DEPARTMENT OF HUMANITIES AND SOCIAL SCIENCES

INDIAN INSTITUTE OF TECHNOLOGY ROPAR

RUPNAGAR-140001, INDIA

ME512: Robotic Manipulators: Kinematics, Dynamics and Control

Assignment 3

Robotic Manipulator Used:

frankaEmikaPanda

Submitted by

VIKRAM SETTY (2018MED1010)

PARNAVI SHRIKHANDE (2018MED1007)

MECHANICAL ENGINEERING - DUAL DEGREE (SECTION - 4)

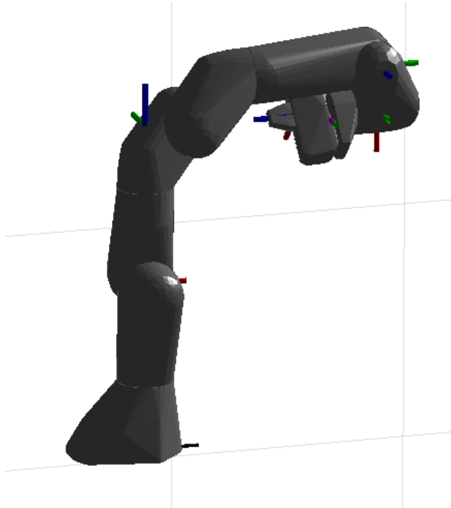
Supervised by

Dr. Ekta Singla

Second Semester, 2021-2022

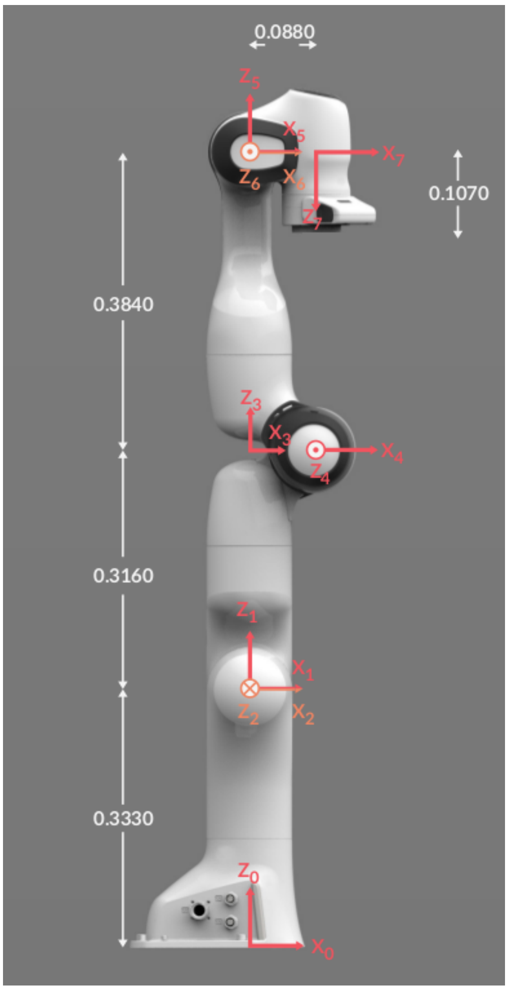
Report Submitted On: 20-02-2022

The manipulator that we have used in this assignment is the **frankaEmikaPanda**. The figure below shows the home configuration of the manipulator as seen in MATLAB. It also has an end effector at the end in the form of two fingers, but has been ignored for this purpose of this assignment. The forward kinematics part has been computed in '**Forward_Kinematics.ipynb**', and the inverse kinematics part in '**Inverse_Kinematics.m**'.



I. Developing the Kinematic Model

We assign the frames to the manipulator as shown in the figure below:



The DH Table looks as follows:

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	$d_1 = 0.333$ m	θ_1
2	0	$-\pi/2$	0	θ_2
3	0	$\pi/2$	$d_3 = 0.316$ m	θ_3
4	$a_4 = 0.0825$ m	$\pi/2$	0	θ_4
5	$a_5 = -0.0825$ m	$-\pi/2$	$d_5 = 0.384$ m	θ_5
6	0	$\pi/2$	0	θ_6
7	$a_7 = 0.088$m	$\pi/2$	0	θ_7

