MEAM 520 Lab1

Sicheng Tao, Pankti Hitesh Parekh

February 15, 2022

1 Forward Kinematics

1.1 Method

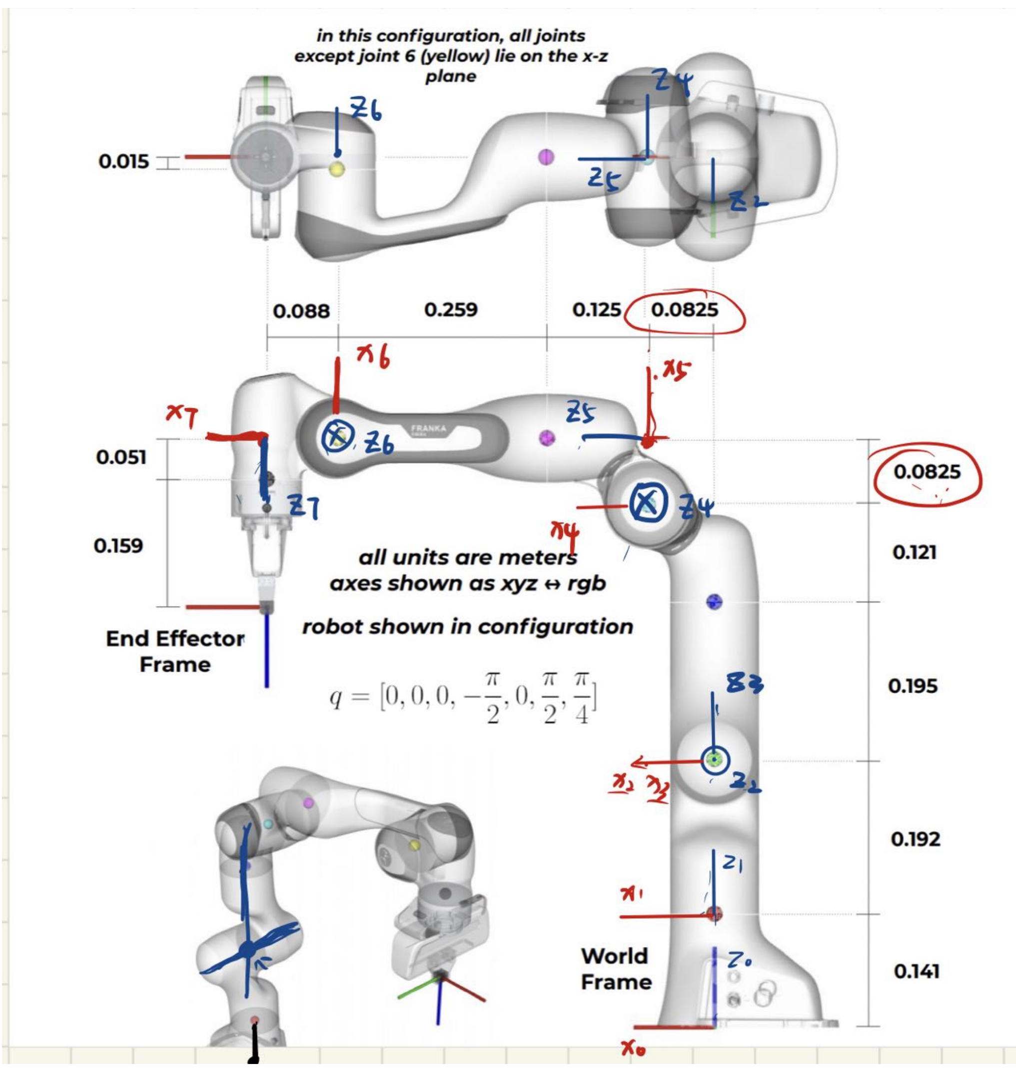


Figure 1: Forward Kinematics using Denavit-Hartenberg (DH) Convention

Steps:

1. Started Schematic at zero pose

2. Attached one frame to each link where frame i-1 is at the start of the link and i at the end of the link. Considered following conditions while allocating frames-

   (a) Located origin o_i where the common normal to z_i and z_{i-1} intersects z_i. It z_i intersects z_{i-1}, then locate o_i at that intersection. If z_i and z_{i-1} are parallel, locate o_i in any convenient position along z_i.
   (b) Joint variable for join i+1 acts along z_i.
   (c) Orientation of z_i defines positive direction.
   (d) Axis x_i is perpendicular to and intersects z_{i-1}.
   (e) Axis y_i is constructed by Right Hand Rule.

