# Kinematic Analysis of Universal Robots Collaborative Robots

*Masters dissertation work*

JOAO GASPAR CUNHA
JOAO QUEIROGA PEREIRA

# Contents

# 1 Introduction

This work was developed in the context of our MSc dissertations: *A Collaborative Work Cell to Improve Ergonomics and Productivity at IKEA Industry Portugal* by João Cunha, and *Human-Like Motion Generation through Waypoints for Collaborative Robots in Industry 4.0* by João Pereira, in which we got to work with the collaborative robotic arm UR10e. Before producing our kinematics solution, we conducted a literature review on the UR robots' kinematics and realised there was a lack of thorough and detailed analysis of their kinematics. Additionally, we found the Universal Robots' documentation confusing and unclear. [1]

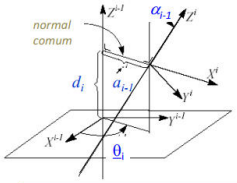
Thus, the main goal of this work is to provide an explicit and transparent guide into the UR robots kinematics (more specifically the UR10e) by expanding on the literature we found: [2, 3, 4, 5, 6, 7, 8, 9] and describing every part of our analysis. We present a forward kinematic solution based on the Modified Denavit-Hartenberg convention and an inverse kinematic solution based on a geometric analysis. These solutions were tested in CoppeliaSim using MATLAB through a remote API.

**Relação entre dois sistemas de eixos coordenados consecutivos, $^{i-1}T_i$: parâmetros de Denavit-Hartenberg**

Um sistema de eixos coordenados $\Sigma_i$ pode ser obtido a partir de um sistema de eixos coordenados $\Sigma_{i-1}$ aplicando as seguintes 4 transformações:

1) Translação ao longo do eixo $X_{i-1}$ de uma distância $a_{i-1}$
2) Rotação em torno de $X_{i-1}$ de um ângulo $\alpha_{i-1}$ *(ângulo ($Z_{i-1}$, $Z_i$))*
3) Translação ao longo de $Z_{i-1}$ depois da rotação em 2 (que é $Z_i$) de uma distância $d_i$
4) Rotação em torno de $Z_{i-1}$, depois da rotação em 2 (que é $Z_i$),de um ângulo $\theta_i$ *(ângulo ($X_{i-1}$, $X_i$))*

Dois "link frames":




Convention: Modified Denavit-Hartenberg Notation

# 2 Theoretical Background

## 2.1 Denavit-Hartenberg Convention

Kinematics consists in studying the motion of a certain system disregarding the forces or torques that cause it. Within kinematics one analyses position, velocity, acceleration (and even high order derivatives of position) with respect to time. [10] [11]

This kinematic analysis is most commonly achieved through the Denavit-Hartenberg convention[1], where the robot is described kinematically by defining four parameters for each link. In essence, the relation between $link_i$ and $link_{i+1}$ is represented by the following four parameters: [12]

- $\alpha_{i-1}$ - rotation around the $x_i$ axis (angle);
- $a_{i-1}$ - translation along the $x_i$ axis (distance);
- $\theta_i$ - rotation around the $z_i$ axis (angle);
- $d_i$ - translation along the $z_i$ axis (distance).

substituir por Estela Bicho, " Cinemática Directa de manipuladores série", Lecture notes, 2017.

The transformations between the reference frames along the robot's links results in the robot's kinematic equations.

However, before determining kinematics it is indispensable to first analyse the robotic manipulator's mechanical structure, *i.e.* identify its joints and links and their lengths. The UR10e has 6 rotational joint that can rotate ±360° and ten displacements, refer to figure 1 and table 1. This mechanical analysis was conducted using documentation available in the Universal Robots' website. Across their documentation joint 1 is referred as "base", joint 2 "shoulder", joint 3 "elbow", joint 4 "wrist 1", joint 5 "wrist 2", and joint 6 "wrist 3". [13]

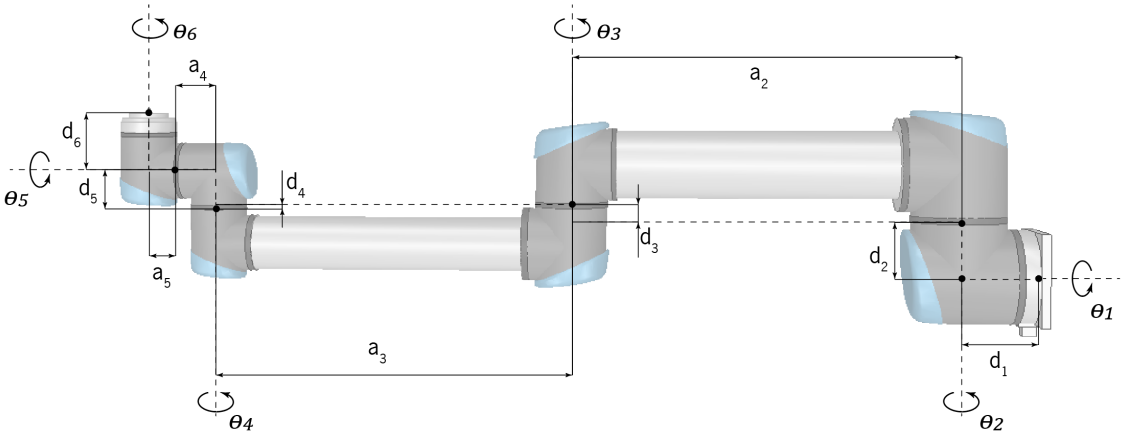



Figure 1: UR10e mechanical structure

[1] A convention introduced by Jacques Denavit and Richard Hartenberg in 1955.

Table 1: UR10e distances and joints

| ***Joints*** | Value (degrees) | ***d*** | Value (meters) | ***a*** | Value (meters) |
|---|---|---|---|---|---|
| $\theta_1$ | $[-360;360]$ | $d_1$ | 0.181 | - | - |
| $\theta_2$ | $[-360;360]$ | $d_2$ | 0.088 | - | - |
| $\theta_3$ | $[-360;360]$ | $d_3$ | 0.020 | $a_3$ | 0.613 |
| $\theta_4$ | $[-360;360]$ | $d_4$ | 0.001 | $a_4$ | 0.571 |
| $\theta_5$ | $[-360;360]$ | $d_5$ | 0.060 | $a_5$ | 0.060 |
| $\theta_6$ | $[-360;360]$ | $d_6$ | 0.117 | $a_6$ | 0.067 |

### 2.1.1 Denavit-Hartenberg Parameters

In order to derive the Denavit-Hartenberg parameters, one must first position the manipulator in its home pose, and secondly attribute a reference frame $\{x_i, y_i, z_i\}$ for every joint. In our case the manipulator's home pose is defined as all joint values equal to zero, which determine an upright pose[2].

The reference frames {0} to {6} are represented in figure 2 (where the x, y and z components are represented with red, green, and blue arrows respectively), as well as the robot's home pose. To attribute these reference frames we followed three basic rules: a) the right hand rule; b) the location of the frame is the centre of the corresponding joint, *e.g.*, reference frame {2} is located at the centre of joint 2; and c) the z-axis must indicate the rotational axis of the joint. [14]

Reference frame {0} is the origin of the robot, and is located on the centre of its base. From henceforth, reference frames [{1}, {2}, {3}, {4}, {5}, {6}] all follow the above stated rules. However, due to the order of multiplication of the transformations for the modified Denavit-Hartenberg convention, it was necessary to add two auxiliary reference frames: {4'}, and {5'} located at the centre of joint 5.

