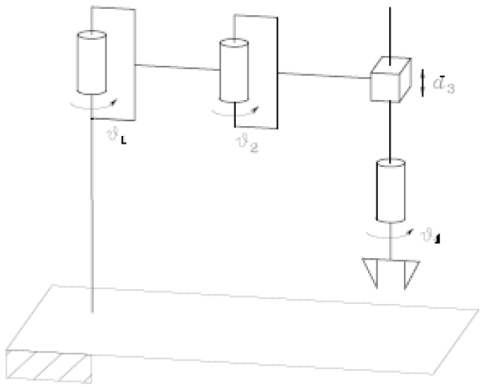
ECE-5223 Sensor-based Robotics
ME-7933 Simulation Tools and Software for Mechatronics and Robotics **Project 1**
New York University, Spring 2019 updated October 11, 2019
Due on October 22, 11:59 PM

Direct and Inverse Kinematics (100 points)

Consider the SCARA manipulator depicted below. Please be aware that not all the parameters are needed at this stage. You will use some of them project 2, which will involve dynamics. For this project only the kinematic parameters are needed. You have received a trajectory for the the manipulator end effector. The trajectory is provided in a file named Kinematic_traj.m and can be read using init.m.



The manipulator parameters are

$$d_0 = 1\ m,\ a_1 = a_2 = 0.5\ m$$

$$\theta_{1_{min}} = -\pi/2\ rad,\ \theta_{1_{max}} = \pi/2\ rad,\ \theta_{2_{min}} = -\pi/2\ rad,\ \theta_{2_{max}} = \pi/4\ rad$$

$$d_{3_{min}} = 0.25\ m,\ d_{3_{max}} = 1\ m,\ \theta_{4_{min}} = -2\pi\ rad,\ \theta_{4_{max}} = 2\pi\ rad$$

Questions:

- Implement in Matlab/Simulink the algorithms for kinematic inversion with inverse and jacobian transpose along the given trajectory. Adopt the Euler integration rule with integration time 1ms.
- Suppose to relax one component in the operational space, implement in Matlab/Simulink the algorithm for kinematic inversion with Jacobian pseudo-inverse along the given trajectory maximizing in two separate cases
  (a) the distance from the mechanical joint limits (relax the orientation component ϕ)
  (b) the distance from an obstacle along the path (relax the z component). Suppose that the obstacle is a sphere centered in $p = \begin{bmatrix}0.4 & -0.7 & 0.5\end{bmatrix}^\top$ of radius 0.2 m.

Instructions:

- Make your code as a combination of matlab and simulink. You should call your initialization in a function named init.m. This function should load the trajectory and all the manipulator variables that have been previously listed. You will then define your Jacobians in another function named Jacobian.m, which will be loaded in simulink as shown during the class
- Make a different folder of each question. Once init.m is ran in each folder, we should be able to automatically play the simulink environment and obtain results. To visualize them implement a function in matlab visualize_results.m for each question. Show the joint trajectories in the joint space and the errors operational space. The final report should be made according to the template.