



UNIVERSITÀ DEGLI STUDI DI GENOVA

DIBRIS

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY,  
BIOENGINEERING, ROBOTICS AND SYSTEM ENGINEERING

MODELING AND CONTROL OF MANIPULATORS

---

## First Assignment

Equivalent representations of orientation matrices

---

*Author:*

Balia Gian Marco  
Polese Carolina  
Salterini Filippo

*Professors:*

Giovanni Indiveri  
Enrico Simetti  
Giorgio Cannata

*Student ID:*

s4398275  
s4862903  
s4516129

*Tutors:*

Andrea Tiranti  
Georgii Kurshakov  
Luca Tarasi

October 31, 2024

## Contents

<b>1 Assignment description</b>	<b>3</b>
1.1 Exercise 1 - Angle-Axis to Rotation Matrix . . . . .	3
1.2 Exercise 2 - Rotation Matrix to Angle-Axis . . . . .	3
1.3 Exercise 3 - Euler Angles to Rotation Matrix . . . . .	4
1.4 Exercise 4 - Rotation Matrix to Euler Angles . . . . .	4
1.5 Exercise 5 - Frame tree . . . . .	4
<b>2 Exercise 1</b>	<b>5</b>
2.1 Q1.1 . . . . .	5
2.2 Q1.2 . . . . .	5
2.3 Q1.3 . . . . .	5
2.4 Q1.4 . . . . .	6
<b>3 Exercise 2</b>	<b>6</b>
3.1 Q2.1 . . . . .	6
3.2 Q2.2 . . . . .	6
3.3 Q2.3 . . . . .	6
3.4 Q2.3 . . . . .	7
3.5 Q2.4 . . . . .	7
<b>4 Exercise 3</b>	<b>7</b>
4.1 Q3.1 . . . . .	7
4.2 Q3.2 . . . . .	7
4.3 Q3.3 . . . . .	7
4.4 Q3.4 . . . . .	7
4.5 Q3.5 . . . . .	7
<b>5 Exercise 4</b>	<b>8</b>
5.1 Q4.1 . . . . .	8
5.2 Q4.2 . . . . .	8
5.3 Q4.3 . . . . .	8
5.4 Q4.4 . . . . .	8
<b>6 Exercise 5</b>	<b>8</b>
6.1 Q5.1 . . . . .	8
<b>7 Appendix</b>	<b>9</b>
7.1 Appendix A . . . . .	9
7.2 Appendix B . . . . .	9

Mathematical expression	Definition	MATLAB expression
$\langle w \rangle$	World Coordinate Frame	w
${}^a_b R$	Rotation matrix of frame $\langle b \rangle$ with respect to frame $\langle a \rangle$	aRb
${}^a_b T$	Transformation matrix of frame $\langle b \rangle$ with respect to frame $\langle a \rangle$	aTb

Table 1: Nomenclature Table

# 1 Assignment description

The first assignment of Modelling and Control of Manipulators focuses on the geometric fundamentals and algorithmic tools underlying any robotics application. The concepts of transformation matrix, orientation matrix and the equivalent representations of orientation matrices (Equivalent angle-axis representation and Euler Angles) will be reviewed.

The first assignment is **mandatory** and consists of 5 different exercises. You are asked to:

- Download the .zip file called MCM-LAB1 from the Aulaweb page of this course.
- Implement the code to solve the exercises on MATLAB by filling the predefined files called "main.m", "AngleAxisToRot.m", "RotToAngleAxis.m", "YPRToRot.m" and "RotToYPR.m".
- Write a report motivating the answers for each exercise, following the predefined format on this document.

## 1.1 Exercise 1 - Angle-Axis to Rotation Matrix

A particularly interesting minimal representation of 3D rotation matrices is the so-called angle-axis representation, where a rotation is represented by the axis of rotation  $\mathbf{h}$  and the angle  $\theta$ . 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

$$\text{function } R = \text{AngleAxisToRot}(h, \theta)$$

Then test it for the following cases and briefly comment the results obtained:

- **Q1.2**  $\mathbf{h} = [1, 0, 0]^T$  and  $\theta = 90^\circ$
- **Q1.3**  $\mathbf{h} = [0, 0, 1]^T$  and  $\theta = \pi/3$
- **Q1.4**  $\rho = [-\pi/3, -\pi/6, \pi/3]$ ;

**Note that**  $\rho = \mathbf{h}\theta$ .

## 1.2 Exercise 2 - Rotation Matrix to Angle-Axis

Given a rotation matrix  $R$ , the problem of finding the corresponding angle-axis representation  $(\mathbf{h}, \theta)$  is called the Inverse Equivalent Angle-Axis Problem.

