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:

Bertille Beaujean
Yui Momiyama
Mamoru Ota

Student ID:

s7899816
s7784921
s7784976

Professors:

Giovanni Indiveri
Enrico Simetti
Giorgio Cannata

Tutors:

Andrea Tiranti
Georgii Kurshakov
Luca Tarasi

October 31, 2024

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

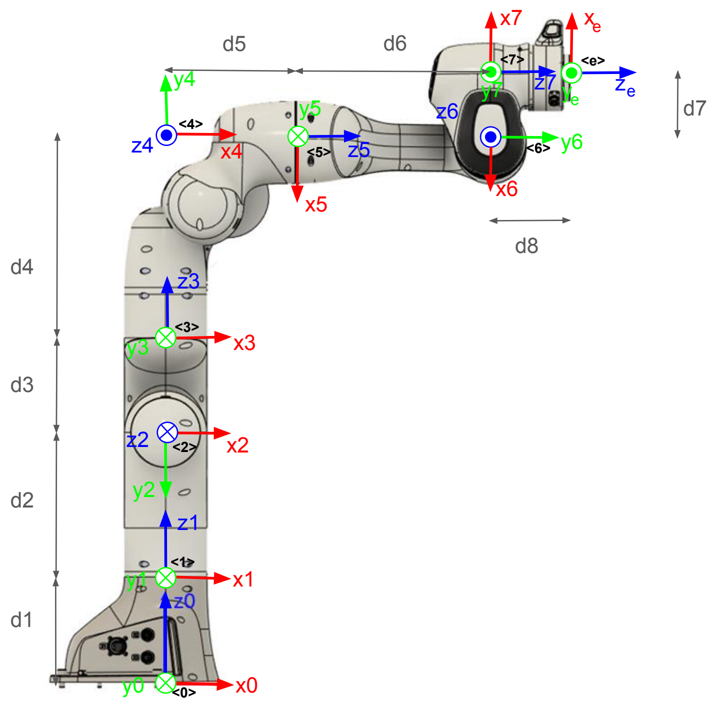

Figure 1: exercise 5 frames

2 Exercise 1

2.1 Q1.1

Given an angle-axis par (h, θ), in order to compute the equivalent rotation matrix $R(h, \theta)$, we used the Rodrigues Formula here under to create the function called "AngleAxisToRot", such that:

$R(h, \theta) = I_{3\times3} + [h\times]\sin\theta + [h\times]^2(1 - \cos\theta)$

, where $[h\times]$ is the skew-symmetric matrix associated to the vector h.

The results from this function are the following:

2.2 Q1.2

Pair angle-axis: $h = [1, 0, 0]^T$ and $\theta = 90$. This is a rotation around the x-axis.

The correspondent matrix computed is : $R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$.

For example, if you rotate the vector "y" [0,1,0] 90 degrees around the x-axis, by multiplying it with R, it becomes [0,0,1], that is so say that it is shifting to the z-axis.

2.3 Q1.3

Pair angle-axis: $h = [0, 0, 1]^T$ and $\theta = \pi/3$. This is a rotation around the z-axis.

The correspondent matrix computed is : $R = \begin{pmatrix} 0.5000 & -0.8660 & 0 \\ 0.8660 & 0.5000 & 0 \\ 0 & 0 & 1.0000 \end{pmatrix}$.

For example, if you rotate the vector "x" [1,0,0] $\pi/3$ radians around the z-axis, it becomes [0.5,0.866,0]. One may recognize [1/2, $\sqrt{3}/2, 0]$ = $[cos(\pi/3), sin(\pi/3), 0]$. In other words, this matrix can be written:

$R = \begin{pmatrix} cos(\pi/3) & -sin(\pi/3) & 0 \\ sin(\pi/3) & cos(\pi/3) & 0 \\ 0 & 0 & 1 \end{pmatrix}$, under the classic formulation of rotation matrix.

2.4 Q1.4

Pair angle-axis: $h\theta = \rho = [-\pi/3, -\pi/6, \pi/3]$.

As a matter of fact, there is an infinity of pairs (h, θ) such that their product is ρ. Let's choose one. In particular, the equality implies that $\|\rho\| = |\theta|\|h\|$. One may choose to have a unit vector, so, given that $\|\rho\| = \pi/2$, the equation gives $\|\rho\| = \pi/2 = |\theta|.1$ Consequently, by definition of $h\theta = \rho$, one possible pair angle-axis is $(h = [-\frac{2}{3}, -\frac{1}{3}, \frac{2}{3}], \theta = \frac{\pi}{2})$.

The correspondent matrix computed is : $R = \begin{pmatrix} 0.4444 & -0.4444 & -0.7778 \\ 0.8889 & 0.1111 & 0.4444 \\ -0.1111 & -0.8889 & 0.4444 \end{pmatrix}$.

