UNIVERSITÀ DEGLI STUDI DI GENOVA

DIBRIS

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY,
BIOENGINEERING, ROBOTICS AND SYSTEM ENGINEERING

MODELLING AND CONTROL OF MANIPULATORS

Third Assignment

Jacobian Matrices and Inverse Kinematics

Professors:

Enrico Simetti
Giorgio Cannata

Author:

Surname Name

Student ID:

s0000000

Tutors:

Andrea Tiranti
Luca Tarasi
George Kurshakov

December 16, 2024

Contents

Mathematical expression	Definition	MATLAB expression
$< w >$	World Coordinate Frame	w
${}^{a}_{b}R$	Rotation matrix of frame $< b >$ with respect to frame $< a >$	aRb
${}^{a}_{b}T$	Transformation matrix of frame $< b >$ with respect to frame $< a >$	aTb
${}^{a}O_{b}$	Vector defining frame $< b >$ wit respect to frame $< a >$	aOb

Table 1: Nomenclature Table

1 Assignment description

The third assignment of Modelling and Control of Manipulators focuses on Inverse Kinematics (IK) control of a robotic manipulator.

The third assignment consists of three exercises. You are asked to:

- Download the .zip file called MCM_assignment3.zip from the Aulaweb page of this course.
- Implement the code to solve the exercises on MATLAB by filling in the predefined files. In particular, you will find two different main files: *"ex1.m"* for the first exercise and *"ex2.m"* for the second exercise.
- Write a report motivating your answers, following the predefined format on this document.
- **Putting code in the report is not an explanation!**

1.1 Exercise 1

Given the geometric model of an industrial manipulator used in the previous assignment, you have to add a tool frame. The tool frame is rigidly attached to the robot end-effector according to the following specifications:

Use the following specifications ${}^{e}\eta_{t/e} = [0, 0, \pi/10], {}^{e}O_t = [0.2, 0, 0]^\top(m)$ where ${}^{e}\eta_{t/e}$ represents the YPR values from end effector frame to tool frame.

To complete this task you should modify the class *geometricModel* by adding a new method called *getToolTransformWrtBase*

1.2 Exercise 2

Implement an inverse kinematic control loop to control the tool of the manipulator. You should be able to complete this exercise by using the MATLAB classes implemented for the previous assignment (*geometricModel*,*kinematicModel*), and also you need to implement a new class *cartesianControl* (see the template attached). The procedure can be split into the following phases

Q2.1 Compute the cartesian error between the robot end-effector frame ${}^{b}_{t}T$ and the goal frame ${}^{b}_{g}T$.

The goal frame must be defined knowing that:

- The goal position with respect to the base frame is ${}^{b}O_g = [-0.14, -0.85, 0.6]^\top$ (m)
- The goal frame is rotated by the YPR angles ${}^{b}\eta_g = [-3.02, -0.40, -1.33]^\top$ (rad) with respect to the base frame.

Q2.2 Compute the desired angular and linear reference velocities of the end-effector with respect to the base: ${}^{b}\nu^*_{t/b} = \begin{bmatrix} \kappa_a & 0 \\ 0 & \kappa_l \end{bmatrix} \cdot {}^{b}e$, such that $\kappa_a = 0.8$,$\kappa_l = 0.8$ is the gain.

Q2.3 Compute the desired joint velocities $\dot{q}$

Q2.4 Simulate the robot motion by implementing the function: *"KinematicSimulation()"* for integrating the joint velocities in time.

2 Exercise 1

[Comment] For the last exercises include an image of the initial robot image of the final robot configuration

[Comment] For each exercise report the results obtained and provide an explanation of the result obtained (even though it might seem trivial). The matlab code is NOT an explanation of the algorithm.

3 Appendix

[Comment] Add here additional material (if needed)

3.1 Appendix A

3.2 Appendix B