

```

function Xdot = test_splines(t,X);
% Simulation of the control of a Two-Link Planar robot
% using Computed-Torque Control
%
% Course: Robotics
% Written by Jose Mireles Jr.
% X(1)= theta1
% X(2)= theta2
% X(3) = (d/dt)theta1
% X(4) = (d/dt)theta2
% X(5) = tracking error 1 (Only for Output purposes)
% X(6) = tracking error 2 (Only for Output purposes)

% Constant values for the desired trajectory:
Amp1=0.1; Amp2=0.1;

% Constant values of the controller:
kp=100; kv=20; ki=500; % Ki < Kp*Kv

% Constant values of the Robot:
m1=1; m2=1; a1=1; a2=1; g=9.8;

% PART 1) COMPUTE DESIRED TRAJECTORY qd(t), qdp(t), qdpp(t)

%A)
a1 = 1;
a2 = 1;
% desired path
x=linspace(2,0,8);
y=linspace(0,2,8);
% D = cos(theta2)
D = (x.*x + y.*y - a1^2 - a2^2)/(2 * a1 * a2);
% theta2 (rads)
theta2 = (atan( sqrt( 1 - D.*D)./(D+0.000001))));
% theta1 (deg)
theta1 = (atan(y./(x+0.0000001)) - atan( (a2 * sin(theta2)) ./ (a1 + a2 * cos(theta2)) ));
x=linspace(-.1,-2,8);
y=linspace(1.9,0,8);
D = (x.*x + y.*y - a1^2 - a2^2)/(2 * a1 * a2);
% theta2 (rads)
theta2b = (atan( sqrt( 1 - D.*D)./(-D-0.000001))));
% theta1 (deg)
theta1b = (atan(y./(x-0.0000001)) - atan( (a2 * sin(theta2b)) ./ (a1 + a2 * cos(theta2b)) ))+pi;
%theta1b = (atan(y./(x-0.0000001)) - atan( (a2 * sin(theta2b)) ./ (a1 + a2 * cos(theta2b)) ));
x = [theta1, theta1b];
y = [theta2, theta2b];

```

```

% Time runs from 0-6 seconds !!
DeltaT = 6/100;
LowInterval = floor(t/DeltaT)+1;
time = (t/DeltaT)-(LowInterval-1);
HighInterval = LowInterval + 1;

[x , y] = splines2(x,y,7); % 106 or 117 total points !!
[Xd, Yd] = slopes(x,y);
qd(1) = Value(x(LowInterval),x(HighInterval),Xd(LowInterval),Xd(HighInterval),time);
qd(2) = Value(y(LowInterval),y(HighInterval),Yd(LowInterval),Yd(HighInterval),time);

Xd = Xd*100/(6); % Consider only 100 samples for six seconds !!
Yd = Yd*100/(6);
clear x;
clear y;

[Xdd, Ydd] = slopes(Xd,Yd);
qdp(1) = Value(Xd(LowInterval),Xd(HighInterval),Xdd(LowInterval),Xdd(HighInterval),time);
qdp(2) = Value(Yd(LowInterval),Yd(HighInterval),Ydd(LowInterval),Ydd(HighInterval),time);

Xdd = Xdd*100/6; % Consider only 100 samples for six seconds !!
Ydd= Ydd*100/6;
clear Xd;
clear Yd;

[Xddd, Yddd] = slopes(Xdd,Ydd);

qdpp(1) = Value(Xdd(LowInterval),Xdd(HighInterval),Xddd(LowInterval),Xddd(HighInterval),time);
qdpp(2) = Value(Ydd(LowInterval),Ydd(HighInterval),Yddd(LowInterval),Yddd(HighInterval),time);
clear Xdd;
clear Ydd;
clear Xddd;
clear Yddd;

% PART 2) COMPUTED-TORQUE CONTROLLER SUBROUTINE

% COMPUTE TRACKING ERRORS:
e(1) = qd(1)-X(1);
e(2) = qd(2)-X(2);
ep(1) = qdp(1)-X(3);
ep(2) = qdp(2)-X(4);

% COMPUTATION OF M(q)
m11=(m1+m2)*a1^2+m2*a2^2+2*m2*a1*a2*cos(X(2));
m12=m2*a2^2+m2*a1*a2*cos(X(2));

```

```

m21=m12;
m22=m2*a2^2;

% COMPUTATION OF N(q,qp)
N1 = -m2*a1*a2*(2*X(3)*X(4)+X(4)^2)*sin(X(2));
N2 = m2*a1*a2*X(3)^2*sin(X(2));
N1 = N1 + (m1+m2)*g*a1*cos(X(1))+m2*g*a2*cos(X(1)+X(2));
N2 = N2 + m2*g*a2*cos(X(1)+X(2));

% COMPUTATION OF CONTROL TORQUES
s1 = qdpp(1) + kv*ep(1) + kp*e(1);
s2 = qdpp(2) + kv*ep(2) + kp*e(2);
f1 = m11*s1 + m12*s2 + N1;
f2 = m21*s1 + m22*s2 + N2;

% Adding these two forces to the plot:
%figure(6)
%hold on
%plot(t,f1,'b',t,f2,'r')

% PART 3) ROBOT ARM DYNAMICS:

Det = (m11*m22)-(m12*m21);

% Inverse Matrix
Im11=m22/Det;
Im12=-m12/Det;
Im21=-m21/Det;
Im22=m11/Det;

% STATE EQUATIONS:
% X(1) = theta1
% X(2) = theta2
% X(3) = (d/dt)theta1
% X(4) = (d/dt)theta2
% X(5) = tracking error 1 (Only for Output purposes)
% X(6) = tracking error 2 (Only for Output purposes)

Xdot=[ X(3);
        X(4);
        (Im11*(f1-N1)+Im12*(f2-N2));
        (Im21*(f1-N1)+Im22*(f2-N2));
        (e(1)-X(5))*6/100;
        (e(1)-X(6))*6/100;
];

```