Let's notice that since this axis does not contain 0, the output matrix does not contain 0 either.

3 Exercise 2

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.

3.1 Q2.1 Implementation

The function `RotToAngleAxis(R)` was implemented in MATLAB to compute the angle-axis representation from a given rotation matrix R. The function calculates the rotation angle θ using the trace of R, and the rotation axis **h** by extracting the elements of the skew-symmetric matrix R. The function also checks that R is a valid rotation matrix by ensuring:

- R is a 3x3 matrix,
- the determinant of R is close to +1,
- R is orthogonal, meaning $R^\top R = I$.

If any of these conditions are not met, an error is raised. Otherwise, the function returns the angle and axis of rotation.

3.2 Q2.2

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

For this matrix, the computed rotation angle θ is 90 degrees (or $\frac{\pi}{2}$ radians), with a rotation axis $\mathbf{h} = [1, 0, 0]$. This indicates a rotation of 90 degrees around the x-axis. If we apply this rotation to a point on the y-axis, it shifts to the z-axis, as expected and described in question Q1.2.

3.3 Q2.3

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

For this matrix, the function returned an error: *"Input matrix must have a determinant of +1"*. This error occurred because the determinant of R was not close enough to +1, which is a requirement for a valid rotation matrix. As a result, no angle-axis representation was computed for this case.

3.4 Q2.4

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

For this identity matrix, the computed rotation angle θ is 0 radians, with a rotation axis $\mathbf{h} = [0, 0, 0]$. This result signifies no rotation, as the identity matrix represents the absence of any transformation.

3.5 Q2.5

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

For this matrix, the computed rotation angle θ is 180 degrees (or π radians), with a rotation axis **h** $= [0, 0, 1]$. This indicates a 180-degree rotation around the z-axis. This rotation effectively mirrors points across the x-y plane, which is consistent with the structure of the matrix.

4 Exercise 3

4.1 Q3.1

In order to compute the equivalent rotation matrix, we created the function called "YPRToRot". The function uses the YPR (yaw-pitch-roll) convention. The matrix is computed by multiplying the rotation matrix associated to the rotation around each one of the Euler angles. The results of the function are the following:

4.2 Q3.2

Euler angles: $\psi = \theta = 0, \phi = \pi/2$.

The associated matrix computed is : $R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$

This result is the same as the one in Q1.2. and Q2.2. Although the description is different, since we used RPY convention, we computed a rotation around the x-axis in both question.

Indeed, a rotation of 90° with respect to axis x may be expressed in 3 different ways: with the rotation matrix, with the pair $(h, \theta) = ([1, 0, 0]^T, 90)$ or with the Euler angles $(\psi, \theta, \phi) = (0, 0, \pi/2)$.

4.3 Q3.3

Euler angles: $\phi = \theta = 0, \psi = 60$.

The associated matrix computed is : $R = \begin{pmatrix} 0.5000 & -0.8660 & 0 \\ 0.8660 & 0.5000 & 0 \\ 0 & 0 & 1.0000 \end{pmatrix}$

This result is also the same as the one in Q1.3 et Q2.3. In this case, in both question, we computed a rotation around the z-axis.

This matrix is associated with the axe-angle pair $(h, \theta) = ([0, 0, 1]^T, \pi/3)$ and the Euler angles $(\psi, \theta, \phi) = (60, 0, 0)$.

4.4 Q3.4

Euler angles: $\psi = \pi/3, \theta = \pi/2, \phi = \pi/4$.

The associated matrix computed is : $R = \begin{pmatrix} 0 & -0.2588 & -0.9659 \\ 0 & 0.9659 & 0.2588 \\ -1.0000 & 0 & 0 \end{pmatrix}$

4.5 Q3.5

Euler angles: $\psi = 0, \theta = \pi/2, \phi = -\pi/12$.

The associated matrix computed is : $R = \begin{pmatrix} 0 & -0.2588 & 0.9659 \\ 0 & 0.9659 & 0.2588 \\ -1.0000 & 0 & 0 \end{pmatrix}$

The result of Q3.4 and Q3.5 is the same. That is because the different angle settings in both inputs apply different rotations for each angle, but the final spatial orientation (the effect of the rotation) ends up being the same due to the specific combination of these angles.

5 Exercise 4

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

5.1 Q4.1 Implementation

The function `RotToYPR(R)` was implemented in MATLAB to compute the YPR angles from a given rotation matrix. The function checks that R is a valid rotation matrix by ensuring:

- R is a 3x3 matrix,
- the determinant of R is close to +1,
- R is orthogonal, meaning $R^\top R = I$.