For forward kinematics, we are considering two use cases, one with seven degrees of freedom as mentioned above, and the second with just the first two degrees of freedom. For the first case, the transformation matrices are as follows:

```
Intermediate Transformation Matrix:
[[ 1. -0.  0.  0.]
 [ 0.  1. -0. -0.]
 [ 0.  0.  1.  0.]
 [ 0.  0.  0.  1.]]
Intermediate Transformation Matrix:
[[ 1.000000e+00 -0.000000e+00  0.000000e+00  8.800000e-02]
 [ 0.000000e+00  6.123234e-17 -1.000000e+00 -0.000000e+00]
 [ 0.000000e+00  1.000000e+00  6.123234e-17  0.000000e+00]
 [ 0.000000e+00  0.000000e+00  0.000000e+00  1.000000e+00]]
Intermediate Transformation Matrix:
[[ 1.000000e+00 -0.000000e+00  0.000000e+00  0.000000e+00]
 [ 0.000000e+00  6.123234e-17 -1.000000e+00 -0.000000e+00]
 [ 0.000000e+00  1.000000e+00  6.123234e-17  0.000000e+00]
 [ 0.000000e+00  0.000000e+00  0.000000e+00  1.000000e+00]]
Intermediate Transformation Matrix:
[[ 1.00000000e+00 -0.00000000e+00  0.00000000e+00 -8.25000000e-02]
 [ 0.00000000e+00  6.12323400e-17  1.00000000e+00  3.84000000e-01]
 [-0.00000000e+00 -1.00000000e+00  6.12323400e-17  2.35132185e-17]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Intermediate Transformation Matrix:
[[ 1.000000e+00 -0.000000e+00  0.000000e+00  8.250000e-02]
 [ 0.000000e+00  6.123234e-17 -1.000000e+00 -0.000000e+00]
 [ 0.000000e+00  1.000000e+00  6.123234e-17  0.000000e+00]
 [ 0.000000e+00  0.000000e+00  0.000000e+00  1.000000e+00]]
Intermediate Transformation Matrix:
[[ 1.00000000e+00 -0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  6.12323400e-17 -1.00000000e+00 -3.16000000e-01]
 [ 0.00000000e+00  1.00000000e+00  6.12323400e-17  1.93494194e-17]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

```
Intermediate Transformation Matrix:
[[ 1.00000000e+00 -0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  6.12323400e-17 -1.00000000e+00 -3.16000000e-01]
 [ 0.00000000e+00  1.00000000e+00  6.12323400e-17  1.93494194e-17]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Intermediate Transformation Matrix:
[[ 1.000000e+00 -0.000000e+00  0.000000e+00  0.000000e+00]
 [ 0.000000e+00  6.123234e-17  1.000000e+00  0.000000e+00]
 [-0.000000e+00 -1.000000e+00  6.123234e-17  0.000000e+00]
 [ 0.000000e+00  0.000000e+00  0.000000e+00  1.000000e+00]]
Intermediate Transformation Matrix:
[[ 1.    -0.     0.     0.   ]
 [ 0.     1.    -0.    -0.   ]
 [ 0.     0.     1.     0.333]
 [ 0.     0.     0.     1.   ]]
```

The final transformation matrix (mentioned as the last intermediate transformation matix) for the case with 7 degrees of freedom (**Case 1**) is as follows:

```
[[ 1.0000  0.0000  0.0000  0.0880]
 [ 0.0000 -1.0000 -0.0000  0.0000]
 [ 0.0000  0.0000 -1.0000  1.0330]
 [ 0.0000  0.0000  0.0000  1.0000]]
```

Thus, the position of origin i.e. (0,0,0) of End Effector's Frame in Ground Coordinates: **(0.088, 0, 1.033)**

Similarly, for the case of two degrees of freedom, the intermediate transformation matrices are:

```
Intermediate Transformation Matrix:
[[ 1. -0.  0.  0.]
 [ 0.  1. -0. -0.]
 [ 0.  0.  1.  0.]
 [ 0.  0.  0.  1.]]
Intermediate Transformation Matrix:
[[ 1.000000e+00 -0.000000e+00  0.000000e+00  0.000000e+00]
 [ 0.000000e+00  6.123234e-17  1.000000e+00  0.000000e+00]
 [-0.000000e+00 -1.000000e+00  6.123234e-17  0.000000e+00]
 [ 0.000000e+00  0.000000e+00  0.000000e+00  1.000000e+00]]
Intermediate Transformation Matrix:
[[ 1.    -0.     0.     0.   ]
 [ 0.     1.    -0.    -0.   ]
 [ 0.     0.     1.     0.333]
 [ 0.     0.     0.     1.   ]]
```

The final transformation matrix (mentioned as the last intermediate transformation matrix) for the case with 2 degrees of freedom (**Case 2**) is as follows:

