating the value of 1 mm^2 on turning moment
```

```
diagram
c=600*%pi/60 //Value of 1 mm^2 on turning moment diagram, N-m
//Calculating the maximum fluctuation of energy
//From the turning moment diagram, maximum fluctuation = E+52, and minimum fluctuation = E-120
deltaE=(52-(-120))*c //N-m
//Calculating the mass of the flywheel
m=deltaE/(R^2*omega^2*CS) //kg
//Results:
printf("\n\n Mass of the flywheel, m = %d kg.\n\n",m)
```

---

**Scilab code Exa 16.6** To find power and speed fluctuation

```
//To find power and speed fluctuation
clc
//Given:
N=250 //rpm
m=500 //kg
k=600/1000 //m
//Solution:
//Refer Fig. 16.8
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Calculating the torque required for one complete cycle
T=(6*%pi*750)+(1/2*%pi*(3000-750))+(2*%pi*(3000-750))+(1/2*%pi*(3000-750)) //N-m
//Calculating the mean torque
Tmean=T/(6*%pi) //N-m
//Calculating the power required to drive the machine
P=Tmean*omega/1000 //kW
```

```
//Coefficient of fluctuation of speed:
//Calculating the value of LM
LM=%pi*((3000-1875)/(3000-750))
//Calculating the value of NP
NP=%pi*((3000-1875)/(3000-750))
//Calculating the value of BM
BM=3000-1875 //N-m
CN=BM
//Calculating the value of MN
MN=2*%pi
//Calculating the maximum fluctuation of energy
deltaE=(1/2*LM*BM)+(MN*BM)+(1/2*NP*CN) //N-m
//Calculating the coefficient of fluctuation of
    speed
CS=deltaE/(m*k^2*omega^2)
//Results:
printf("\n\n Power required to drive the machine, P
    = %.3f kW.\n\n",P)
printf(" Coefficient of speed, CS = %.3f.\n\n",CS)
```

**Scilab code Exa 16.7** To find coefficient of fluctuation

```
//To find coefficient of fluctuation
clc
//Given:
N=100 //rpm
k=1.75 //m
//Solution:
//Refer Fig. 16.9
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Calculating the coefficient of fluctuation of
    speed
CS=1.5/100
//Coefficient of fluctuation of energy:
```

```
AB=2000, LM=1500 //N-m
//Calculating the work done per cycle
WD=(1/2*%pi*AB)+(1/2*%pi*LM) //Work done per cycle, N-m
//Calculating the mean resisting torque
Tmean=WD/(2*%pi) //N-m
//Calculating the value of CD
CD=%pi/2000*(2000-875) //rad
//Calculating the maximum fluctuation of energy
deltaE=1/2*CD*(2000-875) //N-m
//Calculating the coefficient of fluctuation of energy
Ce=deltaE/WD*100 //%
//Calculating the mass of the flywheel
m=deltaE/(k^2*omega^2*CS) //kg
//Crank angles for minimum and maximum speeds:
//Calculating the value of CE
CE=(2000-875)/2000*(4*%pi/9) //rad
//Calculating the crank angle for minimum speed
thetaC=((4*%pi/9)-CE)*180/%pi //degrees
//Calculating the value of ED
ED=(2000-875)/2000*(%pi-(4*%pi/9)) //rad
//Calculating the crank angle for maximum speed
thetaD=((4*%pi/9)+ED)*180/%pi //degrees
//Results:
printf("\n\n Coefficient of fluctuation of energy, CE = %d %c.\n\n",Ce,"%")
printf(" Mass of the flywheel, m = %.1f kg.\n\n",m)
printf(" Crank angle from IDC for the minimum speed, thetaC = %d degrees.\n\n",thetaC)
printf(" Crank angle from IDC for the maximum speed, thetaD = %d degrees.\n\n",thetaD)
```

**Scilab code Exa 16.8** To find power and coefficients