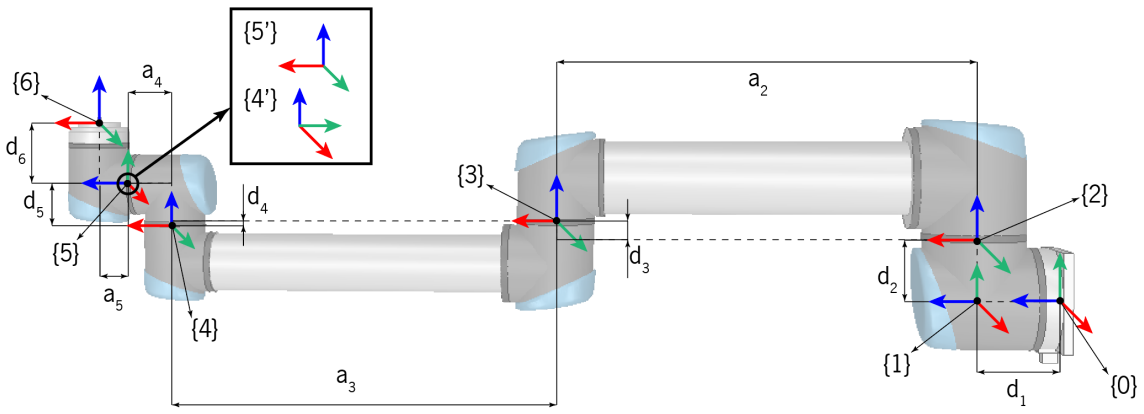



Figure 2: Reference frames along the UR10e structure

After attributing the reference frames, it is possible to compute the transformations that occur between them, using the Denavit-Hartenberg parameters mentioned previously, table 2.

[2]For the *CoppeliaSim* robot model and for the real robot

Table 2: UR10e Denavit-Hartenberg transformations between reference frames

| ${}^{i-1}T_i$ | $\alpha_{i-1}$ | $a_{i-1}$ | $d_i$ | $\theta_i$ |
|---|---|---|---|---|
| $\{0\} \rightarrow \{1\}$ | 0 | 0 | $d_1$ | $\theta_1$ |
| $\{1\} \rightarrow \{2\}$ | $-\pi/2$ | 0 | $d_2$ | $\theta_2 - \pi/2$ |
| $\{2\} \rightarrow \{3\}$ | 0 | $a_2$ | $d_3$ | $\theta_3$ |
| $\{3\} \rightarrow \{4\}$ | 0 | $a_3$ | $d_4$ | $\theta_4$ |
| $\{4\} \rightarrow \{4'\}$ | 0 | $a_4$ | $d_5$ | $\pi/2$ |
| $\{4'\} \rightarrow \{5\}$ | $\pi/2$ | 0 | 0 | $\theta_5$ |
| $\{5\} \rightarrow \{5'\}$ | $-\pi/2$ | 0 | 0 | $-\pi/2$ |
| $\{5'\} \rightarrow \{6\}$ | 0 | $a_5$ | $d_6$ | $\theta_6$ |

### 2.1.2 Modified Denavit-Hartenberg Matrix

Like the homogeneous matrix, the Denavit-Hartenberg matrix is a transformation matrix from a system of coordinates to another. Using a set of multiplications of Denavit-Hartenberg matrices and a table of Denavit-Hartenberg parameters, the final result is a transformation matrix of a final system of coordinates in relation to a initial one. [15]

The modified Denavit-Hartenberg homogeneous transformation ${}^{i-1}T_i$, from frame $i-1$ to $i$ is defined in equations 1, 2, and 3.

$$ {}^{i-1}T_i = Rotation_{x(\alpha_{i-1})} * Translation_{x(a_{i-1})} * Translation_{z(d_i)} * Rotation_{z(\theta_i)} \tag{1} $$

$$ {}^{i-1}T_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_1) & -\cos(\alpha_1) & 0 \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

$$ {}^{i-1}T_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & \alpha_{i-1} \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -\sin(\alpha_{i-1})d_i \\ \sin(\theta_i)\sin(\alpha_{i-1}) & \cos(\theta_i)\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & \cos(\alpha_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3} $$

This matrix is composed by the rotation and translation matrix that generate the required motion from frame $i-1$ to frame $i$:

$$ {}^{i-1}T_i = \begin{bmatrix} {}^{i-1}Rot_{i(3x3)} & {}^{i-1}Trans_{i(3x1)} \\ 0 & 1 \end{bmatrix} \tag{4} $$

In equation 4, ${}^{i-1}Trans_{i(3x1)}$ represents the translation from reference frame $i-1$ to reference frame $i$, and matrix ${}^{i-1}Rot_{i(3x3)}$ represents the rotation matrix in relation to the same frame. From this rotation matrix it is possible to extract the Roll-Pitch-Yaw values of the robot's end-effector in relation to the base.

Only if **{0}^R_{i}**, .i.e. only if referred to the base

## 2.2 Euler Angles

The Euler angles yield the orientation of a rigid body using a combination of three rotations about different axis. Since the rotation matrices are characterised by nine elements related by six constraints due to the orthogonality conditions, only three parameters are sufficient to describe the orientation in space. Therefore, a generic rotation matrix, derived by a sequence of three elementary rotations, implies 12 possible combinations of rotation axes. In the following analysis the ZYX (Roll-Pitch-Yaw) angles will be used. [16]

The rotation resulting from Roll-Pitch-Yaw angles consist in an accumulation of three rotations around the axis X, Y, Z respectively concerning the referential base:

- Rotation by the angle $\gamma$ about the axis X (Yaw);
- Rotation by the angle $\beta$ about the axis Y (Pitch);
- Rotation by the angle $\alpha$ about the axis Z (Roll).

The final orientation frame is achieved by multiplying the matrices in the opposite order of the rotation, refer to equations[3] 5, 6, and 7.

$$R_{XYZ}(\gamma,\beta,\alpha) = R(Z,\alpha) * R(Y,\beta) * R(X,\gamma) \tag{5}$$

$$R_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} \cos(\alpha & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1) \end{bmatrix} \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix} \tag{6}$$

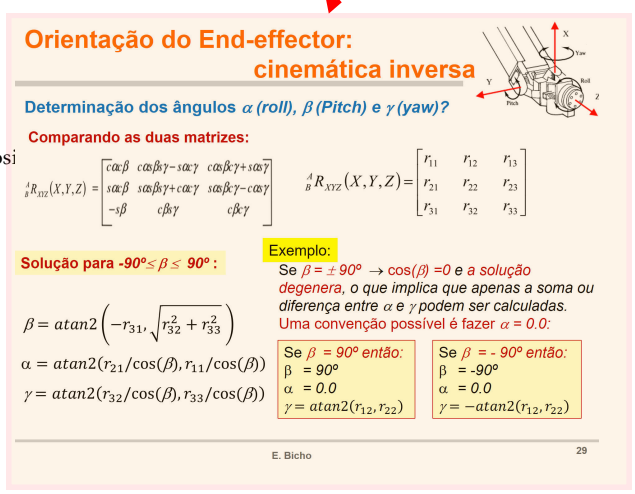
$$R_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} c(\alpha)c(\beta) & c(\alpha)s(\beta)s(\gamma) - s(\alpha)c(\gamma) & c(\alpha)s(\beta)c(\gamma) + s(\alpha)s(\gamma) \\ s(\alpha)c(\beta) & s(\alpha)s(\beta)s(\gamma) + c(\alpha)c(\gamma) & s(\alpha)s(\beta)c(\gamma) - c(\alpha)s(\gamma) \\ -s(\beta) & c(\beta)s(\gamma) & c(\beta)c(\gamma) \end{bmatrix} \tag{7}$$

Knowing the values of the end-effector's orientation, it is possible to determine the corresponding Euler angles. Considering the generic rotation matrix represented in equation 8, the Euler angles XYZ can be obtained by comparing it with the matrix in equation 7, resulting in equation 9.

$$R_{XYZ} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{8}$$

$$\begin{aligned} \beta &= \operatorname{atan2}\left(-r_{31}, \sqrt{(r_{11})^2 + (r_21)^2}\right) \\ \alpha &= \operatorname{atan2}\left(\frac{r_{21}}{c(\beta)}, \frac{r_{11}}{c(\beta)}\right) \\ \gamma &= \operatorname{atan2}\left(\frac{r_{32}}{c(\beta)}, \frac{r_{33}}{c(\beta)}\right) \end{aligned} \tag{9}$$

**Incompleto, falta resolver a " singularidade"**




[3] From henceforth, in order to simplify notation, the sine and cosi
cos(x), and s(x) = sin(x).

