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

Professors:

Giovanni Indiveri
Enrico Simetti
Giorgio Cannata

Author:

Gavagna Veronica

Student ID:

s5487110

Tutors:

Andrea Tiranti
Francesco Giovinazzo

November 19, 2022

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, 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 predefined 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:

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

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 $\theta = 30°$
- **Q1.3** $\mathbf{v} = [0, 1, 0]$ and $\theta = \pi/4$
- **Q1.4** $\mathbf{v} = [0, 0, 1]$ and $\theta = \pi/2$
- **Q1.5** $\mathbf{v} = [0.408, 0.816, -0.408]$ and $\theta = 0.2449$
- **Q1.6** $\rho = [0, \pi/2, 0];$
- **Q1.7** $\rho = [0.4, -0.3, -0.3];$
- **Q1.8** $\rho = [-\pi/4, -\pi/3, \pi/8];$

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 10.

Q2.1 Compute the orientation matrix ${}^a_bR$.
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 ${}^c_bR$.

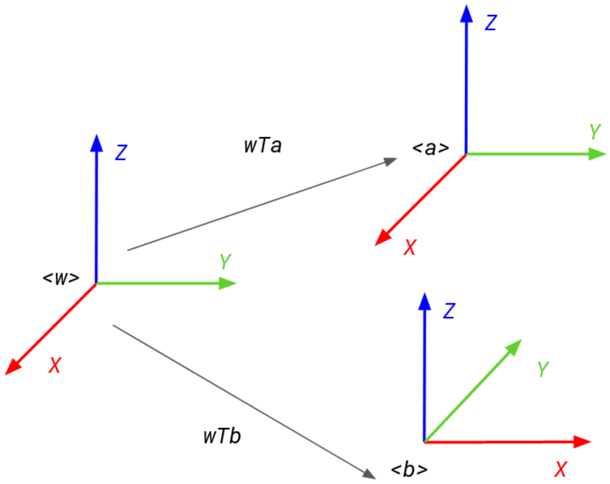

Figure 1: Exercise 2 frames

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 $30°$ around the z-axis of $< w >$
- $< b >$ is rotated of $45°$ around the y-axis of $< w >$
- $< b >$ is rotated of $15°$ 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: $q = 0.8924 + 0.23912i + 0.36964j + 0.099046k$ expressing how a reference frame $< b >$ is rotated with respect to $< a >$:
Q4.1 Compute the equivalent rotation matrix
Q4.2 Solve the Inverse Equivalent Angle-Axis Problem for the obtained orientation matrix

2 Exercise 1

In this exercise we worked on the equivalent Angle-Axis Representation (Exponential representation).
We see that the orientation matrix obtained by the pure rotation around a vector **v** of an angle θ of frame $< b >$ with respect to frame $< a >$ (initially coinciding), is equivalent to the representation of the angle-axis representation described by the Rodrigues Formula:

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

2.1 Q1.1 - Solution

From the Rodrigues Formula I implemented the code of "*ComputeAngleAxis.m*" to compute the rotation matrix ${}^a_bR$ of frame $< b >$ with respect to $< a >$, given as inputs an angle of rotation (θ) and a axis vector (**v**).

To compute the rotational matrix ${}^a_bR$, I need the identity matrix and the skew symmetric matrix of the vector **v**. To get the skew symmetric matrix, I implemented a function called "*SkewDim3.m*" where I get, as input a vector of dimension 3 and, as output the skew symmetric matrix.

The starting configuration is the same for every rotation and it is showed in Figure 2.

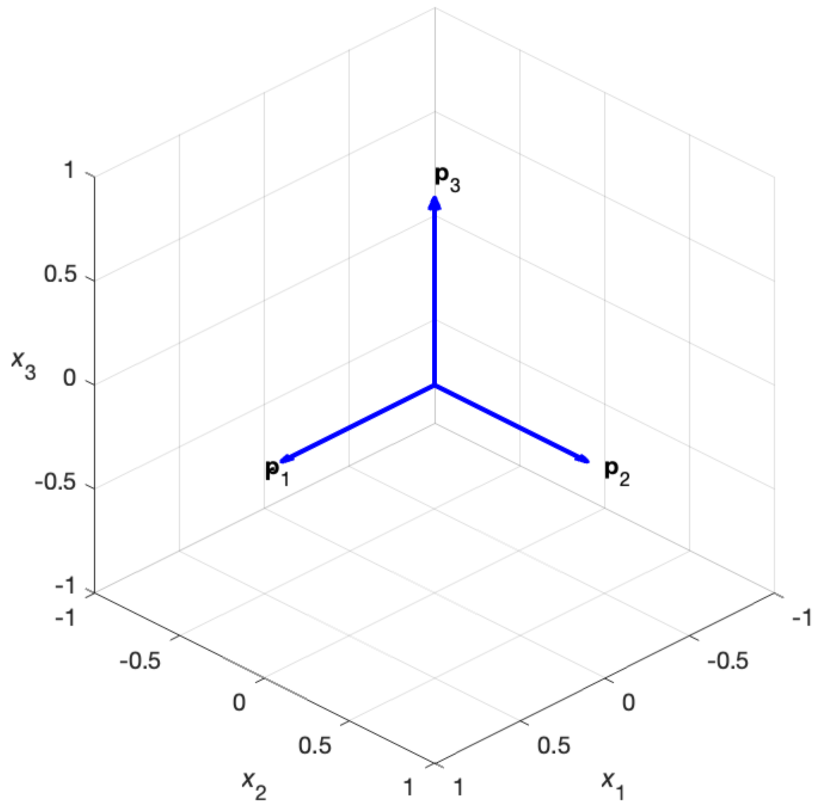

Figure 2: Starting configuration: frames $< a >$ and $< b >$ coinciding

2.2 Q1.2 - Solution

I initialized the axis vector **v** and the angle θ with the following given data: **v** $= [1, 0, 0]$ and $\theta = \pi/6$.
I compute the rotation matrix using the function "*ComputeAngleAxis.m*" and plot the solution displaying the rotation using the function "*plotRotation.m*".
The resulted rotational matrix is:

$$ {}^a_bR = \begin{bmatrix} 1.0000 & 0 & 0 \\ 0 & 0.8660 & -0.5000 \\ 0 & 0.5000 & 0.8660 \end{bmatrix} $$

The result graph is showed in Figure 3.

