Contents

8 Packages for Kinematic Chains

8.1 Driver Link

Packages can be used a to calculate the position, velocity, and acceleration of a driver link in rotational motion [Fig. 8.1.(a)]. For the position analysis, the input data are the coordinates (x_A, y_A) of the start joint A with respect to the reference frame $xOyz$, the length of the link AB, and the angle ϕ with the horizontal axis. For the velocity and the acceleration analysis, the angular velocity $\omega = \dot{\phi}$ and the angular acceleration $\alpha = \ddot{\phi}$ are considered. The output data are the position, velocity, and acceleration of the end point B.

The position equations for the driver link are

$$\begin{aligned} x_B &= x_A + AB\cos\phi, \\ y_B &= y_A + AB\sin\phi, \end{aligned} \tag{8.1}$$

where x_B and y_B are the coordinates of the point B.

The velocity equations for the driver link are

$$\begin{aligned} v_{Bx} &= -AB\omega\sin\phi, \\ v_{By} &= AB\omega\cos\phi, \end{aligned} \tag{8.2}$$

where v_{Bx} and v_{By} are the velocity components of the point B on the x- and y-axes.

The acceleration equations for the driver link are

$$\begin{aligned} a_{Bx} &= -AB\omega^2\cos\phi - AB\alpha\sin\phi, \\ a_{By} &= -AB\omega^2\sin\phi + AB\alpha\cos\phi, \end{aligned} \tag{8.3}$$

where a_{Bx} and a_{By} are the acceleration components of the point B on the x- and y-axes.

In order to compute the position, velocity, and acceleration of the joint B, using *Mathematica*^{TM}, the necessary commands can be can collected in a function. The name of the function is **Driver**.

```
Driver[xA_,yA_,AB_,phi_,omega_,alpha_]:=
Block[{ xB, yB, vBx, vBy, aBx, aBy },
xB = xA + AB Cos[phi] ;
```

```
yB = yA + AB Sin[phi] ;
vBx = - AB omega Sin[phi] ;
vBy = AB omega Cos[phi] ;
aBx = - AB omega^2 Cos[phi] - AB alpha Sin[phi] ;
aBy = - AB omega^2 Sin[phi] + AB alpha Cos[phi] ;
Return[{ xB, yB, vBx, vBy, aBx, aBy } ] ;
] ;
```

The input data, the variable parts of the computation, and the output data are defined as parameters to this function.

All the variables local to **Driver[]** are declared in the **Block[]** statement to isolate them from any values they might have globally. The *Mathematica*^{TM} command **Block[{x, y, ...}, expr]** specifies that **expr** is to be evaluated with local values for the symbols **x, y, ...** . In our case, the local variables are **xB, yB, vBx, vBy, aBx, aBy**, and **expr** is the body of the function.

The *Mathematica*^{TM} command **Return[expr]** returns the value **expr** from a function. For the driver, **expr** represents the output data and it is a vector that contains the elements **xB, yB, vBx, vBy, aBx**, and **aBy**.

The mechanism that *Mathematica*^{TM} provides for keeping the variables used in a package different from those used in the main session is called *context*. As each symbol is read from the terminal or from a file, *Mathematica*^{TM} checks to see whether this symbol has already been used before. If it has been encountered before, the new instance is made to refer to that previously read symbol. If the symbol has not been encountered before, a new entry in the symbol table is created. Each symbol belongs to a certain context. Within one context the names of the symbols are unique, but the same name can occur in two different contexts. For the driver the proper context is

```
Driver::usage = "Driver[xA,yA,AB,phi,omega,alpha]
computes the driver link position,
velocity and acceleration vectors."
Begin["Private`"]
Driver[xA_,yA_,AB_,phi_,omega_,alpha_]:=
Block[ { xB, yB, vBx, vBy, aBx, aBy },
xB = xA + AB Cos[phi] ;
yB = yA + AB Sin[phi] ;
vBx = - AB omega Sin[phi] ;
```

```
vBy = AB omega Cos[phi] ;
aBx = - AB omega^2 Cos[phi] - AB alpha Sin[phi] ;
aBy = - AB omega^2 Sin[phi] + AB alpha Cos[phi] ;
Return[ { xB, yB, vBx, vBy, aBx, aBy } ] ;
]
End[ ]
```

The local variables **xB, yB, vBx, vBy, aBx**, and **aBy** are now created in the context **Private‘** which is not searched when one types a variable name later on.

The usage message defined for the symbol **Driver** is there to provide documentation for the function and to make sure that **Driver** is defined in the current context. If it had not been defined before entering the context **Private‘**, it would not be found later on.

The *Mathematica*^{TM} command **End[]** returns the present context and reverts to the previous one.

The functions that the package provides are put into a separate context which must be visible to be able to use the functions later on. This can be done using the pair of *Mathematica*^{TM} commands **BeginPackage[]** and **EndPackage[]**. Thus, the following *Mathematica*^{TM} package is introduced:

```
BeginPackage["Driver‘"]
Driver::usage = "Driver[xA_,yA_,AB_,phi_,omega_,alpha_]
computes the driver position, velocity and acceleration
vectors."
Begin["‘Private‘"]
Driver[xA_,yA_,AB_,phi_,omega_,alpha_]:=
Block[ { xB, yB, vBx, vBy, aBx, aBy },
xB = xA + AB Cos[phi] ;
yB = yA + AB Sin[phi] ;
vBx = - AB omega Sin[phi] ;
vBy = AB omega Cos[phi] ;
aBx = - AB omega^2 Cos[phi] - AB alpha Sin[phi] ;
aBy = - AB omega^2 Sin[phi] + AB alpha Cos[phi] ;
Return[ { xB, yB, vBx, vBy, aBx, aBy } ] ;
]
End[ ]
```

```
EndPackage[ ]
```

The command **BeginPackage["Driver‘"]** sets **Driver‘** to be the current context, and the command **EndPackage[]** ends the package, prepending **Driver‘** to the context search path.

Note the initial backquote in the context name inside the command **Begin["‘Private‘"]**. This establishes **‘Private‘** as a subcontext of the context **Driver‘** (so its full name is **Driver‘Private‘**).

The name of the source file for the *Mathematica*^{TM} package **Driver** is **Driver.m** as shown in Program 8.1.

Example

A driver link is shown in Fig. 8.1(b). The input data are $AB = 0.20$ m, the angle between the driver link AB and the horizontal axis, $\phi = 30°$, and the angular velocity, $\omega = 5$ rad/s. Calculate the position, velocity, and acceleration components of the joint B. The cartesian reference frame $xOyz$ is chosen with $A \equiv O$.

The *Mathematica*^{TM} package **Driver** is loaded in the main *Mathematica*^{TM} session using the command

```
<<Driver.m ;
```

To compute the numerical values of the position, velocity, and acceleration components for the joint B, the *Mathematica*^{TM} function **Driver** is used.

Position analysis

Since the joint A is the origin of the reference frame $xAyz$, the coordinates of the joint A are

$$x_A = y_A = 0.$$

The coordinates of the joint B are

$$\begin{aligned} x_B &= x_A + AB\cos\phi = 0.173 \text{ m}, \\ y_B &= y_A + AB\sin\phi = 0.10 \text{ m}. \end{aligned} \tag{8.4}$$

The coordinates x_B and y_B are the first and, respectively, the second component of the vector returned by the function **Driver**:

```
xB = Driver[xA,yA,AB,phi,omega,alpha][[1]] ;
yB = Driver[xA,yA,AB,phi,omega,alpha][[2]] ;
```

Velocity analysis

The components v_{Bx} on the x-axis and v_{By} on the y-axis of the velocity for the joint B are

$$v_{Bx} = -AB\omega \sin\phi = -0.50 \text{ m/s},$$
$$v_{By} = AB\omega \cos\phi = 0.866 \text{ m/s}. \quad (8.5)$$

The components v_{Bx} and v_{By} are the third and, respectively, the fourth component of the vector returned by the function **Driver**:

```
vBx = Driver[xA,yA,AB,phi,omega,alpha][[3]] ;
vBy = Driver[xA,yA,AB,phi,omega,alpha][[4]] ;
```

Acceleration analysis

The components a_{Bx} on the x-axis and a_{By} on the y-axis of the velocity for the joint B are

$$a_{Bx} = -AB\omega^2 \cos\phi - AB\alpha \sin\phi = -4.330 \text{ m/s}^2,$$
$$a_{By} = -AB\omega^2 \sin\phi + AB\alpha \cos\phi = -2.50 \text{ m/s}^2. \quad (8.6)$$

The components a_{Bx} and a_{By} are the fifth and, respectively, the sixth component of the vector returned by the function **Driver**:

```
aBx = Driver[xA,yA,AB,phi,omega,alpha][[5]] ;
aBy = Driver[xA,yA,AB,phi,omega,alpha][[6]] ;
```

The *Mathematica*^{TM} program and the numerical results are shown in Program 8.2.

8.2 Position Analysis

RRR dyad

The RRR dyad is shown in Fig. 8.2(a). The input data are the coordinates of the joint $M(x_M, y_M)$, the coordinates of the joint $N(x_N, y_N)$, and the lengths of the segments MP and NP. The output data are the coordinates of the joint $P(x_P, y_P)$.

The position equations for the RRR dyad are

$$(x_M - x_P)^2 + (y_M - y_P)^2 = MP^2,$$
$$(x_N - x_P)^2 + (y_N - y_P)^2 = NP^2, \tag{8.7}$$

where the unknowns are the coordinates x_P and y_P of the joint P. There are two solutions for the position of the joint P: (x_{P1}, y_{P1}) and (x_{P2}, y_{P2}).

The *Mathematica*^{TM} function for the positions of x_{P_1}, y_{P_1}, x_{P_2}, y_{P_2} is

```
PosRRR::usage = "PosRRR[xM,yM,xN,yN,MP,NP]
Computes the position vectors for RRR dyad"
Begin["'Private'"]
PosRRR[xM_,yM_,xN_,yN_,MP_,NP_]:=
Block[
{xPSol,yPSol,xP1,yP1,xP2,yP2,eqRRR1,eqRRR2,solRRR},
eqRRR1 = (xM-xPSol)^2 + (yM-yPSol)^2 == MP^2 ;
eqRRR2 = (xN-xPSol)^2 + (yN-yPSol)^2 == NP^2 ;
solRRR = Solve[{eqRRR1,eqRRR2},{xPSol,yPSol}];
xP1 = xPSol/.solRRR[[1]] ;
yP1 = yPSol/.solRRR[[1]] ;
xP2 = xPSol/.solRRR[[2]] ;
yP2 = yPSol/.solRRR[[2]] ;
Return[xP1, yP1, xP2, yP2] ;
]
End[ ]
```

RRT dyad

The RRT dyad is shown in Fig. 8.2(b). The input data are the coordinates of the joint $M(x_M, y_M)$, the coordinates of the point $N(x_N, y_N)$ on the sliding direction, the length of the segment MP, and the value of the angle θ. The output data are the coordinates of the joint $P(x_P, y_P)$.

The position equations for the RRT dyad are

$$(x_M - x_P)^2 + (y_M - y_P)^2 = MP^2,$$
$$\tan\theta = \frac{y_P - y_N}{x_P - x_N}, \tag{8.8}$$

where the unknowns are the coordinates x_P and y_P of the joint P. There are two solutions for the position of the joint P, those are (x_{P1}, y_{P1}) and (x_{P2}, y_{P2}).

If the value of the angle θ is 90° or 180°, then $x_P = x_N$ and the following equation is used to find the coordinate y_P of the point P:

$$(x_M - x_N)^2 + (y_M - y_P)^2 = MP^2. \tag{8.9}$$

The *Mathematica*^{TM} function for the position analysis is

```
PosRRT::usage = "PosRRT[xM,yM,xN,yN,MP,theta]
Computes the position vectors for RRT dyad"
Begin["‘Private‘"]
PosRRT[xM_,yM_,xN_,yN_,MP_,theta_]:=
Block[
{xPSol, yPSol, xP1, yP1,xP2, yP2, eqRRT, solRRT, eqRRT1,
eqRRT2 },
If[ (theta==Pi/2)||(theta==3*Pi/2),
xP1 = xP2 = xN ;
eqRRT = (xM-xN)^2 + (yM-yPSol)^2 == MP^2 ;
solRRT = Solve[ eqRRT, yPSol ] ;
yP1 = yPSol/.solRRT[[1]] ;
yP2 = yPSol/.solRRT[[2]] ,
eqRRT1 = (xM-xPSol)^2 + (yM-yPSol)^2 == MP^2 ;
eqRRT2 = Tan[theta] == (yPSol-yN)/(xPSol-xN) ;
solRRT = Solve[{eqRRT1,eqRRT2},{xPSol,yPSol}] ;
xP1 = xPSol/.solRRT[[1]] ;
yP1 = yPSol/.solRRT[[1]] ;
xP2 = xPSol/.solRRT[[2]] ;
yP2 = yPSol/.solRRT[[2]] ;
] ;
Return[ { xP1, yP1, xP2, yP2 } ] ;
]
End[ ]
```

The functions **PosRRR** and **PosRRT** are included in the *Mathematica*^{TM} package **Position**. The name of the source file for the package is **Position.m** and is given in Program 8.3.

R-RTR-RRT Mechanism

The planar R-RTR-RRT mechanism considered is shown in Fig. 8.3(a). Given the input data $AB = 0.20$ m, $AD = 0.40$ m, $CD = 0.70$ m, $CE = 0.30$ m, the angle of the driver link AB with the horizontal axis, $\phi = 45°$, and the angular velocity, $\omega = 5$ rad/s, calculate the positions of the joints. The distance from the slider 5 to the horizontal axis Ox is $y_E = 0.35$ m. The cartesian reference frame $xOyz$ is chosen with $A \equiv O$.

