DIBRIS

DEPARTMENT OF INFORMATICS,
BIOENGINEERING, ROBOTICS AND SYSTEM ENGINEERING

MODELLING AND CONTROL OF MANIPULATORS

First Assignment

Equivalent representations of orientation matrices

Author:

Surname Name

Student ID:

s7903986

Professors:

Enrico Simetti
Giorgio Cannata

Tutors:

Luca Tarasi
Simone Borelli

November 11, 2025

Contents

Mathematical expression	Definition	MATLAB expression
$< w >$	World Coordinate Frame	w
${}^{a}_{b}R$	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 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** $R = \begin{pmatrix} 0.5 & -\sqrt{3}/2 & 0 \\ \sqrt{3}/2 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- **Q2.4** $R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- **Q2.5** $R = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

- **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.

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

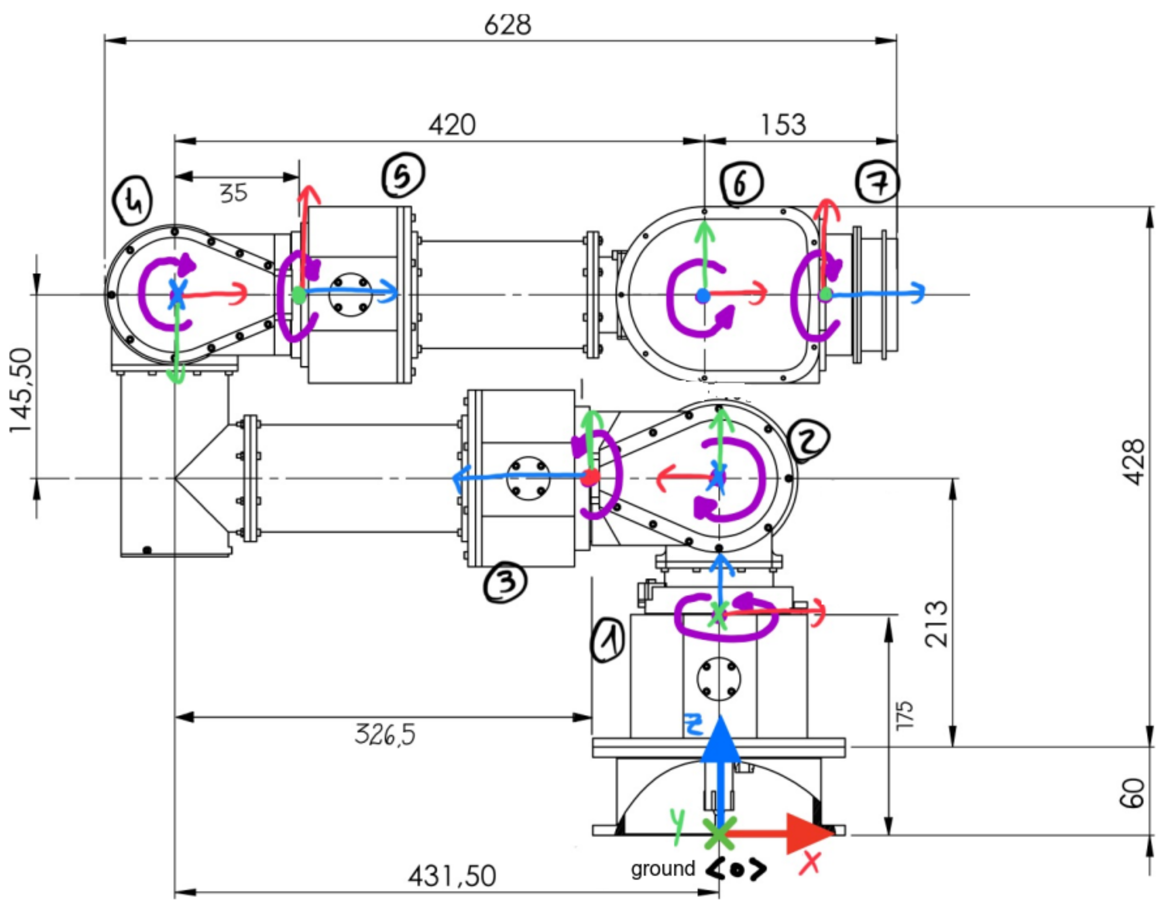

Figure 1: Exercise 6 frames. The unit of measurement is millimeters.

