Direct and Inverse Kinematics of Serial Manipulators

[1]Daniel Fortunato (81498), [2]Teodoro Dias (81723), [3]Pedro Santos (84162)

[1]Instituto Superior Técnico

March 29, 2019

Abstract — In this paper is presented the kinematic analysis of a robot arm with 6 revolute joints and, therefore, 6 DOF (degrees of freedom). Robot kinematics uses the geometry (position and orientation) of rigid bodies (links) and joints to control the movement of the robot. In this project, it is demonstrated the forward and inverse kinematics of a robot to control its movement. Forward kinematics calculates the *end-effector* position of the robot using the angles of the joints. Inverse kinematics calculates the angles of the joints with the *end-effector* position as the reference. In this study the Denavit- Hartenberg (D-H) model of representation was used to model links and joints. Both forward and inverse kinematics solutions for this manipulator were presented.

1 Introduction

Kinematics studies the movement of bodies without considering the forces that produce them, taking into account only the geometric characteristics of the robot. A kinematic model expresses the way in which the various components of a robot move between them, involving exclusively the variables related to the geometry of the robot. Densities related to dynamics, for example the mass of the robot, are not yet considered.

There are two types of kinematic problems: forward kinematics and inverse kinematics. The forward kinematics describes how to find the *end-effector* pose relative to the arm base for the given joint angles. On other hand, the inverse kinematics is based on finding the joint angles for the given pose of the *end-effector* with respect to the arm base.

The inverse kinematics plays an active role in object manipulation because it is an important issue to enable the arm *end-effector* to reach the desired object accurately. Also, there are other issues, which have to be taken into consideration when controlling the robotic arm, such as singularities, joint limits (that we do not take into account in this project) and reachable workspace. Generally, the inverse kinematics problem can be solved using two approaches: analytic and numeric. For this project it was used the analytical method and then divided into geometric and algebraic method.

Regarding notation, we started by calling the first angle θ_0 (base of the robot) and the last angle (*end-effector*) θ_5. So, in this paper the angles are θ_0, θ_1, θ_2, θ_3, θ_4, θ_5.

2 Direct Kinematics

The forward kinematics analysis means that the location and pose of the end of the manipulator in a given reference coordinate system can be worked out with the given geometry parameters of the links and the variables of the joints of the robot. This relationship can be derived using the Denavit-Hartenberg convention - an algorithm that achieves the kinematics of a chain of rigid bodies.

The initial step in the study of manipulator kinematics consists of specifying a set of frames that can capture all the movement characteristics of the manipulator. The basic idea for this method is to assign frames to each of the manipulator joints in order to capture the motion generated by them.

Each transformation between two frames uses a total of 4 parameters to define the role of each frame (Denavit-Hartenberg convention); 2 angles to describe the relative orientation between frames and 2 distances to describe their relative position. The main elements used by this convention are:

a_i - distance between Z_i and Z_{i+1} measured along X_i

α_i - angle between Z_i and Z_{i+1} measured around X_i

d_i - distance between X_{i-1} and X_i measured along Z_i

θ_i - angle between X_{i-1} and X_i measured around Z_i

Thus, the transformation matrix through which the coordinate system i can be represented in the frame of the coordinate system $i-1$ is given by (1):

$$T_{i-1}^{i} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The set must be able to capture all the physical dimensions of the robot in the four parameters used to describe each frame. The rest position that was used to specify the set of frames of this manipulator is represented in the figure (1).

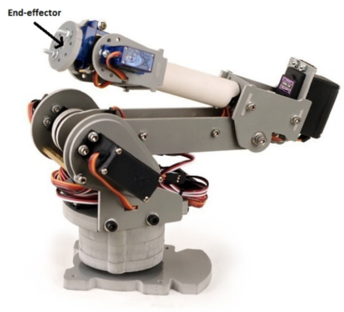

Figure 1: Initial position of the robot (Rest position - $\theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5 = 0$)

Instead of attributing only 6 frames for the manipulator, it was used 4 extra frames to do simple translations and rotations so there is only one rotation between frames and to make sure that there is no mistakes in the computation of the D-H parameters, in order to capture all the physical dimensions of the robot.

The representation of the coordinate frames chosen for the D-H convention are show in the figure (2) and the D-H parameters are represented in the table (1).

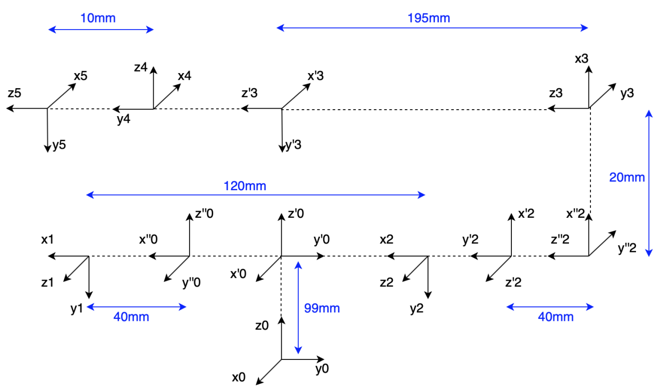

Figure 2: Coordinate frames of all the joints

Link i	a_{i-1}	α_{i-i}	d_i	θ_i
1	0	0	99	θ_0
2	0	0	0	$-\frac{\pi}{2}$
3	40	$-\frac{\pi}{2}$	0	θ_1
4	-120	0	0	θ_2
5	0	0	0	$-\frac{\pi}{2}$
6	0	$-\frac{\pi}{2}$	-40	0
7	20	0	0	θ_3
8	0	0	195	$\frac{\pi}{2}$
9	0	$\frac{\pi}{2}$	0	θ_4
10	0	$-\frac{\pi}{2}$	10	θ_5

Table 1: D-H parameters

Each line in the table (1) results on a transformation matrix between frames. Multiplying every matrix results on the final matrix that shows the homogeneous transformation between the initial frame and the *end-effector*. This multiplication is represented in the equation (2).

$$T_0^6 = T_0^1 * T_1^2 * T_2^3 * T_3^4 * T_4^5 * T_5^6 * T_6^7 * T_7^8 * T_8^9 * T_9^{10} \quad (2)$$

This matrix, because it represents an homogeneous transformation, can be represented as shown in equation (2).

$$T_0^6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

To discover the orientation between the initial frame and the *end-effector* it was used the Euler angles, in this case, in the Z-Y-Z method (as shown in the figure (3)).