**Q2.1** Given a rotation matrix  $R$ , implement on MATLAB the Equivalent Angle-Axis equations **WITHOUT** using built-in matlab functions. The function signature will be

$$\text{function } [h, \theta] = \text{RotToAngleAxis}(R)$$

You **MUST** check that the input is a valid rotation matrix. Test it for the following cases and briefly comment the results obtained:

- **Q2.2**  $R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$
- **Q2.3**  $R = \begin{pmatrix} 0.5 & -\sqrt{3}/2 & 0 \\ \sqrt{3}/2 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- **Q2.4**  $R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- **Q2.5**  $R = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

### 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  $(\psi, \theta, \phi)$ , compute the equivalent rotation matrix representation **WITHOUT** using built-in matlab functions. The function signature will be

$$\text{function } R = \text{YPRToRot}(\text{psi}, \text{theta}, \text{phi})$$

Then test it for the following cases and briefly comment the results obtained:

- **Q3.2**  $\psi = \theta = 0, \phi = \pi/2$
- **Q3.3**  $\phi = \theta = 0, \psi = 60^\circ$
- **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  $(\psi, \theta, \phi)$ , 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

$$\text{function } [\text{psi}, \text{theta}, \text{phi}] = \text{rotToYPR}(R)$$

You **MUST** check that the input is a valid rotation matrix. Test it for the following cases and briefly comment the results obtained:

- **Q4.2**  $R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$
- **Q4.3**  $R = \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- **Q4.4**  $R = \begin{pmatrix} 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0.5 & \frac{\sqrt{2}\sqrt{3}}{4} & \frac{\sqrt{2}\sqrt{3}}{4} \\ -\frac{\sqrt{3}}{2} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} \end{pmatrix}$

### 1.5 Exercise 5 - Frame tree

Figure 1 shows the frame tree for the 7 joints of the Franka robot. With reference to the figure, use the geometric definition of the transformation matrix to compute by hand the following matrices.

- **Q5.1**  ${}^0_1T$
- **Q5.2**  ${}^1_2T$
- **Q5.3**  ${}^2_3T$
- **Q5.4**  ${}^3_4T$
- **Q5.5**  ${}^4_5T$
- **Q5.6**  ${}^5_6T$
- **Q5.7**  ${}^6_7T$
- **Q5.8**  ${}^7_eT$

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

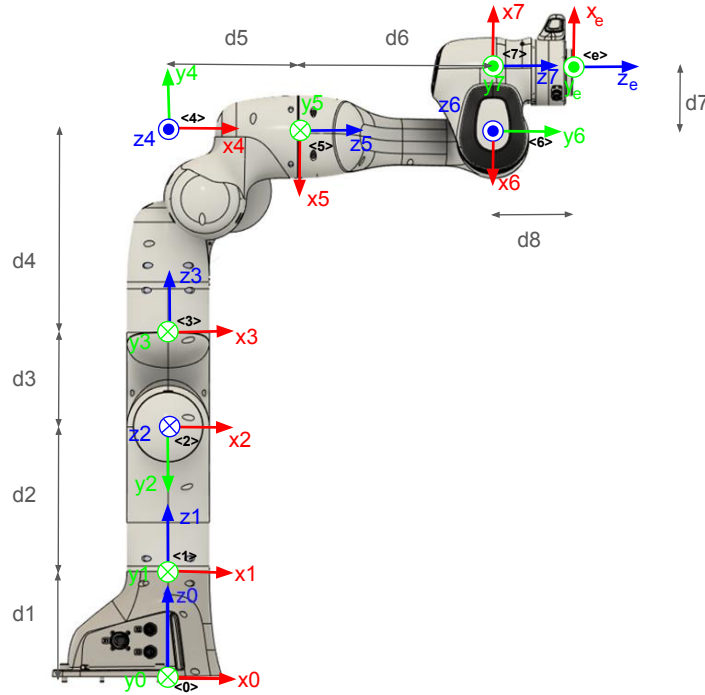


Figure 1: exercise 5 frames

## 2 Exercise 1

### 2.1 Q1.1

The *AngleAxisToRot* is a function that, compute the *rotation matrix*  $\mathbf{R}$  with Rodrigues' formula