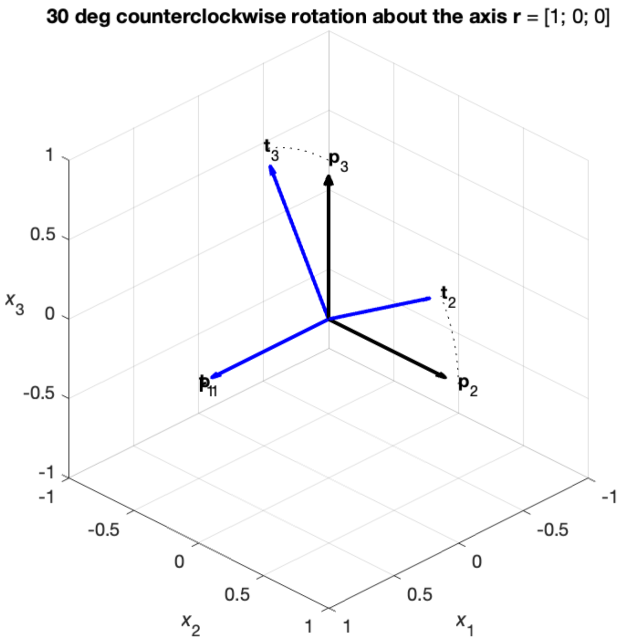

Figure 3: Graph rotation 1.2: Rotation around x axis of $30°$
Black: initial configuration - Blue: rotated configuration

As showed in the graph above, a rotation of $30°$ is computed around the x axis.

2.3 Q1.3 - Solution

I initialized the vector **v** and the angle θ with the following given data: **v** $= [0, 1, 0]$ and $\theta = \pi/4$.
I compute the rotation matrix using the function "*ComputeAngleAxis.m*" and plot the solution displaying the rotation using the function "*plotRotation.m*".
The resulted rotational matrix is:

$$ {}^a_bR = \begin{bmatrix} 0.7071 & 0 & 0.7071 \\ 0 & 1.0000 & 0 \\ -0.7071 & 0 & 0.7071 \end{bmatrix} $$

The result graph is showed in Figure 4.

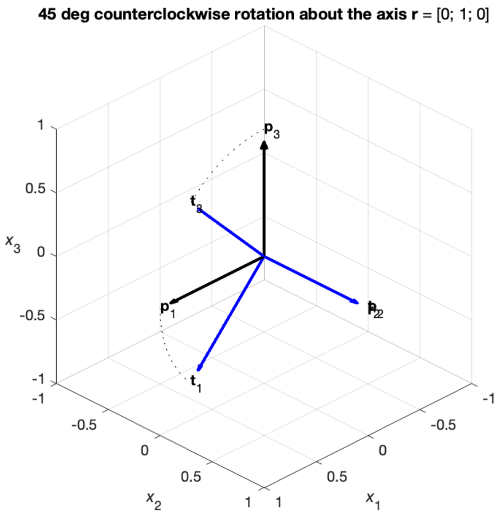

Figure 4: Graph rotation 1.3: Rotation around y axis of $45°$
Black: initial configuration - Blue: rotated configuration

As showed in the graph above, a rotation of $45°$ is computed around the y axis.

2.4 Q1.4 - Solution

I initialized the axis vector **v** and the angle θ with the following given data: **v** $= [0, 0, 1]$ and $\theta = \pi/2$.
I compute the rotation matrix using the function "*ComputeAngleAxis.m*" and plot the solution displaying the rotation using the function "*plotRotation.m*".
The resulted rotational matrix is:

$$ {}^a_bR = \begin{bmatrix} 0.0000 & -1.0000 & 0 \\ 1.0000 & 0.0000 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix} $$

The result graph is showed in Figure 5.

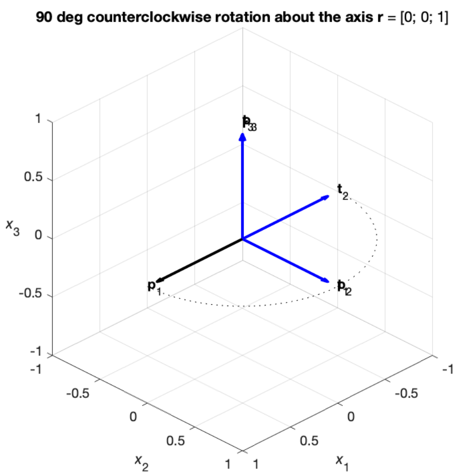

Figure 5: Graph rotation 1.4: Rotation around z axis of $90°$
Black: initial configuration - Blue: rotated configuration

As showed in the graph above, a rotation of $90°$ is computed around the z axis.

2.5 Q1.5 - Solution

I initialized the axis vector **v** and the angle θ with the following given data: **v** $= [0.408, 0.816, -0.408]$ and $\theta = 0.2449$.
I compute the rotation matrix using the function "*ComputeAngleAxis.m*" and plot the solution displaying the rotation using the function "*plotRotation.m*".

The resulted rotational matrix is:

$$ {}^a_bR = \begin{bmatrix} 0.9752 & 0.1089 & 0.1929 \\ -0.0890 & 0.9901 & -0.1089 \\ -0.2028 & 0.0890 & 0.9752 \end{bmatrix} $$

The result graph is showed in Figure 6.

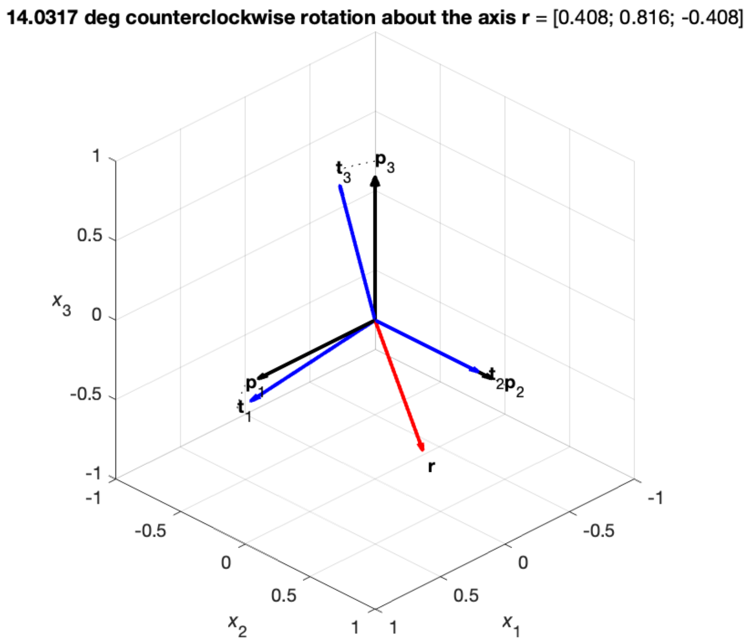

Figure 6: Graph rotation 1.5: Rotation around **v**=[0.408,0.816,-0.408] vector of $14.03°$
Black: initial configuration - Blue: rotated configuration - Red: axis vector

As showed in the graph above, a rotation of $14.03°$ is computed around the **r** axis vector.