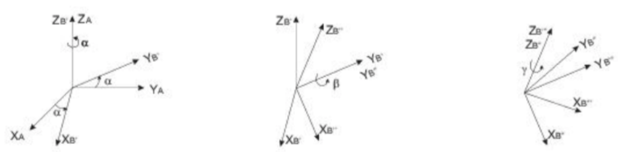

Figure 3: Z-Y-Z method

$$R_0^6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (4)$$

This method gives the orientation of the angles α, β, γ from the rotation matrix (R_0^6) in T_0^6.

From the Euler angles it is known that the next equality, in equation (5), is true.

$$R_0^6 = R_{ZYZ} = \begin{bmatrix} c_\alpha c_\beta c_\gamma - s_\alpha s_\gamma & -c_\alpha c_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta \\ s_\alpha c_\beta c_\gamma + c_\alpha s_\gamma & -s_\alpha c_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha c_\beta \\ -s_\beta c_\gamma & s_\beta c_\gamma & c_\beta \end{bmatrix} \quad (5)$$

From the equation (5), the angles α, β and γ can now be computed. The first angle to be found is β through the equation (6):

$$\beta = \pm \text{atan2}(\sqrt{r_{13}^2 + r_{23}^2}, r_{33}) \tag{6}$$

The α and γ are found with a condition refered to the β angle, as shown in the equations (7) and (8), respectively:

$$\alpha = \begin{cases} \text{atan2}(r_{23}, r_{13}), & \text{if } s_\beta > 0 \\ \text{atan2}(-r_{23}, -r_{13}), & \text{if } s_\beta < 0 \end{cases} \tag{7}$$

$$\gamma = \begin{cases} \text{atan2}(r_{32}, -r_{31}), & \text{if } s_\beta > 0 \\ \text{atan2}(-r_{32}, r_{31}), & \text{if } s_\beta < 0 \end{cases} \tag{8}$$

For the case when $s_\beta = 0$ it was used the equation (9):

$$\alpha + \gamma = \begin{cases} \text{atan2}(r_{21}, r_{11}), & \text{if } c_\beta = 1 \\ \text{atan2}(-r_{21}, -r_{11}), & \text{if } c_\beta = -1 \end{cases} \tag{9}$$

Putting $\alpha = 0$, γ can be computed through the equation (10):

$$\gamma = \begin{cases} \text{atan2}(r_{21}, r_{11}), & \text{if } c_\beta = 1 \\ \text{atan2}(-r_{21}, -r_{11}), & \text{if } c_\beta = -1 \end{cases} \tag{10}$$

2.0.1 Code Explanation

In our MATLAB file the function that refers to the direct kinematics is called *direct_kin.m*. It receives the 6 angles as input parameters (θ_0, θ_1, θ_2, θ_3, θ_4, θ_5) and gives the position and the orientation of the *end-effector* from the initial frame . It is also mandatory to have the function *Transform.m* that is a function that receives the D-H parameters as input and computes the transformation matrix of those parameters. We used this function to simplify the computation of the various transformation matrices that we had to compute for the direct and for the inverse kinematics.

Having both of this functions, to run the program for the direct kinematics, we just have to write *direct_kin* in the command window and put the angles that we want to attribute to each joint.

3 Inverse Kinematics

The transformation between the position and orientation space (x, y, z, α, β, γ) and the joint space is called the inverse kinematics.

The inverse kinematics represents the inverse transformation of the direct kinematics. The main characteristic of this transformation it is the fact that, in general, it has multiple solutions.

For manipulators which have six joints with the last three rotation axis intersecting at a point, it is possible to decouple the inverse kinematics problem into two simpler problems (arm and wrist).

The main approaches to compute the closed-form solutions for the inverse kinematics can be divided in two ways: (i) algebraic method and (ii) geometric method. The algebraic method is characterized by the algebraic manipulation of the frame transformations. The geometric method use basic trigonometric relationships to obtain relations involving the joint variables. In order to simplify the calculations, for this project in particular, it was used both of the methods. For the first three angles (θ_0, θ_1, θ_2) (arm) it was used the geometric approach and for the last angles (θ_3, θ_4, θ_5) (wrist) it was used the algebraic approach .

The input parameters of the inverse kinematics are the position and the orientation of the *end-effector*. In order to simplify the geometric method, as the location of the *end-effector* (x_6, y_6, z_6) is known, it is possible to find the position of the previous joint (x_5, y_5, z_5) through a translation of d_{10} (10mm) in the opposite direction of the z_6 axis, obtaining a position that only depends on the first three angles (θ_0, θ_1, θ_2).

3.0.1 Geometric Method

Looking for the manipulator from above it can be seen that the angle θ_0 it is only dependent of the position of the point (x_5, y_5, z_5), as shown in the figure (4). In other words, θ_0 is responsible for putting the robot arm in the direction of the point (x_5, y_5, z_5).

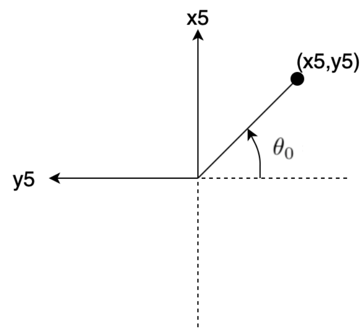

Figure 4: Scheme of the above view of the arm robot

In this particular project, in the rest position (that is represented in the figure (1)), the point of the joint 5 is in a space where $x_5 = 0$, $y_5 = -75$ and $z_5 = 119$. As we increase θ_0, the point (x_5, y_5, z_5) moves for the quadrant where x is positive but y is negative, so the solution for the angle θ_0 is given by the equation (11):

$$\theta_0 = \begin{cases} \text{atan2}(x_5, -y_5) \\ \text{atan2}(x_5, -y_5) + \pi \end{cases} \tag{11}$$

For the computation of the next angles it is also needed to compute another measure. This measure is the distance between the reference frame and the point (x_5, y_5) (seen from above), that is the hypotenuse formed by the cathets x_5 and y_5, and it is given by the equation (12).

$$\rho_5 = \sqrt{x_5^2 + y_5^2} \tag{12}$$

In order to find the next two angles (θ_1, θ_2) it was made a drawing of the manipulator seen from the side, as shown in the figure (5).

