Assignment 7: Denavit-Hartenberg & Inverse Kinematics

Robot Kinematics and Dynamics
Prof. Jeff Ichnowski
Shahram Najam Syed
Yuemin Mao

Contents

1 Overview 3

2 Background 4
2.1 Parameterizations of 3D Rotations 4
2.2 Denavit-Hartenberg Parameters 4

3 Instructions 6

4 In Class Question 7

5 Written Questions 9

6 Coding Section 17

7 Hands-On Section 18

8 Submission Checklist 20

1 Overview

This assignment reinforces the following topics:

- Parameterizations of 3D Rotations
- Denavit-Hartenberg Parameters
- Inverse Kinematics

2 Background

2.1 Parameterizations of 3D Rotations

Along with the 3x3 rotation matrices, there exists the Euler angles, Roll-Pitch-Yaw, axis-angle, and quaternion conventions for representing 3D rotations. The quaternion convention is used frequently in robotics because it avoids the Euler angles' issues with gimbal lock, uses only 4 numbers as compared to the 9 in rotation matrices, and is overall more numerically stable and more efficient. The conversion between the axis-angle representation and quaternion representation is shown below:

$q_x = a_x * \sin(\frac{\theta}{2})$
$q_y = a_y * \sin(\frac{\theta}{2})$
$q_z = a_z * \sin(\frac{\theta}{2})$
$q_w = \cos(\frac{\theta}{2})$

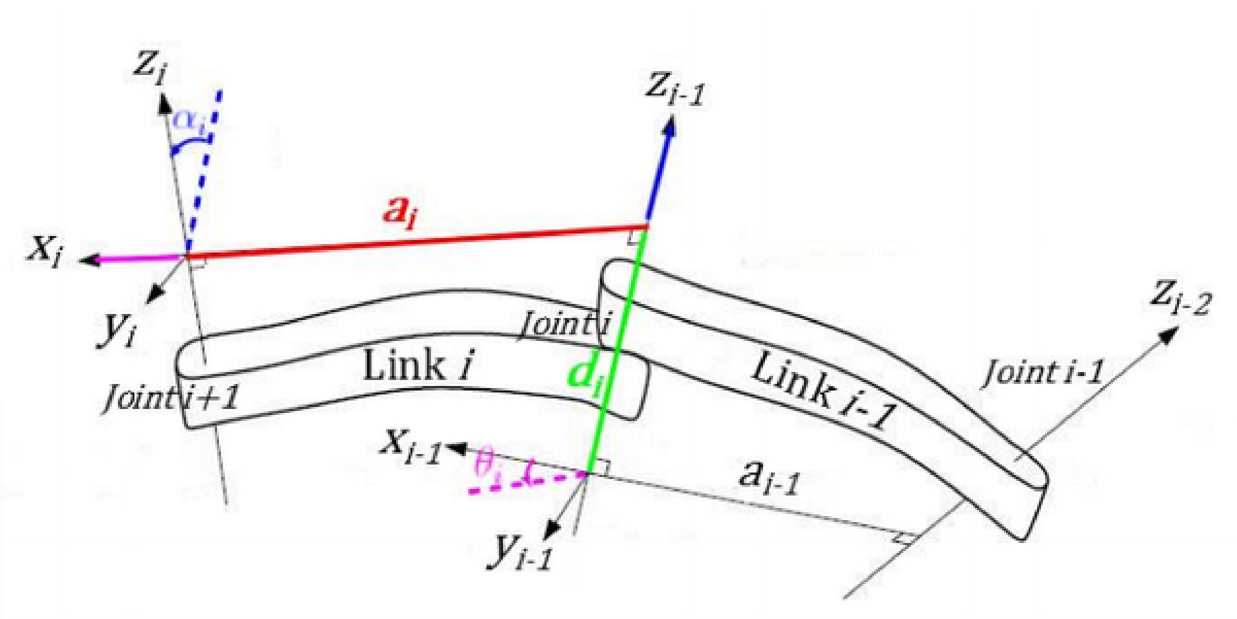
where:

the axis is normalized so: $a_x^2 + a_y^2 + a_z^2 = 1$

the quaternion is also normalized so $q_x^2 + q_y^2 + q_z^2 + q_w^2 = 1$

2.2 Denavit-Hartenberg Parameters

Recall that the Denavit-Hartenberg convention is a systematic and standard way for selecting reference frames for the links of a kinematic chain. There are 4 Denavit-Hartenberg parameters that are used to represent the transformation between the reference frames. We will briefly illustrate the parameters between again below:



a_i	α_i	d_i	θ_i
Link length	Link twist	Link offset	Joint angle

To construct the homogeneous transformation matrix between frames using the 4 D-H parameters, we utilize the following convention:

$$H_i^{i-1} = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3 Instructions

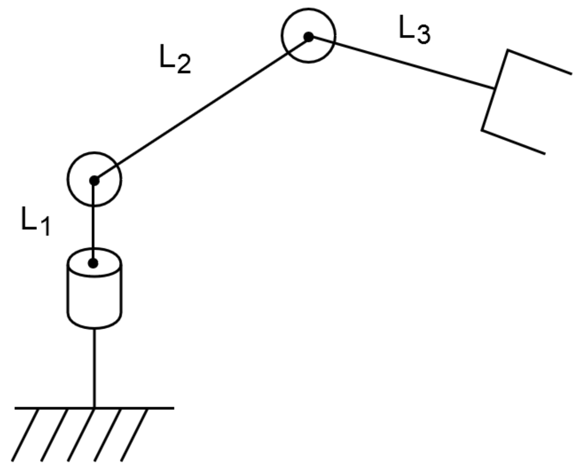
- The deadline for this project is 26th September, 2024 09:00 P.M.
- Zip your code into a single file named `<AndrewId>.zip`. See the complete submission checklist at the end, to ensure you have everything. Submit your PDF file to Gradescope.
- Each question (for points) is marked with a **points** heading.
- **Start early!** This homework may take a long time to complete.
- **During submission indicate the answer/page correspondence carefully when submitting on Gradescope.** If you skip a written question, just submit a blank page for it. This makes our work much easier to grade.
- If you have any questions or need clarifications, please post in Piazza or visit the TAs during the office hours.
- Unless otherwise specified, **all units are in radians, meters, and seconds**, where appropriate.

4 In Class Question

The following question will be done in class, as a part of a group. Your group's answer will still need to be turned in with the rest of your assignment, however unlike the rest of the work this is allowed to be done in groups.

1) Denavit-Hartenberg Practice

Please use the diagram of the arm below for the following question:



(1) [4 points] Label the X and Z axes for each reference frame using the Denavit-Hartenberg convention.

(2) [6 points] Determine the Denavit-Hartenberg Parameters for the robot above.

Link	a_i	α_i	d_i	θ_i
1				
2				
3				
4				

5 Written Questions

For all of the following questions, please fully compute all answers and show your work for full credit, unless otherwise specified. Submit your answers in a PDF entitled `writeup.pdf` in your handin directory. All answers must be typed, but diagrams may be hand-drawn and scanned in. However, they must be tidy and fully legible! Consider drawing them in a black or blue pen. All units are in radians and meters, where appropriate.

1) Parameterizations of Rotations Practice

Please convert between the different parameterizations of rotations below. **Show all of your work.**

(1) [8 points] Rotation Matrix: $R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$

ZYZ Euler Angles: $\phi =?, \theta =?, \psi =?$

Roll-Pitch-Yaw Angles: $Z =?, Y =?, X =?$

Axis-Angle: Axis = $[?, ?, ?]$, Angle = ?

Quaternion: $[?, ?, ?, ?]$

(2) [8 points] ZYZ Euler Angles: $\phi = \frac{\pi}{2}$, $\theta = -\frac{\pi}{2}$, $\psi = \pi$

Rotation Matrix:

$$R = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$

Roll-Pitch-Yaw Angles: $Z = ?, Y = ?, X = ?$

Axis-Angle: Axis = $[?, ?, ?]$, Angle = ?

Quaternion: $[?, ?, ?, ?]$

(3) [8 points] Roll-Pitch-Yaw Angles: $X = 0$, $Y = -\frac{\pi}{2}$, $Z = 0$

Rotation Matrix:

$$R = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$

Roll-Pitch-Yaw Angles: $Z =?, Y =?, X =?$

Axis-Angle: Axis = $[?, ?, ?]$, Angle = ?

Quaternion: $[?, ?, ?, ?]$

(4) [8 points] Axis-Angle: Axis = $[0, 0, 1]$ Angle = π

Rotation Matrix:

$$R = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$

Roll-Pitch-Yaw Angles: $Z = ?, Y = ?, X = ?$

Axis-Angle: Axis = $[?, ?, ?]$, Angle = ?

Quaternion: $[?, ?, ?, ?]$

(5) [8 points] Quaternion: $[0, 0, 0, 1]$

Rotation Matrix:

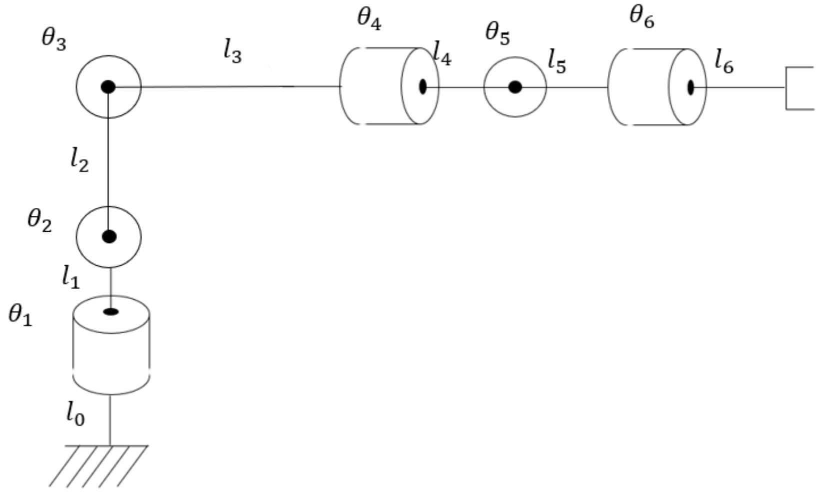
$$R = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$

Roll-Pitch-Yaw Angles: $Z = ?, Y = ?, X = ?$

Axis-Angle: Axis = $[?, ?, ?]$, Angle = ?