3. Computed 4 DH Parameters:

   (a) a_i = Distance between z_{i-1} and z_i measured along x_i to o_i.
   (b) α_i = Angle between z_{i-1} and z_i measured in the plane normal to x_i.
   (c) d_i = Distance between x_{i-1} and x_i measured along z_{i-1} axis.
   (d) θ_i = Angle between z_i and z_{i-1} measured about z_{i-1}.

4. From the homogeneous transformation matrices, we substitute above parameters in

$$A_i = \begin{bmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Link	a_i	α_i	d_i	θ_i
1	0	0	0.141	$\theta_i = 0$
2	0	$-\pi/2$	0.192	θ_2^*
3	0	$\pi/2$	0	θ_3^*
4	0.0825	$\pi/2$	0.316	θ_4^*
5	0.0825	$\pi/2$	0	$\theta_5^* + \pi$
6	0	$-\pi/2$	0.384	θ_6^*
7	0.088	$\pi/2$	0	$\theta_7^* - \pi$
8	0	0	0.21	$\theta_8^* - \pi/4$

Table 1: 4 DH Parameters

5. Hence, to find the transformation of end effector with respect to the base frame,

$$T_n^0 = A_1 * A_2 * A_3 ...... * A_n$$

   In Franka Panda, i = 8 (number of links), Hence we directly substitute all the 4 parameters link wise that we found using DH convention in A_i to get the transformation of end effector with respect to base frame.

6. Additonally, as we are given

$$q = [0, 0, 0, -\pi/2, 0, \pi/2, \pi/4]$$

   Hence, we concatenate 0 as we have 8 links at the start of the array, and subtract the intitally given angles of the position of Franka Panda associate with the respective links.

7. The one last thing for us to get the joints' positions based on frame 0 is the local co-ordinations for each joint based on corresponded frames, and the idea is that:

$$J_i^0 = T_i^0 * J_i^i$$

where J_i^0 is the final result of the $joint_i$ coordination based on frame 0, and J_i^i is the $joint_i$ local coordination based on frame i.

Based on our DH convention, local co-ordinations are:

Joint	x_i^i	y_i^i	z_i^i
1	0	0	0
2	0	0	0
3	0	0	0.195
4	0	0	0
5	0	0	0.125
6	0	0	-0.015
7	0	0	0.051
8	0	0	0

Table 2: DH local co-ordinations

8. (a) For coding task, we apply analysis above using python. There is a method, $get_A(a_i, \alpha_i, d_i, theta_i)$, to calculate transformation matrix between frame i-1 and i using equation of A_i, which is shown above.
   (b) We call the above method for each frame can get A_1, A_2, A_3 ... A_n (A_e for end effector).
   (c) Later, we multiply the matrices using equation of T_n^0 shown above can get the transformation matrix from frame i to 0, T_i^0, and the matrix from frame 8 (or frame e) to frame 0 is one of the final result T0e and then save to return it.
   (d) We multiply the matrix with the corresponding local coordinates using equation of J_i^0 as shown above can and get the final result, J_i^0, and save into joint positions array then return it.

