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:

Galvagni Gianluca

Student ID:

s552188

Professors:

Giovanni Indiveri
Enrico Simetti
Giorgio Cannata

Tutors:

Andrea Tiranti
Francesco Giovinazzo

November 14, 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 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:

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

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 1.

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 ${}^c_bR$.

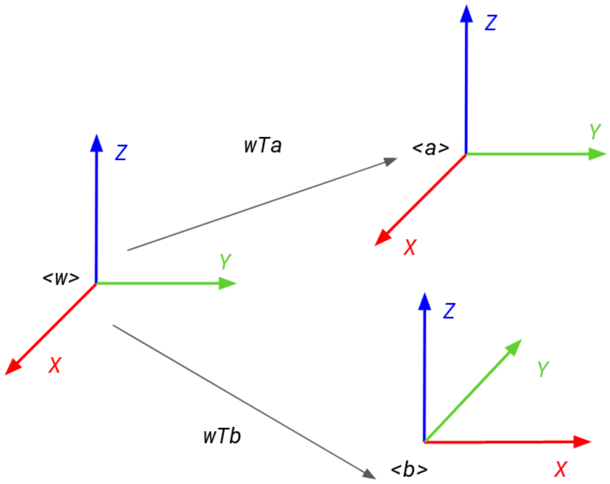


Figure 1: exercise 2 frames

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 the first exercise, I wrote a function called *ComputeAngleAxis()* which implements the Rodrigues Formula. It takes in input the geometric unit vector *v* and the rotation angle θ and it returns the ***orientation matrix***(I called this matrix aRb).

The Rodrigues formula is:

$$aRb = I_{3x3} + K \cdot sin(\theta) + (1 - cos(\theta)) \cdot K^2$$

$$with \quad K = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \quad and \quad v = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}$$

I am going to show my result, in black there are the principal frame, in blue there are the final frame, and in some graphics there are a red vector, it is the reference rotation axis.

2.1 Q1.1

The *Figure 1.1* portrays a rotation about the X_1 axis (r $= [1, 0, 0]$ with a $\theta = 30°$).

$$\theta = 0.5236rad \qquad v = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

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

2.2 Q1.2

The *Figure 1.2* portrays a rotation about the X_2 axis (r $= [0; 1; 0]$ with a $\theta = 45°$).

$$\theta = 0.7854rad \qquad v = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

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

2.3 Q1.3

The *Figure 1.3* portrays a rotation about the X_3 axis (r $= [0; 0; 1]$ with a $\theta = 90°$).

$$\theta = 1.5708rad \qquad v = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

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

2.4 Q1.4

The *Figure 1.4* portrays a rotation about the axis r $= [0.408; 0.816; -0.408]$ with a $\theta = 14.0317°$.

$$\theta = 0.2449rad \qquad v = \begin{bmatrix} 0.4080 & 0.8160 & -0.4080 \end{bmatrix}$$

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

2.5 Q1.5

The *Figure 1.5* portrays a rotation about the axis r $= [0; 1; 0]$ with a $\theta = 90°$. r and θ are calculated from $\rho = [0; \pi/2; 0]$.

$$\theta = \sqrt{\rho_1^2 + \rho_2^2 + \rho_3^2} \qquad and \qquad v = \frac{\rho}{\theta}$$

$$\theta = 1.5708rad \qquad\qquad v = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

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

2.6 Q1.6

The *Figure 1.6* portrays a rotation about the axis r $= [0.68599; -0.5145; -0.5145]$ with a $\theta = 33.4089°$.

$$\theta = 0.5831rad \qquad\qquad v = \begin{bmatrix} 0.6860 & -0.5145 & -0.5145 \end{bmatrix}$$

$$aRb = \begin{bmatrix} 0.9125 & 0.2250 & -0.3416 \\ -0.3416 & 0.8785 & -0.3340 \\ 0.2250 & 0.4215 & 0.8785 \end{bmatrix}$$

2.7 Q1.7

The *Figure 1.7* portrays a rotation about the axis r $= [-0.5747; -0.76626; 0.28735]$ with a $\theta = 78.3023°$.

$$\theta = 1.3666rad \qquad\qquad v = \begin{bmatrix} -0.5747 & -0.7663 & 0.2873 \end{bmatrix}$$

$$aRb = \begin{bmatrix} 0.4661 & 0.0697 & -0.8820 \\ 0.6325 & 0.6709 & 0.3872 \\ 0.6187 & -0.7383 & 0.2686 \end{bmatrix}$$

3 Exercise 2

For the second point, I made a function calls *ComputeInverseAngleAxis()*. It does the inverse process of the *ComputeAngleAxis()*, that means the new function takes a matrix and converts it in an angle and a vector.