The *Mathematica*^{TM} packages **`Driver`** and **`Position`** are loaded in the main program using the commands

```
<<Driver.m ;
<<Position.m ;
```

Position of the joint A

Since the joint A is the origin of the reference frame $xAyz$, the coordinates of the joint A are

$$x_A = y_A = 0.$$

Position of the joint B

The coordinates of the joint B are

$$x_B = x_A + AB\cos\phi = 0.141 \text{ m},$$
$$y_B = y_A + AB\sin\phi = 0.141 \text{ m}. \tag{8.10}$$

The numerical values for the coordinates of the joint B are obtained using the *Mathematica*^{TM} function **`Driver`**:

```
xB = Driver[xA,yA,AB,phi,omega,alpha][[1]] ;
yB = Driver[xA,xB,AB,phi,omega,alpha][[2]] ;
```

The RTR (BBD) dyad is represented in Fig. 8.3(b).

Position of the joint D

The coordinates of the joint D are

$$x_D = 0, \; y_D = -AD = -0.400 \text{ m}.$$

The angle ϕ_3 is

$$\phi_3 = \arctan\frac{y_B - y_D}{x_B - x_D} = 75.36°.$$

The next dyad RRT (CEE) is represented in Fig. 8.3(c).

Position of the joint C
The coordinates of the joint C are

$$x_C = x_D + CD \cos \phi_3 = 0.176 \text{ m},$$
$$y_C = y_D + CD \sin \phi_3 = 0.277 \text{ m}.$$

Position of the joint E
In this particular case, the coordinate y_E of the joint E is constant:

$$y_E = 0.350\text{m}.$$

The coordinate x_E of the joint E is calculated using the equation

$$(x_C - x_E)^2 + (y_C - y_E)^2 = CE^2. \tag{8.11}$$

There are two solutions for x_E:

$$x_{E1} = -0.114 \text{ m, and } x_{E2} = 0.467 \text{ m}.$$

The correct solution for x_E is selected using the condition $x_E < x_C$:

$$x_E = -0.114 \text{ m}.$$

The numerical solution for x_E is obtained using the *Mathematica*^{TM} function **PosRRT**

```
xE1 = PosRRT[xC,yC,0,yE,CE,phi5][[1]] ;
xE2 = PosRRT[xC,yC,0,yE,CE,phi5][[3]] ;
(* Choose the correct solution *)
If[ (xE1<xC), xE=xE1, xE=xE2 ] ;
```

The input data are the coordinates of the joint $C(x_C, y_C)$, the coordinates of the point $P(0, y_E)$ located on the sliding direction, the length of the link CE, and the angle between the sliding direction and the horizontal axis Ox, **phi5**$=180°$.

The output data are the first and the third element of the vector returned by the function **PosRRT**, which are the x-coordinates of the joint E. The second and the fourth element are constant and equal to the y-coordinate of the joint E.

The numerical values are printed using the *Mathematica*^{TM} commands:

```
Print["rB = ",{xB,yB,0}," [m]" ];
Print["rC = ",{xC,yC,0}," [m]" ];
Print["rE = ",{xE,yE,0}," [m]" ];
```

The *Mathematica*^{TM} program and the numerical results are shown in Program 8.4.

R-RRR-RRT Mechanism

The planar R-RRR-RRT mechanism considered is shown in Fig. 8.4(a). Given the input data $AB = 0.15$ m, $BC = 0.40$ m, $CD = 0.37$ m, $CE = 0.23$ m, $EF = CE$, $La = 0.30$ m, $Lb = 0.45$ m, $Lc = CD$, and the angle of the driver link AB with the horizontal axis, $\phi = 45°$, calculate the positions of the joints. The distance from the slider 5 to the horizontal axis Ox is $y_E = 0.35$ m. The cartesian reference frame $xOyz$ is chosen, with $A \equiv O$.

The *Mathematica*^{TM} packages **`Driver`** and **`Position`** are loaded in the main program using the commands:

```
<<Driver.m ;
<<Position.m ;
```

Position of the joint A

Since the joint A is the origin of the reference frame $xAyz$, the coordinates of the joint A are

$$x_A = y_A = 0.$$

Position of the joint B

The coordinates of the joint B are

$$\begin{aligned} x_B &= x_A + AB\cos\phi = 0.106 \text{ m}, \\ y_B &= y_A + AB\sin\phi = 0.106 \text{ m}. \end{aligned} \tag{8.12}$$

The numerical values for the coordinates of the joint B are obtained using the *Mathematica*^{TM} function **`Driver`**:

```
xB = Driver[xA,yA,AB,phi,omega,alpha][[1]] ;
```

```
yB = Driver[xA,xB,AB,phi,omega,alpha][[2]] ;
```

The RRR (BCD) dyad is represented in Fig. 8.4(b).

Position of the joint D
The coordinates of the joint D are

$$x_D = L_a = 0.300 \text{ m}, \; y_D = L_b = 0.450 \text{ m}.$$

Position of the joint C
The coordinates x_C and y_C of the joint C are calculated using the equations

$$\begin{aligned}(x_B - x_C)^2 + (y_B - y_C)^2 &= BC^2,\\ (x_D - x_C)^2 + (y_D - y_C)^2 &= CD^2.\end{aligned} \tag{8.13}$$

There are two solutions for the coordinates of the joint C:

$$\begin{aligned}&x_{C1} = -0.069 \text{ m}, \; y_{C1} = 0.465 \text{ m},\\ &x_{C2} = 0.504 \text{ m} \; , y_{C2} = 0.141 \text{ m}.\end{aligned}$$

The correct solution is selected using the condition $y_C > y_B$:

$$x_{C1} = -0.069 \text{ m}, \; y_{C1} = 0.465 \text{ m}.$$

The numerical solutions for the coordinates of the joint C using the *Mathematica*^{TM} commands are

```
xC1 = PosRRR[xB,yB,xD,yD,BC,CD][[1]] ;
yC1 = PosRRR[xB,yB,xD,yD,BC,CD][[2]] ;
xC2 = PosRRR[xB,yB,xD,yD,BC,CD][[3]] ;
yC2 = PosRRR[xB,yB,xD,yD,BC,CD][[4]] ;
(* Choose the correct solution *)
If[ (yC1>yB), xC=xC1;yC=yC1, xC=xC2;yC=yC2 ] ;
```

The input data are the coordinates of the joint $B(x_B, y_B)$, the coordinates of the joint $D(x_D, y_D)$, and the lengths of the links BC and CD. The output data are the four elements of the vector returned by the function **PosRRR**, which are the coordinates of the joint C.

The angle ϕ_3 between the link 3 and the horizontal axis Ox is

$$\phi_3 = \arctan\frac{y_C - y_D}{x_C - x_D} + \pi = 3.099 \text{ rad.}$$

Position of the joint E
The coordinates of the joint E are

$$x_E = x_C + CE\cos\phi_3 = -0.299 \text{ m},$$
$$y_E = y_C + CE\sin\phi_3 = 0.474 \text{ m}.$$

The next dyad RRT (CEE) is represented in Fig. 8.4(c).

Position of the joint F
In this particular case, the x-coordinate x_F of the joint F is constant:

$$x_F = -L_c = -0.370 \text{ m}.$$

The y-coordinate y_F of the joint F is calculated using the equation

$$(x_E - x_F)^2 + (y_E - y_F)^2 = EF^2. \qquad (8.14)$$

There are two solutions for y_F:

$$y_{F1} = 0.256 \text{ m, and } y_{F2} = 0.693 \text{ m}.$$

The correct solution for y_F is chosen using the condition $y_F < y_E$:

$$y_F = 0.256 \text{ m}.$$

The numerical solutions for y_F using the *Mathematica*^{TM} commands are

```
yF1 = PosRRT[xE,yE,-Lc,0,EF,phi5][[2]] ;
yF2 = PosRRT[xE,yE,-Lc,0,EF,phi5][[4]] ;
(* Choose the correct solution *)
If[ (yF1<yE), yF=yF1, yF=yF2 ] ;
```

The input data are the coordinates of the joint $E(x_E, y_E)$, the coordinates of the point $P(-L_c, 0)$ on the sliding direction, the length of the link EF, and the angle between the sliding direction and the horizontal axis Ox, **phi5**=90°. The output data are the second and the fourth element of the vector returned by the function **PosRRT**, which are the y-coordinates of the joint F. The first and third elements are constant and equal to the x-coordinate of the joint F. The numerical values are printed using the *Mathematica*^{TM} commands:

```
Print["rB = ",{xB,yB,0}," [m]" ];
Print["rC = ",{xC,yC,0}," [m]" ];
Print["rE = ",{xE,yE,0}," [m]" ];
Print["rF = ",{xF,yF,0}," [m]" ];
```

The angle ϕ_2 between the link 2 and the horizontal axis Ox is

$$\phi_2 = \arctan\frac{y_C - y_B}{x_C - x_B} + \pi = 2.025 \text{ rad}.$$

The angle ϕ_4 between the link 4 and the horizontal axis Ox is

$$\phi_4 = \arctan\frac{y_E - y_F}{x_E - x_F} = 1.259 \text{ rad}.$$

The *Mathematica*^{TM} program and the numerical results are shown in Program 8.5.

R-RRT Mechanism

The planar R-RRT mechanism considered is shown in Fig. 8.5. Given the input data $AC = 0.10$ m, $BC = 0.30$ m, $AP = 0.50$ m, and the angle of the driver link AB with the horizontal axis, $\phi = 45°$, calculate the positions of the joints. The cartesian reference frame $xOyz$ is chosen with $A \equiv O$.

The *Mathematica*^{TM} packages **Driver** and **Position** are loaded in the main program using the commands

```
<<Driver.m ;
<<Position.m ;
```

Position of the joint A

Since the joint A is the origin of the reference frame $xAyz$, the coordinates of the joint A are

$$x_A = y_A = 0.$$

Position of the joint B

In order to calculate the position of the point B, the position of the point P located on the driver link AB is calculated with

$$x_P = x_A + AP\cos\phi = 0.353 \text{ m},$$
$$y_P = y_A + AP\sin\phi = 0.353 \text{ m}. \quad (8.15)$$

The numerical values for the coordinates of the point P are obtained using the *Mathematica*^{TM} function **Driver**:

```
xP = Driver[xA,yA,AP,phi,omega,alpha][[1]] ;
yP = Driver[xA,xB,AP,phi,omega,alpha][[2]] ;
```

The coordinates of the point B are calculated using the equations

$$(x_C - x_B)^2 + (y_C - y_B)^2 = BC^2,$$
$$\tan\theta = \frac{y_B - y_P}{x_B - x_P}.$$

There are two solutions for the coordinates of the point B:

$$x_{B1} = -0.156 \text{ m}, \quad y_{B1} = -0.156 \text{ m},$$
$$x_{B2} = 0.256 \text{ m}, \quad y_{B2} = 0.256 \text{ m}.$$

The correct solution is selected using the condition $y_B > y_C = 0$:

$$y_B = y_{B2} = 0.256 \text{ m}.$$

The numerical solutions for the coordinates of the point B are obtained using the *Mathematica*^{TM} commands

```
xB1 = PosRRT[xC,yC,xP,yP,BC,phi][[1]] ;
yB1 = PosRRT[xC,yC,xP,yP,BC,phi][[2]] ;
xB2 = PosRRT[xC,yC,xP,yP,BC,phi][[3]] ;
yB2 = PosRRT[xC,yC,xP,yP,BC,phi][[4]] ;
(* Choose the correct solution *)
If[ (yB1>yC), xB=xB1;yB=yB1, xB=xB2;yB=yB2 ] ;
```

The numerical values are printed using the *Mathematica*^{TM} command

```
Print["rB = ",{xB,yB,0}," [m]" ];
```

The *Mathematica*^{TM} program and the numerical results are shown in Program 8.6.

8.3 Velocity and Acceleration Analysis

RRR dyad

The input data are the coordinates $x_M, y_M, x_N, y_N, x_P, y_P$ of the joints M, N, and P, the velocities $\dot{x}_M, \dot{y}_M, \dot{x}_N, \dot{y}_N$, and the accelerations $\ddot{x}_M, \ddot{y}_M, \ddot{x}_N, \ddot{y}_N$ of the joints M and N. The output data are the velocities $\dot{x}_P, \dot{y}_P$ and acceleration components $\ddot{x}_P$ and $\ddot{y}_P$ of the joint P.

The velocity equations for the RRR dyad are obtained taking the derivative of the position equations

$$
\begin{aligned}
&(x_M - x_P)(\dot{x}_M - \dot{x}_P) + (y_M - y_P)(\dot{y}_M - \dot{y}_P) = 0,\\
&(x_N - x_P)(\dot{x}_N - \dot{x}_P) + (y_N - y_P)(\dot{y}_N - \dot{y}_P) = 0, \qquad (8.16)
\end{aligned}
$$

where the unknowns are the velocity components $\dot{x}_P$ and $\dot{y}_P$ of the joint P.

The acceleration equations for the RRR dyad are obtained taking the derivative of the velocity equations

$$
\begin{aligned}
&(x_M - x_P)(\ddot{x}_M - \ddot{x}_P) + (\dot{x}_M - \dot{x}_P)^2 + (y_M - y_P)(\ddot{y}_M - \ddot{y}_P)\\
&+(\dot{y}_M - \dot{y}_P)^2 = 0,\\
&(x_N - x_P)(\ddot{x}_N - \ddot{x}_P) + (\dot{x}_N - \dot{x}_P)^2 + (y_N - y_P)(\ddot{y}_N - \ddot{y}_P)\\
&+(\dot{y}_N - \dot{y}_P)^2 = 0, \qquad (8.17)
\end{aligned}
$$

where the unknowns are the acceleration components $\ddot{x}_P$ and $\ddot{y}_P$ of the joint P.

