Milestone 1: Forward Kinematics

Robot Kinematics and Dynamics

Prof. Jeff Ichnowski
Kartik Agrawal
Aman Tambi

Contents

1 Overview

In this capstone, you will form a team of 3 students to program a Franka Emika Panda robot to draw with lights to create a long-exposure image.

2 Introduction

2.1 Notation and Terminology

Throughout this document we will use the following terms consistently:

Configuration (or config for short) specifies the settable degrees of freedom of a Robot. For Franka robots, this is a 7-element vector where the first coefficient is the angle of the first joint, etc.

Cartesian coordinates refers to the x, y, z position in workspace coordinates for translation, and a rotation relative to a fixed workspace identity rotation.

Position refers to the Cartesian translation of a an object.

Orientation refers to the rotation of an object.

Pose refers the the position and orientation of an object.

Linear interpolation given a start and end vector, this is an interpolation that starts at the start vector at $t = 0$ and ends at the end vector at $t = 1$. Mathematically $f(x_0, x_1, t) = x_0(1-t) + x_1 t$, for $x_0, x_1 \in \mathbb{R}^n$.

2.2 Picking / Grasping

The picking/grasping task in the capstone is the same as in the hands-on section of Assignment 7: moving the robot to a config where the target object is at the center of the gripper and closing the gripper.

2.3 Franka Kinematics

Most robot manufacturers, including Franka Robotics, use DH parameters to represent the robot kinematics. DH parameters of the Franka Emika Panda robot see Fig. 1 and Table 1. (Ref: Franka DH parameters, note that Panda and Research 3 have the same DH parameters.)

You will implement a forward kinematics function using the method introduced on Slide 16 of DH parameter slides. While the given DH parameters allow you to compute the transformation from the robot's base frame to its flange frame, you can implement your forward kinematics function to operate to any arbitrary point by applying additional transformations. For instance, the distance between the center of grasp and the flange frame is 0.1034 m in the z direction, so you can apply an additional transformation using $a = 0, d = 0.1034, \alpha = 0, \theta = 0$ to compute the transformation from the robot's base frame to its center of grasp. Similarly, you can also get the transformation from the robot's base frame to the tip of a grasped light source. Assuming you follow the slides, you will compute forward kinematic map in the form:

$$H_8^0 = H_1^0 H_2^1 \cdots H_8^7.$$

We will define $\text{fk}(q)$ to be the forward kinematic function that, given a robot configuration $q \in \mathbb{R}^7$, outputs H_0^8.