```
//To find power and coefficients
clc
//Given:
N=600 //rpm
Tmax=90 //N-m
m=12 //kg
k=80/1000 //m
//Solution:
//Refer Fig. 16.10
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Power developed:
//Calculating the work done per cycle
WD=3*1/2*%pi*90 //Work done per cycle, N-m
//Calculating the mean torque
Tmean=WD/(2*%pi) //N-m\
//Calculating the power developed
P=Tmean*omega/1000 //Power developed, kW
//Coefficient of fluctuation of speed:
//Calculating the maximum fluctuation of energy
//From the torque-crank angle diagram, maximum
    energy=E+5.89, and minimum energy=E-5.89
deltaE=5.89-(-5.89) //N-m
//Calculating the coefficient of fluctuation of
    speed
CS=round(deltaE/(m*k^2*omega^2)*100) //%
//Calculating the coefficient of fluctuation of
    energy
CE=deltaE/WD*100 //%
//Calculating the maximum angular acceleration of
    the flywheel
alpha=(Tmax-Tmean)/(m*k^2) //rad/s^2
//Results:
printf("\n\n Power developed = %.2f kW.\n\n",P)
printf(" Coefficient of fluctuation of speed, CS =
    %d %c.\n\n",CS,"%")
printf(" Coefficient of fluctuation of energy, CE =
    %.2f %c.\n\n",CE,"%")
```

```
printf(" Maximum angular acceleration of the
    flywheel , alpha = %d rad/s^2.\n\n",alpha)
```

**Scilab code Exa 16.9** To find moment of inertia

```
//To find moment of inertia
clc
//Given:
P=20*1000 //W
N=300 //rpm
//Solution:
//Refer Fig. 16.11
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //ra/s
//Calculating the coefficient of fluctuation of
    speed
CS=4/100
//Calculating the number of working strokes per
    cycle for a four stroke engine
n=N/2
//Calculating the work done per cycle
WD=P*60/n //Work done per cycle , N-m
//Calculating the work done during expansion cycle
WE=WD*3/2 //N-m
//Calculating the maximum turning moment
Tmax=WE*2/%pi //N-m
//Calculating the mean turning moment
Tmean=WD/(4*%pi) //N-m
//Calculating the excess turning moment
Texcess=Tmax-Tmean //N-m
//Calculating the value of DE
DE=Texcess/Tmax*%pi //rad
//Calculating the maximum fluctuation of energy
deltaE=(1/2*DE*Texcess) //N-m
//Calculating the moment of inertia of the flywheel
```

```
I=deltaE/(omega^2*CS) //kg-m^2
//Results:
printf("\n\n Moment of inertia of the flywheel, I = %.1f kg-m^2.\n\n",I)
```

**Scilab code Exa 16.10** To find mass of the rim

```
//To find mass of the rim
clc
//Given:
a1=0.45*10^-3, a2=1.7*10^-3, a3=6.8*10^-3, a4=0.65*10^-3 //m^2
N1=202, N2=198 //rpm
R=1.2 //m
//Solution:
//Refer Fig. 16.12
//Calculating the net area
a=a3-(a1+a2+a4) //Net area, m^2
//Calculating the energy scale constant
c=3*10^6 //Energy scale constant, N-m
//Calculating the net work done per cycle
WD=a*c //Net work done per cycle, N-m
//Calculating the mean torque
Tmean=WD/(4*%pi) //N-m
//Calculating the value of FG
FG=Tmean //N-m
//Calculating the work done during expansion stroke
WDe=a3*c //Work done during expansion stroke, N-m
//Calculating the value of AG
AG=WDe/(1/2*%pi) //N-m
//Calculating the excess torque
Texcess=AG-FG //N-m
//Calculating the value of AF
AF=Texcess //N-m
//Calculating the value of DE
```

```
DE=AF/AG*%pi //rad
//Calculating the maximum fluctuation of energy
deltaE=1/2*DE*AF //N-m
//Mass of the rim of a flywheel:
//Calculating the mean speed of the flywheel
N=(N1+N2)/2 //rpm
//Calculating the mass of the rim of a flywheel
m=deltaE/(%pi^2/900*R^2*N*(N1-N2)) //kg
//Results:
printf("\n\n Mass of the rim of the flywheel, m = %d
    kg.\n\n",m)
```

**Scilab code Exa 16.12** To find fluctuation of energy and speed

