Dynamics of Mechanical Systems
Academic Year 2023/2024
Homework 1

Afonso Henrique, 98555 - afonso.henrique@tecnico.ulisboa.pt

1 Objectives

Write the vector loop closure for the mechanism presented in the figure and:

1. Develop and present the position constraint equations, the velocity constraint equations and the acceleration constraint equations and report them.
2. Identify the Jacobian matrix and the right-hand-sides of the velocity and acceleration equations and report them.
3. Build the preprocessor script and the kinematic evaluation function in Matlab and use the program developed in class to solve the kinematic analysis of the mechanical system for 2 revolutions of the crank, with a constant angular velocity of 2π rad/s, and answer the question associated to your mechanism. Report the script, the kinematic evaluation function and the requested results of the kinematic analysis.

If any dimension is missing or you feel as being required just measure it in the drawing and scale it appropriately.

2 Defining Vector Loops

For my student number, the mechanism to be used was number 4, and the chosen vector loops can be found in the following figure:

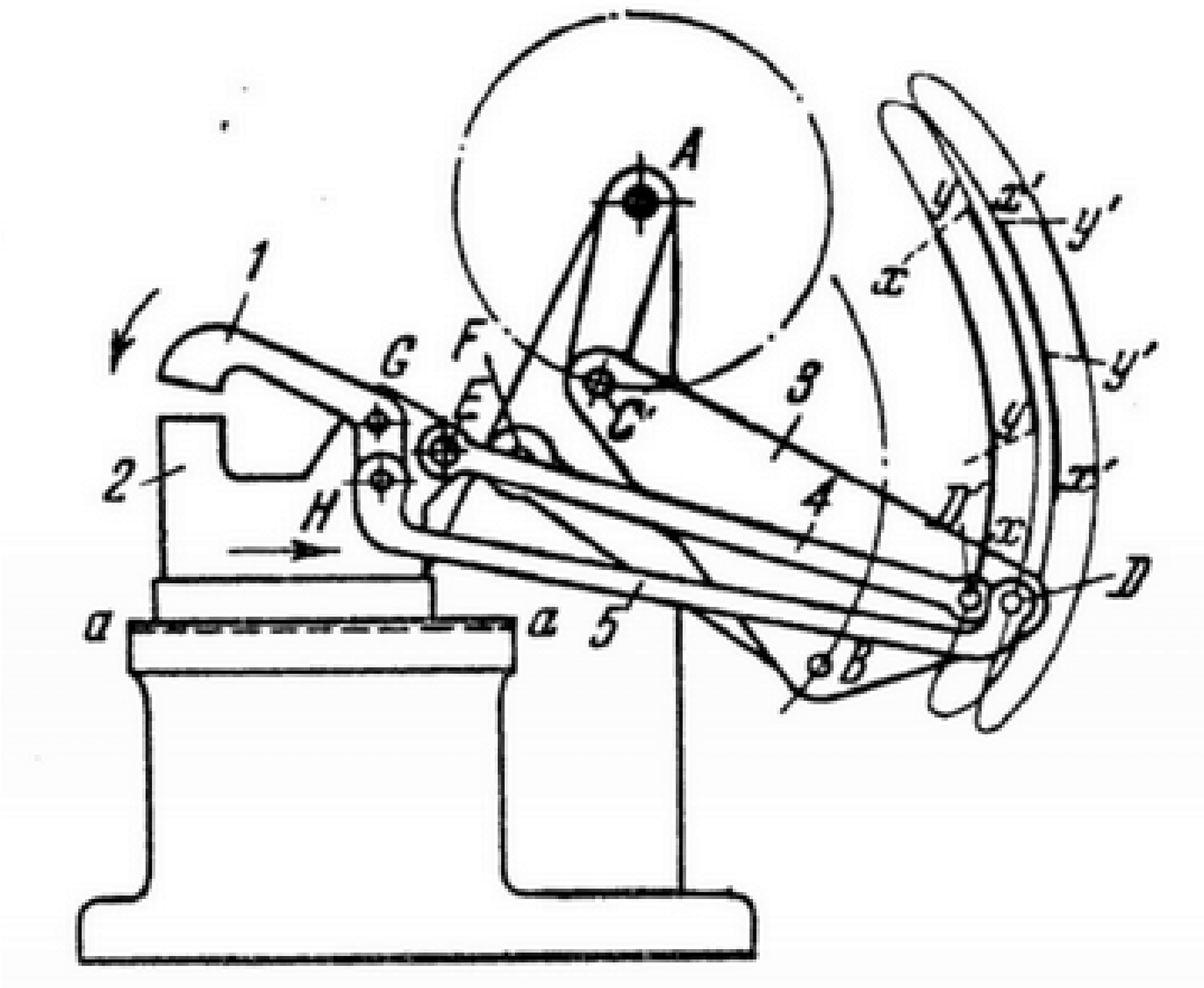

Figure 1: Mechanism number 4: Slider-crank gripping and feeding mechanism.

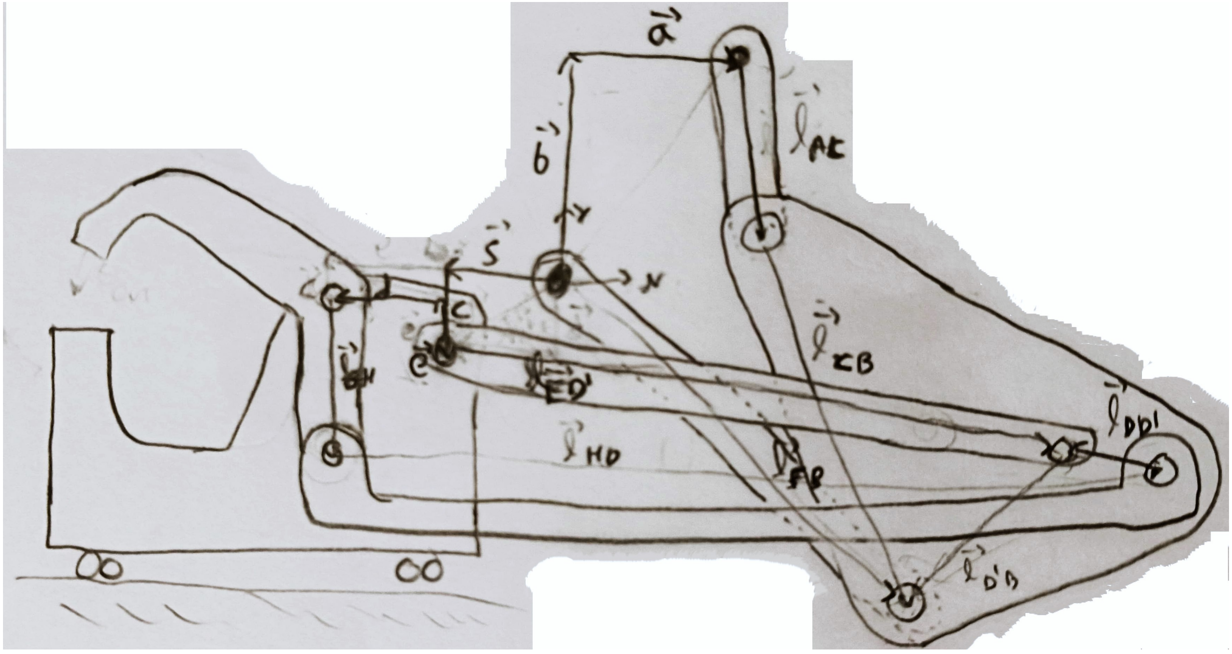

Figure 2: Drawing of the mechanism with the chosen vector loops.

