Geometric modeling in MATLAB: Slider crank mechanism

Description

The objective of this exercise is to design and animate a slider crank mechanism in MATLAB. The instructions are provided in this document on how to design and animate the mechanism step by step. A well commented code will be expected for this work as well as for the assignment work. This exercise will be carried out during lab session and a seperate assignment will be provided. Please note, this is an **individual work and no team work**.

Slider crank design in MATLAB

The exercise will focus only on planar modeling (like the sketching module of CATIA). The slider crank mechanism along with its design parameters is represented below in Figure 1.

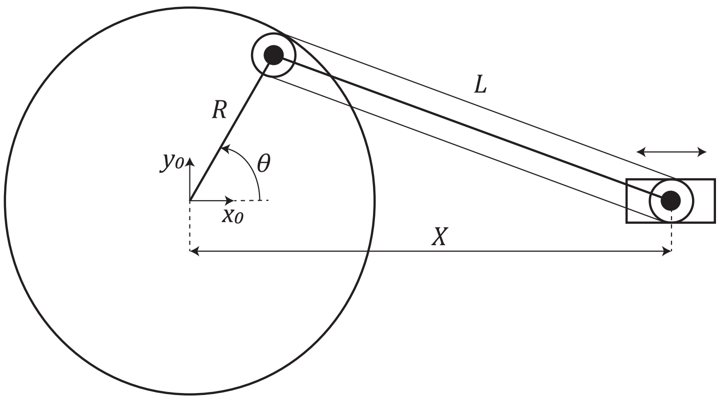

Figure 1: Slider crank mechanism with its design parameters

The reference plane of the mechanism is at x_0 and y_0, similar to the H and V axis of the CATIA sketching environment. The radius of crank is given by R, the length of connecting rod is given by L. A value of 25 and 100 can be assumed for these parameters. The forward kinematics for a mechanism provides the position of the end-effector, which in this case is the slider. Forward kinematics can be computed provided there is information about the joint angle θ. For any value of θ, write the forward kinematic equation for the mechanism.

Using the forward kinematics equation, the crank slider mechanism must be constructed and simulated in MATLAB for three complete revolutions.

Modeling guideline in MATLAB

The guidelines will have information about constructing the mechanism as well as commenting important steps. The initialization of design variables is done as provided below

Listing 1: Initialization of variables

```
1  clear all;
2  clc;
3
4  %% INITIALIZATION %%
5
6  R= 25; %Radius of crank wheel
7  L= 100; %Length of connecting rod
8  theta= 0:pi/15:6*pi; %Discretization of angle from 0 to 360 degrees
```

In Line 7, a discretization is performed which ensures that the angle is split from 0 to 6π radians with a step size of $\pi/15$ radians. This can be changed by user if necessary. The variable θ will be stored as an array and can be viewed in the workspace upon execution (click Run if the file is saved or Run & advance if not saved). This file could be saved as **Crank.m** in order to execute the code. Followed by the initialization, the forward kinematics can be estimated. For each position of θ, the value of X must be computed. The for loop can be used to extract the values of X for each discretized value of θ. An alternate way is to use the dot operator (.) of MATLAB. Write the forward kinematics equation in MATLAB which will be assigned to variable X with the dot operator or for loop.

Listing 2: Forward kinematics

```
9  %% FORWARD KINEMATICS
10 X = ......; %TO BE WRITTEN BY THE STUDENT
11 x = 0 +R.*cos(theta);
12 y = 0 +R.*sin(theta);
```

It is important to note that the dot operator precedes the mathematical operator. The dot operator is usually used when a mathematical operation is performed for an array and not a scalar. Now, the program can be executed and normally, X will also display in the workspace with its size similar to θ. After all the design parameters are defined, the mechanism can be traced. In this case, the for loop is employed to animate each instance of the mechanism for three complete revolutions.

Listing 3: Plotting the mechanism

```
13 %% PLOTTING THE MECHANISM %%
14 % A loop to trace the mechanism for all discretized points of theta
15 for i = 1 : length(x)
16
17         figure(1); %Setting a figure number is ideal
18         set(gca,'FontSize',20,'FontName','Times New Roman','FontWeight'
               ,'Bold')
19
20         % Plotting screen size setting
21         x0=50;y0=40; %Origin for the plot screen
22         largeur=650; %Length of plot screen from origin
23         hauteur=450; %Width of plot screen from origin
24         set(gcf,'units','points','position',[x0,y0,largeur,hauteur])
25
26         % Plotting the mechanism
27         h1= scatter(x(i),y(i),'b'); %Trace of angles
28
29         hold on; %Important to plot multiple traces in a single window
30
31         viscircles([0,0],R); % Tracing a circle
32
33         %Plot of connecting rod
34         h2= plot([0,x(i)],[0,y(i)],'b','Linewidth',2);
35         h3= plot([x(i),X(i)],[y(i),0],'k','Linewidth',2);
```

```
36
37         %Setting x & y limits for the plot area
38         xlim([-50,120]);
39         ylim([-50,50]);
40
41         grid on;
42         grid minor;
43
44
45         %Commands to generate animation
46         drawnow;
47         pause(0.01)
48
49         %Deleting instances for each iteration of loop
50         ......
51                 .....
52                 .....
53         ......
54
55 end
```

Copy-paste listing 3 on your MATLAB editor screen. The loop runs for the length of x. The loop can also run for length of θ as both have the same size. Lines 17 to 24 provide information about setting your plot area. For more information, the *help* syntax of MATLAB could be used in the command window to understand the significance of a function. Variable $h1$ traces the point for every revolution of crank. For constructing the crank circle, *viscircles* is used which requires a centre and radius. The syntax for the circle is provided in line 31. Lines 34 and 35 completes the rest of the mechanism. Now, the program can be executed. It could be noted that for each position, superimposed plots are obtained.

To be completed (Mandatory):

1. What is the significance of discretization? How does it affect the algorithm under study?
2. How many DOF does the mechanism have?
3. Perform the plotting step-by-step. Firstly plot only $h1$, then add $h2$ and so on.
4. What is the significance of *hold on* (Line 29) in the algorithm?
5. Complete lines 50 to 53 that deletes each instances to avoid superimposed plotting.
6. Replace lines 9 to 12 by a for loop.

Additional:

Replace the value of R by 24. A circle must be constructed at crank pin point with a diameter of 2 mm. This circle must be tangential and inscribed with the crank at all instances. With X as the origin, construct a rectangle of width 2 mm and length 10 mm. A circle of 2 mm diameter will then be inscribed within this rectangle with its centroid at X. Simulate the modified mechanism. For constructing the rectangle, the rotation matrix must be used and not the *rectangle* command of MATLAB. Using *plot* command, the rectangle can then be constructed.

6. Provide the rotation matrix calculation for the rectangle.
7. Derive the inverse kinematic equation for the mechanism.

Note: Completing the bonus might be useful for the assignment work.

Submission for grading: 19th March 13h15. Hippocampus portal. The .m file and .doc/.pdf file must be zipped and submitted as “Student name.zip”.