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:

Delucchi Manuel
Cappellini Matteo

Student ID:

s4803977
s4822622

Professors:

Enrico Simetti
Giorgio Cannata

Tutors:

Andrea Tiranti
Francesco Giovinazzo
George Kurshakov

October 23, 2023

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, Euler Angles and Quaternions) will be reviewed.

The first assignment is **mandatory** and consists of 4 different exercises. You are asked to:

- Download the .zip file called MOCOM-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*", "*ComputeAngleAxis.m*", "*ComputeInverseAngleAxis.m*", and "*QuatToRot.m*".
- Write a report motivating the answers for each exercise, following the predefind format on this document.

1.1 Exercise 1 - Equivalent Angle-Axis Representation (Exponential representation)

A particularly interesting minimal representation of 3D rotation matrices is the so-called "*angle-axis representation*" or "*exponential representation*". Given two frames $< a >$ and $< b >$, initially coinciding, let's consider an applied geometric unit vector $(\mathbf{v}, O_a) = (\mathbf{v}, O_b)$, passing through the common origin of the two frames, whose initial projection on $< a >$ is the same of that on $< b >$. Then let's consider that frame $< b >$ is purely rotated around **v** of an angle θ, even negative, accordingly with the right-hand rule. We note that the axis-line defined by $(\mathbf{v}, O_a) = (\mathbf{v}, O_b)$ remains common to both the reference systems of the two frames $< a >$ and $< b >$ and we obtain that the orientation matrix constructed in the above way is said to be represented by its equivalent angle-axis representation that admits the following equivalent analytical expression, also known as Rodrigues Formula:

$$\mathbf{R}(^*\mathbf{v}, \theta) = e^{[^*\mathbf{v}\times]\theta} = e^{[\rho\times]} = \mathbf{I}_{3x3} + [^*\mathbf{v}\times]\sin(\theta) + [^*\mathbf{v}\times]^2(1-\cos(\theta))$$

Q1.1 Given two generic frames $< a >$ and $< b >$, given the geometric unit vector $(\mathbf{v}, O_a) = (\mathbf{v}, O_b)$ and the angle θ, implement on MATLAB the Rodrigues formula, computing the rotation matrix ${}^a_bR$ of frame $< b >$ with respect to $< a >$.

Then test it for the following cases and comment the results obtained, including some sketches of the frames configurations:

- **Q1.2** $\mathbf{v} = [1, 0, 0]$ and $\theta = 45°$
- **Q1.3** $\mathbf{v} = [0, 1, 0]$ and $\theta = \pi/6$
- **Q1.4** $\mathbf{v} = [0, 0, 1]$ and $\theta = 3\pi/4$
- **Q1.5** $\mathbf{v} = [0.3202, 0.5337, 0.7827]$ and $\theta = 2.8$
- **Q1.6** $\boldsymbol{\rho} = [0, 2\pi/3, 0]$;
- **Q1.7** $\boldsymbol{\rho} = [0.25, -1.3, 0.15]$;
- **Q1.8** $\boldsymbol{\rho} = [-\pi/4, -\pi/3, \pi/6]$;

Note that $\boldsymbol{\rho}$ is $\boldsymbol{\rho} = \mathbf{v} * \theta$

1.2 Exercise 2 - Inverse Equivalent Angle-Axis Problem

Given two reference frames $< a >$ and $< b >$, referred to a common world coordinate system $< w >$, their orientation with respect to the world frame $< w >$ is expressed in Figure 1.

Q2.1 Compute the orientation matrix ${}^a_bR$, by inspection of Figure 1, without using the Rodriguez formula.
Q2.2 Solve the Inverse Equivalent Angle-Axis Problem for the orientation matrix ${}^a_bR$.
Q2.3 Given the following Transformation matrix:

$${}^w_cT = \begin{bmatrix} 0.835959 & -0.283542 & -0.46986 & 0 \\ 0.271321 & 0.957764 & -0.0952472 & -1.23 \\ 0.47703 & -0.0478627 & 0.877583 & 14 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solve the Inverse Equivalent Angle-Axis Problem for the orientation matrix ${}^c_bR$.

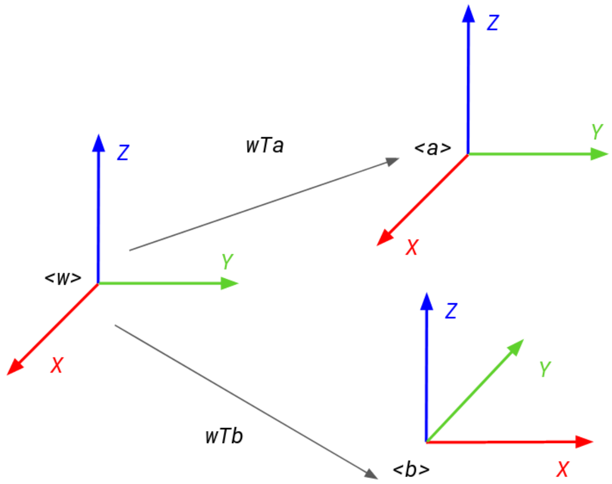


Figure 1: exercise 2 frames

1.3 Exercise 3 - Euler angles (Z-X-Z) vs Tait-Bryan angles (Yaw-Pitch-Roll)

Any orientation matrix can be expressed in terms of three elementary rotations in sequence. These can occur either about the axes of a fixed coordinate system (extrinsic rotations), or about the axes of a rotating coordinate system (intrinsic rotations) initially aligned with the fixed one. Then we can distinguish:

- Proper Euler angles: X-Z-X, Y-Z-Y, ...
- Tait-Bryan angles: Z-Y-X, X-Y-Z, ...