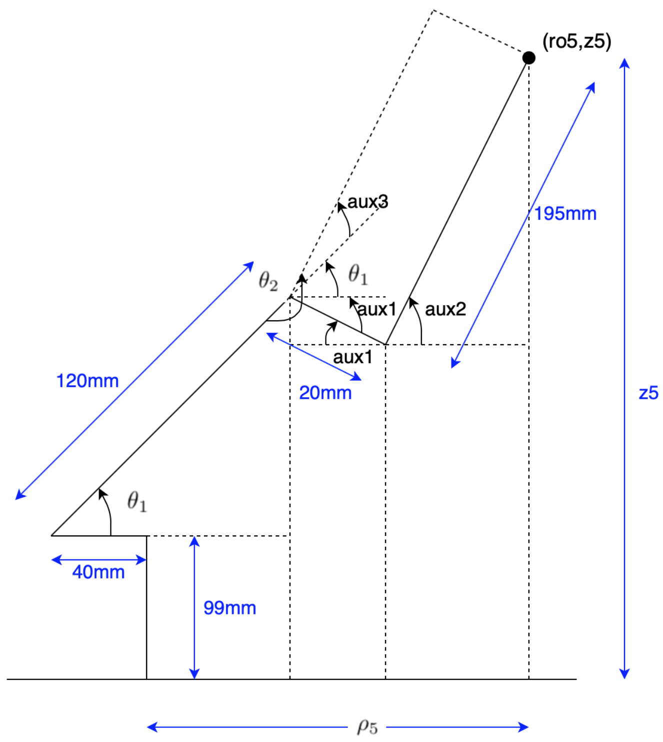

Figure 5: First scheme of the side view of the arm robot

Given z_5 and ρ_5 it can be used basic trigonometric relationships to obtain θ_1 and θ_2. From the figure (5) it can be easily seen that:

$$\begin{cases} \theta_2 = \pi + aux_3 \\ \frac{\pi}{2} = aux_1 + \theta_1 + aux_3 \\ aux_2 = \theta_1 + aux_3 \end{cases} \tag{13}$$

So,

$$\begin{cases} aux_1 = \frac{3\pi}{2} - \theta_1 - \theta_2 \\ aux_2 = \theta_1 + \theta_2 - \pi \end{cases} \tag{14}$$

And the equations that relate z_5 and ρ_5 with θ_1 and θ_2 are given by the equation system (15):

$$\begin{cases} z_5 = 99 + 120\sin(\theta_1) - 20\sin(aux_1) + 155\sin(aux_2) \\ \rho_5 = -40 + 120\cos(\theta_1) + 20\cos(aux_1) + 155\cos(aux_2) \end{cases} \tag{15}$$

As can be seen in the figure (5), this drawing just shows the solution when θ_0 is in the direction of the point (x_5, y_5, z_5). The drawing showing the solution when θ_0 is in

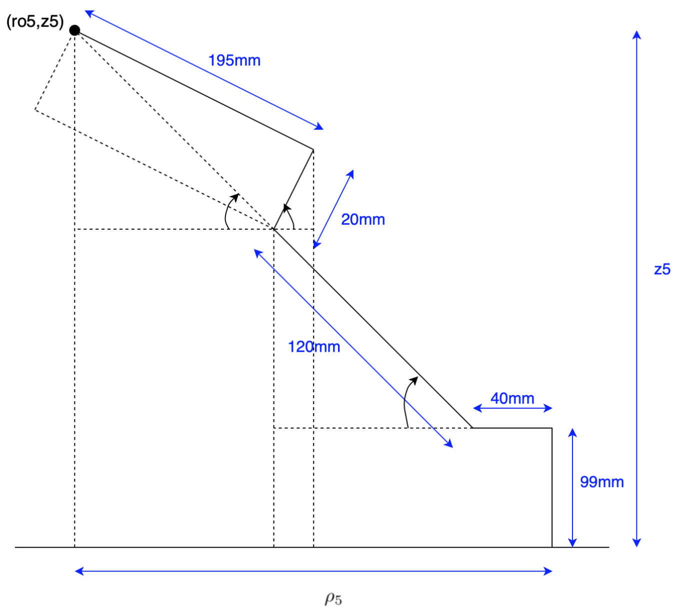

Figure 6: Second scheme of the side view of the arm robot

the reverse direction of the point (x_5, y_5, z_5) is represented in the figure (6).

For this case the equations that relate z_5 and ρ_5 with θ_1 and θ_2 are given by the equation system (16). The representation of this side is simplified because we already know the equations the equations that relate z_5 and ρ_5 with θ_1 and θ_2 from the previous scheme. The only difference is in the measuring of ρ_5, that is measured in the opposite direction.

$$\begin{cases} z_5 = 99 + 120\sin(\theta_1) - 20\sin(aux_1) + 155\sin(aux_2) \\ -\rho_5 = -40 + 120\cos(\theta_1) + 20\cos(aux_1) + 155\cos(aux_2) \end{cases} \tag{16}$$

Being:

$$\begin{cases} aux_1 = \frac{3\pi}{2} - \theta_1 - \theta_2 \\ aux_2 = \theta_1 + \theta_2 - \pi \end{cases} \tag{17}$$

It can be seen that the previous equations are constituted by sines and cosines, so, for each solution of θ_0, there are two solutions of θ_1 and θ_2. Because there are two solutions for θ_0 there are, in total, 4 combined solutions for these three angles.

3.0.2 Algebraic Method

Now that the first three angles are computed by the geometric method, the last three angles can be computed by the algebraic method.

With the first three angles the transformation matrix between the reference frame and the and the joint 3 (T_0^3) can be now computed, as shown in the equation (18)

$$T_0^3 = T_0^1 * T_1^2 * T_2^3 * T_3^4 * T_4^5 * T_5^6 \tag{18}$$

In order to simplify the algebraic method it was also included, in this multiplication, the auxiliary transformation matrices between the joint 3 and the joint 4. To compute all of these transformation matrices it was used the direct kinematics (using the D-H parameters) and the angles that were computed by the geometric method.

