Università degli studi di Genova

DIBRIS

Department of Computer Science and Technology,
Bioengineering, Robotics and System Engineering

Modelling and Control of Manipulators

Exam session: 07/01/2024

Professors:

Enrico Simetti
Giorgio Cannata

Tutors:

Andrea Tiranti
Luca Tarasi
George Kurshakov

January 6, 2025

Contents

Mathematical expression	Definition	MATLAB expression
$< w >$	World Coordinate Frame	w
${}^a_bR$	Rotation matrix of frame $< b >$ with respect to frame $< a >$	aRb
${}^a_bT$	Transformation matrix of frame $< b >$ with respect to frame $< a >$	aTb
${}^aO_b$	Vector defining frame $< b >$ with respect to frame $< a >$	aOb
${}^b\eta_g$	Vector defining the orientation of frame $< b >$ with respect to frame $< a >$ given in terms of Yaw-Pitch-Roll angles	b_eta_g
${}^bv_{g/b}$	Velocity of frame $< g >$ with respect to frame $< b >$	bVg

Table 1: Nomenclature Table

1 Exam description

The Modeling and Control of Manipulator exam consists of a single exercise that summarizes all the topics discussed during the lectures. It should be solved using MATLAB, using the tools implemented during the practical lessons.

We have prepared a compressed folder in which you can find the code template, which consists of a $main.m$ and a folder $/include$. The main file should be compiled following the template. if it is completed correctly, the code can be launched and should output the results of the exam. The $/include$ folder contains the template of the MATLAB classes implemented during the practical lessons:

- *geometricModel*: which implements methods to calculate the direct geometry of the manipulator.
- *kinematicModel* which implements methods to calculate the Jacobian matrix of the manipulator.
- *cartesianControl*: which implements methods for the control of the manipulator.

You can add additional functions to the $/include$ folder. You are not bound to follow a standard template for the functions. Is it strictly forbidden to use built-in MATLAB functions instead of implementing the methods we have seen during the practical lessons.

2 Exercise

Given the geometric model of the industrial manipulator in Figure 1, you have to implement an inverse kinematic control loop to control the tool of the manipulator identified by the frame $< t >$. Consider that the tool frame is rigidly attached to the robot end-effector.The procedure can be split into the following phases:

Q1 Compute the Cartesian error between the robot tool frame ${}^b_tT$ and the goal frame ${}^b_gT$.
The goal frame must be defined knowing that:

- The goal position with respect to the base frame is ${}^bO_g = [-0.05; -1.2; 0.75]^\top(m)$.
- The goal frame is rotated by the YPR angles $\eta_{g/b} = [-\pi/8, 0, 0]^\top(rad)$ with respect to the base frame.
- The goal moves with velocity ${}^bv_{g/b} = [-0.01, 0, 0]^\top(\frac{m}{s})$.

Q2 The initial configuration of the robot must be set to $q = [0, -\pi/6, \pi/3, 0, \pi/2, \pi/2, 0]^\top$. Compute the desired angular and linear reference velocities of the end-effector with respect to the base: ${}^b\nu^*_{t/b} = \begin{bmatrix} \kappa_a & 0 \\ 0 & \kappa_l \end{bmatrix} {}^be$, where $\kappa_a = 0.4$,$\kappa_l = 0.4$ are the angular and linear gains.

Q3 Compute the desired joint velocities $\dot{q}$

Q4 Simulate the robot motion by implementing the function: *"KinematicSimulation()"* to integrate the joint velocities in time. We expect to sea the manipulator moving the tool frame, reaching the moving goal frame with no static error.

3 Appendix

[Comment] Add here additional material (if needed)

3.1 Appendix A

3.2 Appendix B

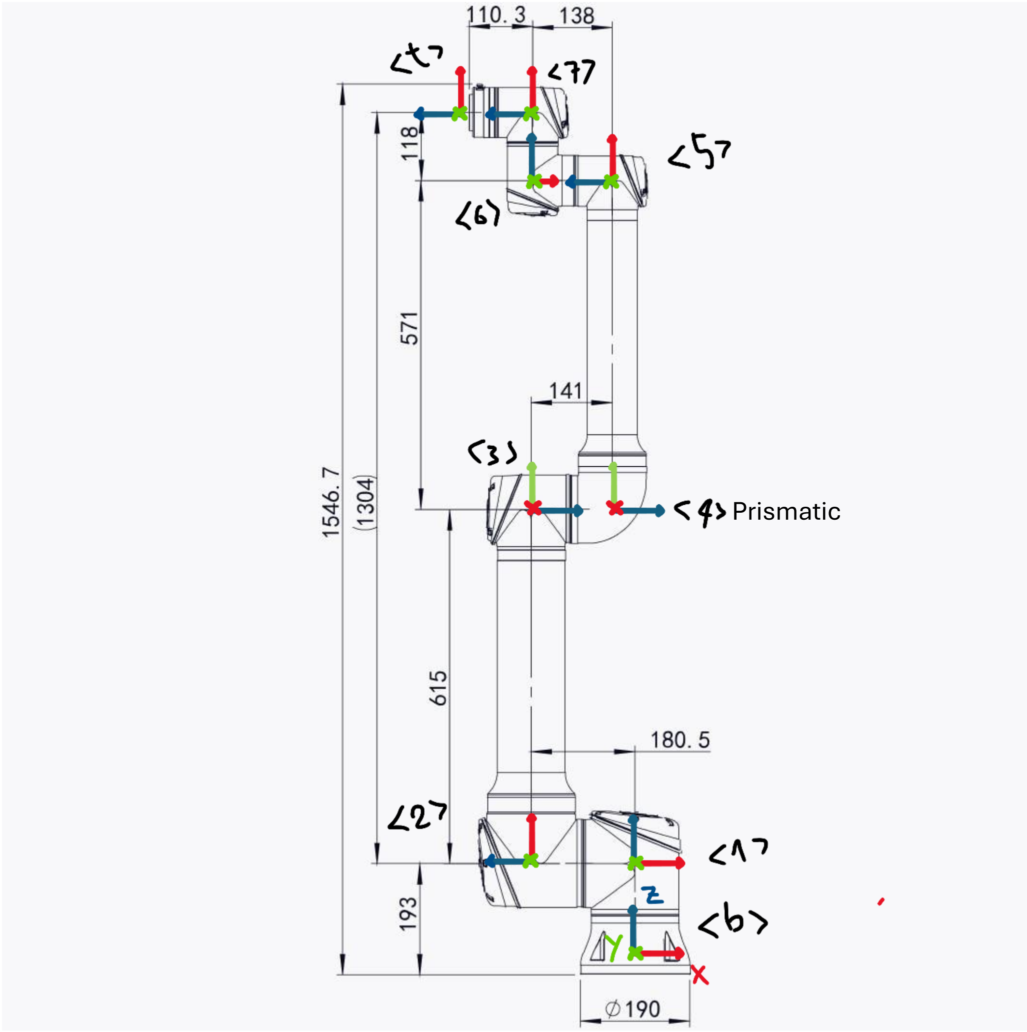

Figure 1: CAD model of the robot