Lab 1: Forward Kinematic

MEAM 5200

Bruke Baraki, Hongyi Lai

Methods

The forward kinematics for the Panda robot arm was derived using a series of transformations, as shown in Equation 1. The transformation T_e^0 is the end effector frame, represented in the base frame. These frame matrices were derived using the DH convention, whose parameters, as shown in Table 1, were generated using the schematic in Figure 1. Figure 1

(Equation 1): $T_e^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 T_e^6$

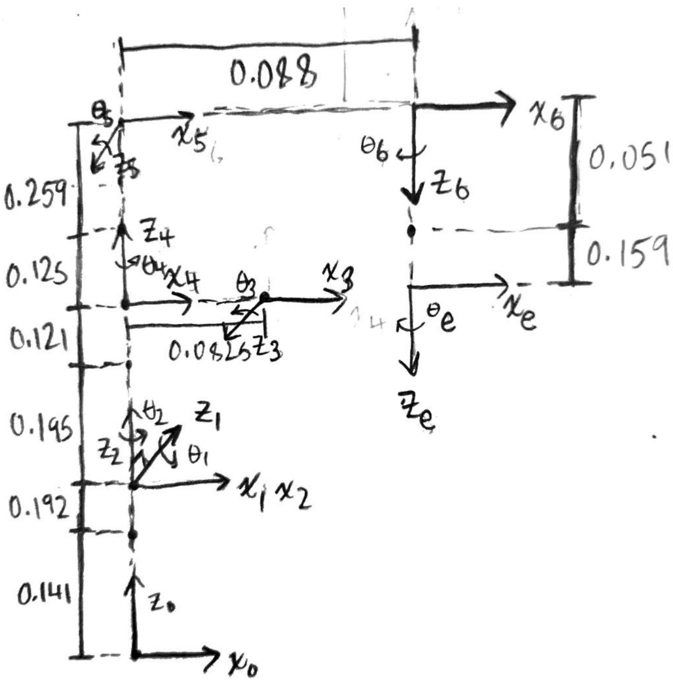

Figure 1: Drawn Schematic of FRANKA EMIKA Panda in meters

Link	α [º]	a [m.]	d [m.]	Θ [º]
1	-90	0	0.333	Θ_1
2	90	0	0	Θ_2
3	90	0.0825	0.316	Θ_3
4	-90	-0.0825	0	Θ_4
5	90	0	0.384	Θ_5
6	-90	0.088	0	Θ_5
7	0	0.051	0	Θ_e

Table 1: DH Parameters

The code that was utilized belonged to Bruke Baraki. The FK class takes 7 joint angles as inputs and returns the positions of the 8 rotational joints of the robot in the world frame, and the end effector pose in the world frame represented as a 4x4 homogeneous transformation matrix. The DH parameters were saved in a list, and a for loop generated all 7 transformation matrices. The forward kinematics calculation is performed using the 7 matrices, as shown in equation 1. For instance, the second joint along the kinetic chain utilizes the first two transformation in equation 1, which is multiplied by a vector from Frame 2 (i.e joint 2 $= T_1^0 T_2^1 u$).

Evaluation

Position of End Effector at Zero Pose

At zero pose, all of the angles of the joints are 0º. The input vector would have the form: q = [0,0,0,0,0,0,0].

$$T0e = \begin{bmatrix} 0.7071 & 0.7071 & 0 & 0.0825 \\ 0.7071 & -0.7071 & 0 & 0 \\ 0 & 0 & -1 & 0.9820 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 2: T_e^0 at zero pose

$$\begin{bmatrix} 0.088 \\ 0 \\ 0.982 \\ 1 \end{bmatrix} = T0e \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Equation 3: Position of end effector

This does match our expectations since at zero pose. When all the angles are at zero, a majority of the robotic arm is positioned upwards, while joint 6 and end effector is positioned downwards, resulting in the end effector's displacement of 0.926m. from the base frame. The simulation output, as shown in equation 3, has 0.982 in the z-component. Although the values differ slightly, this comparison verifies our expectations.

Comparison for 3 other inputs

End Effector Pos.	Input Position [rad.] [q_0, q_1, q_2, q_3, q_4, q_5, q_6]	Experimental Data for EE's position [m.] [x, y, z]	Simulation Results for EE's position [m.] [x, y, z]
1	[1, 1, -1, -1, 1, 1, 1]	[0.559, 0.368, 0.254]	[0.515, 0.378, 0.274]
2	[0, 1, 2, -0.5, 0, 1.5, 0]	[0.470, 0.357, 0.813]	[0.456, 0.380, 0.832]
3	[-0.5, -1, -0.5, -0.5, 0 ,1,0]	[-0.787, 0.191, 1.0414]	[-0.337, -0.019, 0.905]

These configurations were chosen since they covered the range values for each joint. For instance, joint 1(q_1) ranges from -2.8973 to 2.8973. The values for q_1 were 1, 0, and -0.5.