From the equation (2) it is know that:

$$T_0^6 = T_0^3 * T_3^6 \tag{19}$$

So, given the transformation matrix between the reference frame and the joint 3 (T_0^3), the transformation matrix between the joint 3 and the joint 6 (T_3^6) can be isolated in order to compute the angles that are left. And, because the transformation matrix between the reference frame and the *end-effector* (T_3^6) is known from the input parameters of the inverse kinematics, it is know that the equation (20) is true.

$$(T_0^3)^{-1} * T_0^6 = T_3^6 \tag{20}$$

The left side of the equation is a constant because it is known T_0^6 from the input parameters and $(T_0^3)^{-1}$ from the geometric method. So, in order to simplify the next algebraic manipulations, this multiplication can be substituted by T_{left}, as shown in the equation (21):

$$T_{left} = (T_0^3)^{-1} * T_0^6 \tag{21}$$

Looking at the table (1) and knowing the auxiliary matrices that were multiplied in the T_0^3 matrix it can be concluded that the transformation matrix that shows the transformation between the joint 3 and the joint 6 (T_3^6) is represented by the equation (22).

$$T_3^6 = T_6^7 * T_7^8 * T_8^9 * T_9^{10} \tag{22}$$

From the table (1) and from the direct kinematics the $T_6^7, T_7^8, T_8^9, T_9^{10}$ matrices can be algebraically calculated, as shown in the equations (23), (24), (25), (26), respectively. The matrix T_7^8 is an auxiliary matrix between the joint 4 and the joint 5.

$$T_6^7 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 20 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{23}$$

$$T_7^8 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 195 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{24}$$

$$T_8^9 = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{25}$$

$$T_9^{10} = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ -\sin\theta_5 & -\cos\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{26}$$

From here the initial procedure can be repeated in order to leave just one variable on the left. It was chosen first the variable θ_3 to put on left side of the equation, as shown in the equation (27).

$$(T_6^7)^{-1} * T_{left} = T_7^8 * T_8^9 * T_9^{10} \tag{27}$$

Inverting T_6^7 (28) and multiplying T_7^8, T_8^9, T_9^{10} (29), as shown bellow:

$$(T_6^7)^{-1} = \begin{bmatrix} \cos\theta_3 & \sin\theta_3 & 0 & -20\cos\theta_3 \\ -\sin\theta_3 & \cos\theta_3 & 0 & 20\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{28}$$

$$T_7^8 * T_8^9 * T_9^{10} = \begin{bmatrix} -s\theta_5 & -c\theta_5 & 0 & 0 \\ c\theta_4 c\theta_5 & -c\theta_4 s\theta_5 & -s\theta_4 & -10s\theta_4 \\ s\theta_4 c\theta_5 & -s\theta_4 s\theta_5 & c\theta_4 & 195 + 10c\theta_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{29}$$

Replacing the matrix T_{left} entries for variables, the result is:

$$T_{left} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{30}$$

Multiplying the third row of the $(T_6^7)^{-1}$ matrix with the third column of the T_{left} matrix and solving the equation (27), it is given the equation (31).

$$r_{33} = \cos\theta_4 \tag{31}$$

From this equation it is now known the solution for θ_4, as shown in the equation (32).

$$\theta_4 = [\arccos r_{33}, -\arccos r_{33}] \tag{32}$$

There are 2 solutions for θ_4 ($\arccos r_{33}$ and $-\arccos r_{33}$) and, therefore, 2 solutions for θ_4 for every combination of θ_0, θ_1, θ_2. In total, for every angle that was already computed, there are 8 solutions (not counting with singularities) for every point that is reachable by the robot arm.

Placing the matrix T_6^7 again on the right side of the the equation (27) we have:

$$T_{left} = T_6^7 * T_7^8 * T_8^9 * T_9^{10} \tag{33}$$

Being, $T_{left} = (T_0^3)^{-1} * T_0^6$.

The multiplication of the $T_6^7, T_7^8, T_8^9, T_9^{10}$ is given by the equation (34).

$$T_6^7 * T_7^8 * T_8^9 * T_9^{10} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{34}$$

Where,

$$\begin{cases} k_{11} = -c\theta_3 s\theta_5 - s\theta_3 c\theta_4 c\theta_5 \\ k_{12} = -c\theta_3 c\theta_5 + s\theta_3 c\theta_4 s\theta_5 \\ k_{13} = s\theta_3 s\theta_4 \\ k_{14} = 10 s\theta_3 s\theta_4 + 20 \\ k_{21} = -s\theta_3 s\theta_5 + c\theta_3 c\theta_4 c\theta_5 \\ k_{22} = -s\theta_3 c\theta_5 - c\theta_3 c\theta_4 s\theta_5 \\ k_{23} = -c\theta_3 s\theta_4 \\ k_{24} = -10 c\theta_3 s\theta_4 \\ k_{31} = s\theta_4 c\theta_5 \\ k_{32} = -s\theta_4 s\theta_5 \\ k_{33} = c\theta_4 \\ k_{34} = 195 + 10 c\theta_4 \end{cases} \tag{35}$$

So, to find θ_3 and θ_5 it is needed to solve the equation (36):

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{36}$$

Knowing θ_4, to find θ_3 we can look at the k_{13} and k_{23} entries of the right sided matrix.

$$\begin{cases} r_{13} = k_{13} = s\theta_3 s\theta_4 \\ r_{23} = k_{23} = -c\theta_3 s\theta_4 \end{cases} \tag{37}$$

Dividing r_{13} by r_{23} we have in the equation (38):

$$\frac{r_{13}}{r_{23}} = \frac{s\theta_3 s\theta_4}{-c\theta_3 s\theta_4} = \frac{s\theta_3}{-c\theta_3} \tag{38}$$

We can finally compute the solution of θ_3 through the equation (39).

$$\theta_3 = \text{atan2}(\frac{r_{13}}{-r_{23}}) \tag{39}$$

It can be done the same thing for the angle of the *end-effector* (θ_5) but now looking at the k_{32} and k_{31} entries of the right sided matrix.

$$\begin{cases} r_{32} = k_{32} = -s\theta_4 s\theta_5 \\ r_{31} = k_{31} = s\theta_4 c\theta_5 \end{cases} \tag{40}$$

Dividing r_{32} by r_{31} we have in the equation (41):

$$\frac{r_{32}}{r_{31}} = \frac{-s\theta_4 s\theta_5}{s\theta_4 c\theta_5} = \frac{-s\theta_5}{c\theta_5} \tag{41}$$

We can finally compute the solution of θ_5 through the equation (42).

$$\theta_5 = \text{atan2}(\frac{-r_{32}}{r_{31}}) \tag{42}$$

Now that we have all the angles computed we can conclude that, in total, there are 8 solutions (not counting with singularities) for every point that is reachable by the robot arm. Two solutions for θ_0, two solutions for θ_1 and θ_2 for each θ_0, two solutions for θ_4 for each combination of θ_0, θ_1 and θ_2, and, finally, one solution for θ_3 and θ_5 for each solution of θ_4.