- **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** ${}^{0}_{1}T$
- **Q6.2** ${}^{1}_{2}T$
- **Q6.3** ${}^{2}_{3}T$
- **Q6.4** ${}^{3}_{4}T$
- **Q6.5** ${}^{4}_{5}T$
- **Q6.6** ${}^{5}_{6}T$
- **Q6.7** ${}^{6}_{6}T$
- **Q6.8** ${}^{7}_{6}T$

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

2 Exercise 1

2.1 Q1.1

These sets of exercises demonstrate the computation of a rotation matrix derived from its axial vector and angle of rotation. With the assistance of the Rodrigues formula, this can be computed. The *Rodrigues Formula*[1] is shown below:

$$\mathbf{R} = \mathbf{I} + \sin(\theta)[\underline{h}\mathbf{x}] + (1 - \cos(\theta))[\underline{h}\mathbf{x}]^2 \tag{1}$$

The components of the equation are as follows:

- $\mathbf{I}$ is the identity matrix.
- $\sin(\theta)[\underline{h}\mathbf{x}]$ is the multiplication of $\sin(\theta)$ with the anti-symmetrical matrix. $[\underline{h}\mathbf{x}]$ is the anti-symmetric matrix as it has a power of 1.
- $(1 - \cos(\theta))[\underline{h}\mathbf{x}]^2$ is the multiplication of $(1 - \cos(\theta))$ with the symmetrical matrix. $[\underline{h}\mathbf{x}]^2$ is the symmetric matrix due to its even exponent.

The MATLAB code initially defines the identity matrix. Subsequently, it transforms the vector h into a matrix, utilizing the formula defined in the equation [2].

$$[\underline{h}\mathbf{x}] = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \tag{2}$$

To calculate the third component of the Rodrigues equation [1], it is essential to determine the symmetric matrix. This is indicated by the subsequent formula:

$$[\underline{h}\mathbf{x}]^{2(i+1)} = (-1)^i \underline{h}\underline{h}^T - \mathbf{I} \tag{3}$$

This code will suffice for the first two questions. However, for question 1.4, an additional step is required.

For question 1.4, the input is a rotation vector that needs to be computed into its respective axial vector and rotation angle.

To calculate θ, it is necessary to first obtain the norm of the input. h is subsequently determined by dividing the vector (ρ) by θ. To check the validity of the axial vector computed, it must adhere to the angle-axis parameter, where the norm of h is equal to 1. The subsequent steps for question 1.4 adhere to the same methodology as the previous questions.

2.2 Q1.2

The input represents a frame rotated 90 degrees about the x-axis. The rotation axis is clearly seen by the rotation matrix produced, wherein R(1,1) equals one, while the corresponding columns and rows contain zeros. The rotation matrix can also be computed by the angle about the x-axis into the respective matrix shown in figure 4.

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \tag{4}$$

2.3 Q1.3

The input represents a frame rotated 60 degrees about the z-axis. The rotation axis is clearly seen in the rotation matrix produced, wherein R(3,3) equals one, while the corresponding columns and rows contain zeros. The rotation matrix can also be computed by the angle about the z-axis into the respective matrix shown in figure 4.

$$\mathbf{R} = \begin{bmatrix} 0.5 & -0.866 & 0 \\ 0.866 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

2.4 Q1.4

Initially, this rotation matrix is more challenging to comprehend. The indication of no entry, where there exists only a single value in the column and row, suggests a rotation along many axes. The theta is 90 degrees, and the axial vector is [-0.667, -0.333, 0.6667]. This value indicates that the frame rotates 90 degrees around the specified axial vector.