The *Mathematica*^{TM} function for the velocity and acceleration analysis is

```
VelAccRRR::usage = "VelAccRRR[xM,yM,xN,yN,xP,yP,
vMx,vMy,vNx,vNy,aMx,aMy,aNx,aNy] computes the velocity
and acceleration vectors for RRR dyad"
Begin["‘Private‘"]
VelAccRRR[xM_,yM_,xN_,yN_,xP_,yP_,vMx_,vMy_,vNx_,vNy_,
aMx_,aMy_, aNx_,aNy_]:=
Block[
```

```
   { vPxSol, vPySol, aPxSol, aPySol, vPx, vPy, aPx, aPy,
eqRRR1v, eqRRR2v, solRRRv, eqRRR1a, eqRRR2a, solRRRa },
   (* Velocities *)
   eqRRR1v=(xM-xP) (vMx-vPxSol)+(yM-yP) (vMy-vPySol)==0;
   eqRRR2v=(xN-xP) (vNx-vPxSol)+(yN-yP) (vNy-vPySol)==0;
   solRRRv=Solve[{eqRRR1v, eqRRR2v},{vPxSol, vPySol}];
   vPx = vPxSol/.solRRRv[[1]] ;
   vPy = vPySol/.solRRRv[[1]] ;
   (* Accelerations *)
   eqRRR1a = (xM-xP) (aMx-aPxSol) + (vMx-vPx)^2 +
(yM-yP) (aMy-aPySol) + (vMy-vPy)^2 == 0 ;
   eqRRR2a = (xN-xP) (aNx-aPxSol) + (vNx-vPx)^2 +
(yN-yP) (aNy-aPySol) + (vNy-vPy)^2 == 0 ;
   solRRRa=Solve[{eqRRR1a, eqRRR2a},{aPxSol, aPySol}];
   aPx = aPxSol/.solRRRa[[1]] ;
   aPy = aPySol/.solRRRa[[1]] ;
   Return[ { vPx, vPy, aPx, aPy } ] ; ]
   End[ ]
```

RRT dyad

The input data are the coordinates $x_M, y_M, x_N, y_N, x_P, y_P$ of the joints M, N, and P, the velocities $\dot{x}_M, \dot{y}_M, \dot{x}_N, \dot{y}_N$, the accelerations $\ddot{x}_M, \ddot{y}_M, \ddot{x}_N, \ddot{y}_N$ of the joints M and N, the angle θ, the angular velocity, and acceleration $\dot{\theta}$ and $\ddot{\theta}$. The output data are the velocities $\dot{x}_P, \dot{y}_P$, and accelerations $\ddot{x}_P, \ddot{y}_P$ of the joint P.

The velocity equations for the RRT dyad are obtained taking the derivative of the position equations

$$
\begin{aligned}
&(x_M - x_P)(\dot{x}_M - \dot{x}_P) + (y_M - y_P)(\dot{y}_M - \dot{y}_P) = 0,\\
&(\dot{x}_P - \dot{x}_N)\sin\theta + \dot{\theta}(x_P - x_N)\cos\theta - (\dot{y}_P - \dot{y}_N)\cos\theta\\
&+\dot{\theta}(y_P - y_N)\sin\theta = 0,
\end{aligned}
\tag{8.18}
$$

where the unknowns are the velocity components $\dot{x}_P$ and $\dot{y}_P$ of the joint P.

The acceleration equations for the RRT dyad are obtained taking the derivative of the velocity equations

$$
\begin{aligned}
&(x_M - x_P)(\ddot{x}_M - \ddot{x}_P) + (\dot{x}_M - \dot{x}_P)^2 + (y_M - y_P)(\ddot{y}_M - \ddot{y}_P)\\
&+(\dot{y}_M - \dot{y}_P)^2 = 0,
\end{aligned}
$$

$$(\ddot{x}_P - \ddot{x}_N)\sin\theta - (\ddot{y}_P - \ddot{y}_N)\cos\theta + [2(\dot{x}_P - \dot{x}_N)\cos\theta - \dot{\theta}(x_P - x_N)\sin\theta$$
$$+2(\dot{y}_P - \dot{y}_N)\sin\theta + \dot{\theta}(y_P - y_N)\cos\theta]\dot{\theta} + [(x_P - x_N)\cos\theta$$
$$+(y_P - y_N)\sin\theta]\ddot{\theta} = 0, \quad (8.19)$$

where the unknowns are the acceleration components $\ddot{x}_P$ and $\ddot{y}_P$ of the joint P.

The *Mathematica*^{TM} function for the velocity and acceleration analysis is

```
   VelAccRRT::usage = "VelAccRRT[xM,yM,xN,yN,xP,yP,
vMx,vMy,vNx,vNy,aMx,aMy,aNx,aNy,theta,omega,alpha]
computes the velocity and acceleration vectors for
RRT dyad"
   Begin["‘Private‘"]
   VelAccRRT[xM_,yM_,xN_,yN_,xP_,yP_,vMx_,vMy_,vNx_,vNy_,
aMx_,aMy_,aNx_,aNy_,theta_,omega_,alpha_]:=
   Block[
   { vPxSol, vPySol, aPxSol, aPySol, vPx, vPy, aPx, aPy,
eqRRTv, eqRRTa, eqRRT1v, eqRRT2v, solRRTv ,
eqRRR1a, eqRRR2a, solRRRa },
   (* Velocity *)
   eqRRT1v=(xM-xP) (vMx-vPxSol)+(yM-yP) (vMy-vPySol)==0;
   eqRRT2v = Sin[theta] (vPxSol-vNx) +
Cos[theta] omega (xP-xN) - Cos[theta] (vPySol-vNy) +
Sin[theta] omega (yP-yN) == 0 ;
   solRRTv=Solve[{eqRRT1v, eqRRT2v},{vPxSol, vPySol}] ;
   vPx = vPxSol/.solRRTv[[1]] ;
   vPy = vPySol/.solRRTv[[1]] ;
   (* Acceleration *)
   eqRRT1a = (xM-xP) (aMx-aPxSol) + (vMx-vPx)^2 +
(yM-yP) (aMy-aPySol) + (vMy-vPy)^2 == 0 ;
   eqRRT2a = Sin[theta] (aPxSol-aNx) -
Cos[theta] (aPySol-aNy) + ( 2 Cos[theta] (vPx-vNx) -
Sin[theta] dtheta (xP-xN) + 2 Sin[theta] (vPy-vNy) +
Cos[theta] dtheta (yP-yN)) dtheta +
(Cos[theta] (xP-xN) + Sin[theta] (yP-yN)) ddtheta == 0;
   solRRTa=Solve[{eqRRT1a, eqRRT2a},{aPxSol, aPySol}];
   aPx = aPxSol/.solRRTa[[1]] ;
   aPy = aPySol/.solRRTa[[1]] ;
```

```
Return[ { vPx, vPy, aPx, aPy } ] ; ]
End[ ]
```

Angular velocities and accelerations

A *Mathematica*^{TM} function is used to compute the angular velocity and acceleration of a link. The input data are the coordinates x_M, y_M, x_N, y_N, the velocities $\dot{x}_M, \dot{y}_M, \dot{x}_N, \dot{y}_N$, the accelerations $\ddot{x}_M, \ddot{y}_M, \ddot{x}_N, \ddot{y}_N$ of two points M and N located on the link direction, and the angle θ between the link direction and the horizontal axis. The output data are the angular velocity $\omega = \dot{\theta}$ and the angular acceleration $\alpha = \ddot{\theta}$ of the link.

The slope of the line MN is

$$\tan\theta = \frac{y_M - y_N}{x_M - x_N}. \tag{8.20}$$

The derivative with respect to time of the Eq. (8.20) is

$$[(y_M - y_N)\sin\theta + (x_M - x_N)\cos\theta]\omega = (v_{My} - v_{Ny})\cos\theta - (v_{Mx} - v_{Nx})\sin\theta. \tag{8.21}$$

The angular velocity ω is calculated from Eq. (8.21). The derivative with respect to time of Eq. (8.21) is

$$\begin{aligned} &[(x_M - x_N)\cos\theta + (y_M - y_N)\sin\theta]\alpha = (a_{My} - a_{Ny})\cos\theta \\ &-(a_{Mx} - a_{Nx})\sin\theta - [(y_M - y_N)\omega\cos\theta + 2(v_{My} - v_{Ny})\sin\theta \\ &-(x_M - x_N)\omega\sin\theta + 2(v_{Mx} - v_{Nx})\cos\theta]\omega. \end{aligned} \tag{8.22}$$

Solving Eq. (8.22), the angular acceleration α is obtained.

The *Mathematica*^{TM} function for the angular velocity and acceleration analysis is

```
AngVelAcc::usage = "AngVelAcc[xM,yM,xN,yN,vMx,vMy,
vNx,vNy,aMx,aMy,aNx,aNy,theta] computes the angular velocity
and acceleration of a link."
Begin["‘Private‘"]
AngVelAcc[xM_,yM_,xN_,yN_,vMx_,vMy_,vNx_,vNy_,aMx_,aMy_,
aNx_,aNy_,theta_]:=
Block[
{ dtheta, ddtheta },
```

```
  dtheta = ( Cos[theta] (vMy-vNy) - Sin[theta] (vMx-vNx))/
( Sin[theta] (yM-yN) + Cos[theta] (xM-xN) ) ;
  ddtheta=(Cos[theta](aMy-aNy)-Sin[theta](aMx-aNx)-
(Cos[theta] dtheta (yM-yN) + 2 Sin[theta] (vMy-vNy) -
Sin[theta]dtheta(xM-xN)+2Cos[theta](vMx-vNx))dtheta)/
( Cos[theta] (xM-xN) + Sin[theta] (yM-yN) ) ;
  Return[ { dtheta, ddtheta } ] ; ]
  End[ ]
```

Absolute velocities and accelerations

A function is used to compute the velocity and acceleration of the point N, knowing the velocity and acceleration of the point M, both points N and M are located on a rigid link. The input data are the coordinates x_M, y_M, x_N, and y_N of the points M and N, the velocity and acceleration components $\dot{x}_M$, $\dot{y}_M$, $\ddot{x}_M$, $\ddot{y}_M$ of the point M, and the angular velocity and acceleration θ and α of the link. The output data are the velocity and acceleration components $\dot{x}_N$, $\dot{y}_N$, $\ddot{x}_N$, $\ddot{y}_N$ of the point N.

The following vectorial equation between the velocities $\mathbf{v}_N$ and $\mathbf{v}_M$ of the points N and M exists as

$$\mathbf{v}_N = \mathbf{v}_M + \boldsymbol{\omega} \times \mathbf{r}_{MN}, \tag{8.23}$$

where $\mathbf{v}_N = \dot{x}_N\mathbf{\imath} + \dot{y}_N\mathbf{\jmath}$, $\mathbf{v}_M = \dot{x}_M\mathbf{\imath} + \dot{y}_M\mathbf{\jmath}$, $\boldsymbol{\omega} = \omega\mathbf{k}$, and $\mathbf{r}_{MN} = (x_N - x_M)\mathbf{\imath} + (y_N - y_M)\mathbf{\jmath}$.

Equation (8.23) is projected on the $\mathbf{\imath}$ and $\mathbf{\jmath}$ directions to find the velocity components of the point N:

$$\begin{aligned} \dot{x}_N &= \dot{x}_M - \omega(y_N - y_M), \\ \dot{y}_N &= \dot{y}_M + \omega(x_N - x_M). \end{aligned} \tag{8.24}$$

The following vectorial equation between the accelerations $\mathbf{a}_N$ and $\mathbf{a}_M$ of the points N and M can be written as

$$\mathbf{a}_N = \mathbf{a}_M + \boldsymbol{\alpha} \times \mathbf{r}_{MN} - \omega^2 \mathbf{r}_{MN}, \tag{8.25}$$

where $\mathbf{a}_N = \ddot{x}_N\mathbf{\imath} + \ddot{y}_N\mathbf{\jmath}$, $\mathbf{a}_M = \ddot{x}_M\mathbf{\imath} + \ddot{y}_M\mathbf{\jmath}$, and $\boldsymbol{\alpha} = \alpha\mathbf{k}$.

The acceleration components of the point N are obtained from Eq. (8.25):

$$\begin{aligned} \ddot{x}_N &= \ddot{x}_M - \alpha(y_N - y_M) - \omega^2(x_N - x_M), \\ \ddot{y}_N &= \ddot{y}_M + \alpha(x_N - x_M) - \omega^2(y_N - y_M). \end{aligned} \tag{8.26}$$

The *Mathematica*^{TM} function for the absolute velocity and acceleration analysis is

```
AbsVelAcc::usage = "AbsVelAcc[xM,yM,xN,yN,vMx,vMy,
aMx,aMy, dtheta,ddtheta] computes the absolute velocity
and acceleration vectors."
Begin["'Private'"]
AbsVelAcc[xM_,yM_,xN_,yN_,vMx_,vMy_,aMx_,aMy_,
dtheta_,ddtheta_]:=
Block[
{ vNx, vNy, aNx, aNy },
vNx = vMx - dtheta (yN-yM) ;
vNy = vMy + dtheta (xN-xM) ;
aNx = aMx - ddtheta (yN-yM) - dtheta^2 (xN-xM) ;
aNy = aMy + ddtheta (xN-xM) - dtheta^2 (yN-yM) ;
Return[ { vNx, vNy, aNx, aNy } ] ; ]
End[ ]
```

The functions **VelAccRRR, VelAccRRT, AngVelAcc**, and **AbsVelAcc** are included in the *Mathematica*^{TM} package **VelAcc**. The name of the source file for the package is **VelAcc.m** and is given in Program 8.7.