$$\mathbf{R} = \mathbf{I}_{3 \times 3} + \sin \theta [\mathbf{h} \times] + (1 - \cos \theta) [\mathbf{h} \times]^2 \quad (1)$$

Given the *axis of rotation*  $\mathbf{h}$ , which checked if it is a unit vector, can be considered three different cases for Equation 1 due to the values of *angle of rotation*  $\theta$ :

- $\theta = 0 \Rightarrow \mathbf{R} = \mathbf{I}_{3 \times 3}$  ;
- $\theta = \pi \Rightarrow \mathbf{R} = \mathbf{I}_{3 \times 3} + 2[\mathbf{h} \times]^2$  where the square of a skew-symmetric matrix is  $[\mathbf{h} \times]^2 = (\mathbf{h}\mathbf{h}^T - \mathbf{I}_{3 \times 3})$  ;
- $\theta \in (0, \pi)$  where, since there are not simplifications of Equation 1 was needed to compute the axis of rotation's skew-symmetric matrix

$$[\mathbf{h} \times] = \begin{bmatrix} 0 & -h_3 & h_2 \\ h_3 & 0 & -h_1 \\ -h_2 & h_1 & 0 \end{bmatrix}$$

### 2.2 Q1.2

In the first case  $\mathbf{h} = [1, 0, 0]^T$  and  $\theta = \pi/2$  represent a 90-degree rotation around the  $x$ -axis. In fact,  $\mathbf{R} = \mathbf{R}_z \mathbf{R}_y \mathbf{R}_x = \mathbf{I}_{3 \times 3} \mathbf{I}_{3 \times 3} \mathbf{R}_x = \mathbf{R}_x$  where

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

### 2.3 Q1.3

In this configuration, which  $\mathbf{h} = [0, 0, 1]^T$  and  $\theta = \pi/3$ , likely the previous case, it is a rotation around the  $z$ -axis but of 60-degree

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

## 2.4 Q1.4

For the last case,  $\rho = [-\pi/3, -\pi/6, \pi/3]$ , before to call the function *AngleAxisToRot* have been computed  $\mathbf{h}$  and  $\theta$  knowing that  $\rho = \theta \mathbf{h}$ , and by definition  $\|\mathbf{h}\| = 1$  this entail  $\|\rho\| = \|\theta \mathbf{h}\| = \theta \|\mathbf{h}\| = \theta$ .

$$\theta = \sqrt{\rho_x^2 + \rho_y^2 + \rho_z^2} = \frac{\pi}{2} \quad \mathbf{h} = \frac{\rho}{\theta} = \left[-\frac{2}{3}, -\frac{1}{3}, \frac{2}{3}\right]^T$$

This means that  $R$  is a 90-degree rotation around the vector  $\mathbf{h}$ .

$$\mathbf{R} = \begin{bmatrix} \frac{4}{9} & -\frac{4}{9} & -\frac{7}{9} \\ \frac{8}{9} & \frac{1}{9} & \frac{4}{9} \\ -\frac{1}{9} & -\frac{8}{9} & \frac{4}{9} \end{bmatrix}$$

