ECE4560 - Homework #3

Due: Sep. 11 2016

Problem 1. [20 pts] Figure 1 depicts a three-link rotational, planar manipulator. Let the link lengths be $l_1 = 1$, $l_2 = \frac{1}{2}$, and $l_3 = \frac{1}{4}$.

(a) Work out the 2D forward kinematics for the manipulator. This should be a function from the three joint angles to planar coordinates (the output is only the point coordinates in $\mathbb{E}(2)$ with orientation ignored). The full symbolic answer should be worked out.

(b) Given the joint angles $\alpha = (-\frac{\pi}{1}2, \frac{\pi}{6}, -2\frac{\pi}{3})^T$, what is the end-effector position (x_e, y_e)?

(c) Work out the (manipulator) Jacobian for the function from part (a).

(d) Using the joint angles from part (b), and the joint velocities $\dot{\alpha} = (\frac{1}{6}, -\frac{1}{2}, \frac{1}{4})^T$, what is the end-effector velocity?

There should be a function called `planarR3_display` that can be used to visualize the manipulator. It is available through the class wiki page. You can even plot the vector in the figure using the `quiver` function in Matlab. Given a set of base points and an associated set of vector coordinates, the `quiver` function will plot the vectors at their associated base points. In this example case, the forward kinematics position would be the base point, and the end-effector velocity would be the vector.

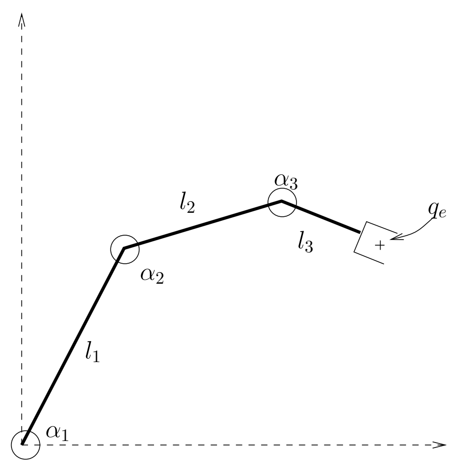

Figure 1: Planar 3R Manipulator.

Problem 2. [15 pts] Every planar transformation going from one frame to another is equivalent to a pure rotation about a unique point in the plane called the *pole*, see Figure 2. Basically, the pole is the point in the plane whose coordinates do not change when the point is rigidly transformed by g. Let q_p denote the location of the pole.

(a) First off, take the above statements and write down what they mean as an equation. In particular, what is the mathematical equation associated to the english of the second sentence? I neglected to include the super- and sub-scripts for the frames. Fill those in properly in your equation. (There are actually two equivalent interpretations, I just want one of them)

(b) If the planar transformation is given symbolically by $g = (\vec{d}, R)$, find the location of the pole symbolically as a function of $\vec{d}$ and R. In what frame did you compute the location of the pole?

(c) Suppose that the initial configuration of the object was $g_{\mathcal{A}}^{\mathcal{O}} = (7.0, 2.0, -3\pi/4)$ and the final configuration was $g_{\mathcal{B}}^{\mathcal{O}} = (0, 8.0, \pi/2)$. Where is the pole located? In what frame did you find the location of the pole?

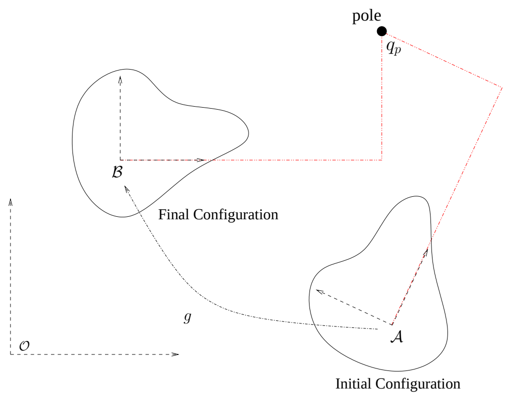

Figure 2: Pole of a planar transformation.

Problem 3. [10 pts] Consider the planar manipulator depicted in Figure 3. It has four rotational joint variables.

(a) What is the end-effector configuration of the following manipulator (symbolically), as a function of the angles $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$? (Recall vector form is (x, y, θ) form).

(b) How would you write it down as a kinematic chain, e.g., as a product of Lie group operations? (Use homogeneous coordinates.)

(c) What is the end-effector's configuration for $\alpha_1 = \pi$, $\alpha_2 = \pi/8$, $\alpha_3 = -\pi/4$, $\alpha_4 = \pi/8$, given that $l_1 = 1$, $l_2 = 0.75$, and $l_3 = 0.75$. (Acceptable in any representation/form)

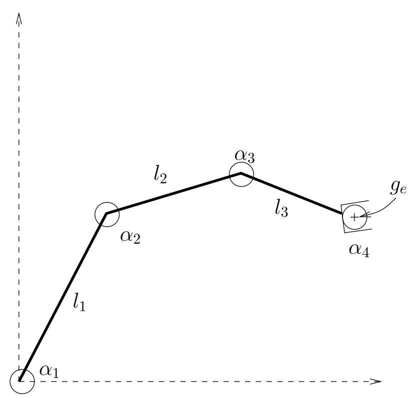

Figure 3: Problem 3 manipulator

Problem 4. [10 pts] Now is the time to start modifying the empty or incomplete functions in the SE2 Matlab class. Using what is known about $SE(2)$ and its operations, complete the functions (they should be inv, leftact, and mtimes). Note that this will define two product operations in Matlab, `*` and `.*` plus inversion. The two operations correspond to the `frame * frame` and `frame * point` operations (as well as the other operations of the matrix from class that have the same structure). Matlab nicely allows us to define these things so that the coded math follows our written math. How nice!

Verify that your coded functions work by showing that, given

$$g_1 = \left(\left\{ \begin{array}{c} 1 \\ 2 \end{array} \right\}, R(\pi/3) \right) \quad \text{and} \quad g_2 = \left(\left\{ \begin{array}{c} -2 \\ 1 \end{array} \right\}, R(\pi/6) \right),$$

the code returns

$$g_1^{-1} = \left(\left\{ \begin{array}{c} -2.23 \\ -0.13 \end{array} \right\}, R(-\pi/3) \right), \quad \text{and} \quad g_1 * g_2 = \left(\left\{ \begin{array}{c} -0.87 \\ 0.77 \end{array} \right\}, R(\pi/2) \right).$$

In practice, this code can be used to verify or compute all of the work from the previous problems, and even to visualize the entire set of solutions. Using the code, also verify your work from Problem 3. If you used the code to solve Problem 3, then you cannot really verify without a second way to compute the answer. In that case, work it out by hand to verify that your code is correct with regards to the point transformations.

Representation: The code is configured to use the Real matrix homogeneous form, and is what has historically been used. However, the Complex matrix homogeneous form works just as well. Decide which of the two you want to use, then go ahead and make the code consistent with that formulation. For those choosing the complex form, the external use will still operate normally, it is just that internally complex numbers will be used. The whole point of classes is to abstract the underlying representation and keep that somewhat secret from the user perspective. The Complex version would be funner to implement, but the course will mostly cover the Real version with some discussion of the Complex equivalent after the Real derivations are done.