Q3.1 Given two generic frames $< w >$ and $< b >$, define the elementary orientation matrices for frame $< b >$ with respect to frame $< w >$, knowing that:

- $< b >$ is rotated of $45°$ around the z-axis of $< w >$
- $< b >$ is rotated of $60°$ around the y-axis of $< w >$
- $< b >$ is rotated of $-30°$ around the x-axis of $< w >$

Q3.2 Compute the equivalent angle-axis representation for each elementary rotation
Q3.3 Compute the z-y-x (yaw,pitch,roll) representation and solve the Inverse Equivalent Angle-Axis Problem for the obtained orientation matrix
Q3.4 Compute the z-x-z representation and solve the Inverse Equivalent Angle-Axis Problem for the obtained orientation matrix

1.4 Exercise 4 - Quaternions

Given the following quaternion: $q = 0.1647 + 0.31583i + 0.52639j + 0.77204k$ expressing how a reference frame $< b >$ is rotated with respect to $< a >$:
Q4.1 Compute the equivalent rotation matrix, **WITHOUT** using built-in matlab functions.
Q4.2 Solve the Inverse Equivalent Angle-Axis Problem for the obtained orientation matrix

2 Exercise 1

2.1 Q1.1

The *angle-axis representation* or *exponential representation* provides a great way to represent 3D rotation matrices. Given two initially overlapping frames, denoted as $< a >$ and $< b >$, we introduce a geometric unit vector $(\mathbf{v}, O_a) = (\mathbf{v}, O_b)$ that starts from the shared origin of both frames. Now, let's consider frame $< b >$ being rotated entirely around **v** by an angle θ, even if it's negative, following the right-hand rule.
It's worth noting that the axis-line defined by $(\mathbf{v}, O_a) = (\mathbf{v}, O_b)$ remains consistent in both reference systems of frames $< a >$ and $< b >$. Thanks to this, we can establish that the orientation matrix, constructed in this manner, is equivalently represented in the angle-axis form. This form can be expressed using the Rodrigues Formula.

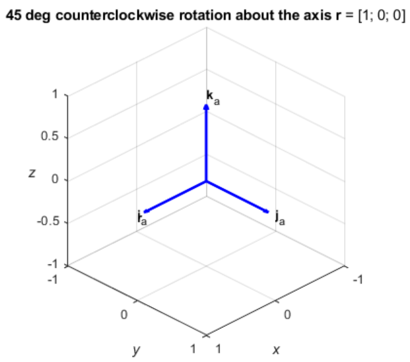
For this part of the assignment we need to implement the function *ComputeAngleAxis()* which implements the Rodrigues Formula. This function takes the geometric unit vector **v** and the rotation angle θ as inputs and uses **v** to calculate the **skew matrix**, as follows:

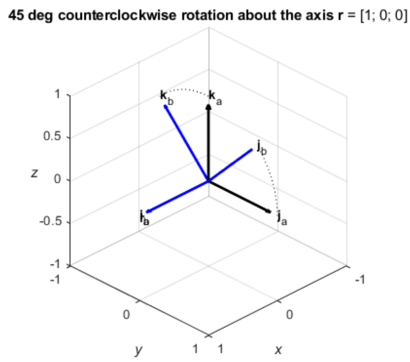
$$v_{skew} = \begin{bmatrix} 0 & -v(3) & v(2) \\ v(3) & 0 & -v(1) \\ -v(2) & v(1) & 0 \end{bmatrix} \tag{1}$$

Then it implements the **Rodrigues Formula**, returning the value of the corresponding rotation matrix ${}^a_bR$:

$$\mathbf{R}({}^*\mathbf{v}, \theta) = e^{[{}^*\mathbf{v}\wedge]\theta} = e^{[\rho\wedge]} = \mathbf{I}_{3x3} + [{}^*\mathbf{v}\wedge]\sin(\theta) + [{}^*\mathbf{v}\wedge]^2(1 - \cos(\theta))$$

2.2 Q1.2

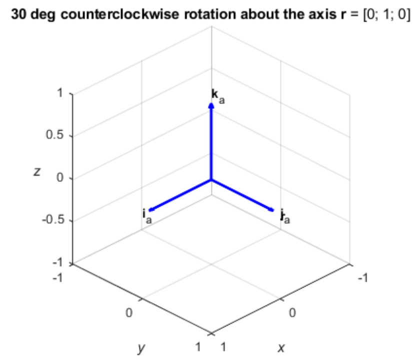


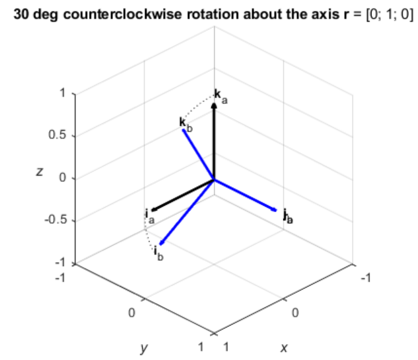


We can observe that the frame $< a >$ undergoes a rotation around the axis defined by **v** = [1 0 0], with an associated rotation angle θ = 45°, corresponding to $\frac{\pi}{4}$ *rad*. This transformation results in the establishment of the new frame, denoted as $< b >$. The resulting rotation matrix is:

$$ {}^a_bR = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.7071 & -0.7071 \\ 0 & 0.7071 & 0.7071 \end{bmatrix} \tag{2}$$

