Università di Genova

DIBRIS

DEPARTMENT OF INFORMATICS,
BIOENGINEERING, ROBOTICS AND SYSTEM ENGINEERING

MODELLING AND CONTROL OF MANIPULATORS

First Assignment

Equivalent representations of orientation matrices

Authors:

Magnasco Chiara
Masaniello Chiara
Polese Carolina

Students ID:

s5657434
s8434670
s4862903

Professors:

Enrico Simetti
Giorgio Cannata

Tutors:

Luca Tarasi
Simone Borelli

November 11, 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

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*", "*RotToYPR.m*" and "*IsRotationMatrix.m*".
- Write a report motivating the answers for each exercise, following the predefind format on this document.
- The usage of built-in MATLAB functions is strictly forbidden except for basic mathematical operations such as *det*, *eig*.

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** $\rho = [-\pi/3, -\pi/6, \pi/3];$

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 by filling in and utilizing the function *IsRotationMatrix(R)*.
Hint: utilize a suitable tolerance (e.g. 10^{-3}) to check the properties of the matrices.
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** $R = \begin{pmatrix} \frac{1}{2} & \frac{-\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- **Q2.4** $R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

- **Q2.5** $R = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- **Q2.6** $R = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

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 by filling in and utilizing the function *IsRotationMatrix(R)*. **Hint:** utilize a suitable tolerance (e.g.10^{-3}) to check the properties of the matrices.
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 - Rotation Matrix to Angle-Axis using Eigenvectors

Consider a rotation matrix R and the corresponding angle-axis representation ($\mathbf{h}$, θ). Then, $R\mathbf{h} = \mathbf{h}$, which can be verified by computing $R\mathbf{h}$ with the Rodrigues Formula and observing that the terms containing $\mathbf{h} \times \mathbf{h}$ vanish. Therefore, $\mathbf{h}$ is an eigenvector of R corresponding to the eigenvalue +1.

Given the following rotation matrices, compute the corresponding angle-axis representation with two different methods. In the first method, compute $(\mathbf{h}, \theta)$ using the function *RotToAngleAxis* implemented before. In the second one, compute $\mathbf{h}$ as the eigenvector of R corresponding to eigenvalue +1, and then use the function *RotToAngleAxis* to compute θ. Compare and briefly discuss the results.

