Contents

2 Position Analysis **1**

2 Position Analysis

2.1 Absolute Cartesian Method

The position analysis of a kinematic chain requires the determination of the joint positions and/or the position of the center of gravity (CG) of the link. A planar link with the end nodes A and B is considered in Fig. 2.1(a). Let $(x_A,\, y_A)$ be the coordinates of the joint A with respect to the reference frame xOy, and $(x_B,\, y_B)$ be the coordinates of the joint B with the same reference frame. Using Pythagoras the following relation can be written

$$(x_B - x_A)^2 + (y_B - y_A)^2 = AB^2 = L_{AB}^2, \tag{2.1}$$

where L_{AB} is the length of the link AB.
Let ϕ be the angle of the link AB with the horizontal axis Ox. Then, the slope m of the link AB is defined as

$$m = \tan\phi = \frac{y_B - y_A}{x_B - x_A}. \tag{2.2}$$

Let n be the intercept of AB with the vertical axis Oy. Using the slope m and the intercept n, the equation of the straight link, in the plane, is

$$y = m\,x + n, \tag{2.3}$$

where x and y are the coordinates of any point on this link.

For a link with a translational joint, Fig. 2.1(b), the sliding direction (Δ) is given by the equation

$$x\,\cos\alpha - y\,\sin\alpha - p = 0, \tag{2.4}$$

where p is the distance from the origin O to the sliding line (Δ). The position function for the joint $A(x_A,\, y_A)$ is

$$x_A\,\cos\alpha - y_A\,\sin\alpha - p = \pm\,d, \tag{2.5}$$

where d is the distance from A to the sliding line. The relation between the joint A and a point B on the sliding direction, $B \in (\Delta)$ (the symbol $\in$ means ”belongs to”), is

$$(x_A - x_B)\,\sin\beta - (y_A - y_B)\,\cos\beta = \pm\,d, \tag{2.6}$$

where $\beta = \alpha + \dfrac{\pi}{2}$.

If $A\,x + B\,y + C = 0$ is the linear equation of the line (Δ) then the distance d is, Fig. 2.1(b)

$$d = \frac{|A\,x_A + B\,y_A + C|}{\sqrt{A^2 + B^2}} \ . \tag{2.7}$$

2.2 Slider-Crank (R-RRT) Mechanism

The R-RRT (slider-crank) mechanism shown in Fig. 2.2(a) has the dimensions: $AB = 0.1$ m and $BC = 1$ m. The driver link 1 makes an angle $\phi = \phi_1 = 45°$ with the horizontal axis. Find the positions of the joints and the angles of the links with the horizontal axis.

Solution

The MATLAB program starts with the statements

```
clear % clears all variables from the workspace
clc % clears the command window and homes the cursor
close all  % closes all the open figure windows
```

The MATLAB commands for the input data are

```
AB=0.5;  BC=1.;
```

The angle of the driver link 1 with the horizontal axis $\phi = 45°$. The MATLAB command for the input angle is

```
phi=pi/4;
```

where `pi` has a numerical value approximately equal to 3.14159.

Position of joint A

A Cartesian reference frame xOy is selected. The joint A is in the origin of the reference frame, that is, $A \equiv O$,

$$x_A = 0,\ y_A = 0,$$

or in MATLAB

```
xA=0; yB=0;
```

Position of joint B

The unknowns are the coordinates of the joint B, x_B and y_B. Because the joint A is fixed and the angle ϕ is known, the coordinates of the joint B are computed from the following expressions

$$\begin{aligned} x_B &= AB\cos\phi = (0.5)\ \cos 45^\circ = 0.353553\,\text{m}, \\ y_B &= AB\sin\phi = (0.5)\ \sin 45^\circ = 0.353553\,\text{m}. \end{aligned} \tag{2.8}$$

The MATLAB commands for Eq. (2.20) are

```
xB=AB*cos(phi);
yB=AB*Sin(phi);
```

where `phi` is the angle ϕ in radians.

Position of joint C

The unknowns are the coordinates of the joint C, x_C and y_C. The joint C is located on the horizontal axis $y_C = 0$ and with MATLAB

```
yC=0;
```

The length of the segment BC is constant

$$(x_B - x_C)^2 + (y_B - y_C)^2 = BC^2, \tag{2.9}$$

or

$$(0.353553 - x_C)^2 + (0.353553 - 0)^2 = 1^2.$$

Equation (2.9) with MATLAB command is

```
eqnC='(xB-xCsol)^2+(yB-yC)^2=BC^2';
```

where `xCsol` is the unknown.

To solve the equation, a specific MATLAB command will be used.