After validating the matrix, the function calculates the pitch angle θ using the value of $R(3,1)$. If $\cos\theta$ is close to zero (i.e., in the case of gimbal lock), it sets the roll angle ϕ to zero and adjusts the yaw angle ψ. Otherwise, it calculates the yaw, pitch, and roll angles using standard trigonometric relationships.

5.2 Q4.2

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

For this matrix, the computed Yaw (psi) angle is 0.0000 radians, the Pitch (theta) angle is -0.0000 radians, and the Roll (phi) angle is 1.5708 radians. This result represents a 90-degree rotation around the x-axis, as expected for this matrix and seen in previous exercises.

5.3 Q4.3

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

For this matrix, the computed Yaw (psi) angle is approximately 1.0472 radians, the Pitch (theta) angle is -0.0000 radians, and the Roll (phi) angle is 0.0000 radians. This result indicates a 60-degree rotation around the z-axis, aligning with the matrix structure. It corresponds to the matrix described in different ways in Q3.3.

5.4 Q4.4

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

For this matrix, the computed Yaw (psi) angle is approximately 1.5708 radians, the Pitch (theta) angle is approximately 1.0472 radians, and the Roll (phi) angle is approximately 0.7854 radians. This combination of rotations represents a complex rotation involving all three axes, with significant contributions from each angle.

6 Exercise 5

To compute by hand the transformation matrices between the successive joints of the Franka robot, we use the geometric definition of the transformation ${}^i_jT$ between the frames i and j, such that : $\begin{pmatrix} {}^i_jR & {}^i_jP \\ 0_{1\times3} & 1 \end{pmatrix}$, where ${}^i_jR$ et ${}^i_jP$ are respectively the rotation matrix and the translation matrix from frame i to j.

Here, as the translations are unidirectional, all the translation matrices are expressed as a vector equal to the translation parameter d in the direction of the translation and null in other directions.

To compute the rotation matrices, in this particular case, rather than determining the angles and axes of the rotations between two successive frames, one can note on the figure that all rotations make an angle equal to $\pm$90° or $\pm$180°. To put it another way, one may express the vectors of every frame as an easy linear combination of the vectors of the previous frame, with coefficients equal to 0 or ±1.

- **Q5.1** $${}^0_1T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

  For the first transformation, there is clearly no rotation between the two frames: the rotation matrix is the identity. As for the translation matrix, the translation of d_1 with respect to z_0 is $[0, 0, d_1]^T$.

- **Q5.2** $${}^1_2T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

  The matrix ${}^2_3R$ standing for a rotation of -90° with respect to x_2 may be determined from the figure, reading that $x_2 = x_1, y_2 = -z_1, z_2 = y_1$. The translation of d_2 with respect to z_1 is $[0, 0, d_2]^T$.

- **Q5.3** $${}^2_3T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -d_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

  Following an analog method between every two successive joints, one may determine the transformation matrices as follow.
  ${}^2_3R$ stands for a rotation of +90° with respect to x_2 and ${}^2_3P$ for a translation of d_3 with respect to $-y_2$.

- **Q5.4** $${}^3_4T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

  ${}^3_4R$ stands for a rotation of +90° with respect to x_3 and ${}^3_4P$ for a translation of d_4 with respect to z_3.

- **Q5.5** $${}^4_5T = \begin{pmatrix} 0 & 0 & 1 & d_5 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

  In this case, particularly, it is easier to note that $x_5 = -y_4, y_5 = -z_4, z_5 = x_4$, rather than determining the angle of the rotation with respect to x_5, y_5, z_5. One may cross check those results by describing the rotation from frame 4 to 5 as a rotation of -90° with respect to z_4 and then a rotation of -90° with respect to x_4. One then computes the rotation matrix between the frames by multiplying the rotation matrix of the first rotation with the one of the second rotation. Indeed, one would also find:
  $${}^4_5R = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}$$
  ${}^4_5P$ stands for a translation of d_5 with respect to x_4.

- **Q5.6** $${}^5_6T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

  ${}^5_6R$ stands for a rotation of +90° with respect to x_5 and ${}^5_6P$ for a translation of d_6 with respect to z_5.

- **Q5.7** $${}^6_7T = \begin{pmatrix} -1 & 0 & 0 & -d_7 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

  Again, let's see that $x_7 = -x_6, y_7 = z_6, z_7 = y_6$. Otherwise, the rotation matrix can be determined by a +180° rotation w.r.t. z_6 followed by a +90° rotation w.r.t. x_6. ${}^6_7P$ stands for a translation of d_7 with respect to $-x_6$.

- **Q5.8** $${}^7_eT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_8 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

  There is no rotation, so ${}^7_eR$ is the identity matrix, and ${}^7_eP$ stands for a translation of d_e with respect to z_7.

7 Appendix

[Comment] Add here additional material (if needed)

7.1 Appendix A

7.2 Appendix B