# 3 Forward and Inverse Kinematics

## 3.1 Forward Kinematics

Forward kinematics consists on finding the coordinates of the robot's tip $(x_e, y_e)^T$, from its joint positions $q = [\theta_1, \theta_2, \theta_2, \theta_3, \theta_4, \theta_4, \theta_5, \theta_6]^T$. Refer to equation 10 called the manipulator kinematic equation. For every vector of joint values, $q$, the tip's position always exists and is unique. [17]

tb orientaçao do faz parte da cinemática direta!

este areferência é sobre manipulados com 2 DOF

$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = f(q) \tag{10}$$

substituir por uma dos meus slides onde há posição e orientação

In order to compute a manipulator's forward kinematics one must follow the following st
a) assign reference frames along the robot's structure; b) determine the Denavit-Hartenberg parameters; (refer to section 2 for steps a) and b)); c) determine the individual transformation matrices; and d) determine the general transformation matrix to obtain the Cartesian coordinates of the robot's tip in terms of it's joint values. [17]

Forward kinematics can be achieved using a trigonometric approach, which can be considerably difficult for complex robot structures, or by calculating the homogeneous transformation matrix that describes the position and orientation of the end-effector relative to the base. This transformation matrix is obtained by multiplying the transformations between the frames in the adjacent links.

### 3.1.1 Computing the Individual Transformation Matrices

Using the Denavit-Hartenberg parameters specified in subsection 2.1.1 and the modified Denavit-Hartenberg matrix in subsection 2.1.2, it is possible to compute the following transformation matrices: ${}^0T_1$, ${}^1T_2$, ${}^2T_3$, ${}^3T_4$, ${}^4T_5$, and ${}^5T_6$. [12]

$${}^0T_1 = \begin{bmatrix} c(\theta_1) & -s(\theta_1) & 0 & 0 \\ s(\theta_1) & c(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$${}^1T_2 = \begin{bmatrix} c(\theta_2 - \pi/2) & c(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ c(\theta_2) & -c(\theta_2 - \pi/2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

$${}^2T_3 = \begin{bmatrix} c(\theta_3) & -s(\theta_3) & 0 & a_3 \\ s(\theta_3) & c(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

$${}^3T_4 = \begin{bmatrix} c(\theta_4) & -s(\theta_4) & 0 & a_4 \\ s(\theta_4) & c(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

$${}^4T_{4'} = \begin{bmatrix} 0 & -1 & 0 & a_5 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$${}^{4'}T_5 = \begin{bmatrix} c(\theta_5) & -s(\theta_5) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s(\theta_5) & c(\theta_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

$${}^5T_{5'} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

$${}^{5'}T_6 = \begin{bmatrix} c(\theta_6) & -s(\theta_6) & 0 & a_6 \\ s(\theta_6) & c(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

### 3.1.2 Computing the Complete Transformation Matrix

Having defined the homogeneous transformation matrices along the robotic arm, the transformation from the robot base to robot-tip is given by:

$$ {}^{0}T_{6} = {}^{0}T_{1} * {}^{1}T_{2} * {}^{2}T_{3} * {}^{3}T_{4} * {}^{4}T_{4'} * {}^{4'}T_{5} * {}^{5}T_{5'} * {}^{5'}T_{6} \tag{19} $$

Using the individual transformation matrices obtained in subsection 3.1.1, and knowing the trigonometric laws presented in equation 20, one can compute the complete transformation matrix presented in 21, by multiplying the first two terms of the equations iteratively until obtaining ${}^{0}T_{6}$.

$$ \begin{aligned} \cos(\alpha+\beta) &= \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) \\ \sin(\alpha+\beta) &= \cos(\alpha)\sin(\beta) - \cos(\beta)\sin(\alpha) \end{aligned} \tag{20} $$

$$ \begin{aligned} {}^{0}T_{6} &= {}^{0}T_{1} * {}^{1}T_{2} * {}^{2}T_{3} * {}^{3}T_{4} * {}^{4}T_{4'} * {}^{4'}T_{5} * {}^{5}T_{5'} * {}^{5'}T_{6} = \\ &= {}^{0}T_{2} * {}^{2}T_{3} * {}^{3}T_{4} * {}^{4}T_{4'} * {}^{4'}T_{5} * {}^{5}T_{5'} * {}^{5'}T_{6} = \\ &= {}^{0}T_{3} * {}^{3}T_{4} * {}^{4}T_{4'} * {}^{4'}T_{5} * {}^{5}T_{5'} * {}^{5'}T_{6} = \\ &= {}^{0}T_{4} * {}^{4}T_{4'} * {}^{4'}T_{5} * {}^{5}T_{5'} * {}^{5'}T_{6} = \\ &= {}^{0}T_{4'} * {}^{4'}T_{5} * {}^{5}T_{5'} * {}^{5'}T_{6} = \\ &= {}^{0}T_{5} * {}^{5}T_{5'} * {}^{5'}T_{6} = \\ &= {}^{0}T_{5'} * {}^{5'}T_{6} = \\ &= {}^{0}T_{6} \end{aligned} \tag{21} $$

### 3.1.3 Obtaining the Position and Orientation of the Robot's Tip

Now that the transformation matrix ${}^{0}T_{6}$ is known, the position and orientation of the robot's end-effector can be easily obtained. As presented earlier in equation 4, a transformation matrix is composed by a rotation matrix of dimension $3x3$ and a translation matrix $3x1$. Thus, the transformation matrix ${}^{0}T_{6}$ can be represented as:

Colocar o superfix 0 atrá para indicar que é em relação ao referencial {0}

$$ {}^{0}T_{6} = \begin{bmatrix} [\boldsymbol{R_6}]_{3x3} & [\boldsymbol{P_6}]_{3x1} \\ 0_{1x3} & 1 \end{bmatrix} \tag{22} $$

Therefore, the position $(x_e, y_e, z_e)^T$ of the robot's end-effector is given by the ~~matrix~~ $P_6$ and the orientation $(\gamma_e, \beta_e, \alpha_e)^T$ is determined from the matrix $R_6$ using equation 9.

## 3.2 Inverse Kinematics

In terms of task space what is specified to the robot is the desired position, $(x_e, y_e, z_e)^T$, of its tip, or a trajectory of points which he must follow. In these situations the problem consists in determining the vector $q = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$ that implies that the robot's tip position is $(x_e, y_e, z_e)^T$. [17]

However, it just so happens that $q$ might not always exist, and when it does, the solution still might not be unique. The problem of obtaining the vector $q$ that makes the robot's tip position equal to $(x_e, y_e, z_e)^T$ is called inverse kinematics, and it is not as linear to solve as the forward kinematics. [17]

### 3.2.1 Computing $\theta_1$

In order to find the value of joint 1 ($\theta_1$) we considered a geometric approach, which consists of first determining the value of ${}^0P_5$ (the position of frame {5} in reference to frame {0}) using an overhead view of the robot.

${}^0P_5$ can be calculated using the known values of ${}^0T_6$, $d_6$, and $a_5$. Considering the fourth column of ${}^0T_6$, ${}^0P_6 = [p_x, p_y, p_z, 1]^T$ (the position of the robot's tip), we can correctly infer that the position of frame {5} in relation to frame {0} consists in a translation of $-a_5$ in the x-axis, and $-d_6$ in the z-axis from the $6^{th}$ frame, refer to figure 3.

