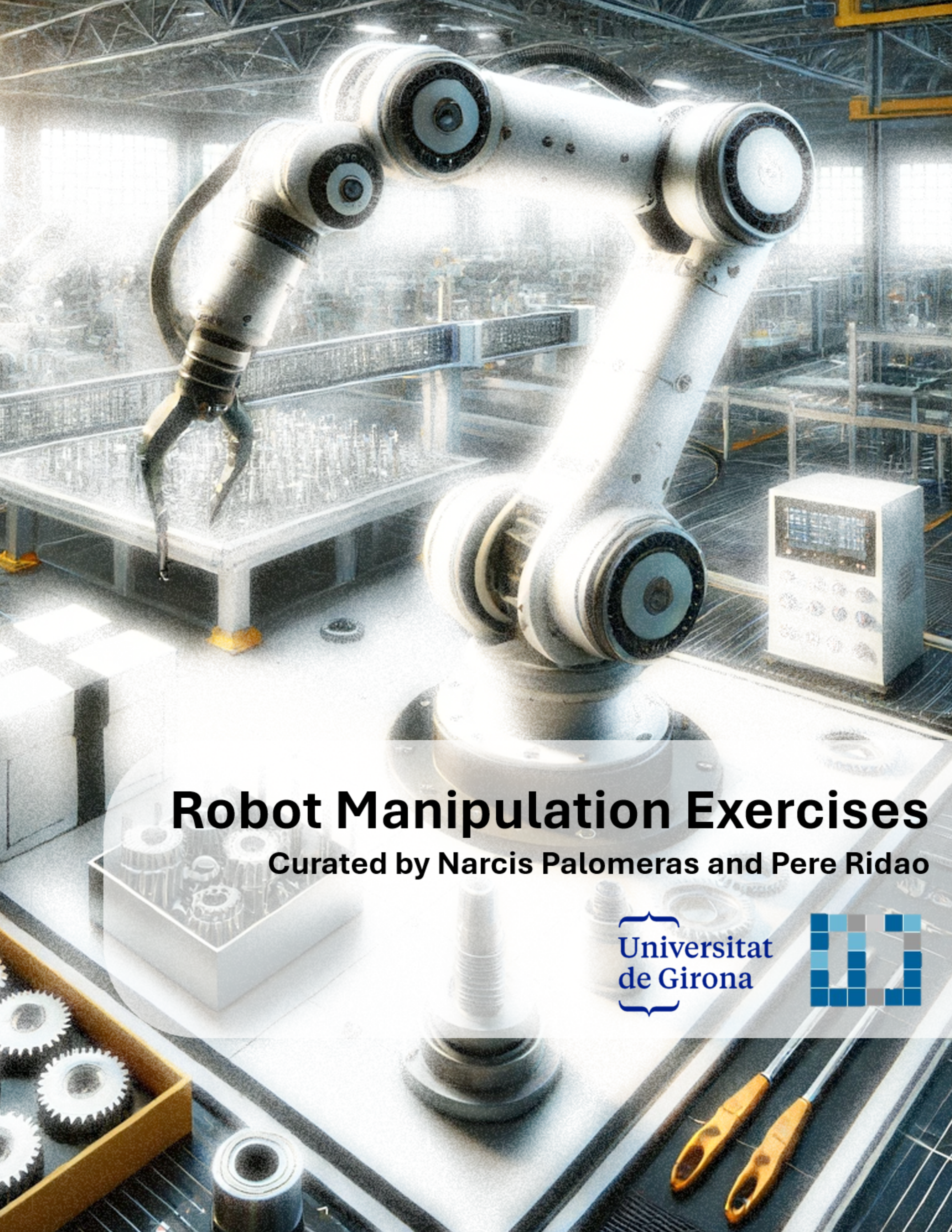


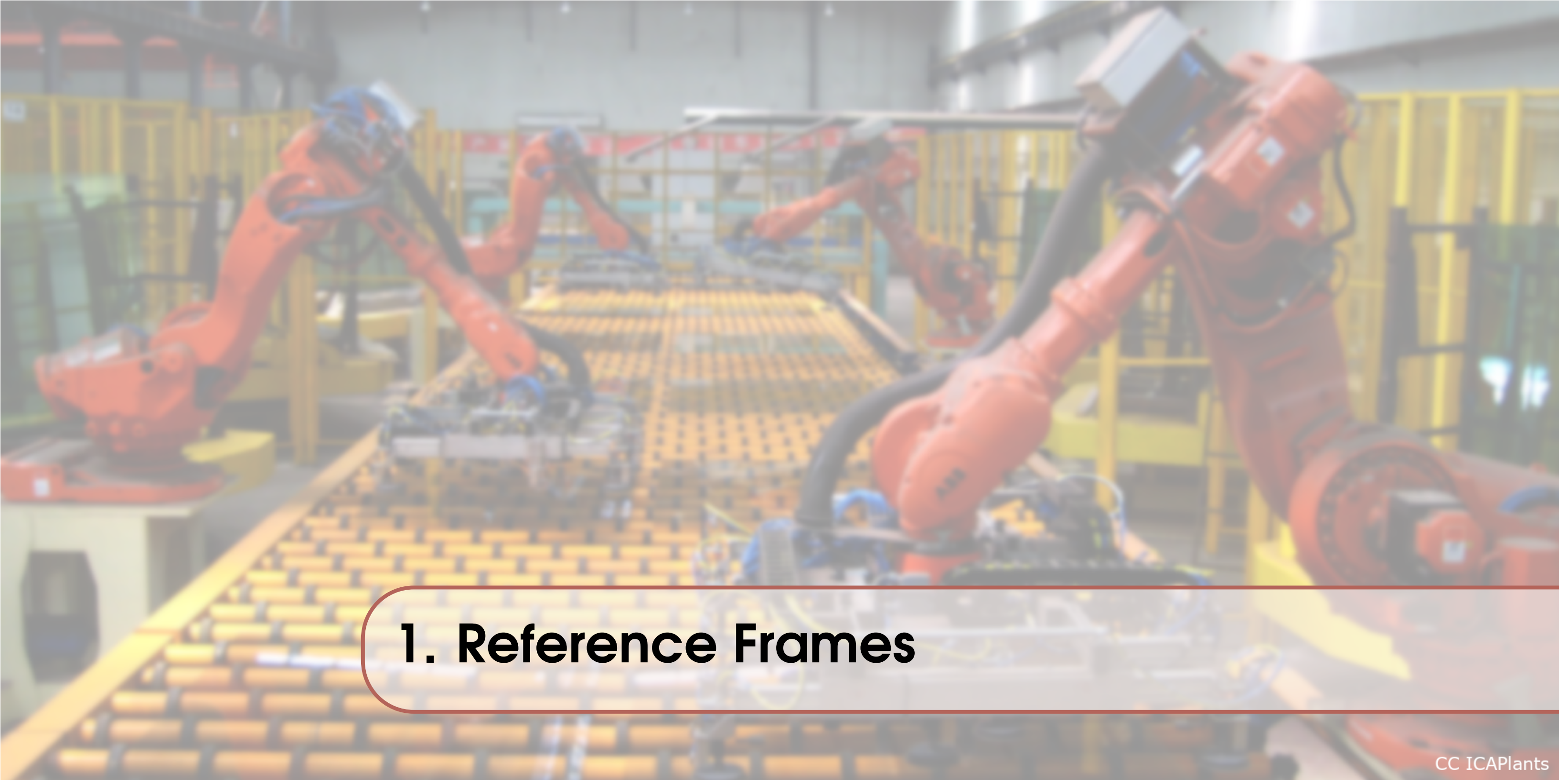
Robot Manipulation Exercises

Curated by Narcis Palomeras and Pere Ridao

Universitat de Girona

Authors: Narcís Palomeras and Pere Ridao
This book contains exercises from professors: Marc Carreras, Josep Forest, Xavier Cufí, Esteve-Amadeu Hernandez, Narcís Palomeras, and Pere Ridao
List of Exercices.
First release, September 2022

Contents



1. Reference Frames

1.1 Reference Frames Exercices

Exercise 1 Robot Matrix

Define the transformation matrix that describes the end effector pose with respect to the base according to Figure 1.1.

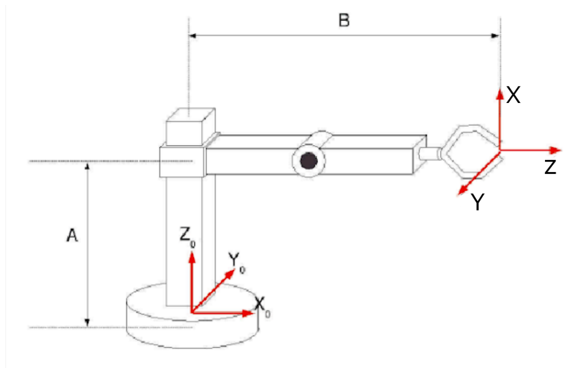


Figure 1.1: **R** and **B** of a 3 DoF robot arm.

Answer of exercise 1

1. The matrix transformation matrix that describes the end effector pose with respect to the baseis:

$$^{R}T_{H} = \begin{bmatrix} 0 & 0 & 1 & B \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & A \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1.1}$$

Exercise 2 Homogeneous Transformation I

Provide the numeric value of the $^{A}T_{B}$ homogeneous transformation matrix corresponding to the **A** and **B** shown in Figure 1.2.

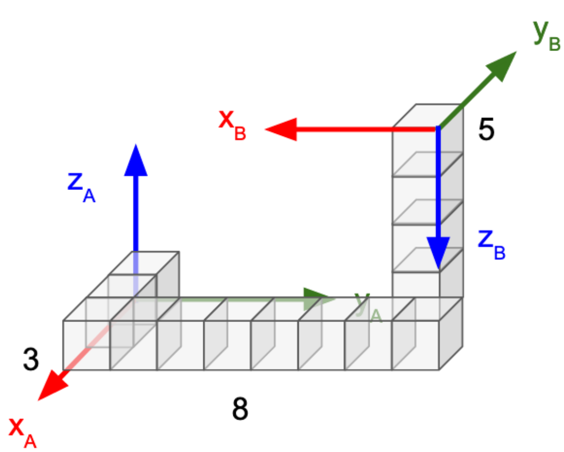


Figure 1.2: **A** and **B** according to ${}^{A}T_{B}$

Answer of exercise 2

We know that:

$$ {}^{A}T_{B} = \begin{bmatrix} {}^{A}x_{B} & {}^{A}y_{B} & {}^{A}z_{B} & {}^{A}p_{B} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.2} $$

where ${}^{A}\boldsymbol{x_B}$, ${}^{A}\boldsymbol{y_B}$ and ${}^{A}\boldsymbol{z_B}$ are the unitary vectors of the **B**, represented in **A**, and ${}^{A}\boldsymbol{p_B}$ is the origin of **B** represented in **A**. According to Figure 1.2 ${}^{A}\boldsymbol{x_B} = [0\ -1\ 0]^T$, ${}^{A}\boldsymbol{y_B} = [-1\ 0\ 0]^T$, ${}^{A}\boldsymbol{z_B} = [0\ 0\ -1]^T$ and ${}^{A}\boldsymbol{p_B} = [3\ 8\ 5]^T$. Therefore, ${}^{\mathbf{A}}\mathbf{T_B}$ is given by:

$$ {}^{A}T_{B} = \begin{bmatrix} 0 & -1 & 0 & 3 \\ -1 & 0 & 0 & 8 \\ 0 & 0 & -1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.3} $$

Exercise 3 Homogeneous Transformation II

Given the homogeneous transformation

$$ {}^{\mathbf{A}}\mathbf{T_B} = \mathbf{Trans}(\mathbf{3},\mathbf{4},\mathbf{5}) \cdot \mathbf{Rot}(\mathbf{x},\mathbf{90}) \cdot \mathbf{Rot}(\mathbf{y},-\mathbf{90}) \cdot \mathbf{Rot}(\mathbf{z},\mathbf{270}) $$

you are requested to:

1. Draw the **A** and the **B**.
2. Provide the numerical value of the matrix ${}^{\mathbf{B}}\mathbf{T_A}$.

Answer of exercise 3

1. Figure 1.3 shows **A** and **B** according to ${}^{A}T_{B}$.

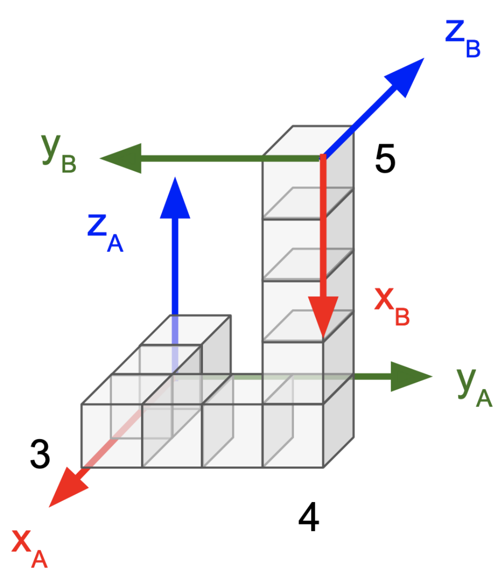


Figure 1.3: **A** and **B** according to ${}^{A}T_{B}$

2. We can take profit of Figure 1.3 to extract $^{\mathbf{B}}\mathbf{T}_{\mathbf{A}}$ by geometrical inspection. We know that:

$$^{B}\boldsymbol{T}_{A} = \begin{bmatrix} ^{B}\boldsymbol{x}_{A} & ^{B}\boldsymbol{y}_{A} & ^{B}\boldsymbol{z}_{A} & ^{B}\boldsymbol{p}_{A} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.4}$$

where $^{B}\boldsymbol{x}_{A}$, $^{B}\boldsymbol{y}_{A}$ and $^{B}\boldsymbol{z}_{A}$ are the unitary vectors of the **B**, represented in **A**, and $^{B}\boldsymbol{p}_{A}$ is the origin of **B** represented in **A**. According to Figure 1.3 $^{B}\boldsymbol{x}_{A} = [0\ 0\ -1]^T$, $^{B}\boldsymbol{y}_{A} = [0\ -1\ 0]^T$, $^{B}\boldsymbol{z}_{A} = [-1\ 0\ 0]^T$ and $^{B}\boldsymbol{p}_{A} = [5\ 4\ 3]^T$. Therefore, $^{\mathbf{B}}\mathbf{T}_{\mathbf{A}}$ is given by:

$$^{B}\boldsymbol{T}_{A} = \begin{bmatrix} 0 & 0 & -1 & 5 \\ 0 & -1 & 0 & 4 \\ -1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.5}$$

Exercise 4 Homogeneous Transformation III

Given a F_0 with axis $< x_0, y_0, z_0 >$ we apply the following sequence of transformations:

- A Rotation of 90° around the axis x_0 obtaining F_1 with axis $< x_1, y_1, z_1 >$.
- A Rotation of 90° around the axis y_1 obtaining F_2 with axis $< x_2, y_2, z_2 >$.
- A 10 m translation along z_0 obtaining F_3 with axis $< x_3, y_3, z_3 >$.

1. Provide the numerical value of the transformation matrix $^0T_3$ which transforms the points represented in F_3 to F_0.
2. Draw F_0, F_1, F_2 and F_3.

Answer of exercise 4

1. The requested $^{\mathbf{0}}\boldsymbol{H}_{\mathbf{2}}$ homogeneous matrix is given by (1.6)

$$\begin{aligned} ^{\mathbf{0}}\boldsymbol{H}_{\mathbf{2}} &= (\boldsymbol{z}_{\mathbf{0}}, \mathbf{10}) \cdot \boldsymbol{Rot}(\boldsymbol{x}_{\mathbf{0}}, 90^\circ) \cdot \boldsymbol{Rot}(\boldsymbol{y}_{\mathbf{1}}, 90^\circ) \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & -0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{1.6}$$

2. F_0, F_1, F_2 and F_3 are drawn in Figure 1.4

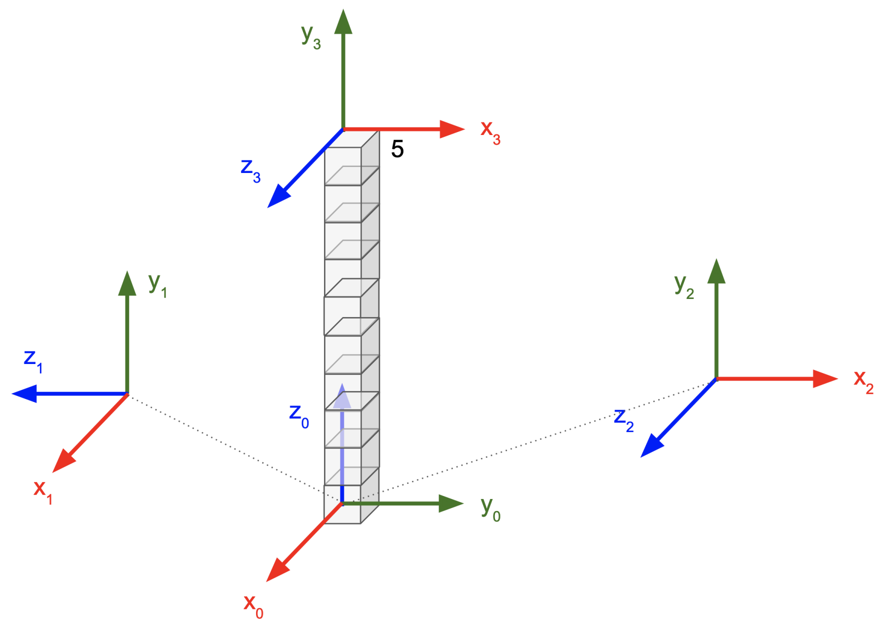


Figure 1.4: F_0, F_1, F_2 and F_3 according to ${}^{0}\boldsymbol{H}_{\boldsymbol{2}}$

Exercise 5 RVM1

Figure 1.5 shows the a sequence of simple transformations that must be applied to the base **R** in order to reach the hand **H** of the RV-M1 robot. The transformations are:

- Translation t between frames $\{R\}$ and $\{a\}$
- Rotation of α between frames $\{a\}$ and $\{b\}$
- Rotation between frames $\{b\}$ and $\{c\}$ (use visual inspection)
- Rotation of θ between frames $\{c\}$ and $\{d\}$
- Rotation of ϕ between frames $\{d\}$ and $\{H\}$

1. Find the symbolic expression of the ${}^{\boldsymbol{R}}\boldsymbol{T}_{\boldsymbol{H}}$ matrix, depending on t, α, θ, and ϕ, obtained from the product of the individual transformations.

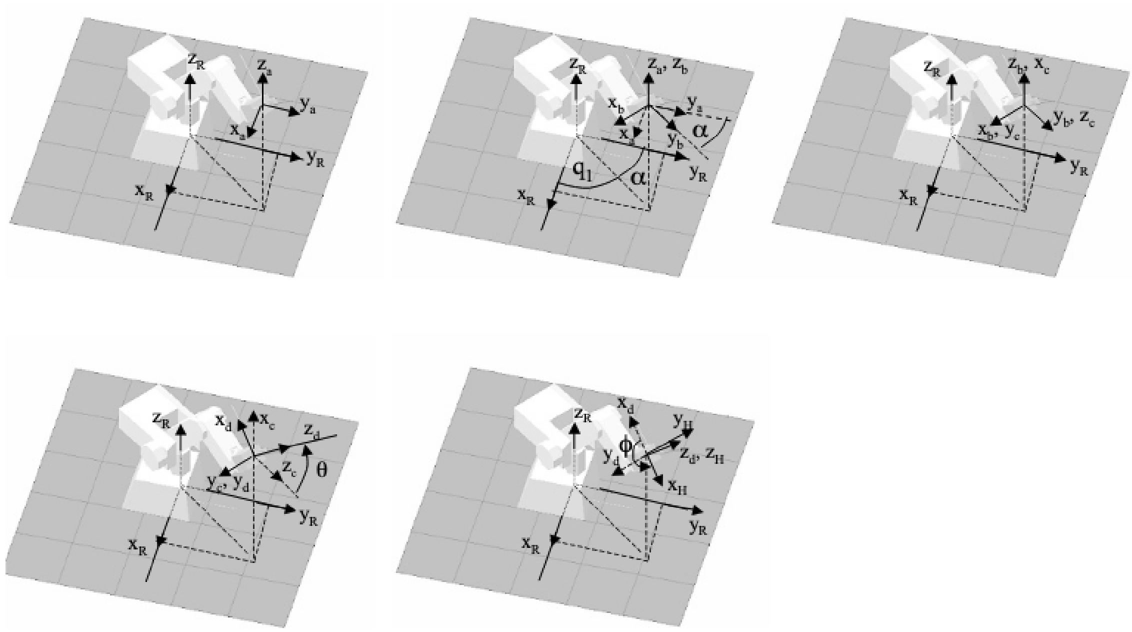


Figure 1.5: Reference frame transformation of the RVM1 end effector.

Answer of exercise 5

1.

$$ {}^{R}\boldsymbol{T}_{\boldsymbol{H}} = {}^{R}\boldsymbol{T}_{\boldsymbol{A}}{}^{A}\boldsymbol{T}_{\boldsymbol{B}}{}^{B}\boldsymbol{T}_{\boldsymbol{C}}{}^{C}\boldsymbol{T}_{\boldsymbol{D}}{}^{D}\boldsymbol{T}_{\boldsymbol{R}} \tag{1.7} $$

$$ {}^{R}\boldsymbol{T}_{\boldsymbol{A}} = \begin{bmatrix} 1 & 0 & 0 & {}^{R}h_x \\ 0 & 1 & 0 & {}^{R}h_y \\ 0 & 0 & 1 & {}^{R}h_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.8} $$

$$ {}^{A}\boldsymbol{T}_{\boldsymbol{B}} = \begin{bmatrix} c(-\alpha) & -s(-\alpha) & 0 & 0 \\ s(-\alpha) & c(-\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.9} $$

$$ {}^{B}\boldsymbol{T}_{\boldsymbol{C}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.10} $$

$$ {}^{C}\boldsymbol{T}_{\boldsymbol{D}} = \begin{bmatrix} c\theta & 0 & s\theta & 0 \\ 0 & 0 & 0 & 0 \\ -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.11} $$

$$
{}^{D}\boldsymbol{T}_{\boldsymbol{H}} = \begin{bmatrix} c\phi & -s\phi & 0 & 0 \\ s\phi & c\phi & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.12}
$$

finally, solving for the product in 1.7 we get:

$$
{}^{R}\boldsymbol{T}_{\boldsymbol{H}} = \begin{bmatrix} -cq_1 s\theta c\phi + sq_1 s\phi & cq_1 s\theta s\phi + sq_1 c\phi & cq_1 c\theta & {}^{R}h_x \\ -sq_1 s\theta c\phi - cq_1 s\phi & sq_1 s\theta s\phi - cq_1 c\phi & sq_1 c\theta & {}^{R}h_y \\ c\theta c\phi & -c\theta s\phi & s\theta & {}^{R}h_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.13}
$$

Exercise 6 GPS

A GPS receiver provides measurements with respect to the ECEF **E** (see Figure 1.6). To be able to perform local measurements, we define the **N** fixed in a point ${}^{\mathbf{E}}\mathbf{p}$ of the earth's surface. The XY plane of the **N** is tangent to the surface of the earth, and $\mathbf{z_N}$ points towards the centre of the earth (slightly misaligned with the centre of the earth due to its eccentricity as shown in Figure 1.6.

1. Assuming that the ECEF coordinates of the origin of **N** (point ${}^{E}\boldsymbol{p} = [\boldsymbol{a}\ \boldsymbol{b}\ \boldsymbol{c}]^{\boldsymbol{T}}$) are known find the symbolic expression of the ${}^{N}\boldsymbol{T}_{\boldsymbol{E}}$ matrix that given a GPS measurement, allows to reference it to the local coordinate system **N**.