2.6 Q1.6 - Solution

I initialized the axis vector **v** and the angle θ with the following given data: $\rho = [0, \pi/2, 0]$. From the rotation vector ρ, I calculate the angle of rotation $\theta = \|\rho\|)$ and the axis vector **v** $= \rho \,/\, \theta = \rho \,/\, \|\rho\|$.
I compute the rotation matrix using the function "*ComputeAngleAxis.m*" and plot the solution displaying the rotation using the function "*plotRotation.m*".

The resulted rotational matrix is:

$$ {}^a_bR = \begin{bmatrix} 0.0000 & 0 & 1.0000 \\ 0 & 1.0000 & 0 \\ -1.0000 & 0 & 0.0000 \end{bmatrix} $$

The result graph is showed in Figure 7.

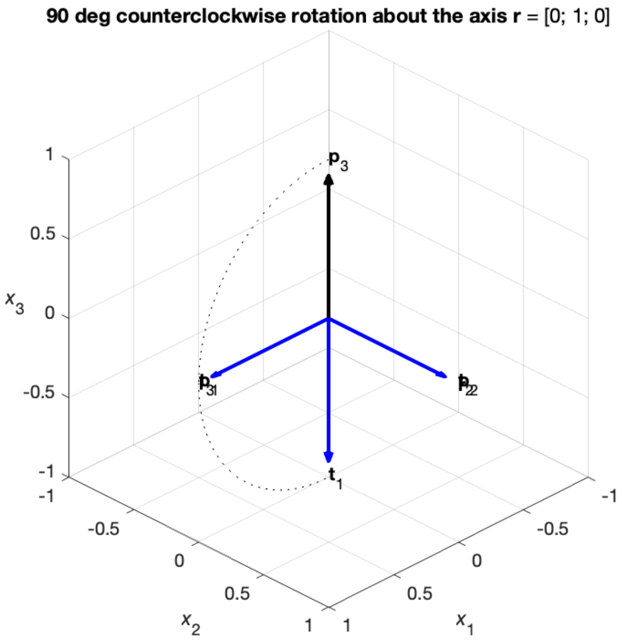

Figure 7: Graph rotation 1.6: Rotation around **v**=[0,1,0] vector of $90°$
Black: initial configuration - Blue: rotated configuration

As showed in the graph above, a rotation of $90°$ is computed around the y axis vector.

2.7 Q1.7 - Solution

I initialized the axis vector **v** and the angle θ with the following given data: $\rho = [0.4, -0.3, -0.3];$. From the rotation vector ρ, I calculate the angle of rotation $\theta = \|\rho\|)$ and the axis vector **v** $= \rho \,/\, \theta = \rho \,/\, \|\rho\|$.
I compute the rotation matrix using the function "*ComputeAngleAxis.m*" and plot the solution displaying the rotation using the function "*plotRotation.m*".

The resulted rotational matrix is:

$$
{}^{a}_{b}R = \begin{bmatrix} 0.9125 & 0.2250 & -0.3416 \\ -0.3416 & 0.8785 & -0.3340 \\ 0.2250 & 0.4215 & 0.8785 \end{bmatrix}
$$

The result graph is showed in Figure 8.

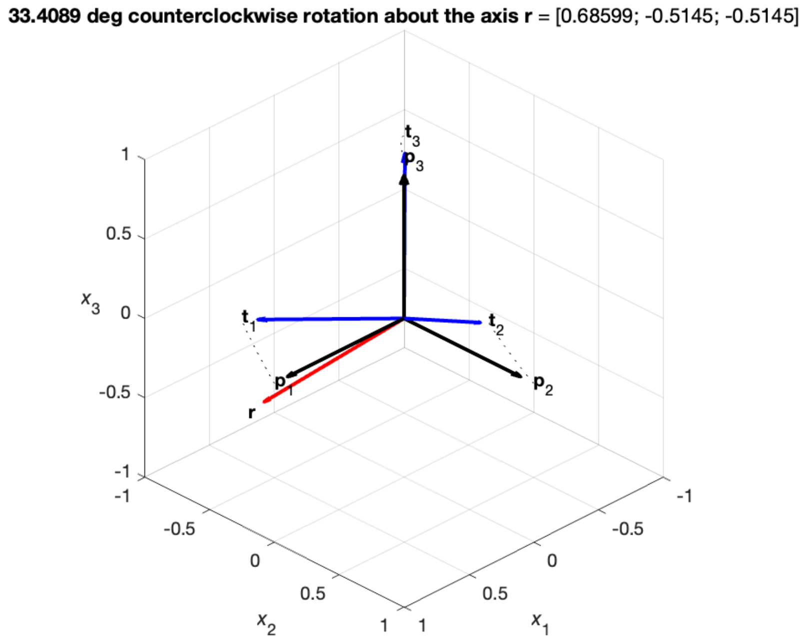

Figure 8: Graph rotation 1.7: Rotation around **v**=[0.68,-0.51,-0.45] axis of $33.04°$
Black: initial configuration - Blue: rotated configuration - Red: axis vector

As showed in the graph above, a rotation of $33.04°$ is computed around the **r** axis vector.

2.8 Q1.8 - Solution

I initialized the vector **v** and the angle θ with the following given data: $\rho = [-\pi/4, -\pi/3, \pi/8];$. From the rotation vector ρ, I calculate the angle of rotation $\theta = \|\rho\|)$ and the axis vector **v** $= \rho \,/\, \theta = \rho \,/\, \|\rho\|$.
I compute the rotation matrix using the function "*ComputeAngleAxis.m*" and plot the solution displaying the rotation using the function "*plotRotation.m*".