3.1 Q2.1

I calculated two orientation matrices the wRa and the wRb, because I need to find the aRb matrix (Figure 4). For do that i have to calculate the inverse matrix of wRa and I know $wRa = wRa^{-1}$ because, in this example, $wRa = I_{3x3}$ matrix (identical matrix). Then I have also $v = [0, 0, 1]$ and $\theta = \pi/2$.

$$wRa = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I_{3x3} \qquad and \qquad wRb = ComputeAngleAxis(\theta, v)$$

$$aRb = wRa^{-1} \cdot wRb = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3.2 Q2.2

At that point; I used the function *ComputeInversalAngleAxis()*. This function is composed by three cycles if; the first one controls if the size of aRb is $3x3$ matrix, the second one checks if it is orthogonal and the last one looks the determinate of the matrix can be 1 (one). Then i use two loop for and a if to check all row and all column inside the matrix aRb if there are eigenvalues equal to 1 (one) and at the end of this function, I create the vector v equal to their eigenvectors.

$$NB: \quad orthogonal\ matrix: \quad aRb^{\perp} = |aRb \cdot aRb^{-1}|$$

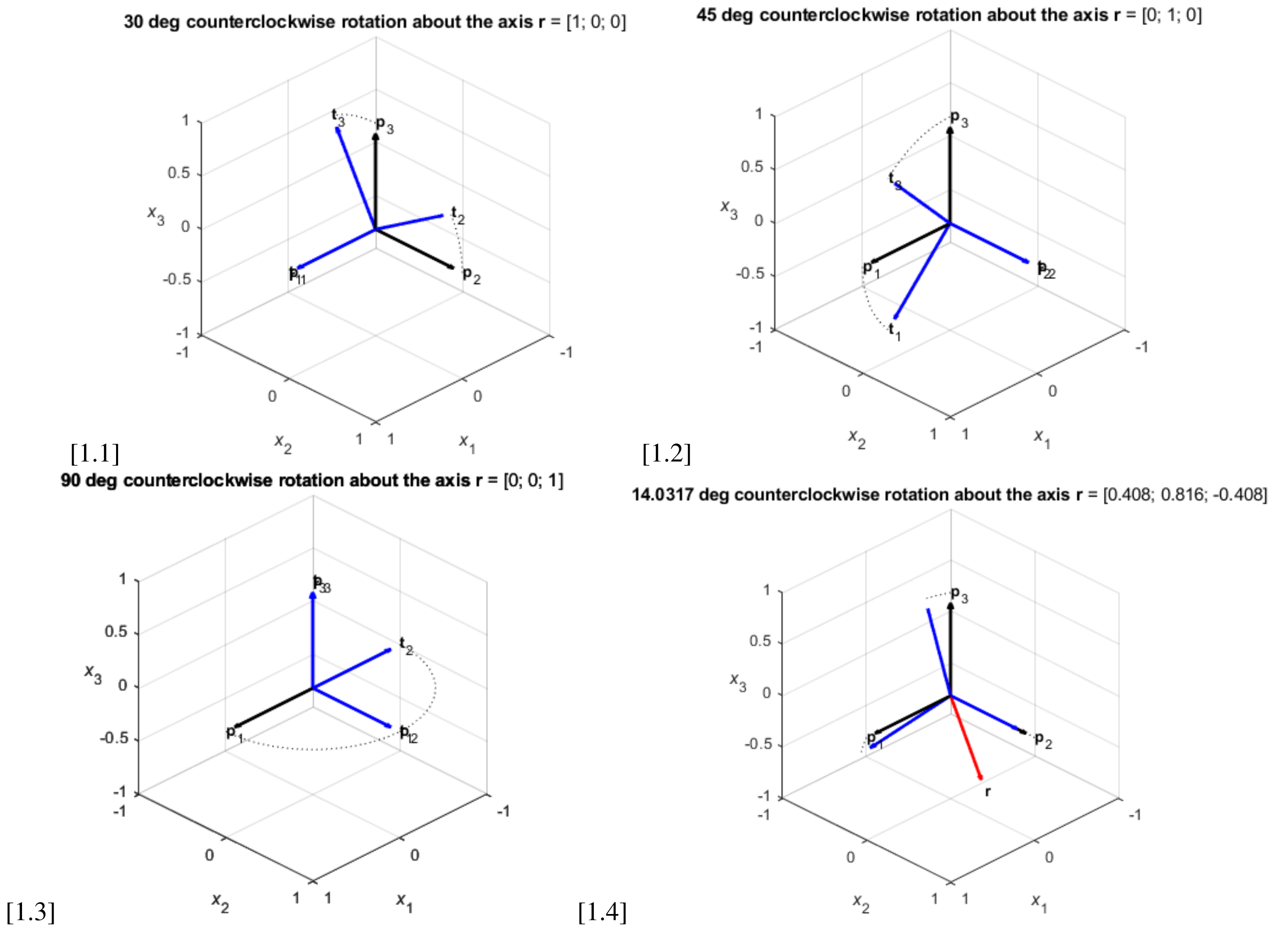
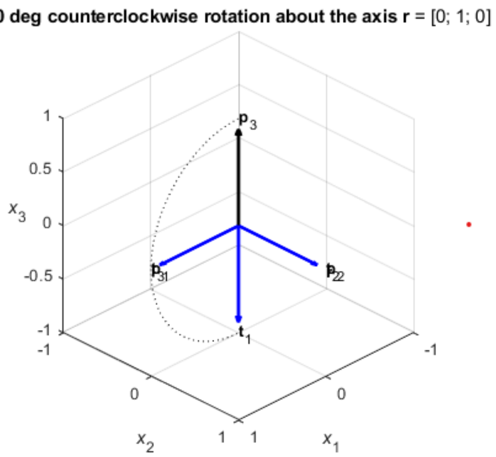
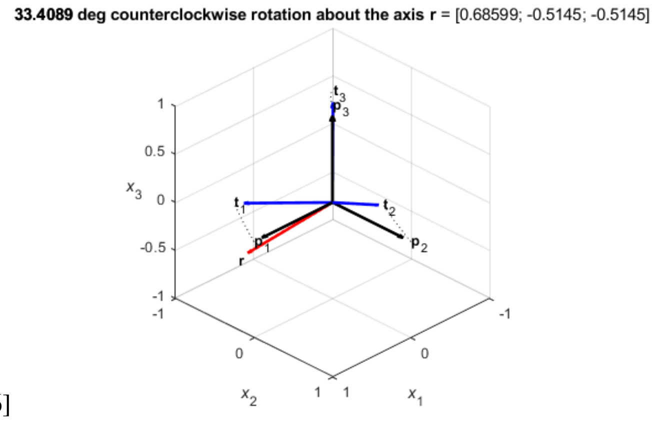


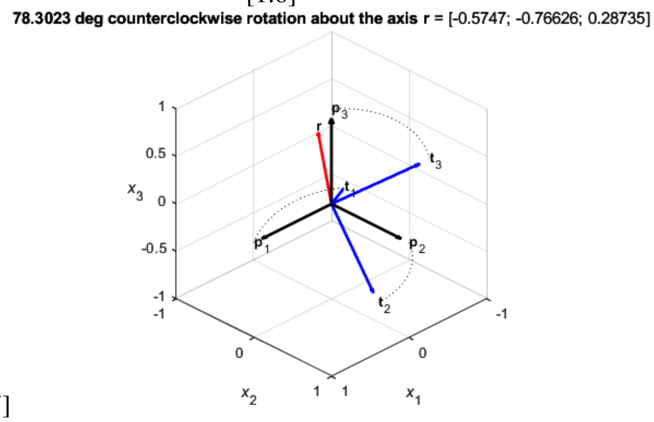
Figure 2:
(1.1) 30.0 deg counterclockwise rotation about the axis r = [1;0;0]
(1.2) 45.0 deg counterclockwise rotation about the axis r = [0;1;0]
(1.3) 90.0 deg counterclockwise rotation about the axis r = [0;0;1]
(1.4) 14.0 deg counterclockwise rotation about the axis r = [0.408;0.816;-0.408]



