Università degli studi di Genova

DIBRIS

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

Modelling and Control of Manipulators

First Assignment

Equivalent representations of orientation matrices

Author:

Surname Name

Student ID:

s0000000

Professors:

Giovanni Indiveri
Enrico Simetti
Giorgio Cannata

Tutors:

Andrea Tiranti
Georgii Kurshakov
Luca Tarasi

October 18, 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 first assignment of Modelling and Control of Manipulators focuses on the geometric fundamentals and algorithmic tools underlying any robotics application. The concepts of transformation matrix, orientation matrix and the equivalent representations of orientation matrices (Equivalent angle-axis representation and Euler Angles) will be reviewed.

The first assignment is **mandatory** and consists of 5 different exercises. You are asked to:

- Download the .zip file called MCM-LAB1 from the Aulaweb page of this course.
- Implement the code to solve the exercises on MATLAB by filling the predefined files called "*main.m*", "*AngleAxisToRot.m*", "*RotToAngleAxis.m*", "*YPRToRot.m*" and "*RotToYPR.m*".
- Write a report motivating the answers for each exercise, following the predefind format on this document.

1.1 Exercise 1 - Angle-Axis to Rotation Matrix

A particularly interesting minimal representation of 3D rotation matrices is the so-called angle-axis representation, where a rotation is represent by the axis of rotation **h** and the angle θ. Any rotation matrix can be represented by its equivalent angle-axis representation by applying the Rodrigues Formula.

Q1.1 Given an angle-axis pair $(\mathbf{h}, \theta)$, implement on MATLAB the Rodrigues formula, computing the equivalent rotation matrix, **WITHOUT** using built-in matlab functions. The function signature will be

function R = AngleAxisToRot(h,theta)

Then test it for the following cases and briefly comment the results obtained:

- **Q1.2** $\mathbf{h} = [1, 0, 0]^T$ and $\theta = 90°$
- **Q1.3** $\mathbf{h} = [0, 0, 1]^T$ and $\theta = \pi/3$
- **Q1.4** $\rho = [-\pi/3, -\pi/6, \pi/3];$

Note that $\rho = \mathbf{h}\theta$.

1.2 Exercise 2 - Rotation Matrix to Angle-Axis

Given a rotation matrix R, the problem of finding the corresponding angle-axis representation $(\mathbf{h}, \theta)$ is called the Inverse Equivalent Angle-Axis Problem.

Q2.1 Given a rotation matrix R, implement on MATLAB the Equivalent Angle-Axis equations **WITHOUT** using built-in matlab functions. The function signature will be

function [h,theta] = RotToAngleAxis(R)

You **MUST** check that the input is a valid rotation matrix. Test it for the following cases and briefly comment the results obtained:

- **Q2.2** $R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$
- **Q2.3** $R = \begin{pmatrix} 0.5 & -\sqrt{3}/2 & 0 \\ \sqrt{3}/2 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- **Q2.4** $R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- **Q2.5** $R = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

1.3 Exercise 3 - Euler Angles to Rotation Matrix

Any orientation matrix can be expressed in terms of three elementary rotations in sequence. Consider the Yaw Pitch Roll (YPR) representation, where the sequence of the rotation axes is Z-Y-X.

Q3.1 Given a triplet of YPR angles (ψ, θ, ϕ), compute the equivalent rotation matrix representation **WITHOUT** using built-in matlab functions. The function signature will be

function R = YPRToRot(psi, theta, phi)

Then test it for the following cases and briefly comment the results obtained:

- **Q3.2** $\psi = \theta = 0,\, \phi = \pi/2$
- **Q3.3** $\phi = \theta = 0,\, \psi = 60°$
- **Q3.4** $\psi = \pi/3,\, \theta = \pi/2,\, \phi = \pi/4$
- **Q3.5** $\psi = 0,\, \theta = \pi/2,\, \phi = -\pi/12$

1.4 Exercise 4 - Rotation Matrix to Euler Angles

Given a rotation matrix R, it is possible to compute an equivalent triplet of YPR angles (ψ, θ, ϕ), provided that the configuration is not singular (that is, $\cos\theta \neq 0$).

