Differential (Forward) Kinematics

This project will expand upon your program that computes direct/forward (position) kinematics and displays a manipulator in three dimensions with DH frames. The ability to compute and display differential kinematics will be added.

1. Implement the Jacobian via the direct method along with the ability to compute and display the end-effector's (frame n's) linear and angular velocity.

   The program should take the following as inputs:

   (a) DH table with parameters, and

   (b) joint variables $\vec{q}$ along with their derivatives (joint velocities) $\dot{\vec{q}}$ keeping in mind we will likely use a sequence of them in the future for animation.

   The program should output or show:

   (a) homogeneous transformation matrix representing the end-effector's pose;

   (b) Jacobian that maps joint velocities to end-effector velocities;

   (c) end-effector's velocity (linear and angular);

   (d) 3D visualization of robot using lines, cylinders or fancier objects to represent joints and links (see figure below as simple example);

   (e) display of frames 0 to n (with ability to turn them on/off); and

   (f) display of linear and angular velocity vectors at/for the end-effector (with ability to turn them on/off).

2. Implement the following robots (with details provided in handout) in your program: Stanford Arm, Puma 260, Adept SCARA, and planar RRP. Note it should be easy to implement a new manipulator through specification of its DH table, and joint variables and velocities.

3. Test your forward kinematics and visualizations for a variety of values of joint variables and velocities including moving joints with time-varying functions such as $q_i = 1 + \sin(t)$, $\dot{q}_i = \cos(t)$.

4. Turn in results (specifically, values of the Jacobian and end-effector's velocity plus 3D visualization of manipulator with end-effector's velocity shown) for the cases when all $q_i = \dot{q}_i = 0$, all $q_i = \dot{q}_i = 1$, and all $q_i = \dot{q}_i = -1$.