KINOVA

DH Parameters & Forward Kinematics

Last update: 2021-04-21

Gen3 lite Cursus

Table of contents

Objectives **3**

About Gen3 lite **3**

TABLE 1: DH parameters of Gen3 lite 3

FIGURE 1: DH frames and dimensions 4

Theory **5**

Manipulations **5**

Part 1: Implementation of the forward kinematics 5

Part 2: Validation with the robot 6

TABLE 2: Angular positions to test 6

Acknowledgements **7**

Objectives

During this lab, we will build a model for the forward kinematics of Gen3 lite using the Denavit-Hartenberg (DH) parameters, a common systematic way of describing the geometry of a robot with just a handful of dimensions. Then, we will validate our model experimentally by sending commands to the robot.

About Gen3 lite

All the information required to build the kinematic model of Gen3 lite is available in the User Guide in the section *Guidance for advanced users*. For convenience, the relevant information is transcribed here. Table 1 contains the DH parameters and Figure 1 illustrates the coordinate systems and dimensions used to fill the table.

i	a_i (mm)	d_i (mm)	α_i (rad)	θ_i (rad)
0	0.0	(128.25 + 115.00)	$\pi/2$	θ_0
1	280.0	30.00	π	$\theta_1 + \pi/2$
2	0.0	20.00	$\pi/2$	$\theta_2 + \pi/2$
3	0.0	(140.00 + 105.00)	$\pi/2$	$\theta_3 + \pi/2$
4	0.0	(28.50 + 28.50)	$\pi/2$	$\theta_4 + \pi$
5	0.0	(105.00 + 130.00)	0.0	$\theta_5 + \pi/2$