The command

```
solve('eqn1','eqn2',...,'eqnN','var1','var2',...'varN')
```

attempts to solve an equation or set of equations `'eqn1','eqn2',...,'eqnN'` for the variables `'eqnN','var1','var2',...'varN'`. The set of equations are symbolic expressions or strings specifying equations.
For the R-RRT mechanism

```
solC=solve(eqnC,'xCsol');
```

Because it is a quadratic equation two solutions are found for the position of C. The two solutions are given in a vector form: `solC` is a vector with two components `solC(1)` and `solC(2)`. To obtain the numerical solutions the `eval` command has to be used

```
xC1=eval(solC(1));
xC2=eval(solC(2));
```

The command `eval(s)`, where `s` is a string, executes the string as an expression or statement.

These two solutions for x_C are located at the intersection of the horizontal axis $0x$ with the circle centered in B and radius CB, as shown in Fig. 2.2(b), and they have the following numerical values:

$$x_{C_1} = 1.2890 \text{ m} \quad \text{and} \quad x_{C_2} = -0.5819 \text{ m}.$$

To determine the correct position of the joint C for the mechanism, an additional condition is needed. For the first quadrant, $0 \leq \phi \leq 90°$, the condition is $x_C > xB$.
This condition with MATLAB is

```
if xC1 > xB xC = xC1; else xC = xC2; end
```

The general form of the `if` statement is
`if` *expression statements* `else` *statements* `end`
The x-coordinate of the joint C ise $x_C = x_{C_1} = 1.2890\,\text{m}$.

The angle of the link 2 (link BC) with the horizontal is

$$\phi_2 = \arctan\frac{y_B - y_C}{x_B - x_C},$$

or in MATLAB

```
phi2 = atan((yB-yC)/(xB-xC));
```

The statement `atan(s)` is the arctangent of the elements of `s`.
The numerical solutions for B, C, and ϕ_2 are printed using the statements

```
fprintf('xB = %g (m) \n', xB);
fprintf('yB = %g (m) \n', yB);
fprintf('xC = %g (m) \n', xC);
fprintf('yC = %g (m) \n', yC);
fprintf('phi2 = %g (degrees) \n', phi2*180/pi);
```

The statement `fprintf(f,format,s)` writes data in the real part of array `s` to the file `f`. The data is formated under control of the specified `format` string.

The results of the program are displayed as

```
xB = 0.353553 (m)
yB = 0.353553 (m)
xC = 1.06821 (m)
yC = 1.28897 (m)
phi2 = -20.7048 (degrees)
>>
```

The mechanism is plotted with the help of the command `plotf`. The statement `plotf(x,y,c)` plots vector `x` versus vector `x`, and `c` is a character string.
For the R-RRT mechanism two straight lines AB and BC are plotted

```
plot( [xA,xB],[yA,yB],'r-o', [xB,xC],[yB,yC],'b-o' ),...
```

The line AB is a red (`r` red), solid line (`-` solid), with a circle (`o` circle) at each data point and the line BC is a blue (`b` blue), solid line with a circle at each data point and the. The graphic of the mechanism obtained with MATLAB is shown in Fig. 2.3.

The x-axis and y-axis are labeled using the commands

```
xlabel('x (m)'),...
ylabel('y (m)'),...
```

and a title is added

```
title('positions for \phi = 45 (deg)'),...
```

On the figure the joints A, B, and C are identified with the statements

```
text(xA,yA,' A'),...
text(xB,yB,' B'),...
text(xC,yC,' C'),...
axis([-0.2 1.4 -0.2 1.4]),...
grid
```

The commas and ellipses (...) after the command are used to execute the commands together. Otherwise, the data will be plotted, then the labels will be added and the data replotted, and so on.

The statement `axis([xMIN xMAX yMIN yMAX])` sets scaling for the x and y axes on the current plot. To improve the graph a background grid was added with the command `grid`.

The MATLAB program for the positions is given in Program 2.1.

2.3 R-RRR-RRT Mechanism

The considered planar R-RRR-RRT mechanism is shown in Fig. 2.4. The driver link is the rigid link 1 (the element AB). The following data are given: AB=0.150 m, BC=0.400 m, CD=0.370 m, CE=0.230 m, EF=CE, L_a=0.300 m, L_b=0.450 m, and L_c=CD. The angle of the driver link 1 with the horizontal axis is $\phi = \phi_1 = 45°$.
Find the positions of the joints and the angles of the links with the horizontal axis.