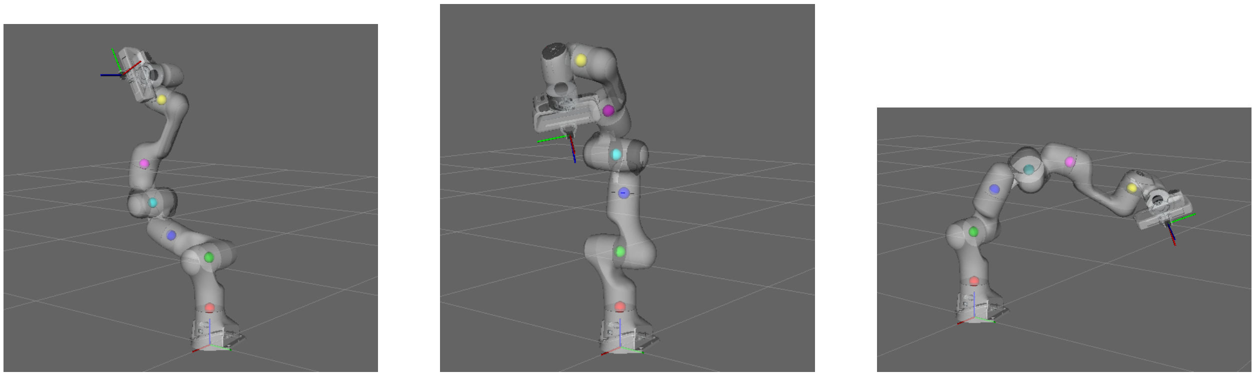
1.2 Evaluation of Forward Kinematics

To evaluate our solution, we tested the program with different inputs. Basically, we used control variate method.

Within the limits, we changed one joint configuration per test to see if each joint works well, in another word, if all joint points locate on the correct joints and positions. The test results are good and correct. Below are three featured test result:

test	q_0	q_1	q_2	q_3	q_4	q_5	q_6
1	$\pi/2$	-0.4*π	0	$-\pi/2$	$-\pi/2$	$\pi/2$	0
2	0	0	0	-1	0	0	0
3	$\pi/2$	$\pi/6$	$\pi/4$	$-\pi/2$	$-\pi/2$	$\pi/2$	0

Table 3: test parameters for FK



(a) Test 1 (b) Test 2 (c) Test 3

Figure 2: Evaluation of Inverse Kinematics

In addition, our code passed all tests in Gradescope.