3 Position, Velocity & Acceleration Equations

The chosen vectors loops resulted in the 6 following kinematic constraint equations, and the last line is the driver equation for element AC:

$$
\Phi \begin{cases}
a + l\cdot\cos(\theta_2) + 2l\cdot\cos(\theta_4) - 2l\cdot\cos(\theta_3) = 0 \\
b + l\cdot\sin(\theta_2) + 2l\cdot\sin(\theta_4) - 2l\cdot\sin(\theta_3) = 0 \\
-d + 0.35\cdot l\cdot\cos(\theta_1) + 3,6l\cdot\cos(\theta_6) + 0,2428\cdot l\cdot\cos(\theta_4 - 137,85) - 2,8l\cdot\cos(\theta_5) = 0 \\
c + 0,35l\cdot\sin(\theta_1) + 3,6l\cdot\sin(\theta_6) + 0,2428\cdot l\cdot\sin(\theta_4 - 137,85) - 2,8l\cos(\theta_5) = 0 \\
x_1 + 2,8l\cdot\cos(\theta_5) + 0,9l\cdot\cos(\theta_4 - 99,15) - 2l\cdot\cos(\theta_3) = 0 \\
-e + 2,8l\sin(\theta_5) + 0,9l\sin(\theta_4 - 99,15) - 2l\sin(\theta_3) = 0 \\
\theta_2(t) = \omega\cdot t + \theta_2^0
\end{cases}
$$

Figure 3: 6 kinematic constraint equations and one driver equation.

Where the length AC was defined as l, and the remaining lengths were defined according to the relations given in the statement or found by importing the picture of the mechanism into SolidWorks and measuring. The angles in element BCDD' were also found by measuring in SolidWorks.

To simplify the equations the angles of the members were replaced in the following manner:

- $\theta_1 = \theta_{GH}$
- $\theta_2 = \theta_{AC}$
- $\theta_3 = \theta_{FB}$
- $\theta_4 = \theta_{CB} = \theta_{DD'} + 137.85^\circ = \theta_{D'B} + 99.15^\circ$
- $\theta_5 = \theta_{ED'}$
- $\theta_6 = \theta_{HD}$
- $x_1 = s$

And the lengths were also named according to the angles.

9

$$\dot{\Phi} \Rightarrow \begin{cases} -l\sin(\theta_2)\dot{\theta}_2 - 2l\sin(\theta_4)\dot{\theta}_4 + 2l\sin(\theta_3)\dot{\theta}_3 = 0 \\ l\cos(\theta_2)\dot{\theta}_2 + 2l\cos(\theta_4)\dot{\theta}_4 - 2l\cos(\theta_3)\dot{\theta}_3 = 0 \\ -0{,}35l\sin(\theta_1)\dot{\theta}_1 - 3{,}6l\sin(\theta_6)\dot{\theta}_6 - 0{,}2428l\sin(\theta_4-137{,}85)\dot{\theta}_4 + 2{,}8l\sin(\theta_5)\dot{\theta}_5 = 0 \\ 0{,}35l\cos(\theta_1)\dot{\theta}_1 + 3{,}6l\cos(\theta_6)\dot{\theta}_6 + 0{,}2428l\cos(\theta_4-137{,}85)\dot{\theta}_4 - 2{,}8l\cos(\theta_5)\dot{\theta}_5 = 0 \\ \dot{x}_1 - 2{,}8l\sin(\theta_5)\dot{\theta}_5 - 0{,}9l\sin(\theta_4-99{,}15)\dot{\theta}_4 + 2l\cdot\sin(\theta_3)\dot{\theta}_3 = 0 \\ 2{,}8l\cos(\theta_5)\dot{\theta}_5 + 0{,}9l\cos(\theta_4-99{,}15)\dot{\theta}_4 - 2l\cos(\theta_3)\dot{\theta}_3 = 0 \\ \dot{\theta}_2 = \omega \end{cases}$$