## 3 Exercise 2

### 3.1 Q2.1

The *RotToAngleAxis* function takes  $\mathbf{R}$  and computes  $\mathbf{h}$  and  $\theta$ . Before computing  $\mathbf{R}$  were done three preliminary checks:  $\mathbf{R}$  must be a  $3 \times 3$  matrix,  $\det \mathbf{R} = 1$ , and  $\mathbf{R}\mathbf{R}^T = \mathbf{I}_{3 \times 3}$ . Due to prevent floating point, e.g. Subsection 3.3, was important implementing a tolerance for the computation of  $\det \mathbf{R}$  and  $\mathbf{R}\mathbf{R}^T = \mathbf{I}_{3 \times 3}$ , we have chosen a value of  $10^{-10}$ . Considering the

$$\begin{aligned} \text{Tr} R &= \text{Tr}(\mathbf{I}_{3 \times 3} + \sin \theta [\mathbf{h} \times] + (1 - \cos \theta) [\mathbf{h} \times]^2) = 1 + 2 \cos \theta \\ \Rightarrow \theta &= \arccos \left( \frac{\text{Tr} R - 1}{2} \right) \end{aligned}$$

and the *axial vector*

$$\begin{aligned} \mathbf{a} &= \sin \theta \cdot \mathbf{h} = \text{Vex} \left( \frac{\mathbf{R}\mathbf{R}^T}{2} \right) \\ \Rightarrow \mathbf{h} &= \frac{\mathbf{a}}{\sin \theta} \end{aligned}$$

Notice that *theta* could be equal to zero or  $\pi$ :

- $\theta = 0$ : in this case  $\mathbf{h}$  can be arbitrary than we decided to rise an error.
- $\theta = \pi$ : this represent a rotation of 180-degree around one of the axes. In order to find  $\mathbf{h}$ , the script compute one

$$h_i = \pm \sqrt{\frac{r_{ii} + 1}{2}} \quad i \in \mathbb{N} \setminus \{0\}, i \leq 3$$

Determinate the signs of other components  $h_j$  ( $j \in \mathbb{N} \setminus \{0\}, j \leq 3, i \neq j$ )

$$h_j = \text{sgn}(h_i) \text{sgn}(r_{ij}) \sqrt{\frac{r_{jj} + 1}{2}}$$

This let us to find the two possible configurations of the vector  $\mathbf{h}$ .

### 3.2 Q2.2

The rotation matrix

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

represents a rotation around  $x$ -axis. In fact, the function returns  $\mathbf{h} = [1, 0, 0]^T$  and  $\theta = \frac{\pi}{2}$ .

### 3.3 Q2.3

In this case the rotation is around the  $z$ -axis, as show in the rotation matrix

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

Since it's clear that  $\arccos(\frac{1}{2}) = \arcsin(\frac{\sqrt{3}}{2}) = \frac{\pi}{3}$ , then  $\mathbf{h} = [0, 0, 1]^T$  and  $\theta = \frac{\pi}{3}$ .

### 3.4 Q2.3

In the third case,  $R = I_{3 \times 3}$ , it means there is no rotation, then  $\mathbf{h}$  is arbitrary.

### 3.5 Q2.4

With the rotation matrix

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

the angle of rotation  $\theta = \pi$ , this mean that there are two possible configurations of the vector  $\mathbf{h} \rightarrow \mathbf{h}_+ = [0, 0, 1]^T$ ,  $\mathbf{h}_- = [0, 0, -1]^T$ .

## 4 Exercise 3

### 4.1 Q3.1

The *YPRToRot* function computes  $\mathbf{R}$  using the *Yaw-Pitch-Roll* representation

$$\begin{aligned} \mathbf{R}_{YPR} &= \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi) = \\ &= \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \end{aligned} \quad (2)$$

After the *rotation matrix* was checked as in Subsection 3.1.

### 4.2 Q3.2

We have given to the function the inputs  $\psi = 0, \theta = 0, \phi = \frac{\pi}{2}$ , it means that  $\mathbf{R}_z(\psi) = \mathbf{R}_y(\theta) = \mathbf{I}_{3 \times 3}$  then there will be no rotation around  $Z$  and  $Y$  axes. The only rotational contribution is around  $x$ -axis.

$$\mathbf{R}_{YPR} = \mathbf{R}_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

### 4.3 Q3.3

The angles' composition  $\psi = \frac{\pi}{3}, \theta = 0, \phi = 0$ , as discussed above has only the rotational contribution around  $z$ -axis.

$$\mathbf{R}_{YPR} = \mathbf{R}_z(\psi) = \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

