
```

%Shishir Khanal
%ME3320
%Recitation - 3
%Date: 9/12/2021
%Purpose: Evaluate the parameters for the closing door mechanism
%Problem-2
clc;clear;close all;
%-----
% parameters of the mechanism
q = 3;
h = 4;
a = 10;
d = 8;
alpha = deg2rad(30);
theta = 0:0.01:2*pi;

% the allowable theta values based on the limits for 't'
thetamin = 0.3597;
thetamax = 2.2143;

%-----
%2.
%rootnumber can be either 1 or 2 based on the root we would like to
  explore
rootnumber = 1;

%evaluate the value of length 't' for one complete rotation of theta
t = evaluatet(h, q, theta, d, rootnumber);
xlabel = 'theta';
ylabel = 't';
plottitle = 'Plot of Joint length(t) for changing theta(0,2*pi)';
figure(1)
createplot(theta, t,xlbl, ylabel,plottitle)
hold on
%plot the boundaries for the allowable range of theta
plot(thetamin,t,'o')
hold on
plot(thetamax,t,'o')
hold off

%-----
%evaluate dependent angle 'beta(theta)'
beta = evaluatebeta(h, q,t, theta);
figure(2)
createplot(theta, beta, 'theta', 'beta', 'Plot of Dependent
  angle(beta) with respect to theta')
%plot the boundaries for the allowable range of theta
hold on
plot(thetamin,beta,'o')
hold on
plot(thetamax,beta,'o')
hold off

```

```

%-----
%3.
% generate data between the allowable region for theta
meaningfultheta = thetamin:0.01:thetamax;
meaningfulbeta = evaluatebeta(h, q,t, meaningfultheta);

% generate the position vector coordinates for P
[Px, Py] = evaluateP(a, alpha, meaningfulbeta);
figure(3)
createplot(Px,Py, 'Px', 'Py', 'Plot of the Position vector with
    respect to allowable theta')
hold on
plot(0,0,'*r')
%-----
%Functions:
function [t] = evaluatet(h, q, theta, d, rootnumber)
%evaluatet() takes in the length arguments and the theta vector and
    outputs
% the 't' vector
% Detailed explanation goes here
t = zeros(length(theta),1);
    for i= 1: length(theta)
        A = 1;
        B = 2*(q*cos(theta(i)) - h*sin(theta(i)));
        C = q^2 + h^2 - d^2;
        % t_total = roots([A B C]);
        %the roots are not put in the same order in this command
        % t = t_total(rootnumber);
        if rootnumber == 1
            t(i) = (-B + sqrt(B^2 - 4 * C)) / 2;
        elseif rootnumber == 2
            t(i) = (-B - sqrt(B^2 - 4 * C)) / 2;
        end
    end
end

function [beta] = evaluatebeta(h, q,t, theta)
%evaluatet() takes in the length arguments and the theta vector and
    outputs
% the 't' vector
% Detailed explanation goes here
beta = zeros(length(theta),1);
    for i= 1: length(theta)
        num = -h + t(i)*sin(theta(i));
        den = q + t(i)*cos(theta(i));
        beta(i) = atan2(num, den);
    end
end

function createplot(xparam, yparam,xlbl, ylbl,plottitle)
%createplot() doesn't output anything but creates a formatted plot of
%parameters based on the given parameters
% Detailed explanation goes here

