UNIVERSITÀ DEGLI STUDI DI GENOVA

DIBRIS

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

MODELLING AND CONTROL OF MANIPULATORS

Second Assignment

Manipulator Geometry and Direct Kinematics

Author:

Ashoori Sarvenaz
Khadka Chhetri Rubin

Student ID:

s6878764
s6558048

Group:

4

Professors:

Enrico Simetti
Giorgio Cannata

Tutors:

Andrea Tiranti
Luca Tarasi
George Kurshakov

December 5, 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

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 called *template_MATLAB-assignment2* 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 matrices, by filling the structures in the *BuildTree()* function. Be careful to define the z-axis coinciding with the joint rotation axis, and such that the positive rotation is the same as showed in the CAD model you received. 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 should compute the model 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 should compute the transformation matrix from the base to a given frame. Calculate the following transformation matrices: ${}^b_eT$, ${}^5_3T$. Explain the method used and comment on the results obtained.

Q1.4 Implement the method of *kinematicModel* called *updateJacobian()* which should compute the jacobian of a given geometric model considering the possibility of having *rotational* or *prismatic* joints. Compute the Jacobian matrix of the manipulator for the end-effector. Explain the method used and comment on the results obtained.

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

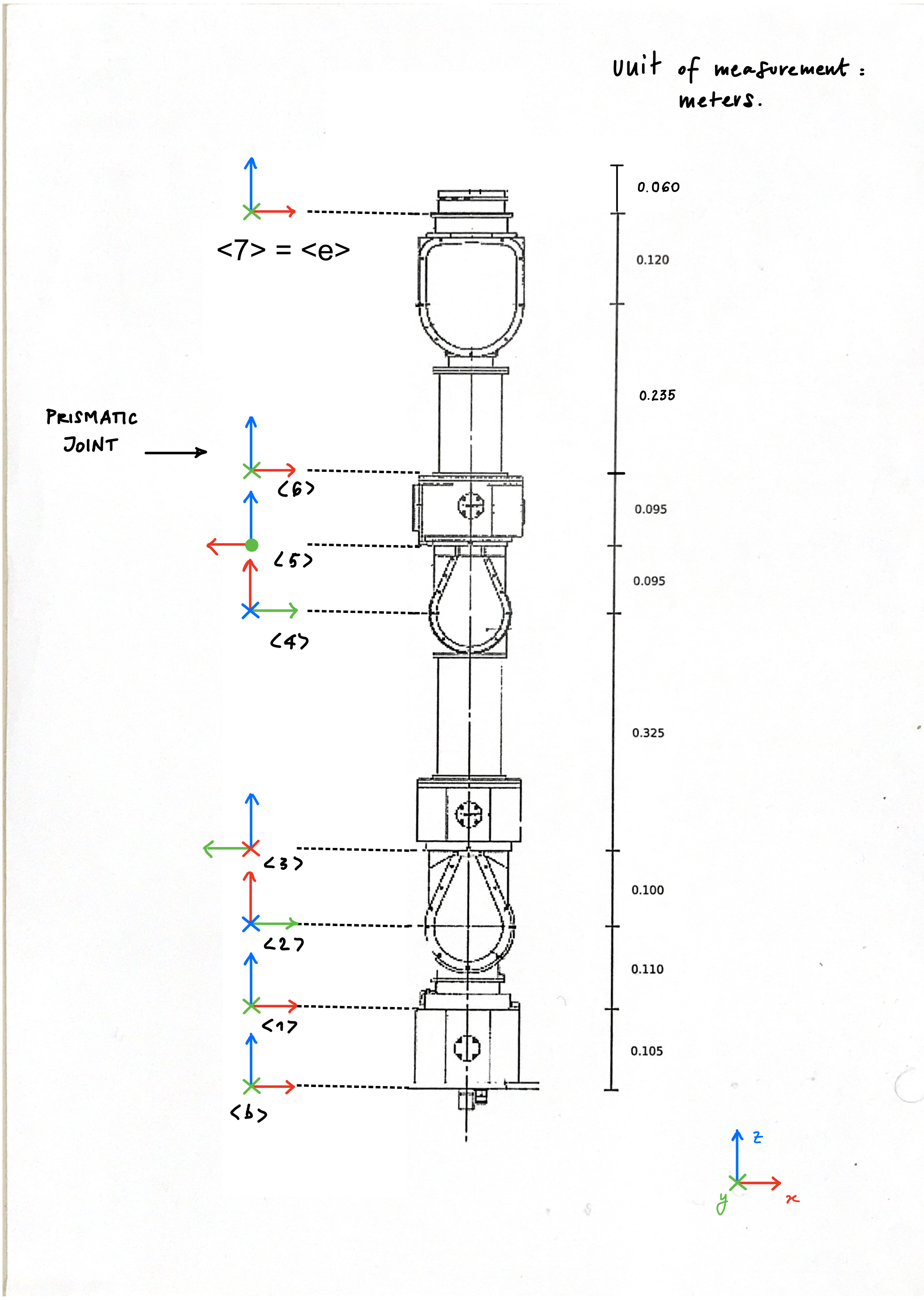

Figure 1: CAD model of the robot

2 Exercise 1

This assignment focuses on the geometric modeling and direct kinematics of a 7-DOF manipulator. The main objectives include defining the transformation matrices for the manipulator in its zero configuration and computing them as a function of joint positions. Additionally, transformations from the base to specific joint frames are calculated to analyze the spatial relationships between different parts of the manipulator. Finally, the Jacobian matrix for the end-effector is derived, which is essential for understanding the manipulator's motion capabilities. These computations provide the foundation for trajectory planning and control within the scope of kinematics.

2.1 Q 1.1: Transformation Matrices in Zero Configuration

To analyze the geometry of the manipulator in its zero configuration, transformation matrices between consecutive frames were derived from the provided CAD model. The zero configuration refers to the default pose of the manipulator, with all joint variables q set to zero. These matrices define the spatial relationships between adjacent links and are essential for kinematic analysis.

In MATLAB, the function `BuildTree()` is used to define these transformation matrices between consecutive frames in zero configuration. However, the transformation matrices must be manually filled in the function before it can be used for other kinematic calculations.

2.1.1 Methodology for Deriving Transformation Matrices

