CS4610/CS5335: Homework 1

Out: 1/24/2019, Due: 2/5/2019

Please turn in this homework via email by the due date. HW Q1 - Q5 should be submitted as a PDF. HW PA Q1-Q5 should be submitted in the form of a set of five files named Q1.m ... Q5.m. All this should be zipped up into a single file and emailed to the TAs.

Have a look at the accompanying zip file. Stub files for Q1.m ... Q5.m are provided to you. You should implement each of these. Once implemented, you should be able to run "hw1(X)" in order to run code for question "X". hw1.m is given to you and should not need to be modified. The only thing you need to do is to insert code into the stub functions in Q1.m .. Q5.m.

HW Q1: (Spong, Problem 2-15) If the coordinate frame A is obtained from the coordinate frame B by a rotation of $\pi/2$ about the x-axis followed by a rotation of $\pi/2$ about the fixed y-axis, find the rotation matrix R representing the composite transformation. Sketch the initial and final frames.

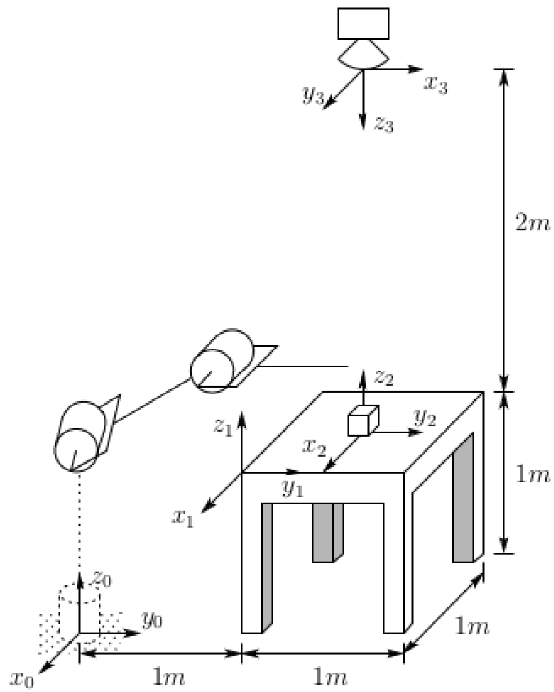


Figure 1:

HW Q2: (Spong, Problem 2-37) Consider the diagram in Figure 1. A robot is set up 1 meter from a table. The table top is 1 meter high and 1 meter square. A frame $o_1\ x_1, y_1, z_1$ is fixed to the edge of the table as shown. A cube measuring 20 cm on a side is placed on top of the table and at the center of the table with frame $o_2\ x_2, y_2, z_2$ established at the center of the cube as shown. A camera is situated directly above the center of the block 2m above the table top with frame $o_3\ x_3, y_3, z_3$ attached as shown. Find the homogeneous transformations relating each of these frames to the base frame $o_0\ x_0, y_0, z_0$. Find the homogeneous transformation relating the frame $o_2\ x_2, y_2, z_2$ to the camera frame $o_3\ x_3, y_3, z_3$.

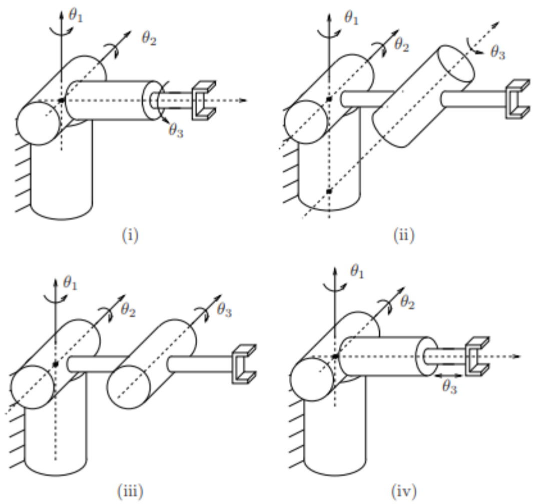


Figure 2:

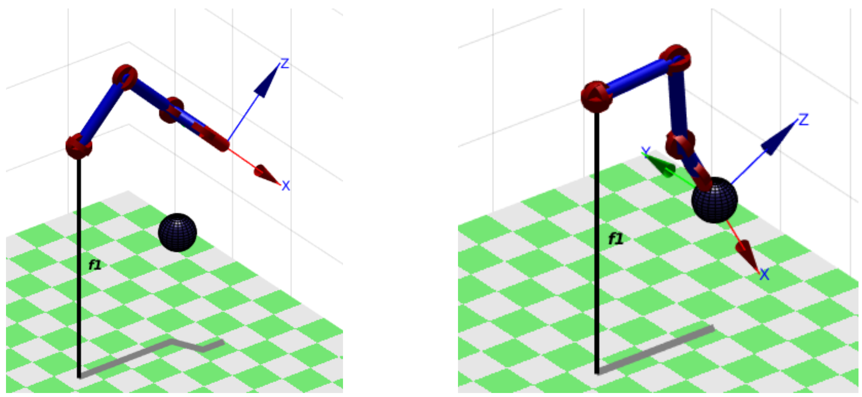
HW Q3: For each of the three degree of freedom manipulators shown in Figure 2, find the forward kinematics map. Assume that the link lengths are given to you in as l_1, l_2, l_3, etc. Specifically, in (i), l_1 is the vertical height of the first joint and l_2 is the length of the arm segment after $\theta_{1,2}$. In (ii) and (iii), l_1 is the vertical height of the first joint, l_2 is the length of the arm segment after $\theta_{1,2}$ and before θ_3, and l_3 is the length of the arm after θ_3. In (iv), the first two link lengths are as defined above. The last link length is determined by the translational joint, θ_3, which determines the additional amount by which the hand extends beyond the l_2 part of the arm.