3.0.3 Code Explanation

In our MATLAB file the function that refers to the inverse kinematics is called *inverse_kin.m*. It receives the 6 angles as input parameters the position and the orientation of the *end-effector* (x, y, z, α, β, γ) and gives the value that all the joints must have to reach that position and that orientation. It is also mandatory to have the function *Transform_ZYZ.m* that is a function that receives the position and the orientation of the *end-effector* as input and computes the transformation matrix between the reference frame and the *end-effector*. We used this function to compute the transformation from the initial frame to the last frame in order to transform the last point in the point (x_5, y_5, z_5), so we could do the geometric method in a simplified way.

Having both of this functions, to run the program for the inverse kinematics, we just have to write *inverse_kin* in the command window and put the position and the orientation of the *end-effector*. The function will print a matrix 6x8 with all the angle solutions for every joint. It also prints the direct kinematics for all the solutions so we could confirm that all them were correct.

4 Singularities

Singularities in robotics are caused by the inverse kinematics of the robot. When placed at a singularity there may be an infinite number of ways for the kinematics to achieve the same tip position/orientation of the robot. Another possible definition for singularities are the configurations of the robot where it looses one or more degrees of freedom - the robot looses mobility and we can't impose an arbitrary movement on the end efector. In order to check for the existence of singularities we use as a tool the Jacobian matrix, which maps the velocity of the joints θ_i to the velocity space of the Cartesian coordinates (x,y,z) and Euler coordinates (α, β, γ):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{v} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = J(\theta) \begin{bmatrix} \dot{\theta}_0 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{bmatrix} \tag{43}$$

The Jacobian can be separated in two matrixces, J_P and J_o. J_P is the linear velocity jacobian, which give us the position singularities (corresponds to the first 3 lines of the Jacobian) and J_o is the angular velocity jacobian, which give us the orientation singularities (corresponds to the last 3 lines of the jacobian). This way, we can resume the structure of the Jacobian in Figure7 :

$$J = \begin{bmatrix} J_P \\ J_o \end{bmatrix}$$

Figure 7: Jacobian Matrix Structure

The terms of the Linear Velocity Jacobian (J_P) can be determined by the differentiation of the direct kinematics equations. In direct kinematics we reach to a transformation matrix that relates the efector with the base of the robotic arm. The first 3 members of the last column of this transformation matrix give us the position of the efector (x, y, z). Differentiating each of this positions in order of each theta we can get the values of J_P, where each line of the matrix corresponds to a position derivated by all thetas. We can resume this in the following Figure:

$$\begin{bmatrix} \delta y_1 \\ \vdots \\ \delta y_n \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \vdots \\ \delta x_m \end{bmatrix}$$

Figure 8: Linear Velocity Jacobian Determination

δy_n corresponds to the derivative of each position and δx_m corresponds to the derivative of the θ's.

The terms of the Angular Velocity Jacobian (J_o) can be determined by applying the following equation:

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R \, {}^{i}\omega_i + \dot{\theta}_{i+1} \, {}^{i+1}Z_{i+1}$$

Figure 9: Angular Velocity Jacobian Terms Determination

Finally, we can resume that the Jacobian matrix terms will be determined in the following way:

$$J = \begin{pmatrix} \frac{\partial}{\partial q_1}({}^0x_P) & \frac{\partial}{\partial q_2}({}^0x_P) & \cdots & \frac{\partial}{\partial q_n}({}^0x_P) \\ \overline{\epsilon}_1.({}^0_1R.Z) & \overline{\epsilon}_2.({}^0_2R.Z) & \cdots & \overline{\epsilon}_n.({}^0_nR.Z) \end{pmatrix}$$

Figure 10: Jacobian Matrix Determination

We can say that when one of the lines of the Jacobian is zero we are loosing degrees of freedom which is equal to saying that the singularities are discovered by making

$$det(J(\theta)) = 0 \tag{44}$$

But since these 2 sub matrices (J_P and J_o) are now 3x6 we need to do combinations of the 6 columns in order to make 3x3 matrices to discover all the singularities which gives us 10 cases to check for position singularities (since θ_5 does not influence the position of the end efector) and 20 for the orientation singularities (5 combinations 3 by 3 and 6 combinations 3 by 3 respectively).

As a final introductory remark we would like to point out that this type of arm robot exhibits three types of singularities: wrist singularities, shoulder singularities, and elbow singularities.

A wrist singularity occurs when the rotation axis of θ_3 and θ_5 are coincident(which happens when $\theta_4 = 0$) and the effect is that we can turn θ_5 to any angle and as long as $\theta_5 = -\theta_3$ the arm will always have the same position and orientation.

A shoulder singularity occurs when the wrist center lies on a cylinder centered about axis of θ_0 and with a radius equal to the distance between axes of θ_0 and θ_4 (which happens for example when the position of the end efector has x = y = 0).

Finally an elbow singularity occurs when the wrist center lies in the same plane as axes of theta 1 and theta 2 (which happens with an outstretched arm).

4.1 Position singularities

The obtained Jacobian matrix for the linear velocities is in the appendix of this report. Since we have 10 determinants to analyse for the presence of singularities we only present here the simpler/most important ones. Is important to note that several determinants present the same singularities.

By using a matrix composed by the columns 1,4 and 5 we get the expression:

$$\begin{aligned} 500cos(\theta_3)(0.5cos(2\theta_4) - 0.5)(31.3cos(\theta_1 + \theta_2 - 0.128) \\ - 24cos(\theta_1) + 8) = 0 \end{aligned} \tag{45}$$

This gives us 3 values for possible singularities of the systems, namely:

$$500cos(\theta_3) = 0 \leftrightarrow \theta_3 = \frac{\pi}{2} + k\pi, k \in \mathbb{Z} \tag{46}$$

$$(0.5cos(2\theta_4) - 0.5) = 0 \leftrightarrow \theta_4 = k\pi, k \in \mathbb{Z} \tag{47}$$

Where the first one can be explained by the fact that the end efector of the arm is always in the same plane so we cant obtain any point outside of this plane, which leads to reduced mobility (singularity), and the second one is the wrist singularity previously discussed. The last one determines the necessary relationships between θ_1 and θ_2 in order to have the arm outstretched.

