Midterm - Group 1

Software Development for Robotics

October 17, 2023

Students:
Jerry Pittman, Jr. (117707120)

Aaqib Barodawala (119348710)

Vedant Ranade (119211588)

Instructors:
T. Chang

TA:
Jay Prajapati

Semester:
Fall 2023

Course code:
ENPM808X

Contents

1 Overview

We will be designing the Manipulator arm path planner that solves Inverse Kinematics (IK) for Acme. We plan on creating the design for an UR5 robot which has 6 degrees of freedom that are RRRRRR joints. This will be an open-source implementation so roboticists will not be required to use third-party libraries or middleware frameworks when controlling for UR5 and can be further used for other robots that have 6 degrees of freedom with RRRRRR type joints. The license we will be using most likely will be Apache based on common-math (cmath) library. We will ensure our design does not cause singularities and provides continuous (not jumpy) motion of the robot's manipulator. We will be using C++ programming language that can be integrated with ROS2 for visual simulation. Also, we will be using the C++ standard libraries including the cmath library specifically for calculations. Furthermore, we are building our code using Linux Ubuntu 22.04.

2 Algorithms or Techniques to be Used/Developed

We will be implementing the following algorithms for implementing Path Planning using Inverse Kinematics and matrix math: Denavit-Hartenberg (DH) Parameters for Forward Kinematics and Jacobian mathematics for Inverse Kinematics.

3 Risks/unknowns (technical, project) and Mitigations

Some unknowns is to ensure our unit tests had adequate coverage of the implementation and which parameters/methods will be private or public as well as whether we will need virtual methods. We also may need to fix/correct the class relationships/dependencies after we start implementing. We will be using methods of [2] for self-collision to aid us in creating our own methods for IK so that those features can be implemented in future works since we will not be implementing collision avoidance.

4 Final Deliverables to Acme

Our github repo is here which includes our QuadChart, UML/Activity Diagrams, and video. Our video and QuadChart are also located here.

5 How Pair Programming will be Executed and Documented

Vedant will be the UML/activity diagram creator and design keeper. He will also be the owner and maintainer of the GitHub Repository. Jerry and Aaqib will be using pair programming for the creation of the base class (based on the UML), unit tests, and stub implementation. Roles will then be swapped for the implementation (Phase 1 and phase 2 of the project). As each portion is completed Vedant will review the code and compare to the UML and make adjustments to the UML/activity diagrams (as needed) as the development progresses. Depending on availability, Vedant may act as Driver with Jerry or Aaqib as Navigator.

6 Unit Tests

Our unit test suites will test Forward Kinematics and Inverse Kinematics (for invertability and ensuring no singularity).

References

[1] T. Chang, "Enpm808x fall 2023 notes," University of Maryland - College Park, MAGE, College Park, MD, Tech. Rep. Lectures, Sep. 2023.

[2] Z. Liu, L. Zhang, X. Qin, and G. Li, "An effective self-collision detection algorithm for multi-degree-of-freedom manipulator," *Measurement Science and Technology*, vol. 34, no. 1, p. 015 901, Oct. 2022. DOI: 10.1088/1361-6501/ac9920. [Online]. Available: https://doi.org/10.1088/1361-6501/ac9920.

[3] V. Rajlich, *Software Engineering: The Current Practice*. Apr. 2016, pp. 1–281, ISBN: 9780429088759. DOI: 10.1201/b11678.