Animaniacs

Scientific report
Cam-follower mechanism

Matlab project

Submitted to: Dr./ Maha Hassanein

27-Apr-2019

Table of contents

Abstract

This is a scientific report about a matlab coding project of a cam-follower mechanism discussing the problem we're solving and its importance besides the method of solution & the code used .

Team introduction

A group of students from faculty of engineering-Cairo university studying in mechanical department working on a mathematical application course project for using Matlab program in a useful application .

The team members are :

- Hazem Elawa
- Abdelraman Gamal
- Tarek Elsantir
- Sherif Said
- Dina hassan
- Sarah Elganzoury
- Tony Tamer
- Islam Farag

Project objective

It’s about designing a code that calculates the dimensions and the shape of a cam that draws a certain motion desired for the follower according to the cam-follower mechanism as well as motion analysis using graphs and simulation.

(The concept will be introduced through this report.)

Importance of the project

Cam-follower systems are frequently used in all kinds of machines. The valves in our automobile engines are open by cams. Machines used in the manufacture of many consumer goods are full of cams.

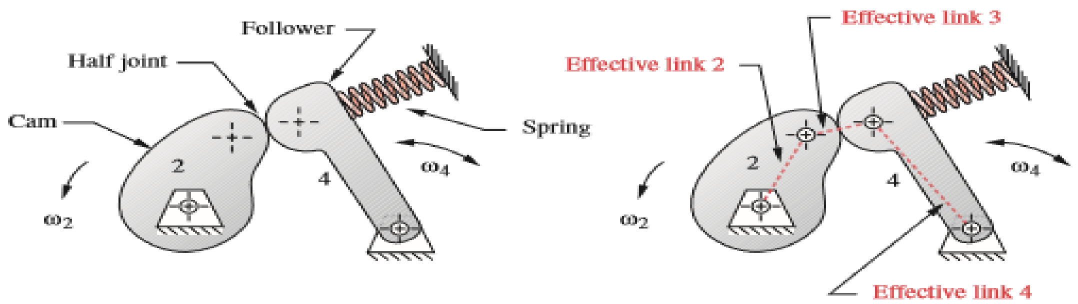
Compared to linkages, cams are easier to design to give a specific output function, but they are much more difficult and expensive to manufacture than linkages.

It is this conceptual difference that makes the cam-follower such a flexible and useful function generator. We can specify virtually any output function we desire and quite likely create a curved surface on the cam to generate that function in the motion of the follower. We are not limited to fixed-length links as we were in linkage synthesis. The cam-follower is an extremely useful mechanical device, without which the machine designer’s tasks would be more difficult to accomplish.

What is a cam?

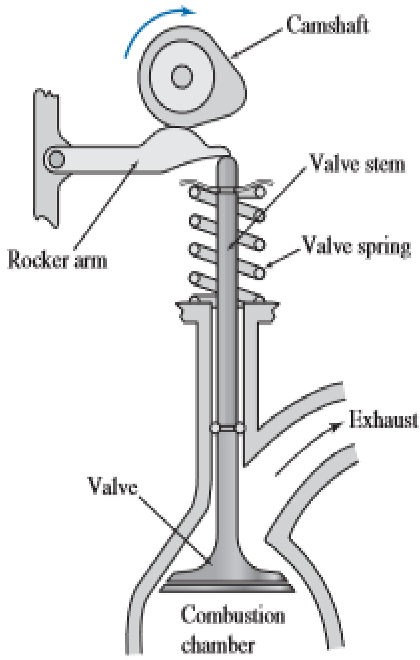
A cam is a common mechanism element that drives a mating component known as a follower. From a functional viewpoint, a cam-and-follower arrangement is very similar to the linkages connected to execute a certain motion but in compact form.

The cam accepts an input motion similar to a crank and imparts a resultant motion similar to that of a follower.