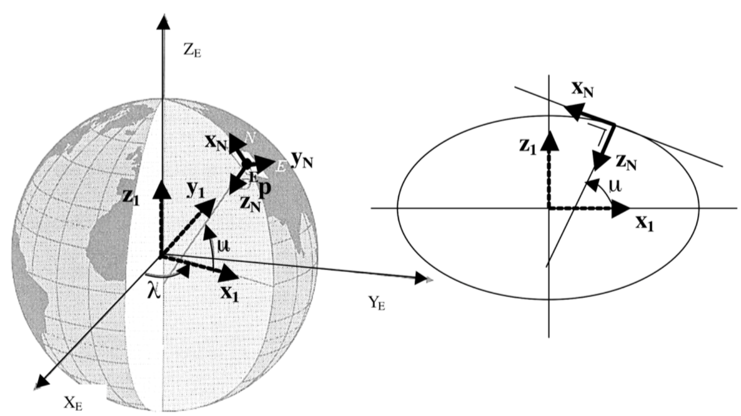


Figure 1.6: a) Relationship between the ECEF **E** and the **N**; b) Detail of the section of the earth

Answer of exercise 6

The requested transformation ${}^{N}\boldsymbol{T}_{\boldsymbol{E}}$ is the inverse of ${}^{E}\boldsymbol{T}_{\boldsymbol{N}}$. Therefore we can compute ${}^{E}\boldsymbol{T}_{\boldsymbol{N}}$ and invert it later on. ${}^{E}\boldsymbol{T}_{\boldsymbol{N}}$ can be computed as follows as follows:

$$
{}^{E}\boldsymbol{T}_{\boldsymbol{N}} = Trans(a,b,c) \cdot Rot(z,\lambda) \cdot Rot\left(y, -\mu - \frac{\pi}{2}\right) \tag{1.14}
$$

$$
{}^{E}\boldsymbol{T}_{\boldsymbol{N}} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\lambda & -s\lambda & 0 & a \\ s\lambda & c\lambda & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\left(-\mu - \frac{\pi}{2}\right) & 0 & s\left(-\mu - \frac{\pi}{2}\right) & 0 \\ 0 & 1 & 0 & 0 \\ -s\left(-\mu - \frac{\pi}{2}\right) & 0 & c\left(-\mu - \frac{\pi}{2}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.15}
$$

now, since

$$c\left(-\mu-\frac{\pi}{2}\right)=c\left(\mu+\frac{\pi}{2}\right)=-s(\mu) \tag{1.16}$$

and

$$s\left(-\mu-\frac{\pi}{2}\right)=-s\left(\mu+\frac{\pi}{2}\right)=-c(\mu) \tag{1.17}$$

(1.14) can be rewritten as:

$$\begin{aligned}{}^{E}\boldsymbol{T}_{N} &= \begin{bmatrix}1&0&0&a\\0&1&0&b\\0&0&1&c\\0&0&0&1\end{bmatrix}\begin{bmatrix}c\lambda&-s\lambda&0&a\\s\lambda&c\lambda&0&b\\0&0&1&c\\0&0&0&1\end{bmatrix}\begin{bmatrix}-s\mu&0&-c\mu&0\\0&1&0&0\\c\mu&0&-s\mu&0\\0&0&0&1\end{bmatrix}\\ &= \begin{bmatrix}-c\lambda s\mu&-s\lambda&-c\lambda c\mu&a\\-s\lambda s\mu&c\lambda&-s\lambda c\mu&b\\c\mu&0&-s\mu&c\\0&0&0&1\end{bmatrix}.\end{aligned} \tag{1.18}$$

No we have to invert it to get ${}^{N}\boldsymbol{T}_{E}$. Given the homogeneous matrix:

$${}^{E}\boldsymbol{T}_{N}=\begin{bmatrix}{}^{E}\boldsymbol{R}_{N}&{}^{E}\boldsymbol{o}_{N}\\\mathbf{0}_{1\times3}&1\end{bmatrix} \tag{1.19}$$

where

$${}^{E}\boldsymbol{R}_{N}=\begin{bmatrix}-c\lambda s\mu&-s\lambda&-c\lambda c\mu\\-s\lambda s\mu&c\lambda&-s\lambda c\mu\\c\mu&0&-s\mu\end{bmatrix} \tag{1.20}$$

and

$${}^{E}\boldsymbol{p}=\begin{bmatrix}a\\b\\c\end{bmatrix}. \tag{1.21}$$

we can use the formula of the inversion of an homogeneous transformation:

$${}^{N}\boldsymbol{T}_{E}={}^{E}\boldsymbol{T}_{N}^{-1}=\begin{bmatrix}{}^{E}\boldsymbol{R}_{N}^{T}&-{}^{E}\boldsymbol{R}_{N}^{T}{}^{E}\boldsymbol{p}\\\mathbf{0}_{1\times3}&1\end{bmatrix}. \tag{1.22}$$

$${}^{N}\boldsymbol{R}_{E}^{T}=\begin{bmatrix}-c\lambda s\mu&-s\lambda s\mu&c\mu\\-s\lambda&c\lambda&0\\-c\lambda c\mu&-s\lambda c\mu&-s\mu\end{bmatrix} \tag{1.23}$$

$$-{}^{E}\boldsymbol{R}_{N}^{T}{}^{E}\boldsymbol{p}=\begin{bmatrix}-c\lambda s\mu&-s\lambda s\mu&c\mu\\-s\lambda&c\lambda&0\\-c\lambda c\mu&-s\lambda c\mu&-s\mu\end{bmatrix}\begin{bmatrix}a\\b\\c\end{bmatrix}=\begin{bmatrix}c\lambda s\mu a+s\lambda s\mu b-c\mu c\\s\lambda a-c\lambda b\\c\lambda c\mu a+s\lambda c\mu b+s\mu c\end{bmatrix} \tag{1.24}$$

$$
{}^{N}\boldsymbol{T_E} = \begin{bmatrix} -c\lambda s\mu & -s\lambda s\mu & c\mu & c\lambda s\mu a + s\lambda s\mu b - c\mu c \\ -s\lambda & c\lambda & 0 & s\lambda a - c\lambda b \\ -c\lambda c\mu & -s\lambda c\mu & -s\mu & c\lambda c\mu a + s\lambda c\mu b + s\mu c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.25}
$$

Exercise 7 The Prestige

The french submarine Nautile is performing the damage assessment operations (Figure 1.7) of the oil tanker Prestige. Nautile is in contact supervision with a vessel that guides the submarine to the sunken oil tanker. The Prestige position ${}^{\mathbf{V}}\mathbf{p}$ with respect to this vessel (**V**) is known. Let:

- **V** the coordinate system located on the vessel.
- **N** the coordinate system located at the Nautile submarine.
- **P** the coordinate system located in the Prestige.
- The X-axis of frame **V** is aligned with the Earth's magnetic North.
- $\psi = 270°$ is the orientation of the submarine Nautile (X-axis) with respect to the magnetic North.
- λ is the orientation of the Prestige (X-axis) with respect to the magnetic North.
- ${}^{\mathbf{V}}\mathbf{n} = [0.8\ -0.5\ 2.5]^T$ the origin of **N** with respect to **V**
- ${}^{\mathbf{V}}\mathbf{p} = [-0.2\ 0.3\ 3.5]^T$ the position of the Prestige with respect to **V**.

Note that measurements are given in km and degrees.

1. Calculate the numerical value of the homogeneous matrix ${}^{\mathbf{V}}\mathbf{T_N}$.
2. Calculate the position of ${}^{\mathbf{N}}\mathbf{p}$.
3. From the new position ${}^{\mathbf{V}}\mathbf{n}' = [-0.15\ 0.25\ 3.4]^T$ with $\psi = 180°$, the Nautile has detected a point of interest in the Prestige indicated as ${}^{\mathbf{N}}\mathbf{x} = [-0.05\ 0.01\ 0.02]^T$. We want to know what part of the tanker is being inspected (i.e., to know ${}^{\mathbf{P}}\mathbf{x}$). Explain, without performing any calculation, all operations required to find ${}^{\mathbf{P}}\mathbf{x}$.

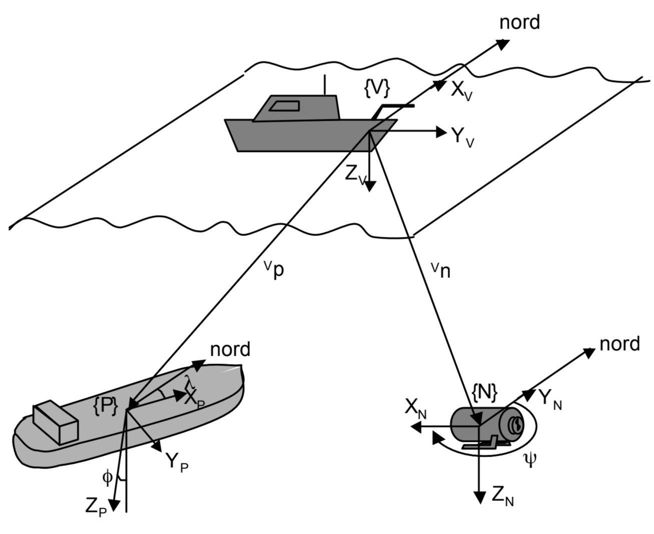


Figure 1.7: Prestige damage assessment operations.

Answer of exercise 7

1. $\mathbf{^VT_N}$ can be computed as follows:

$$\mathbf{^VT_N} = \mathbf{Trans}(^{\mathbf{V}}\mathbf{n}) \cdot \mathbf{I} \cdot \mathbf{Rot}(\mathbf{z}, \psi) = \begin{bmatrix} 1 & 0 & 0 & {}^Vn_x \\ 0 & 1 & 0 & {}^Vn_y \\ 0 & 0 & 1 & {}^Vn_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\psi & -s\psi & 0 & 0 \\ s\psi & c\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0.8 \\ -1 & 0 & 0 & -0.5 \\ 0 & 0 & 1 & 2.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.26}$$

2. $\mathbf{^Np}$ can be computed as follows:

$$^N\boldsymbol{p} = {}^N\boldsymbol{T_V} \cdot {}^V\boldsymbol{p} \tag{1.27}$$

$^N\boldsymbol{T_V}$ can be computed inverting $^V\boldsymbol{T_N}$ using the formula for inverting an homogeneous transformation:

$$^A\boldsymbol{H_B}^{-1} = \begin{bmatrix} {}^A\boldsymbol{R_B^T} & -{}^A\boldsymbol{R_B^T}{}^A\boldsymbol{p} \\ \boldsymbol{0} & 1 \end{bmatrix}^{-1} \tag{1.28}$$

$$^N\boldsymbol{T_V} = [{}^V\boldsymbol{T_N}]^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0.8 \\ -1 & 0 & 0 & -0.5 \\ 0 & 0 & 1 & 2.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 0 & -1 & 0 & -0.5 \\ 1 & 0 & 0 & -0.8 \\ 0 & 0 & 1 & -2.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.29}$$

$$^N\boldsymbol{p} = {}^N\boldsymbol{T_V} \cdot {}^V\boldsymbol{p} = \begin{bmatrix} 0 & -1 & 0 & -0.5 \\ 1 & 0 & 0 & -0.8 \\ 0 & 0 & 1 & -2.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.2 \\ 0.3 \\ 3.5 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.8 \\ -1 \\ 1 \end{bmatrix} \tag{1.30}$$

3. The position of $^P\boldsymbol{x}$ can be computed as:

$$^P\boldsymbol{x} = {}^P\boldsymbol{T_N} \cdot {}^N\boldsymbol{x} \tag{1.31}$$

The point $^N\boldsymbol{x}$ is already known, and the transformation $^P\boldsymbol{T_N}$ is given by:

$$^P\boldsymbol{T_N} = {}^P\boldsymbol{T_V} \cdot {}^V\boldsymbol{T_N} \tag{1.32}$$

where

$$^P\boldsymbol{T_V} = [{}^V\boldsymbol{T_P}]^{-1} \tag{1.33}$$

can be computed as,

$$ {}^{V}\boldsymbol{T}_{\boldsymbol{P}} = \boldsymbol{Trans}({}^{V}\boldsymbol{p}) \cdot \boldsymbol{I} \cdot \boldsymbol{Rot}(z, \boldsymbol{\lambda}) \tag{1.34} $$

and ${}^{V}\boldsymbol{T}_{\boldsymbol{N}}$ is:

$$ {}^{V}\boldsymbol{T}_{\boldsymbol{N}} = \boldsymbol{Trans}(\mathbf{-0.05\ 0.01\ 0.02}) \cdot \boldsymbol{I} \cdot \boldsymbol{Rot}(z, \mathbf{180}) \tag{1.35} $$

Exercise 8 Mars Rover

The prestigious GIT research centre (Girona Institute of Technology) has been commissioned from perform the calculation of positions to be accessible by the manipulator installed on a mobile robot sent to Mars. This robot has 3 coordinate systems (Figure 1.8), the one for robot (**R**), the one for camera (**C**) and the one for manipulator (**M**).

The following information is known:

- The position of the **M** with respect to **R** is ${}^{R}\boldsymbol{T}_{\boldsymbol{M}} = [20\ 0\ -10]^T$.
- The position of the **C** with respect to **R** is ${}^{R}\boldsymbol{T}_{\boldsymbol{C}} = [-30\ -15\ 80]^T$.

Note that all axes are either parallel or orthogonal to **R** axes, with the exception of x_C and z_C that depend on the angle α.

1. Compute the ${}^{R}\boldsymbol{T}_{\boldsymbol{M}}$ homogeneous matrix.

2. Compute the ${}^{R}\boldsymbol{T}_{\boldsymbol{C}}$ homogeneous matrix as a function of α.

3. Assuming ${}^{C}\boldsymbol{p} = [20\ 100\ 70]^T$ and $\alpha = 30°$, which are the coordinates of ${}^{M}\boldsymbol{p}$.

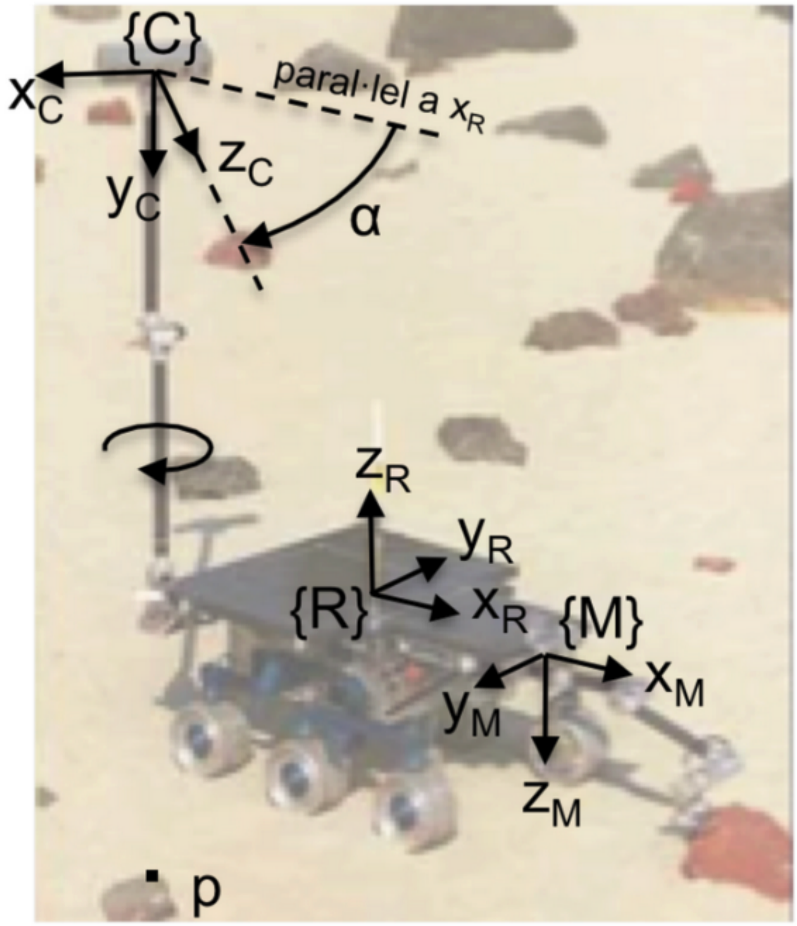


Figure 1.8: Reference frames for the Mars Rover.

Answer of exercise 8

1. To compute ${}^{R}\boldsymbol{T_M}$ we can apply visual inspection. In (1.36) we state how to compute the matrix using the unitary vectors of the target frame represented in the original frame.

$$ {}^{R}\boldsymbol{T_M} = \begin{bmatrix} {}^{R}\boldsymbol{x_M} & {}^{R}\boldsymbol{y_M} & {}^{R}\boldsymbol{z_M} & {}^{R}\boldsymbol{p_M} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.36} $$

From 1.8 we can observe that ${}^{R}\boldsymbol{x_M} = [1\ 0\ 0]^T$, ${}^{R}\boldsymbol{y_M} = [0\ -1\ 0]^T$, ${}^{R}\boldsymbol{z_M} = [0\ 0\ -1]^T$ and ${}^{R}\boldsymbol{T_M} = [20\ 0\ -10]^T$. Therefore, applying (1.37) we get:

$$ {}^{R}\boldsymbol{T_M} = \begin{bmatrix} 1 & 0 & 0 & 20 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.37} $$

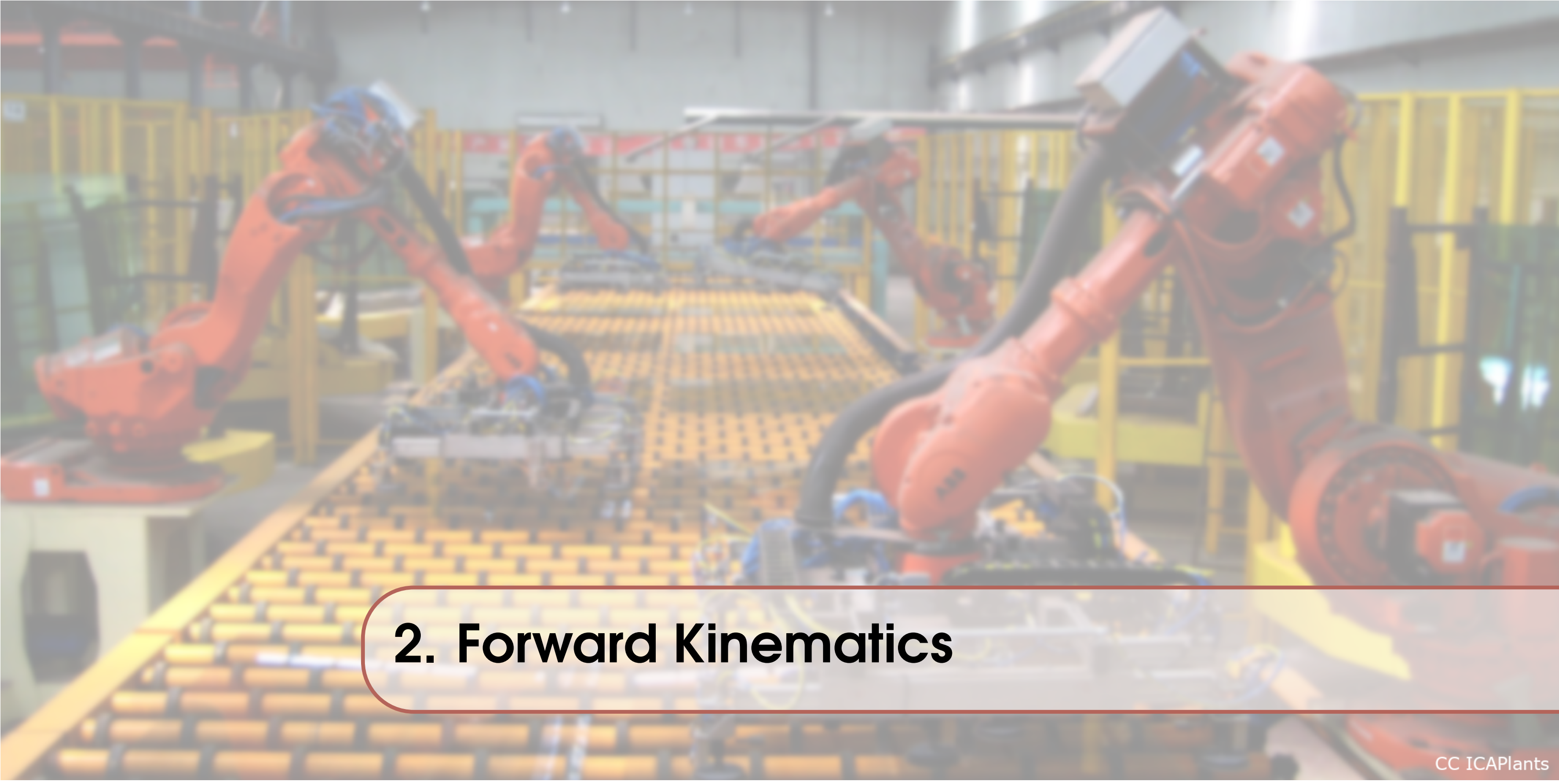
2. To compute the ${}^{R}\boldsymbol{T_C}$ homogeneous matrix as a function of α, we can apply the following transformation:

$$ {}^{R}T_C = t(-30,-15,80)\cdot Rot(x,-\frac{\pi}{2})\cdot Rot(y,\pi+\alpha) \tag{1.38} $$

3. Assuming ${}^{C}\boldsymbol{p} = [20\ 100\ 70]^T$ and $\alpha = 30°$, the coordinates of ${}^{M}\boldsymbol{p}$ can be computed as:

$$ {}^{M}p = \left({}^{R}T_{M}\right)^{-1} \cdot {}^{R}T_{C} \cdot {}^{C}p \tag{1.39} $$

$$ {}^{m}p = \begin{bmatrix} 1 & 0 & 0 & -20 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -0.87 & 0 & -0.5 & -30 \\ 0.5 & 0 & -0.87 & -15 \\ 0 & -1 & 0 & 80 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 20 \\ 100 \\ 70 \\ 1 \end{bmatrix} = \begin{bmatrix} -102.32 \\ 65.62 \\ 10. \\ 1. \end{bmatrix} \tag{1.40} $$



2. Forward Kinematics

2.1 Forward Kinematics Theory

An industrial robot is a system used for industrial applications such as welding, painting, assembly, palletizing, or product inspection, to name a few. Robots are programmable systems that can perform a series of motions, with great force, speed, and precision, in three or more axes.

Industrial robots are composed of links and joints. While links are rigid parts, joints are moving parts that connect two links. In general, it can be assumed that there are only two types of joints: angular and prismatic. Angular joints produce a motion of rotation with respect to an axis and are usually limited by a minimum and maximum angle of rotation, although there are joints that can rotate freely. On the other hand, prismatic joints generate linear motion along a single axis.

There are joints that can move linearly or rotate in more than one axis at a time. However, to simplify the description of a robot and to perform calculations in a systematic way, it is usually assumed that each joint produces a single motion and, therefore, joints that combine motions in different axis are usually transformed into a combination of joints with a single motion each.

An industrial robot is instrumented in such a way that it is possible to measure the position of each joint. For this purpose, encoder-type sensors capable of measuring with high precision the degrees that an angular joint has rotated or the millimeters that a prismatic joint has moved are typically used. However, industrial robots do not usually have any system to measure the position and orientation of their terminal element with respect to a fixed reference frame such as the robot base.

When we talk about direct kinematics in a robot, we refer to the use of a set of equations that allows to calculate the position and orientation of the terminal element of a robot with respect to its base using only the values of the robot's joints, which is what can actually be measured, and some other constant parameters such as the length of each link.