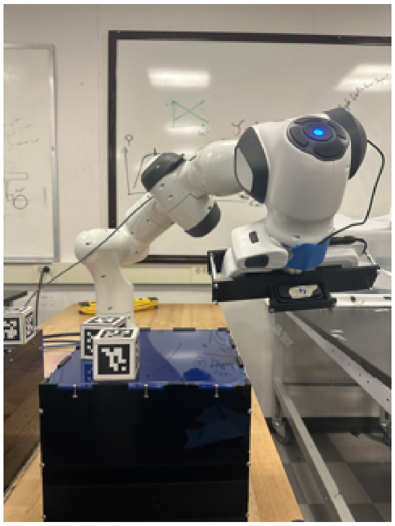

Figure 2: The Picture of Effector **Pos. 1**

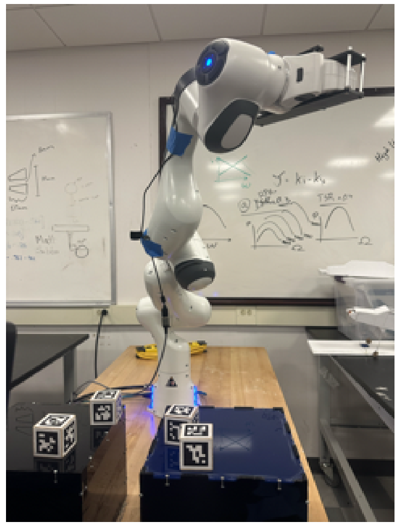

Figure 3: The Picture of Effector **Pos. 2**

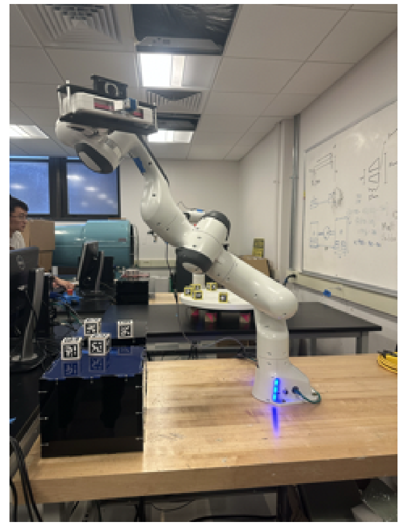

Figure 5: The Picture of Effector **Pos. 2**

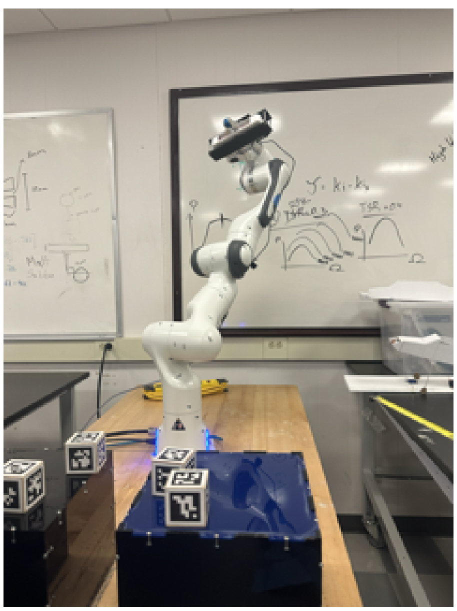

Figure 6: The Picture of Effector **Pos. 3**

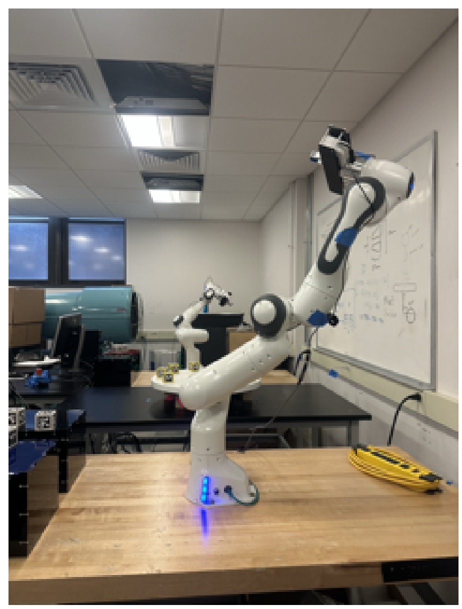

Figure 7: The Picture of Effector **Pos. 3**

A simulation of the robot in a specific configuration can provide additional information that the forward kinematics calculation cannot provide. Workplace.py was called to simulate the reachable workspace of the robot's end effector by inputting a wide range of angle values into the Fk.forward() function, as shown in figure 8. A simulation can show the collision between the robot and its environment, the dynamic response of the robot to external forces, the stability of the robot, and even the measurement.