Solution
Position of joint A

A cartesian reference frame $xOyz$ with the versors $[\mathbf{\imath}, \mathbf{\jmath}, \mathbf{k}]$ is selected, Fig. 2.4. Since the joint A is in the origin of the reference system $A \equiv O$ the coordinates of A are $x_A = 0$, $y_A = 0$ and the position vector of A is

$\mathbf{r}_A = x_A\,\imath + y_A\,\jmath$. The vector $\mathbf{r}_A$ is introduced in MATLAB as

```
rA = [xA yA 0];
```

Position of joint D

The coordinates of the joint D are $x_D = L_a$, $y_D = L_b$ and the position vector of D is $\mathbf{r}_D = x_D\,\imath + y_D\,\jmath$.

Position of joint B

The unknowns are the coordinates of the joint B, x_B and y_B. Because the joint A is fixed and the angle ϕ is known, the coordinates of the joint B are computed from the following expressions

$x_B = AB\cos\phi = 0.106\,\text{m}, \quad y_B = AB\sin\phi = 0.106\,\text{m}.$

The position vector of B is $\mathbf{r}_B = x_B\,\imath + y_B\,\jmath$.

The MATLAB program for this part is

```
AB=0.15; BC=0.40; CD=0.37; CE=0.23; EF=CE;
La=0.30; Lb=0.45; Lc=CD;
phi = pi/6 ;
xA = 0; yA = 0; rA = [xA yA 0];
xD = La ; yD = Lb ; rD = [xD yD 0];
xB = AB*cos(phi); yB = AB*sin(phi); rB = [xB yB 0];
```

Position of joint C

The unknowns are the coordinates of the joint C, x_C and y_C. Knowing the positions of the joints B and D, the position of the joint C can be computed using the fact that the lengths of the links BC and CD are constants

$$
\begin{aligned}
&(x_C - x_B)^2 + (y_C - y_B)^2 = BC^2,\\
&(x_C - x_D)^2 + (y_C - y_D)^2 = CD^2,
\end{aligned}
\tag{2.10}
$$

or

$$
\begin{aligned}
&(x_C - 0.106)^2 + (y_C - 0.106)^2 = 0.400^2,\\
&(x_C - 0.300)^2 + (y_C - 0.450)^2 = 0.370^2,
\end{aligned}
\tag{2.11}
$$

Equations (2.11) consist of two quadratic equations. Solving this system of equations, two sets of solutions are found for the position of the joint C.

These solutions are

$$x_{C_1} = -0.069\,\text{m}, \quad y_{C_1} = 0.465\,\text{m},$$
$$x_{C_2} = 0.504\,\text{m}, \quad y_{C_2} = 0.141\,\text{m}. \tag{2.12}$$

The MATLAB program for calculating the coordinates of C_1 and C_2 is

```
eqnC1 = '( xCsol - xB )^2 + ( yCsol - yB )^2 = BC^2';
eqnC2 = '( xCsol - xD )^2 + ( yCsol - yD )^2 = CD^2';
solC = solve(eqnC1, eqnC2, 'xCsol, yCsol');
xCpositions = eval(solC.xCsol);
yCpositions = eval(solC.yCsol);
xC1 = xCpositions(1); % first component of the vector xCpositions
xC2 = xCpositions(2); % second component of the vector xCpositions
yC1 = yCpositions(1); % first component of the vector yCpositions
yC2 = yCpositions(2); % second component of the vector yCpositions
```

The points C_1 and C_2 are the intersections of the circle of radius BC (with its center at B) with the circle of radius CD (with its center at D), as shown in Fig. 2.5. To determine the position of the joint C for this mechanism, an additional constraint condition is needed: $x_C < x_D$. Because $x_D = 0.300\,\text{m}$, the coordinates of joint C have the following numerical values

$$x_C = x_{C_1} = -0.069\,\text{m}, \qquad y_C = y_{C_1} = 0.465\,\text{m}. \tag{2.13}$$

The MATLAB program for selecting the correct position of C is

```
if xC1 < xD
   xC = xC1; yC=yC1;
else
   xC = xC2; yC=yC2;
end
rC = [xC yC 0]; % Position vector of C
```

Position of joint E

The unknowns are the coordinates of the joint E, x_E and y_E. The position of the joint E is determined from the equation

$$(x_E - x_C)^2 + (y_E - y_C)^2 = CE^2, \tag{2.14}$$

or

$$(x_E + 0.069)^2 + (y_E - 0.465)^2 = 0.230^2.$$

The joints D, C and E are located on the same straight element DE. For these joints, the following equation can be written

$$\frac{y_D - y_C}{x_D - x_C} = \frac{y_E - y_C}{x_E - x_C}, \tag{2.15}$$

or

$$\frac{0.450 - 0.465}{0.300 + 0.069} = \frac{y_E - 0.465}{x_E + 0.069}.$$

Equations (2.14) and (2.15) form a system from which the coordinates of the joint E can be computed. Two solutions are obtained, Fig. 2.6, and the numerical values are

$$\begin{aligned} x_{E_1} &= -0.299\,\text{m}, \quad y_{E_1} = 0.474\,\text{m}, \\ x_{E_2} &= 0.160\,\text{m}, \quad y_{E_2} = 0.455\,\text{m}. \end{aligned} \tag{2.16}$$