HW Q4: Let J (θ) : $R^n \rightarrow R^6$ be the Jacobian of a manipulator. Show that the manipulability measure $\mu = \sqrt{det(JJ^T)}$ is given by the product of the singular values J(θ); that is,

$$\mu = \prod_{i=1}^{6} \sigma_i(\theta).$$

Thus, $\mu_3(\theta)$ is zero if and only if the Jacobian is singular.

HW Q5: A point in a manipulator's workspace is said to be isotropic if the condition number (the ratio of the largest to smallest singular values) of the Jacobian is 1.
(a) Calculate conditions under which a two-link planar manipulator has isotropic points and sketch their location in the plane.
(b) Discuss why isotropic points are useful for tasks which involve applying forces against the environment.



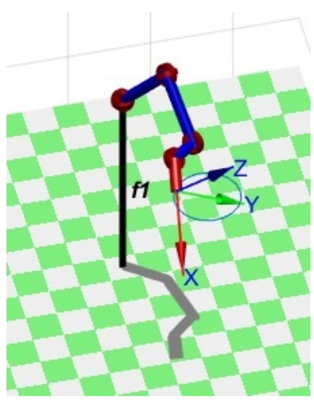
(a) (b)

Figure 3: Illustration of Q1. (a) is before moving the arm. (b) is after moving the arm to the configuration calculated in your function.

PA Q1: Implement the function in Q1.m. This function will use the built-in inverse kinematics function in the RTB to calculate a joint configuration that corresponds to a desired end effector position (just position, not orientation). The function will take as input a robot (encoded as a SerialLink class) and a desired position (encoded as a 3x1 vector). It will calculate a target joint configuration that will cause the end of the robot arm to reach a point at the center of the sphere (see Figure 3). This function should work for arbitrary desired positions. Note that in order to use the inverse kinematics functions in the toolbox, you need to apply the appropriate mask vector.

PA Q2: Achieve the same result as in PA Q1, but this time using Jacobian pseudoinverse or Jaboian transpose iterations. The exact solution found by your function will probably be different from what you found in PA Q1. However, the end effector should reach the same position.

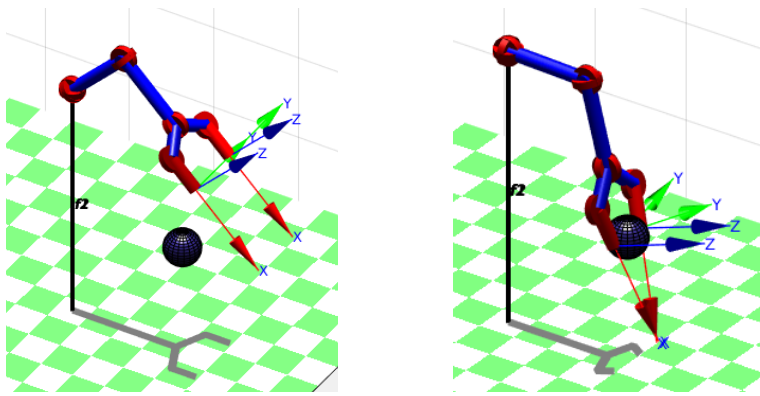
PA Q3: In Q2, you used Jacobian pseudoinverse or transpose iterations to solve for the IK. However, notice that you function returns only a single vector of joint angles corresponding to a final configuration. In this question, I want you to use Jacobian Pseudoinverse control to find a trajectory that moves the end effector to the goal position in precisely a straight line and with a specific velocity. The input to the function is a 3×1 goal position, a parameter EPSILON that specifies the maximum allowed distance of the manipulator from the goal position at termination, and a parameter VELOCITY that specifies exactly how far the end effector should move on each time step. The output of this function should be an $n \times 9$ trajectory that moves the end effector exactly VELOCITY distance per row of the matrix.



(a)

Figure 4: Tracing out a circle in the workspace

PA Q4: Using the result of Q3, write a function that finds a trajectory (i.e. an $n \times 9$ matrix of joint angles) that moves the end effector in a circle as specified in the hw1.m code (Figure 4) at constant velocity. Each row of the output trajectory matrix should cause the end effector to move its position by exactly the velocity specified.



(a) (b)

Figure 5: Joint configurations found by my code before (a) and after (b) running the code in Q3.
(a)

PA Q5: Now, imagine that there is a two-fingered hand on the end of the arm. Use Jacobian pseudoinverse control to move the arm/hand so that the two fingers capture the sphere by moving each finger to one side of the sphere as shown in hw1.m. There are now TWO objectives in this problem (to move each finger to the desired spot), not just one. I want you to solve this problem by formulating a new Jacobian matrix that reflects the two-part objective. Notice that the configuration of the arm and two fingers is now encoded as an 11-dof (degree of freedom) configuration rather than a 9-dof configuration. The first seven joints are the arm joints. The next two joints are for finger f1. The final two joints are for finger f2.