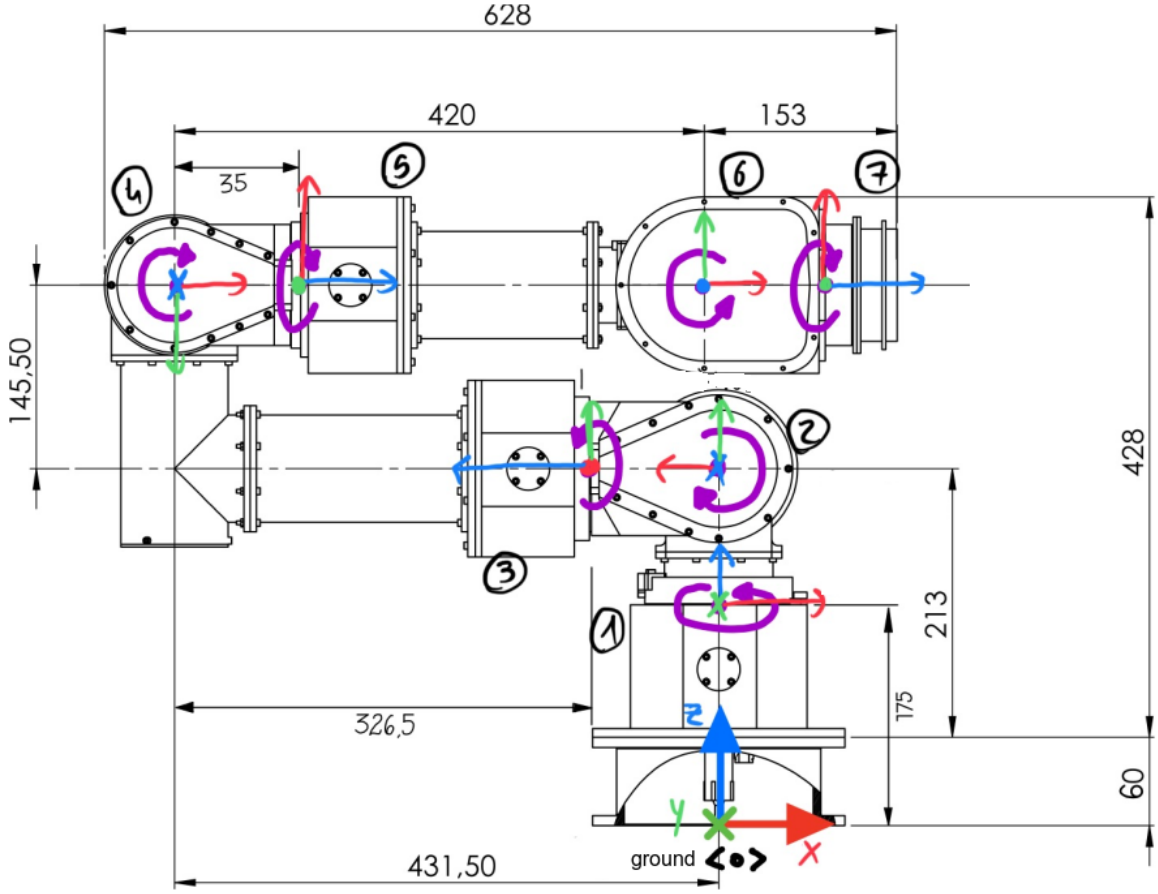

Figure 1: Exercise 6 frames. The unit of measurement is millimeters.

- **Q5.1** $R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$

- **Q5.2** $R = \frac{1}{9}\begin{pmatrix} 4 & -4 & -7 \\ 8 & 1 & 4 \\ -1 & -8 & 4 \end{pmatrix}$

1.6 Exercise 6 - 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.

- **Q6.1** ${}^0_1T$
- **Q6.2** ${}^1_2T$
- **Q6.3** ${}^2_3T$
- **Q6.4** ${}^3_4T$
- **Q6.5** ${}^4_5T$
- **Q6.6** ${}^5_6T$
- **Q6.7** ${}^6_6T$
- **Q6.8** ${}^7_6T$

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

2 Exercise 1

In this first exercise, the request was to find the rotation matrices $\mathbf{R} \in \mathbb{R}^{3\times3}$ using the Rodrigues formula:

$$\mathbf{R} = \mathbf{I}_{3\times3} + \sin\theta[\mathbf{h}\times] + (1-\cos\theta)[\mathbf{h}\times]^2 \tag{1}$$

Where:

- $\mathbf{I}_{3\times3}$ is the identity matrix,

$$\mathbf{I}_n = [\delta_{ij}]_{n\times n} \quad \text{with} \quad \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad and \quad \mathbf{I} \in \mathbb{R}^{3\times3}.$$

- $[\mathbf{h}\times]$ is the skew-symmetric matrix

$$[\mathbf{h}\times] = \begin{pmatrix} 0 & -h_3 & h_2 \\ h_3 & 0 & -h_1 \\ -h_2 & h_1 & 0 \end{pmatrix} \quad \in \mathbb{R}^{3\times3}$$

  we can define the square of the skew-symmetric matrix as

$$[\mathbf{h}\times]^2 = (\mathbf{h}\mathbf{h}^T - \mathbf{I}_{3\times3}).$$

In our Matlab implementation, this formula is computed by the *AngleAxisToRot* function. It takes the *axis of rotation* $\mathbf{h}$ and the *angle of rotation* θ from the `main.m` file as inputs, and returns the rotation matrix $\mathbf{R}$.

2.1 Q1.1

Due to the Equation 1, we can consider three different cases based on the value of the angle θ.

- $\theta = 0 \Rightarrow \mathbf{R} = \mathbf{I}_{3\times3}$
- $\theta = \pi \Rightarrow \mathbf{R} = \mathbf{I}_{3\times3} + 2[\mathbf{h}\times]^2$
- $\theta \in (0, \pi)$, in this general case, no simplifications are possible, so the complete Formula (1) must be computed.