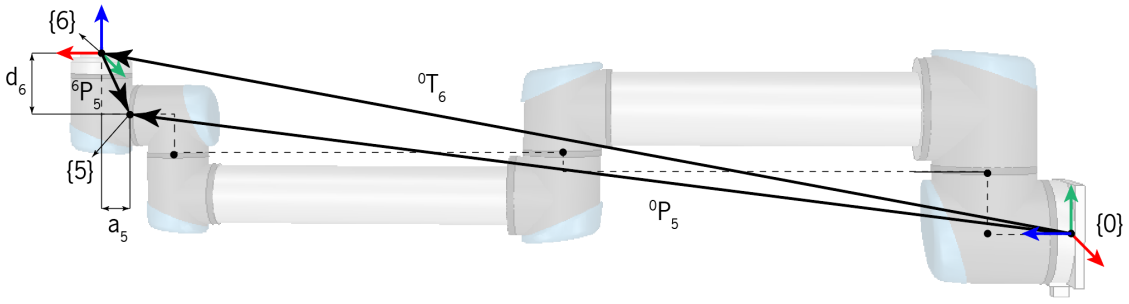



Figure 3: Determining ${}^0P_5$

Algebraically this translation can be written as:

$$ {}^0P_5 = {}^0P_6 * \begin{bmatrix} -a_5 & 0 & -d_6 & 1 \end{bmatrix}^T $$

So as to visualise the appearance of $\theta_1$, we consider an overhead view of the robot, where $\theta_2 = \pi/2$ and $\theta_1 \neq 0$, figure 4. This allows us to see $\theta_1$ and represent the robot in the x-y plane of frame {0}, which is extremely useful for a geometrical analysis.

Empirically, observing figure 4, one can apprehend that the value of $\theta_1$ is equal to the subtraction of $\theta_1^{'}$ by $\pi$, equation 24. Whereas the value of $\theta_1^{'}$ is related to the sum of angle $\psi$ with $\phi$ and $\frac{\pi}{2}$, equation 25.

$$ \theta_1^{'} = \psi + \phi + \frac{\pi}{2} \tag{24} $$

$$ \theta_1 = \theta_1^{'} - \pi \tag{25} $$

Angle $\psi$ is created using ${}^0P_5$ and its x-y components which form triangle 1. $\psi$ can now be directly calculated using the tangent function, since both the values of ${}^0P_{5x}$ and ${}^0P_{5y}$ are known from the previously calculated value of ${}^0P_5$. Thus, $\psi$ is equal to:

$$\psi = \text{atan2}({}^0P_{5y}, {}^0P_{5x}) \tag{26}$$

$\phi$ can be obtained using the values of ${}^0P_5$ and $d_2$, $d_3$, $d_4$, $d_5$, which form triangle 2, equation 4. $\theta_1$ has two possible solutions which are dependant on the configuration of the shoulder joint (joint 2), left or right.

$$\phi = \pm \arccos\left(\frac{d_2 + d_3 - d_4 + d_5}{\sqrt{{}^0P_{5x}^2 + {}^0P_{5y}^2}}\right) \tag{27}$$

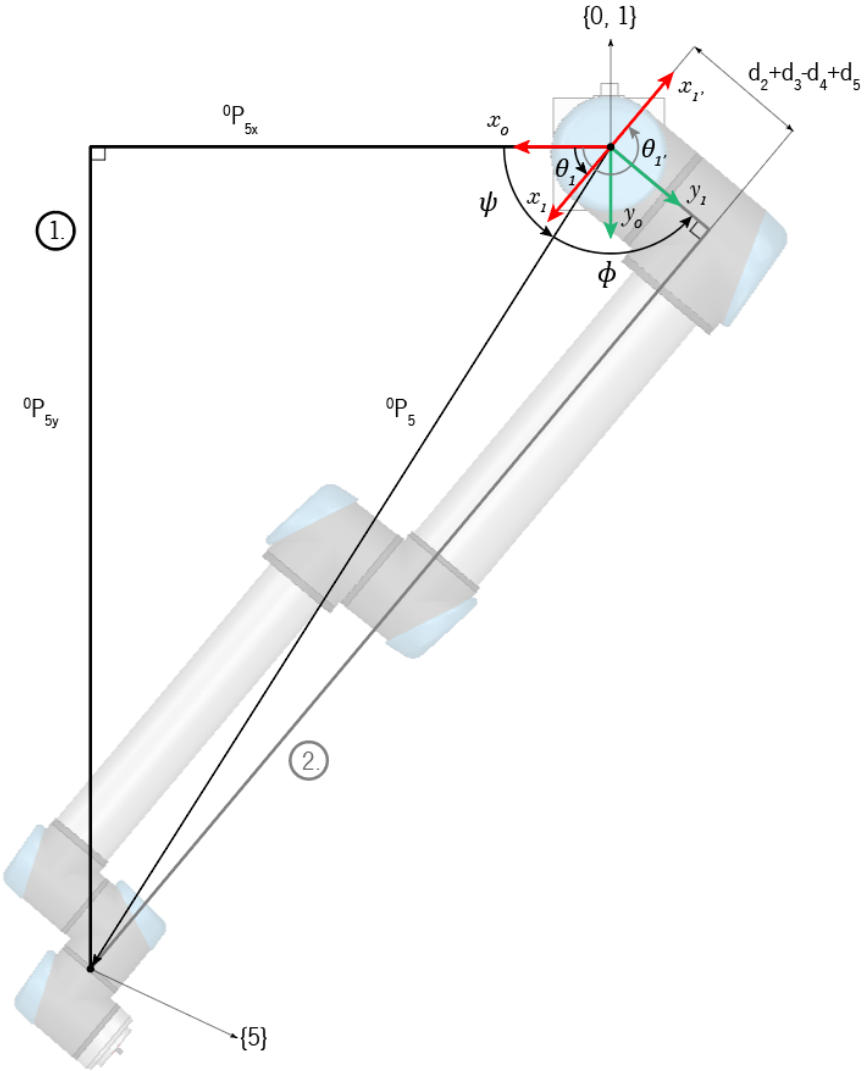



Figure 4: Visualising $\theta_1$

### 3.2.2 Computing $\theta_5$

In order to geometrically visualise the value of joint 5, $\theta_4$ is set to $\pi/2$, which makes the x-axis of frame {4} and {5} form $\theta_5$ in the x-y plane of frame {0}.

Since the value of $\theta_1$ is known, it is possible to calculate the transformation matrix from frame {1} to frame {6}, ${}^1T_6$. Using the fourth column of ${}^1T_6$ we obtain the position of frame {6} from frame {1}, and consequently its $y$ component. As it is possible to see in figure 5, ${}^1P_{6y}$ only depends on $\theta_5$. Analysing this figure, one can conclude that the component ${}^1P_{6y}$ is defined by the sum of the physical length of the robot arm along the $y_1$ referential until frame {5} and the length created by $\theta_5$ in the same direction $d_i$, resulting in equation 28.

$$\begin{aligned} {}^1P_{6y} &= d_i + d_2 + d_3 - d_4 + d_5 \\ where,\ d_i &= d_6 * \cos(\theta_5) \end{aligned} \tag{28}$$

indicar como:
1T6 = 1T0. 0T6 = inv (0T1).0T6

conseguem mostrar as expressões matemáticas desta matriz?