Another determinant that is important to analyse is the one made by building the matrix using the columns 2,3 and 5:

$$6000cos(\theta_3)(cos(\theta_4))(4cos(\theta_2) - 31sen(\theta_2) - 2cos(\theta_4)sen(\theta_2) + 2cos(\theta_2)sen(\theta_3)sen(\theta_4)) = 0 \quad (48)$$

In this case the first 2 terms of the multiplication give us a repeated result for θ_3 and a new one for θ_4:

$$\theta_4 = \frac{\pi}{2} + k\pi, k \in \mathbb{Z} \quad (49)$$

Whereas the last equation gives us the relationships that need to be respected between θ_2, θ_4 and θ_3 in order for the end efector to be in the $x = y = 0$ position.

4.2 Orientation singularities

The angular velocity matrix is also in the appendix ate the end of the report. The method to obtain the singularities is the same as the one that we made for the position singularities. If we select the columns 1,4 and 6 we obtain the determinant:

$$cos(\theta_1 + \theta_2)cos(\theta_3)sen(\theta_4) = 0 \quad (50)$$

Which gives us the results for the angles of:

$$\theta_1 + \theta_2 = \frac{\pi}{2} + k\pi, k \in \mathbb{Z} \quad (51)$$

$$\theta_4 = k\pi, k \in \mathbb{Z} \quad (52)$$

$$\theta_3 = \frac{\pi}{2} + k\pi, k \in \mathbb{Z} \quad (53)$$

Where the first one gives us the condition for an outstretched arm, the second for a wrist singularity and the last one for the elbow singularity. These singularities can also be reached with different angle combinations as seen by the combinations of lines 1,2 and 5 for example:

$$sen(\theta_1 + \theta_2)cos(\theta_3) = 0 \quad (54)$$

And there are also combinations of position singularity configurations that also are valid for orientation singularities, as seen from the above expressions and the solutions for the expressions made from the set of 2,4,6 and 3,5,6 for example:

$$sen(\theta_3)sen(\theta_4) = 0 \quad (55)$$

$$cos(\theta_3)cos(\theta_4) = 0 \quad (56)$$

4.2.1 Code Explanation

To determine the Jacobian Matrix and the determinants of the Linear Velocity Jacobian (J_P) and Angular Velocity Jacobian (J_o) (for determination of position and angular singularities, respectively), a MATLAB program was developed (jacobianDetermination.m).

The function *jacobianDetermination* recieves as an input 3 column numbers of the Jacobian Matrix. This column numbers are used to create the submatrices, for the determinant determination. It outputs in the console (prints) the Jacobian Matrice and the equation corresponding to the determinants (J - Jacobian matrix, detjp - determinant of linear velocity submatrix, detjr - determinant of angular velocity submatrix). The function also returns this values (J, detjp and detjr).

To run the program, simply call the function (jacobianDetermination) and give the column numbers as input.

5 Results

The results for the direct kinematics are in the table (2) and the results for the inverse kinematics are in the table (3) to (12) in section (A) and (B) of the Appendix.

We started by testing the direct kinematics first with some angles that were easy to confirm if the program was working without errors and to confirm if the position and the orientation given in the output were right, and then moved to some more difficult ones, as seen in the table (2).

To test the inverse kinematics we put in the input parameters the output of the direct kinematics so we could confirm that the angles given in the output of the inverse kinematics were write. Also, as mentioned in the code explanation, we ran the direct kinematics for all the solutions of the inverse so we could check, once again, if the solutions that the program gave us were correct.

As can be seen in the section (A), in the table (2), almost in all the tests, θ_4 had the value zero. That was intentional. We know that θ_3 and θ_5 rotate in the same direction. So, if the θ_4 is zero, there are infinite combinations between θ_3 and θ_5 to reach one orientation. What we did for this case, instead of printing that are infinite solutions for the orientation of θ_3 and θ_5, we forced θ_3 to go to zero and the only angle that changes is θ_5. This only happens when θ_4 is zero.

Also, in the section (B) there are some tests that have only 4 solutions instead of 8. That is due to the geometry of the robot. As can be seen in the figures (5) and (6) there is a 40 mm segment that can restrict a certain position not to reach the point that was put in the input parameters. We can give the example of our own arms. If we stretch our right arm as much as possible for the right side we reach a point that cannot be reached by the left arm. That is what happened to the to the tests in section (B) that only have 4 solutions: a point could be reached if the arm is fully stretched for one side, but when θ_0 rotates π that point cannot be reached because of the 40mm segment.

6 References

[1] Mark W. Spong, Seth Hutchinson, M. Vidyasagar (2nd edition), Robot Dynamics and Control

[2] João Silva Sequeira, Introdução à Robótica (cap.1-6)

[3] Lecture Notes, EN3562, Department of Electronic Telecommunication Engineering, University of Moratuwa

7 Appendix

A Results of the Direct Kinematics

Test i	θ_0, θ_1, θ_2, θ_3, θ_4, θ_5	Solution (x, y, z, α, β, γ)
1	0, 0, 0, 0, 0, 0	0, -85, 119, $-\frac{\pi}{2}$, $\frac{\pi}{2}$, $-\frac{\pi}{2}$
2	$\frac{\pi}{2}$, 0, 0, 0, 0, 0	85, 0, 119, 0, $\frac{\pi}{2}$, $-\frac{\pi}{2}$
3	$\frac{\pi}{2}$, 0, 0, 0, 0, $\frac{\pi}{2}$	85, 0, 119, 0, $\frac{\pi}{2}$, 0
4	0, $\frac{\pi}{2}$, $\frac{\pi}{2}$, 0, 0, 0	0, 125, 199, $\frac{\pi}{2}$, $\frac{\pi}{2}$, $\frac{\pi}{2}$
5	0, $\frac{\pi}{2}$, $\frac{\pi}{2}$, $\frac{\pi}{2}$, 0, $\frac{\pi}{2}$	0, 125, 199, $\frac{\pi}{2}$, $\frac{\pi}{2}$, $-\frac{\pi}{2}$
6	$-\frac{\pi}{7}$, $\frac{\pi}{9}$, $\frac{5\pi}{4}$, 0, 0, $-\frac{7\pi}{9}$	69.6910, 144.7148, 281.1308, 1.1220, 0.4363, -0.8727
7	π, $\frac{\pi}{2}$, $\frac{\pi}{2}$, $\frac{\pi}{2}$, 0, $\frac{\pi}{2}$	0, -125, 199, $-\frac{\pi}{2}$, $\frac{\pi}{2}$, $-\frac{\pi}{2}$
8	$\frac{\pi}{2}$, π, π, 0, 0, $\frac{\pi}{2}$	325, 0, 119, 0, $\frac{\pi}{2}$, 0
9	0, π, π, 0, 0, 0	0, -325, 119, $-\frac{\pi}{2}$, $\frac{\pi}{2}$, $-\frac{\pi}{2}$
10	$\frac{\pi}{2}$, $\frac{\pi}{2}$, $\frac{\pi}{2}$, $\frac{\pi}{2}$, $\frac{\pi}{2}$, $\frac{\pi}{2}$	-115, 0, 189, 0, π, $\frac{\pi}{2}$

