

Forward Kinematics using DH notation – Pybullet practice

MECH439 Intro. to Robotics

TA Session

2024.04.03.(W)

Forward Kinematics

How to get end-effector's homogeneous transformation(orientation, position) from joint variables?

PoE, DH notation, ...

DH notation

Solve FK problem using relative transformation between two adjacent coordinate frames, with predefined convention

$${}^iD_{i+1} = T_{z_i}(d_{i+1})R_{z_i}(\theta_{i+1})T_{x_{i+1}}(r_{i+1})R_{x_{i+1}}(\alpha_{i+1})$$

$${}^iD_{i+1} = \begin{bmatrix} C\theta_{i+1} & -S\theta_{i+1}C\alpha_{i+1} & S\theta_{i+1}S\alpha_{i+1} & r_{i+1}C\theta_{i+1} \\ S\theta_{i+1} & C\theta_{i+1}C\alpha_{i+1} & -C\theta_{i+1}S\alpha_{i+1} & r_{i+1}S\theta_{i+1} \\ 0 & S\alpha_{i+1} & C\alpha_{i+1} & d_{i+1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D_n = D_0 {}^0D_1 {}^1D_2 \cdots {}^{n-1}D_n$$

Mission

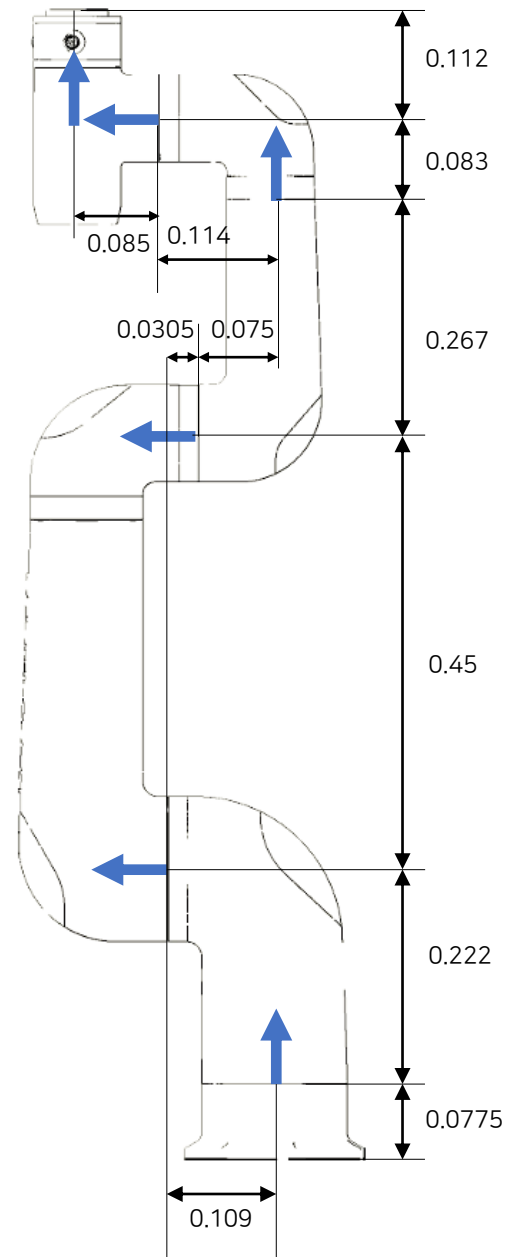
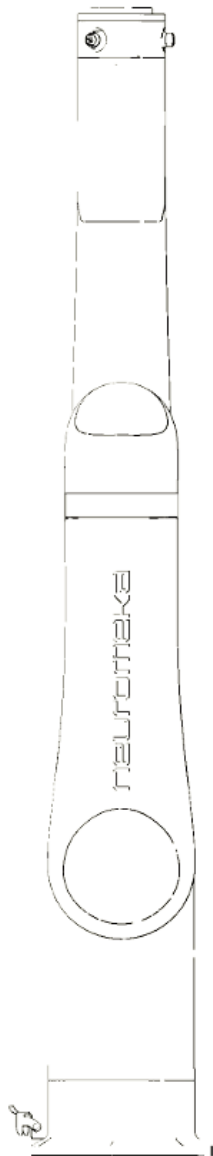
1. Implement FK of Indy7 robot using DH notation in Python
2. Compare the result of 1 with that of simulation(Pybullet)

Provided:

1. Pybullet skeleton code
2. Indy7 urdf file
3. Indy7 geometric parameters

If your work is not done until the end of today's class, that will be your homework due to April 17th (W)

Indy7



Overall structure of skeleton code

- ▼ Start simulation (Do not touch)
 - Import libraries
 - Numpy print options (precision)
 - Connect to GUI
 - GUI configurations
 - Simulation settings
 - Load plane
 - coordinate frame visualization function
 - Load robot
- ▼ Forward Kinematics
 - DH notation**
 - Functions required to get end-effector's transformation
- ▼ Test
 - Set target joint variables
 - Get desired transformation using DH method
 - Apply joint variables to robot and get current transformation

[View>Table of Contents](#)