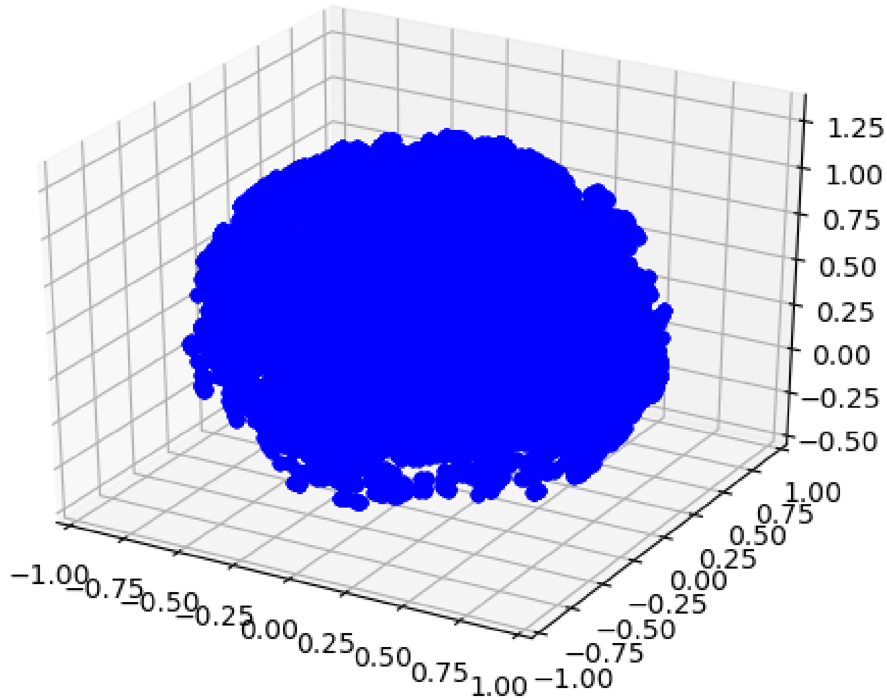

Figure 8: Reachable Workspace [m.]

During our experiments, for example, in position 3, the angle of the robot arm is relatively large, so it leads to some inconvenience when the TA helps us in the measurement, which is inconceivable in the simulator. These additional pieces of information can be crucial in understanding the behavior and performance of the robot in a specific scenario and can help in the design and optimization of the robot.

Analysis

The difference between the experimental data and simulation results for the end effector's position is significant in some cases and does not meet our expectations. For example, the difference in x, y, and z between the first sets of data is [0.044, 0.010, -0.039] which is a significant difference in the x and z directions. The difference in x, y, and z between the second sets of data is [0.029, 0.023, 0.109], which is a significant difference in the z direction. I think a lot of this error comes from the measurement method, because we use the kind of tape measure in the toolbox, and only through the naked eye to aim. So it is easy to error, coupled with some position is very "open", so when measured again will produce more error.

There may be differences between the workspaces of the theoretical and simulated robot due to several factors, such as differences in the geometric parameters of the robot, the accuracy of the joint angles in the simulation, and the precision of the forward kinematics calculation. These differences can result in points that are predicted as reachable in the theoretical model but are not reachable in the simulation, or vice versa.

For example, if the geometric parameters in the simulation differ from the actual robot, the forward kinematics calculation may not accurately reflect the end effector's position. To account for these differences, the geometric parameters used in the simulation should be verified and updated as necessary to match the actual robot. Additionally, if the joint angles in the simulation are not accurate, the forward kinematics calculation may result in incorrect end effector positions. To improve the accuracy of the joint angles, the simulation can be calibrated or the joint angles can be measured directly on the actual robot.

The software to hardware hurdle refers to the challenges of implementing a solution that works in simulation on a physical robot. There are several factors that can cause a solution that works in simulation to fail on hardware, including:

1. Sensor noise: The sensors on the physical robot may introduce noise or measurement error, which can impact the performance of the solution.
2. Environmental factors: The physical robot may be subject to external forces, such as friction, gravity, or wind, that are not present in the simulation, leading to differences in the behavior of the system.
3. Latency: The physical robot may have latency in its control loop, leading to delays in the response of the system.

In summary, testing the solution on hardware can provide valuable information that cannot be obtained from simulation tests. We do learn lots of things in our labs, after we finished this lab, we will be more thoughtful when we encounter the same problem again in the future