EECS4421/5324: Lab 4

Thu Feb 8, 2018
Due: In class Fri Mar 2, 2018

You will require your inverse kinematics solutions from Lab 2 to complete this lab. Consider creating a method that solves the inverse kinematics problem for the arm, rather than putting all of the inverse kinematics code inside of your solutions to Questions 1 and 2.

1. (Everybody)
In the Matlab simulator for the A150 robot implement the method with signature `move(obj, T)` that takes as input a goal pose `T` (expressed in the base frame of the robot; i.e., `T` $= T_5^0$); the function should then move frame 5 to the goal pose `T` using a cubic polynomial joint space path for each joint variable θ_i^* (where $i = 1, 2, ..., 5$). If `T` is not reachable then the robot should not move from its current pose and an error message should be output indicating that `T` is not reachable.

To compute each polynomial joint space path, assume that the velocity of each joint is 0 radians$/s$ at the start of the path and at the end of the path, the value of θ_i^* at the start of the path is obtained from current pose of the robot (using `getpose`), and the value of θ_i^* at the end of the path is obtained using `T`. Also, assume that the time $t = 0$ at the start of the path and $t = 1$ the end of the path.

To produce the motion of the robot, evaluate the polynomial path for each joint variable at $t = 0, 0.1, 0.2, ..., 1$ and set all 5 joint angles using `dhmadeg` at $t = 0.1, 0.2, ..., 1$.

2. (Undergraduate students only)
In the Matlab simulator for the A150 robot implement the method with signature `movelinear(obj, p)` that takes as input a goal location `p` (expressed in the base frame of the robot; i.e., `p` $= p_5^0$); the function should then move the origin of frame 5 to the input location in a straight line, or output a message indicating that `p` is not reachable. You should assume that the orientation of the end effector is constant and equal to

$$R_5^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

i.e., the gripper is always pointing straight down; this constraint means that you do not need to interpolate the orientation of the end effector.

The path of the end effector should be a straight line. If a straight line path is not possible then the robot should complete as much of the straight line path as is possible; i.e., it should move from its current position towards the goal in a straight line until it can go no further and then stop (see the next paragraph for what "no further" means).

You should compute the number of steps along the path so that the end effector moves by no more than 5 cm at each step. If taking the next step in the path would cause the robot to move to an unreachable position then the robot should not take the next step.

2. (Graduate students only)
In the Matlab simulator for the A150 robot implement the method with signature `movelinear(obj, T)` that takes as input a goal pose `T` (expressed in the base frame of the robot; i.e., $\texttt{T} = T_5^0$); the function should then move frame 5 to the goal pose `T` in a straight line, or output a message indicating that `p` is not reachable. Furthermore, the orientation of the end effector should change smoothly as the robot moves along the path.

To solve this problem, you need to find a way to interpolate the orientation of the end effector along the path. To perform the interpolation, implement a quaternion slerp; your implementation should include functions to convert a rotation matrix to a quaternion, convert a quaternion to a rotation matrix, and perform the slerp operation.

The path of the end effector should be a straight line. If a straight line path is not possible then the robot should complete as much of the straight line path as is possible; i.e., it should move from its current position towards the goal in a straight line until it can go no further and then stop (see the next paragraph for what ”no further” means).

You should compute the number of steps along the path so that the end effector moves by no more than 5 cm at each step. If taking the next step in the path would cause the robot to move to an unreachable position then the robot should not take the next step.

1 Submit

Submit the Matlab files needed to run your solution:

```
submit 4421 lab3 <your files>
```