2.1.1 Denavit-Hartenberg algorithm

To define the forward kinematic equations for a particular robot, there is a systematic set of steps, i.e., an algorithm, that can be applied. This algorithm is called Denavit-Hartenberg after their creators. [Hartenberg, Richard Scheunemann; Denavit, Jacques (1965). Kinematic synthesis of linkages. McGraw-Hill series in mechanical engineering. New York: McGraw-Hill. p. 435.]. This algorithm describes a particular convention for attaching reference frames to the links of a robot manipulator. The algorithm assigns a reference frame to each robot link. Then, it proposes a systematic way to compute a trasformation matrix (i.e., a homogeneous 3D rotation and translation) that relates each reference frame with the previous. These transformation matrices depend only on 4 parameters:

- Joint angle (θ_i): The angle of rotation from X_{i-1} to X_i measured on Z_{i-1}. This is the joint variable if the joint is angular.
- Joint distance (d_i): The distance from the origin of reference frame $i-1^{th}$ to the intersection of Z_{i-1} axis and X_i axis measured along Z_{i-1} axis. This is the joint variable if the joint is prismatic.
- Link length (a_i): The distance from the intersection of Z_{i-1} axis and the X_i axis to the origin of the i^{th} reference system measured along the X_i axis.
- Link twist angle α_i: The angle of rotation from axis Z_{i-1} to axis Z_i measured around axis X_i.

Once all the transformation matrices that relate a link i^{th} with link $i-1^{th}$ (i.e., ${}^{i-1}A_i$) have been defined, it is required to multiply all these matrices together in order to obtain the transformation matrix that relates the robot base (i.e., $i=0$) to the end-effector link (i.e., $i=N$ being N the number of degrees in the robot manipulator).

$$^{R}T_H = \prod_{i=1}^{N} {}^{i-1}A_i \tag{2.1}$$

The ${}^{i-1}A_i$ matrix is created from the four variables θ_i, d_i, a_i, and α_i according to the following scheme.

$${}^{i-1}A_i = \mathrm{Trans}(d_i, Z) \cdot \mathrm{Rot}(\theta_i, Z) \cdot I \cdot \mathrm{Trans}(a_i, X) \cdot \mathrm{Rot}(\alpha_i, X) \tag{2.2}$$

The first transformation ($\mathrm{Trans}(d_i, Z)$) and rotation ($\mathrm{Rot}(\theta_i, Z)$) are respect to the Z axis of the $i-1^{th}$ reference frame (i.e., pre-multiplication) while the second translation ($\mathrm{Trans}(a_i, X)$) and rotation ($\mathrm{Rot}(\alpha_i, X)$) are done with respect to the (i^{th}) reference frame (i.e., post-multiplication).

The matrix obtained when computing (2.2) is:

$${}^{i-1}A_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

To obtain these four parameters for each joint the following algorithm must be applied:

1. Enumerate joints from 1 to n starting for the base and finalizing for the yaw, pitch, and roll of the end-effector (in this order).
2. Assign reference frame L_0 to the base. Ensure that Z_0 is coincident with the axis of joint 1. Initialize k = 1.
3. Set Z_k coincident with the axis of joint k + 1.
4. Set the origin of L_k at the intersection of the axis Z_k and Z_{k-1}. If Z_k and Z_{k-1} do not intersect, use the intersection of Z_k with a shared normal of Z_k and Z_{k-1}.
5. Set X_k orthogonal to Z_k and Z_{k-1}. If Z_k and Z_{k-1} are parallel, set X_k perpendicular to Z_{k-1}, place it along the link and heading outside.
6. Set Y_k to complet the dextrorotatory reference system L_k.
7. Set k = k + 1. If k < n, return to step 3.
8. Set the origen of L_n at the tip of the end effector. Z_n must be coincident with the approach vector a, Y_n must be coincident with the orientation vector o, and X_n must be coincident with the normal vector n. Set k = 1.
9. Define a point b_k at the intersection of the axis X_k and Z_{k-1}. If they do not intersect, set it at the intersection of X_k with a shared normal of X_k and Z_k.
10. θ_k is the rotation angle from X_{k-1} to X_k measured with respect to axis Z_{k-1}.
11. d_k is the distance from the origen of the reference system L_{k-1} to b_k measured with respect to axis Z_{k-1}.
12. a_k is the distance from point b_k to the origen of the reference system L_k measured with respect to axis X_k.
13. α_k is the rotation angle from Z_{k-1} to Z_k measured with respect to axis X_k.
14. Set k = k + 1. If k $\leq$ n go to step 9.

2.2 Forward Kinematics Exercises

Exercise 9 3 DoF angular manipulator

Use the DH algorithm to set the axes of the 3 DoF manipulator shown in Figure 2.1. Fill in Table 2.1 with the DH parameters. Red arrows indicate the axis of rotation of each joint. Indicate the required distances to complete the DH table.

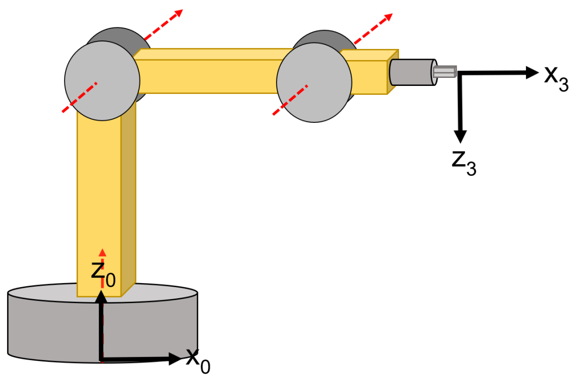


Figure 2.1: 3 DoF angular manipulator.

[ht]

Table 2.1: DH parameters table.

DoF	θ	**d**	**a**	α
1				
2				
3				

Calculate, symbolically, the transformation matrices that relate each joint with the previous (i.e., ${}^{0}A_1$, ${}^{1}A_2$, ${}^{2}A_3$), as well as the *Hand to Robot* transformation matrix (i.e., ${}^{R}T_H$ or ${}^{0}A_3$) using the pattern:

$$
{}^{i-1}A_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Answer of exercise 9

The axis for the 3 DoF angular manipulator are shown in Figure 2.2 while the DH parameters can be seen at Table 2.2.

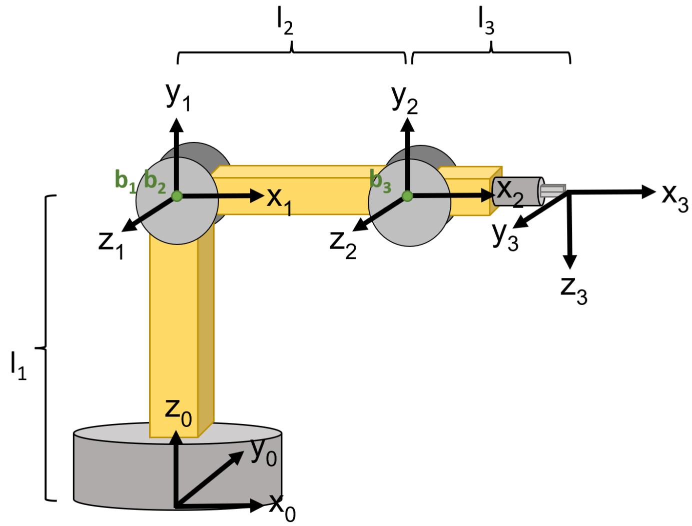


Figure 2.2: DH solution of the 3 DoF angular manipulator.

Table 2.2: DH parameters table.

DoF	θ	d	a	α
1	$q_1 = 0°$	l_1	0	90°
2	$q_2 = 0°$	0	l_2	0°
3	$q_3 = 0°$	0	l_3	90°

The transformation matrices obtained from Table 2.2 are shown next. The order in which matrices are merged is important to simplifuy the final result. Here, for instance, we first combine matrices ${}^1A_2$ and ${}^2A_3$ to obtain ${}^1A_3$, then combine ${}^0A_1$ and ${}^1A_3$ to obtain ${}^0A_3$. Following this order it is easier to apply trigonometric rules to simplify the final result.

$$
{}^0A_1 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
{}^1A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & l_2c_2 \\ s_2 & c_2 & 0 & l_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
{}^2A_3 = \begin{bmatrix} c_3 & 0 & s_3 & l_3c_3 \\ s_3 & 0 & -c_3 & l_3s_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^1A_3 = \begin{bmatrix} c_{23} & 0 & s_{23} & l_3c_{23}+l_2c_2 \\ s_{23} & 0 & -c_{23} & l_3s_{23}+l_2s_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^0A_3 = \begin{bmatrix} c_1c_{23} & s_1 & c_1s_{23} & c_1\left(l_3c_{23}+l_2c_2\right) \\ s_1c_{23} & -c_1 & s_1s_{23} & s_1\left(l_3c_{23}+l_2c_2\right) \\ s_{23} & 0 & -c_{23} & l_3s_{23}+l_2s_2+l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Exercise 10 3 DoF angular manipulator v2

Use the DH algorithm to set the axes of the 3 DoF manipulator shown in Figure 2.3. Fill in Table 2.3 with the DH parameters. Red arrows indicate the axis of rotation of each joint. Indicate the required distances to complete the DH table.

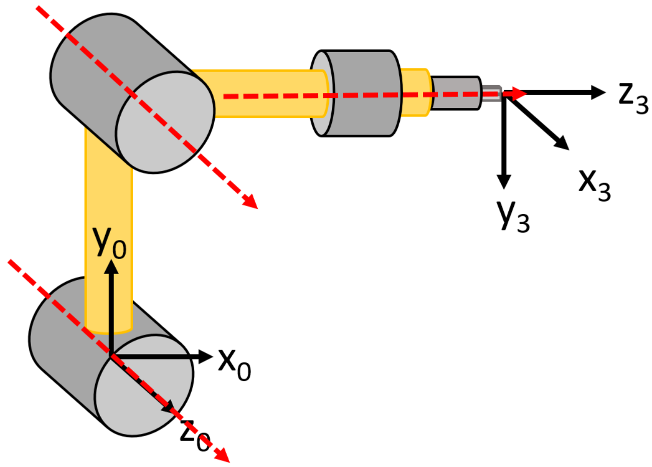


Figure 2.3: 3 DoF angular manipulator.

[ht]

Table 2.3: DH parameters table.

DoF	θ	d	a	α
1				
2				
3				

Answer of exercise 10

The axis for the 3 DoF angular manipulator are shown in Figure 2.4 while the DH parameters can be seen at Table 2.4.

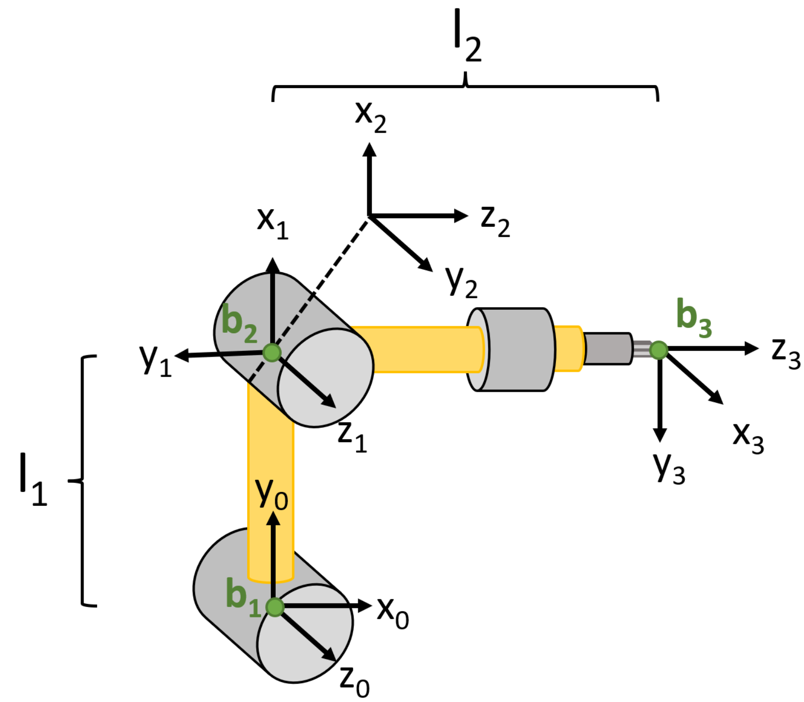


Figure 2.4: DH solution of the 3 DoF angular manipulator.

Table 2.4: DH parameters table.

DoF	θ	d	a	α
1	$q_1 = 90°$	0	l_1	0°
2	$q_2 = 0°$	0	0	90°
3	$q_3 = 90°$	l_2	0	0°

Exercise 11 3 DoF RTR manipulator

Use the DH algorithm to set the axes of the 3 DoF manipulator shown in Figure 2.5. Note that the first and last joints are angular (rotational movement) but the second joint is prismatic (translational movement). Fill in Table 2.5 with the DH parameters. Red arrows indicate the axis of rotation of each joint. Indicate the required distances to complete the DH table.

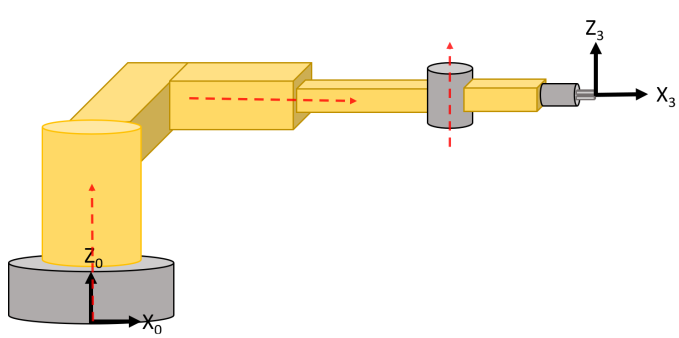


Figure 2.5: 3 DoF angular manipulator.

[ht]

Table 2.5: DH parameters table.

DoF	θ	d	a	α
1				
2				
3				

Calculate, symbolically, the transformation matrices that relate each joint with the previous (i.e., ${}^0A_1$, ${}^1A_2$, ${}^2A_3$), as well as the *Hand to Robot* transformation matrix (i.e., ${}^RT_H$ or ${}^0A_3$) using the pattern:

$$
{}^{i-1}A_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Answer of exercise 11

The axis for the 3 DoF RTR manipulator are shown in Figure 2.6 while the DH parameters can be seen at Table 2.6.

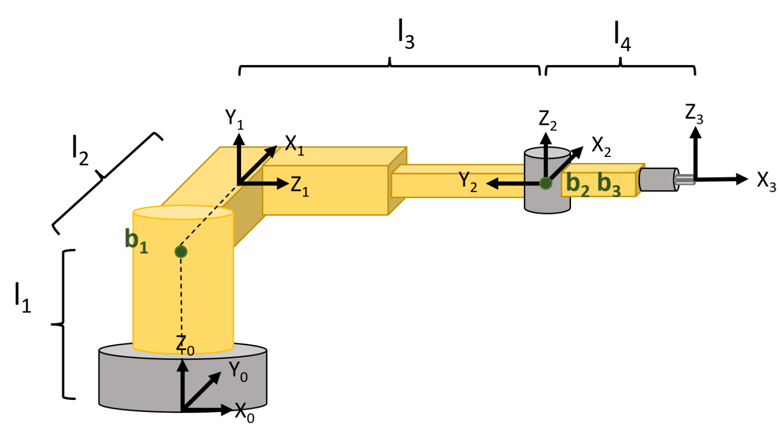


Figure 2.6: DH solution of the 3 DoF angular manipulator.

Table 2.6: DH parameters table.

DoF	θ	d	a	α
1	$q_1 = 90°$	l_1	l_2	$90°$
2	$0°$	$q_2 = l_3$	0	$-90°$
3	$q_3 = -90°$	0	l_4	$0°$

The transformation matrices obtained from Table 2.6 are:

$$
{}^{0}A_1 = \begin{bmatrix} c_1 & 0 & s_1 & l_2c_1 \\ s_1 & 0 & -c_1 & l_2s_1 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \; {}^{1}A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{2}A_3 = \begin{bmatrix} c_3 & 0 & s_3 & l_4c_3 \\ s_3 & 0 & -c_3 & l_4s_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{0}A_3 = \begin{bmatrix} c_{13} & 0 & s_{13} & l_4c_{13} + l_2c_1 + q_2s_1 \\ s_{13} & 0 & -c_{13} & l_4s_{13} + l_2s_1 - q_2c_1 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Exercise 12 4 DoF RTTR manipulator

Use the DH algorithm to set the axes of the 4 DoF manipulator shown in Figure 2.7. Note that the first and last joints are angular (rotation) and the 2^{nd} and 3^{rd} are prismatic (translation). Fill in Table 2.7 with the DH parameters. Red arrows indicate the axis of rotation/translation of each joint. Indicate the required distances to complete the DH table.

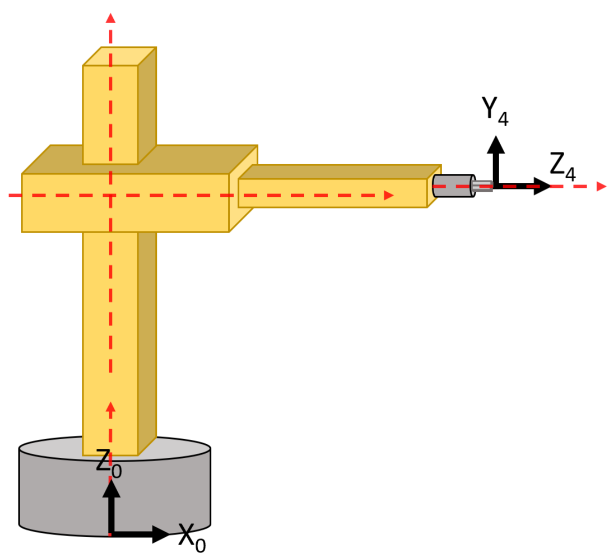


Figure 2.7: 4 DoF RTTR manipulator.

Table 2.7: DH parameters table.

DoF	θ	d	a	α
1				
2				
3				
4				

Answer of exercise 12

The axis for the 4 DoF manipulator are shown in Figure 2.8 while the DH parameters can be seen at Table 2.8.

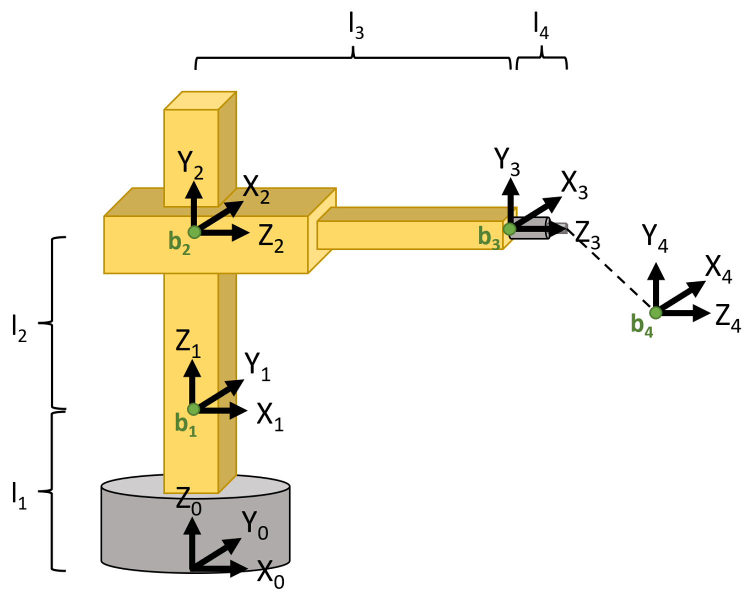


Figure 2.8: DH solution of the 4 DoF RTTR manipulator.

Table 2.8: DH parameters table.

DoF	θ	d	a	α
1	$q_1 = 0°$	l_1	0	0°
2	90°	$q_2 = l_2$	0	90°
3	0°	$q_3 = l_3$	0	0°
4	$q_4 = 0°$	l_4	0	0°

Exercise 13 4 DoF RRTR manipulator

Use the DH algorithm to set the axes of the 4 DoF manipulator shown in Figure 2.9. Note that all joints are angular (rotation) except the 3rd joint which is prismatic (translation). Fill in Table 2.9 with the DH parameters. Red arrows indicate the axis of rotation of each joint. Indicate the required distances to complete the DH table.

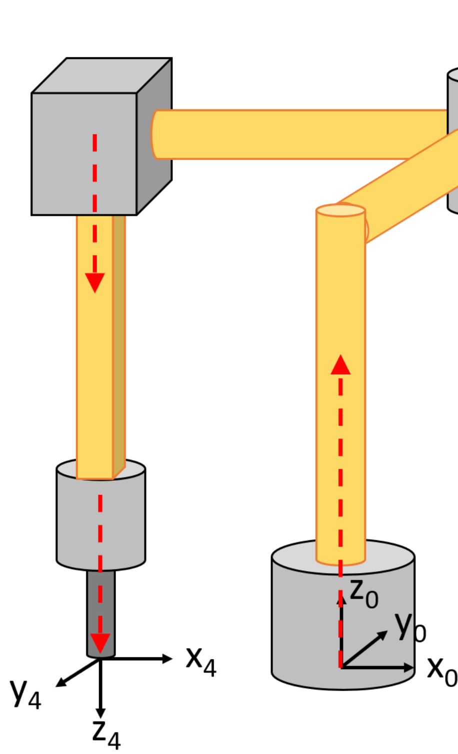


Figure 2.9: 4 DoF RRTR manipulator.

Table 2.9: DH parameters table.

DoF	θ	**d**	**a**	α
1				
2				
3				
4				

Answer of exercise 13

The axis for the 4 DoF angular manipulator are shown in Figure 2.10 while the DH parameters can be seen at Table 2.10.

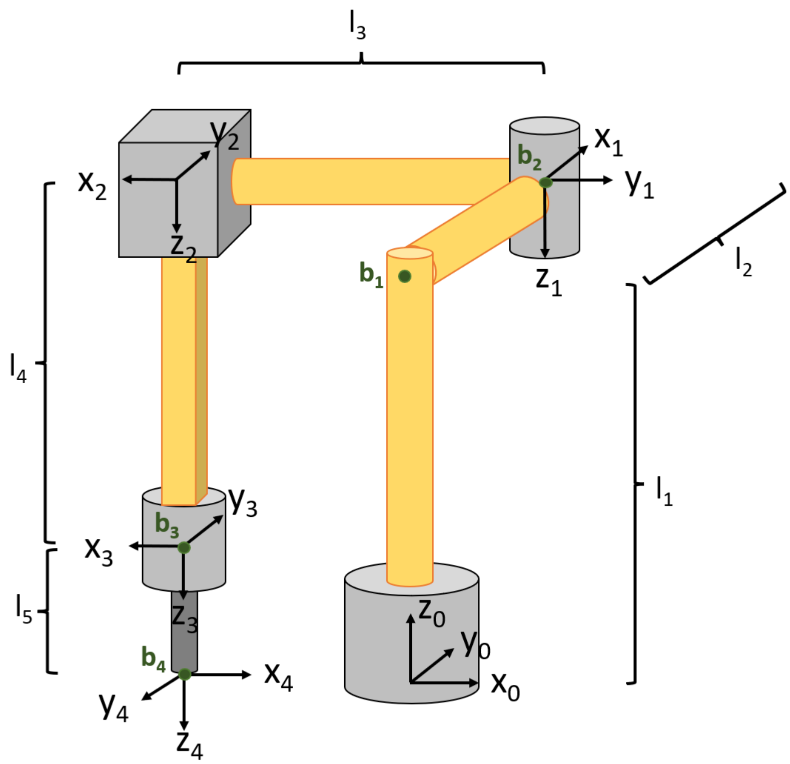


Figure 2.10: DH solution of the 4 DoF RRTR manipulator.

Table 2.10: DH parameters table.

