MEAM 520 Lab 1

Raima Sen, Renu Reddy Kasala

October 1, 2022

1 Methodology

The following methodology was adopted to calculate the forward kinematics of the Franka panda arm:

- Assign a coordinate frame at each joint of the arm
- Assign z axis of each frame along the axis of rotation of the joint
- Assign x axis of each frame as per the coordinate frames shown on RViz
- Assign y axis as per RHR
- Calculate the relative transformation between consecutive pairs of coordinate frames.
  For rotation : Consider consecutive frames i and i+1. Rotate frame i+1 about its z axis by angle θ_{i+1}. Take projections of this rotated i+1 frame on previous frame i. This gives us the rotation matrix.
  For translation : Consider the shift of the origin of frame i+1 with respect to frame i. This gives us the translation vector.
  Combining the rotation and translation we get the homogeneous transformation matrix for each consecutive frame.
- Using post multiplication of the homogeneous matrices obtained by this method we get the pose of the end effector in the base frame.

The homogeneous transformation matrices are as follows :

$$T_0^1 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0.141 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_1^2 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_2 & -c\theta_2 & 0 & 0.192 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_2^3 = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & 0 \\ 0 & 0 & 1 & -0.195 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^4 = \begin{bmatrix} s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -c\theta_4 & s\theta_4 & 0 & 0.121 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_4^5 = \begin{bmatrix} 0 & 0 & -1 & -0.125 \\ c\theta_5 & -s\theta_5 & 0 & -0.0825 \\ -s\theta_5 & -c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_5^6 = \begin{bmatrix} -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & -1 & 0.015 \\ c\theta_6 & -s\theta_6 & 0 & 0.259 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6^7 = \begin{bmatrix} 0 & 0 & -1 & -0.051 \\ -c\theta_7 & s\theta_7 & 0 & -0.088 \\ s\theta_7 & c\theta_7 & 0 & 0.015 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_7^e = \begin{bmatrix} 0.707 & 0.707 & 0 & 0 \\ -0.707 & 0.707 & 0 & 0 \\ 0 & 0 & 1 & 0.159 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The code written by Raima Sen was used for experiments. The transformation matrices are defined in the forward function of FK class. Using post multiplication the final transformation matrix of the end effector in the base frame is obtained. The intermediate joint positions are obtained by doing post multiplication in a step by step manner so that each joint's transformation matrix can be obtained with respect to the base frame. The last column of these intermediate matrices give the x y and z translation of each joint w.r.t the base frame. The main function also plots the x y and z coordinate of each joint in space as a scatter plot.

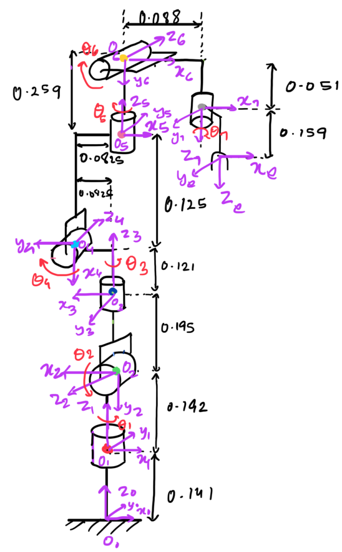


Figure 1: Schematic diagram with assigned coordinate frames

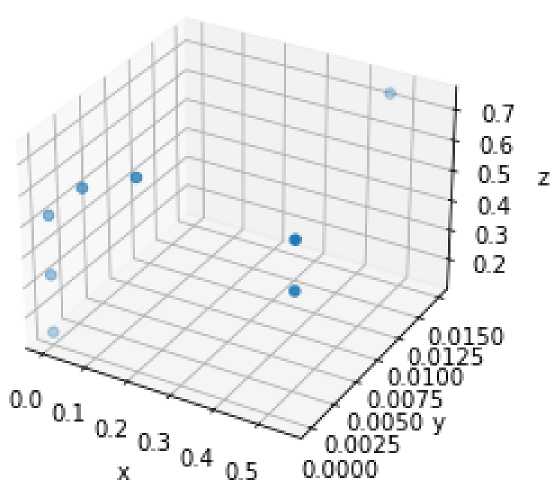


Figure 2: Unit test : Plotting joint positions for $q = [0, 0, 0, -\pi/2, 0, \pi/2, \pi/4]$