2.3 Q1.3

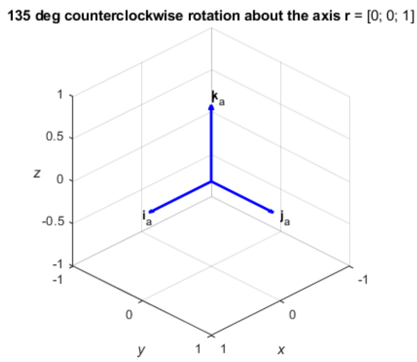


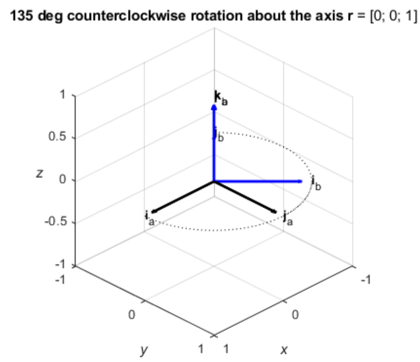


We can observe that the frame $< a >$ undergoes a rotation around the axis defined by **v** = [0 1 0], with an associated rotation angle $\frac{\pi}{6}$ *rad*. This transformation results in the establishment of the new frame, denoted as $< b >$. The resulting rotation matrix is:

$$ {}^a_bR = \begin{bmatrix} 0.866 & 0 & 0.5 \\ 0 & 1 & 0 \\ -0.5 & 0 & 0.866 \end{bmatrix} \tag{3} $$

2.4 Q1.4

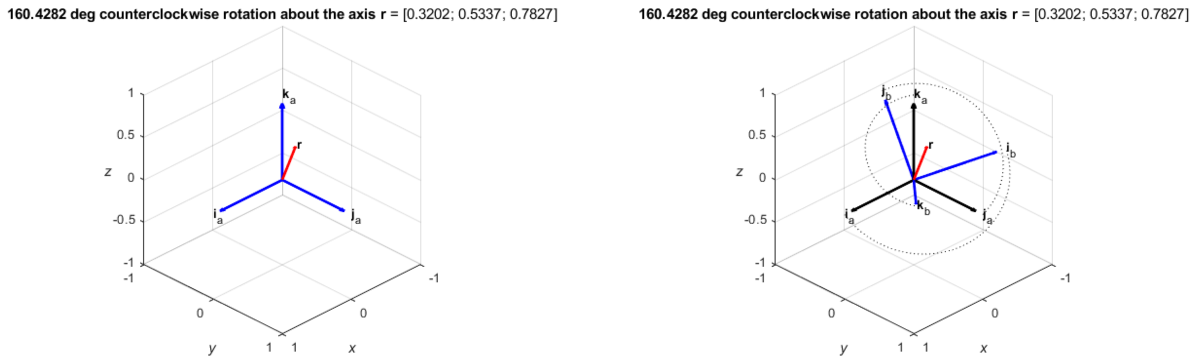




We can observe that the frame $< a >$ undergoes a rotation around the axis defined by **v** = [0 0 1], with an associated rotation angle $\frac{3}{4}\pi$ *rad*. This transformation results in the establishment of the new frame, denoted as $< b >$. The resulting rotation matrix is:

$$ {}^a_bR = \begin{bmatrix} -0.7071 & -0.7071 & 0 \\ 0.70710 & -0.7071 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4} $$

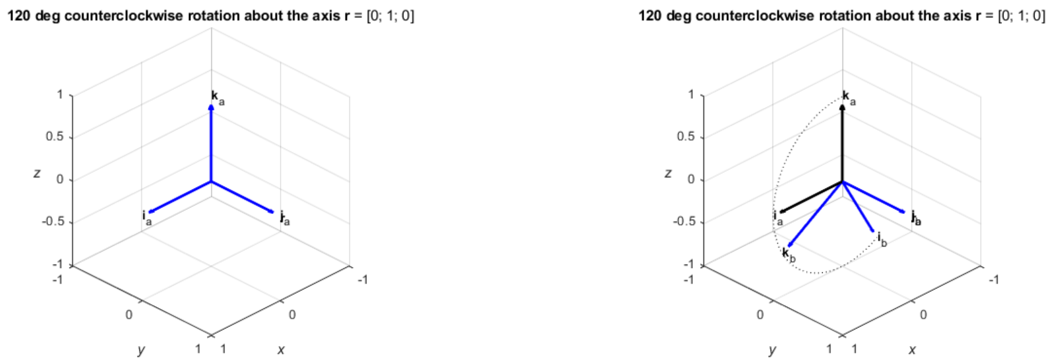
2.5 Q1.5



We can observe that the frame $< a >$ undergoes a rotation around the axis defined by **v** = [0.3202 0.5337 0.7827], with an associated rotation angle 2.8 *rad*. This transformation results in the establishment of the new frame, denoted as $< b >$. The resulting rotation matrix is:

$$
{}^a_bR = \begin{bmatrix} -0.7431 & 0.0697 & 0.6655 \\ 0.5941 & -0.3890 & 0.7041 \\ 0.3080 & 0.9186 & 0.2477 \end{bmatrix} \tag{5}
$$