DoF	θ	**d**	**a**	α
1	$q_1 = 90°$	l_1	l_2	$180°$
2	$q_2 = -90°$	0	l_3	$0°$
3	$0°$	$q_3 = l_4$	0	$0°$
4	$q_4 = 180°$	l_5	0	$0°$

Exercise 14 4 DoF TRTR manipulator

Use the DH algorithm to set the axes of the 4 DoF manipulator shown in Figure 2.11. Note that joints one and third are prismaic (translation) while two and four are angular (rotation). Fill in Table 2.11 with the DH parameters. Red arrows indicate the axis of rotation/translation of each joint. Indicate the required distances to complete the DH table.

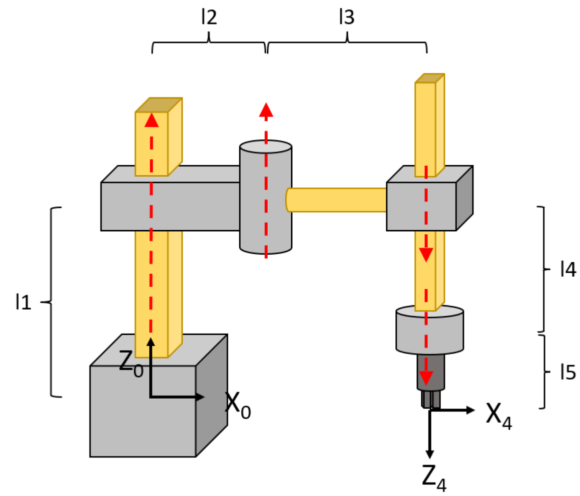


Figure 2.11: 4 DoF TRTR manipulator.

Table 2.11: DH parameters table.

DoF	θ	**d**	**a**	α
1				
2				
3				
4				

Answer of exercise 14

The axis for the 4 DoF manipulator are shown in Figure 2.12 while the DH parameters can be seen at Table 2.12.

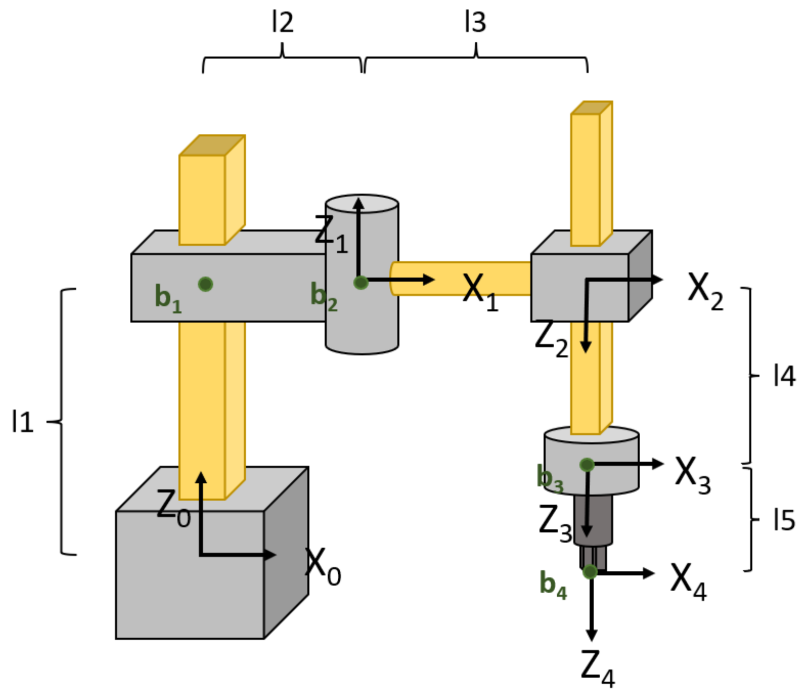


Figure 2.12: DH solution of the 4 DoF TRTR manipulator.

Table 2.12: DH parameters table.

DoF	θ	**d**	**a**	α
1	0°	$q_1 = l_1$	l_2	0°
2	$q_2 = 0^\circ$	0	l_3	180°
3	0°	$q_3 = l_4$	0	0°
4	$q_4 = 0^\circ$	l_5	0	0°

Exercise 15 5 DoF angular manipulator

Use the DH algorithm to set the axes of the 5 DoF manipulator shown in Figure 2.13. Fill in Table 2.13 with the DH parameters. Red arrows indicate the axis of rotation of each joint. Indicate the required distances to complete the DH table.

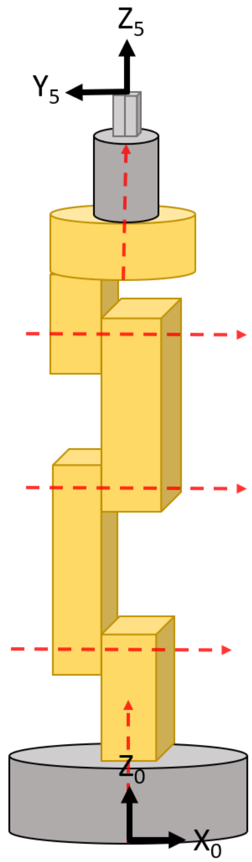


Figure 2.13: 6 DoF angular manipulator.

Table 2.13: DH parameters table.

DoF	θ	**d**	**a**	α
1				
2				
3				
4				
5				

Answer of exercise 15

The axis for the 5 DoF angular manipulator are shown in Figure 2.14 while the DH parameters can be seen at Table 2.14.

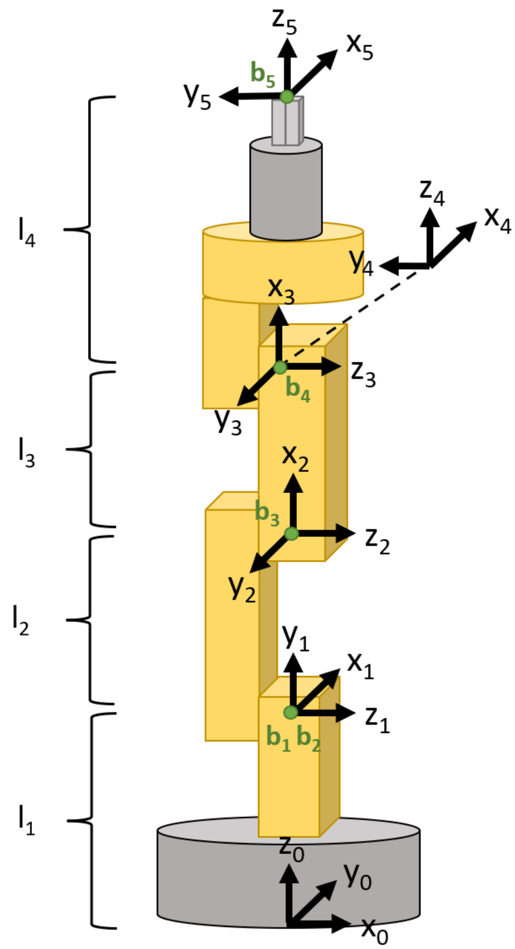


Figure 2.14: DH solution of the 5 DoF angular manipulator.

Table 2.14: DH parameters table.

DoF	θ	**d**	**a**	α
1	$q_1 = 90°$	l_1	0	$90°$
2	$q_2 = 90°$	0	l_2	$0°$
3	$q_3 = 0°$	0	l_3	$0°$
4	$q_4 = -90°$	0	0	$-90°$
5	$q_5 = 0°$	l_4	0	$0°$

Exercise 16 6 DoF angular manipulator

Use the DH algorithm to set the axes of the 6 DoF manipulator shown in Figure 2.15. Fill in Table 2.15 with the DH parameters. Red arrows indicate the axis of rotation of each joint. Indicate the required distances to complete the DH table.

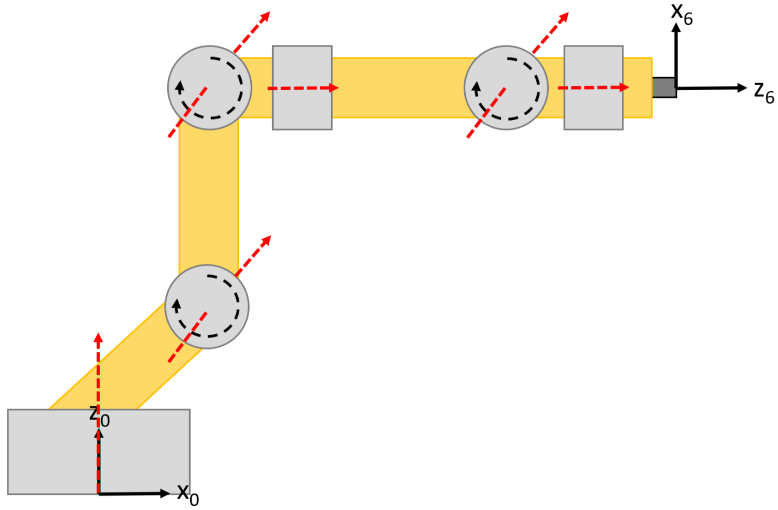


Figure 2.15: 6 DoF angular manipulator.

Table 2.15: DH parameters table.

DoF	θ	d	a	α
1				
2				
3				
4				
5				
6				

Answer of exercise 16

The axis for the 6 DoF angular manipulator are shown in Figure 2.16 while the DH parameters can be seen at Table 2.16.

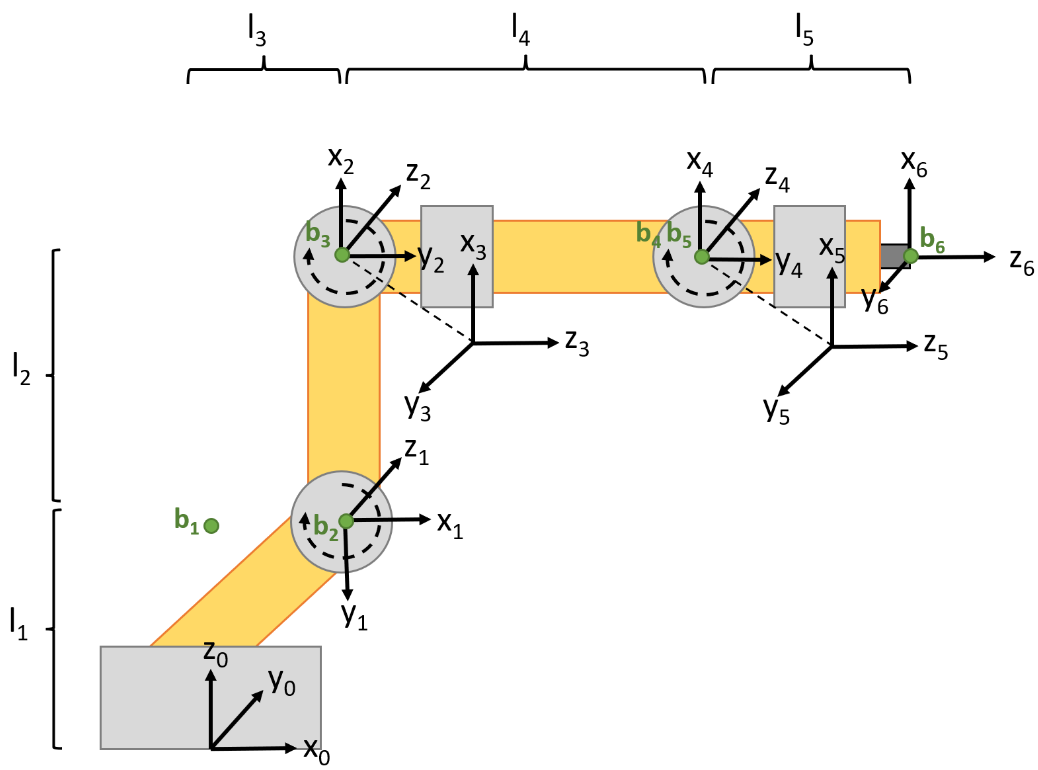


Figure 2.16: DH solution of the 6 DoF angular manipulator.

Table 2.16: DH parameters table.

DoF	θ	d	a	α
1	$q_1 = 0°$	l_1	l_3	$-90°$
2	$q_2 = -90°$	0	l_2	$0°$
3	$q_3 = 0°$	0	0	$-90°$
4	$q_4 = 0°$	l_4	0	$90°$
5	$q_5 = 0°$	0	0	$-90°$
6	$q_6 = 0°$	l_5	0	$0°$

Exercise 17 6 DoF angular manipulator v2

Use the DH algorithm to set the axes of the 6 DoF manipulator shown in Figure 2.17. Fill in Table 2.17 with the DH parameters. Red arrows indicate the axis of rotation of each joint. Indicate the required distances to complete the DH table.

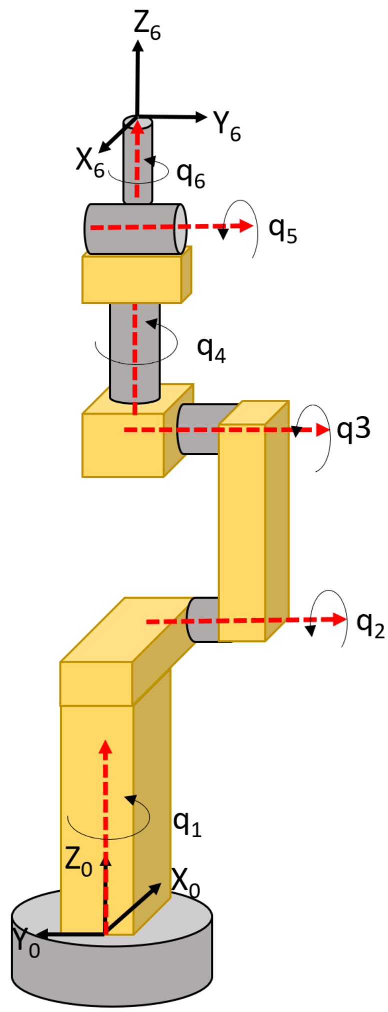


Figure 2.17: 6 DoF angular manipulator.

Table 2.17: DH parameters table.

DoF	θ	d	a	α
1				
2				
3				
4				
5				
6				

Answer of exercise 17

The axis for the 6 DoF angular manipulator are shown in Figure 2.18 while the DH parameters can be seen at Table 2.18.

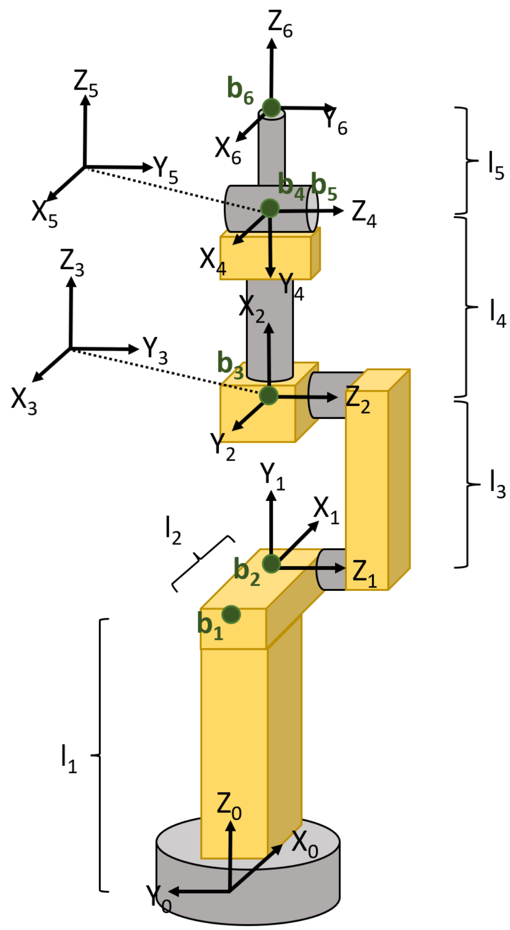


Figure 2.18: DH solution of the 6 DoF angular manipulator.

Table 2.18: DH parameters table.

DoF	θ	**d**	**a**	α
1	$q_1 = 0°$	l_1	l_2	90°
2	$q_2 = 90°$	0	l_3	0°
3	$q_3 = 90°$	0	0	90°
4	$q_4 = 0°$	l_4	0	−90°
5	$q_5 = 0°$	0	0	90°
5	$q_6 = 0°$	l_5	0	0°

Exercise 18 7 DoF RTRRRR manipulator

Use the DH algorithm to set the axes of the 6 DoF manipulator shown in Figure 2.15. Notice that all joints are angular (produce a rotation) excep the second one that is prismatic (produces a translation). Fill in Table 2.19 with the DH parameters. Red arrows indicate the axis of rotation of each joint. Indicate the required distances to complete the DH table.

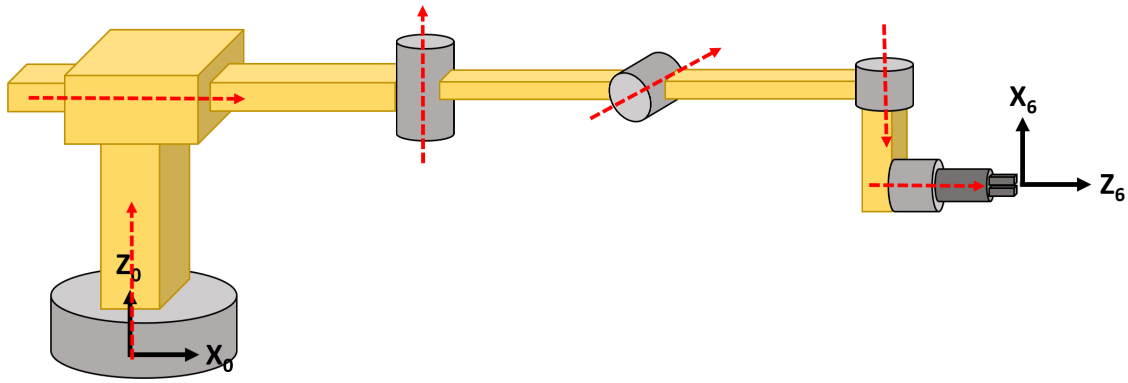


Figure 2.19: 6 DoF RTRRRR manipulator.

Table 2.19: DH parameters table.

DoF	θ	**d**	**a**	α
1				
2				
3				
4				
5				
6				

Answer of exercise 18

The axis for the 6 DoF RTRRRR manipulator are shown in 2.20 while the DH table can be seen at 2.20.

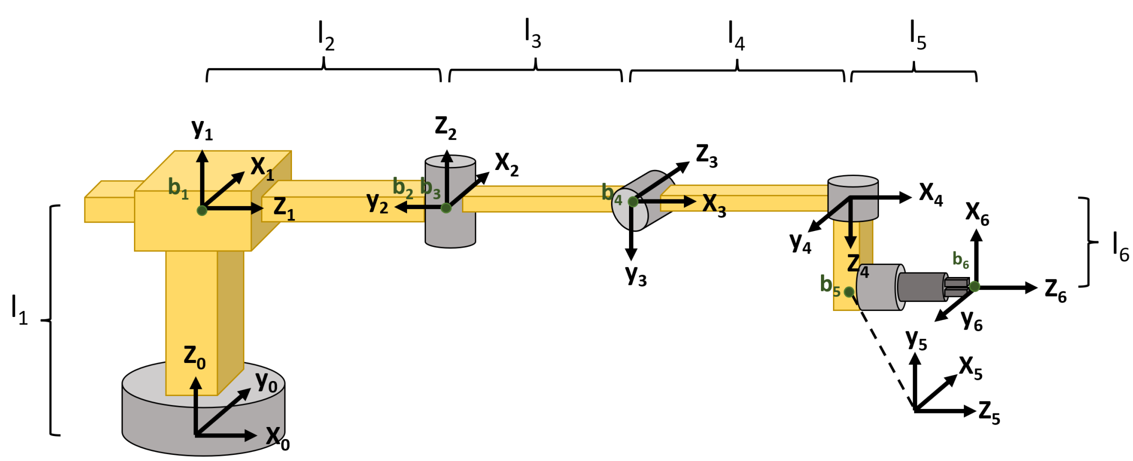


Figure 2.20: DH solution of the 6 DoF RTRRRR manipulator.

Table 2.20: DH parameters table.

DoF	θ	d	a	α
1	$q_1 = 90°$	l_1	0	$90°$
2	$0°$	$q_2 = l_2$	0	$-90°$
3	$q_3 = -90°$	0	l_3	$-90°$
4	$q_4 = 0°$	0	l_4	$-90°$
5	$q_5 = -90°$	l_6	0	$-90°$
6	$q_6 = 90°$	l_5	0	$0°$

Exercise 19 7 DoF manipulator

Use the DH algorithm to set the axes of the 7 DoF manipulator shown in Figure 2.21. Notice that all joints are angular (produce a rotation) excep the first one that is prismatic (produces a translation). Fill in Table 2.21 with the DH parameters. Dashed lines indicate the axis of rotation of each joint. Indicate the required distances to complete the DH table.

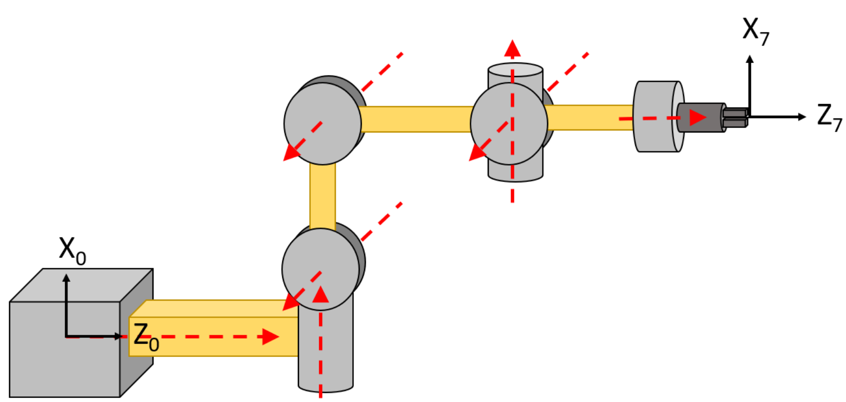


Figure 2.21: 7 DoF robot arm.

Table 2.21: DH parameters table.

DoF	θ	d	a	α
1				
2				
3				
4				
5				
6				
7				

Answer of exercise 19

The axis for the 7 DoF TRRRRRR manipulator are shown in Figure 2.22 while the DH parameters can be seen at Table 2.22.

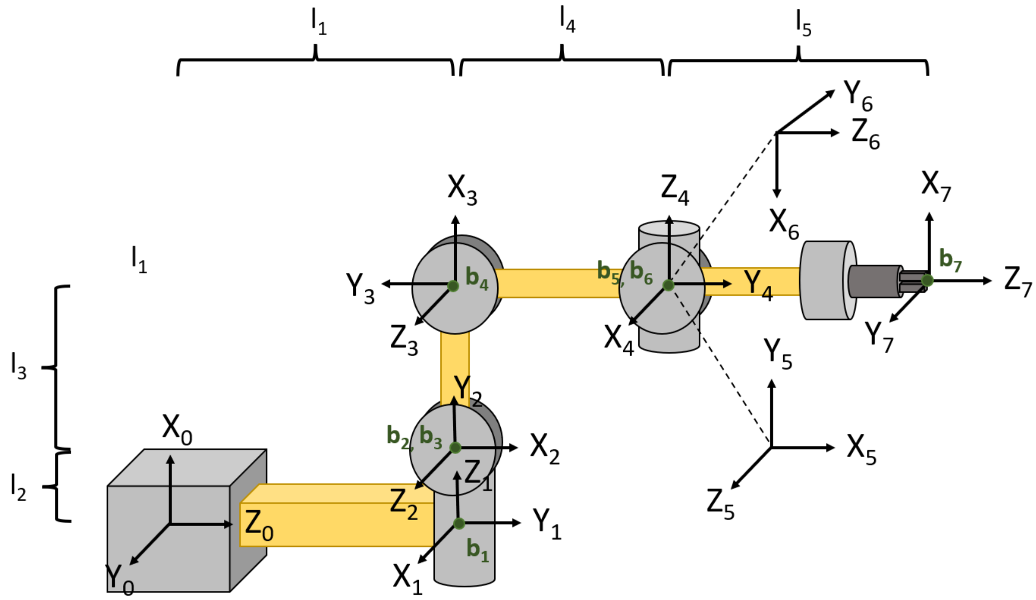


Figure 2.22: DH solution of the 7 DoF robot arm.