(a) An oscillating cam-follower has an effective pin-jointed fourbar equivalent

Applications :



One of the most common cam applications is the valve train of an automotive engine. In this application, an oblong-shaped cam is machined on a shaft. This camshaft is driven by the engine. As the cam rotates, a rocker arm drags on its oblong surface. The rocker arm, in turn, imparts a linear, reciprocating motion to a valve stem. The motion of the valve must be such as that the exhaust pathway is closed during a distinct portion of the combustion cycle and open during another distinct portion. Thus, the application is perfect for a cam because timing and motion must be precisely sequenced.

Cams are often used in factory automation equipment because they can sequence displacements in a cost-efficient manner.

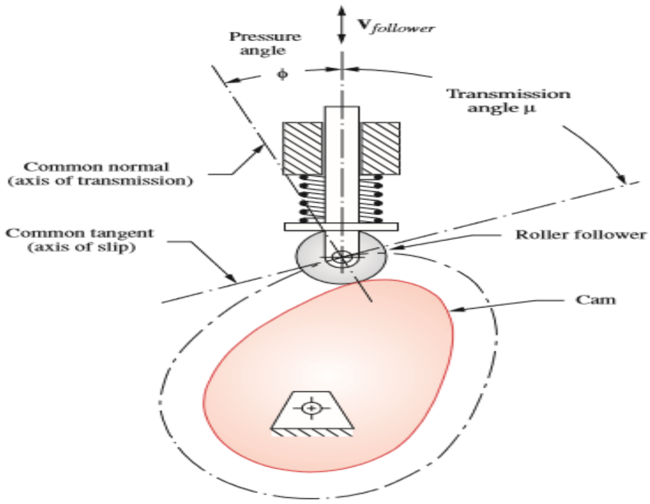
Challenges

- Distributing several tasks among the team members concerning hazy goal.
- Determining the inputs needed for a fair design.
- Understanding matlab coding language to translate the equations into a useful program.
- Determining the limitations of coding.
- Special cases of used application and assumptions needed.
- Translating the output data from the code into easy form so that people can understand and deal with.
- Creating a smart interface that clearly deals with the user guiding him through possible input mistakes

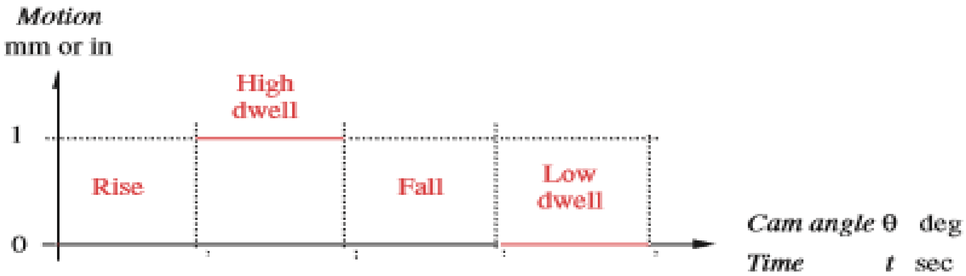
Method of solution

Description:

Regarding cam-follower mechanism design using references indexed at the end of the report , this should pass through several stages :



1) Knowing the specifications needed for the mechanism through:



- Rise angle : the angle or time range for the follower to rise for the high dwell position.

- High dwell : the angle or time range for the follower to stay in the high position.

- Fall : the angle or time range for the follower to fall from the high dwell position .

- Low dwell :the angle or time range for the follower to stay in the low dwell before cycle repetition.

- Total lift : the amount of rise or fall of the follower.

- **Min. radius of curvature** : the least radius to be used on rise or fall phase.

- **Min. pressure angle** : the least angle between the vertical line and the tangent line between the cam and the follower through the rise or fall phase.

