MEAM 5200 - Lab 2: Inverse Kinematics

Raima Sen, Renu Reddy Kasala

1 Methods

1.1 Inverse Kinematics

The strategy for computing the full inverse kinematics for the simplified Panda robot arm is as follows: The Panda robot arm is a 7-DOF robot, redundant manipulator with one free degree of freedom. While calculating inverse kinematics we will be fixing joint 5 at an angle of $\theta_5 = 0$, making the Panda arm a 6-DOF system.

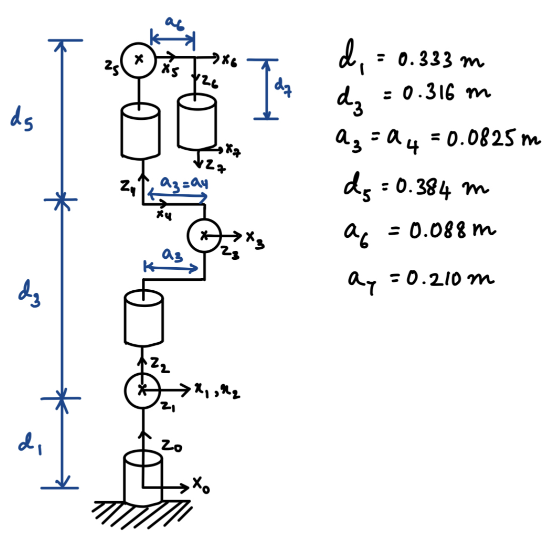

Figure 1: Schematic diagram with assigned coordinate frames

The Panda arm has an offset wrist, which means that axes of rotation for the last three joints do not intersect at a single point. This means that the joints 1,2 and 3 are in spherical wrist configuration with wrist center at joint 2. By treating it as if it were upside down we can do the kinematic decoupling on the arm.

The transformation matrix T_7^0 can be obtained by using DH parameters.
$T_7^0 = T_1^0 \, T_2^1 \, T_3^2 \, T_4^3 \, T_5^4 \, T_6^5 \, T_7^6$

$$T_7^0 = \begin{bmatrix} \mathrm{R}_7^0 & \mathrm{O}_7^0 \\ 0 & 1 \end{bmatrix}$$

We did the inverse of the transformation T_7^0 to turn the robot in upside down configuration.

$(T_0^7) = \begin{bmatrix} \text{-R}_7^0 & \text{-(R}_7^0)^T \; O_7^0 \\ 0 & 1 \end{bmatrix}$

Then kinematic decoupling is performed using T_0^7. The position of wrist center which is the joint 2 in frame 7 can be obtained from robot geometry as:

$O_2^7 = O_0^7 + d_1 \; R_0^7 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

Considering the robot geometry we get the joints 4, 6, and 7. Only these joint variables affect the vector O_2^7 since joint 5 is fixed, and joints 1, 2, 3 do not affect this vector. This is because joints 1, 2 and 3 constitute a spherical wrist. We know that the motion of the three links about those axes will not change the position of wrist center, and the position of the wrist center is a function of only the other 4 joint variables out of which we fixed joint 5 as $\theta 5 = 0$. This means that only joints 4, 6 and 7 affect the O_2^7 vector.

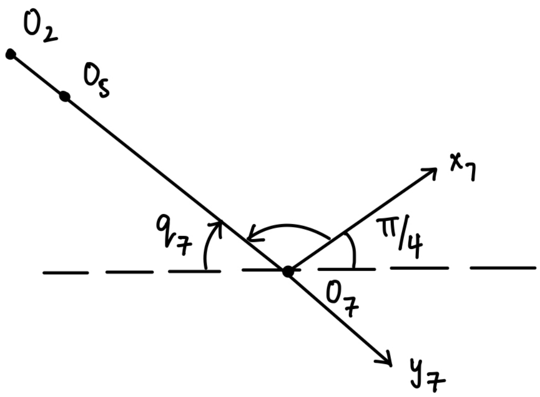

Figure 2: Top view of Franka Panda arm

From geometry, we get that:

$\theta_7 = \pi$ - $(atan2(\frac{-O_2,y^7}{-O_2,x^7}) + \frac{\pi}{4})$

And, $\theta_6 = \theta_1$ - $\frac{\pi}{4} + tan^{-1}1(\frac{a_3}{d_5})$

$\theta_4 = \theta_2 + tan^{-1}1(\frac{d_3}{a_3}) + tan^{-1}1(\frac{d_5}{a_3})$ - π

$O_{2-5}^6 = O_2^6 + \begin{bmatrix} \text{a}_6 \\ 0 \\ 0 \end{bmatrix}$

where θ_1 = atan2$(\frac{O_y}{O_x})$ - atan2$(\frac{a_2 sin(\theta_2)}{a_1 + a_2 cos(\theta_2)})$

$\theta_2 = cos^{-1}(\frac{O_x^2 + O_y^2 - a_1^2 - a_2^2}{2a_1a_2})$

In order to find the spherical wrist angles 1, 2, and 3 we compute R_2^0.
$R_7^0 = R_2^0 \; R_7^2$
$R_2^0 = (R_7^2)^{-1} \; R_0^7$