This interpretation further explains the Euler theorem, which declares that a frame, in the Euclidean space, can be described by a singular rotation of an angle around a single fixed axis (the axial vector).

$$\mathbf{R} = \begin{bmatrix} 0.44 & -0.44 & -0.78 \\ 0.89 & 0.11 & 0.44 \\ -0.11 & -0.89 & 0.44 \end{bmatrix} \tag{6}$$

3 Exercise 2

The objective of converting a rotation matrix to its equivalent angle-axis representation ($\underline{h}$, θ) is to identify the axis of rotation ($\underline{h}$) and the angle of rotation about that specified axis. This improves our comprehension of a rotation matrix. The calculation of θ is shown in Figure 3.

Nonetheless, there are special requirements for $\underline{h}$ that are affected when theta is either 0 or π.
If theta equals zero, it signifies that a frame has not undergone rotation; hence, the axial vector is arbitrary.

If $\theta = \pi$, then according to the Rodrigues formula [1], the output is "$2\underline{hh^T} - I$." This outcome is a symmetric matrix. From this, $\underline{h}$ can be computed. To calculate $\underline{h}$, it is necessary to initially select a non-zero diagonal variable and designate its sign (positive or negative), as seen in figure 7a. The allocation of the sign guarantees the consistency of the remaining computed h values, as seen in Figure 2.

Utilizing the equation in Figure 7b, it is possible to calculate the remaining $\underline{h}$ values. The initial part of the equation multiplies the sign of the first selected $\underline{h}$ by the sign of the off-diagonal element to obtain the sign of the second $\underline{h}$ value. The rest of the equation calculates the value of $\underline{h}_j$.

Prior to this computation, it is essential to first establish whether the input matrix is a rotation matrix. For a matrix to be classified as a rotation matrix, it must belong to the special orthogonal group ($So(n)$). The special orthogonal group states that: $R\epsilon R^{nxn}$ such that $R^T R = I$ and $det(R) = 1$. Which will satisfy the properties of:

- Closure: If R_1, R_2 belong to $So(n)$. Then their multiplication also belongs to $So(n)$.
- Identity: where there exists an element I in $So(n)$, then $IR = R$ also belongs to $So(n)$.
- Associativity: where any three elements R_1, R_2, R_3 in $So(n)$, then $R_1(R_2R_3) = (R_1R_2)R_3$
- Inverse: for every R in $So(n)$ there exists R^- in $So(n)$ such that $RR^- = I$
- For any vector (x) that exist in R^3 and where R is an element in $So(n)$. Then the norm of Rx will equal the norm of x. This indicates that the rotation matrix did not affect the length of the vector.

The properties of the special orthogonal group ensure that no important information about a rotation matrix or vector is affected by another rotation matrix. It also assists in calculations and assumptions. To ensure that the rotation matrix was identified, the testing parameters were given a tolerance of 0.0001.

$$\pm\underline{h}_i = \pm\sqrt{(R_{ii}+1)/2} \tag{7a}$$

$$\underline{h}_j = \operatorname{sign}(\underline{h}_i)\operatorname{sign}(R_{ij})\sqrt{(R_{jj}+1)/2} \tag{7b}$$

Figure 2: Formula for calculating the values of the axial vector $\underline{h}$

$$\theta = \arccos((trace(R) - 1)/2) \tag{8}$$

Figure 3: Formula for calculating θ

3.1 Q2.2

$$[\underline{h}] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \theta = 90 \tag{9}$$

Indicates that a frame is rotated 90 degrees about the x-axis, represented by the axial vector shown.

3.2 Q2.3

$$[\underline{h}] = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \theta = 60 \tag{10}$$

Indicates that a frame is rotated 60 degrees about the z-axis, represented by the axial vector displayed.

3.3 Q2.4

$$[\underline{h}] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \theta = 0 \tag{11}$$

This indicates that there is no rotation of the frame. Therefore, an arbitrary axis is defined, where its norm equals 1.

3.4 Q2.5

$$[\underline{h}] = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \theta = \pi \tag{12}$$

This shows that the frame is rotated by π about the z-axis.