```

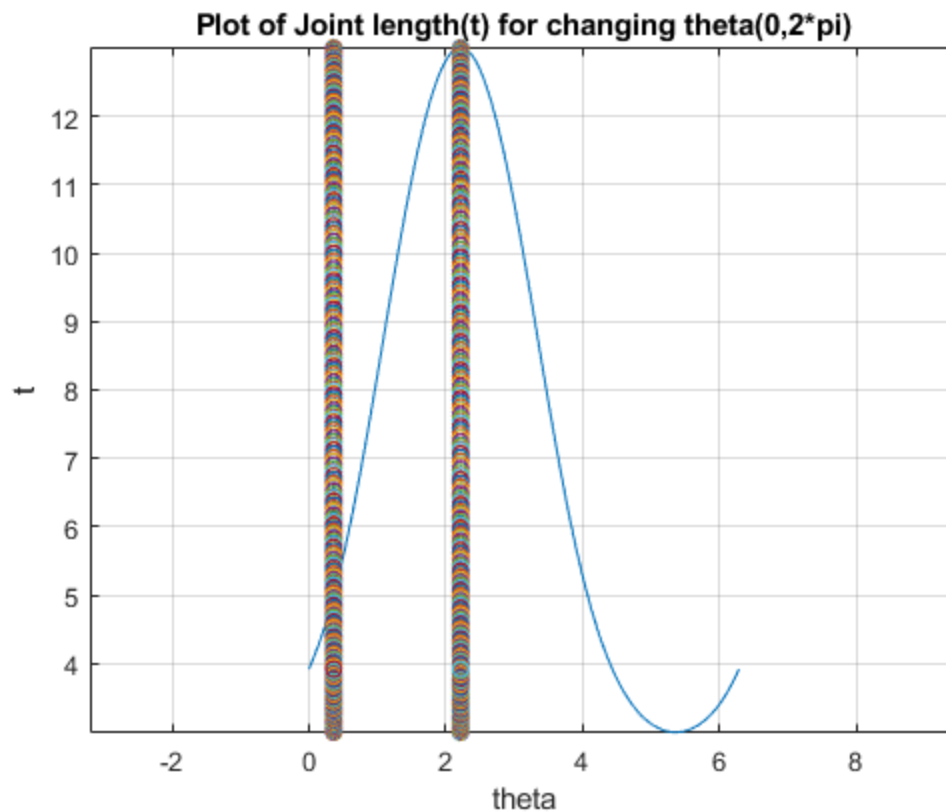
```

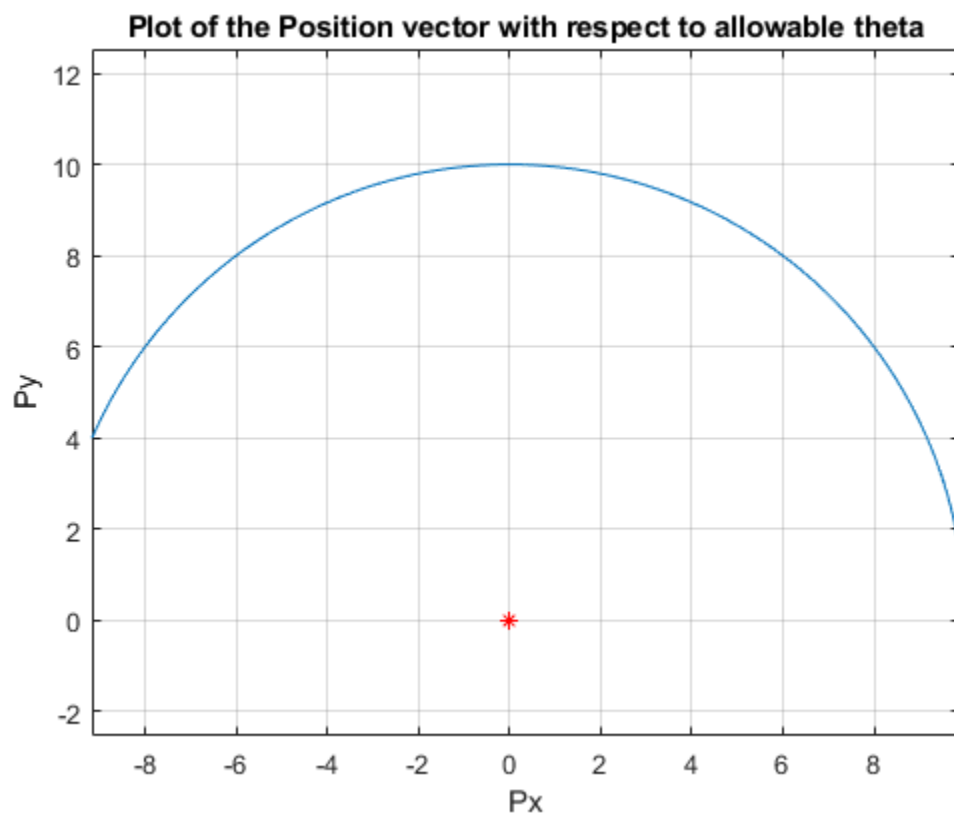
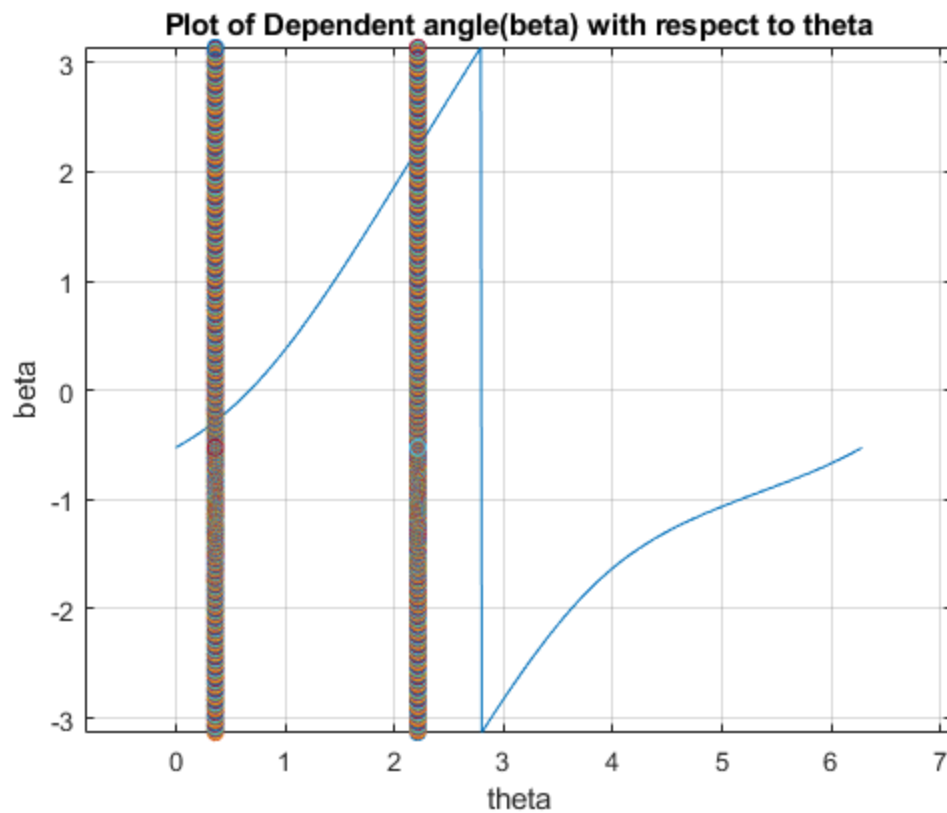
plot(xparam,yparam)
xlabel(xlbl)
ylabel(ylbl)
title(plottitle)
grid on;
axis equal
end

function [Px, Py] = evaluateP(a, alpha, beta)
%evaluateP() takes in the length arguments and the theta vector and
% outputs
% the 't' vector
% Detailed explanation goes here
Px = zeros(length(beta),1);
Py = zeros(length(beta),1);
for i= 1: length(beta)
    Px(i) = a*cos(alpha + beta(i));
    Py(i) = a*sin(alpha + beta(i));
end
end

%-----

```





Published with MATLAB® R2019b