R-RRR-RRT Mechanism

The position analysis of the planar R-RRR-RRT mechanism considered [see Fig. 8.4(a)] is presented in Subsection 8.2. Given the angular velocity $\omega = \dot{\phi} = 3.14$ rad/s, calculate the velocities and the accelerations of the joints and the angular velocities and the accelerations of the links.

The *Mathematica*^{TM} packages **Driver**, **PosVec**, and **VelAcc** are loaded in the main program using the commands

```
<<Driver.m ;
<<PosVec.m ;
<<VelAcc.m ;
```

The angular velocity of the driver link is zero:

$$\alpha = \ddot{\phi} = 0.$$

Velocity and acceleration of the joint A

Since the joint A is the origin of the reference frame $xAyz$, the velocity and acceleration of the joint A are

$$\mathbf{v}_A = \mathbf{a}_A = \mathbf{0}.$$

Velocity and acceleration of the joint B

The velocity and acceleration components of the joint B are

$$\begin{aligned}
v_{Bx} &= -AB\omega \sin\phi = -1.110 \text{ m/s},\\
v_{By} &= \ AB\omega \cos\phi = 1.110 \text{ m/s},\\
a_{Bx} &= -AB\omega^2 \cos\phi - AB\alpha \sin\phi = -11.631 \text{ m/s}^2,\\
a_{By} &= -AB\omega^2 \sin\phi + AB\alpha \cos\phi = -11.631 \text{ m/s}^2.
\end{aligned}$$

The numerical values for the velocity and acceleration components of the joint B are obtained using the *Mathematica*^{TM} function **Driver**:

```
vBx = Driver[xA,yA,AB,phi,omega,alpha][[3]] ;
vBy = Driver[xA,yA,AB,phi,omega,alpha][[4]] ;
aBx = Driver[xA,yA,AB,phi,omega,alpha][[5]] ;
aBy = Driver[xA,yA,AB,phi,omega,alpha][[6]] ;
```

Velocity and acceleration of the joint D

The velocity and acceleration of the joint D are

$$\mathbf{v}_D = \mathbf{a}_D = \mathbf{0}.$$

Velocity and acceleration of the joint C

To calculate the velocity components v_{Cx} and v_{Cy} of the joint C, the following equations are used:

$$\begin{aligned}
&(x_B - x_C)(v_{Bx} - v_{Cx}) + (y_B - y_C)(v_{By} - v_{Cy}) = 0,\\
&(x_D - x_C)(v_{Dx} - v_{Cx}) + (y_D - y_C)(v_{Dy} - v_{Cy}) = 0. \qquad (8.27)
\end{aligned}$$

The acceleration components a_{Cx} and a_{Cy} of the joint C are calculated from the equations

$$(x_B - x_C)(a_{Bx} - a_{Cx}) + (v_{Bx} - v_{Cx})^2 + (y_B - y_C)(a_{By} - a_{Cy})$$

$$+(v_{By}-v_{Cy})^2=0,$$
$$(x_D-x_C)(a_{Dx}-a_{Cx})+(v_{Dx}-v_{Cx})^2+(y_D-y_C)(a_{Dy}-a_{Cy})$$
$$+(v_{Dy}-v_{Cy})^2=0. \tag{8.28}$$

The numerical solutions for the velocity and acceleration components of the joint C using the *Mathematica*^{TM} function **VelAccRRR** are

```
   vCx=VelAccRRR[xB,yB,xD,yD,xC,yC,vBx,vBy,vDx,vDy,
aBx,aBy,aDx,aDy][[1]];
   vCy=VelAccRRR[xB,yB,xD,yD,xC,yC,vBx,vBy,vDx,vDy,
aBx,aBy,aDx,aDy][[2]];
   aCx=VelAccRRR[xB,yB,xD,yD,xC,yC,vBx,vBy,vDx,vDy,
aBx,aBy,aDx,aDy][[3]];
   aCy=VelAccRRR[xB,yB,xD,yD,xC,yC,vBx,vBy,vDx,vDy,
aBx,aBy,aDx,aDy][[4]];
```

The input data are the coordinates of the joints B, D, and C, and the velocities and acceleration components of the joints B and D. The output data are the four elements of the vector returned by the function **VelAccRRR**, which are the velocity and acceleration components of the joint C.

Velocity and acceleration of the joint E

The numerical values for the angular velocity and acceleration of the link 3 using the *Mathematica*^{TM} function **AngVelAcc** are

```
   omega3=AngVelAcc[xC,yC,xD,yD,vCx,vCy,vDx,vDy,aCx,aCy,
aDx,aDy,phi3][[1]];
   alpha3=AngVelAcc[xC,yC,xD,yD,vCx,vCy,vDx,vDy,aCx,aCy,
aDx,aDy,phi3][[2]];
```

The input data are the coordinates, velocities, and accelerations of the joints C and D, and the angle ϕ_3. The output data are the two components of the vector returned by the function **AngVelAcc**.

The velocity and the acceleration of the joint E are calculated with

$$\mathbf{v}_E=\boldsymbol{\omega}_3\times\mathbf{r}_{DE} \quad \text{and} \quad \mathbf{a}_E=\boldsymbol{\alpha}_3\times\mathbf{r}_{DE}-\omega^2\mathbf{r}_{DE}.$$

The numerical solutions for the velocity and acceleration components of the joint E are obtained using the *Mathematica*^{TM} function **AbsVelAcc**:

```
vEx=AbsVelAcc[xD,yD,xE,yE,vDx,vDy,aDx,aDy,omega3,
alpha3][[1]];
vEy=AbsVelAcc[xD,yD,xE,yE,vDx,vDy,aDx,aDy,omega3,
alpha3][[2]];
aEx=AbsVelAcc[xD,yD,xE,yE,vDx,vDy,aDx,aDy,omega3,
alpha3][[3]];
aEy=AbsVelAcc[xD,yD,xE,yE,vDx,vDy,aDx,aDy,omega3,
alpha3][[4]];
```

The input data are the coordinates of the joints D and E, and the velocity and acceleration components of the joint D. The output data are the four elements of the vector returned by the function **AbsVelAcc**, which are the velocity and acceleration components of the joint E.

Velocity and acceleration of the joint F

In this particular case, the angular velocity and acceleration of the link 5 are zero:

$$\boldsymbol{\omega}_5 = \boldsymbol{\alpha}_5 = \mathbf{0}.$$

The velocity and the acceleration of the point $P(-L_c, 0)$, on the sliding direction, are zero:

$$\mathbf{v}_P = \mathbf{a}_P = \mathbf{0}.$$

The velocity and acceleration components of the joint F are calculated using the *Mathematica*^{TM} function **VelAccRRT**:

```
vFx=VelAccRRT[xE,yE,xP,yP,xF,yF,vEx,vEy,vPx,vPy,
aEx,aEy,aPx,aPy,phi5,omega5,alpha5][[1]];
vFy=VelAccRRT[xE,yE,xP,yP,xF,yF,vEx,vEy,vPx,vPy,
aEx,aEy,aPx,aPy,phi5,omega5,alpha5][[2]];
aFx=VelAccRRT[xE,yE,xP,yP,xF,yF,vEx,vEy,vPx,vPy,
aEx,aEy,aPx,aPy,phi5,omega5,alpha5][[3]];
aFy=VelAccRRT[xE,yE,xP,yP,xF,yF,vEx,vEy,vPx,vPy,
aEx,aEy,aPx,aPy, phi5,omega5,alpha5][[4]];
```

The input data are the coordinates of the joints E, P, and F, the velocities and acceleration components of the joints E and P, the angle ϕ_5, the angular velocity ω_5, and the angular acceleration α_5. The output data are the four elements of the vector returned by the function **VelAccRRT**, which are the velocity and acceleration components of the joint F. The numerical values for the angular velocity ω_2 and the angular acceleration α_2 of the link 2 using the *Mathematica*^{TM} function **AngVelAcc** are

```
omega2=AngVelAcc[xB,yB,xC,yC,vBx,vBy,vCx,vCy,aBx,aBy,
aCx,aCy,phi2][[1]];
alpha2=AngVelAcc[xB,yB,xC,yC,vBx,vBy,vCx,vCy,aBx,aBy,
aCx,aCy,phi2][[2]];
```

The input data are the coordinates, velocities, and accelerations of the joints B and C, and the angle ϕ_2. The numerical values for the angular velocity ω_4 and the angular acceleration α_4 of the link 4 are calculated with the *Mathematica*^{TM} function **AngVelAcc**

```
omega4=AngVelAcc[xE,yE,xF,yF,vEx,vEy,vFx,vFy,aEx,aEy,
aFx,aFy,phi4][[1]];
alpha4=AngVelAcc[xE,yE,xF,yF,vEx,vEy,vFx,vFy,aEx,aEy,
aFx,aFy,phi4][[2]];
```

The input data are the coordinates, velocities, and accelerations of the joints E and F, and the angle ϕ_4. The output data are the two components of the vector returned by the function **AngVelAcc**.

The *Mathematica*^{TM} program and the numerical results are shown in Program 8.8.

R-RRT Mechanism

The position analysis of the planar R-RRT mechanism considered (Fig. 8.5), has been presented in Subsection 8.2. Given the angular velocity $\omega = 3.141$ rad/s, calculate the velocities and accelerations of the joints and the angular velocities and accelerations of the links.

The *Mathematica*^{TM} packages **Driver**, **Position**, and **VelAcc** are loaded in the main program using the commands

```
<<Driver.m ;
<<Position.m ;
```

```
<<VelAcc.m ;
```

The angular velocity of the driver link is zero:

$$\alpha = \ddot{\phi} = 0.$$

Velocity and acceleration of the joint A

Since the joint A is the origin of the reference frame $xAyz$, the velocity and acceleration of the joint A are

$$\mathbf{v}_A = \mathbf{a}_A = \mathbf{0}.$$

The velocity and the acceleration of the point C are zero:

$$\mathbf{v}_C = \mathbf{a}_C = \mathbf{0}.$$

Velocity and acceleration of the joint B

In order to calculate the velocity and acceleration of the point B, you need to calculate the velocity and acceleration of the point P, located on the driver link AB:

$$\begin{aligned}
v_{Px} &= -AP \sin\phi\omega = -1.110 \text{ m/s},\\
v_{Py} &= \ AP \cos\phi\omega = 1.110 \text{ m/s},\\
a_{Px} &= -AP\omega^2 \cos\phi - AP\alpha \sin\phi = -3.489 \text{ m/s}^2,\\
a_{Py} &= -AP\omega^2 \sin\phi + AP\alpha \cos\phi = -3.489 \text{ m/s}^2.
\end{aligned}$$

The numerical values for the velocity and acceleration components of the point P using the *Mathematica*^{TM} function **Driver** are

```
vPx = Driver[xA,yA,AP,phi,omega,alpha][[3]] ;
vPy = Driver[xA,yA,AP,phi,omega,alpha][[4]] ;
aPx = Driver[xA,yA,AP,phi,omega,alpha][[5]] ;
aPy = Driver[xA,yA,AP,phi,omega,alpha][[6]] ;
```

The velocity components v_{Bx} and v_{By} of the point B are calculated using the equations

$$\begin{aligned}
&(x_C - x_B)(v_{Cx} - v_{Bx}) + (y_C - y_B)(v_{Cy} - v_{By}) = 0,\\
&(v_{Bx} - v_{Px}) \sin\phi + \omega(x_B - x_P) \cos\phi - (v_{By} - v_{Py}) \cos\theta +\\
&\omega(y_B - y_P) \sin\phi = 0. \qquad (8.29)
\end{aligned}$$

The acceleration components a_{Bx} and a_{By} of the point B are calculated using the equations

$$(x_C - x_B)(a_{Cx} - a_{Bx}) + (v_{Cx} - v_{Bx})^2 + (y_C - y_B)(a_{Cy} - a_{By})$$
$$+(v_{Cy} - v_{By})^2 = 0,$$
$$(a_{Bx} - a_{Px}) \sin\phi + \omega(v_{Bx} - v_{Px}) \cos\phi - (a_{By} - a_{Py}) \cos\phi$$
$$+\omega(v_{By} - v_{Py}) \sin\phi + [(x_B - x_P) \cos\phi + (y_B - y_P) \sin\phi]\alpha$$
$$+((v_{Bx} - v_{Px}) \cos\phi - \omega(x_B - x_P) \sin\phi + (v_{By} - v_{Py}) \sin\phi +$$
$$\omega(y_B - y_P) \cos\phi)\omega = 0. \quad (8.30)$$

The solutions for the velocity and acceleration components of the joint F are obtained using the *Mathematica*^{TM} function **VelAccRRT**

```
  vBx=VelAccRRT[xC,yC,xP,yP,xB,yB,vCx,vCy,vPx,vPy,
aCx,aCy,aPx,aPy,phi,omega,alpha][[1]];
  vBy=VelAccRRT[xC,yC,xP,yP,xB,yB,vCx,vCy,vPx,vPy,
aCx,aCy,aPx,aPy,phi,omega,alpha][[2]];
  aBx=VelAccRRT[xC,yC,xP,yP,xB,yB,vCx,vCy,vPx,vPy,
aCx,aCy,aPx,aPy,phi,omega,alpha][[3]];
  aBy=VelAccRRT[xC,yC,xP,yP,xB,yB,vCx,vCy,vPx,vPy,
aCx,aCy,aPx,aPy,phi,omega,alpha][[4]];
```

The input data are the coordinates of the points C, P, and B, the velocities and acceleration components of the points C and P, the angle ϕ, the angular velocity ω, and the angular acceleration α. The output data are the four elements of the vector returned by the function **VelAccRRT**, which are the velocity and acceleration components of the point B.

The *Mathematica*^{TM} program and the numerical results are shown in Program 8.9.