The MATLAB program for calculating the coordinates of E_1 and E_2 is

```
eqnE1 = '( xEsol - xC )^2 + ( yEsol - yC )^2 = CE^2 ';
eqnE2 = '(yD-yC)/(xD-xC)=(yEsol-yC )/(xEsol-xC)';
solE = solve(eqnE1, eqnE2, 'xEsol, yEsol');
xEpositions=eval(solE.xEsol); yEpositions=eval(solE.yEsol);
xE1 = xEpositions(1); xE2 = xEpositions(2);
yE1 = yEpositions(1); yE2 = yEpositions(2);
```

For continuous motion of the mechanism, a constraint condition is needed, $x_E < x_C$. Using this condition, the coordinates of the joint E are

$$x_E = x_{E_1} = -0.300\,\text{m}, \ \ y_E = y_{E_1} = 0.475\,\text{m}.$$

The MATLAB program for selecting the correct position of E is

```
if xE1 < xC
   xE = xE1; yE=yE1;
else
   xE = xE2; yE=yE2;
end
```

```
rE = [xE yE 0]; % Position vector of E
```

Position of joint F

The joint F is restricted to move in a vertical direction, i.e. $x_F = -L_c = 0.370$ m. The coordinate y_F of the joint F can be calculated from the following quadratic equation

$$(x_F - x_E)^2 + (y_F - y_E)^2 = EF^2, \tag{2.17}$$

or

$$(0.370 + 0.300)^2 + (y_F - 0.475)^2 = 0.230^2,$$

The solutions of Eq. (2.17) are

$$y_{F_1} = 0.256\,\text{m}, \qquad y_{F_2} = 0.693\,\text{m}. \tag{2.18}$$

The points F_1 and F_2 are the intersections between the circle of radius EF (centered at E) and the vertical line with $x = x_F$, Fig. 2.7. For the mechanism depicted in Fig. 2.4, with $\theta = \pi/4$ the y coordinate of the joint F should be smaller that the y coordinate of the joint E, $y_F < y_E$. The y coordinate of the joint F is

$$y_F = y_{F_1} = 0.256\,\text{m}. \tag{2.19}$$

The MATLAB program for the position of F is

```
xF = - Lc ;
eqnF = '( xF - xE )^2 + ( yFsol - yE )^2 = EF^2 ';
solF = solve(eqnF,'yFsol');
yFpositions=eval(solF);
yF1 = yFpositions(1); yF2 = yFpositions(2);
if yF1 < yE
   yF=yF1;
else
   yF=yF2;
end
rF = [xF yF 0];
```

The angles of the links 2, 3, and 4 with the horizontal are

$$\phi_2 = \arctan\frac{y_B - y_C}{x_B - x_C}, \quad \phi_3 = \arctan\frac{y_D - y_C}{x_D - x_C}, \quad \phi_4 = \arctan\frac{y_F - y_E}{x_F - x_E},$$

and in MATLAB

```
phi2 = atan((yB-yC)/(xB-xC));
phi3 = atan((yD-yC)/(xD-xC));
phi4 = atan((yF-yE)/(xF-xE));
```

The results are printed using the statements

```
fprintf('rA = [ %g, %g, %g ] (m) \n', rA);
fprintf('rD = [ %g, %g, %g ] (m) \n', rD);
fprintf('rB = [ %g, %g, %g ] (m) \n', rB);
fprintf('rC = [ %g, %g, %g ] (m) \n', rC);
fprintf('rE = [ %g, %g, %g ] (m) \n', rE);
fprintf('rF = [ %g, %g, %g ] (m) \n', rF);
fprintf('phi2 = %g (degrees) \n', phi2*180/pi);
fprintf('phi3 = %g (degrees) \n', phi3*180/pi);
fprintf('phi4 = %g (degrees) \n', phi4*180/pi);
```

The graph of the mechanism using MATLAB for $\phi = \pi/4$ is given by

```
plot([xA,xB],[yA,yB],'r-o','LineWidth',1.5)
hold on % holds the current plot
plot([xB,xC],[yB,yC],'b-o','LineWidth',1.5)
hold on
plot([xD,xE],[yD,yE],'g-o','LineWidth',1.5)
hold on
plot([xE,xF],[yE,yF],'b-o','LineWidth',1.5)
grid on,...  % adds major grid lines to the current axes
xlabel('x (m)'), ylabel('y (m)'),...
title('positions for \phi = 45 (deg)'),...
text(xA,yA,'\leftarrow A = ground','HorizontalAlignment','left'),...
text(xB,yB,' B'),...
text(xC,yC,'\leftarrow C = ground','HorizontalAlignment','left'),...
text(xD,yD,'\leftarrow D = ground','HorizontalAlignment','left'),...
text(xE,yE,' E'), text(xF,yF,' F'), axis([-0.4 0.45 -0.1 0.6])
```

The graph of the R-RRR-RRT mechanism using MATLAB is shown in Fig. 2.8. The MATLAB program for the positions and the results is given in

Program 2.2.