```
//To find fluctuation of energy and speed
clc
//Given:
m=500 //kg
k=0.4 //m
N=150 //rpm
//Solution:
//Refer Fig. 16.14
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Fluctuation of energy:
//Equating the change in torque to zero and
    calculating the value of theta
thetaA=asind(0), thetaC=asind(0)+180, thetaE=asind
    (0)+360 //degrees
thetaB=acosd(1/(2*(600/500))), thetaD=360-acosd
    (1/(2*(600/500))) //degrees
//Calculating the maximum fluctuation of energy
deltaE=round(integrate('(5000+600*sin(2*theta))
    -(5000+500*sin(theta))','theta',thetaC*%pi/180,
    thetaD*%pi/180)) //N-m
```

```
//Calculating the total percentage fluctuation of speed
CS=deltaE/(m*k^2*omega^2)*100 //%
//Maximum and minimum angular acceleration of the flywheel and the corresponding shaft positions:
//Calculating the maximum or minimum values of theta
//Differentiating (600*sin(2*theta))-500*sin(theta) = 0 with respect to theta and equating to zero, we get 12*2*(cosd(theta))^2-5*cosd(theta)-12 = 0
a=12*2, b=-5, c=-12
cosdtheta1=(-b+sqrt(b^2-4*a*c))/(2*a)
cosdtheta2=(-b-sqrt(b^2-4*a*c))/(2*a)
theta1=round(acosd(cosdtheta1)), theta2=acosd(cosdtheta2) //degrees
//Calculating the maximum torque
Tmax=600*sind(2*theta1)-500*sind(theta1) //N-m
//Calculating the minimum torque
Tmin=600*sind(2*theta2)-500*sind(theta2) //N-m
//Calculating the maximum acceleration
alphamax=Tmax/(m*k^2) //rad/s^2
//Calculating the minimum acceleration
alphamin=abs(Tmin)/(m*k^2) //rad/s^2
//Results:
printf("\n\n Fluctuation of energy, deltaE = %d N-m.\n\n",deltaE)
printf(" Total percentage fluctuation of speed, CS = %.1f %c.\n\n",CS,"%")
printf(" Shaft position corresponding to maximum and minimum accelerations, theta = %d degrees and %.1f degrees.\n\n",theta1,theta2)
printf(" Maximum acceleration, alphamax = %.2f rad/s^2.\n\n",alphamax)
printf(" Minimum acceleration, alphamin = %.1f rad/s^2.\n\n",alphamin)
```

**Scilab code Exa 16.13** To find power fluctuation and torque

```
//To find power, fluctuation and torque
clc
//Given:
I=1000 //kg-m^2
N=300 //rpm
//Solution:
//Refer Fig. 16.15 and Fig. 16.16
//Calculating the angular speed of the crank
omega=2*%pi*N/60 //rad/s
//Power of the engine:
//Calculating the work done per revolution
WD=integrate('5000+1500*sin(3*theta)','theta',0,2*%pi) //Work done per cycle, N-m
//Calculating the mean resisting torque
Tmean=WD/(2*%pi) //N-m
//Calculating the power of the engine
P=Tmean*omega/1000 //kW
//Maximum fluctuation of the speed of the flywheel when resisting torque is constant:
//Calculating the value of theta
sind3theta=(5000-5000)/1500
theta=1/3*(asind(sind3theta)+180) //degrees
//Calculating the maximum fluctuation of energy
deltaE=integrate('5000+1500*sin(3*theta)-5000','theta',0,60*%pi/180) //N-m
//Calculating the maximum fluctuation of speed of the flywheel
CS1=deltaE/(I*omega^2)*100 //%
//Maximum fluctuation of speed of the flywheel when resisting torque (5000+600*sin(theta)) N-m:
//Calculating the values of theta, thetaB and thetaC
thetaB=asind(sqrt((1/4*(3-600/1500)))) //degrees
thetaC=180-thetaB //degrees
//Calculating the maximum fluctuation of energy
deltaE=round(integrate('(5000+1500*sin(3*theta))-(5000+600*sin(theta))','theta',thetaB*%pi/180,
```