$$\ddot{\Phi} \Rightarrow \begin{cases} -l\sin(\theta_2)\ddot{\theta}_2 - 2l\sin(\theta_4)\ddot{\theta}_4 + 2l\sin(\theta_3)\ddot{\theta}_3 = l\cos(\theta_2)\dot{\theta}_2^2 + 2l\cos(\theta_4)\dot{\theta}_4^2 - 2l\cos(\theta_3)\dot{\theta}_3^2 \\ l\cos(\theta_2)\ddot{\theta}_2 + 2l\cos(\theta_4)\ddot{\theta}_4 - 2l\cos(\theta_3)\ddot{\theta}_3 = l\sin(\theta_2)\dot{\theta}_2^2 + 2l\sin(\theta_4)\dot{\theta}_4^2 - 2l\sin(\theta_3)\dot{\theta}_3^2 \\ -0{,}35l\sin(\theta_1)\ddot{\theta}_1 - 3{,}6l\sin(\theta_6)\ddot{\theta}_6 - 0{,}2428l\sin(\theta_4-137{,}85)\ddot{\theta}_4 + 2{,}8l\sin(\theta_5)\ddot{\theta}_5 = \\ \quad = 0{,}35l\cos(\theta_1)\dot{\theta}_1^2 + 3{,}6l\cos(\theta_6)\dot{\theta}_6^2 + 0{,}2428l\cos(\theta_4-137{,}85)\dot{\theta}_4^2 - 2{,}8l\cos(\theta_5)\dot{\theta}_5^2 \\ 0{,}35l\cos(\theta_1)\ddot{\theta}_1 + 3{,}6l\cos(\theta_6)\ddot{\theta}_6 + 0{,}2428l\cos(\theta_4-137{,}85)\ddot{\theta}_4 - 2{,}8l\cos(\theta_5)\ddot{\theta}_5 = \\ \quad = 0{,}35l\sin(\theta_1)\dot{\theta}_1^2 + 3{,}6l\sin(\theta_6)\dot{\theta}_6^2 + 0{,}2428l\sin(\theta_4-137{,}85)\dot{\theta}_4^2 - 2{,}8l\sin(\theta_5)\dot{\theta}_5^2 \\ \ddot{x}_1 - 2{,}8l\sin(\theta_5)\ddot{\theta}_5 - 0{,}9l\sin(\theta_4-99{,}15)\ddot{\theta}_4 + 2l\sin(\theta_3)\ddot{\theta}_3 = 2{,}8l\cos(\theta_5)\dot{\theta}_5^2 + 0{,}9l\cos(\theta_4-99{,}15)\dot{\theta}_4^2 - \\ \quad - 2l\cos(\theta_3)\dot{\theta}_3^2 \\ 2{,}8l\cos(\theta_5)\ddot{\theta}_5 + 0{,}9l\cos(\theta_4-99{,}15)\ddot{\theta}_4 - 2l\cos(\theta_3)\ddot{\theta}_3 = 2{,}8l\sin(\theta_5)\dot{\theta}_5^2 + 0{,}9l\sin(\theta_4-99{,}15)\dot{\theta}_4^2 - 2l\sin(\theta_3)\dot{\theta}_3^2 \\ \ddot{\theta}_2 = 0 \end{cases}$$

Figure 4: Velocity and acceleration constraint equations.

4 Jacobian Matrix and R.H.S. of Velocity and Acceleration Equations

The previous equations resulted in the following Jacobian matrix and the vectors of R.H.S. of the velocity and acceleration equations:

$$\mathrm{J} = \begin{bmatrix} 0 & -l2\sin(\theta 2) & l3\sin(\theta 3) & -l4\sin(\theta 4) & 0 & 0 & 0 \\ 0 & l2\cos(\theta 2) & -l3\cos(\theta 3) & l4\cos(\theta 4) & 0 & 0 & 0 \\ -l1\sin(\theta 1) & 0 & 0 & -0.2428l\sin(\theta 4-137.85) & l5\sin(\theta 5) & -l6\sin(\theta 6) & 0 \\ l1\cos(\theta 1) & 0 & 0 & 0.2428l\cos(\theta 4-137.85) & -l5\cos(\theta 5) & l6\cos(\theta 6) & 0 \\ 0 & 0 & l3\sin(\theta 3) & -0.9l\sin(\theta 4-99.15) & -l5\sin(\theta 5) & 0 & 1 \\ 0 & 0 & -l3\cos(\theta 3) & 0.9l\cos(\theta 4-99.15) & l5\cos(\theta 5) & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\nu = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \omega \end{bmatrix}$$