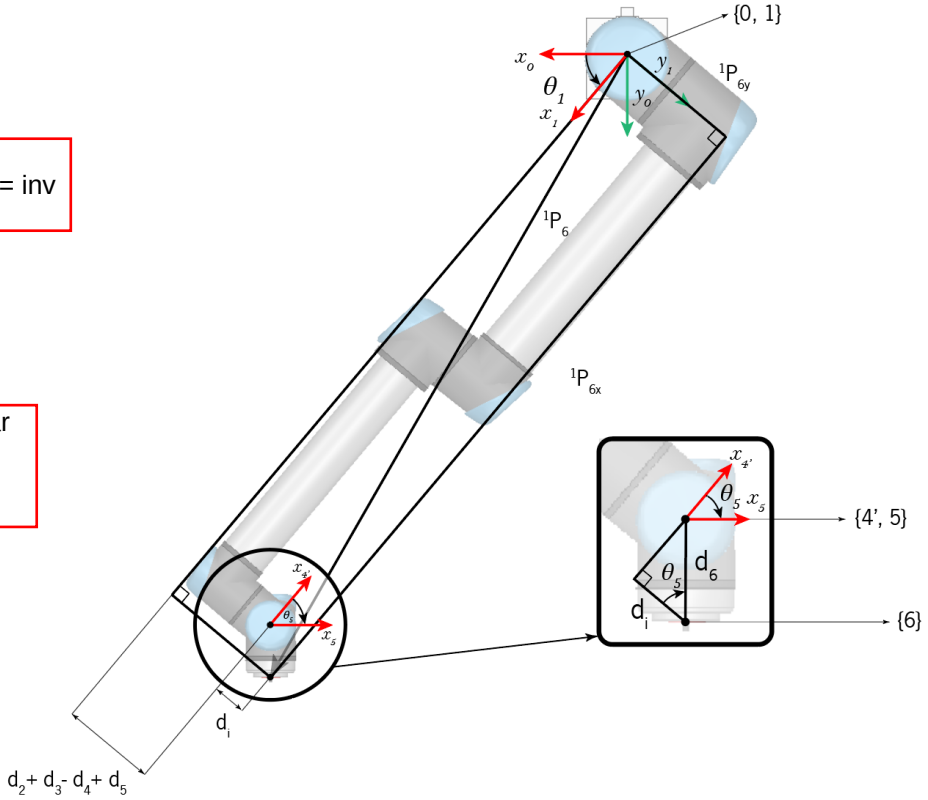



Figure 5: Visualising $\theta_5$

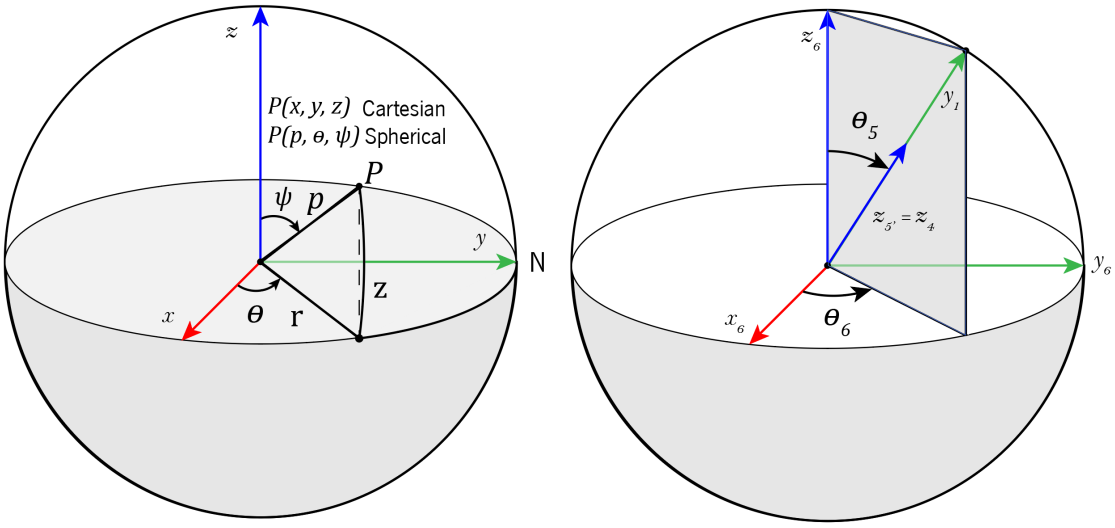
Therefore, by solving the equation 28, the expression of $\theta_5$ is given by equation 5. $\theta_5$ also has two possible values which are determined by the wrist being up or down.

$$\theta_5 = \pm \arccos\left(\frac{{}^1P_{6y} - (d_2 + d_3 + d_4 + d_5)}{d_6}\right) \tag{29}$$

### 3.2.3 Computing $\theta_6$

The calculation of $\theta_6$ is based on the analysis of $y_1$ seen from frame {6}, ${}^6y_1$. By inspecting the orientation of the reference frames along the robot's structure present in figure 2 and abstracting the translations between frame {6} and {1}, one can conclude that the orientation of $y_1$, only depends on $\theta_5$ and $\theta_6$, since it will always be parallel to axes $z_{\{2,3,4,4',5'\}}$.

To examine this, we found out that it would be extremely useful to represent ${}^6y_1$ using a spherical coordinate system, where the azimuthal angle is $\theta_6$ and the polar angle is $\theta_5$. Figure 6a portrays a unit sphere showing the radial distance (r), polar angle ($\psi$) and azimuthal angle ($\theta$) of a point P. Using this representation, figure 6b is easily obtained, moreover it simplifies the conversion from spherical to Cartesian coordinates.




(a) Representation of point $P$ using spherical and Cartesian coordinates

(b) Spherical representation of ${}^6y_1$, where the azimuthal angle is $\theta_6$ and the polar angle is $\theta_5$

Figure 6: Coordinate analysis of ${}^6y_1$

$$Cartesian \leftrightarrow Spherical$$
$$\begin{cases} x = p\cos(\theta)\sin(\psi) \\ y = p\sin(\theta)\sin(\psi) \\ z = p\cos(\theta) \end{cases} \tag{30}$$

Knowing that equation 30 translates the conversion from spherical to Cartesian coordinates, it is possible to obtain the $x$, $y$, $z$ components of ${}^6y_1$:

$${}^6Y_1 = \begin{bmatrix} \sin(\theta_5)\cos(\theta_6) \\ \sin(\theta_5)\sin(\theta_6) \\ \cos(\theta_5) \end{bmatrix} \tag{31}$$

Since the value of $\theta_1$ is already known, and consequently the transformation matrix ${}^1T_6$, one can

explicitly deduce the value of vector ${}^{6}y_1$. Hence, the system present in equation 30 can be solved for $\theta_6$:

$$\begin{cases} \sin(\theta_5)\cos(\theta_6) = {}^{6}y_{1_x} \\ \sin(\theta_5)\sin(\theta_6) = {}^{6}y_{1_y} \end{cases} \Rightarrow \begin{cases} \cos(\theta_6) = \left({}^{6}y_{1_x}\right) / \sin(\theta_5) \\ \sin(\theta_6) = \left({}^{6}y_{1_y}\right) / \sin(\theta_5) \end{cases} \tag{32}$$

$$\theta_6 = \operatorname{atan2}\left(\frac{-{}^{6}y_{1_y}}{\sin(\theta_5)}, \frac{{}^{6}y_{1_x}}{\sin(\theta_5)}\right) \tag{33}$$

However, equation 33 is not entirely correct. Between frame {5} and {6}, besides a transformation of $\theta_6$ along the z-axis we considered an auxiliary frame ({5'}) that presumes a transformation of $-\pi/2$ along the z-axis first. So, this means that there needs to be an adjustment to equation 33 that describes this transformation:

$$\theta_6 = -\pi/2 + \operatorname{atan2}\left(\frac{-{}^{6}y_{1_y}}{\sin(\theta_5)}, \frac{{}^{6}y_{1_x}}{\sin(\theta_5)}\right) \tag{34}$$

It is important to note that equation 34 only has a solution if $\sin(\theta_5) \neq 0$. Otherwise, joints 2, 3, 4 and 6 are parallel and the solution for $\theta_6$ is undetermined. When this occurs, an arbitrary value for $\theta_6$ can be given.

verdade para posição mas não para a orientação?!...

