UNIVERSITÀ DEGLI STUDI DI GENOVA

DIBRIS

DEPARTMENT OF INFORMATICS,
BIOENGINEERING, ROBOTICS AND SYSTEM ENGINEERING

MODELLING AND CONTROL OF MANIPULATORS

Second Assignment

Manipulator Geometry and Direct Kinematics

Author:

Surname Name

Student ID:

s0000000

Professors:

Enrico Simetti
Giorgio Cannata

Tutors:

Simone Borelli
Luca Tarasi

December 1, 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

Table 1: Nomenclature Table

1 Assignment description

The second assignment of Modelling and Control of Manipulators focuses on manipulators' geometry and direct kinematics.

- Download the .zip file from the Aulaweb page of this course.
- Implement the code to solve the exercises on MATLAB by filling the template classes called *geometricModel* and *kinematicModel*
- Write a report motivating your answers, following the predefined format on this document.

1.1 Exercise 1

Given the following CAD model of an industrial 7 DoF manipulator:

Q1.1 Define all the model transformation matrices, by filling the structures in the *BuildTree()* function. Be careful to define the z-axis coinciding with the joint rotation axis, such that the positive rotation is the same as showed in the CAD model you received; and the x-axis, when possible, pointing to the next link. Draw on the CAD model the reference frames for each link and insert it into the report.

Q1.2 Implement the method of *geometricModel* called *updateDirectGeometry()* which computes the model transformation matrices as a function of the joint position q. Explain the method used and comment on the results obtained.

Q1.3 Implement the method of *geometricModel* called *getTransformWrtBase()* which computes the transformation matrix from the base to a given frame. Using this method, compute the following transformation matrices: ${}^{b}_{e}T$, ${}^{6}_{2}T$. Explain the method used and comment on the results obtained.

Q1.4 Run the MATLAB section Q1.4 in the given script *main.m*, which computes the configurations of the manipulators moving from an initial joint position $q_i = [\frac{\pi}{4}, -\frac{\pi}{4}, 0, -\frac{\pi}{4}, 0, 0.15, \frac{\pi}{4}]^\top$ to $q_f = [\frac{5\pi}{12}, -\frac{\pi}{4}, 0, -\frac{\pi}{4}, 0, 0.18, \frac{\pi}{5}]^\top$. The script generates plots using methods from the given class *plotManipulators*, which must not be modified. Check that it behaves reasonably and show the obtained plots in the report.

Q1.5 Implement the method of *kinematicModel* called *getJacobianOfLinkWrtBase()* which takes as argument the link number, and returns the Jacobian of the corresponding link with respect to the base. Compute the Jacobian of **link 6** with respect to base for the final configuration q_f given in Q1.4. Explain the method used and comment on the result obtained.

Q1.6 Implement the method of *kinematicModel* called *updateJacobian()* which updates the Jacobian matrix of the manipulator for the end-effector. Compute the Jacobian for the final configuration q_f given in Q1.4. Explain the method used and comment on the results obtained.

Q1.7 Given the following joint position, $q = [0.7, -0.1, 1, -1, 0, 0.03, 1.3]^\top$ expressed in radians and meters, and the following joint velocities $\dot{q} = [0.9, 0.1, -0.2, 0.3, -0.8, 0.5, 0]^\top$ expressed in rad/s and m/s, compute the velocities of the end effector with respect to the base projected on the end effector frame, ${}^{e}v_{e/b}$ and ${}^{e}\omega_{e/b}$, using **forward kinematics**.

Remark: All the methods must be implemented for a generic serial manipulator. For instance, joint types, and the number of joints should be parameters.

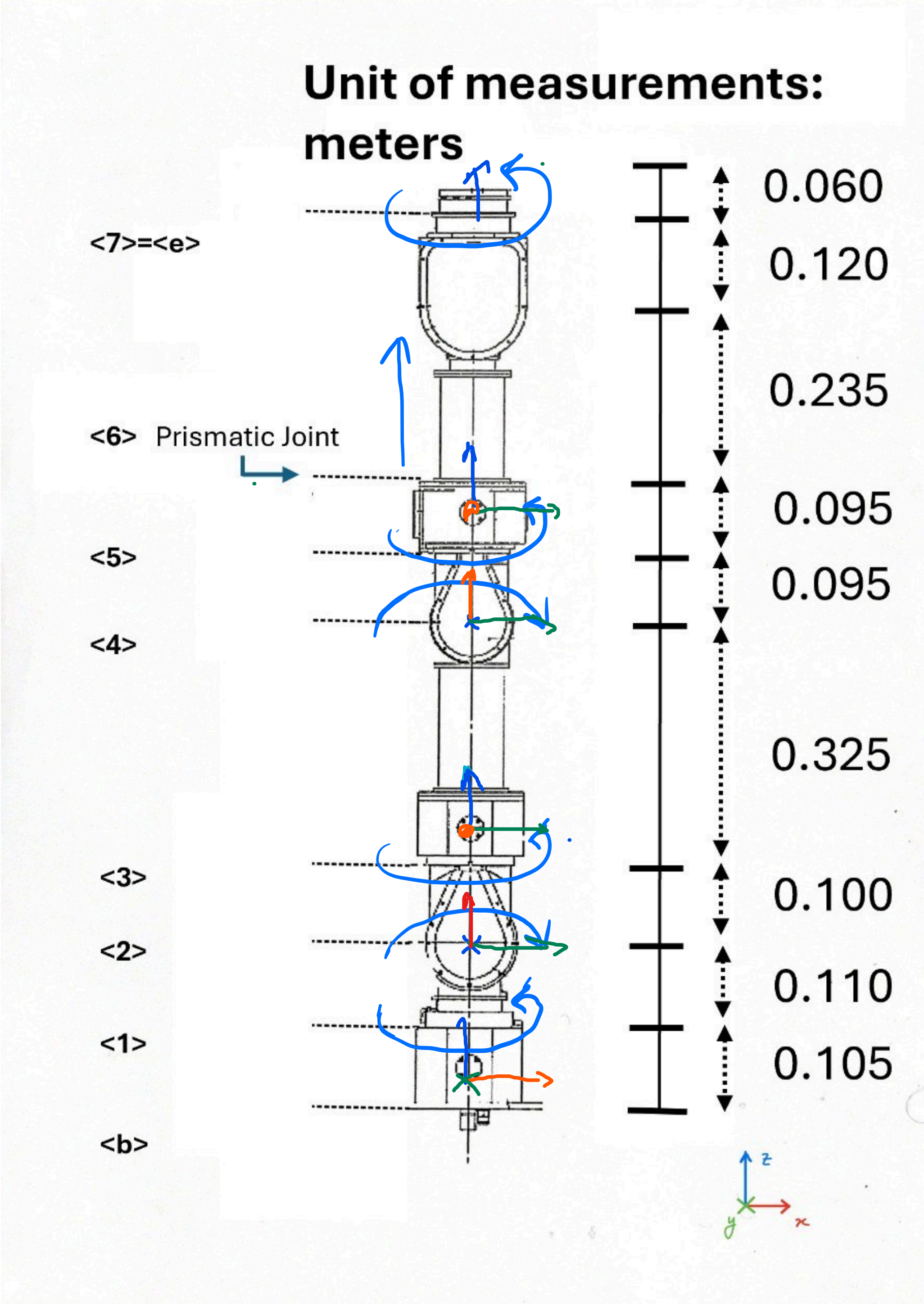

Figure 1: CAD model of the robot. The directions of motion of each joint are indicated by the blue arrows.

2 Exercise 1