The resulted rotational matrix is:

$$
{}^{a}_{b}R = \begin{bmatrix} 0.4661 & 0.0697 & -0.8820 \\ 0.6325 & 0.6709 & 0.3872 \\ 0.6187 & -0.7383 & 0.2686 \end{bmatrix}
$$

The result graph is showed in Figure 9.

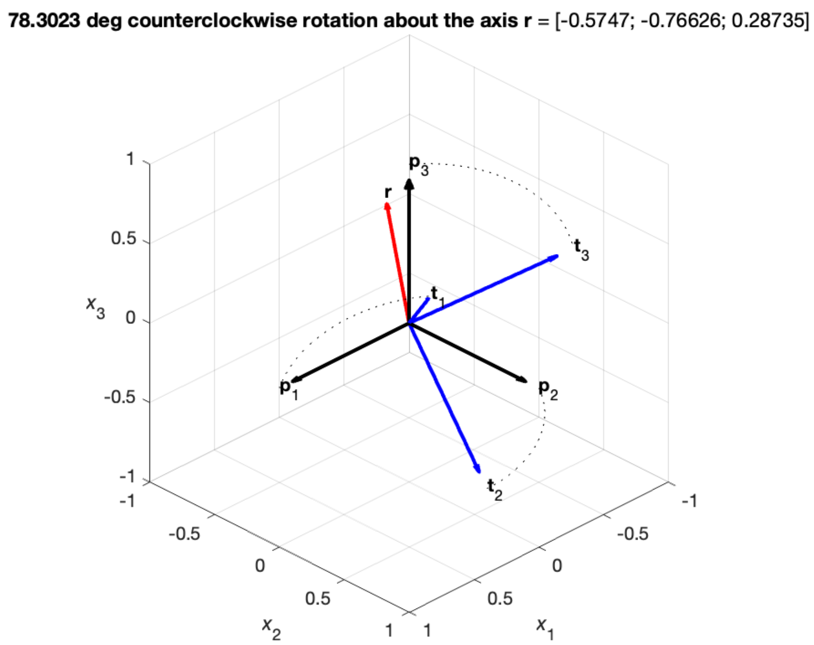

Figure 9: Graph rotation 1.2: Rotation around **v**=[-0.57,-0.76,0.29] axis of $78.30°$
Black: initial configuration - Blue: rotated configuration - Red: axis vector

As showed in the graph above, a rotation of $78.30°$ is computed around the **r** axis vector.

3 Exercise 2

In the exercise 2 we work on the inverse equivalent Angle-Axis Problem.
We see that given the rotation matrix ${}^a_bR$, it is possible to compute the angle of rotation θ and a axis vector **v** by implementing the code of "*ComputeInverseAngleAxis.m*".

3.1 Q2.1 - Solution

The orientation of frame $< a >$ and frame $< b >$ with respect to the world frame $< w >$ is expressed in Figure 10.

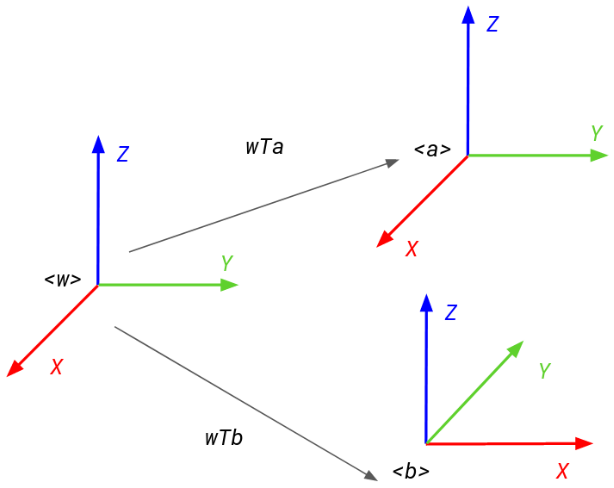

Figure 10: Exercise 2 frames orientation

The orientation matrix ${}^a_bR$ is given by multiplying the transposed matrix ${}^w_aR^T$ with the matrix ${}^w_bR$.

$$ {}^a_b\mathbf{R} = {}^w_a\mathbf{R}^T * {}^w_b\mathbf{R} $$

These two matrices are the result of "*ComputeAngleAxis.m*" giving as inputs $\theta = 0$ and **v**=[0, 0, 0] for ${}^w_aR^T$ and $\theta = \pi/2$ and **v**=[0, 0, 1] for ${}^w_bR$.

The resultant ${}^a_bR$ matrix is: $\quad {}^a_bR = \begin{bmatrix} 0.0000 & -1.0000 & 0 \\ 1.0000 & 0.0000 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}$

3.2 Q2.2 - Solution

Computed ${}^a_bR$, I implemented the code of "*ComputeInverseAngleAxis.m*". Before computing the angle of rotation θ and a axis vector **v** it is necessary to check that the input matrix has dimensions 3x3, that is orthogonal ($R * R^T$ = identity matrix) and that the determinant is not 0. If the input matrix is 3x3, orthogonal with determinant not equal to 0, then it is possible to compute θ and **v** using the following formulas:

$$ \theta = \arccos \frac{(tr(R) - 1)}{2} $$

$$ \mathbf{v} = \frac{vex(R)}{\sin\theta} $$

To compute $vex(R)$, I implemented a function "*vex3.m*" where, given as input a matrix R, it returns the column vector associated to the matrix using the following formula:

$$ vex(R) = \frac{1}{2} \begin{bmatrix} R_{3,2} - R_{2,3} \\ R_{1,3} - R_{3,1} \\ R_{2,1} - R_{1,2} \end{bmatrix} $$

The results obtained are:
θ = 1.5708 = $\pi/2$ **v** = [0; 0; 1]. That verifies our expectations.

The result graph is showed in Figure 12.

