Middle East Technical University Department of Computer Engineering

CENG 460

Introduction to Robotics

Fall 2019-2020

Homework 3 - Inverse Kinematics and Quaternion Interpolation

Due date: 25 12 2019, Wednesday, 23:59

1 Theory (30pts)

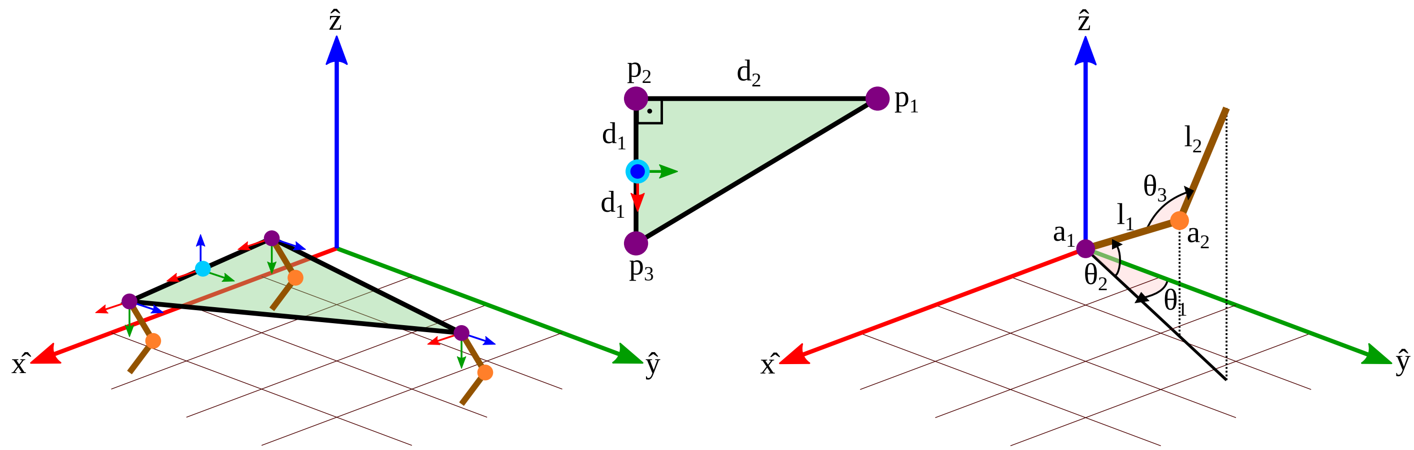

Figure 1: The Sphinx, with its leg design on the right and body plane shape in the middle. Notice that the local frames of the legs are oriented in a different way than the body frame of the robot.

(a) **Inverse Kinematics of the Sphinx Leg (15pts):** The sphinx leg has 2 joints a_1, a_2. The base joint a_1 is a spherical joint described with the angles $\theta_1,\ \theta_2$ with respect to the local frame of the arm as shown in Figure 1. The knee joint a_2 is described with the angle θ_3. a_2 constraints the motion of the links $l_1,\ l_2$ with length l such that their projections on the xy plane of the local frame always end up colinear. Find all possible angles $\theta_1,\ \theta_2,\ \theta_3$ for the tip of the leg to be at the coordinates $[x\ y\ z]$ relative to the local frame of the leg. The orientation of the tip does not matter. (**Hint:** This coordinate must lie in a particular sphere for a solution to exist.)

(b) **Quaternion Interpolation**

(a) **Angle between Quaternions (5pts):** Let q, q_h, q_k be unit quaternions with $q_h = q \circ q_k = h_1 < h_2, h_3, h_4 >$, $q_k = k_1 < k_2, k_3, k_4 >$ and $q = \cos\theta < \sin\theta\hat{v} >$, where $\hat{v}$ is a unit vector. Show that $cos(\theta) = \sum_{i=1}^{4} h_i k_i$ (A regular dot product!)

(b) **SLERP (10pts):** Let $\hat{v}_1, \hat{v}_2$ be linearly independent unit vectors pointing out from the global origin with angle θ from $\hat{v}_1$ to $\hat{v}_2$. Let $\hat{v}$ be the unit vector with angle $\alpha\theta$ measured from $\hat{v}_1$ to $\hat{v}$ ($\alpha \in [0, 1]$). Show that:

$$\hat{v} = \frac{\sin(\alpha\theta)}{\sin\theta}\hat{v}_2 + \frac{\sin((1-\alpha)\theta)}{\sin\theta}\hat{v}_1$$

(**Hint:** Try to represent $\hat{v}_2$ as a linear combination of orthonormal vectors, one being $\hat{v}_1$. The same representation will work for $\hat{v}$ also.) Curiously, the above equation works for the quaternions too, but that takes too many multiplications and simplifications to show.

2 Programming (70pts)

In this part, you are going to implement the stationary kinematics of Sphinx, a three legged robot with legs p_1, p_2, p_3 shown in Figure 1 with their own local frames. It has a body shaped like an upright triangle with sides of lengths $2d_1$ and d_2 with its center situated in the middle of one side.

2.1 The Leg (30pts)

Implement the forward and inverse kinematics of the `Leg` class in the given stub. For the IK to have a single solution and realism on the knee joint, $\theta_1, \theta_2 \in (-\frac{\pi}{2}, \frac{\pi}{2})$ and $\theta_3 \in (-\pi, 0)$.

2.2 The Sphinx (30pts)

Implement the class `Sphinx` so that the feet stay at global $z = 0$ when the body changes its transform.

2.3 Interpolation Algorithms (10pts)

Implement the interpolation functions. (SciPy interpolation API is obviously not allowed, but other things like representation conversions are fair, such as quaternion to rotation matrix etc.). Do not mind the degenerate cases like the orientations being opposite or the same for the quaternion SLERP.

Implementation Specifications

Implement the functions in the given stub file with Python3. You may use libraries other than NumPy, SciPy, and Matplotlib, but you need to get permission from the course assistant first (preferably by opening a topic at the ODTUCLASS forum). The code will be reviewed with white-box technique, therefore please **comment** your intents within the hard-to-understand parts of your code.

3 Submission

You can submit handwritten solutions for the Theory part, however they need to be scanned (to pdf format), **and the hardcopies should be delivered to B202 within the next day after submission**. Submit all your files, including the scans and the filled in stubs within a zipfile named `eXXXXXXX_hw3_460_2019f.zip` to ODTUCLASS.