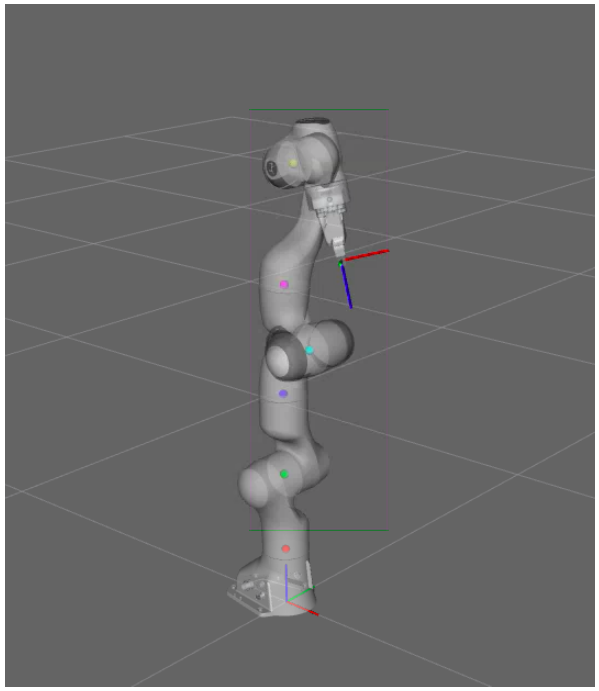


Figure 3: Zero pose

2 Evaluation

Zero Pose

Plugging in 0 for each joint angle leads to the following transformation :

$$T_0^e = \begin{bmatrix} 0.707 & 0.707 & 0 & 0.0875 \\ 0.707 & -0.707 & 0 & 0 \\ 0 & 0 & -1 & 0.823 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

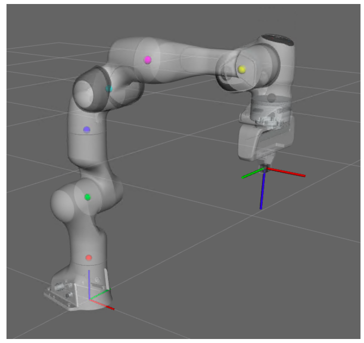
As per the schematic diagram (where the dimensions of the robot are not taken into account), the end effector is rotated by $\pi/4$ and there is no translation present in the y direction. Along the z direction, the translation is maximum, which is obtained both through the simulation and analytically. As seen in Figure 3, the simulation leads to a self collision warning due to the small value of translation in the x direction. This is because of the dimensions of the robot. Overall, the result matches our expectations.

New Configurations

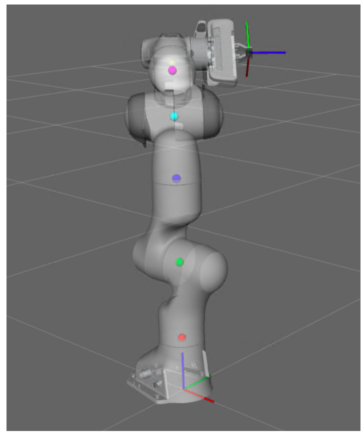
The following 4 other configurations were chosen (Refer Figure 4) :
$q = [0, 0, 0, -\pi/2, 0, \pi/2, \pi/4]$
$q = [\pi/2, 0, \pi/4, -\pi/2, -\pi/2, \pi/2, 0]$
$q = [0, 0, -pi/2, -\pi/4, \pi/2, \pi, \pi/4]$
$q = [\pi/2, 0, 0, -\pi/2, \pi, \pi/2, 0]$

These configurations were chosen because they are also easy to calculate manually through inspection. The calculation of forward kinematics does not provide any information about self collision of the arm. This can be only visualized in the simulation. Additionally, the slight difference in z dimension of the end effector w.r.t the base frame can be accounted for only by hardware testing. It can neither be confirmed in the analytical calculation nor in the simulation (unless a parameter to consider it is set).

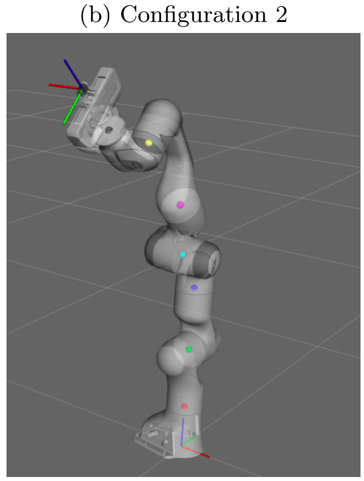
For Configuration 3 the following observation was made from the forward kinematic calculation and