3.5 Q2.6

The given rotation matrix was not an element of the special orthogonal group; therefore, it was not a rotation matrix, and no axial vector nor theta could be computed.

4 Exercise 3

Euler angles to a rotation matrix assists in describing the motion of a frame about each axis (z,y,x). The sequence of rotation is yaw-pitch-roll. This rotation sequence corresponds to an intrinsic rotation. An intrinsic rotation is when a rotation about an axis is complete, and the next rotation is computed on that rotated frame. Each rotation about its respective axis corresponds to the respective rotation matrices shown in Figure 4. The complete rotation matrix is the multiplication of each rotation about an axis. Every rotation matrix has a unique set of rotation sequences.

$$\mathbf{R}_z = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad \mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{13}$$

Figure 4: Rotation matrices about the Z(yaw), Y(pitch), and X(roll) axes

4.1 Q3.2

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \tag{14}$$

The rotation matrix represents the rotation about the x-axis by 90 degrees. This is computed by substituting the rotation angle (phi) into the R_x rotation matrix seen in Figure 5.

4.2 Q3.3

$$\mathbf{R} = \begin{bmatrix} 0.5 & -0.866 & 0 \\ 0.866 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

The rotation matrix represents a rotation about the z-axis by 60 degrees. This is computed by substituting the rotation angle (psi) into the R_z rotation matrix seen in Figure 5.

4.3 Q3.4

$$\mathbf{R} = \begin{bmatrix} 0 & -0.259 & 0.966 \\ 0 & 0.966 & 0.259 \\ -1 & 0 & 0 \end{bmatrix} \tag{16}$$

The rotation matrix represents the rotation first about the z-axis by 60 degrees, then about the y-axis by 90 degrees, and finally about the x-axis by 45 degrees. The final rotation matrix is computed by substituting each rotation angle into the respected rotation matrices shown in Figure 5 and performing the multiplication of the rotation sequence (zyx).

4.4 Q3.5

$$\mathbf{R} = \begin{bmatrix} 0 & -0.259 & 0.966 \\ 0 & 0.966 & 0.259 \\ -1 & 0 & 0 \end{bmatrix} \tag{17}$$

The rotation matrix represents the rotation first about the y-axis by 90 degrees, and then about the x-axis by -15 degrees. This yields the same rotation matrix seen in 3.4. This demonstrates how unique sequences can lead to the same final rotation matrices. This is essentially useful when a link of a robot can not rotate a certain amount about a specific axis, but a desired final frame is still required. Therefore, a different rotation sequence can still achieve the desired frame.

5 Exercise 4

The yaw-pitch-roll values can be calculated from a given rotation matrix. This calculation is performed using the generalised rotation matrix shown in Figure 5. This matrix corresponds to the yaw-pitch-roll rotation sequence. The matrix facilitates the calculation of yaw-pitch-roll angles. Theta is computed using the formula seen in Figure 6. Φ and ψ are computed using the subsequent formula shown in Figure 7, provided that cos(theta) is non-zero.

If $\cos(theta) = 0$ (θ being +/- 90 degrees), this results in a singularity. The formulas shown in Figure 8 and Figure 9 demonstrate the calculation for ψ and ϕ when $\theta = +/-90$. The formulas were calculated by substituting $\theta = +/-90$ into the rotation matrix shown in Figure 5.

A singularity indicates that a rotation around the roll axis produces an effect on the frame similar to a rotation around the yaw axis. Consequently, the rotation about either axis (roll or yaw) becomes indistinguishable and results in a loss of a degree of freedom. The Figures (9 and 8) indicate that there are infinitely many sets of -yaw and -roll angles that will produce the same rotation matrix when $\theta = +/-90$. The singularity will also impact the robot's control algorithm. [[2],[1]] Suppose that a joint of a robot experiences a singularity during rotation. It will then result in a rapid alteration in movement, which will potentially compromise the control algorithm and the surroundings (particularly in an uncontrolled environment).

