UNIVERSITÀ DEGLI STUDI DI GENOVA

DIBRIS

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

MODELLING AND CONTROL OF MANIPULATORS

Exam session: 28/01/2024

Professors:

Enrico Simetti
Giorgio Cannata

Tutors:

Andrea Tiranti
Luca Tarasi
George Kurshakov

January 28, 2025

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 $< b >$ 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
${}^{b}v_{g/b}$	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 (the measurements are given in (mm)), 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 ${}^bO_g = [-1.3, 0, 1.65]^\top(m)$.
- The goal frame is rotated by the YPR angles $\eta_{g/b} = [0, 0, -\frac{3\pi}{5}]^\top(rad)$ with respect to the base frame.

Q2 The initial configuration of the robot must be set to $q = [0, \frac{\pi}{12}, \frac{\pi}{3}, \frac{\pi}{12}, \frac{\pi}{12}, \frac{\pi}{6}, 0]^\top(rad)$.
Develop a Cartesian Controller that regulates the distance to zero and the yaw-pitch-roll angles to zero.

Remark: The error must be defined using Euler angles. The desired angular reference velocites MUST be computed using the relationship between angular velocity vector and YPR derivatives. A solution based on the control of the angle-axis vector will be considered to be only partially correct.

Q3 Compute the inverse kinematics and hence 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 see the manipulator moving the tool frame, reaching the moving goal frame with no static error.

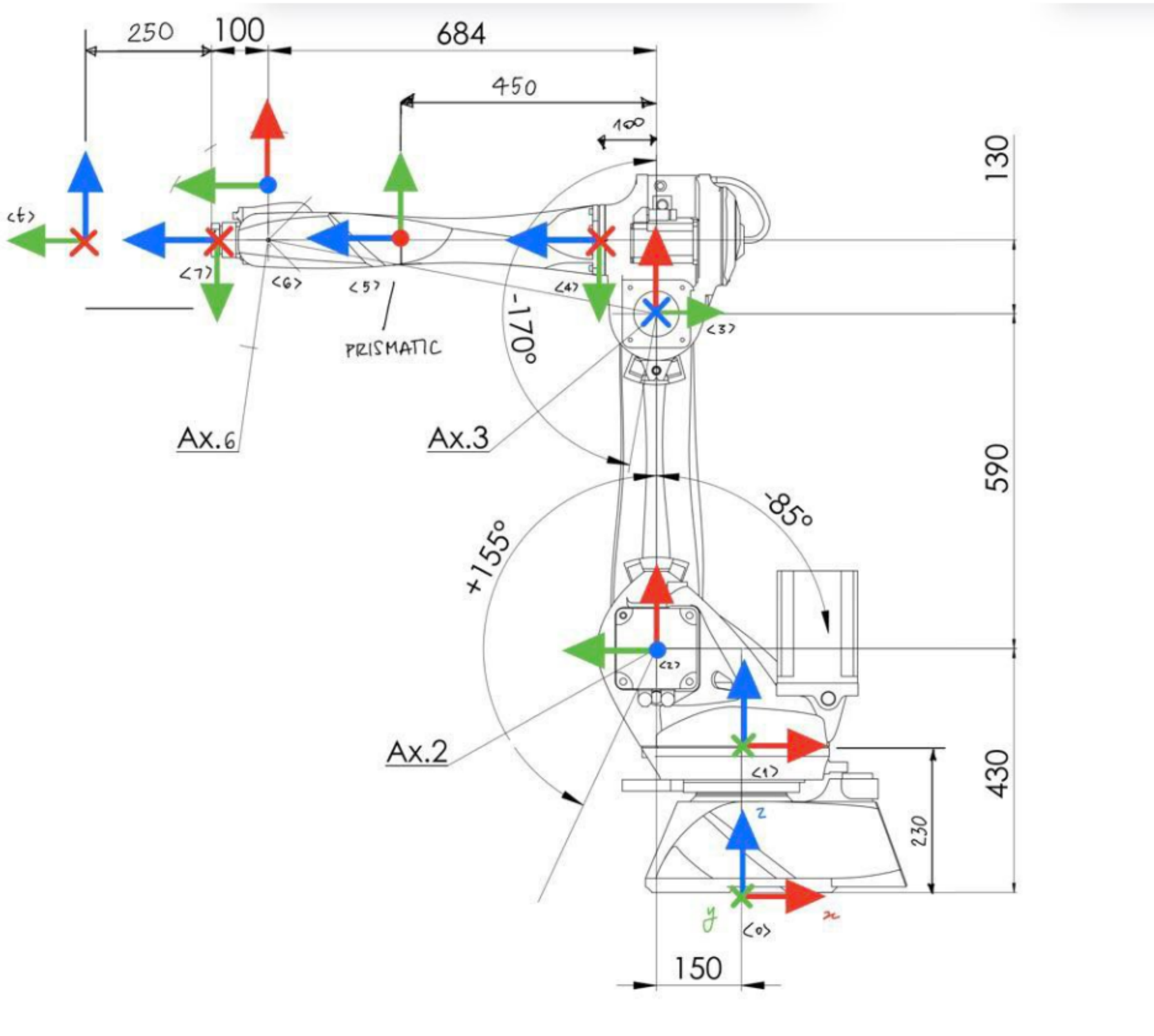

Figure 1: CAD model of the robot, the measurements are given in (mm)