2.6 Q1.6



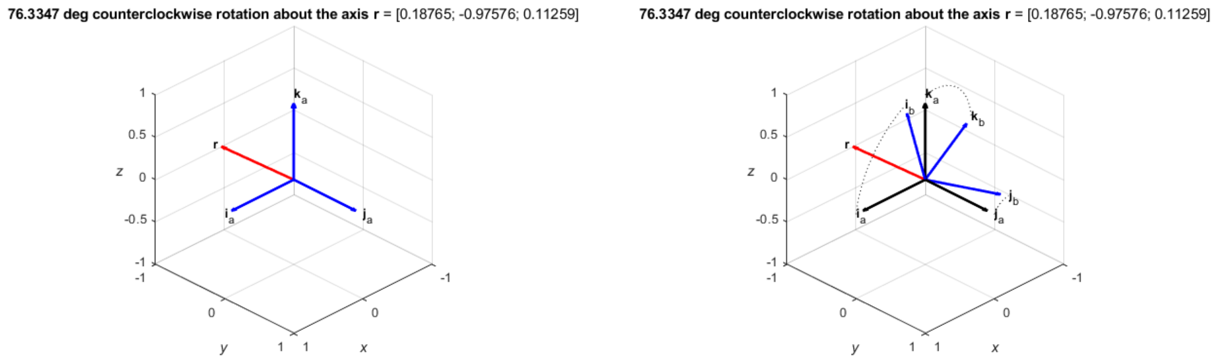
Here we are given $\rho = \begin{bmatrix} 0 & \frac{2}{3}\pi & 0 \end{bmatrix}$, which we can use to obtain v and θ in the following way:

$$
\theta = \sqrt{(\rho_1)^2 + (\rho_2)^2 + (\rho_3)^2} \quad \text{and} \quad \mathbf{v} = \frac{\rho}{\theta} \tag{6}
$$

By applying these formulae we get that $\mathbf{v} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$ and $\theta = \frac{2}{3}\pi$ *rad*. Resulting in the rotation matrix :

$$
{}^a_bR = \begin{bmatrix} -0.5 & 0 & 0.866 \\ 0 & 1 & 0 \\ 0.866 & 0 & -0.5 \end{bmatrix} \tag{7}
$$

2.7 Q1.7



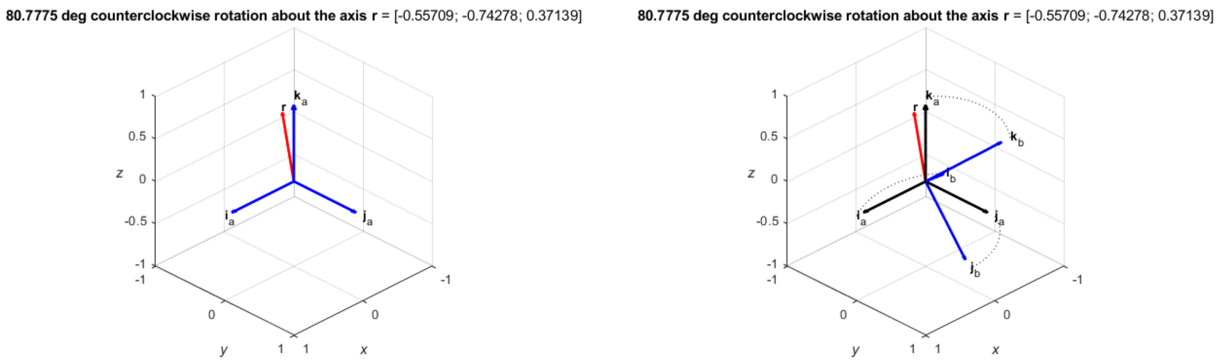
Here we are given $\rho = \begin{bmatrix} 0.25 & -1.3 & 0.15 \end{bmatrix}$, which we can use to obtain v and θ in the following way:

$$\theta = \sqrt{(\rho_1)^2 + (\rho_2)^2 + (\rho_3)^2} \quad \text{and} \quad v = \frac{\rho}{\theta} \tag{8}$$

By applying these formulae we get that $v = \begin{bmatrix} 0.1876 & -0.9758 & 0.1126 \end{bmatrix}$ and $\theta = 1.3323$ *rad*. Resulting in the rotation matrix :

$$ {}_b^a R = \begin{bmatrix} 0.2631 & -0.2492 & -0.932 \\ -0.0304 & 0.9634 & -0.2662 \\ 0.9643 & 0.0984 & 0.2459 \end{bmatrix} \tag{9}$$

2.8 Q1.8



Here we are given $\rho = \begin{bmatrix} -\frac{\pi}{4} - \frac{\pi}{3} - \frac{\pi}{6} \end{bmatrix}$, which we can use to obtain v and θ in the following way:

$$\theta = \sqrt{(\rho_1)^2 + (\rho_2)^2 + (\rho_3)^2} \quad \text{and} \quad v = \frac{\rho}{\theta} \tag{10}$$