```
thetaC*%pi/180)) //N-m
//Calculating the maximum fluctuation of speed of the flywheel
CS2=abs(deltaE)/(I*omega^2)*100 //%
//Results:
printf("\n\n Power of the engine, P = %.1f kW.\n\n",P)
printf(" Maximum fluctuation of the speed of the flywheel when resisting torque is constant, CS = %.1f %c.\n\n",CS1,"%")
printf(" Maximum fluctuation of speed of the flywheel when resisting torque (5000+600*sin(theta)) N-m, CS = %.3f %c.\n\n",CS2,"%")
```

**Scilab code Exa 16.14** To find diameter and cross section

```
//To find diameter and cross section
clc
//Given:
N=800 //rpm
stroke=300 //mm
sigma=7*10^6 //N/m^2
rho=7200 //kg/m^3
//Solution:
//Refer Fig. 16.18
//Calculating the angular speed of the engine
omega=2*%pi*N/60 //rad/s
//Calculating the coefficient of fluctuation of speed
CS=4/100
//Diameter of the flywheel rim:
//Calculating the peripheral velocity of the flywheel rim
v=sqrt(sigma/rho) //m/s
//Calculating the diameter of the flywheel rim
```

```
D=v*60/(%pi*N) //m
//Cross-section of the flywheel rim:
//Calculating the value of 1 mm^2 on the turning moment diagram
c=500*%pi/30 //Value of 1 mm^2 on the turning moment diagram, N-m
//Calculating the maximum fluctuation of energy
deltaE=round((420-(-30))*c) //N-m
//Calculating the mass of the flywheel rim
m=deltaE/(v^2*CS) //kg
//Calculating the thickness of the flywheel rim
t=sqrt(m/(%pi*D*5*rho))*1000 //mm
//Calculating the width of the flywheel rim
b=5*t //mm
//Results:
printf("\n\n Diameter of the flywheel rim, D = %.3f m.\n\n",D)
printf(" Thickness of the flywheel rim, t = %d mm.\n\n",t)
printf(" Width of the flywheel rim, b = %d mm.\n\n",b)
```

---

**Scilab code Exa 16.15** To find mass and cross section

```
//To find mass and cross section
clc
//Given:
P=150*1000 //W
N=80 //rpm
CE=0.1
D=2, R=D/2 //m
rho=7200 //kg/m^3
//Solution:
//Calculating the angular speed of the engine
omega=2*%pi*N/60 //rad/s
```

```
//Calculating the coefficient of fluctuation of speed
CS=4/100
//Mass of the flywheel rim:
//Calculating the work done per cycle
WD=P*60/N //Work done per cycle, N-m
//Calculating the maximum fluctuation of energy
deltaE=WD*CE //N-m
//Calculating the mass moment of inertia of the flywheel
I=deltaE/(omega^2*CS) //kg-m^2
//Calculating the mass moment of inertia of the flywheel rim
Irim=0.95*I //kg-m^2
//Calculating the mass of the flywheel rim
k=R //Radius of gyration, m
m=Irim/k^2 //kg
//Calculating the cross-sectional area of the flywheel rim
A=m/(2*%pi*R*rho) //m^2
//Resilts:
printf("\n\n Mass of the flywheel rim, m = %d kg.\n\n",m)
printf(" Cross-sectional area of the flywheel rim, A = %.3f m^2.\n\n",A)
```

**Scilab code Exa 16.16** To find MI and dimensions

```
//To find MI and dimensions
clc
//Given:
N=600 //rpm
rho=7250 //kg/m^3
sigma=6*10^6 //N/m^2
//Solution:
```