[1.5]



[1.6]



[1.7]

Figure 3:
(1.5) 90.0 deg counterclockwise rotation about the axis r = [0;1;0]
(1.6) 33.4089 deg counterclockwise rotation about the axis r = [0.68599;-0.5145;-0.5145]
(1.7) 78.3023 deg counterclockwise rotation about the axis r = [-0.5747;-0.76626;0.28735]

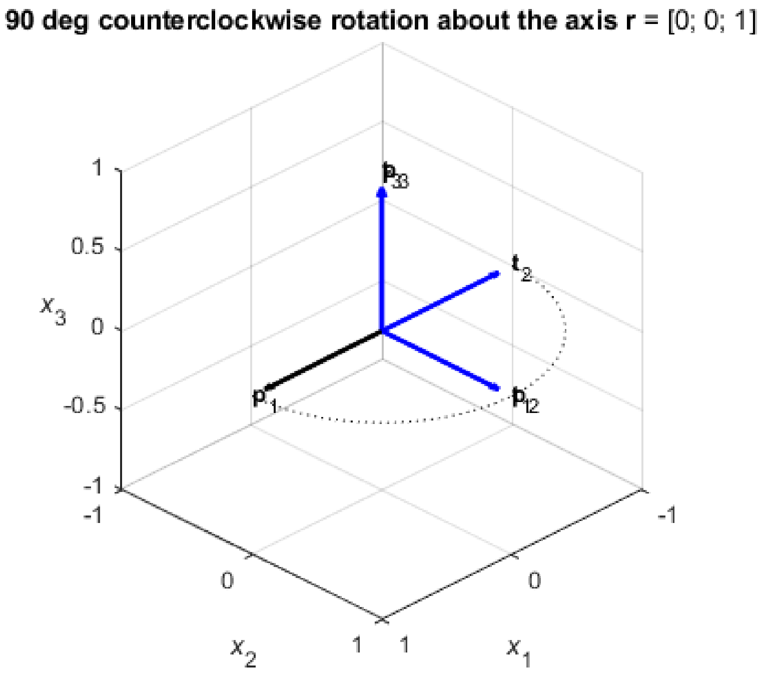


Figure 4: The rotation matrix between frame $< a >$ and frame $< b >$.

$$v = [eigenvectors] = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

$$\theta = \arccos(\frac{tr(aRb) - 1}{2}) = 1.5708 rad$$
$$with \quad tr(aRb) = aRb_{11} + aRb_{22} + aRb_{33}$$

3.3 Q2.3

Now, I have to calculate the rotation matrix between frame $< c >$ and frame $< b >$. The problem gives me the matrix wRc and with the function *ComputeInverseAngleAxis()* I found that rotation matrix (Figure 5).

$$wRc = \begin{bmatrix} 0.835959 & -0.283542 & -0.469869 \\ 0.271321 & 0.957764 & -0.0952472 \\ 0.47703 & -0.0478627 & 0.877583 \end{bmatrix}$$

$$cRb = wRc^{-1} \cdot wRb \qquad -> \qquad \theta = 1.3529 rad \qquad v = \begin{bmatrix} 0.2651 & 0.2931 & 0.9186 \end{bmatrix}$$

4 Exercise 3

From two generic frames $< w >$ and $< b >$, I defined *elementary orientation matrices* for the frame $< b >$ respect to frame $< w >$. I generated the *ComputeElementaryOrientationMatrix()*, it takes a vector, an angle and creates elementary rotation matrix.

