Program Assignment 3 (Due March 12, 2015)

1. Write a joint space, cubic splined trajectory planner.

   One subroutine that you system should include is

   Procedure CUBCOEF(VAR qs, qg, qdots, qdotg, T: real; Var cc: vec4);

   where the input parameters are:

   - qs = the position of q_i at the beginning of a path segment;
   - qg = the position of q_i at the end of the path segment;
   - qdots = velocity at the beginning of the path segment;
   - qdotg = velocity at the end of the path segment;
   - T = the time duration of the segment.

   The output "cc" is an array of 4 cubic coefficients.

   Suppose that your manipulator is the 3-D planar one used for Program #2. The input of your trajectory planner, which is itself a procedure, should be a path with up to five knot points, where each knot point is specified in the usual form of (x, y, ϕ). Clearly, given a path, your program needs to call the inverse kinematics procedure(s) that you did in Program #2 to convert the knot points into joint space representations. Your system can assume that the initial velocity and the final velocity of the joints equal to zero. It should accept user-specified time durations for each path segment. Then, it should use the simple heuristic described in the Handout #3 to specify the joint velocities at the via points. The output of your planner should be the coefficients of cubic polynomials for all the path segments.

2. Write a trajectory generator which calculates the joint position, velocity and acceleration at a user-specified frequency r for all the joints.

   One subroutine the system should include is

   Procedure TRAJEC(VAR t: real; VAR cc: vec4; VAR q, qdot, qddot: real);

where the input parameters are: the time "t" and the cubic coefficients (an array of 4 elements) "cc". The output parameters "q", "qdot", and "qddot" are $q_i(t)$, $\dot{q}_i(t)$, and $\ddot{q}_i(t)$ respectively for the joint q_i.

3. Write a program to test your trajectory planner and generator. Use the following data as input:

   - a path with $n = 4$ knot points starting at $(x, y, \phi) = (0.758, 0.173, 0.0)$, moving through the via points $(0.6, -0.3, 45.0)$ and $(-0.4, 0.3, 120)$, and ending at $(0.758, 0.173, 0.0)$ (which is the same as the starting point in this case)
   - the time duration of each path segment is $T = 3.0$ seconds.
   - the frequency for trajectory calculation is $r = 5$Hz (which means that you need to calculate the trajectory every 0.2 seconds and you need an array of $(n-1) * T * r = 45$ elements to store the calculated trajectory.)

   Your program should first call the trajectory planner and then the trajectory generator. Finally, the program should call the forward kinematics routines for all the trajectory points at $t = 0, 1, 2, ..., 9$ seconds to find the corresponding (x, y, ϕ) representations. The output should be the printout of all these (x, y, ϕ) values.

Attention: **Turn in a hard copy of the well-documented source codes of your program and a well-formatted printout of your output**.