R-RTR-RTR Mechanism

The planar R-RTR-RTR mechanism considered is shown in Fig. 8.6. Given the input data $AB = 0.14$ m, $AC = 0.06$ m, $AE = 0.25$ m, $CD = 0.15$ m, the angle of the driver link AB and the horizontal axis $\phi = 30°$, and the angular velocity $\omega = \dot{\phi} = 5.235$ rad/s, calculate the velocities and accelerations of the joints and the angular velocities and accelerations of the links. The cartesian reference frame $xOyz$ is chosen, $A \equiv O$.

The *Mathematica*^{TM} packages **Driver** and **VelAcc** are loaded in the main program using the commands

```
<<Driver.m ;
<<VelAcc.m ;
```

Position analysis

Position of the joint A

Since the joint A is the origin of the reference frame $xAyz$, the coordinates of the joint A are zero:

$$x_A = y_A = 0.$$

Position of the joint B

The coordinates of the joint B are

$$\begin{aligned} x_B &= x_A + AB\cos\phi = 0.121 \text{ m}, \\ y_B &= y_A + AB\sin\phi = 0.070 \text{ m}. \end{aligned} \tag{8.31}$$

The coordinates of the joint B are obtained using the *Mathematica*^{TM} function **Driver**:

```
xB = Driver[xA,yA,AB,phi,omega,alpha][[1]] ;
yB = Driver[xA,xB,AB,phi,omega,alpha][[2]] ;
```

Position of the joint C

The coordinates of the joint C are

$$x_C = 0 \text{ m}, \; y_C = AC = 0.06 \text{ m}.$$

The angle ϕ_3 between the link 3 and the horizontal axis Ox is

$$\phi_3 = \arctan\frac{y_B - y_C}{x_B - x_C}.$$

Position of the joint D

The coordinates of the joint D are

$$\begin{aligned} x_D &= x_C - CD\cos\phi_3 = -0.149 \text{ m}, \\ y_D &= y_C - CD\sin\phi_3 = 0.047 \text{ m}. \end{aligned} \tag{8.32}$$

Position of the joint E

The coordinates of the joint E are

$$x_E = 0 \text{ m}, \; y_E = -AE = -0.25 \text{ m}.$$

The angle ϕ_5 between the link 3 and the horizontal axis Ox is

$$\phi_5 = \pi + \arctan \frac{y_D - y_E}{x_D - x_E}.$$

Velocity and acceleration analysis

The angular velocity of the driver link is zero:

$$\alpha = \ddot{\phi} = 0.$$

Velocity and acceleration of the joint A

Since the joint A is the origin of the reference frame $xAyz$, the velocity and acceleration of the joint A are zero:

$$\mathbf{v}_A = \mathbf{a}_A = \mathbf{0}.$$

Velocity and acceleration of the joint B

The velocity and acceleration components of the joint B are:

$$\begin{aligned}
v_{Bx} &= -AB\omega \sin\phi = -0.366 \text{ m/s}, \\
v_{By} &= \;\; AB\omega \cos\phi = 0.634 \text{ m/s}, \\
a_{Bx} &= -AB\omega^2 \cos\phi - AB\alpha \sin\phi = -3.323 \text{ m/s}^2, \\
a_{By} &= -AB\omega^2 \sin\phi + AB\alpha \cos\phi = -1.919 \text{ m/s}^2.
\end{aligned}$$

The velocity and acceleration components of the joint B using the *Mathematica*^{TM} function **`Driver`** are

```
vBx = Driver[xA,yA,AB,phi,omega,alpha][[3]] ;
vBy = Driver[xA,yA,AB,phi,omega,alpha][[4]] ;
aBx = Driver[xA,yA,AB,phi,omega,alpha][[5]] ;
aBy = Driver[xA,yA,AB,phi,omega,alpha][[6]] ;
```

Velocity and acceleration of the joint C

The velocity and acceleration of the joint C are zero

$$\mathbf{v}_C = \mathbf{a}_C = \mathbf{0}.$$

Velocity and acceleration of the joint D

The angular velocity ω_3 and angular acceleration α_3 of the link 3 are obtained using the *Mathematica*^{TM} function **AngVelAcc**:

```
omega3=AngVelAcc[xB,yB,xC,yC,vBx,vBy,vCx,vCy,aBx,aBy,
aCx,aCy,phi3][[1]];
alpha3=AngVelAcc[xB,yB,xC,yC,vBx,vBy,vCx,vCy,aBx,aBy,
aCx,aCy,phi3][[2]];
```

The input data are the coordinates, velocities, and accelerations of the joints B and C, and the angle ϕ_3. The output data are the two components of the vector returned by the function **AngVelAcc**. To obtain the numerical values for the velocity and acceleration components of the joint D, the *Mathematica*^{TM} function **AbsVelAcc** is used:

```
vDx=AbsVelAcc[xC,yC,xD,yD,vCx,vCy,aCx,aCy,
omega3,alpha3][[1]];
vDy=AbsVelAcc[xC,yC,xD,yD,vCx,vCy,aCx,aCy,
omega3,alpha3][[2]];
aDx=AbsVelAcc[xC,yC,xD,yD,vCx,vCy,aCx,aCy,
omega3,alpha3][[3]];
aDy=AbsVelAcc[xC,yC,xD,yD,vCx,vCy,aCx,aCy,
omega3,alpha3][[4]];
```

The input data are the coordinates of the joints C and D, the velocity and acceleration components of the joint C, and the angular velocity and acceleration of the link 3. The output data are the four components of the vector returned by the function **AbsVelAcc**.

Velocity and acceleration of the joint E

The velocity and acceleration of the joint E are zero

$$\mathbf{v}_E = \mathbf{a}_E = \mathbf{0}.$$

The numerical values for the angular velocity ω_5 and angular acceleration α_5 of the link 5 using the *Mathematica*^{TM} function **AngVelAcc** are given by the commands

```
omega5=AngVelAcc[xD,yD,xE,yE,vDx,vDy,vEx,vEy,
aDx,aDy,aEx,aEy,phi5][[1]];
alpha5=AngVelAcc[xD,yD,xE,yE,vDx,vDy,vEx,vEy,
aDx,aDy,aEx,aEy,phi5][[2]];
```

The input data are the coordinates, velocities, and accelerations of the joints D and E, and the angle ϕ_5. The output data are the two components of the vector returned by the function **AngVelAcc**. The *Mathematica*^{TM} program and the numerical results are shown in Program 8.10.

8.4 Force Analysis

Force and Moment

A rigid link is shown in Fig. 8.7. The input data are the mass m, the acceleration vector of the center of mass $\mathbf{a}_{CM}$, the mass moment of inertia I_{CM}, and the angular acceleration $\boldsymbol{\alpha}$ of the link. The output data are the total force $\mathbf{F}$ and the moment of inertia $\mathbf{M}$ of the link.

The total force $\mathbf{F}$ is

$$\mathbf{F} = \mathbf{F}_{in} + \mathbf{G},$$

where $\mathbf{F}_{in} = -m\,\mathbf{a}_{CM}$ is the inertia force, $\mathbf{G} = -m\,\mathbf{g}$ is the gravitational force, and $\mathbf{g} = -9.807\mathbf{k}$ m/s^2 is the gravitational acceleration.

The moment of inertia M of the link is

$$\mathbf{M} = -I_{CM}\boldsymbol{\alpha},$$

where $\boldsymbol{\alpha} = \alpha\mathbf{k}$.

The *Mathematica*^{TM} function **ForceMomentum** for the force analysis is

```
ForceMomentum::usage="ForceMomentum[m,aCM,ICM,ddtheta]
computes
the total force and moment of a rigid link."
Begin["‘Private‘"]
ForceMomentum[m_,aCM_,ICM_,ddtheta_]:=
Block[
```

```
g, Fin, G, F, M ,
g = 9.807 ;
Fin = - m aCM ;
G = { 0, -m g, 0 } ;
F = Fin + G ;
M = - ICM { 0, 0, ddtheta } ;
Return[ { F, M } ] ; ]
End[ ]
```

Joint Force Computation

RRR dyad

Figure 8.8 shows an RRR dyad with two links 2 and 3, and three pin joints M, N, and P. The input data are the total forces $\mathbf{F}_2,\ \mathbf{F}_3$ and the moments $\mathbf{M}_2,\ \mathbf{M}_3$ on the links 2 and 3, the position vectors $\mathbf{r}_M,\ \mathbf{r}_N,\ \mathbf{r}_P$ of the joints M, N, P, and position vectors $\mathbf{r}_{C2}, \mathbf{r}_{C3}$ of the centers of mass of the links 2 and 3. The output data are the joint reaction forces $\mathbf{F}_{12},\ \mathbf{F}_{43}$, and $\mathbf{F}_{32}$.

The unknown joint reaction forces are

$$\begin{aligned} \mathbf{F}_{12} &= F_{12x}\imath + F_{12y}\jmath, \\ \mathbf{F}_{43} &= F_{43x}\imath + F_{43y}\jmath, \\ \mathbf{F}_{23} &= -\mathbf{F}_{32} = F_{23x}\imath + F_{23y}\jmath. \end{aligned} \tag{8.33}$$

To determine $\mathbf{F}_{12}$ and $\mathbf{F}_{43}$, the following equations are written:

• sum of all forces on links 2 and 3 is zero

$$\sum \mathbf{F}^{(2\&3)} = \mathbf{F}_{12} + \mathbf{F}_2 + \mathbf{F}_3 + \mathbf{F}_{43} = \mathbf{0},$$

or

$$\begin{aligned} \sum \mathbf{F}^{(2\&3)} \cdot \imath &= F_{12x} + F_{2x} + F_{3x} + F_{43x} = 0, \\ \sum \mathbf{F}^{(2\&3)} \cdot \jmath &= F_{12y} + F_{2y} + F_{3y} + F_{43y} = 0. \end{aligned} \tag{8.34}$$

• sum of moments of all forces and moments on link 2 about P is zero

$$\sum \mathbf{M}_P^{(2)} = (\mathbf{r}_M - \mathbf{r}_P) \times \mathbf{F}_{12} + (\mathbf{r}_{C2} - \mathbf{r}_P) \times \mathbf{F}_2 + \mathbf{M}_2 = \mathbf{0}. \tag{8.35}$$

• sum of moments of all forces and moments on link 3 about P is zero

$$\sum \mathbf{M}_P^{(3)} = (\mathbf{r}_N - \mathbf{r}_P) \times \mathbf{F}_{43} + (\mathbf{r}_{C3} - \mathbf{r}_P) \times \mathbf{F}_3 + \mathbf{M}_3 = \mathbf{0}. \tag{8.36}$$

The components F_{12x}, F_{12y}, F_{43x}, and F_{43y} are calculated from Eqs. (8.34), (8.35), and (8.36).
The reaction force $\mathbf{F}_{32} = -\mathbf{F}_{23}$ is computed from the sum of all forces on the link 2:

$$\sum \mathbf{F}^{(2)} = \mathbf{F}_{12} + \mathbf{F}_2 + \mathbf{F}_{32} = \mathbf{0},$$

or

$$\sum \mathbf{F}^{(2)} \cdot \imath = F_{12x} + F_{2x} + F_{32x} = 0,$$
$$\sum \mathbf{F}^{(2)} \cdot \jmath = F_{12y} + F_{2y} + F_{32y} = 0. \quad (8.37)$$

The *Mathematica*^{TM} function **ForceRRR** for the RRR dyad joint force analysis is

```
ForceRRR::usage = "ForceRRR[F2,M2,F3,M3,rM,rN,rP,rC2,rC3]
computes the joint reactions for the RRR dyad."
Begin["‘Private‘"]
ForceRRR[F2_,M2_,F3_,M3_,rM_,rN_,rP_,rC2_,rC3_]:=
Block[
{ F12, F12Sol, F12xSol, F12ySol, F43, F43Sol, F43xSol,
F43ySol, rPC2, rPC3, rPM, rPN, F32, eqRRR1, eqRRR2, eqRRR3,
eqRRR4, solRRR},
F12Sol = { F12xSol, F12ySol, 0 } ;
F43Sol = { F43xSol, F43ySol, 0 } ;
rPC2 = rC2 - rP ;
rPC3 = rC3 - rP ;
rPM = rM - rP ;
rPN = rN - rP ;
eqRRR1=(F12Sol+F43Sol+F2+F3)[[1]]==0;
eqRRR2=(F12Sol+F43Sol+F2+F3)[[2]]==0;
eqRRR3=(Cross[rPC2,F2]+Cross[rPM,F12Sol]+M2)[[3]]==0;
eqRRR4=(Cross[rPC3,F3]+Cross[rPN,F43Sol]+M3)[[3]]==0;
solRRR = Solve[ {eqRRR1, eqRRR2, eqRRR3, eqRRR4},
{F12xSol,F12ySol,F43xSol,F43ySol} ] ;
F12 = F12Sol/.solRRR[[1]] ;
F43 = F43Sol/.solRRR[[1]] ;
F32 = - F2 - F12 ;
```

```
Return[ { F12, F43, F23 } ] ;
]
End[ ]
```

RRT dyad

Figure 8.9 shows an RRT dyad with two links 2 and 3, two pin joints M and P and one slider joint P. The input data are the total forces $\mathbf{F}_2$, $\mathbf{F}_3$ and moments $\mathbf{M}_2$, $\mathbf{M}_3$ on the links 2 and 3, the position vectors $\mathbf{r}_M$, $\mathbf{r}_N$, $\mathbf{r}_P$ of the joints M, N, P, and the position vector of the center of mass $\mathbf{r}_{C2}$ of the link 2. The output data are the joint reaction forces $\mathbf{F}_{12}$, $\mathbf{F}_{43}$, $\mathbf{F}_{23} = -\mathbf{F}_{32}$ and the position vector $\mathbf{r}_Q$ of the application point of the joint reaction force $\mathbf{F}_{43}$.