2) SVAJ functions calculation through :

a) Equation of displacement of rise or fall phases

$$s = h\left[\frac{\theta}{\beta} - \frac{1}{2\pi}\sin\left(2\pi\frac{\theta}{\beta}\right)\right]$$

b) Equation of velocity through rise or fall phases

$$v = \frac{h}{\beta}\left[1 - \cos\left(2\pi\frac{\theta}{\beta}\right)\right]$$

c) Equation of acceleration through rise or fall phases

$$a = 2\pi\frac{h}{\beta^2}\sin\left(2\pi\frac{\theta}{\beta}\right)$$

d) Equation of jerk through rise or fall phases

$$j = 4\pi^2\frac{h}{\beta^3}\cos\left(2\pi\frac{\theta}{\beta}\right)$$

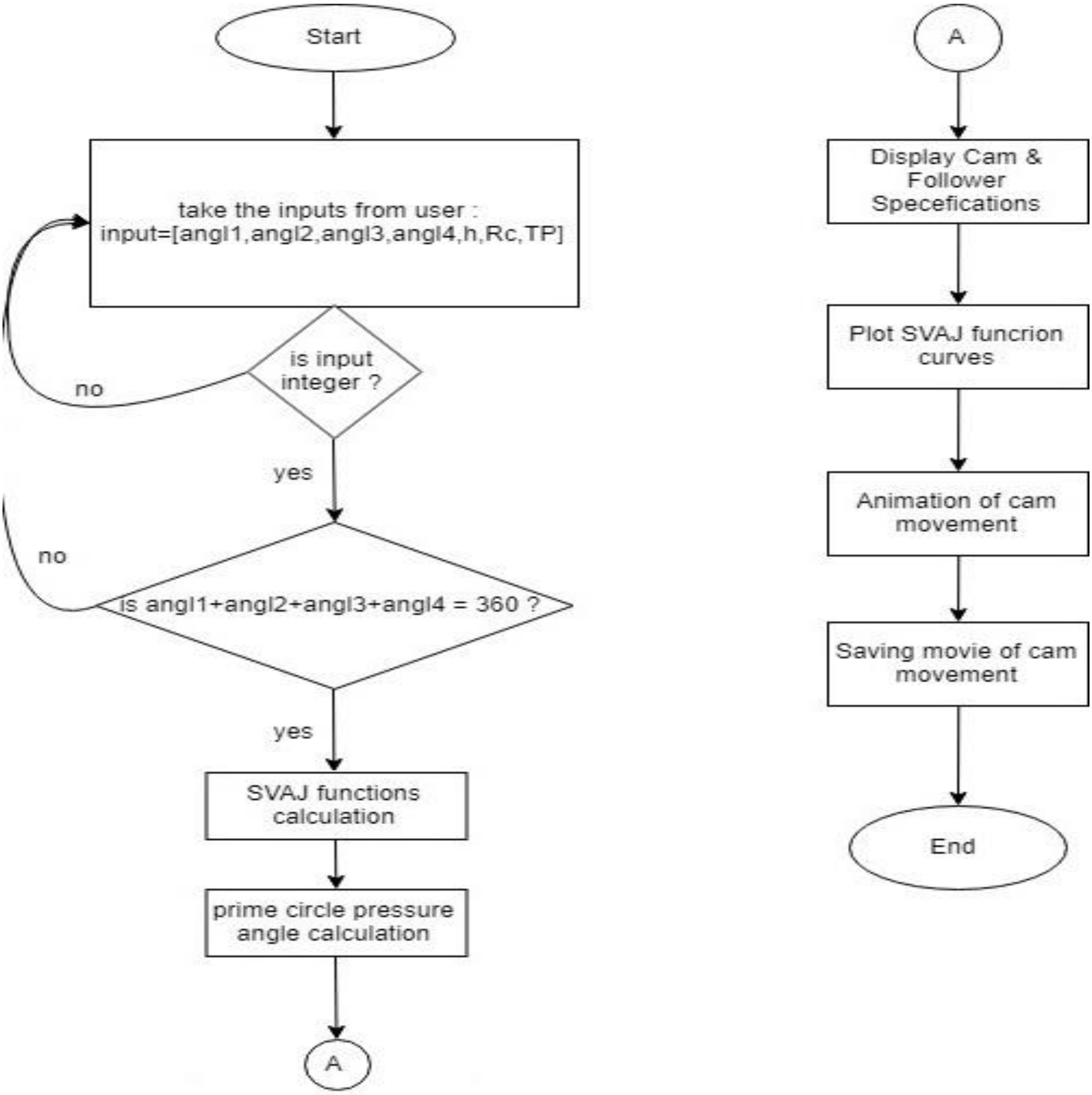
3) Prime circle radius calculation through :