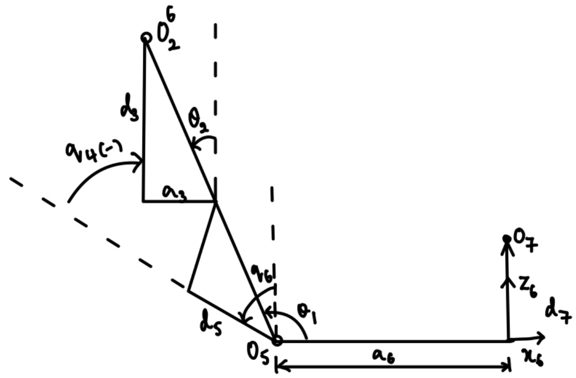

Figure 3: Front view of Franka Panda arm

By using Euler angles concept we get: R = $R_{z,\phi}$ $R_{y,\theta}$ $R_{z,\psi}$

$$R = \begin{bmatrix} \cos(\phi)cos(\theta)cos(\psi) - sin(\phi)sin(\psi) & -\cos(\phi)cos(\theta)sin(\psi) - sin(\phi)cos(\psi) & \cos(\phi)sin(\theta) \\ \sin(\phi)cos(\theta)cos(\psi) + cos(\phi)sin(\psi) & -\sin(\phi)cos(\theta)sin(\psi) - cos(\phi)cos(\psi) & \sin(\phi)sin(\theta) \\ -\sin(\theta)cos(\psi) & \sin(\theta)sin(\psi) & \cos(\theta) \end{bmatrix}$$

where $\phi = \theta_1$, $\theta = \theta_2$, $\psi = \theta_3$ and R = R_0^3

From this we know $cos(\theta_2) = R_0^3$[2,2], so we'll get θ_2

From $cos(\phi)sin(\theta) = R_0^3$[0,2] and $sin(\phi)sin(\theta) = R_0^3$[1,2] , we get θ_1

And from $-sin(\theta)cos(\psi) = R_0^3$[2,0] and $sin(\theta)sin(\psi) = R_0^3$[2,1] , we get θ_3

We can see that we get multiple solutions for some joint variables. In some end effector configurations we will get infinite solutions. For these cases we used Q0 as the joint variable value for that joint.

1.2 Code

We used Renu's code. The code structure is summarized below:

- First, the R_7^0 and t_7^0 are extracted from the dictionary. Using these the R_0^7 is found. Then T_7^0 is used to calculate T_0^7.
- Using kinematic decoupling equations we found the wrist center positions O_2^7. Then we computed the angles 7, 6, and 4 by geometry.
- Then calculated the angles 1, 2, and 3 using the Euler angle formulation.

  But this solution was incorrect as the matrix difference did not equate to zero as expected. The joint variables were outputting correct angles in some configurations but did not in many other case. This could be because there were some calculation errors as all the possible joint angles were not taken into account in cases where there were multiple solutions.

  Our angles seem suitable for passing for the first quadrant. We had expected arctan2 to generate multiple values of for each angles but the values are limited to -π to π. This is why, we must have explicitly written individual multiple solutions for each angle. Using arctan instead of arctan2 also did not yield multiple solutions.

After thoroughly inspecting the code, we found that there was an error in cases where there were multiple solutions for each joint. If there was more time, then we would've tried out different configurations and debugged it to perform better.

2 Evaluation

To ensure correctness of our testing process we considered the following evaluation tests:

- The first thing that we did to check the correctness of the code is to see if the Matrix difference = zero. That will tell us that the end-effector transformation matrix calculated from the joint variables obtained from the Inverse Kinematics is the same as the the end-effector transformation matrix obtained from Forward Kinematics.
- Another way to check the correctness of the code is plug in the joint variable values obtained from the Inverse Kinematics into the Forward Kinematics code and check if the end-effector is in the correct position or not.

Using these test cases we can check if the joint angles outputted by our code produced the expected position and orientation of the end effector.
When the desired position is outside of the robot's reachable workspace then there will be no solution.
For end effector positions within the reachable workspace, there are some orientations that the end effector cannot attain, these are the singular configurations or the positions that violate the joint limits.

3 Analysis

The solutions obtained by geometric analysis did not yield to correct result for all quadrants, which is why we also attempted to solve it using algebraic decomposition.

- Since the IK function did not perform as expectT_0^7ed, we were unable to make judgments on the simulation and hardware. However, if it had worked appropriately, we should have been able to compare the simulation and hardware end effector position with the one obtained by using the IK output angles in the forward kinematic calculation.
- The main difference between the simulation and hardware experiments would have been the singularity configurations and self collision configurations. The simulation would have still output some values but the hardware would have failed. Also the hardware would have motor torque constraints, link mass, and gravity as additional factors in determining the end effector pose. The same will not be the case for the simulation and code outputs.
- For all the available valid solutions for each angle, the selection process will be as follows :
  1. remove the angle that leads to loss of DOF in the manipulator (singularity)
  2. select only the angles within the joint limits of the robot link to prevent self collision
  Therefore, singularity and self collision would be the 2 issues that would mainly determine selection from all the possible solutions.