4.1 Q3.1

Rotation matrix from $< w >$ to frame $< b >$ by rotating around z-axes (Figure 3.z):

$$v_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \qquad \theta = 30 deg$$

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

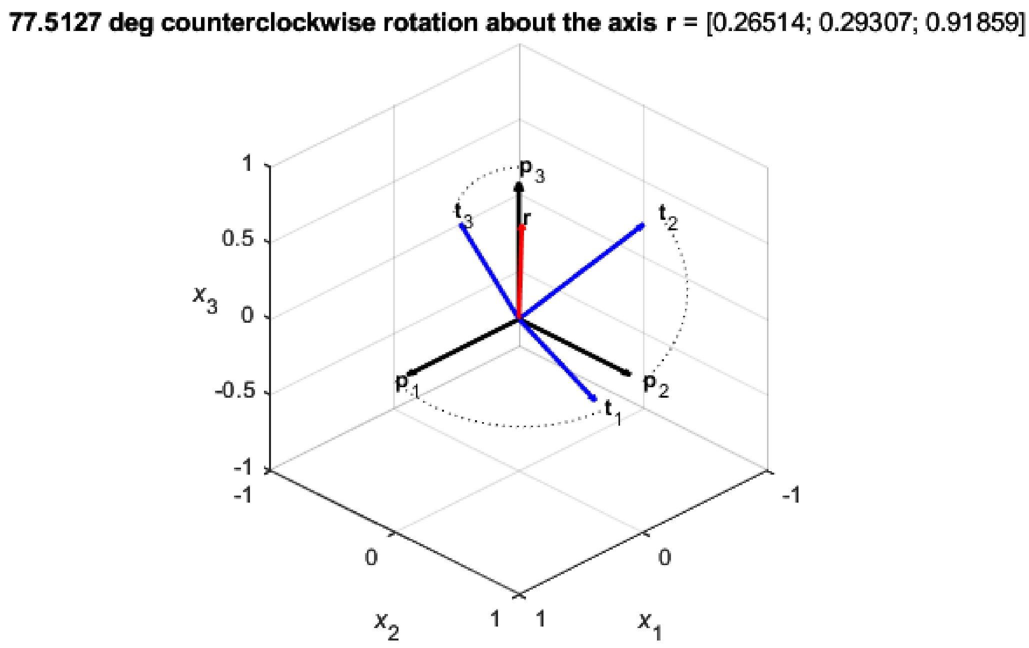


Figure 5: The rotation matrix between frame $< c >$ and frame $< b >$.

Rotation matrix from $< w >$ to frame $< b >$ by rotating around y-axes (Figure 3.y):

$$v_y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \qquad \theta = 45deg$$

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

Rotation matrix from $< w >$ to frame $< b >$ by rotating around x-axes(Figure3.x):

$$v_x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad \theta = 15deg$$

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

4.2 Q3.2

Now, I calculated the θ and the vector v with the formula *ComputeInverseAngleAxis()*.

- For wRb_z: $\theta = 0.5236$ and $v_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$
- For wRb_x: $\theta = 0.7854$ and $v_y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$
- For wRb_x: $\theta = 0.2618$ and $v_x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$

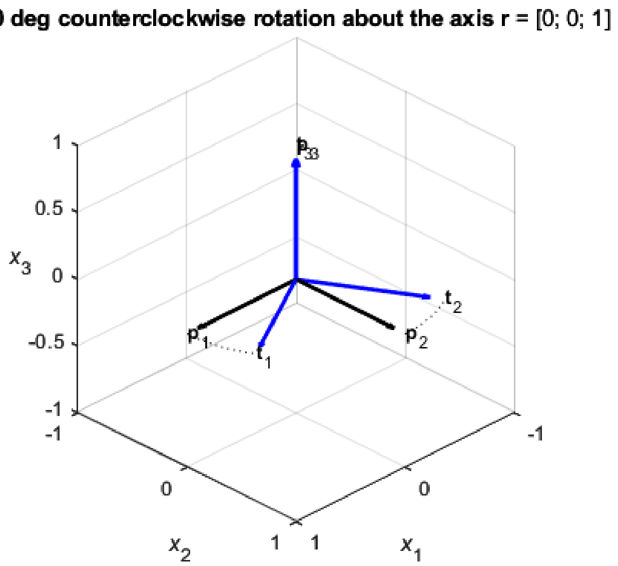
4.3 Q3.3

I calculated the rotation matrix corresponding to the z-y-x representation (Figure 7) in this way.