Table 2.22: DH parameters table.

DoF	θ	**d**	**a**	α
1	$90°$	$q_1 = l_1$	0	$90°$
2	$q_2 = 90°$	l_2	0	$90°$
3	$q_3 = 90°$	0	l_3	$0°$
4	$q_4 = -90°$	0	l_4	$-90°$
5	$q_5 = 0°$	0	0	$90°$
6	$q_6 = -90°$	0	0	$-90°$
7	$q_7 = 180°$	l_7	0	$0°$

3. Inverse Kinematics

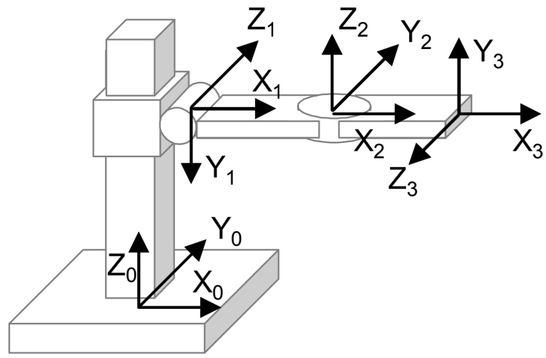


Figure 3.1: 3 DoF robot arm.

Exercise 20 TRR Robot Manipulator

Given the DH of the 3 DoF robot arm shown in Figure 3.1, and its DH table 3.1 and the corresponding ${}^{i-1}A_i$ matrices:

$$\boldsymbol{{}^0A_1} = \begin{bmatrix} 1 & 0 & 0 & a_1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & q_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \boldsymbol{{}^1A_2} = \begin{bmatrix} c_2 & 0 & s_2 & a_2 \cdot c_2 \\ s_2 & 0 & -c_2 & a_2 \cdot s_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \boldsymbol{{}^2A_3} = \begin{bmatrix} c_3 & 0 & s_3 & a_3 \cdot c_3 \\ s_3 & 0 & -c_3 & a_3 \cdot s_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

$$\boldsymbol{{}^RT_H} = \begin{bmatrix} c_2c_3 & s_2 & c_2s_3 & a_3c_3c_2 + a_2c_2 + a_1 \\ s_3 & 0 & -c_3 & a_3s_3 \\ -s_2c_3 & c_2 & -s_2s_3 & q_1 - s_2a_3c_3 - a_2s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

1. Solve the inverse kinematics in an analytical form.

Table 3.1: DH Table of the TRR Robot Manipulator.

i	θ	d	a	α	**Home**
1	0	q_1	a_1	$\frac{-\pi}{2}$	d_1
2	q_2	0	a_2	$\frac{\pi}{2}$	0
3	q_3	0	a_3	$\frac{\pi}{2}$	0

Answer of exercise 20

1. We begin equating the arm eq. (3.2) to the numeric $\boldsymbol{{}^RT_H}$:

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_2c_3 & s_2 & c_2s_3 & a_3c_3c_2 + a_2c_2 + a_1 \\ s_3 & 0 & -c_3 & a_3s_3 \\ -s_2c_3 & c_2 & -s_2s_3 & q_1 - s_2a_3c_3 - a_2s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

solving then for q_1, q_2 and q_3:

$$\left.\begin{array}{l} o_x = s_2 \\ o_z = c_2 \end{array}\right\} \Rightarrow q_2 = atan2(o_x, o_z) \tag{3.4}$$

$$\left.\begin{array}{l} n_y = s_3 \\ a_y = -c_3 \end{array}\right\} \Rightarrow q_3 = atan2(n_y, -a_y) \tag{3.5}$$

$$z = q_1 - s_2 a_3 c_3 - a_2 s_2 \Rightarrow q_1 = z + s_2 a_3 c_3 + a_2 s_2 \tag{3.6}$$

Exercise 21 RTT robot Manipulator

Given the DH of the RTT robot manipulator (Table 3.2) and its corresponding corresponding arm equation given by (3.7).

$$\boldsymbol{{}^{R}T_{H}} = \begin{bmatrix} s_1 & -c_1 & 0 & d_3 c_1 - d_2 s_1 + d_1 c_1 \\ -c_1 & -s_1 & 0 & d_3 s_1 + d_2 c_1 + d_1 s_1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

1. Solve the inverse kinematic problem using either the geometric, or the analytical method.

Table 3.2: DH Table of the RTT robot arm.

i	θ	d	a	α	**Home**
1	q_1	0	a_1	$\frac{-\pi}{2}$	0
2	$\frac{-\pi}{2}$	q_2	0	$\frac{-\pi}{2}$	100
3	$\frac{-\pi}{2}$	q_3	0	$\frac{-\pi}{2}$	150

Answer of exercise 21

We begin by equating the numerical $\boldsymbol{{}^{R}T_{H}}$ matrix to the arm equation given in (3.7):

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_1 & -c_1 & 0 & d_3 c_1 - d_2 s_1 + a_1 c_1 \\ -c_1 & -s_1 & 0 & d_3 s_1 + d_2 c_1 + a_1 s_1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

We begin solving for q_1 as follows:

$$q_1 = atan2(n_x, -n_y) \tag{3.9}$$

To solve for q_2 we can combine the equations corresponding to the x and y position of the robot:

$$\left.\begin{array}{l} x = d_3 c_1 - d_2 s_1 + a_1 c_1 \\ y = d_3 s_1 + d_2 c_1 + a_1 s_1 \end{array}\right\} \Rightarrow \left.\begin{array}{l} x = (d_3 + a_1) c_1 - d_2 s_1 \\ y = (d_3 + a_1) s_1 + d_2 c_1 \end{array}\right\} \Rightarrow \begin{array}{l} s_1 x = (d_3 + a_1) c_1 s_1 - d_2 s_1^2 \\ c_1 y = (d_3 + a_1) s_1 c_1 + d_2 c_1^2 \end{array} \tag{3.10}$$

Now, substracting the first equation from the second we get:

$$c_1y - s_1x = d_2(s_1^2 + c_1^2) \rightarrow c_1y - s_1x = d_2 \rightarrow c_1y - s_1x = q_2 \tag{3.11}$$

In a similar way it is possible to compute q_3 as follows:

$$\left.\begin{aligned} x &= (d_3 + a_1)c_1 - d_2s_1 \\ y &= (d_3 + a_1)s_1 + d_2c_1 \end{aligned}\right\} \Rightarrow \begin{aligned} c_1x &= (d_3 + a_1)c_1^2 - d_2s_1c_1 \\ s_1y &= (d_3 + a_1)s_1^2 + d_2c_1s_1 \end{aligned} \tag{3.12}$$

now adding both equations we get:

$$c_1x + s_1y = (d_3 + a_1)(c_1^2 + s_1^2) \rightarrow c_1x + s_1y = d_3 + a_1 \rightarrow c_1x + s_1y - a_1 = q_3 \tag{3.13}$$

Exercise 22 RTR robot Manipulator

Given the arm equation of the RTR robot manipulator, solve the inverse kinematic problem using the analytical method.

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & p_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{13} & 0 & s_{13} & a_3c_{13} + s_1q_2 \\ s_{13} & 0 & -c_{13} & a_3s_{13} - c_1q_2 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

Answer of exercise 22

We can begin solving for q_{13} combining he n_x and n_y elements of the ${}^RT_H$ matrix:

$$q_{13} = atan2(n_y, n_x) \tag{3.15}$$

Now we can use the x and y equations:

$$\begin{aligned} x &= a_3c_{13} + s_1q_2 \\ y &= a_3s_{13} - c_1q_2 \end{aligned} \tag{3.16}$$

and leave in the left hand of the equation the known terms:

$$\begin{aligned} x - a_3c_{13} &= s_1q_2 \\ -y + a_3s_{13} &= c_1q_2 \end{aligned} \tag{3.17}$$

now squaring both equations and adding both them together:

$$\begin{array}{lrl} & (x - a_3c_{13})^2 & = s_1^2q_2^2 \\ + & (-y + a_3s_{13})^2 & = c_1^2q_2^2 \\ \hline & x^2 + y^2 + a_3^2(s_{13}^2 + c_{13}^2) - 2xa_3c_{13} - 2ya_3s_{13} & = (s_1^2 + c_1^2)q_2^2 \end{array} \tag{3.18}$$

which simplifies to:

$$x^2 + y^2 + a_3^2 - 2xa_3c_{13} - 2ya_3s_{13} = q_2^2 \tag{3.19}$$

so taking the root square, we can solve for q_2:

$$q_2 = \sqrt{x^2 + y^2 + a_3^2 - 2xa_3c_{13} - 2ya_3s_{13}}. \tag{3.20}$$

Now, to solve for q_1, let us begin with eq. (3.17) and solve for s_1 and c_1 assuming that $q_2 > 0$:

$$\begin{aligned} \frac{x-a_3c_{13}}{q_2} &= s_1 \\ \frac{-y+a_3s_{13}}{q_2} &= c_1 \end{aligned} \tag{3.21}$$

Then, we can solve for q_1 first:

$$q_1 = atan2\left(\frac{x-a_3c_{13}}{q_2}, \frac{-y+a_3s_{13}}{q_2}\right) = atan2(x-a_3c_{13}, -y+a_3s_{13}) \tag{3.22}$$

and then for q_3:

$$q_3 = q_{13} - q_1 \tag{3.23}$$

Exercise 23 RTR robot Manipulator (2)

Given the arm equation of the RTR robot manipulator, solve the inverse kinematic problem using the analytical method.

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & p_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -s_1s_3 & -s_1c_3 & c_1 & -s_1q_2 - s_1a_3s_3 \\ c_1s_3 & c_1c_3 & s_1 & c_1q_2 + c_1a_3s_3 \\ ? & ? & 0 & d_1 - a_2 - a_3c_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.24}$$

Answer of exercise 23

We begin solving for q_1 combining he a_x and a_y elements of the ${}^RT_H$ matrix:

$$q_1 = atan2(a_y, a_x) \tag{3.25}$$

Now to obtain the expression of s_3, we begin from:

$$\begin{aligned} -n_x &= s_1s_3 \\ n_y &= c_1s_3 \end{aligned} \tag{3.26}$$

since s_1 are known c_1, we can combine the 2 previous equations as follows:

$$\begin{array}{lrl} & -n_xs_1 & = s_1^2s_3 \\ + & n_yc_1 & = c_1^2s_3 \\ \hline & n_yc_1 - n_xs_1 & = s_3(s_1^2+c_1^2) = s_3 \end{array} \tag{3.27}$$

It is also possible to solve for c_3 in a similar way by combining o_x and o_y:

$$\begin{aligned} -o_x &= s_1c_3 \\ o_y &= c_1c_3 \end{aligned} \tag{3.28}$$

$$\begin{array}{lrl} & -o_xs_1 & = s_1^2c_3 \\ + & o_yc_1 & = c_1^2c_3 \\ \hline & o_yc_1 - o_xs_1 & = c_3(s_1^2+c_1^2) = c_3 \end{array} \tag{3.29}$$

allowing to solve for q_3:

$$q_3 = atan2(n_y c_1 - n_x s_1, o_y c_1 - o_x s_1) \tag{3.30}$$

To solve for q_2 we begin with the x and y equations:

$$\begin{aligned} x &= -s_1 q_2 - s_1 a_3 s_3 \\ y &= c_1 q_2 + c_1 a_3 s_3 \end{aligned} \tag{3.31}$$

we isolate in the left hand side of the equations the known terms:

$$\begin{aligned} -x - s_1 a_3 s_3 &= s_1 q_2 \\ y - c_1 a_3 s_3 &= c_1 q_2 \end{aligned} \tag{3.32}$$

and then multiply both sides by s_1 and c_1 respectively, and add the equations to get ride of s_1 and c_1:

$$\begin{array}{lrl} & -xs_1 - s_1^2 a_3 s_3 & = s_1^2 q_2 \\ + & yc_1 - c_1^2 a_3 s_3 & = c_1^2 q_2 \\ \hline & yc_1 - xs_1 - a_3 s_3 (s_1^2 + c_1^2) & = (s_1^2 + c_1^2) q_2 \end{array} \tag{3.33}$$

which simplifies to:

$$q_2 = yc_1 - xs_1 - a_3 s_3. \tag{3.34}$$

Exercise 24 Spherical Wrist

Given the arm equation of the spherical wrist, solve the inverse kinematic problem using the analytical method.

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & p_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & s_4s_5d_6 \\ -s_5c_6 & s_5s_6 & c_5 & c_5d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.35}$$

Answer of exercise 24

- If $a_x \neq 0$ and $a_y \neq 0$ (meaning $s_5 \neq 0$, meaning q_5 is not 0 neither π) then
  - we can solve for q_4 and q_6 easily:

$$\begin{aligned} q_4 &= atan2(a_y, a_x) \\ q_6 &= atan2(o_z, -n_z). \end{aligned} \tag{3.36}$$

To solve for q_5 we can combine the a_x, a_y and a_z components:

$$\left.\begin{aligned} a_x &= c_4 s_5 \\ a_y &= s_4 s_5 \end{aligned}\right\} \Rightarrow \left.\begin{array}{lrl} & c_4 a_x & = c_4^2 s_5 \\ + & s_4 a_y & = s_4^2 s_5 \\ \hline & c_4 a_x + s_4 a_y & = (c_4^2 + s_4^2) s_5 \end{array}\right\} \tag{3.37}$$

$$\left.\begin{aligned} s_5 &= c_4 a_z + s_4 a_y \\ c_5 &= a_z \end{aligned}\right\} \Rightarrow q_5 = atan2(c_4 a_z + s_4 a_y, a_z) \tag{3.38}$$

- Otherwise (i.e., is $s_5 = 0$)
  - $a_x = a_y = 0 \Rightarrow s_5 = 0$, meaning that $q_5 = 0$ or $q_5 = \pi$. In this case, since $a_z = c_5$, if $a_z = c_5 = 1$ then $q_5 = 0$ and if $a_z = c_5 = -1$, then $q_5 = \pi$.
  - When $q_5 = 0$, eq. (3.35) simplifies to:

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_4c_6 - s_4s_6 & -c_4s_6 - s_4c_6 & 0 & 0 \\ s_4c_6 + c_4s_6 & -s_4s_6 + c_4c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{46} & -s_{46} & 0 & 0 \\ s_{46} & c_{46} & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.39}$$

In this situation, we can solve for $q_{46} = atan2(n_y, n_x)$. It is not possible to solve for q_4 and q_5 independently. This happens because $q_5 = 0$ defines a singularity so, rotating q_4 or q_6 alone, generates the same motion of the end effector frame.

  - In case $q_5 = \pi$, then eq. (3.35) simplifies to:

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -c_4c_6 - s_4s_6 & c_4s_6 - s_4c_6 & 0 & 0 \\ -s_4c_6 + c_4s_6 & s_4s_6 + c_4c_6 & 0 & 0 \\ 0 & 0 & -1 & -d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -c_{4-6} & -s_{4-6} & 0 & 0 \\ -s_{4-6} & c_{4-6} & 0 & 0 \\ 0 & 0 & 1 & -d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.40}$$

In this situation, we can solve for $q_{46} = atan2(-o_x, o_y)$. It is not possible to solve for q_4 and q_6 independently. This happens because $q_5 = \pi$ defines a singularity so, rotating q_4 or q_6 alone, generates the same motion of the end effector frame.

Exercise 25 RRR robot Manipulator

Given arm equation (eq. (3.41)) of the 3 DoF RRR robot manipulator, solve the inverse kinematic problem:

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -c_1c_{23} & s_1 & c_1s_{23} & Cc_1c_{23} + Bc_1c_2 \\ -s_1c_{23} & -c_1 & s_1s_{23} & Cs_1c_{23} + Bs_1c_2 \\ s_{23} & 0 & -c_{23} & Cs_{23} + Bs_2 + A \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.41}$$

Answer of exercise 25

We begin solving for q_1 by combining o_x and o_y:

$$q_1 = atan2(o_x, -o_y) \tag{3.42}$$

Then, we can solver for q_{23}:

$$q_{23} = atan2(n_z, -a_z). \tag{3.43}$$

Now, we take the x and y equations:

$$\begin{aligned} x &= Cc_1c_{23} + Bc_1c_2 \\ y &= Cs_1c_{23} + Bs_1c_2 \end{aligned} \tag{3.44}$$

and multiply both sides of the first and second equation by c_1 ad s_1 respectively, and add both them together:

$$\begin{array}{rrl} & xc_1 & = Cc_1^2c_{23} + Bc_1^2c_2 \\ + & ys_1 & = Cs_1^2c_{23} + Bs_1^2c_2 \\ \hline & xc_1 + ys_1 & = C(s_1^2 + c_1^2)c_{23} + Bc_2(s_1^2 + c_1^2) \\ & & = Cc_{23} + Bc_2 \end{array} \tag{3.45}$$

Now, let us define two new known variable $b_1 = xc_1 + ys_1$ and $b_2 = z - A$, so we can define a new system of equations:

$$\begin{aligned} b_1 &= Cc_{23} + Bc_2 \\ b_2 &= Cs_{23} + Bs_2 \end{aligned} \tag{3.46}$$

If we square and add both equations we get:

$$\begin{array}{rrl} & b_1^2 & = (Cc_{23} + Bc_2)^2 = C^2c_{23}^2 + B^2c_2^2 + 2BCc_2c_{23} \\ + & b_2^2 & = (Cs_{23} + Bs_2)^2 = C^2s_{23}^2 + B^2s_2^2 + 2BCs_2s_{23} \\ \hline & b_1^2 + b_2^2 & = B^2(s_2^2 + c_2^2) + C^2(s_{23}^2 + c_{23}^2) + 2BC(c_2c_{23} + s_2s_{23}) \\ & & = B^2 + C^2 + 2BC(c_2c_{23} + s_2s_{23}) \end{array} \tag{3.47}$$

Since $c(\alpha - \beta) = c(\alpha)c(\beta) + s(\alpha)s(\beta)$, the equation simplifies to:

$$b_1^2 + b_2^2 = B^2 + C^2 + 2BCc_3, \tag{3.48}$$

so it is possible to solve for c_3 as follows:

$$q_3 = cos^{-1}\left(\frac{b_1^2 + b_2^2 - B^2 - C^2}{2BC}\right) \tag{3.49}$$

Finally we can compute $q_2 = q_{23} - q_3$.

Exercise 26 RTR robot Manipulator (3)

Given the arm equation of the RTR robot manipulator, solve the inverse kinematic problem using the analytical method.

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & p_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{13} & 0 & s_{13} & l_4c_{13} + l_2c_1 + q_2s_1 \\ s_{13} & 0 & -c_{13} & l_4s_{13} + l_2s_1 - q_2c_1 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.50}$$

Answer of exercise 26

We begin solving for q_{13} combining the n_x and n_y elements of the ${}^RT_H$ matrix:

$$q_{13} = atan2(n_y, n_x) \tag{3.51}$$

Now we can use the x and y equations:

$$\begin{aligned} x &= l_4c_{13} + l_2c_1 + q_2s_1 \\ y &= l_4s_{13} + l_2s_1 - q_2c_1 \end{aligned} \tag{3.52}$$

and leave in the left had of the equation the known terms:

$$\begin{aligned} x - l_4 c_{13} &= l_2 c_1 + q_2 s_1 \\ y - l_4 s_{13} &= l_2 s_1 - q_2 c_1 \end{aligned} \tag{3.53}$$

to simplify, lets say that $b_1 = x - l_4 c_{13}$ and $b_2 = y - l_4 s_{13}$. Now, let us square both equations:

$$\begin{array}{rrl} & b_1^2 & = l_2^2 c_1^2 + q_2^2 s_1^2 + 2 l_2 q_2 s_1 c_1 \\ + & b_2^2 & = l_2^2 s_1^2 + 2 q_2^2 c_1^2 - 2 L_2 q_2 s_1 c_1 \\ \hline & b_1^2 + b_2^2 & = l_2^2 (s_1^2 + c_1^2) + q_2^2 (s_1^2 + c_1^2) \end{array} \tag{3.54}$$

$$q_2^2 = b_1^2 + b_2^2 - l_2^2 \Rightarrow q_2 = \sqrt{b_1^2 + b_2^2 - l_2^2} \tag{3.55}$$

Now that q_2 is known, the eq. (3.53) has become a linear system with 2 equations and 2 unknowns, so we can solve first for s_1:

$$\begin{array}{rll} & q_2 b_1 & = q_2 l_2 c_1 + q_2^2 s_1 \\ + & l_2 b_2 & = l_2^2 s_1 - l_2 q_2^2 c_1 \\ \hline & q_2 b_1 + l_2 b_2 & = (l_2^2 + q_2^2) s_1 \end{array} \tag{3.56}$$

$$s_1 = \frac{q_2 b_1 + l_2 b_2}{l_2^2 + q_2^2} \tag{3.57}$$

and then for c_1:

$$\begin{array}{rll} & l_2 b_1 & = l_2^2 c_1 + l_2 q_2 s_1 \\ - & q_2 b_2 & = q_2 l_2 s_1 - q_2^2 c_1 \\ \hline & l_2 b_1 + q_2 b_2 & = (l_2^2 + q_2^2) c_1 \end{array} \tag{3.58}$$

$$c_1 = \frac{l_2 b_1 + q_2 b_2}{l_2^2 + q_2^2}. \tag{3.59}$$

Finally, we can use eq. (3.57) and eq. (3.59), to obtain q_1:

$$q_1 = atan2(q_2 b_1 + l_2 b_2, l_2 b_1 + q_2 b_2) \tag{3.60}$$

4. Jacobians

DOF	θ_i	d_i	a_i	α_i	Home
1	θ_1	d_1	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$
2	$\frac{\pi}{2}$	d_2	0	0	d_2

Table 4.1: Dh of the RP robot manipulator

Exercise 27 RP Robot Manipulator Jacobian

Given a RP robot manipulator, defined by the DH table 4.1, which has he following link transformations matrices:

$$\begin{aligned} {}^0A_1 &= \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^1A_2 &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.1}$$

whose sequential products are given by:

$$\begin{aligned} {}^0A_1 &= \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^0A_2 &= \begin{bmatrix} 0 & -c_1 & s_1 & d_2\,s_1 \\ 0 & -s_1 & -c_1 & -d_2\,c_1 \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.2}$$

compute the geometric Jacobian matrix.

Answer of exercise 28

For each DOF, we compute the corresponding column of the Jacobian using the velocity propagation equation.
The joint q_1 is angular, so its column Jacobian is given by:

$$J_1 = \begin{bmatrix} {}^0z_0 \times \left({}^0p_2 - {}^0p_0\right) \\ {}^0z_0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} d_2\,s_1 \\ -d_2\,c_1 \\ d_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} d_2\,c_1 \\ d_2\,s_1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.3}$$

The joint q_2 is prismatic so its column Jacobian is given by:

$$J_2 = \begin{bmatrix} {}^0z_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.4}$$

Finally,we can formulate the robot Jacobian by concatenating the column Jacobians:

$$J(q) = \begin{bmatrix} d_2\,c_1 & s_1 \\ d_2\,s_1 & -c_1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \tag{4.5}$$

Exercise 28 Derivative-RP Robot Manipulator Jacobian

Given the arm equation of the RP robot manipulator:

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & p_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -c_1 & s_1 & d_2\,s_1 \\ 0 & -s_1 & -c_1 & -d_2\,c_1 \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

Compute the Robot Jacobian using the method of the partial derivatives.