$$\rho_{pitch} = \frac{\left[(R_P + s)^2 + v^2\right]^{3/2}}{(R_P + s)^2 + 2v^2 - a(R_P + s)}$$

4) Pressure angle calculation through :

$$\phi = \arctan\frac{v - \varepsilon}{s + \sqrt{R_P^2 - \varepsilon^2}}$$

Flow chart :


Start
take the inputs from user :
input=[angl1,angl2,angl3,angl4,h,Rc,TP]
is input integer ?
no
yes
is angl1+angl2+angl3+angl4 = 360 ?
no
yes
SVAJ functions calculation
prime circle pressure angle calculation
A
A
Display Cam & Follower Specefications
Plot SVAJ funcrion curves
Animation of cam movement
Saving movie of cam movement
End

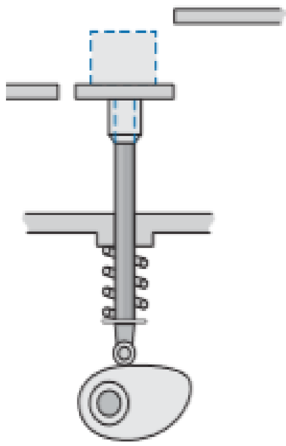
Data description

In order to make sure that the calculations are correct we took some assumptions and certain cases where the calculation are assured to be true :

- The follower is always rolling on the cam
  (No slips or jumps.)

- The follower moves linearly
  (With respect to ground.)

- Eccentricity equals **zero**
  (The follower is not shifted from the cam axis.)

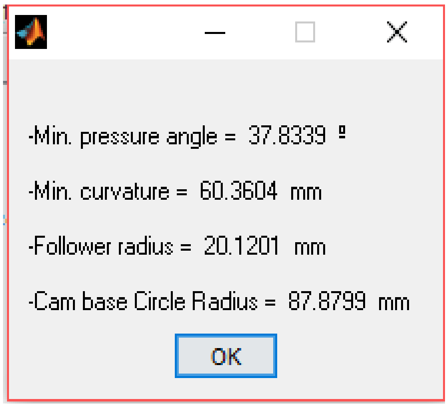


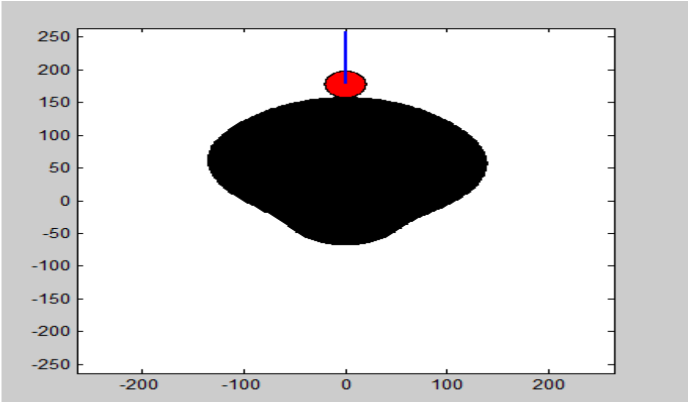
We also made sure that the user finds no difficulties using the code so we added a user interface that smartly detects any input error then prompts the user for it to avoid confusion.