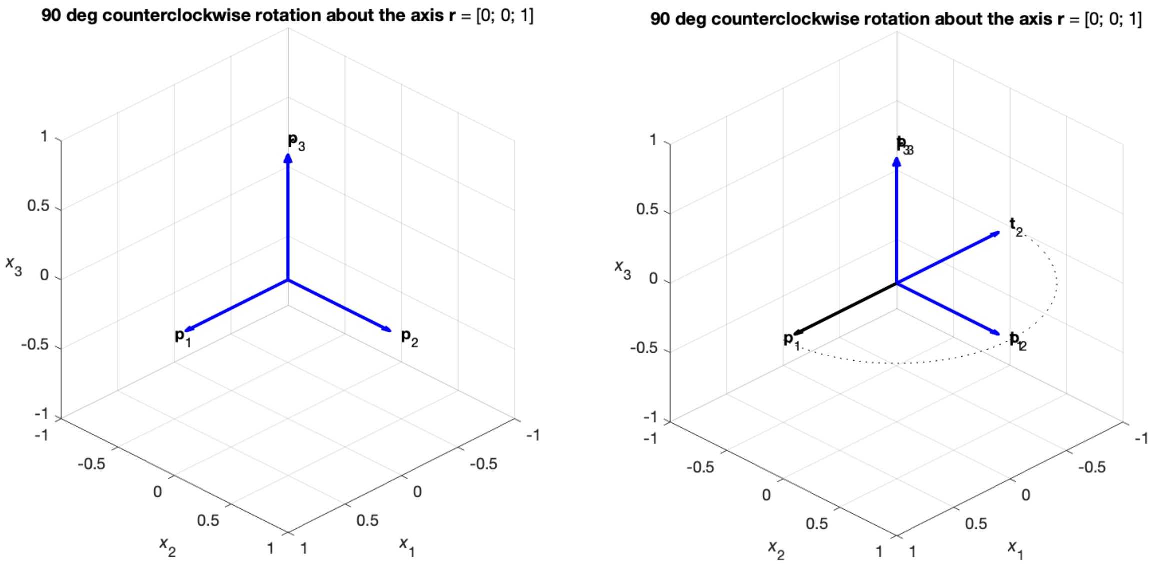

Figure 11: Graph rotation 2.2: Rotation around $\mathbf{v} = [0; 0; 1]$ of $\theta = 1.570$
Left: initial configuration - Right: rotated configuration

As showed in the graphs above, a rotation of $90°$ is computed around the z axis.

3.3 Q2.3 - Solution

The orientation of frame $< w >$ with respect to the world frame $< c >$ is given:

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

The orientation matrix between frame $< c >$ and frame $< b >$ (${}^c_b R$) is given by multiplying the transposed matrix ${}^w_c R^T$ with the matrix ${}^w_b R$.

$$
{}^c_b\mathbf{R} = {}^w_c\mathbf{R}^T * {}^w_b\mathbf{R}
$$

Computed ${}^a_b R$ and before computing the inverse problem, I added to the controls in the function "*ComputeInverseAngleAxis.m*" a rounding to the 5^{th} decimal place when I check if the input matrix is orthogonal and the determinant is not 0, because $R * R^T$ did not came out exactly equal to the identity matrix (diagonal elements where not exactly equal to 1) and the determinant did not came out exactly equal to 0, because of Matlab approximation.

After adding this rounding, the results obtained are:

θ = 1.3529 **v** = [0.2651; 0.2931; 0.9186].

The result graph is showed in Figure 12.

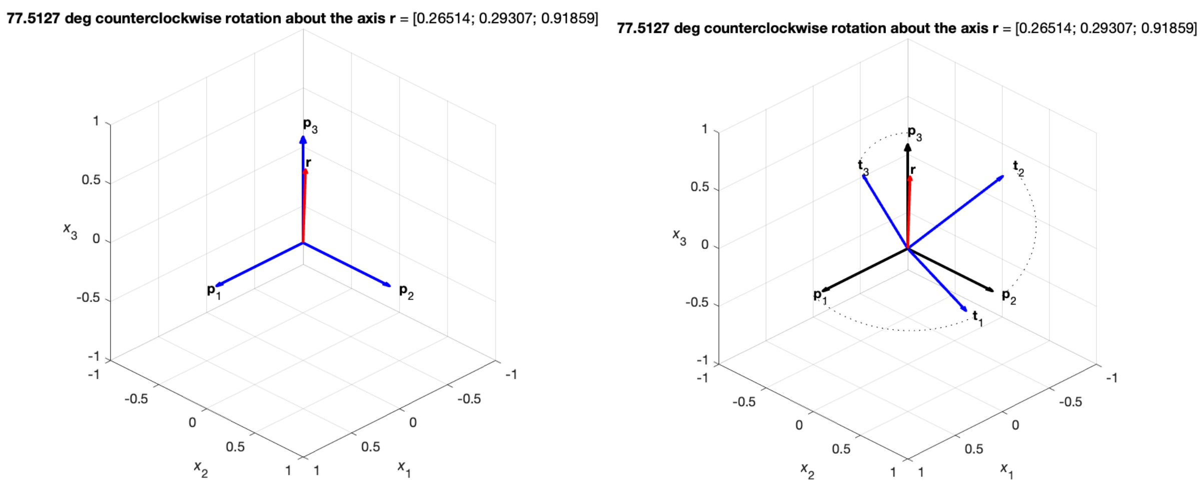

Figure 12: Graph rotation 2.2: Rotation around **v** $= [0.2651; 0.2931; 0.9186]$ of $\theta = 1.3529$
Left: initial configuration - Right: rotated configuration

As showed in the graphs above, a rotation of $77.51°$ is computed around the **r** axis vector.

4 Exercise 3

In the exercise 3 we compare the Euler angles (Z-X-Z) and the Tait-Bryan angles (Yaw-Pitch-Roll).
We see that any orientation matrix can be expressed in terms of three elementary rotations in sequence: it can be about the axis of a fixed coordinate system (extrinsic rotations), or about the axes of a rotating coordinate system (intrinsic rotations) initially aligned with the fixed one.