Answer of exercise 28

Let us begin computing $J_v(q)$ through the partial derivatives of the end effector position $[x\ y\ z]^T$:

$$Jv(q) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \cdots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \cdots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \cdots & \frac{\partial z}{\partial q_n} \end{bmatrix} = \begin{bmatrix} d_2\,c_1 & s_1 \\ d_2\,s_1 & -c_1 \\ 0 & 0 \end{bmatrix} \tag{4.7}$$

To compute $J_\omega(q)$, where have to compute first the angular velocity tensor $\Omega = \dot{R}R^T$:

$$\dot{R} = \begin{bmatrix} 0 & \dot{q}_1\,s_1 & \dot{q}_1\,c_1 \\ 0 & -\dot{q}_1\,c_1 & \dot{q}_1\,s_1 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.8}$$

$$R^T = \begin{bmatrix} 0 & 0 & 1 \\ -c_1 & -s_1 & 0 \\ s_1 & -c_1 & 0 \end{bmatrix} \tag{4.9}$$

$$\Omega = \dot{R}R^T = \begin{bmatrix} 0 & -\dot{q}_1 & 0 \\ \dot{q}_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.10}$$

Ω relates to the angular velocity vector $\omega = [\omega_x\ \omega_y\ \omega_z]^T$ as follows:

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\dot{q}_1 & 0 \\ \dot{q}_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.11}$$

Therefore

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 \end{bmatrix}, \tag{4.12}$$

which can be rewritten as:

$$\omega = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}, \tag{4.13}$$

being

$$J_\omega(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \tag{4.14}$$

the Jacobian of the angular velocity. Now, we can stack the linear and angular velocity Jacobians to form the geometric Jacobian of the robot manipulator:

$$J(q) = \begin{bmatrix} J_v(q) \\ J_\omega(q) \end{bmatrix} = \begin{bmatrix} d_2 c_1 & s_1 \\ d_2 s_1 & -c_1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \tag{4.15}$$

Exercise 29 RR Robot Manipulator Jacobian

Given a RR robot manipulator, defined by the DH table 4.2, which has he following link transformations matrices:

$$\begin{aligned} {}^0A_1 &= \begin{bmatrix} c_1 & -s_1 & 0 & a_1 c_1 \\ s_1 & c_1 & 0 & a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^1A_2 &= \begin{bmatrix} c_2 & 0 & s_2 & a_2 c_2 \\ s_2 & 0 & -c_2 & a_2 s_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.16}$$

DOF	θ_i	d_i	a_i	α_i	Home
1	θ_1	0	a_1	0	0
2	θ_2	0	a_2	$\frac{\pi}{2}$	0

Table 4.2: Dh of the RR robot manipulator

whose sequential products are given by:

$$\begin{aligned} {}^0A_1 &= \begin{bmatrix} c_1 & -s_1 & 0 & a_1\,c_1 \\ s_1 & c_1 & 0 & a_1\,s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^0A_2 &= \begin{bmatrix} c_{1+2} & 0 & s_{1+2} & a_2\,c_{1+2}+a_1\,c_1 \\ s_{1+2} & 0 & -c_{1+2} & a_2\,s_{1+2}+a_1\,s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{4.17}$$

compute the geometric Jacobian matrix.

Answer of exercise 30

For each DOF, we compute the corresponding column of the Jacobian using the velocity propagation equation.
The joint q_1 is angular, so its column Jacobian is given by:

$$J_1 = \begin{bmatrix} {}^0z_0 \times \left({}^0p_2 - {}^0p_0\right) \\ {}^0z_0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} a_2\,c_{1+2}+a_1\,c_1 \\ a_2\,s_{1+2}+a_1\,s_1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -a_2\,s_{1+2}-a_1\,s_1 \\ a_2\,c_{1+2}+a_1\,c_1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.18}$$

The joint q_2 is angular, so its column Jacobian is given by:

$$J_2 = \begin{bmatrix} {}^0z_1 \times \left({}^0p_2 - {}^0p_1\right) \\ {}^0z_1 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} a_2\,c_{1+2}+a_1\,c_1 \\ a_2\,s_{1+2}+a_1\,s_1 \\ 0 \end{bmatrix} - \begin{bmatrix} a_1\,c_1 \\ a_1\,s_1 \\ 0 \end{bmatrix} \right) \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -a_2\,s_{1+2} \\ a_2\,c_{1+2} \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.19}$$

Finally,we can formulate the robot Jacobian by concatenating the column Jacobians:

$$J(q) = \begin{bmatrix} -a_2\,s_{1+2}-a_1\,s_1 & -a_2\,s_{1+2} \\ a_2\,c_{1+2}+a_1\,c_1 & a_2\,c_{1+2} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \tag{4.20}$$

Exercise 30 Derivative-RR Robot Manipulator Jacobian

Given the arm equation of the RR robot manipulator:

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & p_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{1+2} & 0 & s_{1+2} & a_2\, c_{1+2} + a_1\, c_1 \\ s_{1+2} & 0 & -c_{1+2} & a_2\, s_{1+2} + a_1\, s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.21}$$

Compute the Robot Jacobian using the method of the partial derivatives.

Answer of exercise 30

Let us begin computing $J_v(q)$ through the partial derivatives of the end effector position $[x\ y\ z]^T$:

$$Jv(q) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \dots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \dots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \dots & \frac{\partial z}{\partial q_n} \end{bmatrix} = \begin{bmatrix} -a_2\, s_{1+2} - a_1\, s_1 & -a_2\, s_{1+2} \\ a_2\, c_{1+2} + a_1\, c_1 & a_2\, c_{1+2} \\ 0 & 0 \end{bmatrix} \tag{4.22}$$

To compute $J_\omega(q)$, where have to compute first the angular velocity tensor $\Omega = \dot{R}R^T$:

$$\dot{R} = \begin{bmatrix} -\dot{q}_1\, s_{1+2} - \dot{q}_2\, s_{1+2} & 0 & \dot{q}_1\, c_{1+2} + \dot{q}_2\, c_{1+2} \\ \dot{q}_1\, c_{1+2} + \dot{q}_2\, c_{1+2} & 0 & \dot{q}_1\, s_{1+2} + \dot{q}_2\, s_{1+2} \\ 0 & 0 & 0 \end{bmatrix} \tag{4.23}$$

$$R^T = \begin{bmatrix} c_{1+2} & s_{1+2} & 0 \\ 0 & 0 & 1 \\ s_{1+2} & -c_{1+2} & 0 \end{bmatrix} \tag{4.24}$$

$$\Omega = \dot{R}R^T = \begin{bmatrix} 0 & -\dot{q}_1 - \dot{q}_2 & 0 \\ \dot{q}_1 + \dot{q}_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.25}$$

Ω relates to the angular velocity vector $\omega = [\omega_x\ \omega_y\ \omega_z]^T$ as follows:

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\dot{q}_1 - \dot{q}_2 & 0 \\ \dot{q}_1 + \dot{q}_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.26}$$

Therefore

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 + \dot{q}_2 \end{bmatrix}, \tag{4.27}$$

which can be rewritten as:

$$\omega = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}, \tag{4.28}$$

being

$$J_\omega(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \tag{4.29}$$

the Jacobian of the angular velocity. Now, we can stack the linear and angular velocity Jacobians to form the geometric Jacobian of the robot manipulator:

$$J(q) = \begin{bmatrix} J_v(q) \\ J_\omega(q) \end{bmatrix} = \begin{bmatrix} -a_2\, s_{1+2} - a_1\, s_1 & -a_2\, s_{1+2} \\ a_2\, c_{1+2} + a_1\, c_1 & a_2\, c_{1+2} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \tag{4.30}$$

Exercise 31 Scara Robot Manipulator Jacobian

Given a Scara robot manipulator, defined by the DH table 4.3, which has he following link transformations matrices:

$$\begin{aligned}
{}^0A_1 &= \begin{bmatrix} c_1 & -s_1 & 0 & a_1\, c_1 \\ s_1 & c_1 & 0 & a_1\, s_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^1A_2 &= \begin{bmatrix} c_2 & s_2 & 0 & a_2\, c_2 \\ s_2 & -c_2 & 0 & a_2\, s_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^2A_3 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^3A_4 &= \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} , \tag{4.31}$$

DOF	θ_i	d_i	a_i	α_i	Home
1	θ_1	d_1	a_1	0	0
2	θ_2	0	a_2	π	0
3	0	d_3	0	0	d_3
4	θ_4	d_4	0	0	0

Table 4.3: Dh of the Scara robot manipulator

whose sequential products are given by:

$$
\begin{aligned}
{}^0A_1 &= \begin{bmatrix} c_1 & -s_1 & 0 & a_1\,c_1 \\ s_1 & c_1 & 0 & a_1\,s_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^0A_2 &= \begin{bmatrix} c_{1+2} & s_{1+2} & 0 & a_2\,c_{1+2}+a_1\,c_1 \\ s_{1+2} & -c_{1+2} & 0 & a_2\,s_{1+2}+a_1\,s_1 \\ 0 & 0 & -1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^0A_3 &= \begin{bmatrix} c_{1+2} & s_{1+2} & 0 & a_2\,c_{1+2}+a_1\,c_1 \\ s_{1+2} & -c_{1+2} & 0 & a_2\,s_{1+2}+a_1\,s_1 \\ 0 & 0 & -1 & d_1-d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^0A_4 &= \begin{bmatrix} c_{1+2-4} & s_{1+2-4} & 0 & a_2\,c_{1+2}+a_1\,c_1 \\ s_{1+2-4} & -c_{1+2-4} & 0 & a_2\,s_{1+2}+a_1\,s_1 \\ 0 & 0 & -1 & d_1-d_3-d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
, \tag{4.32}
$$

compute the geometric Jacobian matrix.

Answer of exercise 32

For each DOF, we compute the corresponding column of the Jacobian using the velocity propagation equation.
The joint q_1 is angular, so its column Jacobian is given by:

$$
J_1 = \begin{bmatrix} {}^0z_0 \times \left({}^0p_4 - {}^0p_0\right) \\ {}^0z_0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} a_2\,c_{1+2}+a_1\,c_1 \\ a_2\,s_{1+2}+a_1\,s_1 \\ d_1-d_3-d_4 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -a_2\,s_{1+2}-a_1\,s_1 \\ a_2\,c_{1+2}+a_1\,c_1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.33}
$$

The joint q_2 is angular, so its column Jacobian is given by:

$$J_2 = \begin{bmatrix} {}^0z_1 \times \left({}^0p_4 - {}^0p_1\right) \\ {}^0z_1 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} a_2\,c_{1+2} + a_1\,c_1 \\ a_2\,s_{1+2} + a_1\,s_1 \\ d_1 - d_3 - d_4 \end{bmatrix} - \begin{bmatrix} a_1\,c_1 \\ a_1\,s_1 \\ d_1 \end{bmatrix} \right) \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -a_2\,s_{1+2} \\ a_2\,c_{1+2} \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.34}$$

The joint q_3 is prismatic so its column Jacobian is given by:

$$J_3 = \begin{bmatrix} {}^0z_2 \\ 0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.35}$$

The joint q_4 is angular, so its column Jacobian is given by:

$$J_4 = \begin{bmatrix} {}^0z_3 \times \left({}^0p_4 - {}^0p_3\right) \\ {}^0z_3 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \times \left(\begin{bmatrix} a_2\,c_{1+2} + a_1\,c_1 \\ a_2\,s_{1+2} + a_1\,s_1 \\ d_1 - d_3 - d_4 \end{bmatrix} - \begin{bmatrix} a_2\,c_{1+2} + a_1\,c_1 \\ a_2\,s_{1+2} + a_1\,s_1 \\ d_1 - d_3 \end{bmatrix} \right) \\ \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} \tag{4.36}$$

Finally,we can formulate the robot Jacobian by concatenating the column Jacobians:

$$J(q) = \begin{bmatrix} -a_2\,s_{1+2} - a_1\,s_1 & -a_2\,s_{1+2} & 0 & 0 \\ a_2\,c_{1+2} + a_1\,c_1 & a_2\,c_{1+2} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} \tag{4.37}$$

Exercise 32 Derivative-Scara Robot Manipulator Jacobian

Given the arm equation of the Scara robot manipulator:

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & p_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{1+2-4} & s_{1+2-4} & 0 & a_2\,c_{1+2} + a_1\,c_1 \\ s_{1+2-4} & -c_{1+2-4} & 0 & a_2\,s_{1+2} + a_1\,s_1 \\ 0 & 0 & -1 & d_1 - d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.38}$$

Compute the Robot Jacobian using the method of the partial derivatives.

Answer of exercise 32

Let us begin computing $J_v(q)$ through the partial derivatives of the end effector position $[x\ y\ z]^T$:

$$Jv(q) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \dots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \dots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \dots & \frac{\partial z}{\partial q_n} \end{bmatrix} = \begin{bmatrix} -a_2\, s_{1+2} - a_1\, s_1 & -a_2\, s_{1+2} & 0 & 0 \\ a_2\, c_{1+2} + a_1\, c_1 & a_2\, c_{1+2} & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{4.39}$$

To compute $J_\omega(q)$, where have to compute first the angular velocity tensor $\Omega = \dot{R}R^T$:

$$\dot{R} = \begin{bmatrix} \dot{q}_4\, s_{1+2-4} - \dot{q}_2\, s_{1+2-4} - \dot{q}_1\, s_{1+2-4} & \dot{q}_1\, c_{1+2-4} + \dot{q}_2\, c_{1+2-4} - \dot{q}_4\, c_{1+2-4} & 0 \\ \dot{q}_1\, c_{1+2-4} + \dot{q}_2\, c_{1+2-4} - \dot{q}_4\, c_{1+2-4} & \dot{q}_1\, s_{1+2-4} + \dot{q}_2\, s_{1+2-4} - \dot{q}_4\, s_{1+2-4} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.40}$$

$$R^T = \begin{bmatrix} c_{1+2-4} & s_{1+2-4} & 0 \\ s_{1+2-4} & -c_{1+2-4} & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{4.41}$$

$$\Omega = \dot{R}R^T = \begin{bmatrix} 0 & \dot{q}_4 - \dot{q}_2 - \dot{q}_1 & 0 \\ \dot{q}_1 + \dot{q}_2 - \dot{q}_4 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.42}$$

Ω relates to the angular velocity vector $\omega = [\omega_x\ \omega_y\ \omega_z]^T$ as follows:

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & \dot{q}_4 - \dot{q}_2 - \dot{q}_1 & 0 \\ \dot{q}_1 + \dot{q}_2 - \dot{q}_4 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.43}$$

Therefore

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 + \dot{q}_2 - \dot{q}_4 \end{bmatrix}, \tag{4.44}$$

which can be rewritten as:

$$\omega = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix}, \tag{4.45}$$

being

$$J_\omega(q) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} \tag{4.46}$$

the Jacobian of the angular velocity. Now, we can stack the linear and angular velocity Jacobians to form the geometric Jacobian of the robot manipulator:

$$J(q) = \begin{bmatrix} J_v(q) \\ J_\omega(q) \end{bmatrix} = \begin{bmatrix} -a_2 s_{1+2} - a_1 s_1 & -a_2 s_{1+2} & 0 & 0 \\ a_2 c_{1+2} + a_1 c_1 & a_2 c_{1+2} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} \tag{4.47}$$

Exercise 33 Spherical Robot Manipulator Jacobian

Given a Spherical robot manipulator, defined by the DH table 4.4, which has he following link transformations matrices:

$$\begin{aligned} {}^0A_1 &= \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^1A_2 &= \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^2A_3 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.48}$$

whose sequential products are given by:

$$\begin{aligned} {}^0A_1 &= \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^0A_2 &= \begin{bmatrix} c_1 c_2 & -s_1 & c_1 s_2 & -d_2 s_1 \\ c_2 s_1 & c_1 & s_1 s_2 & d_2 c_1 \\ -s_2 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^0A_3 &= \begin{bmatrix} c_1 c_2 & -s_1 & c_1 s_2 & d_3 c_1 s_2 - d_2 s_1 \\ c_2 s_1 & c_1 & s_1 s_2 & d_2 c_1 + d_3 s_1 s_2 \\ -s_2 & 0 & c_2 & d_3 c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.49}$$

compute the geometric Jacobian matrix.

Answer of exercise 34

For each DOF, we compute the corresponding column of the Jacobian using the velocity propagation equation.

DOF	θ_i	d_i	a_i	α_i	Home
1	θ_1	0	0	$-\frac{\pi}{2}$	0
2	θ_2	d_2	0	$\frac{\pi}{2}$	0
3	0	d_3	0	0	d_3

Table 4.4: Dh of the Spherical robot manipulator

The joint q_1 is angular, so its column Jacobian is given by:

$$J_1 = \begin{bmatrix} {}^0z_0 \times ({}^0p_3 - {}^0p_0) \\ {}^0z_0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} d_3 c_1 s_2 - d_2 s_1 \\ d_2 c_1 + d_3 s_1 s_2 \\ d_3 c_2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -d_2 c_1 - d_3 s_1 s_2 \\ d_3 c_1 s_2 - d_2 s_1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.50}$$

The joint q_2 is angular, so its column Jacobian is given by:

$$J_2 = \begin{bmatrix} {}^0z_1 \times ({}^0p_3 - {}^0p_1) \\ {}^0z_1 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} -s_1 \\ c_1 \\ 0 \end{bmatrix} \times \left(\begin{bmatrix} d_3 c_1 s_2 - d_2 s_1 \\ d_2 c_1 + d_3 s_1 s_2 \\ d_3 c_2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \\ \begin{bmatrix} -s_1 \\ c_1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} d_3 c_1 c_2 \\ d_3 c_2 s_1 \\ -d_3 s_2 \\ -s_1 \\ c_1 \\ 0 \end{bmatrix} \tag{4.51}$$

The joint q_3 is prismatic so its column Jacobian is given by:

$$J_3 = \begin{bmatrix} {}^0z_2 \\ 0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} c_1 s_2 \\ s_1 s_2 \\ c_2 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} c_1 s_2 \\ s_1 s_2 \\ c_2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.52}$$

Finally,we can formulate the robot Jacobian by concatenating the column Jacobians:

$$J(q) = \begin{bmatrix} -d_2 c_1 - d_3 s_1 s_2 & d_3 c_1 c_2 & c_1 s_2 \\ d_3 c_1 s_2 - d_2 s_1 & d_3 c_2 s_1 & s_1 s_2 \\ 0 & -d_3 s_2 & c_2 \\ 0 & -s_1 & 0 \\ 0 & c_1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.53}$$

Exercise 34 Derivative-Spherical Robot Manipulator Jacobian

Given the arm equation of the Spherical robot manipulator:

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & p_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_1 c_2 & -s_1 & c_1 s_2 & d_3 c_1 s_2 - d_2 s_1 \\ c_2 s_1 & c_1 & s_1 s_2 & d_2 c_1 + d_3 s_1 s_2 \\ -s_2 & 0 & c_2 & d_3 c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.54}$$

Compute the Robot Jacobian using the method of the partial derivatives.

Answer of exercise 34

Let us begin computing $J_v(q)$ through the partial derivatives of the end effector position $[x\ y\ z]^T$:

$$Jv(q) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \dots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \dots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \dots & \frac{\partial z}{\partial q_n} \end{bmatrix} = \begin{bmatrix} -d_2 c_1 - d_3 s_1 s_2 & d_3 c_1 c_2 & c_1 s_2 \\ d_3 c_1 s_2 - d_2 s_1 & d_3 c_2 s_1 & s_1 s_2 \\ 0 & -d_3 s_2 & c_2 \end{bmatrix} \tag{4.55}$$

To compute $J_\omega(q)$, where have to compute first the angular velocity tensor $\Omega = \dot{R}R^T$:

$$\dot{R} = \begin{bmatrix} -\dot{q}_1 c_2 s_1 - \dot{q}_2 c_1 s_2 & -\dot{q}_1 c_1 & \dot{q}_2 c_1 c_2 - \dot{q}_1 s_1 s_2 \\ \dot{q}_1 c_1 c_2 - \dot{q}_2 s_1 s_2 & -\dot{q}_1 s_1 & \dot{q}_1 c_1 s_2 + \dot{q}_2 c_2 s_1 \\ -\dot{q}_2 c_2 & 0 & -\dot{q}_2 s_2 \end{bmatrix} \tag{4.56}$$

$$R^T = \begin{bmatrix} c_1 c_2 & c_2 s_1 & -s_2 \\ -s_1 & c_1 & 0 \\ c_1 s_2 & s_1 s_2 & c_2 \end{bmatrix} \tag{4.57}$$

$$\Omega = \dot{R}R^T = \begin{bmatrix} 0 & -\dot{q}_1 & \dot{q}_2 c_1 \\ \dot{q}_1 & 0 & \dot{q}_2 s_1 \\ -\dot{q}_2 c_1 & -\dot{q}_2 s_1 & 0 \end{bmatrix} \tag{4.58}$$

Ω relates to the angular velocity vector $\omega = [\omega_x\ \omega_y\ \omega_z]^T$ as follows:

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\dot{q}_1 & \dot{q}_2 c_1 \\ \dot{q}_1 & 0 & \dot{q}_2 s_1 \\ -\dot{q}_2 c_1 & -\dot{q}_2 s_1 & 0 \end{bmatrix} \tag{4.59}$$

Therefore

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} -\dot{q}_2 s_1 \\ \dot{q}_2 c_1 \\ \dot{q}_1 \end{bmatrix}, \tag{4.60}$$

which can be rewritten as:

$$\omega = \begin{bmatrix} 0 & -s_1 & 0 \\ 0 & c_1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}, \tag{4.61}$$

being

$$J_\omega(q) = \begin{bmatrix} 0 & -s_1 & 0 \\ 0 & c_1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.62}$$

the Jacobian of the angular velocity. Now, we can stack the linear and angular velocity Jacobians to form the geometric Jacobian of the robot manipulator:

$$J(q) = \begin{bmatrix} J_v(q) \\ J_\omega(q) \end{bmatrix} = \begin{bmatrix} -d_2\,c_1 - d_3\,s_1\,s_2 & d_3\,c_1\,c_2 & c_1\,s_2 \\ d_3\,c_1\,s_2 - d_2\,s_1 & d_3\,c_2\,s_1 & s_1\,s_2 \\ 0 & -d_3\,s_2 & c_2 \\ 0 & -s_1 & 0 \\ 0 & c_1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.63}$$

Exercise 35 SphericalWrist Robot Manipulator Jacobian

Given a SphericalWrist robot manipulator, defined by the DH table 4.5, which has he following link transformations matrices:

$$\begin{aligned} {}^0A_1 &= \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^1A_2 &= \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad , \\ {}^2A_3 &= \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.64}$$

whose sequential products are given by:

$$\begin{aligned} {}^0A_1 &= \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^0A_2 &= \begin{bmatrix} c_1\,c_2 & -s_1 & c_1\,s_2 & 0 \\ c_2\,s_1 & c_1 & s_1\,s_2 & 0 \\ -s_2 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad , \\ {}^0A_3 &= \begin{bmatrix} c_1\,c_2\,c_3 - s_1\,s_3 & -c_3\,s_1 - c_1\,c_2\,s_3 & c_1\,s_2 & d_3\,c_1\,s_2 \\ c_1\,s_3 + c_2\,c_3\,s_1 & c_1\,c_3 - c_2\,s_1\,s_3 & s_1\,s_2 & d_3\,s_1\,s_2 \\ -c_3\,s_2 & s_2\,s_3 & c_2 & d_3\,c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.65}$$

DOF	θ_i	d_i	a_i	α_i	Home
1	θ_1	0	0	$-\frac{\pi}{2}$	$-\frac{\pi}{2}$
2	θ_2	0	0	$\frac{\pi}{2}$	$-\frac{\pi}{2}$
3	θ_3	d_3	0	0	$\frac{\pi}{2}$

Table 4.5: Dh of the SphericalWrist robot manipulator

compute the geometric Jacobian matrix.