### 3.2.4 Computing $\theta_3$

Concerning the calculation of $\theta_3$ and $\theta_2$ we considered a two link planar manipulator from frame {2} to frame {4} (links $a_2$ and $a_3$), refer to figure 7. This analysis is possible, since these three frames are parallel, which means the angle they form between each other can be seen along the same two axes. Since the values of $\theta_1$, $\theta_5$ and $\theta_6$ are already known, one can explicitly obtain the value of matrix ${}^{1}T_4$ through equation 35.

$$\begin{cases} {}^{1}T_6 = inv({}^{0}T_1) * {}^{0}T_6 \\ {}^{4}T_5 = {}^{4}T_{4'} * {}^{4'}T_5 \\ {}^{5}T_6 = {}^{5}T_{5'} * {}^{5'}T_6 \end{cases} \qquad {}^{1}T_4 = {}^{1}T_6 * inv({}^{4}T_5 * {}^{5}T_6) \tag{35}$$

Analysing figure 7, $\theta_3$ is equal to the subtraction of $\psi$ to $\pi$, equation 36. In its turn, $\psi$ can be derived using the law of cosines and is given by equation 37. The two solutions for $\theta_3$ correspond to the elbow being up or down.

$$\theta_3 = \pi - \psi \tag{36}$$

$$\begin{aligned} \cos(\psi) &= \frac{{}^{1}P_{4_{XZ}}^{2} - {a_3}^2 - {a_2}^2}{-2a_2a_3} \\ \psi &= \pm\arccos\left(\frac{{}^{1}P_{4_{XZ}}^{2} - {a_3}^2 - {a_2}^2}{-2a_2a_3}\right) \end{aligned} \tag{37}$$

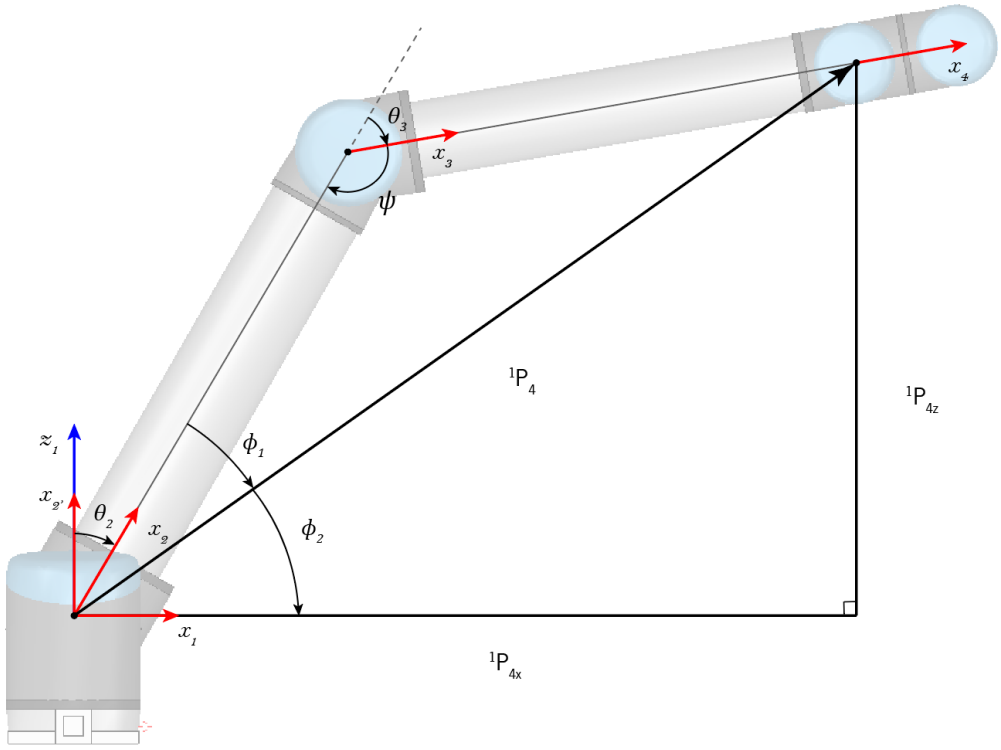



Figure 7: Visualising $\theta_3$ and $\theta_2$

### 3.2.5 Computing $\theta_2$

Now that $\theta_3$ is known, the value of the second joint ($\theta_2$) can be determined. Note that the value of $\theta_2$ is equal to the angle between $x_2$ and $x_{2'}$ which represents the orientation of $x_2$ in the home pose (which forms a right angle with $x_1$). Thus, the value of $\theta_2$ is established by the subtraction of $\pi/2$ with the sum of the auxiliary angles $\phi_1$ and $\phi_2$, equation 38. These two angles can be obtained with the trigonometry concepts of arc tangent and the law of sines, respectively, as demonstrated in equations 39 and 40.

$$\theta_2 = \frac{\pi}{2} - (\phi_1 + \phi_2) \tag{38}$$

$$\phi_2 = \text{atan2}\left({}^1P_{4_Z}, {}^1P_{4_X}\right) \tag{39}$$

$$\frac{a_3}{\sin(\phi_1)} = \frac{{}^1P_{4_{XZ}}}{\sin(\psi)} \Rightarrow \phi_1 = \arcsin\left(\frac{a_3 \sin(\psi)}{{}^1P_{4_{XZ}}}\right) \tag{40}$$

### 3.2.6 Computing $\theta_4$

Being defined as the angle between $x_3$ and $x_4$ and knowing the value of $\theta_{\{1,\ 2\ ,3\ ,5\ ,6\}}$, $\theta_4$ can be easily obtained using the individual transformation matrix ${}^3T_4$, more specifically using the $x$, $y$ components of the first column, ${}^3X_4$. Refer to equation 41, and figure 8.

$$\theta_4 = \text{atan2}({}^3X_{4_y}, {}^3X_{4_x}) \tag{41}$$

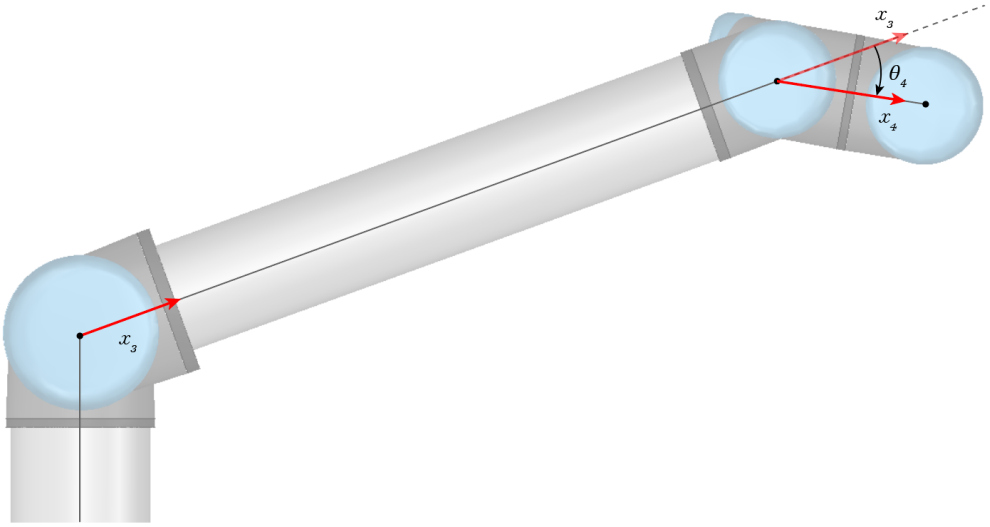
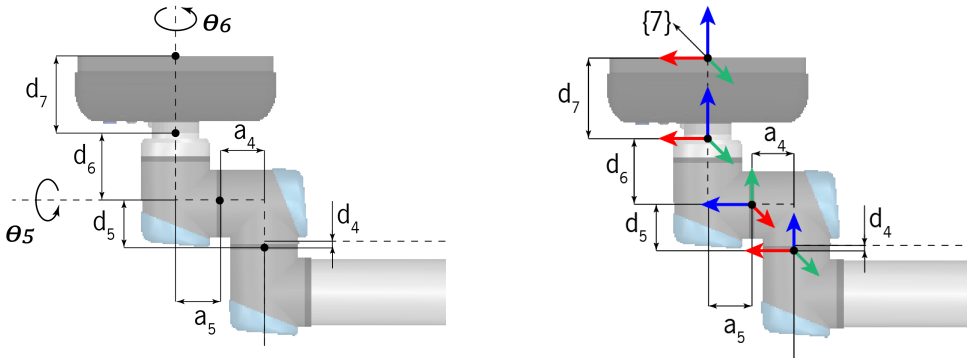



Figure 8: Visualising $\theta_4$

### 3.3 Adding an End-Effector

By attaching an end-effector there are consequent alterations to the machine's structure and kinematics. To explain these alterations we will use, as an example, the Uniggriper Co/Light Regular which is a collaborative vacuum gripper. The gripper has a height of 101 mm which means, the robot's tip is dislocated an additional 101 mm, refer to figure 9a. Kinematically it is necessary to add a new reference frame, figure 9b, and alter the Denavit-Hartenberg parameters by adding a new line that corresponds to a translation on the z-axis, table 3.