```
//Refer Fig. 16.19
//Calculating the angular speed of the engine
omega=2*%pi*N/60 //rad/s
//Calculating the total fluctuation of speed
CS=2/100
//Moment of inertia of the flywheel:
//Calculating the value of 1 mm^2 of turning moment
   diagram
c=250*%pi/60 //Value of 1 mm^2 of turning moment
   diagram, N-m
//Calculating the maximum fluctuation of energy
deltaE=round((162-(-35))*c) //N-m
//Calculating the moment of inertia of the flywheel
I=deltaE/(omega^2*CS) //kg-m^2
//Dimensions of the flywheel rim:
//Calculating the peripheral velocity of the
   flywheel
v=sqrt(sigma/rho) //m/s
//Calculating the mean diameter of the flywheel
D=v*60/(%pi*N) //m
//Calculating the maximum fluctuation of energy of
   the flywheel rim
deltaErim=0.92*deltaE //N-m
//Calculating the mass of the flywheel rim
m=deltaErim/(v^2*CS) //kg
//Calculating the thickness of the flywheel rim
t=sqrt(m/(%pi*D*2*rho))*1000 //mm
//Calculating the breadth of the flywheel rim
b=2*t //mm
//Results:
printf("\n\n Moment of inertia of the flywheel, I =
   %.1f kg-m^2.\n\n",I)
printf(" Thickness of the flywheel rim, t = %.1f mm
   .\n\n",t)
printf(" Breadth of the flywheel rim, b = %.1f mm.\n
   \n",b)
```

**Scilab code Exa 16.17** To find MI and size

```
//To find MI and size
clc
//Given:
a1=5*10^-5, a2=21*10^-5, a3=85*10^-5, a4=8*10^-5 //m
    ^2
N2=98, N1=102 //rpm
rho=8150 //kg/m^3
sigma=7.5*10^6 //N/m^2
//Solution:
//Refer Fig. 16.20
//Calculating the net area
a=a3-(a1+a2+a4) //Net area, m^2
//Calculating the value of 1 m^2 on the turning
    moment diagram in terms of work
c=14*10^6 //Value of 1 m^2 on the turning moment
    diagram, N-m
//Calculating the net work done per cycle
WD=a*c //Net work done per cycle, N-m
//Calculating the mean torque on the flywheel
Tmean=WD/(4*%pi) //N-m
FG=Tmean //N-m
//Calculating the work done during expansion stroke
WDe=a3*c //Work done during expansion stroke, N-m
//Calculating the value of AG
AG=WDe/(1/2*%pi) //N-m
//Calculating the excess torque
Texcess=AG-FG //Excess torque, N-m
AF=Texcess //N-m
//Calculating the value of DE
DE=AF/AG*%pi //rad
//Calculating the maximum fluctuation of energy
deltaE=1/2*DE*AF //N-m
```

```
//Moment of inertia of the flywheel:
//Calculating the mean speed during the cycle
N=(N1+N2)/2 //rpm
//Calculating the corresponding angular mean speed
omega=2*%pi*N/60 //rad/s
//Calculating the coefficient of fluctuation of
    speed
CS=(N1-N2)/N
//Calculating the moment of inertia of the flywheel
I=deltaE/(omega^2*CS) //kg-m^2
//Size of flywheel:
//Calculating the peripheral velocity of the
    flywheel
v=sqrt(sigma/rho) //m/s
//Calculating the mean diameter of the flywheel
D=v*60/(%pi*N) //m
//Calculating the mass of the flywheel rim
m=deltaE/(v^2*CS) //kg
//Calculating the thickness of the flywheel rim
t=sqrt(m/(%pi*D*4*rho))*1000 //mm
//Calculating the width of the flywheel rim
b=4*t //mm
//Results:
printf("\n\n Moment of inertia of the flywheel, I =
    %d kg-m^2.\n\n",I)
printf(" Thickness of the flywheel rim, t = %.1f mm
    .\n\n",t)
printf(" Width of the flywheel rim, b = %.1f mm.\n\n
    ",b)
```

**Scilab code Exa 16.18** To find diameter and cross section