Answer of exercise 36

For each DOF, we compute the corresponding column of the Jacobian using the velocity propagation equation.
The joint q_1 is angular, so its column Jacobian is given by:

$$J_1 = \begin{bmatrix} {}^0z_0 \times \left({}^0p_3 - {}^0p_0\right) \\ {}^0z_0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} d_3\,c_1\,s_2 \\ d_3\,s_1\,s_2 \\ d_3\,c_2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -d_3\,s_1\,s_2 \\ d_3\,c_1\,s_2 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.66}$$

The joint q_2 is angular, so its column Jacobian is given by:

$$J_2 = \begin{bmatrix} {}^0z_1 \times \left({}^0p_3 - {}^0p_1\right) \\ {}^0z_1 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} -s_1 \\ c_1 \\ 0 \end{bmatrix} \times \left(\begin{bmatrix} d_3\,c_1\,s_2 \\ d_3\,s_1\,s_2 \\ d_3\,c_2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \\ \begin{bmatrix} -s_1 \\ c_1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} d_3\,c_1\,c_2 \\ d_3\,c_2\,s_1 \\ -d_3\,s_2 \\ -s_1 \\ c_1 \\ 0 \end{bmatrix} \tag{4.67}$$

The joint q_3 is angular, so its column Jacobian is given by:

$$J_3 = \begin{bmatrix} {}^0z_2 \times \left({}^0p_3 - {}^0p_2\right) \\ {}^0z_2 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} c_1\,s_2 \\ s_1\,s_2 \\ c_2 \end{bmatrix} \times \left(\begin{bmatrix} d_3\,c_1\,s_2 \\ d_3\,s_1\,s_2 \\ d_3\,c_2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \\ \begin{bmatrix} c_1\,s_2 \\ s_1\,s_2 \\ c_2 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ c_1\,s_2 \\ s_1\,s_2 \\ c_2 \end{bmatrix} \tag{4.68}$$

Finally,we can formulate the robot Jacobian by concatenating the column Jacobians:

$$J(q) = \begin{bmatrix} -d_3\,s_1\,s_2 & d_3\,c_1\,c_2 & 0 \\ d_3\,c_1\,s_2 & d_3\,c_2\,s_1 & 0 \\ 0 & -d_3\,s_2 & 0 \\ 0 & -s_1 & c_1\,s_2 \\ 0 & c_1 & s_1\,s_2 \\ 1 & 0 & c_2 \end{bmatrix} \tag{4.69}$$

Exercise 36 Derivative-SphericalWrist Robot Manipulator Jacobian

Given the arm equation of the SphericalWrist robot manipulator:

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & p_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_1\,c_2\,c_3 - s_1\,s_3 & -c_3\,s_1 - c_1\,c_2\,s_3 & c_1\,s_2 & d_3\,c_1\,s_2 \\ c_1\,s_3 + c_2\,c_3\,s_1 & c_1\,c_3 - c_2\,s_1\,s_3 & s_1\,s_2 & d_3\,s_1\,s_2 \\ -c_3\,s_2 & s_2\,s_3 & c_2 & d_3\,c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.70}$$

Compute the Robot Jacobian using the method of the partial derivatives.

Answer of exercise 36

Let us begin computing $J_v(q)$ through the partial derivatives of the end effector position $[x\ y\ z]^T$:

$$Jv(q) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \cdots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \cdots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \cdots & \frac{\partial z}{\partial q_n} \end{bmatrix} = \begin{bmatrix} -d_3\,s_1\,s_2 & d_3\,c_1\,c_2 & 0 \\ d_3\,c_1\,s_2 & d_3\,c_2\,s_1 & 0 \\ 0 & -d_3\,s_2 & 0 \end{bmatrix} \tag{4.71}$$

To compute $J_\omega(q)$, where have to compute first the angular velocity tensor $\Omega = \dot{R}R^T$:

$$\dot{R} = \begin{bmatrix} -\dot{q}_1\,(c_1\,s_3 + c_2\,c_3\,s_1) - \dot{q}_3\,(c_3\,s_1 + c_1\,c_2\,s_3) - \dot{q}_2\,c_1\,c_3\,s_2 & \dot{q}_3\,(s_1\,s_3 - c_1\,c_2\,c_3) - \dot{q}_1\,(c_1\,c_3 - c_2\,s_1\,s_3) + \dot{q}_2\,c_1\,s_2\,s_3 & \dot{q}_2\,c_1\,c_2 - \dot{q}_1\,s_1\,s_2 \\ \dot{q}_3\,(c_1\,c_3 - c_2\,s_1\,s_3) - \dot{q}_1\,(s_1\,s_3 - c_1\,c_2\,c_3) - \dot{q}_2\,c_3\,s_1\,s_2 & \dot{q}_2\,s_1\,s_2\,s_3 - \dot{q}_3\,(c_1\,s_3 + c_2\,c_3\,s_1) - \dot{q}_1\,(c_3\,s_1 + c_1\,c_2\,s_3) & \dot{q}_1\,c_1\,s_2 + \dot{q}_2\,c_2\,s_1 \\ \dot{q}_3\,s_2\,s_3 - \dot{q}_2\,c_2\,c_3 & \dot{q}_2\,c_2\,s_3 + \dot{q}_3\,c_3\,s_2 & -\dot{q}_2\,s_2 \end{bmatrix} \tag{4.72}$$

$$R^T = \begin{bmatrix} c_1\,c_2\,c_3 - s_1\,s_3 & c_1\,s_3 + c_2\,c_3\,s_1 & -c_3\,s_2 \\ -c_3\,s_1 - c_1\,c_2\,s_3 & c_1\,c_3 - c_2\,s_1\,s_3 & s_2\,s_3 \\ c_1\,s_2 & s_1\,s_2 & c_2 \end{bmatrix} \tag{4.73}$$

$$\Omega = \dot{R}R^T = \begin{bmatrix} 0 & -\dot{q}_1 - \dot{q}_3\,c_2 & \dot{q}_2\,c_1 + \dot{q}_3\,s_1\,s_2 \\ \dot{q}_1 + \dot{q}_3\,c_2 & 0 & \dot{q}_2\,s_1 - \dot{q}_3\,c_1\,s_2 \\ -\dot{q}_2\,c_1 - \dot{q}_3\,s_1\,s_2 & \dot{q}_3\,c_1\,s_2 - \dot{q}_2\,s_1 & 0 \end{bmatrix} \tag{4.74}$$

Ω relates to the angular velocity vector $\omega = [\omega_x\ \omega_y\ \omega_z]^T$ as follows:

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\dot{q}_1 - \dot{q}_3\,c_2 & \dot{q}_2\,c_1 + \dot{q}_3\,s_1\,s_2 \\ \dot{q}_1 + \dot{q}_3\,c_2 & 0 & \dot{q}_2\,s_1 - \dot{q}_3\,c_1\,s_2 \\ -\dot{q}_2\,c_1 - \dot{q}_3\,s_1\,s_2 & \dot{q}_3\,c_1\,s_2 - \dot{q}_2\,s_1 & 0 \end{bmatrix} \tag{4.75}$$

Therefore

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{q}_3\,c_1\,s_2 - \dot{q}_2\,s_1 \\ \dot{q}_2\,c_1 + \dot{q}_3\,s_1\,s_2 \\ \dot{q}_1 + \dot{q}_3\,c_2 \end{bmatrix}, \tag{4.76}$$

which can be rewritten as:

$$\omega = \begin{bmatrix} 0 & -s_1 & c_1\,s_2 \\ 0 & c_1 & s_1\,s_2 \\ 1 & 0 & c_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}, \tag{4.77}$$

being

$$J_{\omega}(q)=\begin{bmatrix}0 & -s_1 & c_1\,s_2\\ 0 & c_1 & s_1\,s_2\\ 1 & 0 & c_2\end{bmatrix} \tag{4.78}$$

the Jacobian of the angular velocity. Now, we can stack the linear and angular velocity Jacobians to form the geometric Jacobian of the robot manipulator:

$$J(q)=\begin{bmatrix}J_v(q)\\ J_{\omega}(q)\end{bmatrix}=\begin{bmatrix}-d_3\,s_1\,s_2 & d_3\,c_1\,c_2 & 0\\ d_3\,c_1\,s_2 & d_3\,c_2\,s_1 & 0\\ 0 & -d_3\,s_2 & 0\\ 0 & -s_1 & c_1\,s_2\\ 0 & c_1 & s_1\,s_2\\ 1 & 0 & c_2\end{bmatrix} \tag{4.79}$$

Exercise 37 Antropomorphic Robot Manipulator Jacobian

Given a Antropomorphic robot manipulator, defined by the DH table 4.6, which has he following link transformations matrices:

$$\begin{aligned}
{}^{0}A_1 &= \begin{bmatrix}c_1 & 0 & s_1 & 0\\ s_1 & 0 & -c_1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\\
{}^{1}A_2 &= \begin{bmatrix}c_2 & -s_2 & 0 & a_2\,c_2\\ s_2 & c_2 & 0 & a_2\,s_2\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},\\
{}^{2}A_3 &= \begin{bmatrix}c_3 & -s_3 & 0 & a_3\,c_3\\ s_3 & c_3 & 0 & a_3\,s_3\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}
\end{aligned} \tag{4.80}$$

whose sequential products are given by:

$$\begin{aligned}
{}^{0}A_1 &= \begin{bmatrix}c_1 & 0 & s_1 & 0\\ s_1 & 0 & -c_1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\\
{}^{0}A_2 &= \begin{bmatrix}c_1\,c_2 & -c_1\,s_2 & s_1 & a_2\,c_1\,c_2\\ c_2\,s_1 & -s_1\,s_2 & -c_1 & a_2\,c_2\,s_1\\ s_2 & c_2 & 0 & a_2\,s_2\\ 0 & 0 & 0 & 1\end{bmatrix} \qquad ,\\
{}^{0}A_3 &= \begin{bmatrix}c_{2+3}\,c_1 & -s_{2+3}\,c_1 & s_1 & c_1\,(a_3\,c_{2+3}+a_2\,c_2)\\ c_{2+3}\,s_1 & -s_{2+3}\,s_1 & -c_1 & s_1\,(a_3\,c_{2+3}+a_2\,c_2)\\ s_{2+3} & c_{2+3} & 0 & a_3\,s_{2+3}+a_2\,s_2\\ 0 & 0 & 0 & 1\end{bmatrix}
\end{aligned} \tag{4.81}$$

DOF	θ_i	d_i	a_i	α_i	Home
1	θ_1	0	0	$\frac{\pi}{2}$	0
2	θ_2	0	a_2	0	0
3	θ_3	0	a_3	0	0

Table 4.6: Dh of the Antropomorphic robot manipulator

compute the geometric Jacobian matrix.

Answer of exercise 38

For each DOF, we compute the corresponding column of the Jacobian using the velocity propagation equation.
The joint q_1 is angular, so its column Jacobian is given by:

$$J_1 = \begin{bmatrix} {}^0z_0 \times \left({}^0p_3 - {}^0p_0\right) \\ {}^0z_0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} c_1\,(a_3\,c_{2+3} + a_2\,c_2) \\ s_1\,(a_3\,c_{2+3} + a_2\,c_2) \\ a_3\,s_{2+3} + a_2\,s_2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -s_1\,(a_3\,c_{2+3} + a_2\,c_2) \\ c_1\,(a_3\,c_{2+3} + a_2\,c_2) \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.82}$$

The joint q_2 is angular, so its column Jacobian is given by:

$$J_2 = \begin{bmatrix} {}^0z_1 \times \left({}^0p_3 - {}^0p_1\right) \\ {}^0z_1 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix} \times \left(\begin{bmatrix} c_1\,(a_3\,c_{2+3} + a_2\,c_2) \\ s_1\,(a_3\,c_{2+3} + a_2\,c_2) \\ a_3\,s_{2+3} + a_2\,s_2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \\ \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -c_1\,(a_3\,s_{2+3} + a_2\,s_2) \\ -s_1\,(a_3\,s_{2+3} + a_2\,s_2) \\ a_3\,c_{2+3} + a_2\,c_2 \\ s_1 \\ -c_1 \\ 0 \end{bmatrix} \tag{4.83}$$

The joint q_3 is angular, so its column Jacobian is given by:

$$J_3 = \begin{bmatrix} {}^0z_2 \times \left({}^0p_3 - {}^0p_2\right) \\ {}^0z_2 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix} \times \left(\begin{bmatrix} c_1\,(a_3\,c_{2+3} + a_2\,c_2) \\ s_1\,(a_3\,c_{2+3} + a_2\,c_2) \\ a_3\,s_{2+3} + a_2\,s_2 \end{bmatrix} - \begin{bmatrix} a_2\,c_1\,c_2 \\ a_2\,c_2\,s_1 \\ a_2\,s_2 \end{bmatrix} \right) \\ \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -a_3\,s_{2+3}\,c_1 \\ -a_3\,s_{2+3}\,s_1 \\ a_3\,c_{2+3} \\ s_1 \\ -c_1 \\ 0 \end{bmatrix} \tag{4.84}$$

Finally,we can formulate the robot Jacobian by concatenating the column Jacobians:

$$J(q) = \begin{bmatrix} -s_1\,(a_3\,c_{2+3}+a_2\,c_2) & -c_1\,(a_3\,s_{2+3}+a_2\,s_2) & -a_3\,s_{2+3}\,c_1 \\ c_1\,(a_3\,c_{2+3}+a_2\,c_2) & -s_1\,(a_3\,s_{2+3}+a_2\,s_2) & -a_3\,s_{2+3}\,s_1 \\ 0 & a_3\,c_{2+3}+a_2\,c_2 & a_3\,c_{2+3} \\ 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.85}$$

Exercise 38 Derivative-Antropomorphic Robot Manipulator Jacobian

Given the arm equation of the Antropomorphic robot manipulator:

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & p_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{2+3}\,c_1 & -s_{2+3}\,c_1 & s_1 & c_1\,(a_3\,c_{2+3}+a_2\,c_2) \\ c_{2+3}\,s_1 & -s_{2+3}\,s_1 & -c_1 & s_1\,(a_3\,c_{2+3}+a_2\,c_2) \\ s_{2+3} & c_{2+3} & 0 & a_3\,s_{2+3}+a_2\,s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.86}$$

Compute the Robot Jacobian using the method of the partial derivatives.

Answer of exercise 38

Let us begin computing $J_v(q)$ through the partial derivatives of the end effector position $[x\ y\ z]^T$:

$$Jv(q) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \cdots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \cdots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \cdots & \frac{\partial z}{\partial q_n} \end{bmatrix} = \begin{bmatrix} -s_1\,(a_3\,c_{2+3}+a_2\,c_2) & -c_1\,(a_3\,s_{2+3}+a_2\,s_2) & -a_3\,s_{2+3}\,c_1 \\ c_1\,(a_3\,c_{2+3}+a_2\,c_2) & -s_1\,(a_3\,s_{2+3}+a_2\,s_2) & -a_3\,s_{2+3}\,s_1 \\ 0 & a_3\,c_{2+3}+a_2\,c_2 & a_3\,c_{2+3} \end{bmatrix} \tag{4.87}$$

To compute $J_\omega(q)$, where have to compute first the angular velocity tensor $\Omega = \dot{R}R^T$:

$$\dot{R} = \begin{bmatrix} -\dot{q}_1\,c_{2+3}\,s_1-\dot{q}_2\,s_{2+3}\,c_1-\dot{q}_3\,s_{2+3}\,c_1 & \dot{q}_1\,s_{2+3}\,s_1-\dot{q}_3\,c_{2+3}\,c_1-\dot{q}_2\,c_{2+3}\,c_1 & \dot{q}_1\,c_1 \\ \dot{q}_1\,c_{2+3}\,c_1-\dot{q}_2\,s_{2+3}\,s_1-\dot{q}_3\,s_{2+3}\,s_1 & -\dot{q}_1\,s_{2+3}\,c_1-\dot{q}_2\,c_{2+3}\,s_1-\dot{q}_3\,c_{2+3}\,s_1 & \dot{q}_1\,s_1 \\ \dot{q}_2\,c_{2+3}+\dot{q}_3\,c_{2+3} & -\dot{q}_2\,s_{2+3}-\dot{q}_3\,s_{2+3} & 0 \end{bmatrix} \tag{4.88}$$

$$R^T = \begin{bmatrix} c_{2+3}\,c_1 & c_{2+3}\,s_1 & s_{2+3} \\ -s_{2+3}\,c_1 & -s_{2+3}\,s_1 & c_{2+3} \\ s_1 & -c_1 & 0 \end{bmatrix} \tag{4.89}$$

$$\Omega = \dot{R}R^T = \begin{bmatrix} 0 & -\dot{q}_1 & -c_1\,(\dot{q}_2+\dot{q}_3) \\ \dot{q}_1 & 0 & -s_1\,(\dot{q}_2+\dot{q}_3) \\ c_1\,(\dot{q}_2+\dot{q}_3) & s_1\,(\dot{q}_2+\dot{q}_3) & 0 \end{bmatrix} \tag{4.90}$$

Ω relates to the angular velocity vector $\omega = [\omega_x\ \omega_y\ \omega_z]^T$ as follows:

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\dot{q}_1 & -c_1\,(\dot{q}_2+\dot{q}_3) \\ \dot{q}_1 & 0 & -s_1\,(\dot{q}_2+\dot{q}_3) \\ c_1\,(\dot{q}_2+\dot{q}_3) & s_1\,(\dot{q}_2+\dot{q}_3) & 0 \end{bmatrix} \tag{4.91}$$

Therefore

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} s_1(\dot{q}_2+\dot{q}_3) \\ -c_1(\dot{q}_2+\dot{q}_3) \\ \dot{q}_1 \end{bmatrix}, \tag{4.92}$$

which can be rewritten as:

$$\omega = \begin{bmatrix} 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}, \tag{4.93}$$

being

$$J_\omega(q) = \begin{bmatrix} 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.94}$$

the Jacobian of the angular velocity. Now, we can stack the linear and angular velocity Jacobians to form the geometric Jacobian of the robot manipulator:

$$J(q) = \begin{bmatrix} J_v(q) \\ J_\omega(q) \end{bmatrix} = \begin{bmatrix} -s_1(a_3 c_{2+3} + a_2 c_2) & -c_1(a_3 s_{2+3} + a_2 s_2) & -a_3 s_{2+3} c_1 \\ c_1(a_3 c_{2+3} + a_2 c_2) & -s_1(a_3 s_{2+3} + a_2 s_2) & -a_3 s_{2+3} s_1 \\ 0 & a_3 c_{2+3} + a_2 c_2 & a_3 c_{2+3} \\ 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.95}$$

Exercise 39 PPPR Robot Manipulator Jacobian

Given a PPPR robot manipulator, defined by the DH table 4.7, which has he following link transformations matrices:

$$\begin{aligned} {}^0A_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^1A_2 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^2A_3 &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^3A_4 &= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.96}$$

DOF	θ_i	d_i	a_i	α_i	Home
1	0	d_1	0	$-\frac{\pi}{2}$	d_1
2	$\frac{\pi}{2}$	d_2	0	$\frac{\pi}{2}$	d_2
3	$-\frac{\pi}{2}$	d_3	0	0	d_3
4	θ_1	0	0	0	0

Table 4.7: Dh of the PPPR robot manipulator

whose sequential products are given by:

$$\begin{aligned}
{}^0A_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^0A_2 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & d_2 \\ -1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^0A_3 &= \begin{bmatrix} 0 & 0 & 1 & d_3 \\ -1 & 0 & 0 & d_2 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^0A_4 &= \begin{bmatrix} 0 & 0 & 1 & d_3 \\ -c_1 & s_1 & 0 & d_2 \\ -s_1 & -c_1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}, \tag{4.97}$$

compute the geometric Jacobian matrix.

Answer of exercise 40

For each DOF, we compute the corresponding column of the Jacobian using the velocity propagation equation.
The joint q_1 is prismatic so its column Jacobian is given by:

$$J_1 = \begin{bmatrix} {}^0z_0 \\ 0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.98}$$

The joint q_2 is prismatic so its column Jacobian is given by:

$$J_2 = \begin{bmatrix} {}^0z_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.99}$$

The joint q_3 is prismatic so its column Jacobian is given by:

$$J_3 = \begin{bmatrix} {}^0z_2 \\ 0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.100}$$

The joint q_4 is angular, so its column Jacobian is given by:

$$J_4 = \begin{bmatrix} {}^0z_3 \times \left({}^0p_4 - {}^0p_3\right) \\ {}^0z_3 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \times \left(\begin{bmatrix} d_3 \\ d_2 \\ d_1 \end{bmatrix} - \begin{bmatrix} d_3 \\ d_2 \\ d_1 \end{bmatrix} \right) \\ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{4.101}$$

Finally,we can formulate the robot Jacobian by concatenating the column Jacobians:

$$J(q) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.102}$$

Exercise 40 Derivative-PPPR Robot Manipulator Jacobian

Given the arm equation of the PPPR robot manipulator:

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & p_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & d_3 \\ -c_1 & s_1 & 0 & d_2 \\ -s_1 & -c_1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.103}$$

Compute the Robot Jacobian using the method of the partial derivatives.

Answer of exercise 40

Let us begin computing $J_v(q)$ through the partial derivatives of the end effector position $[x\ y\ z]^T$:

$$Jv(q) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \dots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \dots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \dots & \frac{\partial z}{\partial q_n} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.104}$$

To compute $J_\omega(q)$, where have to compute first the angular velocity tensor $\Omega = \dot{R}R^T$:

$$\dot{R} = \begin{bmatrix} 0 & 0 & 0 \\ \dot{q}_4 s_1 & \dot{q}_4 c_1 & 0 \\ -\dot{q}_4 c_1 & \dot{q}_4 s_1 & 0 \end{bmatrix} \tag{4.105}$$

$$R^T = \begin{bmatrix} 0 & -c_1 & -s_1 \\ 0 & s_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.106}$$

$$\Omega = \dot{R}R^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\dot{q}_4 \\ 0 & \dot{q}_4 & 0 \end{bmatrix} \tag{4.107}$$

Ω relates to the angular velocity vector $\omega = [\omega_x \ \omega_y \ \omega_z]^T$ as follows:

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\dot{q}_4 \\ 0 & \dot{q}_4 & 0 \end{bmatrix} \tag{4.108}$$

Therefore

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{q}_4 \\ 0 \\ 0 \end{bmatrix}, \tag{4.109}$$

which can be rewritten as:

$$\omega = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix}, \tag{4.110}$$

being

$$J_\omega(q) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.111}$$

the Jacobian of the angular velocity. Now, we can stack the linear and angular velocity Jacobians to form the geometric Jacobian of the robot manipulator:

$$J(q) = \begin{bmatrix} J_v(q) \\ J_\omega(q) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.112}$$

Exercise 41 rPPR Robot Manipulator Jacobian

Given a rPPR robot manipulator, defined by the DH table 4.8, which has he following link transformations matrices:

$$
\begin{aligned}
{}^0A_1 &= \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^1A_2 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^2A_3 &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^3A_4 &= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
, \tag{4.113}
$$

whose sequential products are given by:

$$
\begin{aligned}
{}^0A_1 &= \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^0A_2 &= \begin{bmatrix} 0 & -s_1 & c_1 & -d_2\,s_1 \\ 0 & c_1 & s_1 & d_2\,c_1 \\ -1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^0A_3 &= \begin{bmatrix} s_1 & 0 & c_1 & d_3\,c_1 - d_2\,s_1 \\ -c_1 & 0 & s_1 & d_2\,c_1 + d_3\,s_1 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^0A_4 &= \begin{bmatrix} \frac{s_{2(1}}{)}2 & -s_1)^2 & c_1 & d_3\,c_1 - d_2\,s_1 \\ -c_1)^2 & \frac{s_{2(1}}{)}2 & s_1 & d_2\,c_1 + d_3\,s_1 \\ -s_1 & -c_1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
, \tag{4.114}
$$

compute the geometric Jacobian matrix.