The joint reaction $\mathbf{F}_{43}$ is perpendicular to the sliding direction $\mathbf{r}_{PN}$ or

$$\mathbf{F}_{43} \cdot \mathbf{r}_{PN} = (F_{43x}\mathbf{\imath} + F_{43y}\mathbf{\jmath}) \cdot [(x_N - x_P)\mathbf{\imath} + (y_N - y_P)\mathbf{\jmath}] = 0. \quad (8.38)$$

In order to determine $\mathbf{F}_{12}$ and $\mathbf{F}_{43}$, the following equations are written:

- sum of all the forces on links 2 and 3 is zero

$$\sum \mathbf{F}^{(2\&3)} = \mathbf{F}_{12} + \mathbf{F}_2 + \mathbf{F}_3 + \mathbf{F}_{43} = \mathbf{0},$$

or

$$\sum \mathbf{F}^{(2\&3)} \cdot \mathbf{\imath} = F_{12x} + F_{2x} + F_{3x} + F_{43x} = 0,$$
$$\sum \mathbf{F}^{(2\&3)} \cdot \mathbf{\jmath} = F_{12y} + F_{2y} + F_{3y} + F_{43y} = 0. \quad (8.39)$$

- sum of moments of all forces and moments on link 2 about P is zero

$$\sum \mathbf{M}_P^{(2)} = (\mathbf{r}_M - \mathbf{r}_P) \times \mathbf{F}_{12} + (\mathbf{r}_{C2} - \mathbf{r}_P) \times \mathbf{F}_2 + \mathbf{M}_2 = \mathbf{0}. \quad (8.40)$$

The components F_{12x}, F_{12y}, F_{43x}, and F_{43y} are calculated from Eqs. (8.38), (8.39), and (8.40).

The reaction force components F_{32x} and F_{32y} are computed from the sum of all the forces on the link 2:

$$\sum \mathbf{F}^{(2)} = \mathbf{F}_{12} + \mathbf{F}_2 + \mathbf{F}_{32} = \mathbf{0},$$

or

$$\sum \mathbf{F}^{(2)} \cdot \mathbf{\imath} = F_{12x} + F_{2x} + F_{32x} = 0,$$
$$\sum \mathbf{F}^{(2)} \cdot \mathbf{\jmath} = F_{12y} + F_{2y} + F_{32y} = 0. \quad (8.41)$$

To determine the application point $Q(x_Q, y_Q)$ of the reaction force $\mathbf{F}_{43}$, one can write sum of moments of all the forces and the moments on the link 3 about $C_3 \equiv P$ is zero:

$$\sum \mathbf{M}_P^{(3)} = (\mathbf{r}_Q - \mathbf{r}_P) \times \mathbf{F}_{43} + \mathbf{M}_3 = \mathbf{0}. \tag{8.42}$$

If $\mathbf{M}_3 = \mathbf{0}$, then P is identical to Q ($P \equiv Q$).

If $\mathbf{M}_3 \neq \mathbf{0}$, an equation regarding the location of the point Q on the sliding direction $\mathbf{r}_{NP}$ is written as

$$\frac{y_Q - y_P}{x_Q - x_P} = \frac{y_N - y_P}{x_N - x_P}. \tag{8.43}$$

From Eqs. (8.42) and (8.43) the coordinates x_Q and y_Q of the point Q are calculated.

The *Mathematica*^{TM} function **ForceRRT** for the RRT dyad joint force analysis is

```
ForceRRT::usage = "ForceRRT[F2,M2,F3,M3,rM,rN,rP,rC2]
computes the joint reactions for the RRR dyad."
Begin["‘Private‘"]
ForceRRT[F2_,M2_,F3_,M3_,rM_,rN_,rP_,rC2_]:=
Block[
{ F12, F12Sol, F12xSol, F12ySol, F43, F43Sol, F43xSol,
F43ySol, F32, eqRRT1, eqRRT2, eqRRT3, eqRRT4, solRRT, rNC2,
rNM, rNP, rNQ, rQP, rQ, rQSol, xQSol, yQSol, eqRRTQ1, eqRRTQ2,
solRRTQ },
F12Sol = { F12xSol, F12ySol, 0 } ;
F43Sol = { F43xSol, F43ySol, 0 } ;
rPC2 = rC2 - rP ;
rPM = rM - rP ;
rPN = rN - rP ;
eqRRT1 = (F12Sol+F43Sol+F2+F3)[[1]] == 0 ;
eqRRT2 = (F12Sol+F43Sol+F2+F3)[[2]] == 0 ;
eqRRT3 = (F43Sol.rPN) == 0 ;
eqRRT4 = (Cross[rNC2,F2]+Cross[rNM,F12Sol]+M2)[[3]]==0;
solRRT = Solve[ {eqRRT1, eqRRT2, eqRRT3, eqRRT4},
{F12xSol,F12ySol,F43xSol,F43ySol} ] ;
F12 = F12Sol/.solRRT[[1]] ;
```

```
F43 = F43Sol/.solRRT[[1]] ;
F32 = - F2 - F12 ;
F23 = - F32 ;
If[ M3[[3]]==0 , rQ = rP ,
rQSol = { xQSol, yQSol, 0 } ;
rQP = rP - rQSol ;
eqRRTQ1 = rPQ[[2]]/rPQ[[1]] - rPN[[2]]/rPN[[1]]==0;
eqRRTQ2 = Cross[rPQ,F43] + M3 == 0 ;
solRRTQ = Solve[ eqRRTQ1, eqRRTQ2 , xQSol,yQSol ] ;
rQ = rQSol/.solRRTQ[[1]] ; ] ;
Return[ { F12, F43, F23, rQ } ] ;
]
End[ ]
```

RTR dyad

Figure 8.10 shows an RTR dyad with two links 2 and 3, and one pin joint M, one slider joint P, and one pin joint P. The input data are the total forces $\mathbf{F}_2$, $\mathbf{F}_3$ and moments $\mathbf{M}_2$, $\mathbf{M}_3$ of the links 2 and 3, the position vectors $\mathbf{r}_M$, $\mathbf{r}_N$, $\mathbf{r}_P$ of the joints M, N, and the position vector of the center of mass $\mathbf{r}_{C2}$ of the link 2. The output data are the joint reaction forces $\mathbf{F}_{12}$, $\mathbf{F}_{43}$, $\mathbf{F}_{23} = -\mathbf{F}_{32}$ and the position vector $\mathbf{r}_Q$ of the application point of the joint reaction force $\mathbf{F}_{23}$.

The unknown joint reaction forces $\mathbf{F}_{12}$ and $\mathbf{F}_{43}$ are calculated from the relations:

• sum of all the forces on links 2 and 3 is zero

$$\sum \mathbf{F}^{(2\&3)} = \mathbf{F}_{12} + \mathbf{F}_2 + \mathbf{F}_3 + \mathbf{F}_{43} = \mathbf{0},$$

or

$$\sum \mathbf{F}^{(2\&3)} \cdot \imath = F_{12x} + F_{2x} + F_{3x} + F_{43x} = 0,$$
$$\sum \mathbf{F}^{(2\&3)} \cdot \jmath = F_{12y} + F_{2y} + F_{3y} + F_{43y} = 0. \quad (8.44)$$

• sum of the moments of all forces and moments on links 2 and 3 about M is zero

$$\sum \mathbf{M}_M^{(2\&3)} = (\mathbf{r}_P - \mathbf{r}_M) \times (\mathbf{F}_3 + \mathbf{F}_{43}) +$$
$$(\mathbf{r}_{C2} - \mathbf{r}_M) \times \mathbf{F}_2 + \mathbf{M}_2 + \mathbf{M}_3 = \mathbf{0}. \quad (8.45)$$

• sum of all the forces on link 2 projected onto the sliding direction $\mathbf{r}_{MP}$ is zero

$$\mathbf{F}_2 \cdot \mathbf{r}_{MP} = (F_{43x}\mathbf{\imath} + F_{43y}\mathbf{\jmath}) \cdot [(x_P - x_M)\mathbf{\imath} + (y_P - y_M)\mathbf{\jmath}] = 0. \qquad (8.46)$$

The components F_{12x}, F_{12y}, F_{43x}, and F_{43y} are calculated from Eqs. (8.44), (8.45), and (8.46).

The force components F_{32x} and F_{32y} are computed from the sum of all the forces on link 2:

$$\sum \mathbf{F}^{(2)} = \mathbf{F}_{12} + \mathbf{F}_2 + \mathbf{F}_{32} = \mathbf{0},$$

or

$$\sum \mathbf{F}^{(2)} \cdot \mathbf{\imath} = F_{12x} + F_{2x} + F_{32x} = 0,$$
$$\sum \mathbf{F}^{(2)} \cdot \mathbf{\jmath} = F_{12y} + F_{2y} + F_{32y} = 0. \qquad (8.47)$$

To determine the application point $Q(x_Q, y_Q)$ of the reaction force $\mathbf{F}_{23}$, one can write sum of moments of all the forces and the moments on the link 3 about $C_3 \equiv P$ is zero:

$$\sum \mathbf{M}_P^{(3)} = (\mathbf{r}_Q - \mathbf{r}_P) \times \mathbf{F}_{23} + \mathbf{M}_3 = \mathbf{0}. \qquad (8.48)$$

If $\mathbf{M}_3 = \mathbf{0}$, then P is identical to Q ($P \equiv Q$).

If $\mathbf{M}_3 \neq \mathbf{0}$, an equation regarding the location of the point Q on the sliding direction $\mathbf{r}_{NP}$ is written as

$$\frac{y_Q - y_M}{x_Q - x_M} = \frac{y_M - y_P}{x_M - x_P}. \qquad (8.49)$$

From Eqs. (8.48) and (8.49) the coordinates x_Q and y_Q of the point Q are calculated.

The *Mathematica*^{TM} function **ForceRTR** for the RTR dyad joint force analysis is

```
   ForceRTR::usage = "ForceRTR[F2,M2,F3,M3,rM,rP,rC2]
computes the joint reactions for the RTR dyad."
   Begin["‘Private‘"]
   ForceRTR[F2_,M2_,F3_,M3_,rM_,rP_,rC2_]:=
   Block[
```

```
  { rC3, F12, F12Sol, F12xSol, F12ySol, F43, F43Sol, F43xSol,
F43ySol, F32, eqRTR1, eqRTR2, eqRTR3, eqRTR4, solRTR, rMC2,
rMP, rPQ, rQ, rQSol, xQSol, yQSol, eqRTRQ1, eqRTRQ2,
solRTRQ },
  F12Sol = { F12xSol, F12ySol, 0 } ;
  F43Sol = { F43xSol, F43ySol, 0 } ;
  rMC2 = rC2 - rM ;
  rMP = rP - rM ;
  eqRTR1=(F12Sol+F43Sol+F2+F3)[[1]]==0;
  eqRTR2=(F12Sol+F43Sol+F2+F3)[[2]]==0;
  eqRTR3=(F12Sol+F2).rMP==0;
  eqRTR4=(Cross[rMC2,F2]+Cross[rMP,(F3+F43Sol)]+M2+M3)[[3]]==0;
  solRTR = Solve[{ eqRTR1, eqRTR2, eqRTR3, eqRTR4 } ,
{ F12xSol, F12ySol, F43xSol, F43ySol } ];
  F12 = F12Sol/.solRTR[[1]];
  F43 = F43Sol/.solRTR[[1]];
  F32 = - F2 - F12 ;
  F23 = - F32 ;
  If[ M3[[3]]==0 , rQ = rP ,
  rQSol = { xQSol, yQSol, 0 };
  rMQ = rQSol - rM ;
  rPQ = rQSol - rP ;
  eqRRTQ1 = rMQ[[2]]/rMQ[[1]] - rMP[[2]]/rMP[[1]] ==0;
  eqRRTQ2 = Cross[rPQ,F23] + M3 ==0;
  solRTRQ = Solve[{eqRTRQ1, eqRTRQ2}, {xQSol,yQSol}];
  rQ = rQSol/.solRTRQ[[1]];
  ];
  Return[ { F12, F43, F23, rQ } ];
  ]
  End[ ]
```

Driver link

A driver link mechanism is shown in Fig. 8.11. The input data are the total force $\mathbf{F}_1$ and moment $\mathbf{M}_1$ on the driver link, the joint reaction force $\mathbf{F}_{21}$, the position vectors $\mathbf{r}_A$, $\mathbf{r}_B$ of the joints A, B, and the position vector of the center of mass $\mathbf{r}_{C1}$ of the driver link. The output data are the joint reaction force $\mathbf{F}_{01}$, and the moment of the motor $\mathbf{M}_m$ (equilibrium moment).

A force equation for the driver link is written to determine the joint reaction $\mathbf{F}_{01}$:

$$\sum \mathbf{F}^{(1)} = \mathbf{F}_{01} + \mathbf{F}_1 + \mathbf{F}_{21} = \mathbf{0},$$

or

$$\sum \mathbf{F}^{(1)} \cdot \imath = F_{01x} + F_{1x} + F_{21x} = 0,$$
$$\sum \mathbf{F}^{(1)} \cdot \jmath = F_{01y} + F_{1y} + F_{21y} = 0, \tag{8.50}$$

The sum of the moments about A_R for link 1 gives the equilibrium moment $\mathbf{M}_m$:

$$\sum \mathbf{M}_A^{(1)} = (\mathbf{r}_B - \mathbf{r}_A) \times \mathbf{F}_{21} + (\mathbf{r}_{C1} - \mathbf{r}_A) \times \mathbf{F}_1 + \mathbf{M}_m = \mathbf{0}. \tag{8.51}$$

The *Mathematica*^{TM} function **FMDriver** for the driver link joint force analysis is

```
FMDriver::usage = "FMDriver[F1,M1,F21,rA,rB,rC1]
computes the joint reaction and torque of the motor."
Begin["‘Private‘"]
FMDriver[F1_,M1_,F21_,rA_,rB_,rC1_]:=
Block[
{ F01, rAC1, rAB, Mm },
F01 = - F1 - F21 ;
rAC1 = rC1 - rA ;
rAB = rB - rA ;
Mm = - Cross[rAC1,F1] - Cross[rAB,F21] - M1 ;
Return[ { F01, Mm } ] ;
]
End[ ]
```

The functions **ForceMomentum, ForceRRR, ForceRRT, ForceRTR**, and **FMDriver** are included in the *Mathematica*^{TM} package **Force**. The name of the source file for the package is **Force.m** and is given in Program 8.11.