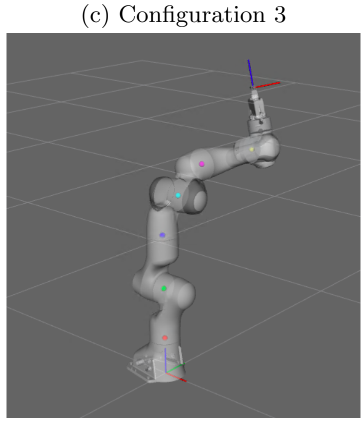
(a) Configuration 1



(b) Configuration 2



(c) Configuration 3



(d) Configuration 4

Figure 4: New configurations

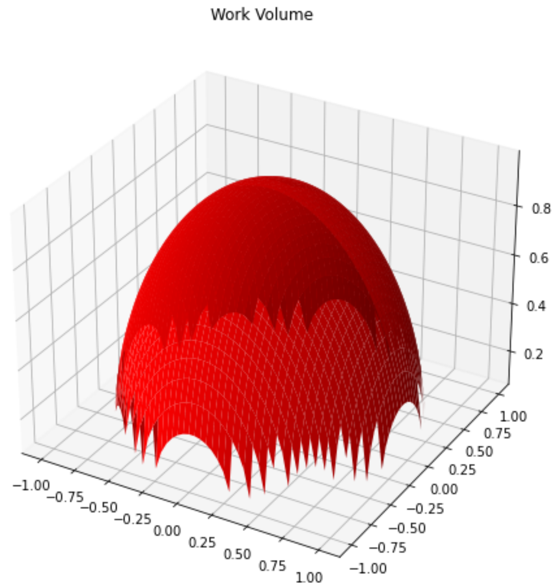


Figure 5: Reachable Workspace

from the RViz simulation platform :

End Effector Pose:

$$T_0^e = \begin{bmatrix} -9.99848989e-01 & -1.04277113e-16 & 1.65762483e-16 & -8.80000000e-02 \\ -5.55111512e-17 & -7.07000000e-01 & -7.07106781e-01 & -4.43685119e-01 \\ 1.89515070e-16 & -7.07000000e-01 & 7.07106781e-01 & 1.12735774 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Observation from Robot lab

According to the transformation, the displacement of the end effector w.r.t the base frame is [0.088, 0.443, 1.127] meters in x,y and z direction. From measurements conducted in the lab for the same configuration for the actual robot, we found out that the end effector is [7, 16, 43] inches ([0.177, 0.406, 1.092] meters) from the base frame along the same directions.

Reachable Workspace

Using the upper limits of the joint angles the forward kinematics were calculated to obtain T_0^e. Similarly, the lower limits of the joint angles were used to again calculate T_{0e}. The last column of these 2 matrices contain the minimum and maximum x, y, and z locations of the end effector. Using the minimum values, a surface was plotted using meshgrid to create the inner work surface. Using the maximum values another surface was plotted using np.meshgrid. Within this 3D boundary, the reachable workspace of the robot is defined.

3 Analysis

The following analysis was done to estimate the performance of obtained solutions:

- Initially, the joint positions were not moving with the robot arm as expected. They either flew away or remained at a constant position in space when the forward kinematics were performed. But this issue was resolved after analysing the intermediate frame transformations and post multiplying successive transformations.

  The functional outputs and the results of the simulation were almost identical in the sense the simulation showed the same positions that were obtained through the calculations. We checked this for some test cases and verified that the results were sound.

- There is some difference between the theoretical and simulated workspace of the robot. Theoretically the Franka Panda arm should have a spherical workspace ignoring the joint limits. But the simulated workspace has some gaps in the workspace that we didn't expect.
When simulating the arm, the zero configuration of the robot arm showed a self collision warning but there was no function output error as such. No points were predicted as reachable that were not in simulation. But we observed that some points like those on the table could not be accessed due to the obstruction present in the real world, whereas such obstacles we not considered in the simulation. Additionally creating an environment for the robot arm will lead to accounting for these prediction differences.

- Software to hardware hurdle: Simulated robot and physical robot differed in the sense that we need to consider various factors like presence of human in the robot workspace, and other obstacles in the path of the robot whereas we could check any configuration without any complications using simulations.

  Some factors which might cause a solution that works in simulation to fail on hardware are power loss, issue in software connection or safety steps like software kill or main kill switch that might be implemented in case of emergencies. From testing our solution on hardware we learnt the importance of being present at the moment and initiating safety steps if there is any unexpected movement in the environment or by the robot itself.

  In hardware self collision is a hard parameter whereas in software it is considered as a warning and not as strict error. In hardware other parameters like gravity, friction are also be considered. Also we learn how using correct parameter aid it maintaining the robot is good condition while no thought on these areas were need when performing simulation tests.

- We measured the end effector position from the base frame and found that X= 7 inches, Y = 16 inches and Z = 43 inches for the $q = [0, 0, -pi/2, -\pi/4, \pi/2, \pi, \pi/4]$ This is equal to [0.177, 0.406,1.092] meters. Through simulation, we found the end effector position to be [0.088,0.443, 1.127] meters ,which is the approximately equal to the actual measured values considering that there might have been some errors during measurement.

Figure 6: Actual Franka panda arm in $[0, 0, -pi/2, -\pi/4, \pi/2, \pi, \pi/4]$ configuration