4.1 Q3.1 - Solution

Given two generic frames $< w >$ and $< b >$, I defined the elementary orientation matrices for frame $< b >$ with respect to frame $< w >$, knowing that:

- $< b >$ is rotated of $30°$ around the z-axis of $< w >$: I initialized $z_{angle} = \pi/6$
- $< b >$ is rotated of $45°$ around the y-axis of $< w >$: I initialized $y_{angle} = \pi/4$
- $< b >$ is rotated of $15°$ around the x-axis of $< w >$: I initialized $x_{angle} = \pi/12$

Then, I computed and displayed the results:

- the rotation matrix ${}^w_bR_z$ from $< w >$ to frame $< b >$ by rotating around z-axes, using "*ComputeAngleAxis.m*" giving as inputs z_{angle} and z axis vector.
- the rotation matrix ${}^w_bR_y$ from $< w >$ to frame $< b >$ by rotating around y-axes, using "*ComputeAngleAxis.m*" giving as inputs y_{angle} and y axis vector.
- the rotation matrix ${}^w_bR_y$ from $< w >$ to frame $< b >$ by rotating around x-axes, using "*ComputeAngleAxis.m*" giving as inputs x_{angle} and x axis vector.

The matrices obtained are:

$$w_bR_z = \begin{bmatrix} 0.8660 & -0.5000 & 0 \\ 0.5000 & 0.8660 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$$w_bR_y = \begin{bmatrix} 0.7071 & 0 & 0.7071 \\ 0 & 1.0000 & 0 \\ -0.7071 & 0 & 0.7071 \end{bmatrix}$$

$$w_bR_x = \begin{bmatrix} 1.0000 & 0 & 0 \\ 0 & 0.9659 & -0.2588 \\ 0 & 0.2588 & 0.9659 \end{bmatrix}$$

4.2 Q3.2 - Solution

I computed the equivalent angle-axis representation for each elementary rotation, using "*ComputeInverseAngleAxis.m*" from the matrices obtained before to obtain the angle of rotation θ and the axis vector **v**.
The results obtained for each case are:

- ${}^w_bR_z \;\rightarrow\; \theta = 0.5236 \qquad$ **v** $= [0;\ 0;\ 1;]$
- ${}^w_bR_y \;\rightarrow\; \theta = 0.7854 \qquad$ **v** $= [0;\ 1;\ 0;]$
- ${}^w_bR_x \;\rightarrow\; \theta = 0.2618 \qquad$ **v** $= [1;\ 0;\ 0;]$

The resulting graphs are showed in Figure 13, Figure 14, Figure 15.

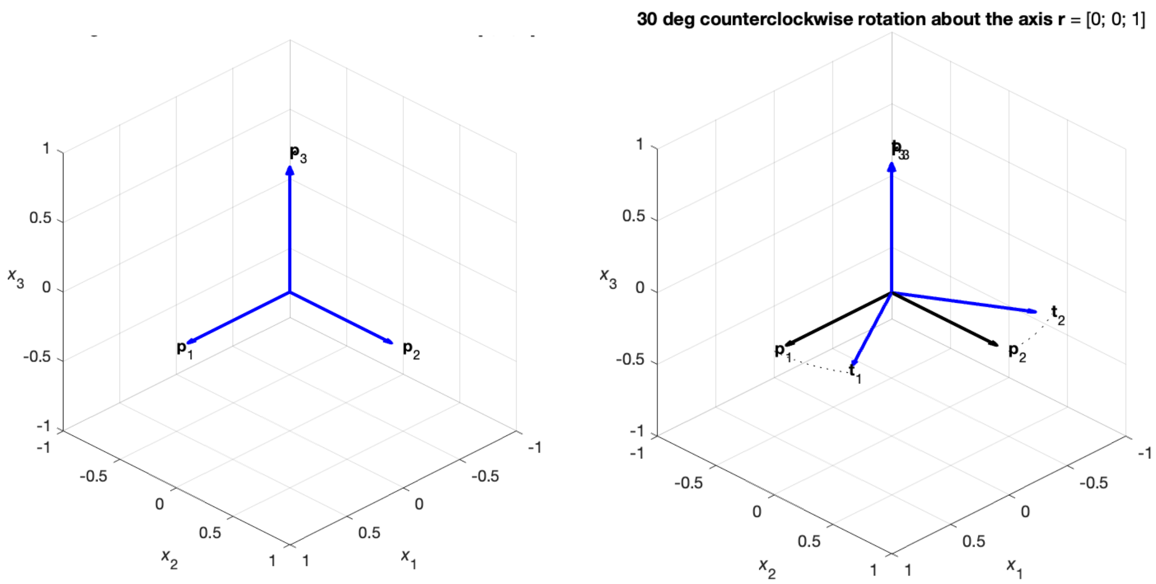

Figure 13: Graph rotation 3.2a: Rotation around z axis of $\theta = 30°$
Left: initial configuration - Right: rotated configuration

As showed in the graphs above, a rotation of $30°$ is computed around the z axis, confirming what I was expected.

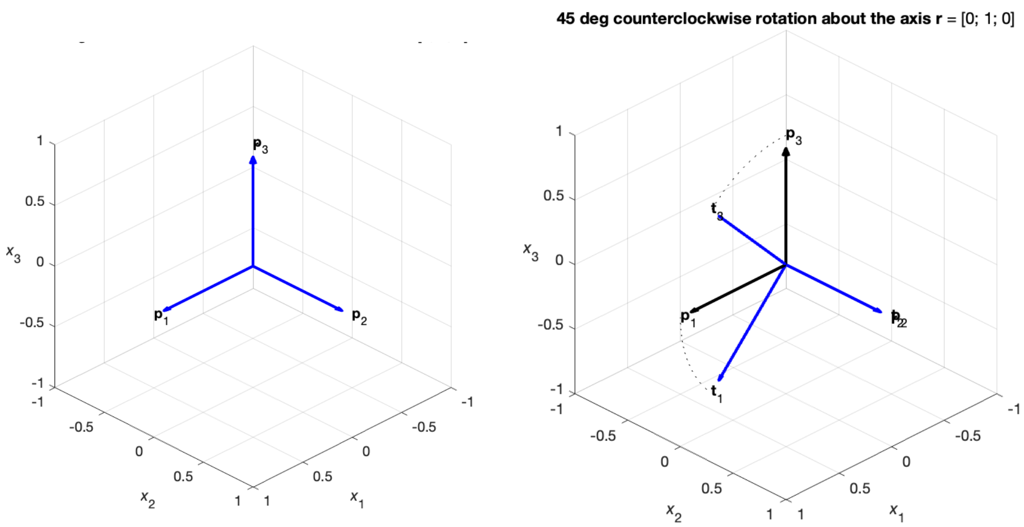

Figure 14: Graph rotation 3.2b: Rotation around y axis of $\theta = 45°$
Left: initial configuration - Right: rotated configuration

As showed in the graphs above, a rotation of $45°$ is computed around the y axis, confirming what I was expected.