R-RRT Mechanism

The position, velocity, and acceleration analysis of the planar R-RRT mechanism considered are presented in Subsections 8.2 and 8.3. Given the external moment $\mathbf{M}_{ext} = -100\,\mathrm{sign}(\omega_3)\,\mathbf{k}$ N·m, applied on the link 3, calculate the motor moment $\mathbf{M}_m$ required for the dynamic equilibrium of the mechanism [Fig. 8.12(a)]. All three links are rectangular prisms with the depth $d = 0.001$ m and the mass density $\rho = 8000$ Kg/m^3. The height of the links 1 and 3 is $h = 0.01$ m. The link 2 has the height $h_S = 0.02$ m, and the width $w_S = 0.05$ m. The center of mass location of the links $i = 1, 2, 3$ are designated by $C_i(x_{Ci}, y_{Ci}, 0)$.

The *Mathematica*^{TM} packages **`Driver`**, **`Position`**, **`VelAcc`**, and **`Force`** are loaded in the main program using the commands

```
<<Driver.m ;
<<Position.m ;
<<VelAcc.m ;
<<Force.m ;
```

Force and moment analysis

Link 1

The mass m_1, the acceleration of the center of mass $\mathbf{a}_{C1}$, and the mass moment of inertia I_{C1} of the link 1 are

$$m_1 = \rho\ AP\ h\ d,$$
$$\mathbf{a}_{C1} = (\mathbf{a}_A + \mathbf{a}_P)/2,$$
$$I_{C1} = m_1(AP^2 + h^2)/12.$$

The total force F_1 and moment M_1 of the link 1 are calculated using the *Mathematica*^{TM} function **`ForceMomentum`**:

```
F1=ForceMomentum[m1,aC1,IC1,alpha][[1]];
M1=ForceMomentum[m1,aC1,IC1,alpha][[2]];
```

The input data are the mass **`m1`**, the acceleration vector of the center of mass **`aC1`**, the mass moment of inertia **`IC1`**, and the angular acceleration **`alpha`** of the link 1. The output data are the two elements of the vector returned by the function **`ForceMomentum`**, which are the total force **`F1`** and moment of inertia **M1** of the link 1.

Link 2

The mass m_2, the acceleration of the center of mass $\mathbf{a}_{C2}$, and the mass moment of inertia I_{C2} of the link 2 are

$$m_2 = \rho\ w_S\ h_S\ d,$$
$$\mathbf{a}_{C2} = \mathbf{a}_B,$$
$$I_{C2} = m_2(w_S^2 + h_S^2)/12.$$

The total force F_2 and moment M_2 of the link 2 are computed using the *Mathematica*^{TM} function **ForceMomentum**:

```
F2=ForceMomentum[m2,aC2,IC2,alpha][[1]];
M1=ForceMomentum[m2,aC2,IC2,alpha][[2]];
```

The input data are the mass **m2**, the acceleration vector of the center of mass **aC2**, the mass moment of inertia **IC2**, and the angular acceleration **alpha2** of the link 2. The output data are the two elements of the vector returned by the function **ForceMomentum**, which are the total force **F2** and moment of inertia **M2** of the link 2.

Link 3

The mass m_3, the acceleration of the center of mass $\mathbf{a}_{C3}$, and the mass moment of inertia I_{C3} of the link 3 are

$$m_3 = \rho\ BC\ h\ d,$$
$$\mathbf{a}_{C3} = (\mathbf{a}_B + \mathbf{a}_C)/2,$$
$$I_{C3} = m_3(AB^2 + h^2)/12.$$

The total force F_3 and moment M_3 of the link 3 are calculated using the *Mathematica*^{TM} function **ForceMomentum**:

```
F3=ForceMoment[m3,aC3,IC3,alpha3][[1]];
M3=ForceMoment[m3,aC3,IC3,alpha3][[2]];
```

The input data are the mass **m3**, the acceleration vector of the center of mass **aC3**, the mass moment of inertia **IC3**, and the angular acceleration

alpha3 of the link 3. The output data are the two elements of the vector returned by the function **ForceMomentum**, which are the total force **F3** and moment of inertia **M3** of the link 3.

Joint reactions

The joint reactions for the dyad RRT (CBB) [Fig. 8.12(b)] are computed using the *Mathematica*^{TM} function **ForceRRT**:

```
F03=ForceRRT[F3,M3+Mext,F2,M2,rC,rB,rA,rC3][[1]];
F12=ForceRRT[F3,M3+Mext,F2,M2,rC,rB,rA,rC3][[2]];
F23=ForceRRT[F3,M3+Mext,F2,M2,rC,rB,rA,rC3][[3]];
```

The input data are the total force **F3** and moment **M3+Mext** of the link 3, the total force **F2** and moment **M2** of the link 2, the position vectors **rC, rB, rA, rC3** of the joints C, B, A, and the center of mass C_3 of the link 3. The output data are the three elements of the vector returned by the function **ForceMomentum**, which are the joint reactions **F03, F12**, and **F23**.

The position vector of the application point Q of the joint reaction **F23** can be also computed using the *Mathematica*^{TM} function **ForceRRT**:

```
rQ=ForceRRT[F3,M3+Mext,F2,M2,rC,rB,rA,rC3][[4]];
```

The joint reaction and the moment of the motor [Fig. 8.11(c)] are calculated using the *Mathematica*^{TM} function **FMDriver**

```
F01=FMDriver[F1,M1,F21,rA,rB,rC1][[1]];
Mm=FMDriver[F1,M1,F21,rA,rB,rC1][[2]];
```

The input data are the total force **F1** and moment **M1** of the link 1, the joint reaction **F21=-F12**, the position vectors **rA, rB, rC1** of the joints A, B, and the center of mass C_1 of the link 1. The output data are the two elements of the vector returned by the function **FMDriver**, which are the joint reaction **F01** and moment **Mm** of the motor.

The *Mathematica*^{TM} program and the numerical results are shown in Program 8.12.

R-RTR-RTR Mechanism

The position, velocity and acceleration analysis of the planar R-RTR-RTR mechanism considered (see Figs 8.6 and 8.13) are presented in Subsection 8.3. Given the external moment $\mathbf{M}_{ext} = -100\,\text{sign}(\omega_5)\,\mathbf{k}$ N·m, applied on the link 5, calculate the motor moment $\mathbf{M}_m$ required for the dynamic equilibrium of the mechanism. All five links are rectangular prisms with the depth $d = 0.001$ m and the mass density $\rho = 8000$ Kg/m^3. The heights of the links 1, 3, and 5 are $h = 0.01$ m. The link 2 has the height $h_S = 0.02$ m, and the width $w_S = 0.05$ m. The center of mass location of the links $i = 1,\ 2,\ 3, ..., 5$ are designated by $C_i(x_{Ci}, y_{Ci}, 0)$.

The *Mathematica*^{TM} packages **Driver, VelAcc**, and **Force** are loaded in the main program using the commands

```
<<Driver.m ;
<<VelAcc.m ;
<<Force.m ;
```

Force and moment analysis

Link 1

The mass m_1, the acceleration of the center of mass $\mathbf{a}_{C1}$, and the mass moment of inertia I_{C1} of the link 1 are

$$m_1 = \rho\ AB\ h\ d,$$
$$\mathbf{a}_{C1} = (\mathbf{a}_A + \mathbf{a}_B)/2,$$
$$I_{C1} = m_1(AB^2 + h^2)/12,$$

The total force F_1 and moment M_1 of the link 1 are computed using the *Mathematica*^{TM} function **ForceMomentum**:

```
F1=ForceMomentum[m1,aC1,IC1,alpha][[1]];
M1=ForceMomentum[m1,aC1,IC1,alpha][[2]];
```

The input data are the mass **m1**, the acceleration vector of the center of mass **aC1**, the mass moment of inertia **IC1**, and the angular acceleration **alpha** of the link 1. The output data are the two elements of the vector returned by the function **ForceMomentum**, which are the total force **F1** and moment of inertia **M1** of the link 1.

Link 2

The mass m_2, the acceleration of the center of mass $\mathbf{a}_{C2}$, and the mass moment of inertia I_{C2} of the link 2 are

$$m_2 = \rho\, w_S\, h_S\, d,$$
$$\mathbf{a}_{C2} = \mathbf{a}_B,$$
$$I_{C2} = m_2(w_S^2 + h_S^2)/12.$$

The total force F_2 and moment M_2 of the link 2 are calculated using the *Mathematica*^{TM} function **ForceMomentum**:

```
F2=ForceMomentum[m2,aC2,IC2,alpha2][[1]];
M1=ForceMomentum[m2,aC2,IC2,alpha2][[2]];
```

The input data are the mass **m2**, the acceleration vector of the center of mass **aC2**, the mass moment of inertia **IC2**, and the angular acceleration **alpha2** of the link 2. The output data are the two elements of the vector returned by the function **ForceMomentum**, which are the total force **F2** and moment of inertia **M2** of the link 2.

Link 3

The coordinates of the center of mass C_3 of the link 3 are

$$x_{C3} = x_C + (DF/2 - CD)\cos\phi_3,$$
$$y_{C3} = y_C + (DF/2 - CD)\sin\phi_3.$$

The acceleration components of the center of mass of the link 3 are compted using the *Mathematica*^{TM} function **AbsVelAcc**:

```
aC3x=AbsVelAcc[xC,yC,xC3,yC3,vCx,vCy,aCx,aCy,
omega3,alpha3][[3]];
aC3y=AbsVelAcc[xC,yC,xC3,yC3,vCx,vCy,aCx,aCy,
omega3,alpha3][[4]];
```

The mass m_3 and the mass moment of inertia I_{C3} of the link 3 are

$$m_3 = \rho\; DF\; h\; d,$$
$$I_{C3} = m_3(DF^2 + h^2)/12.$$

The total force F_3 and moment M_3 of the link 3 using the *Mathematica*^{TM} function **ForceMomentum** are

```
F3=ForceMoment[m3,aC3,IC3,alpha3][[1]];
M3=ForceMoment[m3,aC3,IC3,alpha3][[2]];
```

The input data are the mass **m3**, the acceleration vector of the center of mass **aC3**, the mass moment of inertia **IC3**, and the angular acceleration **alpha3** of the link 3. The output data are the two elements of the vector returned by the function **ForceMomentum**, which are the total force **F3** and moment of inertia **M3** of the link 3.

Link 4

The mass m_4, the acceleration of the center of mass $\mathbf{a}_{C4}$, and the mass moment of inertia I_{C4} of the link 4 are

$$m_4 = \rho \; w_S \; h_S \; d,$$
$$\mathbf{a}_{C4} = \mathbf{a}_D,$$
$$I_{C4} = m_4(w_S^2 + h_S^2)/12.$$

The total force F_4 and moment M_4 of the link 4 using the *Mathematica*^{TM} function **ForceMomentum** are

```
F4=ForceMomentum[m4,aC4,IC4,alpha4][[1]];
M4=ForceMomentum[m4,aC4,IC4,alpha4][[2]];
```

The input data are the mass **m4**, the acceleration vector of the center of mass **aC4**, the mass moment of inertia **IC4**, and the angular acceleration **alpha4** of the link 4. The output data are the two elements of the vector returned by the function **ForceMomentum**, which are the total force **F4** and moment of inertia **M4** of the link 4.

Link 5

The coordinates of the center of mass C_5 of the link 5 are

$$x_{C5} = x_E + EG/2 \cos \phi_5,$$
$$y_{C5} = y_E + EG/2 \sin \phi_5.$$

The acceleration components of the center of mass of the link 3 using the *Mathematica*^{TM} function **AbsVelAcc** are

```
aC5x=AbsVelAcc[xE,yE,xC5,yC5,vEx,vEy,aEx,aEy,
omega5,alpha5][[3]];
aC5y=AbsVelAcc[xE,yE,xC5,yC5,vEx,vEy,aEx,aEy,
omega5,alpha5][[4]];
```

The mass m_5 and the mass moment of inertia I_{C5} of the link 5 are

$$m_5 = \rho \; EG \; h \; d,$$
$$I_{C5} = m_5(EG^2 + h^2)/12.$$

The total force F_5 and moment M_5 of the link 5 using the *Mathematica*^{TM} function **ForceMomentum** are

```
F5=ForceMomentum[m5,aC5,IC5,alpha5][[1]];
M5=ForceMomentum[m5,aC5,IC5,alpha5][[2]];
```

The input data are the mass **m5**, the acceleration vector of the center of mass **aC5**, the mass moment of inertia **IC5**, and the angular acceleration **alpha5** of the link 5. The output data are the two elements of the vector returned by the function **ForceMomentum**, which are the total force **F5** and moment of inertia **M5** of the link 5.