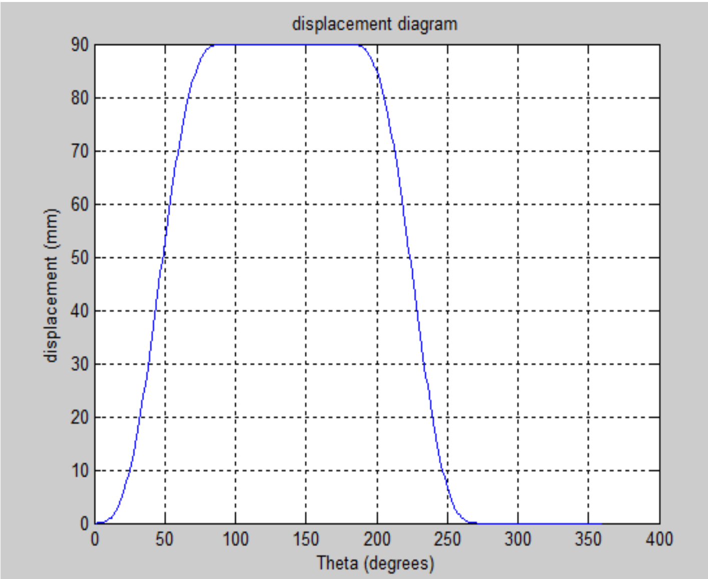
So after testing the code for multiple times using different inputs and cases , the code is able to run smoothly without any issues.

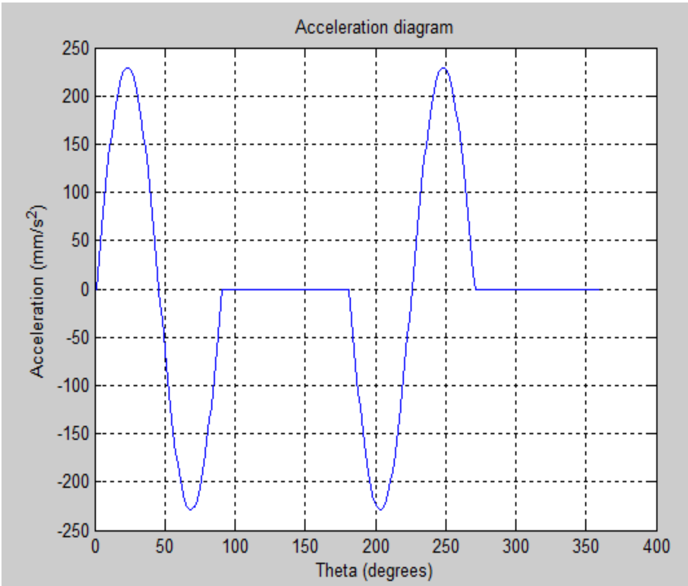
Result analysis

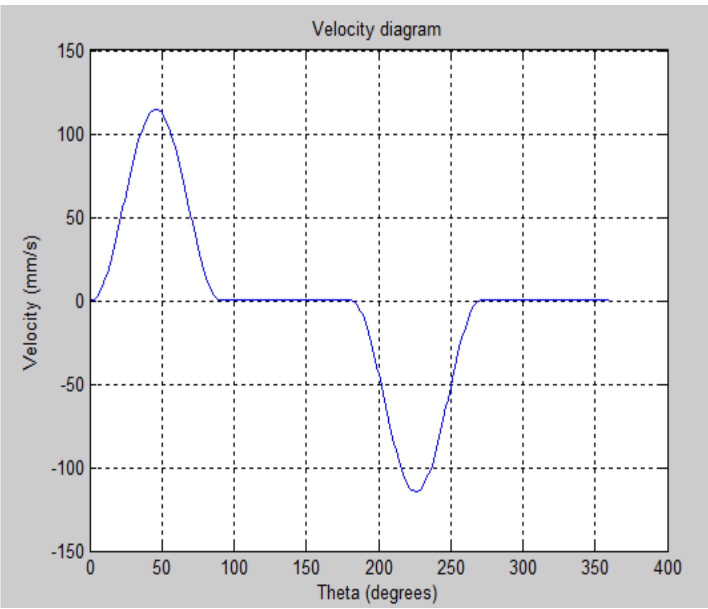
The results are introduced in a form a graphs , animations & text boxes containing the mechanism design specifications as follows:

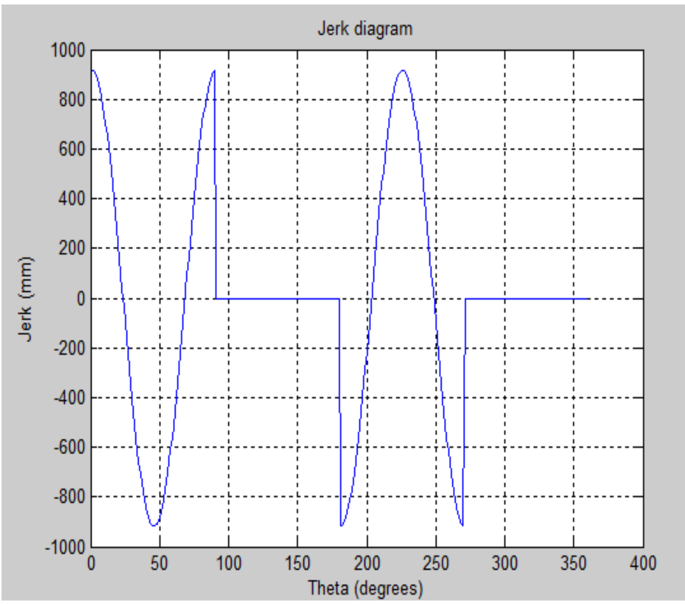












Conclusion

This simple application using MATLAB is like a leap towards cam design as it just gives accurate data in short time providing data-shows like graphs and animations that makes understanding and developing cam-follower mechanism much easier than before .

Recommendations :

when running the code , it’s recommended that the user waits for the cam animation to make a complete revolution in order to insure that the code runs completely and be able to save the animation clip successfully.