Table 2: Results for the Direct Kinematics

B Results of the Inverse Kinematics

B.0.1 Test 1

x, y, z, α, β, γ = (0, -85, 119, $-\frac{\pi}{2}$, $\frac{\pi}{2}$, $-\frac{\pi}{2}$)

θ	sol_1	sol_2	sol_3	sol_4	sol_5	sol_6	sol_7	sol_8
θ_0	0	0	0	0	3.1416	3.1416	3.1416	3.1416
θ_1	0	-1.0383	0	-1.0383	-1.2697	1.6141	-1.2697	1.6141
θ_2	0	0.2566	0	0.2566	-0.7058	0.9624	-0.7058	0.9624
θ_3	0	-1.5708	0	1.5708	1.5708	-1.5708	-1.5708	1.5708
θ_4	0	0.7816	0	-0.7816	1.1661	0.5651	-1.1661	-0.5651
θ_5	0	1.5708	0	-1.5708	1.5708	-1.5708	-1.5708	1.5708

Table 3: Results of test 1 for the Inverse Kinematics

B.0.2 Test 2

x, y, z, α, β, γ = (85, 0, 119, 0, $\frac{\pi}{2}$, $-\frac{\pi}{2}$)

θ	sol_1	sol_2	sol_3	sol_4	sol_5	sol_6	sol_7	sol_8
θ_0	1.5708	1.5708	1.5708	1.5708	4.7124	4.7124	4.7124	4.7124
θ_1	0	-1.0383	0	-1.0383	-1.2697	1.6141	-1.2697	1.6141
θ_2	0	0.2566	0	0.2566	-0.7058	0.9624	-0.7058	0.9624
θ_3	0	-1.5708	0	1.5708	1.5708	-1.5708	-1.5708	1.5708
θ_4	0	0.7816	0	-0.7816	1.1661	0.5651	-1.1661	-0.5651
θ_5	0	1.5708	0	-1.5708	1.5708	-1.5708	-1.5708	1.5708

Table 4: Results of test 2 for the Inverse Kinematics

B.0.3 Test 3

$x,\, y,\, z,\, \alpha,\, \beta,\, \gamma = (85,\, 0,\, 119,\, 0,\, \frac{\pi}{2},\, 0)$

θ	sol_1	sol_2	sol_3	sol_4	sol_5	sol_6	sol_7	sol_8
θ_0	1.5708	1.5708	1.5708	1.5708	4.7124	4.7124	4.7124	4.7124
θ_1	0	-1.0383	0	-1.0383	-1.2697	1.6141	-1.2697	1.6141
θ_2	0	0.2566	0	0.2566	-0.7058	0.9624	-0.7058	0.9624
θ_3	0	-1.5708	0	1.5708	1.5708	-1.5708	-1.5708	1.5708
θ_4	0	0.7816	0	-0.7816	1.1661	0.5651	-1.1661	-0.5651
θ_5	1.5708	-3.1416	1.5708	0.0000	3.1416	0.0000	-0.0000	-3.1416

Table 5: Results of test 3 for the Inverse Kinematics

B.0.4 Test 4

$x,\, y,\, z,\, \alpha,\, \beta,\, \gamma = (0,\, 125,\, 199,\, \frac{\pi}{2},\, \frac{\pi}{2},\, \frac{\pi}{2})$

θ	sol_1	sol_2	sol_3	sol_4	sol_5	sol_6	sol_7	sol_8
θ_0	-3.1416	-3.1416	-3.1416	-3.1416	0	0	0	0
θ_1	0.8313	-2.6859	0.8313	-2.6859	1.5708	-0.4249	1.5708	-0.4249
θ_2	-0.7756	1.0323	-0.7756	1.0323	1.5708	-1.3141	1.5708	-1.3141
θ_3	1.5708	-1.5708	-1.5708	1.5708	0	1.5708	0	-1.5708
θ_4	0.0557	1.6536	-0.0557	-1.6536	0	1.4026	0	-1.4026
θ_5	1.5708	-1.5708	-1.5708	1.5708	0	-1.5708	0	1.5708

Table 6: Results of test 4 for the Inverse Kinematics

B.0.5 Test 5

$x,\, y,\, z,\, \alpha,\, \beta,\, \gamma = (0,\, 125,\, 199,\, \frac{\pi}{2},\, \frac{\pi}{2},\, -\frac{\pi}{2})$

θ	sol_1	sol_2	sol_3	sol_4	sol_5	sol_6	sol_7	sol_8
θ_0	-3.1416	-3.1416	-3.1416	-3.1416	0	0	0	0
θ_1	0.8313	-2.6859	0.8313	-2.6859	1.5708	-0.4249	1.5708	-0.4249
θ_2	-0.7756	1.0323	-0.7756	1.0323	1.5708	-1.3141	1.5708	-1.3141
θ_3	1.5708	-1.5708	-1.5708	1.5708	0	1.5708	0	-1.5708
θ_4	0.0557	1.6536	-0.0557	-1.6536	0	1.4026	0	-1.4026
θ_5	-1.5708	1.5708	1.5708	-1.5708	-3.1416	1.5708	-3.1416	-1.5708

Table 7: Results of test 5 for the Inverse Kinematics

B.0.6 Test 6

$x,\, y,\, z,\, \alpha,\, \beta,\, \gamma = (69.6910,\, 144.7148,\, 281.1308,\, 1.1220,\, 0.4363,\, \text{-}0.8727)$