Joint reactions

The joint reactions for the dyad RTR (EDD) [Fig. 8.14(a)] are computed using the *Mathematica*^{TM} function **ForceRTR**:

```
F05=ForceRTR[F5,M5+Mext,F4,M4,rE,rD,rC5][[1]];
F34=ForceRTR[F5,M5+Mext,F4,M4,rE,rD,rC5][[2]];
F54=ForceRTR[F5,M5+Mext,F4,M4,rE,rD,rC5][[3]];
```

The input data are the total force **F5** and moment **M5+Mext** of the link 5, the total force **F4** and momentum **M4** of the link 4, the position vectors **rE, rD, rC5** of the joints E, D, and the center of mass C_5 of the link 5. The output data are the three elements of the vector returned by the function **ForceMomentum**, which are the joint reactions **F05, F34**, and **F45**.

The position vector of the application point P of the joint reaction **F45** can be also computed using the *Mathematica*^{TM} function **ForceRTR**

```
rP=ForceRTR[F5,M5+Mext,F4,M4,rE,rD,rC5][[4]];
```

Next, consider the dyad RTR (CBB), shown in Fig. 8.14(b). The reaction force $\mathbf{F}_{43}$ acting at point D can be moved to a parallel position at point C_3 by adding the corresponding couple

$$\mathbf{M}_{43} = \mathbf{r}_{C_3D} \times \mathbf{F}_{43}.$$

The joint reactions for the dyad RTR (CBB) using the *Mathematica*^{TM} function **ForceRTR** are

```
F03=ForceRTR[F3+F43,M3+M43,F2,M2,rC,rB,rC3][[1]];
F12=ForceRTR[F3+F43,M3+M43,F2,M2,rC,rB,rC3][[2]];
F32=ForceRTR[F3+F43,M3+M43,F2,M2,rC,rB,rC3][[3]];
```

The input data are the force **F3+F43** and moment **M3+M43** of the link 3, the total force **F2** and moment **M2** of the link 2, the position vectors **rC, rB, rC3** of the joints C, B, and the center of mass C_3 of the link 3. The output data are the three elements of the vector returned by the function **ForceMomentum**, which are the joint reactions **F03, F12**, and **F23**.

The position vector of the application point Q of the joint reaction **F12** can be also computed using the *Mathematica*^{TM} function **ForceRTR**:

```
rQ=ForceRTR[F3+F43,M3+M43,F2,M2,rC,rB,rC3][[4]];
```

The joint reaction and the moment of the motor [Fig. 8.14(c)] are computed using the *Mathematica*^{TM} function **FMDriver**:

```
F01=FMDriver[F1,M1,F21,rA,rB,rC1][[1]];
Mm=FMDriver[F1,M1,F21,rA,rB,rC1][[2]];
```

The input data are the total force **F1** and moment **M1** of the link 1, the joint reaction **F21=-F12**, the position vectors **rA, rB, rC1** of the joints A, B, and center of mass C_1 of the link 1. The output data are the two elements of the vector returned by the function **FMDriver**, which are the joint reaction **F01** and moment **Mm** of the motor.

The *Mathematica*^{TM} program and the numerical results are shown in Program 8.13.

Remark: All the packages must be placed in the *Mathematica*^{TM} folder for PC. For Macintosh OS X the packages must be placed in Home, Library, Mathematica, Applications.

8.5 Problems

8.1 Referring to Problem 3.5 (Fig. 3.20), write a *Mathematica*^{TM} program using packages for the position analysis of the mechanism.

8.2 Referring to Problem 3.7 (Fig. 3.22), write a *Mathematica*^{TM} program using packages for the position analysis of the mechanism.

8.3 Referring to Problem 3.8 (Fig. 3.23), write a *Mathematica*^{TM} program using packages for the position analysis of the mechanism.

8.4 Referring to Problem 3.13 (Fig. 3.28), write a *Mathematica*^{TM} program using packages for the position analysis of the mechanism.

8.5 Referring to Problem 3.14 (Fig. 3.29), write a *Mathematica*^{TM} program using packages for the position analysis of the mechanism.

8.6 Referring to Problem 4.1 (Fig. 3.16), write a *Mathematica*^{TM} program using packages for the velocity and acceleration analysis of the mechanism.

8.7 Referring to Problem 4.3 (Fig. 4.10), write a *Mathematica*^{TM} program using packages for the velocity and acceleration analysis of the mechanism.

8.8 Referring to Problem 6.4 (Fig. 3.19), write a *Mathematica*^{TM} program using packages for the equilibrium moment and the joint forces of the mechanism.

8.9 Referring to Problem 6.16 (Fig. 3.31), write a *Mathematica*^{TM} program using packages for the equilibrium moment and the joint forces of the mechanism.

8.10 Referring to Problem 6.18 (Fig. 3.33), write a *Mathematica*^{TM} program using packages for the equilibrium moment and the joint forces of the mechanism.

8.6 Programs

References

[1] P. Antonescu, *Mechanisms*, Printech, Bucharest, 2003.

[2] P. Appell, *Traité de Mécanique Rationnelle*, Gautier-Villars, Paris, 1941.

[3] I.I. Artobolevski, *Mechanisms in Modern Engineering Design*, MIR, Moscow, 1977.

[4] M. Atanasiu, *Mecanica*, EDP, Bucharest, 1973.

[5] H. Baruh, *Analytical Dynamics*, WCB/McGraw-Hill, Boston, 1999.

[6] A. Bedford and W. Fowler, *Dynamics*, Addison Wesley, Menlo Park, 1999.

[7] A. Bedford and W. Fowler, *Statics*, Addison Wesley, Menlo Park, 1999.

[8] M.I. Buculei, *Mechanisms*, University of Craiova Press, Craiova, 1976.

[9] M.I. Buculei, D. Bagnaru, G. Nanu, D.B. Marghitu, *Analysis of Mechanisms with Bars*, Scrisul romanesc, Craiova, 1986.

[10] T. Demian et al., *Mechanisms - problems*, EDP, Bucharest, 1969.

[11] A.G. Erdman, and G.N. Sandor, *Mechanisms Design*, Prentice-Hall, Upper Saddle River, 1984.

[12] A. Ertas and J.C. Jones, *The Engineering Design Process*, John Wiley & Sons, New York, 1996.

[13] F. Freudenstein, "An Application of Boolean Algebra to the Motion of Epicyclic Drives," *Transaction of the ASME, Journal of Engineering for Industry*, pp.176-182, 1971.

[14] J.H. Ginsberg, *Advanced Engineering Dynamics*, Cambridge University Press, Cambridge, 1995.

[15] D.T. Greenwood, *Principles of Dynamics*, Prentice-Hall, Englewood Cliffs, 1998.

[16] A.S. Hall, Jr., A.R. Holowenko, and H.G. Laughlin, *Theory and problems of machine design*, McGraw-Hill, New York, 1961.

[17] R.C. Hibbeler, *Engineering Mechanics - Statics and Dynamics*, Prentice-Hall, Upper Saddle River, New Jersey, 1995.

[18] R.C. Juvinall and K.M. Marshek, *Fundamentals of Machine Component Design*, John Wiley & Sons, New York, 1983.

[19] T.R. Kane, *Analytical Elements of Mechanics*, Vol. 1, Academic Press, New York, 1959.

[20] T.R. Kane, *Analytical Elements of Mechanics*, Vol. 2, Academic Press, New York, 1961.

[21] T.R. Kane and D.A. Levinson, "The Use of Kane's Dynamical Equtions in Robotics", *MIT International Journal of Robotics Research*, No. 3, pp. 3-21, 1983.

[22] T.R. Kane, P.W. Likins, and D.A. Levinson, *Spacecraft Dynamics*, McGraw-Hill, New York, 1983.

[23] T.R. Kane and D.A. Levinson, *Dynamics*, McGraw-Hill, New York, 1985.

[24] J.T. Kimbrell, *Kinematics Analysis and Synthesis*, McGraw-Hill, New York, 1991.

[25] R. Maeder, *Programming in Mathematica*, Addison–Wesley Publishing Company, Redwood City, California, 1990.

[26] N.H. Madsen, *Statics and Dynamics*, class notes, www.eng.auburn.edu/users/nmadsen/, 2004.

[27] N.I. Manolescu, F. Kovacs, and A. Oranescu, *The Theory of Mechanisms and Machines*, EDP, Bucharest, 1972.

[28] D.B. Marghitu, *Mechanical Engineer's Handbook*, Academic Press, San Diego, California, 2001.

[29] D.B. Marghitu and M.J. Crocker, *Analytical Elements of Mechanisms*, Cambridge University Press, Cambridge, 2001.

[30] D.B. Marghitu and E.D. Stoenescu, *Kinematics and Dynamics of Machines and Machine Design*, class notes, www.eng.auburn.edu/users/marghitu/, 2004.

[31] J.L. Meriam and L.G. Kraige, *Engineering Mechanics: Dynamics*, John Wiley & Sons, New York, 1997.

[32] D.J. McGill and W.W. King, *Engineering Mechanics: Statics and an Introduction to Dynamics*, PWS Publishing Company, Boston, 1995.

[33] R.L. Mott, *Machine elements in mechanical design*, Prentice Hall, Upper Saddle River, New Jersey, 1999.

[34] D.H. Myszka, *Machines and Mechanisms*, Prentice-Hall, Upper Saddle River, New Jersey, 1999.

[35] R.L. Norton, *Machine Design*, Prentice-Hall, Upper Saddle River, New Jersey, 1996.

[36] R.L. Norton, *Design of Machinery*, McGraw-Hill, New York, 1999.

[37] W.C. Orthwein, *Machine Component Design*, West Publishing Company, St. Paul, 1990.

[38] L.A. Pars, *A treatise on analytical dynamics*, Wiley, New York, 1965.

[39] R.M. Pehan, *Dynamics of Machinery*, McGraw-Hill, New York, 1967.

[40] I. Popescu, *Mechanisms*, University of Craiova Press, Craiova, 1990.

[41] I. Popescu and C. Ungureanu, *Structural Synthesis and Kinematics of Mechanisms with Bars*, Universitaria Press, Craiova, 2000.

[42] I. Popescu and D.B. Marghitu, "Dyad Classification for Mechanisms," *World Conference on Integrated Design and Process Technology*, Austin, Texas, December 3-5, 2003.

[43] I. Popescu, E.D. Stoenescu, and D.B. Marghitu, "Analysis of Spatial Kinematic Chains Using the System Groups," *8th International Congress on Sound and Vibration*, St. Petersburg, Russia, July 5-8, 2004.

[44] M. Radoi and E. Deciu, *Mecanica*, EDP, Bucharest, 1981.

[45] F. Reuleaux, *The Kinematics of Machinery*, Dover, New York, 1963.

[46] C.A. Rubin, *The Student Edition of Working Model*, Addison–Wesley Publishing Company, Reading, Massachusetts, 1995.

[47] I.H. Shames, *Engineering Mechanics - Statics and Dynamics*, Prentice-Hall, Upper Saddle River, New Jersey, 1997.

[48] J.E. Shigley and C.R. Mischke, *Mechanical Engineering Design*, McGraw-Hill, New York, 1989.

[49] J.E. Shigley and J.J. Uicker, *Theory of Machines and Mechanisms*, McGraw-Hill, New York, 1995.

[50] R.W. Soutas-Little and D.J. Inman, *Engineering Mechanics: Statics and Dynamics*, Prentice-Hall, Upper Saddle River, New Jersey, 1999.

[51] A. Stan and M. Grumarescu, *Mechanics Problems*, EDP, Bucharest, 1973.

[52] A. Stoenescu, A. Ripianu, and M. Atanasiu, *Theoretical Mechanics Problems*, EDP, Bucharest, 1965.

[53] A. Stoenescu and G. Silas, *Theoretical Mechanics*, ET, Bucharest, 1957.

[54] E.D. Stoenescu, *Dynamics and Synthesis of Kinematic Chains with Impact and Clearance*, Ph.D. Dissertation, Mechanical Engineering, Auburn University, 2005.

[55] L. W. Tsai, *Mechanism Design: Enumeration of Kinematic Structures According to Function*, CRC Press, Boca Raton, Florida, 2001.

[56] R. Voinea, D. Voiculescu, and V. Ceausu, *Mecanica*, EDP, Bucharest, 1983.

[57] K.J. Waldron and G.L. Kinzel, *Kinematics, Dynamics, and Design of Machinery*, John Wiley&Sons, New York, 1999.

[58] C.E. Wilson and J.P. Sadler, *Kinematics and Dynamics of Machinery*, Harper Collins College Publishers, 1991.

[59] C.W. Wilson, *Computer integrated macine design*, Prentice Hall, Inc., Upper Saddle River, New Jersey, 1997.

[60] S. Wolfram, *Mathematica*, Wolfram Media/Cambridge University Press, Cambridge, 1999.

[61] * * * ,*The theory of mechanisms and machines (Teoria mehanizmov i masin)*, Vassaia scola, Minsc, 1970.

[62] * * * ,*Working Model 2D, Users Manual*, Knowledge Revolution, San Mateo, California, 1996.

Figure captions

Figure 8.1. Driver link.
Figure 8.2. (a) RRR dyad and (b) RRT dyad.
Figure 8.3. (a) R-RTR-RRT mechanism, (b) RTR (BBD) dyad, and (c) RRT (CEE) dyad.
Figure 8.4. (a) R-RRR-RRT mechanism, (b) RRR (BCD) dyad, and (c) RRT (CEE) dyad.
Figure 8.5. R-RRT mechanism.
Figure 8.6. R-RTR-RTR mechanism.
Figure 8.7. Rigid link.
Figure 8.8 RRR dyad
Figure 8.9. RRT dyad.
Figure 8.10. RTR dyad.
Figure 8.11. Driver link.
Figure 8.12. (a) R-RRT mechanism, (b) RRT (CBB) dyad, and (c) driver link.
Figure 8.13. R-RTR-RTR mechanism: forces and moments.
Figure 8.14. Joint reactions for: (a) RTR (EDD) dyad, (b) RTR (CBB) dyad, and (c) driver link.