2 Planar Inverse Kinematics

2.1 Method

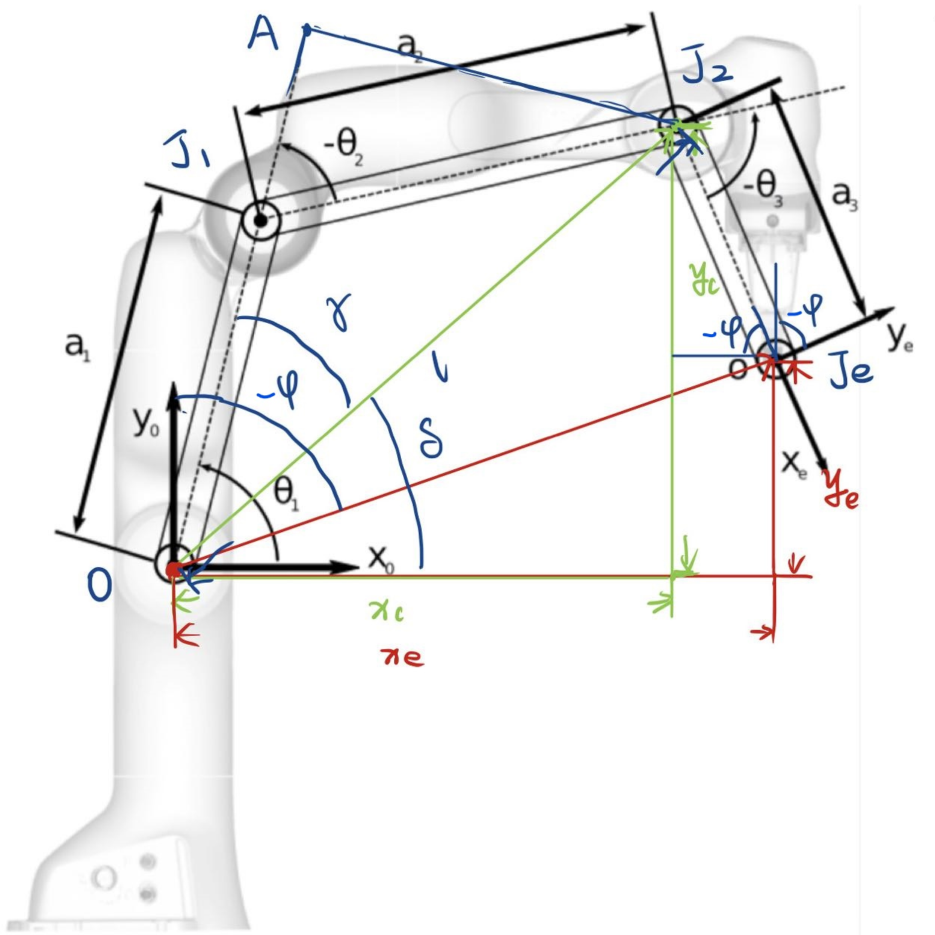


Figure 3: Inverse Kinematics Schematic

The PlanarIK.py can do the task that get the inverse kinematics solutions for a RRR robot which is abstracted from the Panda. The inputs for the function are the end effector position relative to the base frame and the angle of rotation about the base frame's z axis, and the results are all possible configurations for three joints.

The technique we used to solve this problem is the geometric analysis mentioned in class. The geometric graph we analyzed is figure 2, where x_e, y_e, and ϕ are given. To get x_c and y_c, we used kinematic decoupling:

$$\begin{bmatrix} \mathrm{x}_c \\ \mathrm{y}_c \end{bmatrix} = \begin{bmatrix} \mathrm{x}_e \text{ - } a_3\cos(\text{-}\phi) \\ \mathrm{y}_e + a_3\sin(\text{-}\phi) \end{bmatrix}$$

Or,

$$\begin{bmatrix} \mathrm{x}_c \\ \mathrm{y}_c \end{bmatrix} = \begin{bmatrix} \mathrm{x}_e \text{ - } a_3\cos\phi \\ \mathrm{y}_e \text{ - } a_3\sin\phi \end{bmatrix}$$

Consider $\triangle OJ_1J_2$, and use low of cosines to angle between a_1 and a_2, we can easily get:

$$l^2 = a_1^2 + a_2^2 - 2a_1a_2cos(\pi - \theta_2)$$

Then consider cosine angle difference identity:

$$cos(\alpha-\beta) = cos(\alpha)cos(\beta) + sin(\alpha)sin(\beta)$$

get:

$$cos\theta_2 = \frac{x_c^2 + y_c^2 - a_1^2 - a_2^2}{2a_1a_2}$$

Or,

$$\theta_2 = cos^{-1}(\frac{x_c^2 + y_c^2 - a_1^2 - a_2^2}{2a_1a_2}) \tag{1}$$

Apparently, there should have two solutions for θ_2, one positive and one negative.
Now, consider $\triangle OAJ_2$, by observing, can get:

$$\gamma = atan2(\frac{a_2 sin(-\theta_2)}{a_1 + a_2 cos\theta_2})$$

$$\delta = atan2(\frac{y_c}{x_c})$$

and,

$$\theta_1 = \delta + \gamma$$

so,

$$\theta_1 = atan2(\frac{y_c}{x_c}) - atan2(\frac{a_2 sin\theta_2}{a_1 + a_2 cos\theta_2}) \tag{2}$$

Finally, for θ_3, it's apparent that:

$$\phi = \theta_1 + \theta_2 + \theta_3$$

so,

$$\theta_3 = \phi - \theta_1 - \theta_2 \tag{3}$$

and ϕ is known.

For coding task, just apply equations of (1), (2), and (3) in python. Since θ_2 has two solutions, the two solution should both be applied into eq.(2) and eq.(3) and the final results should be corresponded. In our code, label "a" and "2" are connected and represent solutions using negative θ_2, label "b" and "1" are connected and represent solutions using positive θ_2