2.4 R-RTR-RTR Mechanism

The planar R-RTR-RTR mechanism considered is shown in Fig. 2.9. The driver link is the rigid link 1 (the link AB). The following numerical data are given: $AB = 0.140$ m, $AC = 0.060$ m, $AE = 0.250$ m, $CD = 0.150$ m. The angle of the driver link 1 with the horizontal axis is $\phi = 30°$.

Solution

The MATLAB commands for the input data are

```
AB=0.140; AC=0.060; AE=0.250; CD=0.150;   %(m)
phi=pi/6;  %(rad)
```

Next the length of the links DF and EG are selected

```
DF=0.4; EG=0.5;   % (m)
```

A Cartesian reference frame xOy is selected. The joint A is in the origin of the reference frame, that is, $A \equiv O,\ \ x_A = 0,\ y_A = 0$.

Position of joint C

The position vector of C is $\mathbf{r}_C = x_C \mathbf{\imath} + y_C \mathbf{\jmath} = 0.060\ \mathbf{\jmath}$ m.

Position of joint E

The position vector of E is $\mathbf{r}_E = x_E \mathbf{\imath} + y_E \mathbf{\jmath} = -0.250\ \mathbf{\jmath}$ m.

Position of joint B

The unknowns are the coordinates of the joint B, x_B and y_B. Because the joint A is fixed and the angle ϕ is known, the coordinates of the joint B are computed from the following expressions

$$\begin{aligned} x_B &= AB \cos\phi = 0.140 \cos 30° = 0.121 \text{ m}, \\ y_B &= AB \sin\phi = 0.140 \sin 30° = 0.070 \text{ m}, \end{aligned} \tag{2.20}$$

and $\mathbf{r}_B = x_B \mathbf{\imath} + y_B \mathbf{\jmath}$.

The MATLAB statements for the positions of the joints A, C, E, and B are

```
xA = 0 ; yA = 0 ; rA = [xA yA 0] ;  % Position of A
```

```
xC = 0 ; yC = AC ; rC = [xC yC 0] ;  % Position of C
xE = 0 ; yE = -AE ; rE = [xE yE 0] ;   % Position of E
xB=AB*cos(phi); yB=AB*sin(phi); rB=[xB yB 0];  % Position of B
```

Position of joint D

The unknowns are the coordinates of the joint D, x_D and y_D. The length of the segment CD is constant:

$$(x_D - x_C)^2 + (y_D - y_C)^2 = CD^2, \tag{2.21}$$

or

$$(x_D - 0)^2 + (y_D - 0.060)^2 = 0.150^2.$$

The points B, C, and D are on the same straight line with the slope

$$m = \frac{(y_B - y_C)}{(x_B - x_C)} = \frac{(y_D - y_C)}{(x_D - x_C)}, \tag{2.22}$$

or

$$\frac{(0.070 - 0)}{(0.121 - 0.060)} = \frac{(y_D - 0)}{(x_D - 0.060)}.$$

Equations (2.21) and (2.22) form a system from which the coordinates of the joint D can be computed. To solve the system of equations the MATLAB statement `solve` will be used

```
eqnD1 = '( xDsol - xC )^2 + ( yDsol - yC )^2 = CD^2 ';
eqnD2 = '(yB - yC)/(xB - xC) = (yDsol - yC)/(xDsol - xC)';
solD = solve(eqnD1, eqnD2, 'xDsol, yDsol');
xDpositions = eval(solD.xDsol);
yDpositions = eval(solD.yDsol);
xD1 = xDpositions(1); % first component of the vector xDpositions
xD2 = xDpositions(2); % second component of the vector xDpositions
yD1 = yDpositions(1); % first component of the vector yDpositions
yD2 = yDpositions(2); % second component of the vector yDpositions
```

These solutions D_1 and D_2 are located at the intersection of the line BC with the circle centered in C and radius CD (Fig. 2.10), and they have the following numerical values:

$$x_{D1} = -0.149 \text{ m}, \; y_{D1} = 0.047 \text{ m},$$
$$x_{D2} = 0.149 \text{ m}, \; y_{D2} = 0.072 \text{ m}.$$