(a) Mechanical structure

(b) Reference frames

Figure 9: Robotic system mechanical structure and reference frames (UR10e with the UniGripper)

Table 3: Robotic system Denavit-Hartenberg parameters ($d_7$ = 101 mm)

| ${}^{i-1}T_i$ | $\alpha_{i-1}$ | $a_{i-1}$ | $d_i$ | $\theta_i$ |
|---|---|---|---|---|
| $\{0\} \rightarrow \{1\}$ | 0 | 0 | $d_1$ | $\theta_1$ |
| $\{1\} \rightarrow \{2\}$ | $-\pi/2$ | 0 | $d_2$ | $\theta_2 - \pi/2$ |
| $\{2\} \rightarrow \{3\}$ | 0 | $a_2$ | $d_3$ | $\theta_3$ |
| $\{3\} \rightarrow \{4\}$ | 0 | $a_3$ | $d_4$ | $\theta_4$ |
| $\{4\} \rightarrow \{4'\}$ | 0 | $a_4$ | $d_5$ | $\pi/2$ |
| $\{4'\} \rightarrow \{5\}$ | $\pi/2$ | 0 | 0 | $\theta_5$ |
| $\{5\} \rightarrow \{5'\}$ | $-\pi/2$ | 0 | 0 | $-\pi/2$ |
| $\{5'\} \rightarrow \{6\}$ | 0 | $a_5$ | $d_6$ | $\theta_6$ |
| $\{6\} \rightarrow \{7\}$ | 0 | 0 | $d_7$ | 0 |

In terms of forward kinematics this means that there will be an additional transformation, ${}^{6}T_7$:

$$ {}^{6}T_7 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{42} $$

Additionally, it is necessary to compute a new complete transformation matrix ${}^{0}T_7$, that is given by:

$$ {}^{0}T_7 = {}^{0}T_6 * {}^{6}T_7 \tag{43} $$

For the inverse kinematics the only precaution necessary is that the value of ${}^{0}P_6$ is not direct anymore, however, using the values of ${}^{0}T_7$ and $d_7$, it can be computed. Considering the fourth column of ${}^{0}T_7$ (${}^{0}P_7$), ${}^{0}P_6$ is the result of a translation of $d_7$ in the z-axis from the $7^{th}$ frame. Algebraically this translation can be written as seen in equation 44.

$$ {}^{0}P_6 = {}^{0}P_7 * \begin{bmatrix} 0 & 0 & -d_7 & 1 \end{bmatrix}^T \tag{44} $$

The rest of the calculations for the joint angles are the same as seen before.

# 4 Validation

## 4.1 MATLAB Code

The MATLAB code is divided into three function files: *MDHMatrix.m*; *fwdKin.m*; and *invKin.m*, and a single script file: *main.m*.

### 4.1.1 *MDHMatrix.m* Function File

The *MDHMatrix.m* file contains a function capable of computing Modified Denavit-Hartenberg matrices referred in subsection 2.1.2. It receives as input a single line of a Denavit-Hartenberg matrix, *i.e.* a vector of 4 values $[\alpha; a, d, \theta]$ that indicate the rotation around the x axis of frame i (in deg.), the translation along the x axis of frame i (in meters), the translation along the z axis of frame i (in meters), and the rotation around the z axis of frame i (in deg.), respectively, and outputs a 4x4 transformation matrix associated to that line of the Denavit-Hartenberg matrix. Summarily, it computes the transformation matrix from one system of coordinates to another.

### 4.1.2 *fwdKin.m* Function File

The *fwdKin.m* file is responsible for determining the general transformation matrix mentioned in subsection 3.1.2 from the Denavit-Hartenberg parameters matrix in table 3. It returns a bi-dimensional array of arrays of matrices named $M$, where $M\{1\}$ contains an array with the individual transformation matrices in subsection 3.1.1, and $M\{2\}$ an array with the compound transformation matrices mentioned along subsection 3.1.2. To compute the individual transformation matrices the *MDHMatrix* function is used.

### 4.1.3 *invKin.m* Function File

The *invKin.m* file computes the eight inverse kinematic solutions for the robot's joint angles using the equations presented along section 3.2 from the geometric values of the robot's dimensions and the end-effector position. As input it receives two arrays with the robots link dimensions and the position and orientation of the end-effector, as output it returns a 8x6 matrix with the eight inverse solutions for the robot's six joints.

### 4.1.4 *main.m* Script File

The *main.m* script is where the *fwdKin* and *invKin* functions are called. This script can be broken down into three sections:

- **Initialisation**: where variables are initialised and the connection with *CoppeliaSim* is set up;
- **User Interface**: where the user specifies the values of the dimensions of the robot, and the target joint values. It is also here that the Denavit-Hartenberg matrix is coded;
- **Main Program**: section responsible for checking the connection with CoppeliaSim, retrieving the joint handles from the robot model in *CoppeliaSim*, computing the forward and inverse kinematics for the target joint values, and sending the output from the inverse kinematic solution to the robot model in *CoppeliaSim*.

This program is intended to test the correct computation of the end-effector position from a set of target joint values, and vice-versa, compared to the physical representation of the robot in

*CoppeliaSim*.

## 4.2 *CoppeliaSim* Simulation

### 4.2.1 Scene Setup

The scene is composed of an UR10 robot model available in *CoppeliaSim* model browser, with a Unigripper Co/Light Regular model attached to its flange. Additionally, the robot is setup on top of a cylindrical pedestal with 0.3 m diameter and 0.7 m height, figure 10.