```
[[ 1.0000  0.0000  0.0000  0.0000]
 [ 0.0000  0.0000  1.0000  0.0000]
 [ 0.0000 -1.0000  0.0000  0.3330]
 [ 0.0000  0.0000  0.0000  1.0000]]
```

Thus, the position of origin i.e. (0,0,0) of End Effector's Frame in Ground Coordinates: **(0, 0, 0.333)**

II. Verification of Kinematic Model

For case 1 (7 DOF):

For the home parameters, as shown in the first figure, i.e., all θ = 0, we have

1. x-coordinate = $a_4 + a_5 + a_7$ = -0.0825 + 0.0825 + 0.088 = **0.088 m**
2. y-coordinate = 0
3. z-coordinate = $d_1 + d_3 + d_5$ = 0.333 + 0.316 + 0.384 = **1.033 m**

These geometric values, (0.088, 0, 1.033) match with the values which we obtained from the kinematic model, thus verifying the same.

For case 2 (2 DOF):

For the home parameters, i.e., all θ = 0, we have

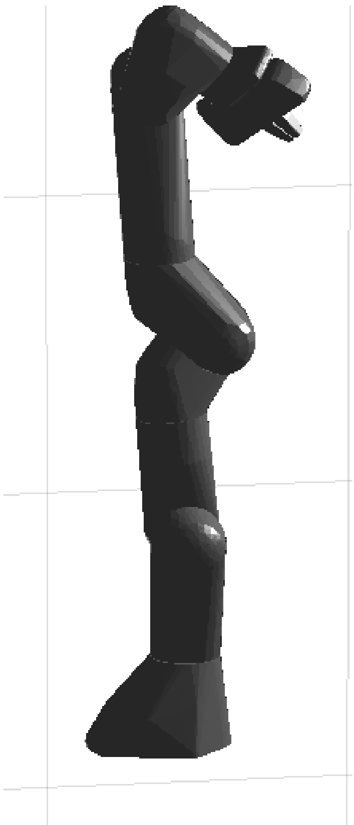
4. x-coordinate = 0
5. y-coordinate = 0
6. z-coordinate = d_1 = **0.333 m**

These geometric values, (0, 0, 0.333) match with the values which we obtained from the kinematic model, thus verifying the same.

III. Inverse Kinematics

Case 1: We Consider all the 7 DoF, and compute inverse kinematics for the origin of the last frame.

As for Case 1 in forward kinematics, we found the position of the end effector (origin of last frame) to be (0.088, 0, 1.033) in the ground frame coordinate system, when we set all the joint parameters (angles in this case since we deal only with revolute joints) to 0.



JointName	JointPosition
{'panda_joint1' }	-7.8124e-12
{'panda_joint2' }	-0.10991
{'panda_joint3' }	-1.567e-11
{'panda_joint4' }	-0.0698
{'panda_joint5' }	-1.5685e-11
{'panda_joint6' }	0.81696
{'panda_joint7' }	0

As we observe, we get all the joint parameters close to zero, on using the inverse kinematics solver for this point, and the picture of the robot for this configuration looks highly accurate to how we'd expect it to be as well.

We do however observe slight deviation here and there due to the complex nature of inverse kinematics at high degrees of freedom, as well as the slight difference in joint ad link dimensions between matlab and the original franka emika panda model (especially during the last few links).

Case 2: We consider only the first 2 DoF, and compute inverse kinematics for the origin of the last frame.

As for Case 2 in forward kinematics, we found the position of the end effector (origin of last frame) to be (0, 0, 0.033) in the ground frame coordinate system, when we set all the joint parameters (angles in this case since we deal only with revolute joints) to 0.

JointName	JointPosition
{'panda_joint1' }	0
{'panda_joint2' }	0
{'panda_joint3' }	0
{'panda_joint4' }	-1.5708
{'panda_joint5' }	0
{'panda_joint6' }	0
{'panda_joint7' }	0

We've considered only with the first two angles since we are only considering two degrees of freedom As we observe, we get all the joint parameters close to zero, on using the inverse kinematics solver for this point, and the picture of the robot for this configuration looks highly accurate to how we'd expect it to be as well.

In this however, we get no discrepancies if any form, and get perfect 0's as the joint parameters for the first two joints (as expected) owing to the lower complexity of 2 DoF systems, as well as no geometrical discrepancies in the matlab model of franka emika panda.