2.2 Q1.2

In the first case $\mathbf{h} = [1, 0, 0]^T$ and $\theta = \frac{\pi}{2}$ represent a $90°$ rotation around the x-axis. In fact, both $\mathbf{R}y$ and $\mathbf{R}_z$ are equivalent to the identity, because we have not rotation around the y-axis and the z-axis. This means that $\mathbf{R} = \mathbf{R}_z\mathbf{R}_y\mathbf{R}_x = \mathbf{I}_{3\times3}\mathbf{I}_{3\times3}\mathbf{R}_x = \mathbf{R}_x$. In this case the rotation matrix is

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

2.3 Q1.3

For this configuration, $\mathbf{h} = [0, 0, 1]^T$ and $\theta = \frac{\pi}{3}$. Similar to the previous case, the rotation occurs about the z-axis, with an angle of $60°$. As in the $Q1.2$ case, we have $\mathbf{R}_x = \mathbf{R}_y = \mathbf{I}_{3\times3}$ and the rotation matrix $\mathbf{R} = \mathbf{R}_z\mathbf{R}_y\mathbf{R}_x = \mathbf{R}_z\mathbf{I}_{3\times3}\mathbf{I}_{3\times3} = \mathbf{R}_z$

$$\mathbf{R}_z = \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2.4 Q1.4

For this last case, we don't have $\mathbf{h}$ and θ but the *rotation vector* $\rho = \theta\mathbf{h} \in \mathbb{R}^{3\times3}$. According to the Lemma *The surjective and non-injected nature of the exponential map*, we have $\parallel \mathbf{h} \parallel = 1$, and this which leads to

$$\parallel \rho \parallel = \parallel \theta\mathbf{h} \parallel = \theta \parallel \mathbf{h} \parallel = \theta$$

knowing $\rho = [-\frac{\pi}{3}, -\frac{\pi}{6}, \frac{\pi}{3}]$, we can compute $\mathbf{h}$ and θ with the following equation

$$\theta = \sqrt{\rho_x^2 + \rho_y^2 + \rho_z^2} = \frac{\pi}{2} \qquad \mathbf{h} = \frac{\rho}{\theta} = \left(-\frac{2}{3}, -\frac{1}{3}, \frac{2}{3} \right)^T$$

this means that $\mathbf{R}$ is a $90°$ rotation along the axis defined by the vector $\mathbf{h}$

$$\mathbf{R} = \begin{pmatrix} \frac{4}{9} & -\frac{4}{9} & -\frac{7}{9} \\ \frac{8}{9} & \frac{1}{9} & \frac{4}{9} \\ -\frac{1}{9} & -\frac{8}{9} & \frac{4}{9} \end{pmatrix}$$

3 Exercise 2

Before solving the Inverse Equivalent Angle-Axis Problem we need to verify that the matrix $\mathbf{R}$ is indeed a rotation matrix.

For this reason, we created the function *IsRotationMatrix* which takes the $\mathbf{R}$ matrix from the main program. The procedure of this function is:

- checks if $\mathbf{R} \in \mathbb{R}^{3\times3}$
- verifies if the matrix is orthogonal by: $\mathbf{R}^T\mathbf{R} = \mathbf{I}_{3\times3}$
- checks if $det(\mathbf{R}) = 1$ is true, so if the determinant property of a rotation matrix is respected.

To prevent floating-point errors, it is important to define a tolerance, in this case of 10^{-3}, when computing the properties of the rotation matrix.

If the input $\mathbf{R}$ is a rotation matrix, we can find the corresponding angle-axis representation ($\mathbf{h}$,θ) using the function *RotToAngleAxis*. This function computes the trace of the matrix, that is the sum of all the diagonal elements, from which we can determine the *rotation angle*

$$\theta = \arccos\left(\frac{\mathrm{tr}(\mathbf{R}) - 1}{2}\right)$$

we can then obtain the *axis of rotation* $\mathbf{h}$.

N.B.: This tolerance introduces a very small numerical margin of error, which is acceptable for the exercises we need to compute. To increase the accuracy of the code, one possibility is to use a smaller tolerance, for example, between 10^{-6} *and* 10^{-10}.

3.1 Q2.1

Before proceeding with the exercise, we can distinguish three different cases based on the value of θ

- $\theta = 0$: in this case the rotation matrix $\mathbf{R}$ is equal to the identity matrix $\mathbf{I}_{3\times3}$ and $\mathbf{h}$ can be arbitrary
- $\theta = \pi$: this represents a rotation of 180° about one of the axes. In order to find $\mathbf{h}$ we need to compute the formula

$$h_i = \pm\sqrt{\frac{r_{ii} + 1}{2}} \quad i \in \mathbb{N} \setminus \{0\}, i \leq 3$$

  computing the sign of the component h_{ij} as h_j ($j \in \mathbb{N} \setminus \{0\}, j \leq 3$, $i \neq j$)

