EE183DA (Winter 2018)

Design of Robotic Systems I

Prof. Ankur Mehta : mehtank@ucla.edu

Lab assignment 1
Lab report due 4pm Thursday Jan. 26, 2018

Lab Overview

Objectives

The goal of this lab is to understand forward and inverse kinematics of simple robotic designs. You will go out into the world and find a mechanical linkage that can be represented by a kinematic chain. You will extract the Denavit-Hartenberg parameters and use them to calculate the transformation between joint (configuration) space parameters and operational space parameters. You will then analyze that transformation to determine the inverse kinematics. Finally, you will implement the inverse kinematics and use your implementation to simulate a sample mission of the robot, assuming each joint is actuated.

Deliverables

This project will require you to write code. If you do not have one already, create an account on http://github.com, and create a project for this class. Make sure the commented code for this lab is committed and pushed, and submit a link to the repository. For some possible resources on git, see below.

You will write up your findings in the format of lab report. The sections of your report will likely follow the sections of this assignment sheet. Submit a .pdf of your report, which should include a link to your github repo, on CCLE by 4pm Thursday Jan. 26, 2018.

You may work in pairs on this assignment. Each person needs to write their own report, but the team can submit common code. Indicate on your report who you worked with, and for each person identify 1) the specific contributions made by each, and 2) an aggregate percentage of the total work done.

Submissions that are up to 24 hours late will be accepted for a 10 percentage point reduction in final grade. No submissions will be accepted more than 24 hours late.

1 Introduction

In this section, you will describe the problem being solved.

Find a real-world kinematic linkage that has 4 or more (1-DOF) joints, such that no two consecutive joints are degenerate — that is, of the same type with the same axis. Some examples of kinematic linkages include construction equipment, stands and mounts, or body parts. Be sure to include in your report a clear picture of this linkage.

Outline the design of this linkage in terms of its joint space: create a simplified schematic of the robot, and characterize all joints and links. Describe the intended functionality of this linkage in terms of it's operational space: explain the desired 6DOF state variables of the end effector. Choose a sample task to analyze, e.g. linear motion between two specified points in operational space.

2 Methods

In this section, you will compute the forward kinematics and implement inverse kinematics.

Determine the Denavit-Hartenberg parameters for the linkage. Define all variables used, and give (approximate) numerical values for the geometric dimensions. Clearly present your parameters. These D-H parameters can be used to generate the operational state variables of the end effectors as a function of the joint parameters of the robots. Implement these forward kinematics using your favorite programming environment.

Choose a method for computing the inverse kinematics of this model, generating joint parameters as a function of a desired output state. Implement this method.

3 Results

In this section, you will present and analyze the results of your computation/experiments.

Your experiments will relate to the sample task you presented in the introduction. Use your IK method to compute the configuration parameters of the endpoints of the desired motion. Use FK to generate a trajectory in operational space arising from a linear interpolation between these in configuration space. How does this resulting trajectory compare to straight-line motion in operational space? Can you use your methods to create a more accurate/desirable trajectory in operational space? Present these results using appropriate values, tables, and/or graphs.

Finally, for my benefit in assessing this class: How long did this lab take you? What sections did you find easiest? Hardest? Most fun? Least fun?

Git Resources

- Getting started with GitHub: `https://guides.github.com/activities/hello-world/`
- Detailed documentation on how to use git: `https://git-scm.com/book/en/v2`
- Try git in the browser: `https://try.github.io/levels/1/challenges/1`