$$\gamma = \begin{bmatrix} l2\cos(\theta 2)\cdot\ddot{\theta_2}^2 + l4\cos(\theta 4)\cdot\ddot{\theta_4}^2 - l3\cos(\theta 3)\cdot\ddot{\theta_3}^2 \\ l2\sin(\theta 2)\cdot\ddot{\theta_2}^2 + l4\sin(\theta 4)\cdot\ddot{\theta_4}^2 - l3\sin(\theta 3)\cdot\ddot{\theta_3}^2 \\ l1\cos(\theta 1)\cdot\ddot{\theta_1}^2 + l6\cos(\theta 6)\cdot\ddot{\theta_6}^2 + 0.2428l\cos(\theta 4 - 137.85)\cdot\ddot{\theta_4}^2 - l5\cos(\theta 5)\cdot\ddot{\theta_5}^2 \\ l1\sin(\theta 1)\cdot\ddot{\theta_1}^2 + l6\sin(\theta 6)\cdot\ddot{\theta_6}^2 + 0.2428l\sin(\theta 4 - 137.85)\cdot\ddot{\theta_4}^2 - l5\sin(\theta 5)\cdot\ddot{\theta_5}^2 \\ l5\cos(\theta 5)\cdot\ddot{\theta_5}^2 + 0.9l\cos(\theta 4 - 99.15)\cdot\ddot{\theta_4}^2 - l3\cos(\theta 3)\cdot\ddot{\theta_3}^2 \\ l5\sin(\theta 5)\cdot\ddot{\theta_5}^2 + 0.9l\sin(\theta 4 - 99.15)\cdot\ddot{\theta_4}^2 - l3\sin(\theta 3)\cdot\ddot{\theta_3}^2 \\ 0 \end{bmatrix}$$

5 Graphical Results

The trajectories of the relevant points can be found in the following figure:

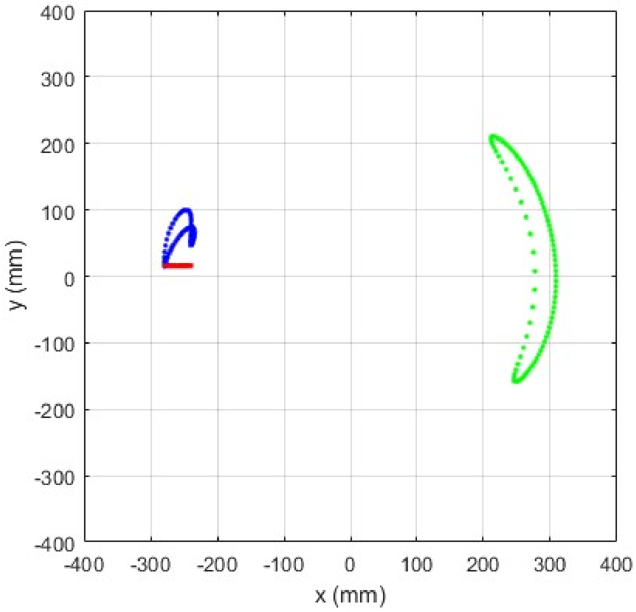

Figure 5: Trajectory of points in opposite faces of the gripper (blue and red) and point D(green).

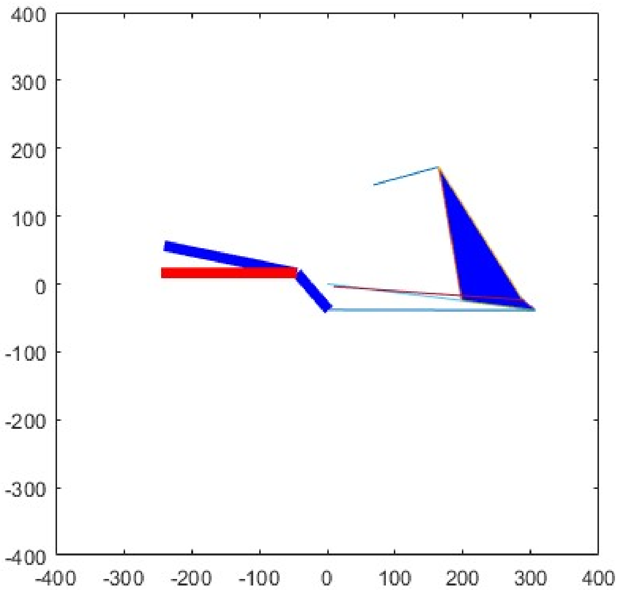

Figure 6: Picture of the mechanism in motion at one time instant (simplified representation).