$$h_j = \mathrm{sgn}(h_i)\mathrm{sgn}(r_{ij})\sqrt{\frac{r_{jj} + 1}{2}}$$

- $\theta \in (0, \pi)$: we can lead to the anti-symmetric properties of the Rodriguez formula and taking into account just the anti-symmetric part of $\mathbf{R}$

$$\mathbf{R}_{anti-symmetric} = \frac{R - R^T}{2}$$

considering the *axial vector* $\mathbf{a}$

$$\mathbf{a} = \sin\theta \cdot \mathbf{h} = \text{vex}(\mathbf{R}_{anti-symmetric})$$

where vex($\cdot$) is used to convert the skew-symmetric matrix to its corresponding vector, it computes the inverse of the skew.
Consequently

$$\mathbf{h} = \frac{\mathbf{a}}{\sin\theta}$$

3.2 Q2.2

In the first case

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

this represents a rotation relative to the x-axis. $\mathbf{R}$ is different from the identity matrix and $\theta = \frac{\pi}{2}$, so we can consider the anti-symmetric part and find $\mathbf{h} = [1, 0, 0]^T$.

3.3 Q2.3

In this case the rotation is about the z-axis

$$\mathbf{R} = \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

like in the previous matrix, this situation belongs to the third case, $\theta \in (0, \pi)$. In this configuration we obtain $\mathbf{h} = [0, 0, 1]^T$ from $\theta = \frac{\pi}{3}$.

3.4 Q2.4

This matrix represents the particular case where $\mathbf{R}$ is equal to the identity matrix

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \mathbf{I}_{3\times3}$$

it means there is no rotation, so $\theta = 0$ and $\mathbf{h}$ is arbitrary.

3.5 Q2.5

The rotation matrix is

$$\mathbf{R} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

in this case the angle $\theta = \pi$, this means there is a rotation of 180° about the z-axis, so we have an inversion of direction with the x and y-axis. We can define two possible configurations of the vector $\mathbf{h}$: $\mathbf{h}_+ = [0, 0, 1]^T$ or $\mathbf{h}_- = [0, 0, -1]^T$.

3.6 Q2.6

In this last case

$$\mathbf{R} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

the conversion to angle-axis cannot be computed because $\mathbf{R}$ is not a rotation matrix. Indeed, we have $det(\mathbf{R}) = -1$, this matrix is not describing a rotation but a reflection of the x-axis.

4 Exercise 3

In this section we want to compute $\mathbf{R}$ given the Euler Angles(ψ, θ, ϕ) using the function *YPRToRot*.
To obtain the rotation matrix we need to consider the Yaw-Pitch-Roll representation

$$\mathbf{R}_{\text{YPR}} = \begin{pmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{pmatrix}$$

4.1 Q3.1

We can also define this matrix $\mathbf{R}$ as the multiplication of three components

$$\mathbf{R}_{\text{YPR}} = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)$$

considering

- $\mathbf{R}_z = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- $\mathbf{R}_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$
- $\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix}$

To verify the correctness of the obtained $\mathbf{R}$ it is possible to use the validation as in the Exercise 2, the function is *IsRotationMatrix*

4.2 Q3.2

In the first case, the rotation is only about ϕ , with the angles set to $\psi = \theta = 0$ and $\phi = \frac{\pi}{2}$. Using these values, we obtain the corresponding rotation matrix

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

this represents a rotation along the x-axis that confirmed the existence of the movement given by the angle ϕ.

4.3 Q3.3

Like in the Q3.2 test, we have a single rotation, but this time it is about the z-axis. Computing the angles $\phi = \theta = 0, \psi = \frac{\pi}{2}$ we obtain the matrix

$$\mathbf{R} = \begin{pmatrix} \frac{1}{2} & \frac{-\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

4.4 Q3.4

Considering $\psi = \frac{\pi}{3}$, $\theta = \frac{\pi}{2}$ and $\phi = \frac{\pi}{2}$ we can describe a particular configuration because θ is equal to $\frac{\pi}{2}$ so the z and x-axis are aligned and dependent on each others. This can be seen from the obtained matrix

$$\mathbf{R} = \begin{pmatrix} 0 & -\frac{\sqrt{6}-\sqrt{2}}{4} & \frac{\sqrt{6}+\sqrt{2}}{4} \\ 0 & \frac{\sqrt{6}+\sqrt{2}}{4} & \frac{\sqrt{6}-\sqrt{2}}{4} \\ -1 & 0 & 0 \end{pmatrix}$$