Appendix

(the Code)

```
clear;clc;
%%input interface
condition1 = true;
while(condition1)
    condition2 = true;
 while(condition2)
    input1 = {'enter angle of first segment (in degrees) :',...
    ' enter angle of second segment (in degrees) :',...
    'enter angle of third segment (in degrees) :',...
    'enter angle of fourth segment (in degrees) :',...
    'enter the lift value (in mm) :',...
    'enter the minimum radius of curvature (in mm) :',...
    'enter the minimum Pressure angle (in degrees) :'};
dlg_title1 = 'Input Table';
num_lines1 = 1;
defult_answer1={'','','','','','',''};
options.Resize= 'on';
options.windowstyle= 'normal';
answer1 =
inputdlg(input1,dlg_title1,num_lines1,defult_answer1,options);
answer1 = char(answer1);
 TH1=   str2double(answer1(1,:));
TH2=   str2double(answer1(2,:));
TH3=   str2double(answer1(3,:));
TH4 = str2double(answer1(4,:));
h=str2double(answer1(5,:));
Rc=str2double(answer1(6,:));
TP=str2double(answer1(7,:));

if(~(ismember(TH1,TH1)&&ismember(TH2,TH2)&&ismember(TH3,TH3)&&ismem
ber(TH4,TH4)&&ismember(h,h)&&ismember(Rc,Rc)&&ismember(TP,TP)))
        uiwait(msgbox('Please make sure the input is number',
'Input error','error'));
 elseif(TH1 + TH2 + TH3 + TH4 ~= 360)
        uiwait(msgbox('Please make sure the sum of segment angles =
360 °', 'Input error','error'));
 else
                 break;
 end
 end
 break;
end
e=0;
%% length of each segment
Th1 = 0:1:TH1-1;
Th2 = 0:1:TH2-1;
```

```
Th3 = 0:1:TH3-1;
Th4 = 0:1:TH4-1;

%% SVAJ functions

s1 = h*(Th1/TH1-sind(360*Th1/TH1)/2/pi);
v1 = h/(TH1/180*pi)*(1-cosd(360*Th1/TH1));
a1 = 2*pi*h/(TH1*pi/180)^2*sind(360*Th1/TH1);
j1 = 4*pi^2*h/(TH1*pi/180)^3*cosd(360*Th1/TH1);

s2 = h*ones(size(Th2));
v2 = 0*Th2;
a2 = 0*Th2;
j2 = 0*Th2;

s3 = h-(h*(Th3/TH3-sind(360*Th3/TH3)/2/pi));
v3 = -(h/TH3*180/pi*(1-cosd(360*Th3/TH3)));
a3 = -(2*pi*h/(TH3*pi/180)^2*sind(360*Th3/TH3));
j3 = -(4*pi^2*h/(TH3*pi/180)^3*cosd(360*Th3/TH3));

s4 = 0 * ones(size(Th4));
v4 = 0*Th4;
a4 = 0*Th4;
j4 = 0*Th4;

Th = 0:1:360;
S = [s1 s2 s3 s4];
V = [v1 v2 v3 v4];
A = [a1 a2 a3 a4];
J = [j1 j2 j3 j4];

%% Prime circle calculation according to pressure angle
Rp = max(sqrt((((V-e)/tand(TP))-S).^2++e^2));
test = true;
while(test)
    %curvature of pitch curve
    RawP = abs((((Rp+S).^2+V.^2).^(3/2))./((Rp+S).^2+2*V.^2-
A.*(Rp+S)));
    %calculating minimum curvature of pitch curve
    RawPMin = min(RawP);
    %calculating minimum curvature of cam body
    RawMin = 2*RawPMin/3;
    if(RawPMin < Rc)
        Rp = Rp + 1;
    else
        break;
    end
end

%% pressure angle of each segment
Phi1 = atand(v1./(s1+Rp));
```

```
Phi2 = atand(v2./(s2+Rp));
Phi3 = atand(v3./(s3+Rp));
Phi4 = atand(v4./(s4+Rp));

%% calculating maximum pressure angle of each segment
[Phimax1,place1] = max(abs(Phi1));
[Phimax2,place2] = max(abs(Phi2));
[Phimax3,place3] = max(abs(Phi3));
[Phimax4,place4] = max(abs(Phi4));
Phimax = max([Phimax1 Phimax2 Phimax3 Phimax4]);
%% calculating Rf and Rb
Rf = RawMin /3;
Rb = Rp - Rf;

%% displaying:Minimum pressure angle,minimum curvature,dimensions
of cam&follower
msgbox({['-Min. pressure angle =  ',num2str(Phimax),'  °']...
             ['   ','   ']...
             ['-Min. curvature =  ',num2str(RawMin),'  mm']...
             ['   ','   ']...
             ['-Follower radius =  ',num2str(Rf),'  mm']...
             ['   ','   ']...
             ['-Cam base Circle Radius =  ',num2str(Rb),'  mm']});
%% plotting & saving SVAJ curves

figure(); %follower displacement plotting
plot(S);
xlabel('Theta (degrees)');
ylabel('displacement (mm)');
title('displacement diagram');
grid on;

figure(); %follower velocity plotting
plot(V);
xlabel('Theta (degrees)');
ylabel('Velocity (mm/s)');
title('Velocity diagram');
grid on;

figure(); % follower acceleration plotting
plot(A);
xlabel('Theta (degrees)');
ylabel('Acceleration (mm/s^2)');
title('Acceleration diagram');
grid on;

figure(); % follower jerk plotting
plot(J);
xlabel('Theta (degrees)');
ylabel('Jerk (mm)');
title('Jerk diagram');
```

```
grid on;


%% plotting cam movement
th1 = 0:1:TH1-1;
th2 = TH1:1:(TH1+TH2)-1;
th3 = (TH1+TH2):1:(TH1+TH2+TH3)-1;
th4 = (TH1+TH2+TH3):1:(TH1+TH2+TH3+TH4)-1;

r1 = (Rb+s1).*sind(th1)+Rf*sind(Phi1-th1);
q1 = (Rb+s1).*cosd(th1)-Rf*cosd(Phi1-th1);

r2 = (Rb+s2).*sind(th2)+Rf.*sind(Phi2-th2);
q2 = (Rb+s2).*cosd(th2)-Rf.*cosd(Phi2-th2);

r3 = (Rb+s3).*sind(th3)+Rf.*sind(Phi3-th3);
q3 = (Rb+s3).*cosd(th3)-Rf.*cosd(Phi3-th3);

r4 = (Rb+s4).*sind(th4)+Rf.*sind(Phi4-th4);
q4 = (Rb+s4).*cosd(th4)-Rf.*cosd(Phi4-th4);

r = [r1 r2 r3 r4];
q = [q1 q2 q3 q4];
k=1;
figure();

for th=1:1:360
    cam = fill(r,q,'k');axis equal;hold on %cam's shape and colour
    rotate(cam,[0 0 1],th,[0 0 1]);%rotate the cam & counter with 1
degree
    Xf = e;
    Yf = sqrt((Rb + S(th))^2-e^2);
    ang=0:0.01:2*pi;
    xc=Rf*cos(ang);
    yc=Rf*sin(ang);
    fill(Xf+xc,Yf+yc,'r');%follower's shape and colour
    plot(0,0,'kx');
    plot([Xf Xf],[Yf Yf+4*Rf],'b','Linewidth',2)
    axis equal;
    axis([-3*Rb 3*Rb -3*Rb 3*Rb]);
    hold off;
 F(k)=getframe;
    k=k+1;
end
%% movie preview and saveing
movie(F,1,60); %movie(frame function,no. of repeat,fps)
%movie2avi(frame function,file name,param.,value,param.,value)
movie2avi(F,'my-cam-simulation.avi','compression','None','fps',60);
```