TABLE 1: DH parameters of Gen3 lite

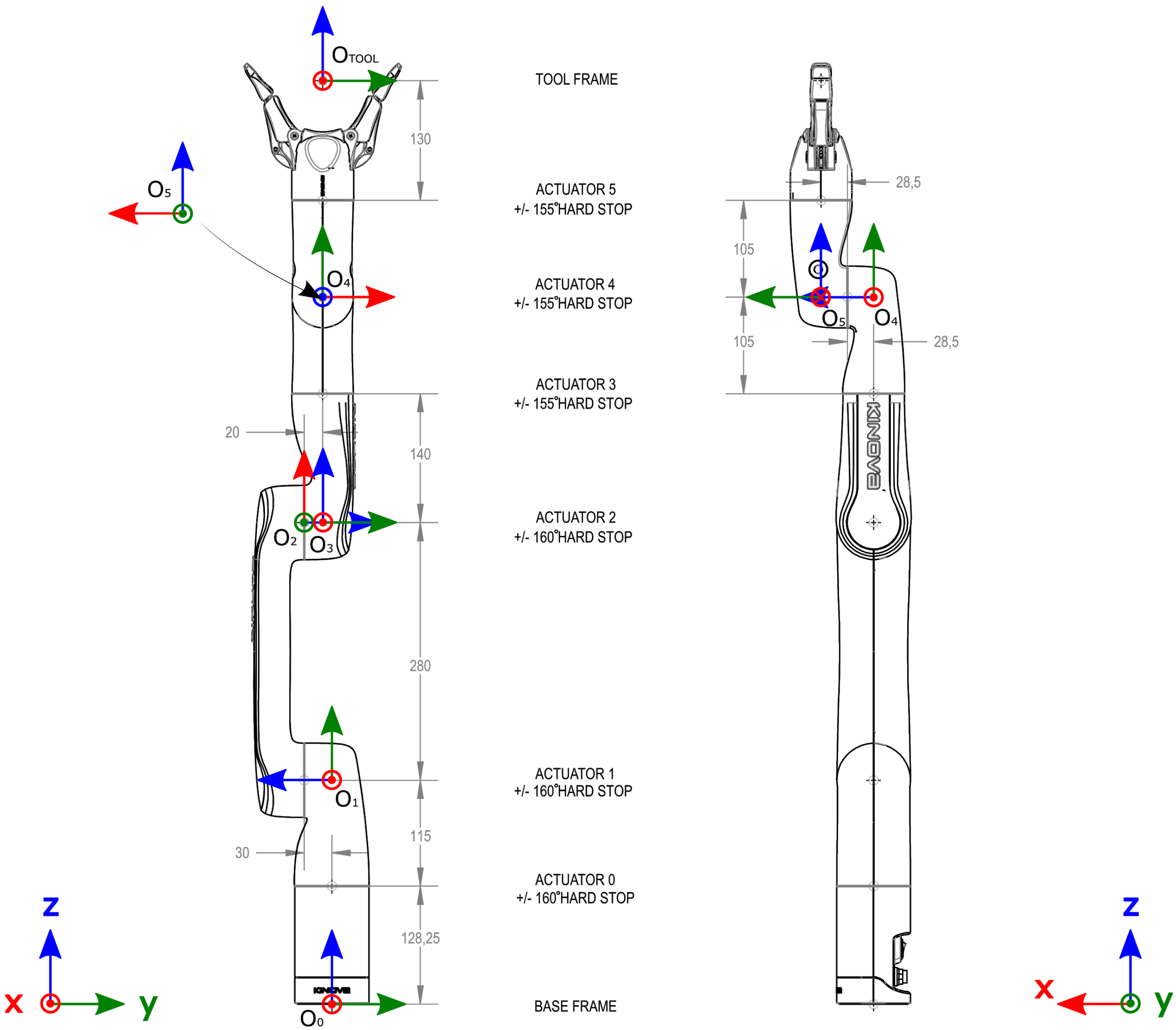

FIGURE 1: DH frames and dimensions (in mm)

Theory

The Classical DH parameters can be used to obtain the homogeneous transformation between two consecutive DH frames using the following definition:

$$^{i-1}T_i = \begin{bmatrix} cos(\theta_i) & -cos(\alpha_i)sin(\theta_i) & sin(\alpha_i)sin(\theta_i) & a_i cos(\theta_i) \\ sin(\theta_i) & cos(\alpha_i)cos(\theta_i) & -sin(\alpha_i)cos(\theta_i) & a_i sin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To obtain a full kinematics model, these transformation matrices can be chained to obtain the transform from the end-effector frame to the base frame.

Manipulations

Part 1: Implementation of the forward kinematics

In this section, we attach to each of Gen3 lite's link a DH frame of reference labeled from $\mathbf{O_0}$ to $\mathbf{O_5}$, with an additional frame $\mathbf{O_{tool}}$ located at the end-effector, as illustrated in Figure 1. We will use these frames and the parameters from Table 1 to implement the forward kinematics calculations in MATLAB.

1.1 Write a function called **transf_dh(**a_i, α_i, d_i, θ_i), which takes as inputs the parameters of one line of the DH parameter table and returns the corresponding homogeneous transformation.

1.2 Use the previous function to create a new function called **fk_gen3_lite_dh**(θ), which takes as an input a vector containing the position of each joint and returns the transformation matrix between the end-effector frame and the base frame.

1.3 Write a new function called **cart2pose**(x,y,z, θ_x, θ_y, θ_z) which takes as inputs the elements of the pose returned by Gen3 lite and returns the corresponding homogeneous transformation matrix.

Part 2: Validation with the robot

In this section, we will write a script to send angular commands to the robot and compare the cartesian pose calculated by the robot to the one obtained from our own calculation. You should reuse the code that we developed during the *Introduction to Gen3 lite* lab. As a reminder, connection to the robot is established via the **CreateRobotApisWrapper**() function, angular commands are sent using **ReachJointAngles**() and the cartesian pose readings are obtained via **RefreshFeedback**(). For more detail on the MATLAB API, consult the documentation.

Reminder: All angle values sent to or received from Gen3 lite should be expressed in degrees, and all lengths are expressed in meters.

2.1 Adapt your script to go to each of the angular configurations defined in Table 2 and record the cartesian pose returned by the robot. Convert these poses to homogeneous transformation matrices.

Test #	$\boldsymbol{\theta}$ (deg)
1	[0, 0, 0, 0, 0, 0,]
2	[90, 0, 0, 45, 45, 45]
3	[0, 344, 75, 0, 300, 0]
4	[7, 21, 150, 285, 340, 270]

TABLE 2: Angular positions to test

2.2 Use your **fk_gen3_lite_dh**($\boldsymbol{\theta}$) function developed in the previous section to compute the pose of the robot yourself and compare it to the results obtained experimentally. **Hint:** A good way to compare two rotation matrices is to take the dot product between corresponding columns. Similar orientations should yield dot products close to 1.

2.3 How do you explain the difference between the experimental results and what you obtained with your own calculations?

Acknowledgements

This document was produced in collaboration with the following establishments

Including direct implication from the following people:

- Prof. David Saussié, Eng., M.A.Sc., Ph.D., Department of Electrical Engineering.
- Alexis Tang, M.Eng., Department of Electrical Engineering
- Prof. Jérôme Le Ny, Eng., M.A.Sc., Ph.D., Department of Electrical Engineering.
- Prof. Richard Gourdeau , B.A.Sc., M.A.Sc., Ph.D., Department of Electrical Engineering.