4.5 Q3.5

As before we have $\theta = \frac{\pi}{2}$ so there is a dependency between z-axis and x-axis. Indeed, despite the different values $\psi = 0$ and $\phi = \frac{-\pi}{12}$ we have the same relationship as the exercise Q3.4.

$$\mathbf{R} = \begin{pmatrix} 0 & -\frac{\sqrt{6}-\sqrt{2}}{4} & \frac{\sqrt{6}+\sqrt{2}}{4} \\ 0 & \frac{\sqrt{6}+\sqrt{2}}{4} & \frac{\sqrt{6}-\sqrt{2}}{4} \\ -1 & 0 & 0 \end{pmatrix}$$

5 Exercise 4

In this section we want to compute the Euler Angles given the rotation matrix $\mathbf{R}$ using the *RotToYPR* function. First of all, we need to verify if the considered $\mathbf{R}$ is a rotation matrix, by performing the usual validation with the *IsRotationMatrix* function.

5.1 Q4.1

Before computing all the angles, it is important include a check on the singularities.
θ is first computed as

$$\theta = \texttt{arctan2}\big(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}\big)$$

Depending on the value of θ we can distinguish two cases:

- $\theta = 0$ the inverse mapping is not unique, so we cannot define a single configuration
- $\theta \neq 0$ there is a unique solution, so it is possible to evaluate the angles as:

$$\psi = \texttt{arctan2}(r_{21}, r_{11})$$

$$\phi = \texttt{arctan2}(r_{32}, r_{33})$$

5.2 Q4.2

In the first case, the matrix $\mathbf{R}$ provided as input to the function is

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

from which we obtain $\psi = \theta = 0$ and $\phi = \frac{\pi}{2}$.
As can be observed from the matrix $\mathbf{R}$, which corresponds to a rotation about the x-axis, indeed the only angle different from zero is ϕ.

5.3 Q4.3

As in the previous case, by observing the input matrix $\mathbf{R}$

$$\mathbf{R} = \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

which represents a rotation around the z-axis, it can be concluded that the only non-zero angle is ψ, as confirmed by the obtained results $\theta = \phi = 0$ and $\psi = \frac{\pi}{3}$.

5.4 Q4.4

In the last case, it's clear that the matrix $\mathbf{R}$

$$\mathbf{R} = \begin{pmatrix} 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{1}{2} & \frac{\sqrt{2}\sqrt{3}}{4} & \frac{\sqrt{2}\sqrt{3}}{4} \\ -\frac{\sqrt{3}}{2} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} \end{pmatrix}$$

is composed by three different rotations. For this reason, all three angles computed by the function are different from zero $\psi = \frac{\pi}{2}$, $\theta = \frac{\pi}{3}$ and $\phi = \frac{\pi}{4}$.

6 Exercise 5

The request of this exercise is to compute corresponding angle-axis representation ($\mathbf{h}$,θ) with two different methods:

1. By using *RotToAngleAxis* (see the Exercise 2)
2. By calculating $\mathbf{h}$ with Matlab function `eigs`, thanks to which we computed only the eigenvector associated with eigenvalue closest to 1, followed by the calculation of θ with *RotToAngleAxis*

6.1 Q5.1

The input matrix and the output values are the following, in particular $\mathbf{h_1}$ and θ_1 correspond to the first method while $\mathbf{h_2}$ and θ_2 to the second one:

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\mathbf{h_1} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \theta_1 = \frac{\pi}{2} \qquad ; \qquad \mathbf{h_2} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \theta_2 = \frac{\pi}{2}$$

As expected, the results are the same.

6.2 Q5.2

The input matrix and the output values are the following, and, as the previous case, $\mathbf{h_1}$ and θ_1 correspond to the first method while $\mathbf{h_2}$ and θ_2 to the second one:

$$R = \frac{1}{9} \begin{pmatrix} 4 & -4 & -7 \\ 8 & 1 & 4 \\ -1 & -8 & 4 \end{pmatrix}$$

$$\mathbf{h_1} = \begin{bmatrix} -\frac{2}{3} \\ -\frac{1}{3} \\ \frac{2}{3} \end{bmatrix}, \theta_1 = \frac{\pi}{2} \qquad ; \qquad \mathbf{h_2} = \begin{bmatrix} \frac{2}{3} \\ \frac{1}{3} \\ -\frac{2}{3} \end{bmatrix}, \theta_2 = \frac{\pi}{2}$$