To determine the correct position of the joint D for the mechanism, an additional condition is needed. For the first quadrant, $0 \leq \phi \leq 90°$, the condition is $x_D \leq x_C$.
This condition with MATLAB is given by

```
if xD1 <= xC
  xD = xD1; yD=yD1;
else
  xD = xD2; yD=yD2;
end
rD = [xD yD 0]; % Position of D
```

Because $x_C = 0$, the coordinates of the joint D are

$$x_D = x_{D1} = -0.149\,\text{m} \quad \text{and} \quad y_D = y_{D1} = 0.047\,\text{m}.$$

The angles of the links 2, 3, and 4 with the horizontal are

$$\phi_2 = \arctan\frac{y_B - y_C}{x_B - x_C}, \quad \phi_3 = \phi_2, \quad \phi_4 = \arctan\frac{y_D - y_E}{x_D - x_E} + \pi, \quad \phi_5 = \phi_4,$$

and in MATLAB

```
phi2 = atan((yB-yC)/(xB-xC));
phi3 = phi2;
phi4 = atan((yD-yE)/(xD-xE))+pi;
phi5 = phi4;
```

The points F and G are calculated in MATLAB with

```
xF = xD + DF*cos(phi3) ; yF = yD + DF*sin(phi3) ;
rF = [xF yF 0]; % Position vector of F
xG = xE + EG*cos(phi5) ; yG = yE + EG*sin(phi5) ;
rG = [xG yG 0]; % Position vector of G
```

The results are printed using the statements

```
fprintf('rA = [ %g, %g, %g ] (m) \n', rA);
fprintf('rC = [ %g, %g, %g ] (m) \n', rC);
```

```
fprintf('rE = [ %g, %g, %g ] (m) \n', rE);
fprintf('rB = [ %g, %g, %g ] (m) \n', rB);
fprintf('rD = [ %g, %g, %g ] (m) \n', rD);
fprintf('phi2 = phi3 =%g (degrees) \n', phi2*180/pi);
fprintf('phi4 = phi5 =%g (degrees) \n', phi4*180/pi);
fprintf('rF = [ %g, %g, %g ] (m) \n', rF);
fprintf('rG = [ %g, %g, %g ] (m) \n', rG);
```

The graph of the mechanism in MATLAB for $\phi = \pi/6$ is given by

```
plot([xA,xB],[yA,yB],'r-o','LineWidth',1.5)
hold on % holds the current plot
plot([xD,xC],[yD,yC],'b-o','LineWidth',1.5)
hold on
plot([xC,xB],[yC,yB],'b-o','LineWidth',1.5)
hold on
plot([xB,xF],[yB,yF],'b-o','LineWidth',1.5)
hold on
plot([xE,xD],[yE,yD],'g-o','LineWidth',1.5)
hold on
plot([xD,xG],[yD,yG],'g-o','LineWidth',1.5)
grid on,...
xlabel('x (m)'), ylabel('y (m)'),...
title('positions for \phi = 30 (deg)'),...
text(xA,yA,'\leftarrow A = ground','HorizontalAlignment','left'),...
text(xB,yB,' B'),...
text(xC,yC,'\leftarrow C = ground','HorizontalAlignment','left'),...
text(xD,yD,' D'),...
text(xE,yE,'\leftarrow E = ground','HorizontalAlignment','left'),...
text(xF,yF,' F'), text(xG,yG,' G'), axis([-0.3 0.3 -0.3 0.3])
```

The graph of the mechanism using MATLAB is shown in Fig. 2.11. The MATLAB program for the positions and the results for the R-RTR-RTR mechanism is given in Program 2.3.

2.5 R-RTR-RTR Mechanism - Complete Rotation

For a complete rotation of the driver link AB, $0 \leq \phi \leq 360°$, a step angle of 60° is selected. To calculate the position analysis for a complete cycle the MATLAB statement `for` *var=startval:step:endval, statement* `end` is used. It repeatedly evaluates *statement* in a loop. The counter variable of the loop is *var*. At the start the variable is initialized to value *startval* and is incremented (or decremented when *step* is negative) by the value *step* for each iteration. The *statement* is repeated until *var* has incremented to the value *endval*. For the considered mechanism the following applies

```
for phi=0:pi/3:2*pi, Program block, end;
```

Method I - constraint conditions

Method I uses constraint conditions for the mechanism for each quadrant. For the mechanism, there are several conditions for the position of the joint D.