Before executing this computation, it is necessary to check whether the input matrix qualifies as a rotation matrix, for reasons discussed previously.

$$\mathbf{R}_{zyx} = \begin{bmatrix} \cos\psi & -\sin\psi\cos\theta + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\phi\sin\theta\sin\psi & -\cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{18}$$

Figure 5: Rotation matrices for the rotation sequence of ZYX

$$\theta = \operatorname{atan2}(-R_{31}, \sqrt{R_{11}^2 + R_{21}^2}) \quad \epsilon[-\pi/2, \pi/2] \tag{19}$$

Figure 6: Formula to calculate theta

$$\psi = \operatorname{atan2}(R_{21}, R_{11}) \quad \phi = \operatorname{atan2}(R_{32}, R_{33}) \tag{20}$$

Figure 7: Calculation of ψ and ϕ **when** $cos\theta$ does not equal zero

$$\psi - \phi = \operatorname{atan2}(-R_{12}, R_{22}) \quad or \quad \psi - \phi = \operatorname{atan2}(R_{23}, R_{13}) \tag{21}$$

Figure 8: Calculation of ψ and ϕ when $\theta = +90$

$$\psi + \phi = \operatorname{atan2}(-R_{12}, R_{22}) \quad or \quad \psi + \phi = \operatorname{atan2}(-R_{23}, -R_{13}) \tag{22}$$

Figure 9: Calculation of ψ and ϕ when $\theta = -90$

5.1 Q4.2

$$\psi = 0 \quad \theta = 0 \quad \phi = 90 \tag{23}$$

The frame is only rotated about the roll axis by 90 degrees. This is a uniquely defined rotation sequence.

5.2 Q4.3

$$\psi = 60 \quad \theta = 0 \quad \phi = 0 \tag{24}$$

The frame is only rotated about the yaw axis by 60 degrees. This is a uniquely defined rotation sequence.

5.3 Q4.4

$$\psi = 90 \quad \theta = 60 \quad \phi = 45 \tag{25}$$

This represents the respective rotation angles of the rotation sequence zxy. Where the frame is first rotated about the yaw axis by 90 degrees, then the pitch axis by 60 degrees, and finally about the roll axis by 45 degrees. This is a uniquely defined rotation sequence.

6 Exercise 5

When a rotation matrix is multiplied by a vector, the result is a new vector. However, when a rotation matrix is multiplied by a vector, which is the axial vector, the results will be identical to the multiplied vector. This observation is confirmed by substituting the axial vector into the Rodrigues formula (1). This indicates that the rotation matrix has no effect on the vector. This result evidently makes $\underline{h}$ an eigenvector of R with a corresponding eigenvalue of +1.

Two methods were computed to verify the angle-axis representation. The first method was discussed in Exercise 2. The second method calculates $\underline{h}$ as the eigenvector of the rotation matrix associated with an eigenvalue of +1. This is accomplished using the formula in Figure 10.

The outcome is a matrix, and we have to compute the values of $\underline{h}$ using the row echelon form. The next step is to normalise the vector, as the axial vector is a unit vector. Theta is calculated using the formula in Figure 3.

Prior to executing the second method, it is necessary to check whether the input matrix qualifies as a rotation matrix, for reasons discussed previously.

$$\underline{0} = (\lambda I_n - R)\underline{h} \tag{26}$$

Figure 10: Eigen vector formula

6.1 Q5.1

$$[\underline{h}] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \theta = 90 \tag{27}$$

Method one revealed the results shown above. The second method revealed the same results, confirming that $\underline{h}$ is an eigenvector of R with an eigenvalue of +1

6.2 Q5.2

$$[\underline{h}] = \begin{bmatrix} -0.67 \\ -0.33 \\ 0.67 \end{bmatrix} \quad \theta = 90 \tag{28}$$

The calculation of the axial vector and theta from the approach outlined in Exercise 2 generated identical results, with the axial vector being an eigenvector of R corresponding to an eigenvalue of +1. This result is identical to Exercise 1-Q1.4, where the rotation vector produced the above axial vector and theta.