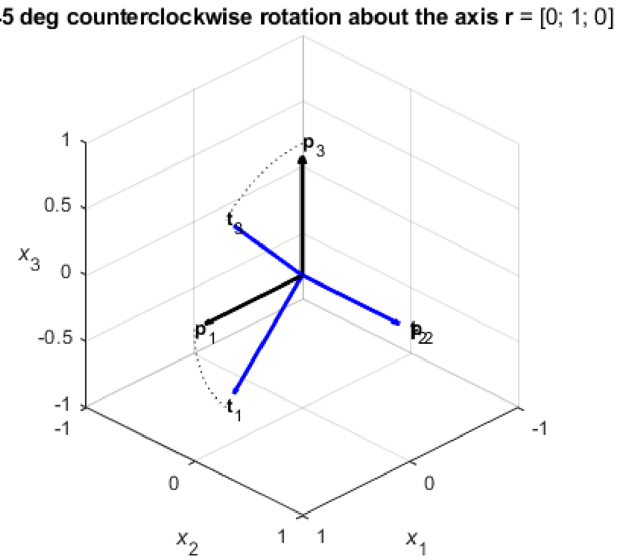
$$R_{zyx} = wRb_z \cdot wRb_y \cdot wRb_x$$

And I used the *ComputeInverseAngleAxis()* formula to calculate θ and v from R_{zyx}.

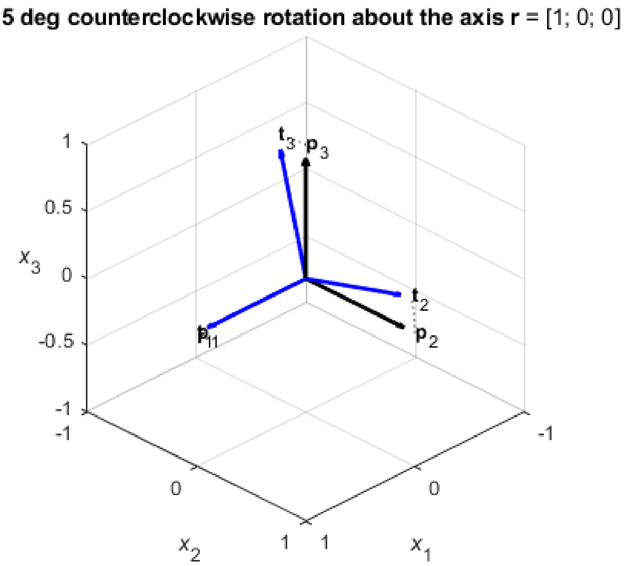
$$\theta = 0.9126rad \qquad v = \begin{bmatrix} 0.0415 & 0.9026 & 0.4285 \end{bmatrix}$$



[3.z]



[3.y]



[3.x]

Figure 6: (3.z) wRb_Z; (3.y) wRb_y; (3.x) wRb_x.