6 Matlab Code

ModelAndSimulation.m was modified to be as following:

```
clear all
%
%... Model & simulation scenario filename
filename = 'SliderCrankSimple.mat';
%
%... Model dimensions
Model.l1 = 35;
Model.l2 = 100;
Model.l3 = 200;
Model.l4 = 200;
Model.l5 = 280;
Model.l6 = 360;
%
%... Driver characteristics
Model.anginit = 256.52*(pi/180);
Model.angvelocity = 2*pi;
Model.NCoordinates = 7;
%
%... Function for equations evaluations
Model.Function = @Slider_Crank_Simple;
%
%... Time analysis parameters
Time.start = 0.0;
Time.end = 2;
Time.step = 0.01;
%
%... Analysis parameters
Parameter.NLTolerance = 0.000001;
Parameter.NLMaxIter = 12;
%
%... Initial positions (estimate)
theta1 = 260.22*(pi/180);
theta2 = Model.anginit;
theta3 = 327.01*(pi/180);
theta4 = 307.98*(pi/180);
theta5 = 345.63*(pi/180);
theta6 = 350.54*(pi/180);
x1 = -24.64;
%
%... Initial coordinates vector
q0 = [theta1; theta2; theta3; theta4; theta5; theta6; x1];
%
%... Save the data in 'mat' file
save(filename);
```