7 Exercise 6

A transformation matrix is constructed as shown in Figure 11. Where ${}^a_bR$ denotes the rotation of frame b relative to frame a, and ${}^aO_b$ denotes the translation of frame b relative to frame a. The final row facilitates the multiplication of transformation matrices.

The rotation about each axis was computed using the rotation matrices shown in Figure 5. The described rotation sequence for each question represented the multiplication order of the rotation matrices. It is important to note that the right-hand rule should be used to compute the positive rotation direction. Each translation from one frame to another is based on the coordinate frame of the 'relative to' frame.

The inverse of a transformation matrix is identical to the transformation matrix when the frames are reversed. The inverse of a transformation matrix can be calculated using the formula shown in Figure 12. This equation enables the rotation of frame A relative to frame B and the positioning of frame A in relation to frame B.

The transformation equation allows for the understanding of how each frame, which represents the joints' coordinate system, is related with respect to one another. This assists in the understanding of the pose of the robot within the workspace. Transformation matrices also assist with the calculation of inverse kinematics.

$${}^a_bT = \begin{pmatrix} & {}^a_bR & & {}^aO_b \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{29}$$

Figure 11: Inverse Transformation matrix

$$ {}^{a}_{b}T^{-1} = {}^{b}_{a}T \quad {}^{a}_{b}T^{-1} = \begin{pmatrix} & {}^{a}_{b}R^{T} & & -{}^{a}_{b}R^{T\,a}O_{b} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{30} $$

Figure 12: Inverse Transformation matrix

7.1 Q6.1

$$ {}^{0}_{1}T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.1750 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

The transformation matrix from frame 0 to frame 1. This has an identity rotation matrix, as there was no rotation from frame 0 to frame 1. Only a translation along the z-axis of frame 0.

7.2 Q6.2

$$ {}^{1}_{2}T = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0.38 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

The transformation matrix from frame 1 to frame 2. The frame was rotated about the x-axis by 90 degrees and then about the y-axis by -180 degrees. The frame was then translated along the z-axis of frame 1 by 0.38m.

7.3 Q6.3

$$ {}^{2}_{3}T = \begin{pmatrix} 0 & 0 & 1 & 1.05 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

The transformation matrix from frame 2 to frame 3. The frame was rotated about the y-axis by 90 degrees. The frame was then translated along the x-axis of frame 2 by 1.05m.

7.4 Q6.4

$$ {}^{3}_{4}T = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1.455 \\ -1 & 0 & 0 & 3.265 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

The transformation matrix from frame 3 to frame 4. The frame was rotated about the y-axis by 90 degrees and then about the x-axis by 180 degrees. The frame was then translated along the z-axis of frame 3 by 3.265m and along the y-axis of frame 3 by 1.455m.

7.5 Q6.5

$$ {}^{4}_{5}T = \begin{pmatrix} 0 & 0 & 1 & 0.35 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

The transformation matrix from frame 4 to frame 5. The frame was rotated about the z-axis by -90 degrees and about the x-axis by -90 degrees. The frame was then translated along the x-axis of frame 4 by 0.35m.

7.6 Q6.6

$$ {}^{5}_{6}T = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 3.85 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

The transformation matrix from frame 5 to frame 6. The frame was rotated about the z-axis by -90 degrees and about the y-axis by -90 degrees. The frame was then translated along the z-axis of frame 5 by 3.85m.

7.7 Q6.7

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

This is an identity transformation matrix, as there is no rotation or translation of the frame itself.

7.8 Q6.8

$$^{7}_{6}T = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1.53 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The transformation matrix from frame 7 to frame 6 is computed using the formula depicted in Figure 12. Therefore, it was essential to calculate the transformation matrix from frame 6 to frame 7 to assist in the computation of the inverse transformation matrix. The transformation matrix from frame 6 to frame 7 included a rotation matrix that initially rotated the frame 90 degrees about the y-axis, followed by a 90-degree rotation about the z-axis. It translated 1.53 m along the x-axis of frame 6.

As illustrated in Figure 1, a translation along the negative z-axis is anticipated. It also corresponds to the magnitude of the translation applied during the computation of $^{6}_{7}T$.