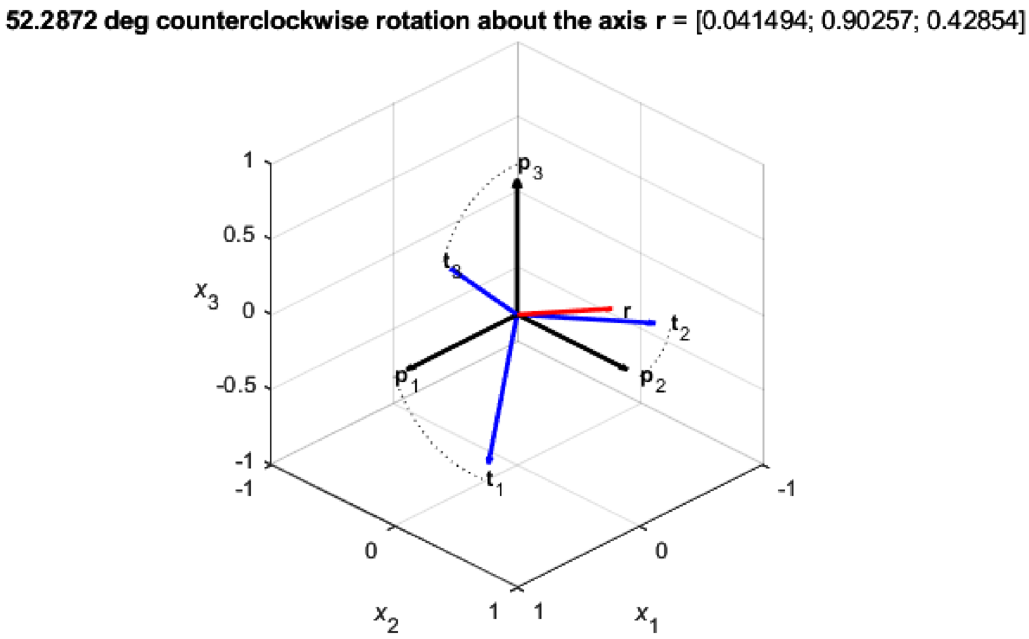


Figure 7: Rotation matrix corresponding to the z-y-x representation

4.4 Q3.4

I calculated the rotation matrix corresponding to the z-x-z representation (Figure 8) in this way.

$$R_{zxz} = wRb_z \cdot wRb_x \cdot wRb_z$$

And I used the *ComputeInverseAngleAxis()* formula to calculate θ and v from R_{zxz}.

$$\theta = 1.0765rad \qquad v = \begin{bmatrix} 0.2546 & 0 & 0.9670 \end{bmatrix}$$

5 Exercise 4

In the last exercise, I generated the *quatToRot()* function. It converts a orientation quaternions, (Q_0, Q_1, Q_2, Q_3), into a rotation matrix. A spatial rotation around a fixed point of θ radians about a unit axis (X, Y, Z) that denotes the Euler axis is given by the quaternion $(C, X\,S, Y\,S, Z\,S)$, where $C = \cos(\theta/2)$ and $S = \sin(\theta/2)$.
The rotation matrix is composed by:

FIRST ROW: $\begin{bmatrix} 2 \cdot [(Q_0^2) + (Q_1^2)] - 1 & 2 \cdot [(Q_1\,Q_2) - (Q_0\,Q_3)] & 2 \cdot [(Q_1\,Q_3) + (Q_0\,Q_2)] \end{bmatrix}$
SECOND ROW: $\begin{bmatrix} 2 \cdot [(Q_1\,Q_2) + (Q_0\,Q_3)] & 2 \cdot [(Q_0^2) + Q_2^2)] - 1 & 2 \cdot [(Q_2\,Q_3) - (Q_0\,Q_1)] \end{bmatrix}$
THIRD ROW: $\begin{bmatrix} 2 \cdot [(Q_1\,Q_3) - (Q_0\,Q_2)] & 2 \cdot [(Q_2\,Q_3) + (Q_0\,Q_1)] & 2 \cdot [(Q_0^2) + (Q_3^2)] - 1 \end{bmatrix}$

5.1 Q4.1

In this point; I used mine function to generate the rotation matrix (Figure 9) and in the point $Q_4.2$ I used the Matlab function to compare if the results were the same. At the end the two matrices are identical ((1) $\equiv$ (2)).
My rotation matrix is:

$$\begin{bmatrix} 0.7071 & -0.0000 & 0.7071 \\ 0.3536 & 0.8660 & -0.3536 \\ -0.6124 & 0.5000 & 0.6124 \end{bmatrix} \tag{1}$$

5.2 Q4.2

The rotation matrix with the Matlab function is:

$$\begin{bmatrix} 0.7071 & -0.0000 & 0.7071 \\ 0.3536 & 0.8660 & -0.3536 \\ -0.6124 & 0.5000 & 0.6124 \end{bmatrix} \tag{2}$$