### 4.4 Q3.4

The third case with those three input angles  $\psi = \frac{\pi}{3}, \theta = \frac{\pi}{2}, \phi = \frac{\pi}{4}$  is a rotation around all the three axes. The result is the following

$$\mathbf{R}_{YPR} = \begin{pmatrix} 0 & \frac{\sqrt{6}}{4} - \frac{\sqrt{2}}{4} & \frac{\sqrt{6}}{4} + \frac{\sqrt{2}}{4} \\ 0 & \frac{\sqrt{6}}{4} + \frac{\sqrt{2}}{4} & \frac{\sqrt{6}}{4} - \frac{\sqrt{2}}{4} \\ -1 & 0 & 0 \end{pmatrix}$$

### 4.5 Q3.5

The parameters are  $\psi = 0, \theta = \frac{\pi}{2}, \phi = -\frac{\pi}{12}$  means that there is no rotation around  $z$ -axis. In fact, the rotation matrix has the contributes of rotation around  $y$  and  $x$  axes.

$$\mathbf{R}_{YPR} = \mathbf{R}_y(\theta)\mathbf{R}_x(\phi) = \begin{pmatrix} 0 & -\frac{\sqrt{6}-\sqrt{2}}{4} & \frac{\sqrt{6}+\sqrt{2}}{4} \\ 0 & \frac{\sqrt{6}+\sqrt{2}}{4} & \frac{\sqrt{6}-\sqrt{2}}{4} \\ -1 & 0 & 0 \end{pmatrix}$$



## 5 Exercise 4

### 5.1 Q4.1

The *RotToYPR* function compute a possible configuration of the angles  $\psi$ ,  $\theta$ , and  $\phi$  from a *Yaw-Pitch-Roll matrix*. After doing the same checks, reported in Subsection 3.1, to verify if the given matrix is a rotation matrix, the script compute angles as

$$\begin{aligned}\theta &= \arctan2(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \\ \psi &= \arctan2(r_{21}, r_{11}) \\ \phi &= \arctan2(r_{32}, r_{33})\end{aligned}$$

checking if the configuration is not singular ( $\cos \theta \neq 0$ ). Notice that  $r_{ij}$  ( $i, j \in \mathbb{N} \setminus \{0\}, i \leq 3, j \leq 3$ )

### 5.2 Q4.2

The matrix  $\mathbf{R}$  passed to the function and the respective output angles are shown below

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \Rightarrow \psi = 0, \theta = 0, \phi = \frac{\pi}{2}$$

It is evident that the matrix  $\mathbf{R}$  corresponds to an  $\mathbf{R}_x(\phi)$  matrix, as confirmed by the fact that only the angle  $\phi$ , characteristic of rotation along the  $x$ -axis, is different from zero.

### 5.3 Q4.3

In this case the input and the respective output angles are

$$\mathbf{R} = \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \Rightarrow \psi = \frac{\pi}{3}, \theta = 0, \phi = 0$$

They represent a rotation around the  $z$ -axis.

### 5.4 Q4.4

As represented in

$$\mathbf{R} = \begin{pmatrix} 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{1}{2} & \frac{\sqrt{2}\sqrt{3}}{4} & \frac{\sqrt{2}\sqrt{3}}{4} \\ -\frac{\sqrt{3}}{2} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} \end{pmatrix} \Rightarrow \psi = \frac{\pi}{2}, \theta = \frac{\pi}{3}, \phi = \frac{\pi}{4}$$

The rotation matrix  $\mathbf{R}$  is a composition of three rotation, one for each axis.

## 6 Exercise 5

### 6.1 Q5.1

$$\begin{aligned}{}^0_1T &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^1_2T &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^2_3T &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^3_4T &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ {}^4_5T &= \begin{pmatrix} 0 & 1 & 0 & d_5 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^5_6T &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^6_7T &= \begin{pmatrix} 1 & 0 & 0 & -d_7 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^7_eT &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_8 \\ 0 & 0 & 0 & 1 \end{pmatrix}\end{aligned}$$

## **7 Appendix**

*[Comment] Add here additional material (if needed)*

### **7.1 Appendix A**

### **7.2 Appendix B**