```
//To find diameter and cross section
clc
//Given:
```

```
P=50*1000 //W
N=150 //rpm
n=75
sigma=4*10^6 //N/m^2
rho=7200 //kg/m^3
//Solution:
//Refer Fig. 16.21
//Calculating the angular speed of the engine
omega=2*%pi*N/60 //rad/s
//Calculating the mean torque transmitted by the
   flywheel
Tmean=P/omega //N-m
FG=Tmean //N-m
//Calculating the work done per cycle
WD=Tmean*4*%pi //Work done per cycle, N-m
//Calculating the work done during power stroke
WDp=1.4*WD //Work done during power stroke, N-m
//Calculating the maximum torque transmitted by the
   flywheel
Tmax=WDp/(1/2*%pi) //N-m
BF=Tmax //N-m
//Calculating the excess torque
Texcess=Tmax-Tmean //N-m
BG=Texcess //N-m
//Calculating the value of DE
DE=BG/BF*%pi //N-m
//Calculating the maximum fluctuation of energy
deltaE=1/2*DE*BG //N-m
//Mean diameter of the flywheel:
//Calculating the peripheral velocity of the
   flywheel
v=sqrt(sigma/rho) //m/s
//Calculating the mean diameter of the flywheel
D=v*60/(%pi*N) //m
//Cross-sectional dimensions of the rim:
//Calculating the coefficient of fluctuation of
   speed
CS=1/100
```

```
//Calculating the total energy of the flywheel
E=deltaE/(2*CS) //N-m
//Calculating the energy of the rim
Erim=15/16*E //N-m
//Calculating the mass of the flywheel rim
m=Erim/(1/2*v^2) //kg
//Calculating the thickness of the rim
t=round(sqrt(m/(%pi*D*4*rho))*1000) //mm
//Calculating the width of the rim
b=4*t //mm
//Results:
printf("\n\n Mean diameter of the flywheel, D = %d m.\n\n",D)
printf(" Thickness of the flywheel rim, t = %d mm.\n\n",t)
printf(" Width of the flywheel rim, b = %d mm.\n\n",b)
```

**Scilab code Exa 16.19** To find power and mass

```
//To find power and mass
clc
//Given:
N1=225, N2=200 //rpm
k=0.5 //m
E1=15*1000 //N-m
HolePunched=720 //per hour
//Solution:
//Power of the motor:
//Calculating the total energy required per second
E=E1*HolePunched/3600 //N-m/s
//Calculating the power of the motor
P=E/1000 //kW
//Minimum mass of the flywheel:
//Calculating the energy supplied by the motor in 2
```

```
seconds
E2=E*2 //N-m
//Calculating the energy supplied by the flywheel during punching
deltaE=E1-E2 //N-m
//Calculating the mean speed of the flywheel
N=(N1+N2)/2 //rpm
//Calculating the minimum mass of the flywheel
m=round(deltaE*900/(%pi^2*k^2*N*(N1-N2))) //kg
//Results:
printf("\n\n Power of the motor, P = %d kW.\n\n",P)
printf(" Minimum mass of the flywheel, m = %d kg.\n\n",m)
```

**Scilab code Exa 16.20** To find power and mass

```
//To find power and mass
clc
//Given:
d=38, t=32, s=100 //mm
E1=7 //N-m/mm^2 of sheared area
v=25 //m/s
//Solution:
//Power of the motor required:
//Calculating the sheared area
A=round(%pi*d*t) //mm^2
//Calculating the total energy required per hole
E1=E1*A //N-m
//Calculating the energy required for punching work per second
E=E1/10 //Energy required for punching work per second, N-m/s
//Calculating the power of the motor required
P=E/1000 //Power of the motor required, kW
//Mass of the flywheel required:
```

```
//Calculating the time required to punch a hole in a 32 mm thick plate
t32=10/(2*s)*t //Time required to punch a hole in 32 mm thick plate, seconds
//Calculating the energy supplied by the motor in t32 seconds
E2=E*t32 //N-m
//Calculating the energy to be supplied by the flywheel during punching
deltaE=E1-E2 //N-m
//Calculating the coefficient of fluctuation of speed
CS=3/100
//Calculating the mass of the flywheel required
m=round(deltaE/(v^2*CS)) //kg
//Results:
printf("\n\n Power of the motor required, P = %.3f kW.\n\n",P)
printf(" Mass of the flywheel required, m = %d kg.\n\n",m)
```

**Scilab code Exa 16.21** To find speed and number of rivets