References

All equations & figures are inserted from the following references

- **Design of machinery** " 4th edition "
  Robert L. Norton
  https://drive.google.com/drive/folders/1mHclVv7LbkgyRHv-yDlSdeg2CBUYsLZb

- **Machines & mechanisms** " 4th edition "
  David H. Myszka
  https://drive.google.com/drive/folders/1md0uHBTi9MlOHdNOKkIr4f_Hk81JZdbe

Notations

t = time, seconds

θ = camshaft angle, degrees or radians (rad)

ω = camshaft angular velocity, rad/sec

β = total angle of any segment, rise, fall, or dwell, degrees or rad

h = total lift (rise or fall) of any one segment, length units

s or S = follower displacement, length units

$v = ds/d\theta$ = follower velocity, length/rad

$V = dS/dt$ = follower velocity, length/sec

$a = dv/d\theta$ = follower acceleration, length/rad^2

$A = dV/dt$ = follower acceleration, length/sec^2

$j = da/d\theta$ = follower jerk, length/rad^3

$J = dA/dt$ = follower jerk, length/sec^3

$s\ v\ a\ j$ refers to the group of diagrams, length units versus radians

$S\ V\ A\ J$ refers to the group of diagrams, length units versus time

R_b = base circle radius, length units

R_p = prime circle radius, length units

R_f = roller follower radius, length units

ε = eccentricity of cam-follower, length units

ϕ = pressure angle, degrees or radians

ρ = radius of curvature of cam surface, length units

ρ_{pitch} = radius of curvature of pitch curve, length units

ρ_{min} = minimum radius of curvature of pitch curve or cam surface, length units