By applying these formulae we get that $v = \begin{bmatrix} -0.5571 & -0.7428 & 0.3714 \end{bmatrix}$ and $\theta = 1.4098$ *rad*. Resulting in the rotation matrix :

$$ {}_b^a R = \begin{bmatrix} 0.4209 & -0.0191 & -0.9069 \\ 0.7141 & 0.6236 & 0.3182 \\ 0.5594 & -0.7815 & 0.2761 \end{bmatrix} \tag{11}$$

3 Exercise 2

In this exercise, we will explore the *Inverse Angle-Axis Representation* of an orientation matrix. This representation offers a mathematical means to depict a rotation in three-dimensional space. It gives information about the axis around which the rotation occurs and about the magnitude of the rotation. The main goal of this exercise is to develop an algorithm capable of taking an orientation matrix as input and extracting the rotation information from it. The process of extracting this information involves checking if the input matrix satisfies certain criteria, such as being a valid rotation matrix. Then, mathematical operations are applied to compute both the axis and angle of rotation.

3.1 Q2.1

Firstly, we want to compute the orientation matrix ${}^a_bR$ which represents a rotation of $\theta = \frac{\pi}{2}$ *rad* around the z-axis of the frame $< a >$, gievn by:

$$
{}^a_bR = \begin{bmatrix} cos(\frac{\pi}{2}) & -sin(\frac{\pi}{2}) & 0 \\ sin(\frac{\pi}{2}) & cos(\frac{\pi}{2}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}
$$

To do so, we consider two rotation matrices, ${}^w_aR$ and ${}^w_bR$ representing the rotation from $< w >$ to $< a >$ and from $< w >$ to $< b >$, respectively. We can then obtain the rotation matrix ${}^a_bR$ with the following formula:

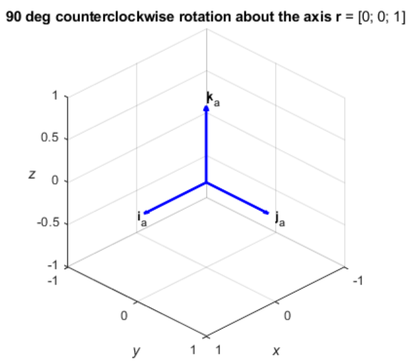
$$
{}^a_bR =^w_a R^T \cdot^w_b R \tag{13}
$$

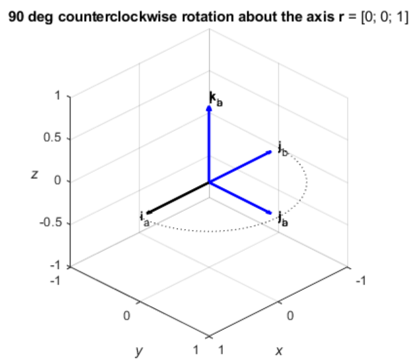
3.2 Q2.2

In this part of the exercise we are prompted to write the function *ComputeInverseAngleAxis()*, which takes a generic rotation matrix as input and returns the values of the axis and the angle of rotation.
In order to correctly construct the algorithm we need to check that the matrix respects certain criteria. In fact, a matrix, to be considered a rotational one, must respect these key factors:

- ***Dimension***: 3x3 matrix
- ***Orthogonality***: $RR^T = R^T R = I_n$
- ***det(R) = 1***

Upon confirming the validity of the input matrix, we need to calculate the angle of rotation θ, which is determined as $\frac{acos((R_1 1+R_2 2+R_3 3)-1)}{2}$. Next, we compute the eigenvalues and eigenvectors, which we will analyze to identify the correct axis of rotation, which is the eigenvector corresponding to the eigenvalue equal to 1.
Then we simply apply this function on the rotation matrix ${}^a_bR$, obtaining the rotation axis **v** $= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$ and the rotation angle $\theta = \frac{\pi}{2}$



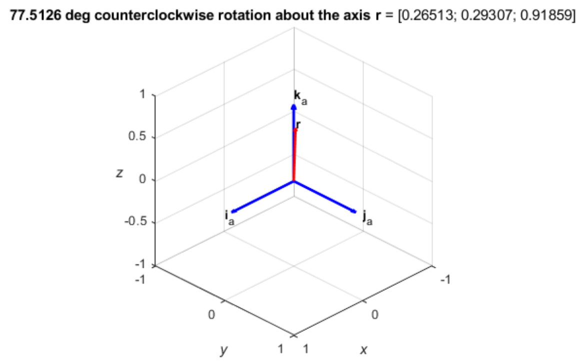


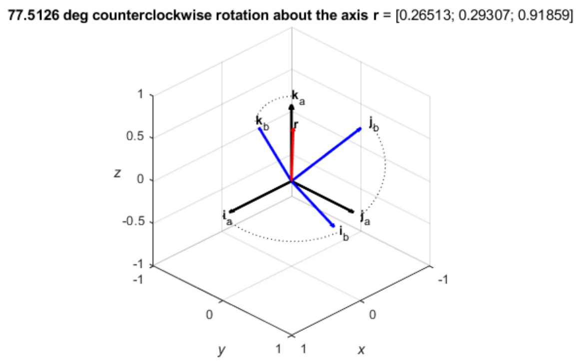
3.3 Q2.3

Here we are able to study the rotation corresponding to the rotation matrix ${}^{c}_{b}R$ that is defined as the rotation from the frame $< c >$ to $< w >$ and again from the frame $< w >$ to $< b >$.
This matrix can be computed as the product between the two smaller rotations, as follows: ${}^{c}_{b}R = {}^{c}_{w}R \cdot {}^{w}_{b}R$, where ${}^{w}_{b}R$ = ${}^{a}_{b}R$, since $< w >$ = $< a >$ and:

$$
{}^{c}_{w}R = \begin{bmatrix} 0.835959 & -0.283542 & -0.46986 \\ 0.271321 & 0.957764 & -0.0952472 \\ 0.47703 & -0.0478627 & 0.877583 \end{bmatrix} \tag{14}
$$