```
//To find speed and number of rivets
clc
//Given:
P=3 //kW
m=150 //kg
k=0.6 //m
N1=300 //rpm
//Solution:
//Calculating the angular speed of the flywheel before riveting
omega1=2*%pi*N1/60 //rad/s
//Speed of the flywheel immediately after riveting:
```

```
//Calculating the energy supplied by the motor
E2=P*1000 //N-m/s
//Calculating the energy absorbed during one riveting operation which takes 1 second
E1=10000 //N-m
//Calculating the energy to be supplied by the flywheel for each riveting operation per second
deltaE=E1-E2 //N-m
//Calculating the angular speed of the flywheel immediately after riveting
omega2=sqrt(omega1^2-(2*deltaE/(m*k^2))) //rad/s
//Calculating the corresponding speed in rpm
N2=omega2*60/(2*%pi) //rpm
//Calculating the number of rivets that can be closed per minute
n=E2/E1*60 //Number of rivets that can be closed per minute
//Results:
printf("\n\n Speed of the flywheel immediately after riveting, N2 = %.1f rpm.\n\n",N2)
printf(" Number of rivets that can be closed per minute = %d rivets.\n\n",n)
```

**Scilab code Exa 16.22** To find mass of the flywheel

```
//To find mass of the flywheel
clc
//Given:
d=40, t=15 //mm
NoofHoles=30 //per minute
EnergyRequired=6 //N-m/mm^2
Time=1/10 //seconds
N1=160, N2=140 //rpm
k=1 //m
//Solution:
```

```
//Calculating the sheared area per hole
A=round(%pi*d*t) //Sheared area per hole , mmˆ2
//Calculating the energy required to punch a hole
E1=EnergyRequired*A //N-m
//Calculating the energy required for punching work
    per second
E=E1*NoofHoles/60 //Energy required for punching
    work per second , N-m/s
//Calculating the energy supplied by the motor
    during the time of punching
E2=E*Time //N-m
//Calculating the energy to be supplied by the
    flywheel during punching a hole
deltaE=E1-E2 //N-m
//Calculating the mean speed of the flywheel
N=(N1+N2)/2 //rpm
//Calculating the mass of the flywheel required
m=round(deltaE*900/(%piˆ2*kˆ2*N*(N1-N2))) //kg
//Results:
printf("\n\n Mass of the flywheel required , m = %d
    kg.\n\n",m)
```

**Scilab code Exa 16.23** To find power and cross section

```
//To find power and cross section
clc
//Given:
n=25
d1=25/1000, t1=18/1000, D=1.4, R=D/2 //m
touu=300*10ˆ6 //N/mˆ2
etam=95/100, CS=0.1
sigma=6*10ˆ6 //N/mˆ2
rho=7250 //kg/mˆ3
//Solution:
//Power needed for the driving motor:
```

```
//Calculating the area of the plate sheared
AS=%pi*d1*t1 //m^2
//Calculating the maximum shearing force required for punching
FS=AS*touu //N
//Calculating the energy required per stroke
E=1/2*FS*t1 //Energy required per stroke, N-m
//Calculating the energy required per minute
E1=E*n //Energy required per minute, N-m
//Calculating the power required for the driving motor
P=E1/(60*etam)/1000 //Energy required for the driving motor, kW
//Dimensions for the rim cross-section:
//Calculating the maximum fluctuation of energy
deltaE=9/10*E //N-m
//Calculating the maximum fluctuation of energy provided by the rim
deltaErim=0.95*deltaE //N-m
//Calculating the mean speed of the flywheel
N=9*25 //rpm
//Calculating the mean angular speed
omega=2*%pi*N/60 //rad/s
//Calculating the mass of the flywheel
m=round(deltaErim/(R^2*omega^2*CS)) //kg
//Calculating the thickness of rim
t=sqrt(m/(%pi*D*2*rho))*1000 //mm
//Calculating the width of rim
b=2*t //mm
//Results:
printf("\n\n Power needed for the driving motor = %.3f kW.\n\n",P)
printf(" Thickness of the flywheel rim, t = %d mm.\n\n",t)
printf(" Width of the flywheel rim, b = %d mm.\n\n",b)
```