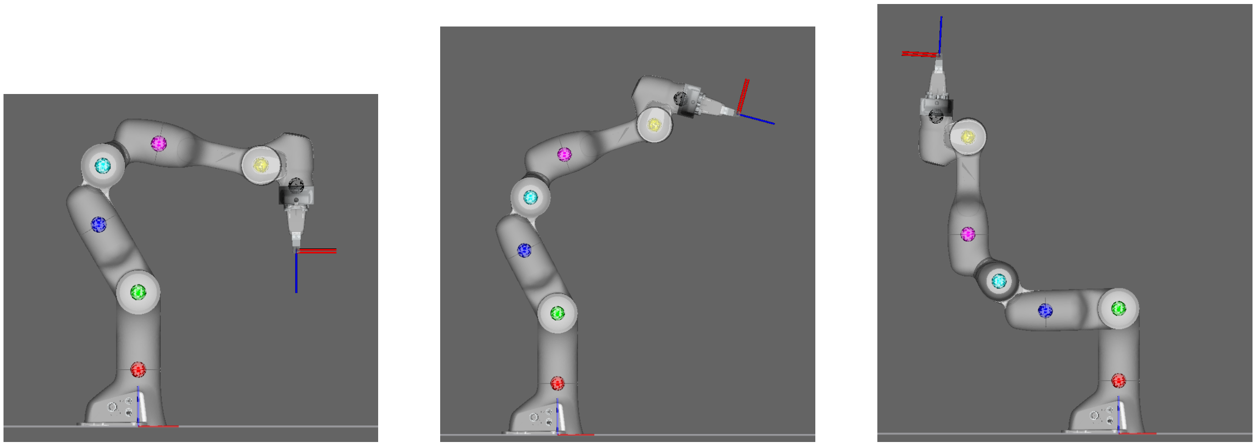
2.2 Evaluation of Inverse Kinematics

To evaluate our solution, we tested different configurations. The idea is that, we use prelab's matrixs to try to get valid configuration for Panda and the coordination for end effector, then, calculate ϕ for the configuration. Take end effector coordination and the angle ϕ into ros or visualize.py and see if the results match.

Within the limits of Panda, we tried as many configurations as possible. Below are three featured new inverse kinematics targets and results:

Test	x_e	y_e	θ_1	θ_2	θ_3	ϕ
1	0.39276	0.43189	$\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$
2	0.4976	0.8783	$\pi/2$	$\pi/4$	$\pi/3$	0.26179
3	-0.477	1.0112569	$\pi * 0.9$	$\pi/4$	$\pi/6$	-1.518436

Table 4: IK test values, notice that all angles above are non-dimensional value based on figure2



(a) Test 1 (b) Test 2 (c) Test 3

Figure 4: Evaluation of Inverse Kinematics

In addition, our code passed all tests in Gradescope.

3 Analysis

3.1 Reachable Workspace

The plot of reachable workspace is shown in figure 5. We use forward kinematics to calculate all possible coordinates of end effector by going through limits of all joints.

There is one assumption that we assume the panda arm has no collision size. However, it is impossible. The true panda arm has volume and some configurations may lead to collision between two parts, for instance, end effector could collide with the arm body if joint 4 is too small and joint 6 is $pi/2$. As a result, the robot can not locate its end effector at some points within our plot.

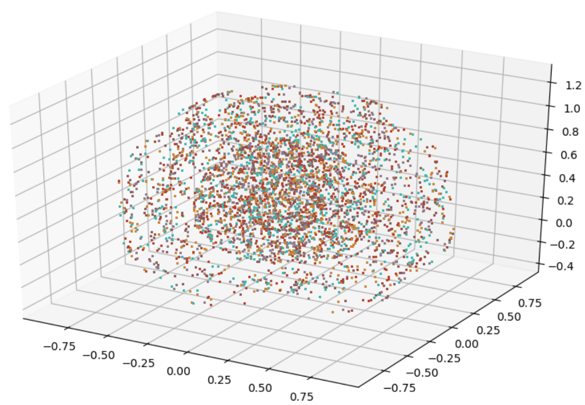


Figure 5: Reachable Workspace

3.2 Extending Inverse Kinematics to 3D

1. No, the Franka Panda Robot arm doesn't have spherical end effector. For an end effector to be spherical, the axis of rotation of three joints should coincide with the wrist center. In this case, the axis of the 7^{th} and the 6^{th} joint are not intersecting. Hence, it's not a spherical end effector.

2. Since the Panda arm does not have a spherical wrist, we cannot do kinematics decoupling and decompose it to simple problems.

3. I believe a 7 degree of freedom arm adds redundancy when a 6 degree of freedom arm can do the same thing with lesser links. Adding more links makes the inverse kinematics problems a bit more complex as we are adding another variable who's value should be determined during inverse kinematics. Our calculations keep on increasing. Hence, it is efficient to solve for six joint variables than seven joint variables.