SliderCrankSimple.m was modified to be as following:

```
    function [phi,Jac,niu,gamma] = Slider_Crank_Simple(time,q,qd)
%
%Summary: This function controls the construction of vectors and
% matrices required to solve the complete kinematic
% analysis of a simple slider-crank mechanism, i.e.,the
% vector with constraint equations, Jacobian matrix,
% r.h.s. of the velocity and acceleration equations.
% Furthermore, it transfers variables from the global
% arrays to local storage.
%
%Input: time - Time in which the positions are evaluated
% q - System positions
% qd - System velocities
%
%Output: Phi - Vector with the kinematic constraints
% Jac - Jacobian matrix
% niu - r.h.s of the velocity equations
```

```
% gamma - r.h.s. of the acceleration equations
%
%Shared: Flag - Flags for the Function Evaluation
% Model - Model parameters
%
%%
%... Access global memory
global Flag Model
%%
%... Initialize workspaces
phi = [];
Jac = [];
niu = [];
gamma = [];
%%
%... Transfer coordinates from global to local storage
o1 = q(1);
o2 = q(2);
o3 = q(3);
o4 = q(4);
o5 = q(5);
o6 = q(6);
x1 = q(7);
l1 = Model.l1;
l2 = Model.l2;
l3 = Model.l3;
l4 = Model.l4;
l5 = Model.l5;
l6 = Model.l6;
a = 68;
b = 146;
c = 20;
d = 54;
e = 4;
l=Model.l2;
%%
%... Evaluate the position constraint vector Phi(q,t)
if Flag.Position == 1
 phi = [
 a+l2*cos(o2)+l4*cos(o4)-l3*cos(o3);
 b+l2*sin(o2)+l4*sin(o4)-l3*sin(o3);
 -d+l1*cos(o1)+l6*cos(o6)+0.2428*l*cos(o4-deg2rad(137.85))-l5*cos(o5);
 c+l1*sin(o1)+l6*sin(o6)+0.2428*l*sin(o4-deg2rad(137.85))-l5*sin(o5);
 x1+l5*cos(o5)+0.9*l*cos(o4-deg2rad(99.15))-l3*cos(o3);
 -e+l5*sin(o5)+0.9*l*sin(o4-deg2rad(99.15))-l3*sin(o3);
 o2-(Model.anginit+Model.angvelocity*time) ];
end
%%
%... Evaluate the constraints Jacobian matrix Phi(q,t)
if Flag.Jacobian == 1
 Jac = [
0                -l2*sin(o2)    l3*sin(o3)   -l4*sin(o4)                              0
                0          0;
0                l2*cos(o2)     -l3*cos(o3)   l4*cos(o4)                              0
                0          0;
-l1*sin(o1)      0             0             -0.2428*l*sin(o4-deg2rad(137.85))     l5*sin(
    o5)   -l6*sin(o6)    0;
l1*cos(o1)       0             0              0.2428*l*cos(o4-deg2rad(137.85))     -l5*cos(
    o5)   l6*cos(o6)    0;
0                0            l3*sin(o3)     -0.9*l*sin(o4-deg2rad(99.15))           -l5*
    sin(o5)     0          1;
0                0            -l3*cos(o3)     0.9*l*cos(o4-deg2rad(99.15))           l5*cos
    (o5)     0          0;
0                1             0             0                                      0
                 0          0];
end
```

```
%%
%... Evaluate vector with r.h.s. of velocity equations
if Flag.Velocity == 1
 niu = [0; 0; 0; 0; 0; 0; Model.angvelocity];
end
%%
%... Evaluate vector with r.h.s. of acceleration equations
if Flag.Acceleration == 1
 o1d = qd(1);
 o2d = qd(2);
 o3d = qd(3);
 o4d = qd(4);
 o5d = qd(5);
 o6d = qd(6);
 x1d = qd(7);
 gamma = [
 l2*cos(o2)*o2d^2+l4*cos(o4)*o4d^2-l3*cos(03)*o3d^2;
 l2*sin(o2)*o2d^2+l4*sin(o4)*o4d^2-l3*sin(03)*o3d^2;
 l1*cos(o1)*o1d^2+l6*cos(o6)*o6d^2+0.2428*l*cos(o4-deg2rad(137.85))*o4d^2-l5*cos(o5)*o5d
     ^2;
 l1*sin(o1)*o1d^2+l6*sin(o6)*o6d^2+0.2428*l*sin(o4-deg2rad(137.85))*o4d^2-l5*sin(o5)*o5d
     ^2;
 l5*cos(o5)*o5d^2+0.9*l*cos(o4-deg2rad(99.15))*o4d^2-l3*cos(o3)*o3d^2;
 l5*sin(o5)*o5d^2+0.9*l*sin(o4-deg2rad(99.15))*o4d^2-l3*sin(o3)*o3d^2;
 0 ];
end
%%
%... Finalize function Slider_Crank_Simple
end
```