The transformation matrices ${}^{i}_{j}T_0$ represent the transformation of frame j relative to frame i in zero configuration, incorporating both rotation and translation. Note that frame j corresponds to the next frame relative to frame i (i.e., frame j is the frame $i+1$). The derivation follows these steps:

1. **Frame Orientation and Offsets:** The provided CAD model of the 7-DOF robot, shown in Figure 1, specifies the orientations and distances (offsets) between consecutive frames. This information is used to calculate the relative rotation matrices and position vectors between the frames.
2. **Rotation Matrices:** The rotation matrix ${}^{i}_{j}R$ defines the relative orientation between frames i and j, describing how the coordinate axes of frame j are oriented relative to frame i.
3. **Position Vectors:** The position vector ${}^{i}_{j}r$ represents the displacement between frames along the x, y, or z-axes. These vectors are derived from the distances specified in the CAD model.
4. **Homogeneous Transformation Matrices:** The transformation matrix ${}^{i}_{j}T_0$ is a 4x4 matrix combining rotation and translation:

$$ {}^{i}_{j}T_0 = \begin{bmatrix} {}^{i}_{j}R & {}^{i}_{j}r \\ 0 & 1 \end{bmatrix} $$

   where:

   - ${}^{i}_{j}R$ is the 3x3 rotation matrix,
   - ${}^{i}_{j}r$ is the 3x1 translation vector,
   - The last row $[0\ 0\ 0\ 1]$ ensures that the matrix is homogeneous.

2.1.2 Transformation Matrices

The transformation matrices ${}^{i}_{j}T_0$ for each joint in the zero configuration are as follows:

Transformation of frame 1 relative to frame b

$$ {}^{b}_{1}T_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.105 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Transformation of frame 2 relative to frame 1

$$ {}^{1}_{2}T_0 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0.110 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Transformation of frame 3 relative to frame 2

$$ {}^{2}_{3}T_0 = \begin{bmatrix} 0 & 0 & 1 & 0.100 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Transformation of frame 4 relative to frame 3

$$ {}^{3}_{4}T_0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0.325 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Transformation of frame 5 relative to frame 4

$$ {}^{4}_{5}T_0 = \begin{bmatrix} 0 & 0 & 1 & 0.095 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Transformation of frame 6 relative to frame 5

$$ {}^{5}_{6}T_0 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0.095 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Transformation of frame 7 relative to frame 6

$$ {}^{6}_{7}T_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.355 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

These transformation matrices describe the relative positions and orientations between adjacent frames in the 7-DOF manipulator. They form the foundation for subsequent tasks such as forward kinematics and Jacobian computation, which are essential for analyzing the manipulator's motion capabilities.

2.2 Q1.2: Transformation Matrices Based on Joint Positions

2.3 Q1.3: Transformation Matrices from Base to a Given Joint Frame

2.4 Q1.4: Jacobian Computation for the End-Effector

3 Appendix

[Comment] Add here additional material (if needed)

3.1 Appendix A

3.2 Appendix B