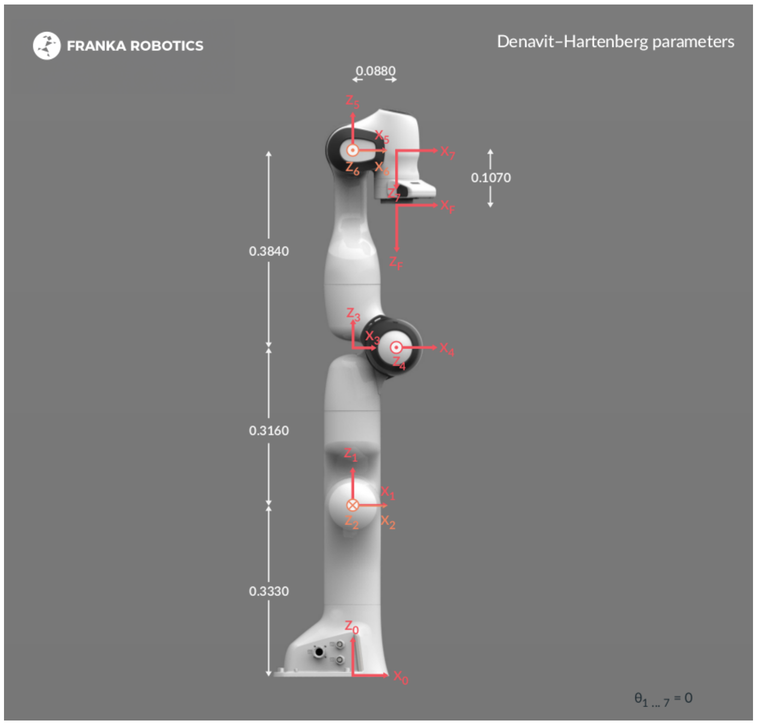

Figure 1: Franka Emika Panda robot joint frames

Joint	a_i (m)	α_i (rad)	d_i (m)	θ_i (rad)
1	0	0	0.333	θ_1
2	0	$-\pi/2$	0	θ_2
3	0	$+\pi/2$	0.316	θ_3
4	0.0825	$+\pi/2$	0	θ_4
5	-0.0825	$-\pi/2$	0.384	θ_5
6	0	$+\pi/2$	0	θ_6
7	0.088	$+\pi/2$	0	θ_7
Flange	0	0	0.107	0

Table 1: DH parameters of Franka Emika Panda robot

3 Milestone 1: Forward Kinematics & Robot

Due: **Thurs, Nov. 13, 2025**

This milestone is meant to (re)familiarize yourself with the Franka in 3D. Using the DH Parameters defined in Table 1 above, develop a forward kinematic map for the robot.

You will implementing this code in `milestone-1.py` The Forward Kinematic map will be a python function `forward_kinematics`, that takes a single argument: a list of the 7 joint angles. It should return the 3D homogeneous transform of the end effector. You will test this function with known configurations and end-effector poses.

Once the code works, you will then test it on the real robot. After putting the robot into free-drive or "teach" mode, validate that your forward kinematics works, by moving the robot to a known position, record the joint angles and validate that your forward kinematics map gives you the expected result.

The codebase has 4 files:

Also in teach mode, move the robot to get it as close as possible to a specific transform.

$$\begin{bmatrix} 1 & 0 & 0 & 0.53 \\ 0 & 1 & 0 & -0.21 \\ 0 & 0 & 1 & 0.42 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Use the `guide_mode.py` file to record the joint angles. Then move the robot into a different configuration while keeping the same end-effector pose and record another set of joint angles. (Hint: have one team member hold the end-effector still, while another moves the arm, and a third records the joint angles.)

Codebase Files

The codebase contains the following four files:

1. `guide_mode.py` This script allows you to interactively control the robot:
   - Press 1 to get the end-effector pose.
   - Press 2 to get the joint angles.
   - Press 3 to stop the current skill.
   - Press 4 to move the robot to the home position.
2. `FrankaRobot16384.py` This is a wrapper class that you will use for most of the capstone to access different functions of the robot.
3. `example.py` This file contains examples demonstrating how to use the wrapper class.
4. `milestone-1.py` This is where you will implement the forward kinematics for this milestone. Note: You do not need the other files to run this; you can use `guide_mode.py` to record poses to test your implementation.

4 Rubric

Your `forward_kinematics()` function will be evaluated in two parts: first using an autograder, and then through real-robot validation.

(1) Autograder Evaluation Your code will be tested automatically on a set of known joint configurations. Points will be assigned based on the number of test cases passed. Partial credit may be given for small numerical errors within tolerance. The autograder will check both the position and orientation accuracy of the end-effector.

(2) Real Robot Validation Once your forward kinematics function works correctly, you will validate it on the actual robot in *free-drive* or *teach* mode.

- Move the robot to a known position, record the joint angles using, and verify that your forward kinematics gives the expected end-effector pose.
- Move the robot into a different configuration while keeping the same end-effector pose, and record another set of joint angles. (Hint: one team member can hold the end-effector steady while others move the arm and record data.)
- Submit a PDF report including photographs of the robot in the tested poses and the recorded joint angles.

(3) Writeup

- Submit a PDF report including photographs of the robot in the tested poses and the recorded joint angles.

5 Submission Checklist

- ☐ Upload `all files` to Gradescope.