FirstPostProcessResults.m was modified to be as following:

```
    function []=First_Post_Process_Results(t, q, qd, qdd)
% Post process data
%ReportData(t,q,qd,qdd);
figure(1);
subplot(3,1,1);plot(t,q(1,:)*180/pi),xlabel('time(s)'),ylabel('theta_1')
grid on; title('Position vector vs time')
subplot(3,1,2);plot(t,q(2,:)*180/pi),xlabel('time(s)'),ylabel('theta_2')
grid on
subplot(3,1,3);plot(t,q(7,:)),xlabel('time(s)'),ylabel('x1')
grid on
%======================================================================
figure(2);
subplot(3,1,1);plot(t,qd(1,:)*180/pi),xlabel('time(s)'),ylabel('dtheta_1')
grid on; title('Velocity vector vs time')
subplot(3,1,2);plot(t,qd(2,:)*180/pi),xlabel('time(s)'),ylabel('dtheta_2')
grid on
subplot(3,1,3);plot(t,qd(7,:)),xlabel('time(s)'),ylabel('dx1')
grid on
%======================================================================
figure(3);
subplot(3,1,1);plot(t,qdd(1,:)*180/pi),xlabel('time(s)'),ylabel('ddtheta_1')
grid on; title('Acceleration velocity vs time')
subplot(3,1,2);plot(t,qdd(2,:)*180/pi),xlabel('time(s)'),ylabel('ddtheta_2')
grid on
subplot(3,1,3);plot(t,qdd(7,:)),xlabel('time(s)'),ylabel('ddx1')
grid on
%======================================================================
global Model

a = 68;
b = 146;
c = 20;
d = 54;
e = 4;

figure(4)
```

```
NTime = length(t);
for i = 1 : NTime
        A = [a , b];
        B =Model.l3*[cos(q(3,i)) , sin(q(3,i))  ];
        C =  A + Model.l2 *[ cos(q(2,i)),  sin(q(2,i))];
        D_linha = B - 0.9*Model.l2*[ cos( q(4,i)-deg2rad(99.15) ),  sin( q(4,i)-deg2rad
            (99.15) ) ];
        D = D_linha  - 0.2428*Model.l2*[ cos( q(4,i)-deg2rad(138.85) ),  sin( q(4,i)-
            deg2rad(138.85) ) ];
        E = D_linha - Model.l5*[ cos(q(5,i)),  sin(q(5,i))];
        F =[0,0];
        G = E+[-d,c];
        H = E+Model.l1 *[ cos(q(1,i)), sin(q(1,i))];
        pinca = G + 200*[ cos(q(1,i)-deg2rad(90)), sin(q(1,i)-deg2rad(90))];
        base =  G - 200*[1 , 0];

    plot(D(1),D(2),'g.');
    hold on;
    plot(pinca(1),pinca(2),'b.');
    hold on;
    plot(base(1),base(2),'r.');
    hold on;
end
    xlabel('x (mm)');
    ylabel('y (mm)');
    grid on;
    title('Trajectory of points in opposite faces of the gripper and point D')
    axis([-400 400 -400 400]);
    pbaspect([1 1 1]);

figure(5)
pause()
    NTime = length(t);
    for i = 1 : NTime
        A = [a , b];
        B =Model.l3*[cos(q(3,i)) , sin(q(3,i))  ];
        C =  A + Model.l2 *[ cos(q(2,i)),  sin(q(2,i))];
        D_linha = B - 0.9*Model.l2*[ cos( q(4,i)-deg2rad(99.15) ),  sin( q(4,i)-deg2rad
            (99.15) ) ];
        D = D_linha  - 0.2428*Model.l2*[ cos( q(4,i)-deg2rad(138.85) ),  sin( q(4,i)-
            deg2rad(138.85) ) ];
        E = D_linha - Model.l5*[ cos(q(5,i)),  sin(q(5,i))];
        F =[0,0];
        G = E+[-d,c];
        H = E+Model.l1 *[ cos(q(1,i)), sin(q(1,i))];
        pinca = G + 200*[ cos(q(1,i)-deg2rad(90)), sin(q(1,i)-deg2rad(90))];
        base =  G - 200*[1 , 0];

        %AC
        plot([A(1) C(1)], [A(2) C(2)]);
        hold on;

        fill([C(1) B(1) D(1) D_linha(1)] , [C(2) B(2) D(2) D_linha(2)] , 'b')

        %CB
        plot([C(1) B(1)], [C(2) B(2)]);
        hold on;

        %CD'
        plot([C(1) D_linha(1)], [C(2) D_linha(2)]);
        hold on;

        %D'D
        plot([D_linha(1) D(1) ], [D_linha(2)  D(2) ]);
        hold on;
```

```
        %BD
        plot([B(1) D(1)], [B(2) D(2)]);
        hold on;

        %FB
        plot([F(1) B(1)], [F(2) B(2)]);
        hold on;

        %ED'
        plot([E(1) D_linha(1)], [E(2) D_linha(2)]);
        hold on;

        %HD
        plot([H(1) D(1)], [H(2) D(2)]);
        hold on;

        %HG
        plot([H(1) G(1)], [H(2) G(2)], 'b','LineWidth',5);

        %Gpinca
        plot([G(1) pinca(1)], [G(2) pinca(2)], 'b','LineWidth',5);
        hold on;

        %Gbase
        plot([G(1) base(1)], [G(2) base(2)], 'r','LineWidth',5);
        hold on;

        hold off;
        axis([-400 400 -400 400]);
        pbaspect([1 1 1]);
        pause(0.01)

    end

end
```