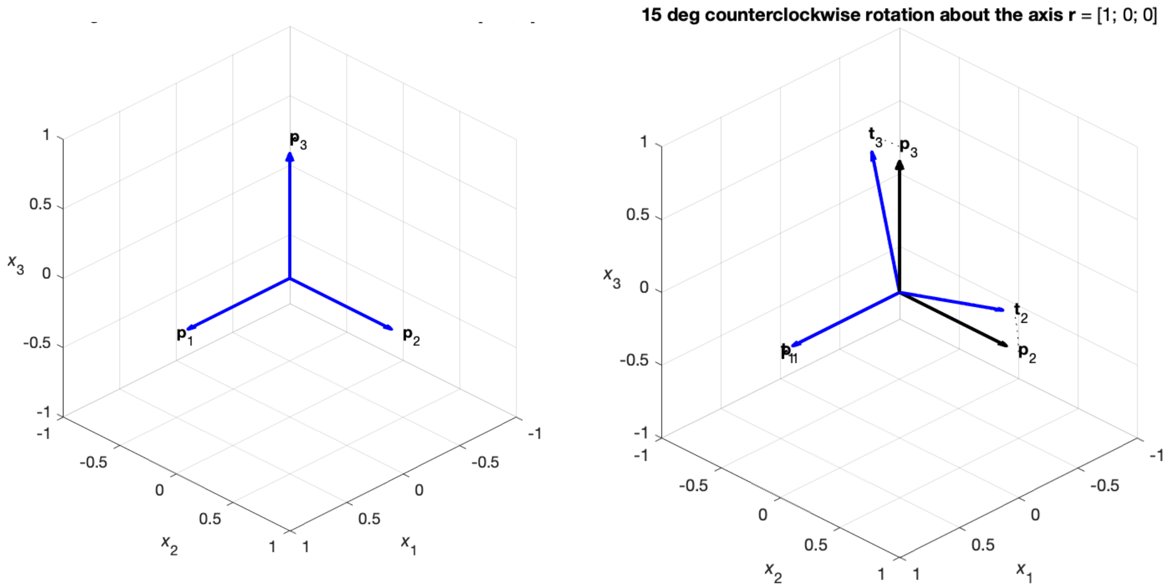

Figure 15: Graph rotation 3.2b: Rotation around x axis of $\theta = 15°$
Left: initial configuration - Right: rotated configuration

As showed in the graphs above, a rotation of $15°$ is computed around the x axis, confirming what I was expected.

4.3 Q3.3 - Solution

I computed the rotation matrix corresponding to the z-y-x representation, using the following formula:

$$R_{zyx} = {}^w_b R_z * {}^w_b R_y * {}^w_b R_x$$

This matrix represents the whole rotation around z axis of $\theta = 30°$, y axis of $\theta = 45°$, and x axis of $\theta = 15°$.

Then, I get the angle of rotation θ and the axis vector **v** using the function "*ComputeInverseAngleAxis.m*".
The result is the following: $\theta = 0.9126$ **v** $= [0.0415;\ -0.7609;\ 0.6475;]$.
The resulting graph is showed in Figure 16.

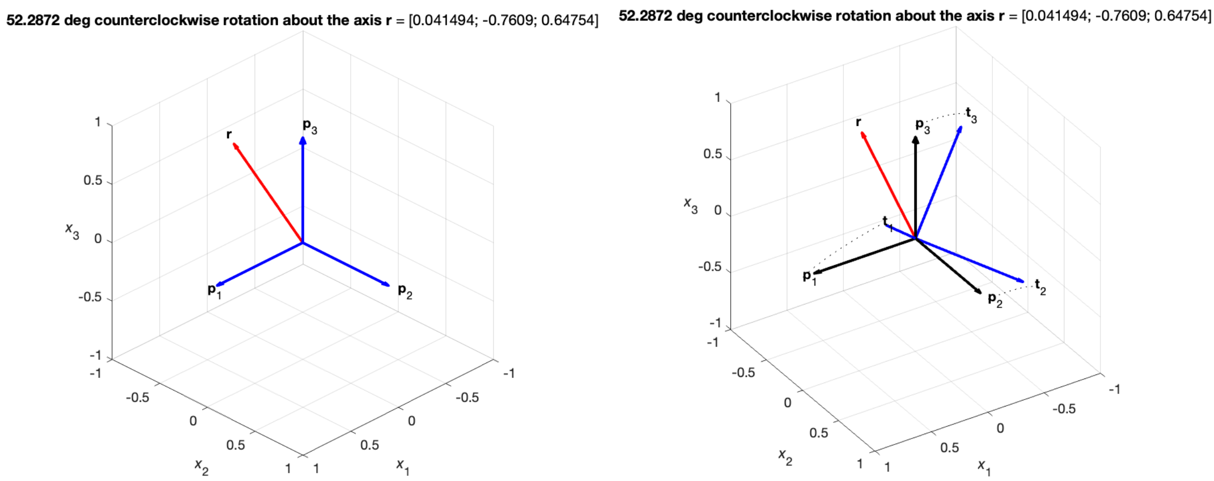

Figure 16: Graph rotation 3.3: Whole rotation around z axis of $\theta_z = 30°$, y axis of $\theta_y = 45°$, x axis of $\theta_x = 15°$
Left: initial configuration - Right: rotated configuration

As showed in the graphs above, a rotation around all the 3 axis of $\theta_z = 30°$, $\theta_y = 45°$, $\theta_x = 15°$ is computed, confirming what I was expected.

Q3.4 - Solution As in the previous point, I computed the rotation matrix corresponding to the z-x-z representation, using the following formula:

$$R_{xyz} =_b^w R_z * {}_b^w R_x * {}_b^w R_z$$

This matrix represents the whole rotation around z axis of $\theta = 30°$, x axis of $\theta = 15°$, and z axis of $\theta = 30°$.

Then, I get the angle of rotation θ and the axis vector **v** using the function "*ComputeInverseAngleAxis.m*".
The result is the following: $\theta = 1.0765$ **v** $= [-0.2205;\ 0.1273;\ 0.9670;]$.
The resulting graph is showed in Figure 17.