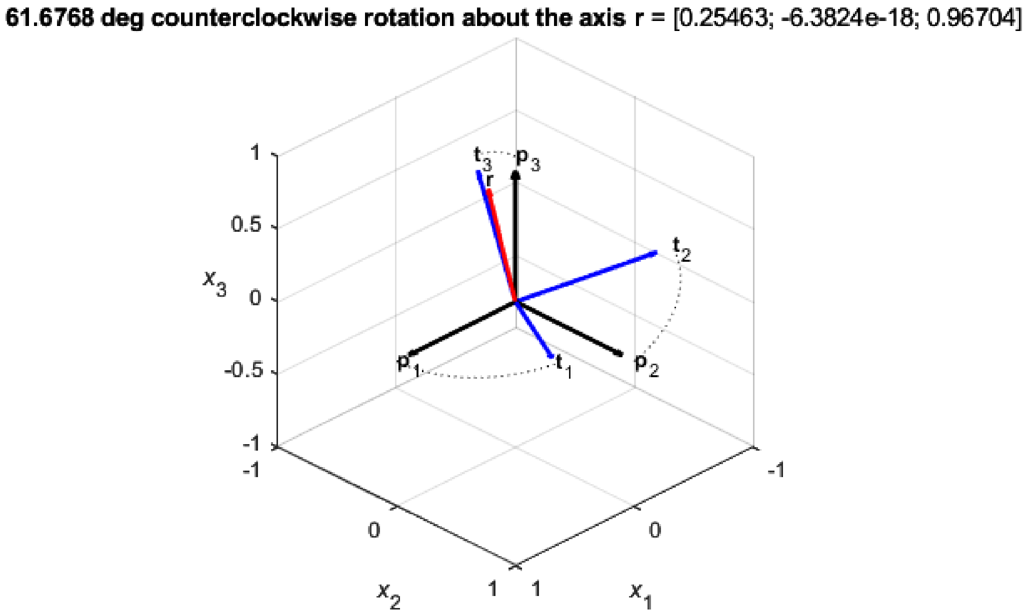


Figure 8: Rotation matrix corresponding to the z-x-z representation

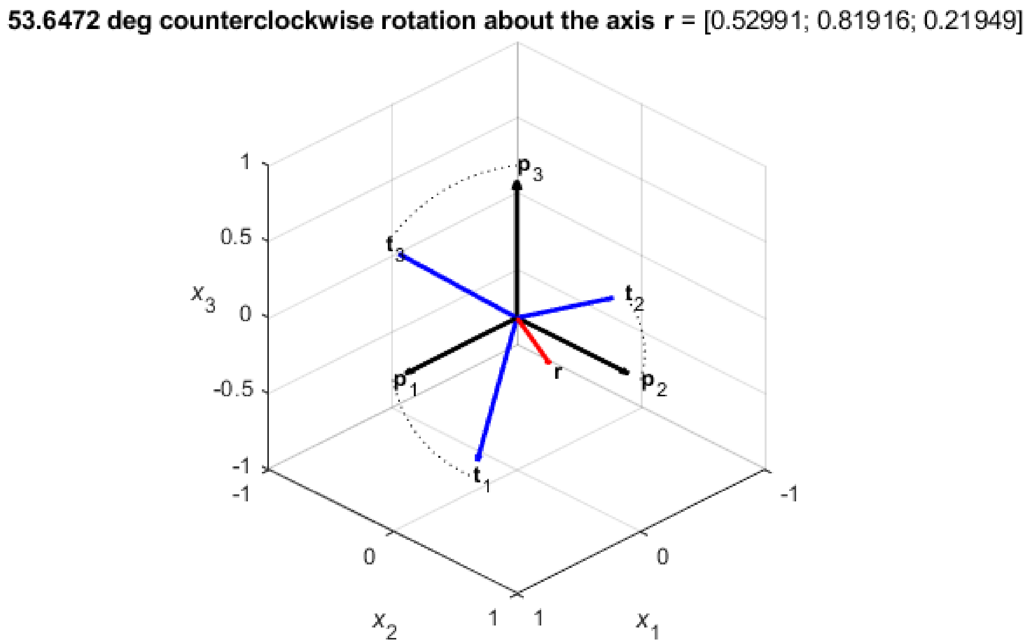


Figure 9: Rotation matrix associated with the given quaternion