MATLAB Assignment

October 5, 2010

Problem 1. Implement a set of MATLAB routines that compute for a generic revolute joint all the information needed to carry out Kinematics Analysis. Specifically, this set of MATLAB routines should be able to evaluate the expression of the algebraic equations defining the joint; i.e., evaluate $\mathbf{\Phi}(\mathbf{q}_i, \mathbf{q}_j)$, $\mathbf{\Phi}_\mathbf{q}$, ν, and γ. The code might look something like this:

```
function [phiVal,Phi_q,nuVal,gammaVal]=revoluteJ(qi,qiDot,sBarPi,qj,qjDot,sBarPj,flagC)
% Computes Kinematics Analysis required quantities associated with the
% presenence of a revolute joint.
% Based on the value of the flagC, one could compute only phiVal; i.e., the
% violation of the algebraic constraint, or Phi_q (the sensitivity matrix; i.e.,
% the Jacobian), or nuVal, the right hand side of the Velocity Equation,
% or gammaVal, the right hand side of the Acceleration Equation

if flagC == 1
    % compute the expression of phiVal only
    r_i = qi(1:2,1);
    phi_i = qi(3,1);
    r_j = qj(1:2,1);
    phi_j = qj(3,1);
    % use r_i,..., phi_j to compute phiVal below...
elseif flagC == 2
    % compute the expression of phiVal only
    % code here to compute a matrix that is 2 by 6
    % The first 2X3 is sensitivity wrt qi, the last 2X3 is wrt qj
elseif flagC == 3
    % compute the expression of nu; this is easy
    nuVal(1) = 0;
    nuVal(2) = 0;
elseif flagC == 4
    % compute the expression of gamma; this is messy
    % needs to use also time derivatives qiDot and qjDot
```

```
    code comes here...
end
```

Problem 2. Like for problem 1, yet this time implement a similar set of MATLAB routines for the translational joint.

Problem 3. Like for problem 1, yet this time implement a similar set of MATLAB routines for the distance constraint.