So, after applying the function *ComputeInverseAngleAxis()* we can affirm that $< c >$ rotates around the axis **v** = $\begin{bmatrix} 0.2651 & 0.2931 & 0.9186 \end{bmatrix}$ with an angle of rotation $\theta = 1.3528$ *rad* resulting in the frame $< b >$.





4 Exercise 3

In the context of spatial orientations and rotations, it is often necessary to represent them using rotation matrices. Any element of the special orthogonal group **SO(3)** can be identified by three parameters. One common approach is to construct a rotation matrix with respect to a fixed reference frame by performing three successive rotations about specific axes. The three rotation angles become the parameters used to define the final rotation matrix.

4.1 Q3.1

In the exercise, we consider two reference frames $< w >$ and $< b >$. We assume that $< w >$ is the fixed frame and $< b >$ is initially aligned with $< w >$. The orientation of $< b >$, initially coincident with $< w >$, can be changed through three successive rotations around its own unit vectors.

Here is the sequence of rotations:

1. Rotate $< b >$ by an angle $\alpha = 30$ around its z-axis. This changes the orientation of $< b >$, giving it a new set of x and y axes while keeping the z-axis the same.
2. Rotate $< b >$ around its current y-axis by an angle $\beta = 45$.
3. Finally, rotate the updated frame by an angle $\gamma = 15$ around the current x-axis.

The elementary rotation matrices for rotations around the x, y, or z axes are defined as follows:

$$R_z(\alpha) = \begin{bmatrix} cos(\frac{\pi}{6}) & -sin(\frac{\pi}{6}) & 0 \\ sin(\frac{\pi}{6}) & cos(\frac{\pi}{6}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$R_y(\beta) = \begin{bmatrix} cos(\frac{\pi}{4}) & 0 & sin(\frac{\pi}{4}) \\ 0 & 1 & 0 \\ -sin(\frac{\pi}{4}) & 0 & cos(\frac{\pi}{4}) \end{bmatrix} \tag{16}$$

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\frac{\pi}{12}) & -sin(\frac{\pi}{12}) \\ 0 & sin(\frac{\pi}{12}) & cos(\frac{\pi}{12}) \end{bmatrix} \tag{17}$$

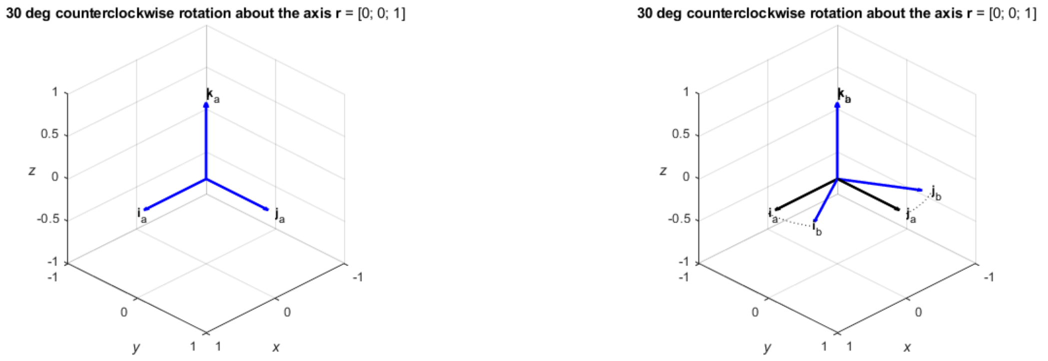
These elementary rotation matrices represent the individual rotations that transform the orientation of $< b >$ with respect to $< w >$.

Remark: The sequence of rotations used in this exercise is reminiscent of yaw, pitch, and roll rotations. Yaw corresponds to rotation around the vertical (z) axis, pitch corresponds to rotation around the lateral (y) axis, and roll corresponds to rotation around the longitudinal (x) axis.

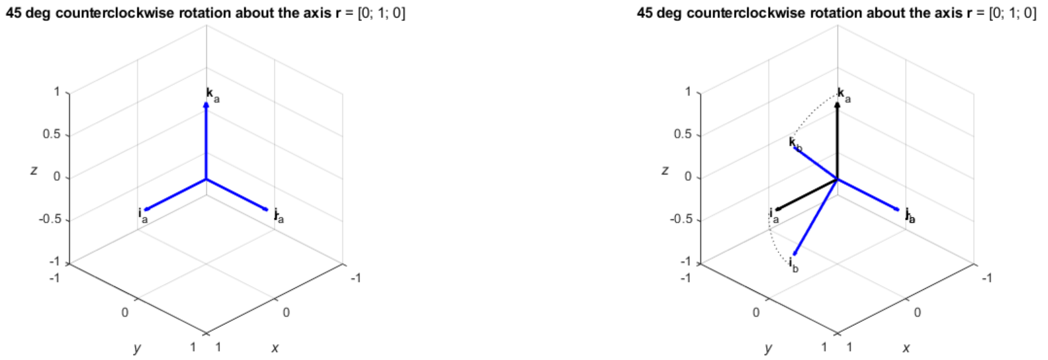
4.2 Q3.2

Now we need to study the axis and the angle of rotation for each matrix obtained in the previous question, obtaining the resulting frames by computing the *Inverse Angle Axis*:

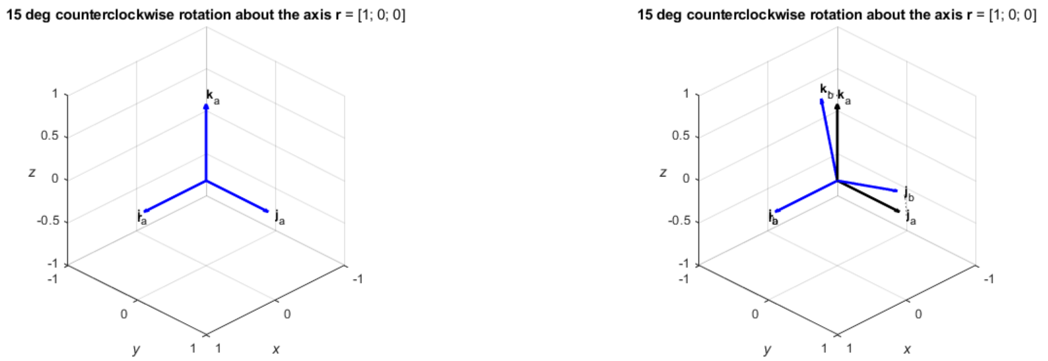
- Rotation matrix from frame $< w >$ to frame $< b >$ by rotating around z-axis of an angle α = 30°:



- Rotation matrix from frame $< w >$ to frame $< b >$ by rotating around y-axis of an angle β = 45°:



- Rotation matrix from frame $< w >$ to frame $< b >$ by rotating around x-axis of an angle γ = 15°:



4.3 Q3.3

In this exercise We obtained the rotational matrix ${}^{a}_{b}R$ which corresponds to the three successive rotations of $< b >$ around its z-y-x current axes, achieved by multiplying the three rotational matrices obtained in Q3.2:

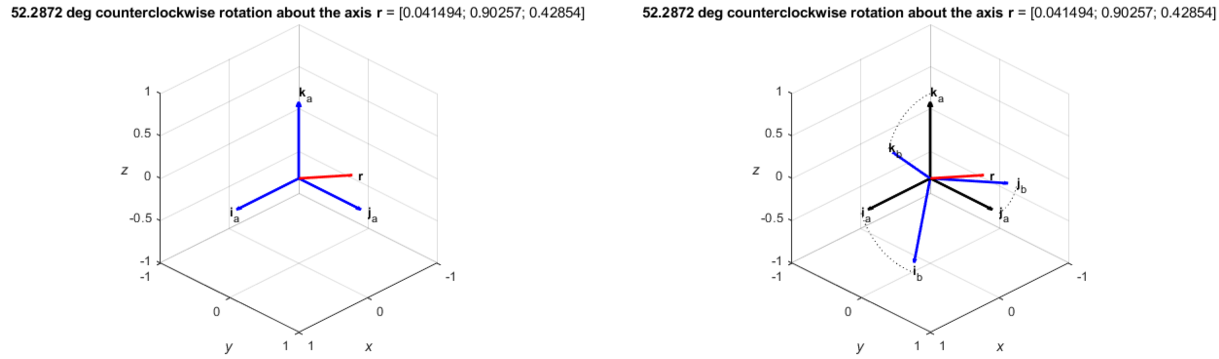
$$ {}^{w}_{b}R(\alpha, \beta, \gamma) = R_z(\alpha) * R_y(\beta) * R_x(\gamma) $$

This new matrix is referred to as an z-y-x Euler angles representation of an element of SO(3). In this representation ”z-y-x” refers to the order of the three successive current rotation axes.

Remark: It’s important to note that there are a total of *12 possible Euler angle choices* corresponding to consecutive rotations. Occasionally, these 12 possible sets of angles are grouped in two sets of 6 elements. The sets related to rotations around two different current axes (*xyx, xzx, yxy, yzy, zxz, zyz*) are commonly called ***Euler angles***, while those related to rotations around three distinct current axes (*xyz, xzy, yxz, yzx, zxy, zyx*) are known as ***Roll*** (rotation around x), ***Pitch*** (rotation around y) and ***Yaw*** (rotation around z) angles.

With that said, we can refer to this matrix as a ***YPR representation***. This represents a sequence of Yaw, Pitch and Roll rotations.