Quaternion: $[?, ?, ?, ?]$

2) Denavit-Hartenberg Parameters Practice



Side View of 6DOF

(1) [10 points] Label the X and Z axes for each reference frame using the Denavit-Hartenberg convention.

(2) [20 points] Determine the Denavit-Hartenberg Parameters for the robot.

Link	a_i	α_i	d_i	θ_i
1				
2				
3				
4				
5				
6				

(3) [12 points] Write down the Transformation Matrices between each link using the Denavit-Hartenberg Parameters you determined above.

6 Coding Section

In this section of the assignment, you are required to implement the dynamics of an inverted pendulum on a cart. The system's dynamics are governed by the equations you will derive from the theoretical model provided below.

Refer to the provided image of an inverted pendulum system for guidance on the physical setup. You will use this conceptual model to write a Python function that calculates the derivatives of the angle and angular velocity of the pendulum.

Tasks

1. Implement the `inverted_pendulum_dynamics` function in Python. This function should take the pendulum's angle (`theta`) and angular velocity (`omega`) as inputs and return their time derivatives.

2. Test your implementation by generating a phase portrait of the system. Ensure your plot clearly shows the behavior of the system across a range of initial conditions.

3. Include the generated plot in your report. An example placeholder for the plot is shown below.

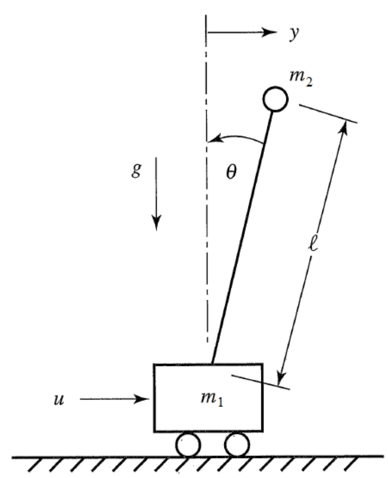


Figure 1: Illustration of an inverted pendulum on a cart.

7 Hands-On Section

For this lab, you will learn how to use the Frankapy API, and you must complete the tasks on real robots in REL.

Task: Jenga Block Pick-and-Place

In this section, you will pick up a Jenga block in the orange "Pick" area and drop it in the blue "Drop" area. See the step-by-step instructions below.

1. Stack the Jenga blocks in the orange "Pick" area as shown in Fig. 2.

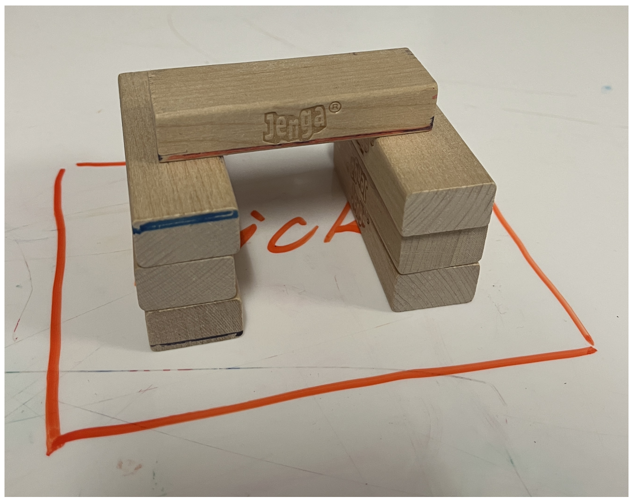


Figure 2: Jenga stack

2. Run `run_guide_mode_custom.py` and manually move the robot's end effector to a position above the Jenga stack. Record the joint angles `q_pre_pick`.
3. Run `run_guide_mode_custom.py` and manually move the robot's end effector to a position where it can grasp the top Jenga block. Record the joint angles `q_pick`.
4. Run `run_guide_mode_custom.py` and manually move the robot's end effector to a position above the blue "Drop" area. Record the joint angles `q_drop`.
5. Write a script to move the robot along the path: `q_home` → `q_pre_pick` → `q_pick` → `q_drop` → `q_home`. Remember to close or open the gripper at certain positions to pick or drop the Jenga block. All required functions can be found in `frankapy/frankapy/franka_arm.py`. Sample code see `sample_script.py` in the code handout. You can also find other examples in the `frankapy/examples` directory.
6. Run your code on the robot and record a video of the robot doing pick-and-place.

Demo see Jenga pick-and-place.

Submission

1. Upload your video to Youtube and attach the link in your writeup.
2. Include the script you write in your code submission.

8 Submission Checklist

- ☐ Upload `<andrew_id>.pdf` to Gradescope.
- ☐ Upload the code `ex1.py` as a zipped folder named `<andrew_id>.zip` to Gradescope.