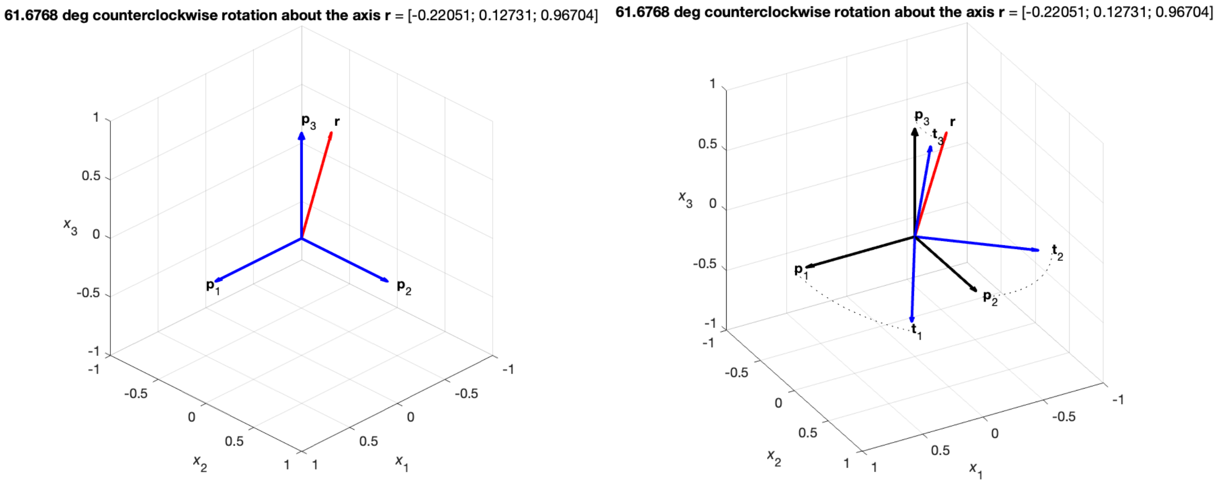

Figure 17: Graph rotation 3.3: Whole rotation around z axis of $\theta_z = 30°$, x axis of $\theta_x = 15°$, z axis of $\theta_z = 30°$
Left: initial configuration - Right: rotated configuration

As showed in the graphs above, a rotation around z axis, x axis and z axis of $\theta_z = 30°$, $\theta_x = 15°$, $\theta_z = 30°$ is computed, confirming what I was expected.

5 Exercise 4

In the exercise 4 we worked with the given following quaternion: $q = 0.8924 + 0.23912i + 0.36964j + 0.099046k$ expressing how a reference frame $< b >$ is rotated with respect to $< a >$.

5.1 Q4.1 - Solution

I implemented the function "*quatToRot.m*" to compute the equivalent rotation matrix of the given quaternion. In this function I get as input the 4 elements representing the quaternion: $[q_0 = 0.8924, q_1 = 0.23912, q_2 = 0.36964, q_3 = 0.099046]$, and computing each element through the following formulas I get, as output, the rotation matrix.

$$r_{00} = 2 * (q_0 * q_0 + q_1 * q_1) - 1$$
$$r_{01} = 2 * (q_1 * q_2 - q_0 * q3)$$
$$r_{02} = 2 * (q_1 * q_3 + q_0 * q_2)$$

$$r_{10} = 2 * (q_1 * q_2 + q_0 * q_3)$$
$$r_{11} = 2 * (q_0 * q_0 + q_2 * q_2) - 1$$
$$r_{12} = 2 * (q_2 * q_3 - q_0 * q_1)$$
$$r_{20} = 2 * (q_1 * q_3 - q_0 * q_2)$$

$$r_{21} = 2 * (q_2 * q_3 + q_0 * q_1)$$
$$r_{22} = 2 * (q_0 * q_0 + q_3 * q_3) - 1$$

$$R = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix}$$

The results of the computation obtained is:

$$R = \begin{bmatrix} 0.7071 & -0.0000 & 0.7071 \\ 0.3536 & 0.8660 & -0.3536 \\ -0.6124 & 0.5000 & 0.6124 \end{bmatrix}$$

The result graph is showed in Figure 18.

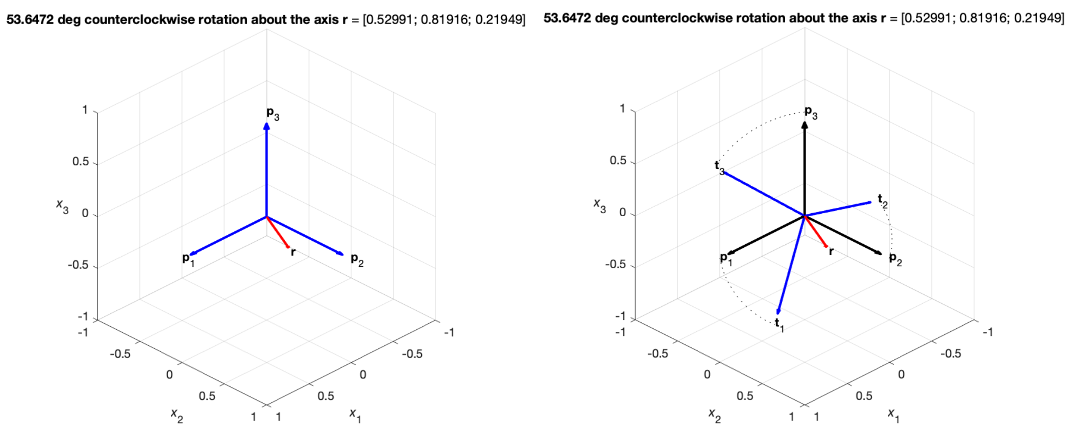

Figure 18: Graph rotation 4.1: Rotation around $\mathbf{v} = [0.52991; 0.81916; 0.21949]$ of $\theta = 53.65°$
Left: initial configuration - Right: rotated configuration

5.2 Q4.2 - Solution

To check if the results are correct I used the build-in Matlab functions in two different way: first with "*quat2rotm*" and "*quaternion*", and second one with "*rotmat*" and "*quaternion*". The rotation matrix and the graph obtained are equal to the results obtained in exercise 4.1 in both cases, as I expected.

6 List of figures

List of Figures