Geometrically, a rotation of angle θ around the axis $\mathbf{h}$ is equivalent to a rotation of the same angle around the opposite axis -$\mathbf{h}$.
Contrary to what one might expect, in this second exercise the sign of the components of $\mathbf{h}$ computed using the second method (`eigs`) turns out to be opposite to those obtained via the *RotToAngleAxis* function. Instead, the values of θ_1 and θ_2 turned out to be the same. This can be caused by the computation that we implemented inside the *RotToAngleAxis*. Indeed, inside the function, θ is calculated using the following formula:

$$\theta = \arccos\left(\frac{\mathrm{tr}(\mathbf{R}) - 1}{2}\right)$$

By definition, this method always returns positive values. Moreover, the calculation of $\mathbf{h_1}$ inside the *RotToAngleAxis* function is made by this formula:

$$\mathbf{a} = \sin\theta \cdot \mathbf{h} = \text{vex}(\mathbf{R}_{anti-symmetric})$$

$$\mathbf{h} = \frac{\mathbf{a}}{\sin\theta}$$

which means that the sign of h depends directly on the sign of $\sin(\theta)$. On the other hand, by calculating the values of $\mathbf{h_2}$ with the Matlab function `eigs`, the eigenvector corresponding to the eigenvalue 1 of $\mathbf{R}$ is not determined with a specific sign. This implies that, for example in this case, the sign of $\mathbf{h_2}$ results with opposite sign with respect to the method of *RotToAngleAxis* . So, to obtain the same rotation as before, also the value of θ_2 should be with opposite sign. This does not happen because the calculation of θ_2 in the second method utilizes again the function *RotToAngleAxis* . To obtain a coherent rotation, the value of θ_2 should be $-\frac{\pi}{2}$.

7 Exercise 6

As requested in the exercise, based on Figure 1, we calculated the matrices corresponding to the movement of the frame transformations by hand.
First of all, we analyzed the frames of the Franka robot and based on that, we defined all the required rotations and translations. After that, we computed all the matrices.
In particular, we computed individually the rotation sub-matrices, including all the products in the cases where the rotations were sequential. Then we considered the translations and composed the complete matrices.
During the process,we computed all the rotations according to right-hand rule. The translations in the matrices are expressed in millimeters [*mm*].

7.1 Q6.1

In the first one, the frame does not rotate and there is only a translation of 175 mm along the z-axis.

$$ {}^0_1T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 175 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

7.2 Q6.2

For the second displacement, we have a translation along the z-axis of 98 mm and a rotation around the same axis of 180° followed by a rotation around the x-axis of 90°

$$ {}^1_2T = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 98 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

7.3 Q6.3

In the third pose, we have only one translation along the x-axis of 105 mm and a rotation around the y-axis of 90°

$$ {}^2_3T = \begin{pmatrix} 0 & 0 & 1 & 105 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

7.4 Q6.4

In this frame, there are two different translations: one on the z-axis of 326.5 mm and the other one on the y-axis of 145.5 mm. Ultimately, we have two sequential rotations: the first around the z-axis of 180° and the second around the y-axis of 90°

$$ {}^3_4T = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 145.5 \\ -1 & 0 & 0 & 326.5 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

7.5 Q6.5

To reach to the position 5 we need a translation of 35 mm on the x-axis and two rotations, the first of 270° around the z-axis and the second around the x-axis of 270°

$$ {}^4_5T = \begin{pmatrix} 0 & 0 & 1 & 35 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

7.6 Q6.6

In this pose, we have one translation on the z-axis of 385 mm and two sequential rotation, each of them of 270° but the first around z-axis and the second around y-axis

$$^{5}_{6}T = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 385 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

7.7 Q6.7

In this case we don't have any rotation or translation, so the rotation matrix is equal to the identity and the translation vector is null. The resulting matrix is

$$^{6}_{6}T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

7.8 Q6.8

To complete the Franka configuration, we assumed that this point is on the outer end of the end effector, so we use a translation on the z-axis of -153 mm, a rotation around the z-axis of 270° and a last rotation of the same angle around the y-axis

$$^{7}_{6}T = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -153 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

To check the correctness of the result we computed also the $^{6}_{7}T$ and verified if the rotation matrices were one the transposed of the other. The resulting $^{6}_{7}T$ matrix is the following:

$$^{6}_{7}T = \begin{pmatrix} 0 & 0 & 1 & 153 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The check was successful, so we concluded that: $^{7}_{6}R^{\mathrm{T}}$ = $^{6}_{7}R$.