For the angle ϕ located in the first quadrant $0° \leq \phi \leq 90°$ (Fig. 2.10), and the fourth quadrant $270° \leq \phi \leq 360°$ (Fig. 2.14), the following relation exists between x_D and x_C:

$$x_D \leq x_C = 0.$$

For the angle ϕ located in the second quadrant $90° < \phi \leq 180°$ (Fig. 2.12), and the third quadrant $180° < \phi < 270°$ (Fig. 2.13), the following relation exists between x_D and x_C:

$$x_D \geq x_C = 0.$$

The following MATLAB commands are used to determine the correct position of the joint D for all four quadrants

```
if (phi>=0 && phi<=pi/2)||(phi >= 3*pi/2 && phi<=2*pi)
if xD1 <= xC xD = xD1; yD=yD1; else xD = xD2; yD=yD2; end
else
if xD1 >= xC xD = xD1; yD=yD1; else xD = xD2; yD=yD2; end
end
```

where `||` is the logical OR function.

The MATLAB program and the results for a complete rotation of the driver link using method I is given in Program 2.3. The graphic of the mechanism for a complete rotation of the driver link is given in Fig. 2.15. To simplify the graphic the points E and G are not shown on the figure.

Method II - Euclidian distance function

Another position analysis method for a complete rotation of the driver link uses constraint conditions only for the initial value of the angle ϕ. Next for the mechanism, the correct position of the joint D is calculated using a simple function, the Euclidian distance between two points P and Q:

$$d = \sqrt{(x_P - x_Q)^2 + (y_P - y_Q)^2}. \tag{2.23}$$

In MATLAB, the following function is introduced with a m-file (Dist.m):

```
function d=Dist(xP,yP,xQ,yQ);
d=sqrt((xP-xQ)^2+(yP-yQ)^2);
end
```

For the initial angle $\phi = 0°$, the constraint is $x_D \leq x_C$, so the first position of the joint D, that is, D_0, is calculated for the first step $D = D_0 = D_k$, $k = 0$. For the next position of the joint, D_{k+1}, there are two solutions D^I_{k+1} and D^{II}_{k+1}, $k = 0,\ 1,\ 2, \ldots$. In order to choose the correct solution of the joint, D_{k+1}, it is compared the distances between the old position, D_k, and each new calculated positions D^I_{k+1} and D^{II}_{k+1}. The distances between the known solution D_k and the new solutions D^I_{k+1} and D^{II}_{k+1} are d^I_k and d^{II}_k. If the distance to the first solution is less than the distance to the second solution, $d^I_k < d^{II}_k$, then the correct answer is $D_{k+1} = D^I_{k+1}$, or else $D_{k+1} = D^{II}_{k+1}$ (Fig. 2.16).

The following MATLAB statements are used to determine the correct position of the joint D using a single condition for all four quadrants:

```
increment=0;   % at the initial moment phi=0 => increment = 0
step=pi/6;     % the step has to be small for this method
for phi=0:step:2*pi,
...
if increment == 0
   if xD1 <= xC xD = xD1; yD=yD1; else xD = xD2; yD=yD2; end
else
   dist1 = Dist(xD1,yD1,xDold,yDold);
   dist2 = Dist(xD2,yD2,xDold,yDold);
if dist1 < dist2 xD = xD1; yD=yD1; else xD = xD2; yD=yD2; end
end
xDold=xD;
yDold=yD;
increment=increment+1;
...
end
```

At the beginning of the rotation the driver link makes an angle `phi=0` with the horizontal and the value of counter `increment` is 0. The MATLAB statement

```
increment=increment+1;
```

specifies that 1 is to be added to the value in `increment` and the result stored back in `increment`. The value `increment` should be incremented by 1.

With this algorithm the correct solution is selected using just one constraint relation for the initial step and then, automatically, the problem is solved. In this way it is not necessary to have different constraints for different quadrants.

For the Euclidian distance method the selection the step of the angle ϕ is very important. If the step of the angle has a large value the method might give wrong answers and that is why it is important to check the graphic of the mechanism.

The MATLAB program and the results for a complete rotation of the driver link using the second method is given in Program 2.5. The graph of the mechanism for a complete rotation of the driver link (the step of the angle is 30°) is given in Fig. 2.17 (the points E and G are not shown).

2.6 Path of a Point on a Link with General Plane Motion

R-RRT mechanism

The mechanism shown in Fig. 2.18(a) has $AB = 0.5$ m and $BC = 1$ m. The link 2 (connecting rod BC) has a general plane motion: translation along x-axis, translation along y-axis, and rotation about z-axis. The mass center of link 2 is located at C_2. Determine the path of point C_2 for a complete rotation of the driver link 1.