Answer of exercise 41

For each DOF, we compute the corresponding column of the Jacobian using the velocity propagation equation.
The joint q_1 is angular, so its column Jacobian is given by:

DOF	θ_i	d_i	a_i	α_i	Home
1	θ_1	d_1	0	$-\frac{\pi}{2}$	d_1
2	$\frac{\pi}{2}$	d_2	0	$\frac{\pi}{2}$	d_2
3	$-\frac{\pi}{2}$	d_3	0	0	d_3
4	θ_1	0	0	0	0

Table 4.8: Dh of the rPPR robot manipulator

$$J_1 = \begin{bmatrix} {}^0z_0 \times \left({}^0p_4 - {}^0p_0\right) \\ {}^0z_0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} d_3 c_1 - d_2 s_1 \\ d_2 c_1 + d_3 s_1 \\ d_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -d_2 c_1 - d_3 s_1 \\ d_3 c_1 - d_2 s_1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.115}$$

The joint q_2 is prismatic so its column Jacobian is given by:

$$J_2 = \begin{bmatrix} {}^0z_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} -s_1 \\ c_1 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -s_1 \\ c_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.116}$$

The joint q_3 is prismatic so its column Jacobian is given by:

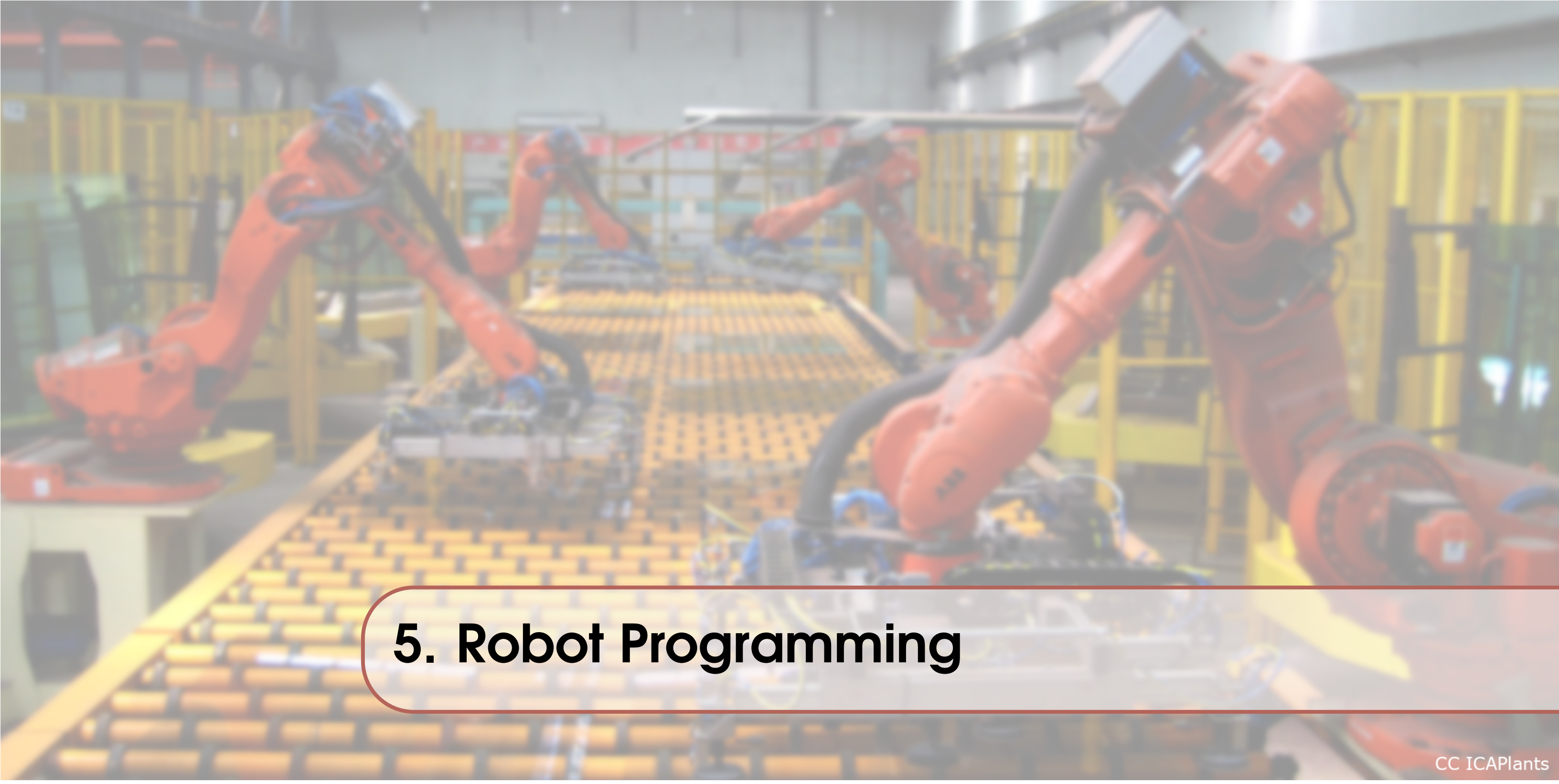
$$J_3 = \begin{bmatrix} {}^0z_2 \\ 0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} c_1 \\ s_1 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} c_1 \\ s_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.117}$$

The joint q_4 is angular, so its column Jacobian is given by:

$$J_4 = \begin{bmatrix} {}^0z_3 \times \left({}^0p_4 - {}^0p_3\right) \\ {}^0z_3 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} c_1 \\ s_1 \\ 0 \end{bmatrix} \times \left(\begin{bmatrix} d_3 c_1 - d_2 s_1 \\ d_2 c_1 + d_3 s_1 \\ d_1 \end{bmatrix} - \begin{bmatrix} d_3 c_1 - d_2 s_1 \\ d_2 c_1 + d_3 s_1 \\ d_1 \end{bmatrix} \right) \\ \begin{bmatrix} c_1 \\ s_1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ c_1 \\ s_1 \\ 0 \end{bmatrix} \tag{4.118}$$

Finally,we can formulate the robot Jacobian by concatenating the column Jacobians:

$$J(q) = \begin{bmatrix} -d_2\,c_1 - d_3\,s_1 & -s_1 & c_1 & 0 \\ d_3\,c_1 - d_2\,s_1 & c_1 & s_1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_1 \\ 0 & 0 & 0 & s_1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.119}$$



5. Robot Programming

5.1 Robot Programming Theory

Every industrial robot company use to develop their own language to program their robots. However, most of them have several similitudes. Next, we present a psudocode that, despite being inspired on the Stäubli VAL3 language, is generic enough to be used in plenty of robots.

5.1.1 Types and variables

To keep thigs easy, we assume that variables can be directly defined in the pseudocode without specifying its type. In addition to **Numeric**, **Boolean** and **String** types the following special types can be used:

- **Point**: 3D poses defined in the Cartesian space. They a vector of 6 elements containing the Cartesian position and orientation using Euler angles (i.e., $[X\ Y\ Z\ Rot_X\ Rot_Y\ Rot_Z]$).
- **Joint**: Vector containing the position of each joint. The lenght of the vector will be the same than the number of articulations the robot has.
- **Tool**: end effector mounted on a robot.
- **Speed**: to define the manipulator displacement parameters. It is assumed tyhat v_slow and v_fast speed parameters are already defined to execute delicate movements and faster movements repectively.
- **DIO**: Defines a digital input output that can be enabled or disabled as well as read.

5.1.2 Control flow

Regarding control flow structures, we assume the following are available:

- **if** condition **then** Statement [**else** Statement] **end if**
- **for** condition **do** Statement **end for**
- **while** condition **do** Statement **end while**
- **function** name ([arguments]) Statement **end function**

5.1.3 Available commands

The following functions are considered defined.

- **movej(Point, Tool, Speed)**: Move the **Tool** from its current position to **Point** at speed **Speed** interpolating joint values.
- **movel(Point, Tool, Speed)**: Move the **Tool** from its current position to **Point** at speed **Speed** following a straight line
- **waitEndMove()**: Normally motions (i.e., both **movej** and **movel**) finalize when they are close to the goal. If is is required wait until the motion finalizes completely this commands must be used.
- **appro(Point, offset)**: **offset** is a 6D vector that is added to the **Point**. The **appro** function returns a **Point** type structure.
- **enable(DIO)**: Enables the digital output DIO.
- **disable(DIO)**: Disables the digital output DIO.
- **read(DIO)**: Read the digital input DIO. It may return True or False.
- **delay(n)**: wait n seconds before continue with the execution.

To open or close the gripper, the **enable(TOOL) disable(TOOL)** functions can be used, assuming that **TOOL** is a digital output connected to the gripper actuator.

5.2 Robot Programming Exercises

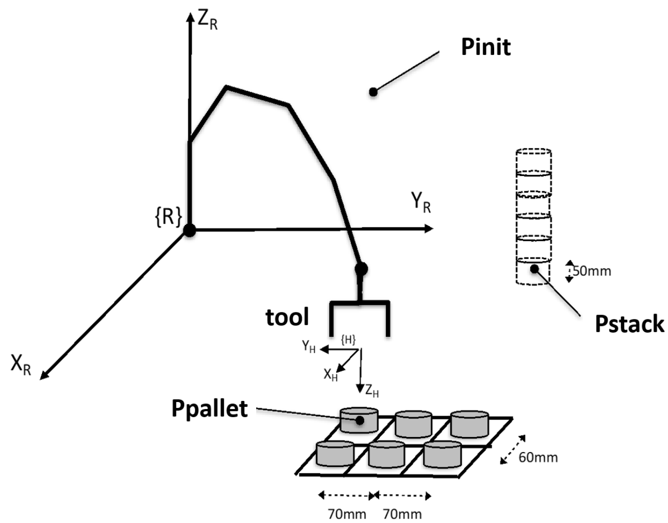
Exercise 42 Palletizer

We are asked for an application able to take 6 pieces placed on a pallet and stack them at another point. The pallet has 2 rows and 3 columns and we know the starting point of the pallet (P_{pallet}) that corresponds to the position row 0 and column 0. The distance between the positions for each row is 60mm and the distance between the positions for each column is 70mm.

We know the point P_{stack} which corresponds to the place where the first piece of the stack must be left. The second piece must be stacked on top of the first and so on until all 6 pieces are stacked. The pieces have a height of 50 mm.

Write an application in pseudocode that takes in consideration the following:

- The points P_{pallet} and P_{stack} are defined using the reference system of the tool called $tool$.
- There is an initial position called P_{init}.
- The robot base and the tool reference system as well as the pallet axis are all parallel.



[ht]

Figure 5.1: Palletizer.

Answer of exercise 42

movej(P_{init}, tool, v_slow) ▷ Go to initial position
waitEndMove()
enable(tool)
for $i \leftarrow 0, 1$ **do**
  for $j \leftarrow 0, 2$ **do**
    movej(**appro**(P_{pallet}, [60×i, -70×j, -100, 0, 0, 0]), tool, v_fast) ▷ Pick piece
    movel(**appro**(P_{pallet}, [60×i, -70×j, 0, 0, 0, 0]), tool, v_slow)
    waitEndMove()
    disable(tool)
    movel(**appro**(P_{pallet}, [60×i, -70×j, -100, 0, 0, 0]), tool, v_slow)
    movej(**appro**(P_{stack}, [0, 0, -400, 0, 0, 0]), tool, v_fast) ▷ Place piece
    movel(**appro**(P_{stack}, [0, 0, -50×(i×3 + j), 0, 0, 0]), tool, v_slow)
    waitEndMove()
    enable(tool)
    movel(**appro**(P_{stack}, [0, 0, -400, 0, 0, 0]), tool, v_slow)
  end for
end for

Exercise 43 Vision-based palletizer

An automotive group contacted us to design a palletizer located at the end of an assembly line. This robotic cell has a vision system to locate the parts placed in the terminal element. Design a program in pseudocode given the following assumptions:

- The presence sensor can be read through the DIO variable $S_{precense}$ that indicates the presence (True) of a part in the scanning area.
- For the vision system to calculate the gripping point of the part, it is necessary to take a picture of it from the P_{scan} position. To do it, it is necessary to stop the robot terminal element in this position for at least 1 second to ensure that there are no vibrations due to the motion.
- The gripping position P_{pick} is provided by the function $P_{pick} \leftarrow$ `ScanRoutine()`.
- Once the part has been picked, it must be positioned on the pallet.
- The pallet has space for 4 pieces that are 300mm apart being P_{pallet} the pose of the first space.
- The pallet is cleared automatically when it is full.
- The robot has only one tool called $gripper$.

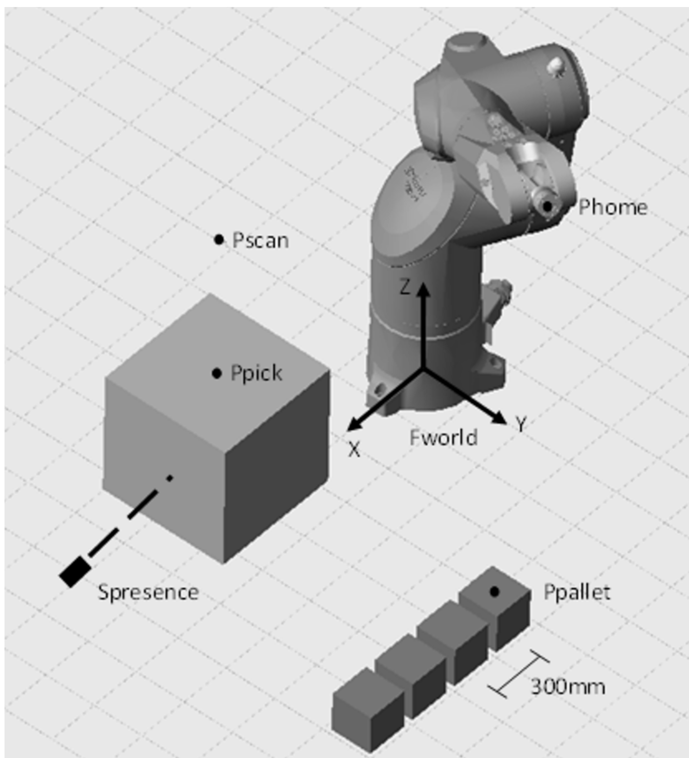
[ht] 

Figure 5.2: Vision-based palletizer.

Answer of exercise 43

```
part_counter ← 0
movej(P_home, gripper, v_fast)
while true do
    if part_counter < 4 then
        if read(S_presence) then                      ▷ Move to Scan position if there is a Piece ready
            movej(P_scan, gripper, v_fast)
            waitEndMove()
            delay(1)
            P_pick ← ScanRoutine()                    ▷ Pick Part from Scan Area
            movej(appro(P_pick, [0, 0, 100, 0, 0, 0]), gripper, v_fast)
            enable(gripper)
            movel(P_pick, gripper, v_slow)
            waitEndMove()
            disable(gripper)
            movel(appro(P_pick, [0, 0, 100, 0, 0, 0]), gripper, v_slow)
            x_offset ← part_counter × 300             ▷ Place Part in pallet
            movej(appro(P_pallet, [x_offset, 0, 100, 0, 0, 0]), gripper, v_fast)
            movel(appro(P_pallet, [x_offset, 0, 0, 0, 0, 0]), gripper, v_slow)
            waitEndMove()
            enable(gripper)
            part_counter ← part_counter + 1
            movel(appro(P_pallet, [0, 0, 100, 0, 0, 0]), gripper, v_slow)
            movej(P_scan, gripper, v_fast)            ▷ Return to Scan position
        end if
    else
        part_counter ← 0                              ▷ Pallet is full
    end if
end while
movej(P_home, gripper, v_fast)
```

Exercise 44 Automatic chess

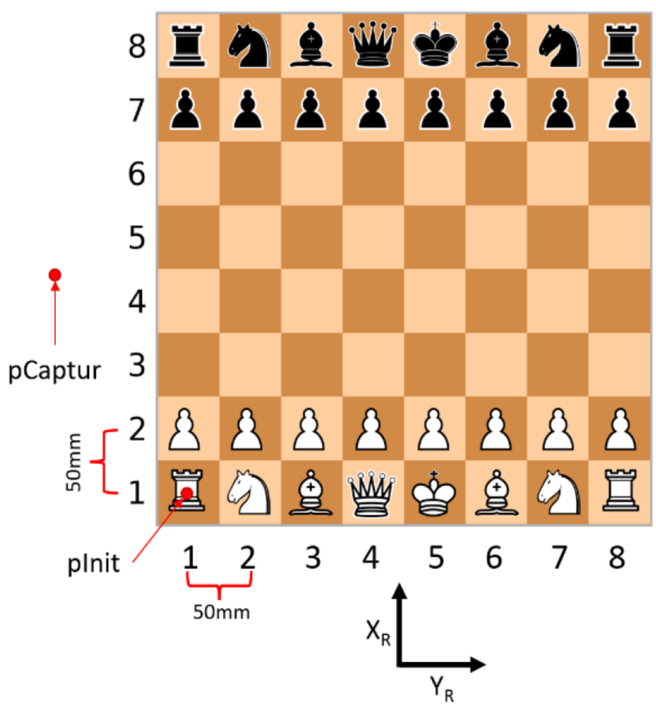
We want to program a system capable of playing chess against a person on a real two-dimensional chess board in a completely autonomous way. We have an artificial vision system capable of recognizing the position of the pieces on a chess board (2D), as well as a chess engine that, given the current state of the board, returns the best move the robot has to make.

We have to program an industrial manipulator, which has a suction cup controllabnle from the DIO *suction_cup* as a terminal element, to execute the motions determined by the chess engine. These motions can be of 2 types:

- Motion (M): move the piece from the initial position (p_i) to the final position (p_f).
- Capture (C): move the piece that is currently at the end position (p_f) to the capture position (p_{Captur}) and then move the piece that is at p_i to p_f.

A motion is described by 3 variables: t, p_i and p_f. Where t can be *'M'* or *'C'* and both p_i and p_f are defined as a 2D tuple: (column, row) (e.g. t=*'M'*, p_i =(5, 2), p_f =(5, 4) move the white pawn in front of the king 2 squares up).

Program the pseudocode of a function that, given (t, p_i, p_f), executes this motion. We only know the positions p_{Captur} (position in which we can discard any number of pieces) and p_{Init} which is the center of the first square of the board as well as that all the squares of the chess board measure 50x50mm. It is assumed that the pieces can be easily picked up with the suction cup and that both p_{Init} and p_{Captur} are referenced to the coordinate system (X_R, Y_R) that corresponds to the coordinate system of the robot terminal element. The robot tool is called *tool*.



[ht]

Figure 5.3: Automatic chess.

Answer of exercise 44

function MOVE(t, p_i, p_f)
  $c_i \leftarrow (p_i[0]-1)\times 50$
  $f_i \leftarrow (p_i[1]-1)\times 50$
  $c_f \leftarrow (p_f[0]-1)\times 50$
  $f_f \leftarrow (p_f[1]-1)\times 50$
  if $t ==$ "C" **then** ▷ If Capture, first remove captured piece
    movej(**appro**(p_{Init}, $[f_f, c_f, -100, 0, 0, 0]$), tool, v_fast)
    movel(**appro**(p_{Init}, $[f_f, c_f, 0, 0, 0, 0]$), tool, v_slow)
    waitEndMove()
    enable(tool) ▷ Enable suction cup
    movel(**appro**(p_{Init}, $[f_f, c_f, -100, 0, 0, 0]$), tool, v_fast)
    movej(p_{Captur}, tool, v_fast)
    waitEndMove()
    disable(tool) ▷ Disable suction cup
  end if ▷ Always move piece from init to final pose
  movej(**appro**(p_{Init}, $[f_i, c_i, -100, 0, 0, 0]$), tool, v_fast)
  movel(**appro**(p_{Init}, $[f_i, c_i, 0, 0, 0, 0]$), tool, v_slow)
  waitEndMove()
  enable(tool)
  movel(**appro**(p_{Init}, $[f_i, c_i, -100, 0, 0, 0]$), tool, v_fast)
  movej(**appro**(p_{Init}, $[f_f, c_f, -100, 0, 0, 0]$), tool, v_fast)
  movel(**appro**(p_{Init}, $[f_f, c_f, 0, 0, 0, 0]$), tool, v_slow)
  waitEndMove()
  disable(tool)
  movel(**appro**(p_{Init}, $[f_f, c_f, -100, 0, 0, 0]$), tool, v_fast)
end function

Exercise 45 Drilling machine

Develop a program to drill a set of parts located on an input pallet (position $p_{InPallet}$) and place them on an output pallet (position $p_{OutPallet}$). A drill is available for drilling the parts. This machine is controlled through DIO $d5$. Write the pseudocode to perform this task considering:

- The following positions are already stored: pRest, pDrilling, pInPallet, pOutPallet.
- In the input and output pallets, each part is placed every 80mm in X and 100mm in Y.
- Once the part is in the $p_{Drilling}$ position, with the robot holding it, DIO $d5$ must be activated for 3 seconds to complete the drill. Once DIO $d5$ is disabled, wait for at least 2 seconds to ensure the piece can be safely removed.
- The robot tool is called *gripper*.

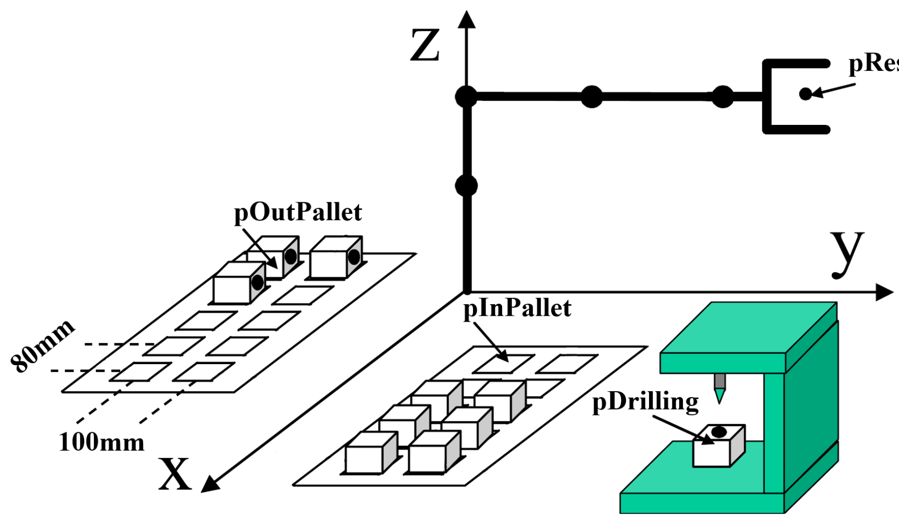


Figure 5.4: Drilling machine.

Answer of exercise 45

```
movej(p_Rest, gripper, v_slow)                                    ▷ Go to initial position
waitEndMove()
enable(tool)
for i ← 0,4 do
    for j ← 0,1 do
        movej(appro(p_InPallet, [80×i, 100×j, 100, 0, 0, 0]), gripper, v_fast)    ▷ Pick up piece
        movel(appro(p_InPallet, [80×i, 100×j, 0, 0, 0, 0]), gripper, v_slow)
        waitEndMove()
        disable(gripper)
        movel(appro(p_InPallet, [80×i, 100×j, 100, 0, 0, 0]), gripper, v_slow)
        movej(appro(p_Drilling, [0, -200, 0, 0, 0, 0]), gripper, v_fast)          ▷ Drill piece
        movel(p_Drilling, gripper, v_slow)
        waitEndMove()
        enable(d5)
        delay(3)
        disable(d5)
        delay(2)
        movel(appro(p_Drilling, [0, -200, 0, 0, 0, 0]), gripper, v_slow)
        movej(appro(p_OutPallet, [80×i, 100×j, 100, 0, 0, 0]), gripper, v_fast)   ▷ Place piece
        movel(appro(p_OutPallet, [80×i, 100×j, 0, 0, 0, 0]), gripper, v_slow)
        waitEndMove()
        enable(gripper)
        movel(appro(p_OutPallet, [80×i, 100×j, 100, 0, 0, 0]), gripper, v_slow)
    end for
end for
```