Q4.1 Given a rotation matrix R, implement in MATLAB the equivalent YPR angles, **WITHOUT** using built-in matlab functions. The function signature will be

function [psi,theta,phi] = rotToYPR(R)

You **MUST** check that the input is a valid rotation matrix. Test it for the following cases and briefly comment the results obtained:

- **Q4.2** $R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$
- **Q4.3** $R = \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- **Q4.4** $R = \begin{pmatrix} 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0.5 & \frac{\sqrt{2}\sqrt{3}}{4} & \frac{\sqrt{2}\sqrt{3}}{4} \\ -\frac{\sqrt{3}}{2} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} \end{pmatrix}$

1.5 Exercise 5 - Frame tree

Figure 1 shows the frame tree for the 7 joints of the Franka robot. With reference to the figure, use the geometric definition of the transformation matrix to compute by hand the following matrices.

- **Q5.1** ${}^{0}_{1}T$
- **Q5.2** ${}^{1}_{2}T$
- **Q5.3** ${}^{2}_{3}T$
- **Q5.4** ${}^{3}_{4}T$
- **Q5.5** ${}^{4}_{5}T$
- **Q5.6** ${}^{5}_{6}T$
- **Q5.7** ${}^{6}_{7}T$
- **Q5.8** ${}^{7}_{e}T$

You **MUST** compute the matrices **WITHOUT** using mathematical software.

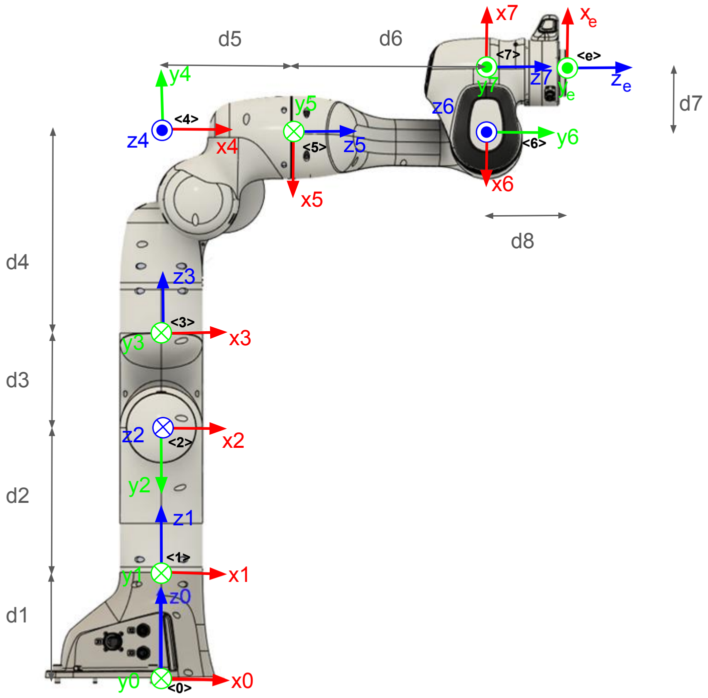

Figure 1: exercise 5 frames

2 Exercise 1

2.1 Q1.1

[Comment] Briefly explain how the MATLAB function was implemented.

2.2 Q1.2

2.3 Q1.3

2.4 Q1.4

[Comment] For each exercise report the results obtained and provide an explanation of the result obtained (even though it might seem trivial)

3 Exercise 2

[Comment] Same structure of exercise 1

4 Exercise 3

[Comment] Same structure of exercise 1

5 Exercise 4

[Comment] Same structure of exercise 1

6 Exercise 5

6.1 Q5.1

$${}^0_1T = \begin{pmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{pmatrix}$$

[Comment] Same structure for the other matrices

7 Appendix

[Comment] Add here additional material (if needed)

7.1 Appendix A

7.2 Appendix B