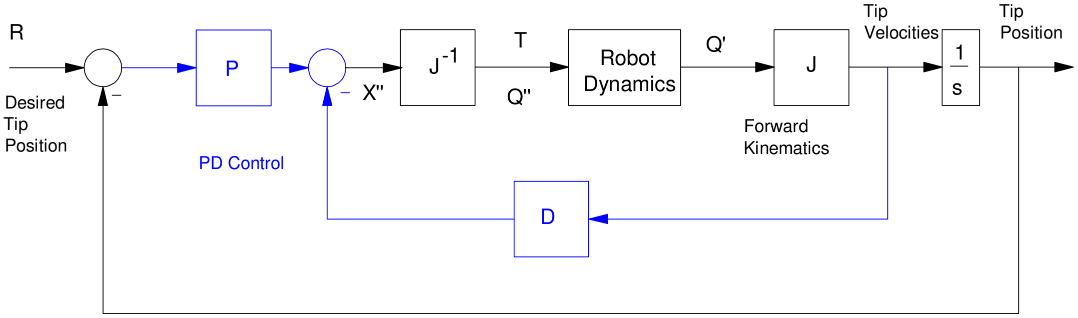
ECE 761: Homework #12: Jacobians and Cartesian Control



For a 2-link arm, assume the tip position is

P2 = (1.1, 0.5)

1) Determine the joint angles at this point

2) Determine the Jacobian at this point

3) Determine the joint velocities if the tip velocity is moving towards point 3:

$$\begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} \quad \text{m/s}$$

4) Determine the tip velocity if the joint velocity is

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} \quad \text{rad/sec}$$

Write a program (modify RR_XY_Control.txt) to trace out a square with corners at

• P0 = (0.1, -0.5)	t = 0 seconds
• P1 = (1.1, -0.5)	t = 4 seconds
• P2 = (1.1, +0.5)	t = 8 seconds
• P3 = (0.1, 0.5)	t = 12 seconds
• P4 = P0 = (0.1, -0.5)	t = 16 seconds

5) Using cosine interpolation between points, and

6) Using no interpolation (step change: a = 1)

```
P0 = [0.5; 0];
P1 = [1.5; 0];
P2 = [1.5; 1];
P3 = [0.5; 1];
P4 = P0;

t = [0:0.01:4.99];
a = t/5;
a = (1 - cos(pi*a))/2;
% for a step change in poistion, add the following line
a = 1*(a>0);

XY1 = P0*(1-a) + P1*a;
XY2 = P1*(1-a) + P2*a;
XY3 = P2*(1-a) + P3*a;
XY4 = P3*(1-a) + P4*a;

XYr = [XY1, XY2, XY3, XY4];
TIP = XYr;


Q = InverseRR(XYr(:,1));
dQ = [0; 0];

t = 0;
dt = 0.001;

% Start the simulation  (dt = 0.001 for stability concerns)

XYq = [];

for i=1:length(XYr)

   Qr = InverseRR(XYr(:,i));

   for j=1:10

      XY = [ cos(Q(1)) + cos(Q(1)+Q(2));
              sin(Q(1)) + sin(Q(1)+Q(2)) ];

      J = [ -(sin(Q(1)) + sin(Q(1)+Q(2))), -sin(Q(1)+Q(2)) ;
             (cos(Q(1)) + cos(Q(1)+Q(2))),  cos(Q(1)+Q(2)) ];

      dXY = J * dQ;

% second derivative terms the Jacobian
      Cxy = [ cos(Q(1))* (dQ(1)^2)  +  cos(Q(1)+Q(2)) * (dQ(1) + dQ(2))^2 ;
               sin(Q(1))* (dQ(1)^2)  +  sin(Q(1)+Q(2)) * (dQ(1) + dQ(2))^2];

      T1 = inv(J) * ( 16*(XYr(:,i) - XY)  + Cxy )     - 6*dQ;

% gravity
      Tg = -9.8 * [ 2*cos(Q(1)) + cos(Q(1) + Q(2));
                    cos(Q(1) + Q(2)) ];

      T = T1 - Tg;

      ddQ = RRDynamics(Q, dQ, T);

      dQ = dQ + ddQ * dt;
      Q = Q + dQ*dt;
      t = t + dt;

      end

   RR(Q, Qr, TIP);
   XYq = [XYq, XY];

   end


t = [1:length(XYr)] * 0.01;
clf
plot(t,XYq,t,XYr);
xlabel('Time (seconds)');
ylabel('Tip  (meters)');
```