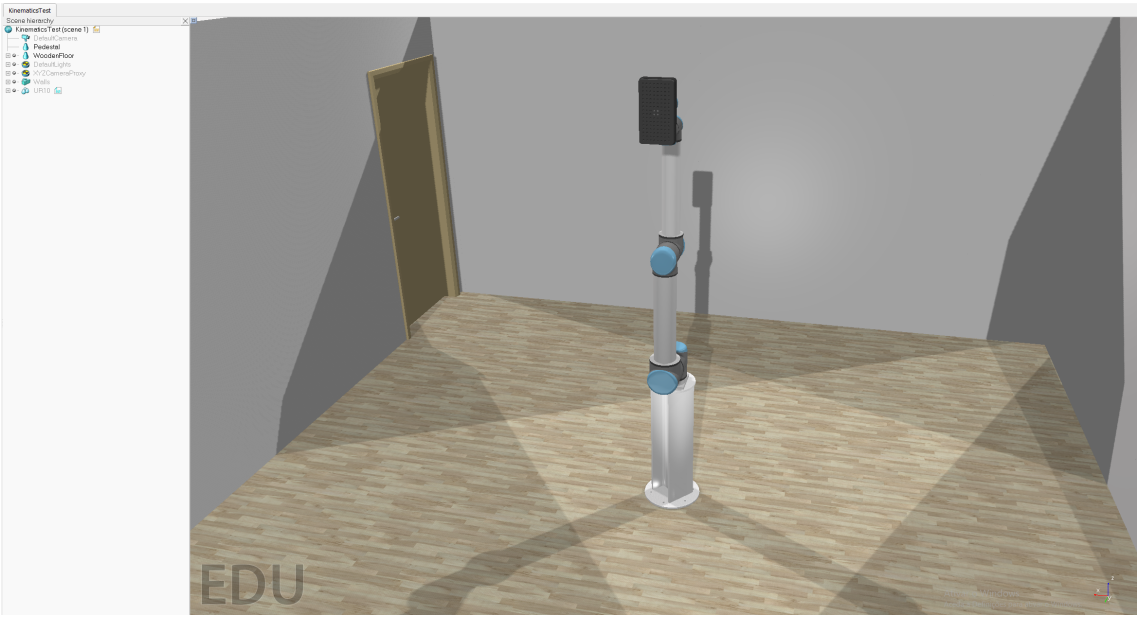



Figure 10: Scene setup and hierarchy

### 4.2.2 Embedded Scripts

In this scene there is only a threaded child script that is associated to the robot model. This script simply starts the MATLAB API remote connection, retrieves the handles of the robot and its joints, and has a while loop that is executed every five seconds where the robot's tip position is printed in the command window - this is useful to compare with our forward kinematics solution.

To execute the simulation launch the *KinematicsTest.ttt* scene and run it. Afterwards, open the *main.m* script, choose the target joint values and run it. You should see the robot move to the eight inverse kinematic solutions and obtain a result similar to figure 11. Comparing the end-effector position calculated in MATLAB we can see it is equal to the value presented in *CoppeliaSim*. Moreover, in the MATLAB command window it is possible to verify that there is one inverse kinematic solution that is equal to the pre-defined target joint values, and that all the other solutions produce the same end-effector position in *CoppeliaSim*. Follow the following link for a video of simulation execution.

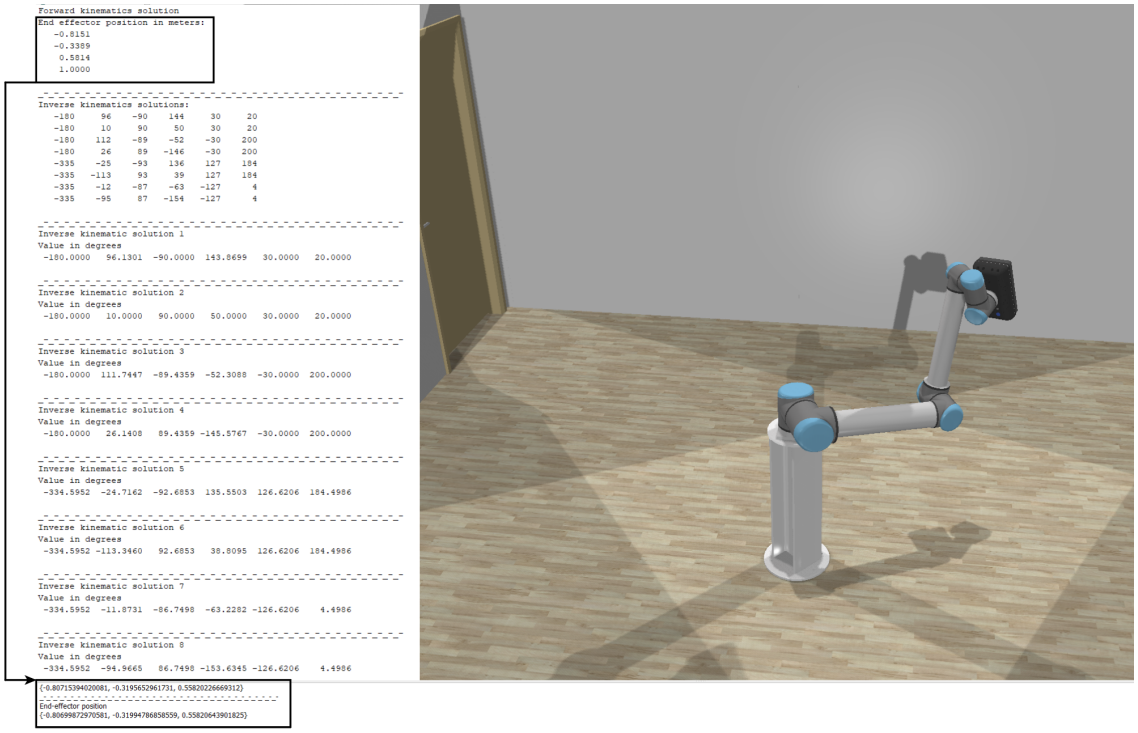



Figure 11: Launching the MATLAB files and the *CoppeliaSim* scene

# 5 Conclusion

This is a conclusion

# References


[1] Universal Robots. Parameters for calculations of kinematics and dynamics; 2020. Available from: https://www.universal-robots.com/articles/ur-articles/parameters-for-calculations-of-kinematics-and-dynamics/.

[2] Kebria PM, Al-Wais S, Abdi H, Nahavandi S. Kinematic and dynamic modelling of UR5 manipulator. 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings. 2017;p. 4229–4234.

[3] Andersen RS. Kinematics of a UR5. Aalborg University, May 2018. 2018;p. 1–12. Available from: http://rasmusan.blog.aau.dk/files/ur5{_}kinematics.pdf.

[4] Hawkins KP. Analytic Inverse Kinematics for the Universal Robots; 2013. p. 1–5. Available from: http://hdl.handle.net/1853/50782.

[5] Sun JD, Cao GZ, Li WB, Liang YX, Huang SD. Analytical inverse kinematic solution using the D-H method for a 6-DOF robot. 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2017. 2017;p. 714–716.

[6] Abdelaziz O, Luo M, Jiang G, Chen S. Multiple configurations for puncturing robot positioning. International Journal of Advance Robotics & Expert Systems (JARES). 2019;1(4):1–19. Available from: http://arxiv.org/abs/1903.02281.

[7] Jian BL, Tsai CS, Kuo YC, Guo YS. An image vision and automatic calibration system for universal robots. Journal of Low Frequency Noise Vibration and Active Control. 2019;.

[8] Oosterwyck NV. Real Time Human Robot Interactions and Speed Control of a Robotic Arm for Collaborative Operations; 2018.

[9] Chen S, Luo M, Abdelaziz O, Jiang G. A general analytical algorithm for collaborative robot (cobot) with 6 degree of freedom (DOF). Proceedings of the 2017 IEEE International Conference on Applied System Innovation: Applied System Innovation for Modern Technology, ICASI 2017. 2017;p. 698–701.

[10] Renfrew A. Book Review: Introduction to Robotics: Mechanics and Control. International Journal of Electrical Engineering & Education. 2004;41(4):388–388.

[11] Khatib O. Robot Manipulator Kinematics. Lecture Notes. 2018;Available from: http://www.ent.mrt.ac.lk/{~}rohan/teaching/ME5144/LectureNotes/Lec5Kinematics.pdf.

[12] Erlhagen EB. Kinematic analysis of an Industrial Serial Robot with 6 DoF. Lecture Notes. 2019;p. 1–11.

[13] Universal Robots. UR10e Working Area. Universal Robots Web Page. 2020;(100455).

[14] Ben-Ari M, Mondada F, Ben-Ari M, Mondada F. Kinematics of a Robotic Manipulator. Elements of Robotics. 2018;p. 267–291.

[15] Erlhagen EB. KUKA 6 DoF robot kinematic analysis. Lecture Notes. 2019;p. 1–10.

[16] Spong MW, Hutchinson S, Vidyasagar M. Robot modeling and control. IEEE Control Systems. 2006;26(6):113–115.

[17] Erlhagen EB. Planar Manipulator Robots with 2 DOFs : Forward and Inverse kinematics. Lecture Notes. 2019;p. 1–7.