Upon applying the inverse angle-axis representation, we can study the rotational matrix and observe that the rotation is along the axis **v** = $\begin{bmatrix} 0.041494 & 0.90257 & 0.42854 \end{bmatrix}$ with a rotation of $\theta = 58.2872.°$

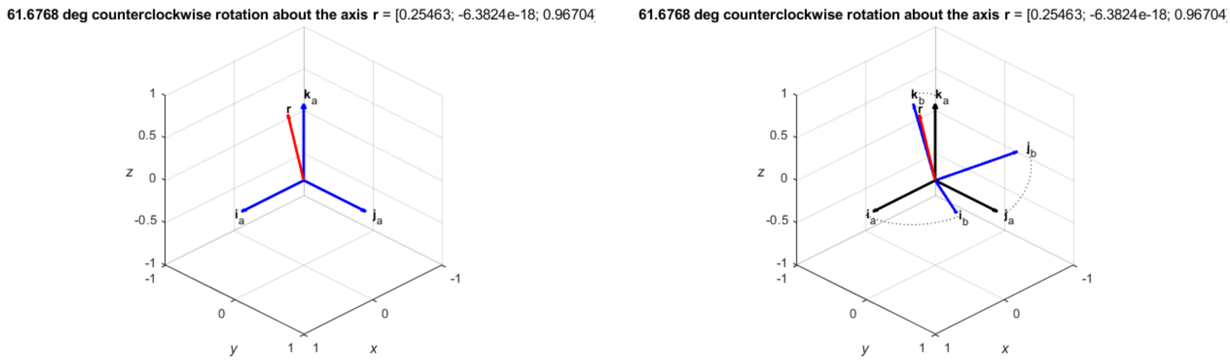


4.4 Q3.4

In this exercise, we compute the z-x-z representation (Euler angle representation) of the rotation matrix:

$$ {}^{w}_{b}R = R_z(\alpha) * R_x(\gamma) * R_z(\alpha) $$

Then, just like we did in Q3.3, we solve the Inverse Equivalent Angle-Axis Problem for the obtained orientation matrix.



In this case, we observe that the rotation corresponds to the axis **v** = $\begin{bmatrix} 0.25463 & -6.3824e-18 & 0.96704 \end{bmatrix}$ with a rotation of $\theta = 61.6788°$.

5 Exercise 4

In the fourth section of this exercise, we will explore the process of transforming a quaternion into its corresponding orientation matrix. Quaternions are a type of hypercomplex number, used to represent rotations and orientations in 3D space. Unlike complex numbers, which have a real part and an imaginary part, quaternions have four components: one real part and three imaginary parts. They are typically represented as follows:

$$q = q_0 + q_1 i + q_2 j + q_3 k$$

Where:

- q is the quaternion;
- q_0 is the real part of the quaternion;
- q_1, q_2, q_3 are the imaginary parts of the quaternion;
- i, j, k are the basis elements of the quaternion.

After computing the rotation matrix associated with the given quaternion we will solve the inverse angle-axis problem for ${}^a_bR$ to study how the quaternion rotate the frame $< a >$ in the resulting frame $< b >$.

5.1 Q4.1

Firstly we need to convert into its corresponding rotation matrix the given quaternion:

$$q = 0.1647 + 0.31583i + 0.52639j + 0.77204k$$

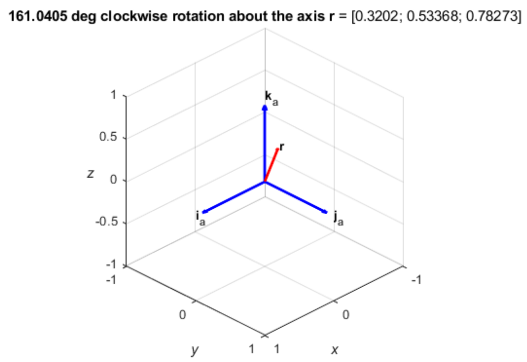
This conversion process is realized through a custom function, *quatToRot()*, which performs the conversion manually, row-by-row. By adding each row to the matrix, we arrive at a 3x3 rotation matrix, denoted as ${}^a_bR$. The resulting matrix effectively represents the same 3D rotation as the original quaternion.

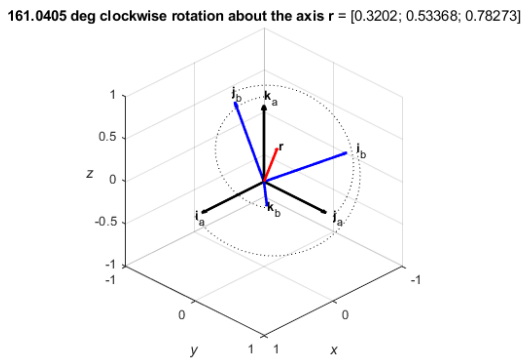
$${}^a_bR = \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix} = \begin{bmatrix} -0.7463 & 0.5868 & 0.3143 \\ 0.0782 & -0.3916 & 0.9168 \\ 0.6611 & 0.7088 & 0.2463 \end{bmatrix} \tag{18}$$

Comparison to MATLAB Functions: It's worth noting that established software tools like MATLAB provide built-in functions for handling quaternion operations. In MATLAB, you can create a quaternion from parameters $q1$, $q2$, $q3$, and $q4$ and then employ the *rotmat()* function to effortlessly convert the quaternion into a rotation matrix.

5.2 Q4.2

Now we just need to study the axis and the angle of rotation of this matrix, obtaining the resulting frames by computing the Inverse Angle Axis:





We can observe that the frame $< a >$ undergoes a rotation around the axis defined by **v** = [0.3202 0.53368 0.78273], with an associated rotation angle θ = 161.0405°. This transformation results in the establishment of the new frame, denoted as $< b >$.