Solution

The coordinates of the joint B are: $x_B = AB \cos\phi$ and $y_B = AB \sin\phi$, where $0 \leq \phi \leq 360°$. The coordinates of the joint C are: $x_C = x_B + \sqrt{BC^2 - y_B^2}$ and $y_C = 0$. The mass center of the link 2 is the midpoint of the segment BC

$$x_{C_2} = \frac{x_B + x_C}{2} \quad \text{and} \quad y_{C_2} = \frac{y_B + y_C}{2}.$$

The MATLAB statements for the coordinates of C_2 are

```
AB = .5; BC = 1; xA = 0; yA = 0; yC = 0;
incr = 0;
for phi=0:pi/10:2*pi,
  xB = AB*cos(phi); yB = AB*sin(phi);
  xC = xB + sqrt(BC^2-yB^2);
  incr = incr + 1;
  xC2(incr)=(xB+xC)/2; yC2(incr)=(yB+yC)/2;
end % end for
```

For the complete rotation of the driver link AB, $0 \leq \phi \leq 360°$, a step angle of $\pi/10$ was selected.
For the coordinates of C_2 two vectors

`xC2=[xC2(1) xC2(2) ...  xC2(incr) ...  ]` and
`yC2=[yC2(1) yC2(2) ...  yC2(incr) ...  ]` are obtained.

The first components `xC2(1)` and `yC2(1)` are calculated for `phi=0` and `incr=1`. The path of C_2 is obtained plotting the vector `yC2` in terms of `xC2`

```
plot(xC2, yC2, '-ko'),...
```

```
xlabel('x (m)'), ylabel('y (m)'),...
title('Path described by C2'), grid
```

Figure 2.18(b) shows two plots: the mechanism for $0 \leq \phi \leq 360°$ and the closed path described by the point C_2 on the link 2 in general plane motion. The plots are obtained using Program 2.6.

R-RRR-RRT mechanism

The mechanism shown in Fig. 2.4 has the dimensions given at subsection 2.3. The link 2 (link BC) has a general plane motion. The positions of the mechanism for $0 \leq \phi \leq 360°$ and the closed path described by the mass center C_2 of the link 2 are shown in Figure 2.19. The plots are obtained using Program 2.7.

2.7 Creating a Movie

The R-RRR-RRT mechanism shown in Fig. 2.4 has the dimensions given at subsection 2.3. This example illustrates the use of movies to visualize the positions of the mechanism for $0 \leq \phi \leq 360°$.

The first step is to create axis the same size as the ones that will display the movie The axis command is

```
axis([-0.3 0.3 -0.3 0.3])
```

The statement, `moviein` is used to create a matrix large enough to hold the 6 frames that compose the movie of the mechanism for 6 values of the driver angle ϕ.

```
M = moviein(6);
```

The `axis` statement must precede the `M = moviein(6);` statement to ensure `M` is initialized to the correct dimensions. Next the command,

```
set(gca,'NextPlot','replacechildren')
```

is employed to prevent the plot function from resetting the axis shaping to axis normal each time it is called.

The program has the structure

```
clear; clc; close all
AB = 0.14; AC = 0.06; AE = 0.25; CD = 0.15;
xA = 0; yA = 0; xC = 0 ; yC = AC; xE = 0; yE = -AE;

axis([-0.3 0.3 -0.3 0.3]);
M = moviein(6);
set(gca,'NextPlot','replacechildren')

incr = 0;
for phi=pi/3:pi/3:2*pi,
xB = AB*cos(phi); yB = AB*sin(phi);
eqnD1 = '( xDsol - xC )^2 + ( yDsol - yC )^2 = CD^2 ';
eqnD2 = '(yB - yC) / (xB - xC) = (yDsol - yC) / (xDsol - xC)';
solD = solve(eqnD1, eqnD2, 'xDsol, yDsol');
xDpositions = eval(solD.xDsol); yDpositions = eval(solD.yDsol);
xD1 = xDpositions(1); xD2 = xDpositions(2);
yD1 = yDpositions(1); yD2 = yDpositions(2);
if (phi>=0 && phi<=pi/2)||(phi >= 3*pi/2 && phi<=2*pi)
if xD1 <= xC xD = xD1; yD=yD1; else xD = xD2; yD=yD2; end else
if xD1 >= xC xD = xD1; yD=yD1; else xD = xD2; yD=yD2; end end
plot([xA,xB],[yA,yB],'r-o',[xB,xD],[yB,yD],'b-o',[xD,xE],[yD,yE],'g-o'),...
text(xA,yA,' A'), text(xB,yB,' B'), text(xC,yC,' C'),...
text(xD,yD,' D'), text(xE,yE,' E')
incr = incr + 1;

M(:,incr) = getframe;

end
```

The statement, `getframe` returns the contents of the current axes, exclusive of the axis labels, title, or tick labels. After generating the movie, it can be play back any number of times. To play it back 5 times, type `movie(M,5)`.