θ	sol_1	sol_2	sol_3	sol_4	sol_5	sol_6	sol_7	sol_8
θ_0	2.6928	2.6928	2.6928	2.6928	5.8344	5.8344	5.8344	5.8344
θ_1	1.3246	3.0011	1.3246	3.0011	0.3491	1.0956	0.3491	1.0956
θ_2	-1.5674	1.8241	-1.5674	1.8241	-2.3562	2.6128	-2.3562	2.6128
θ_3	1.5708	-1.5708	-1.5708	1.5708	0	-1.5708	0	1.5708
θ_4	0.8916	0.3235	-0.8916	-0.3235	0.0000	0.5676	-0.0000	-0.5676
θ_5	-0.8727	2.2689	2.2689	-0.8727	-2.4435	-0.8727	-2.4435	2.2689

Table 8: Results of test 6 for the Inverse Kinematics

B.0.7 Test 7

$x, y, z, \alpha, \beta, \gamma = (0, -125, 199, -\frac{\pi}{2}, \frac{\pi}{2}, -\frac{\pi}{2})$

θ	sol_1	sol_2	sol_3	sol_4	sol_5	sol_6	sol_7	sol_8
θ_0	-0.0000	-0.0000	-0.0000	-0.0000	3.1416	3.1416	3.1416	3.1416
θ_1	0.8313	-2.6859	0.8313	-2.6859	1.5708	-0.4249	1.5708	-0.4249
θ_2	-0.7756	1.0323	-0.7756	1.0323	1.5708	-1.3141	1.5708	-1.3141
θ_3	1.5708	-1.5708	-1.5708	1.5708	0	1.5708	0	-1.5708
θ_4	0.0557	1.6536	-0.0557	-1.6536	0	1.4026	0	-1.4026
θ_5	-1.5708	1.5708	1.5708	-1.5708	-3.1416	1.5708	-3.1416	-1.5708

Table 9: Results of test 7 for the Inverse Kinematics

B.0.8 Test 8

$x, y, z, \alpha, \beta, \gamma = (325, 0, 119, 0, \frac{\pi}{2}, 0)$

θ	sol_1	sol_2	sol_3	sol_4	sol_5	sol_6	sol_7	sol_8
θ_0	1.5708	1.5708	1.5708	1.5708	4.7124	4.7124	4.7124	4.7124
θ_1	3.1416	2.9964	3.1416	2.9964	NaN	NaN	NaN	NaN
θ_2	3.1416	-2.8849	3.1416	-2.8849	NaN	NaN	NaN	NaN
θ_3	0	1.5708	0	-1.5708	NaN	NaN	NaN	NaN
θ_4	0	0.1114	0	-0.1114	NaN	NaN	NaN	NaN
θ_5	1.5708	-0.0000	1.5708	3.1416	NaN	NaN	NaN	NaN

Table 10: Results of test 8 for the Inverse Kinematics

B.0.9 Test 9

$x, y, z, \alpha, \beta, \gamma = (0, -325, 119, -\frac{\pi}{2}, \frac{\pi}{2}, -\frac{\pi}{2})$

θ	sol_1	sol_2	sol_3	sol_4	sol_5	sol_6	sol_7	sol_8
θ_0	-0.0000	-0.0000	-0.0000	-0.0000	3.1416	3.1416	3.1416	3.1416
θ_1	3.1416	2.9964	3.1416	2.9964	NaN	NaN	NaN	NaN
θ_2	3.1416	-2.8849	3.1416	-2.8849	NaN	NaN	NaN	NaN
θ_3	0	1.5708	0	-1.5708	NaN	NaN	NaN	NaN
θ_4	0	0.1114	0	-0.1114	NaN	NaN	NaN	NaN
θ_5	0.0000	-1.5708	0.0000	1.5708	NaN	NaN	NaN	NaN

Table 11: Results of test 9 for the Inverse Kinematics

B.0.10 Test 10

$x, y, z, \alpha, \beta, \gamma =$ (-115, 0, 189, 0, π, $\frac{\pi}{2}$)

θ	sol_1	sol_2	sol_3	sol_4	sol_5	sol_6	sol_7	sol_8
θ_0	-1.5708	-1.5708	-1.5708	-1.5708	1.5708	1.5708	1.5708	1.5708
θ_1	0.8313	-2.6859	0.8313	-2.6859	1.5708	-0.4249	1.5708	-0.4249
θ_2	-0.7756	1.0323	-0.7756	1.0323	1.5708	-1.3141	1.5708	-1.3141
θ_3	-1.5708	1.5708	1.5708	-1.5708	1.5708	1.5708	-1.5708	-1.5708
θ_4	1.5151	3.0588	-1.5151	-3.0588	1.5708	2.9734	-1.5708	-2.9734
θ_5	1.5708	-1.5708	-1.5708	1.5708	1.5708	1.5708	-1.5708	-1.5708

Table 12: Results of test 10 for the Inverse Kinematics

C
Linear Velocity Jacobian Matrix

$$J_p = \begin{bmatrix} 10c_0\left(xc_4 - 12c_1 + x + 2y + 4 + s_{34}y\right) - 10c_3s_{04} & -5s_0\left(31y - 4x - 24s_1 + 2c_4y - 2s_{34}y\right) & -5s_0\left(31y - 4x + 2c_4y - 2s_{34}x\right) & -10c_4\left(c_0s_3 - c_3s_0y\right) & 10c_4\left(c_{03} + s_{03}y\right) - 10s_04x & 0 \\ 5s_0\left(2xc_4 - 24c_2 + 2ys_{34} + 31x + 4y + 8\right) + 10c_{03}s_4 & 5c_0\left(31y - 4x - 24s_1 + 2c_4y + 2s_{34}y\right) & 5c_0\left(31y - 4x + 2c_4y - 2s_{34}x\right) & -10c_4\left(s_{03} + c_{03}y\right) & 10c_4\left(c_3s_0 - s_0c_0y\right) - 10s_4c_0x & 0 \\ 0 & 5\left(24c_1 - 2xc_4 - 2ys_{34} - 31x - 4y\right) & -5\left(2xc_4 + 2ys_{34} + 31x + 4y\right) & 10xc_3s_4 & 10\left(ys_4 + xc_4s_3\right) & 0 \end{bmatrix}$$

$$\text{x} = \cos(\theta_1 + \theta 2)$$
$$\text{y} = \sin(\theta_1 + \theta 2)$$

Figure 11: Determined Linear Velocity Jacobian

D
Angular Velocity Jacobian

$$J_o = \begin{bmatrix} 0 & c_0 & c_0 & cos\left(\theta_1 + \theta_2\right)s_0 & c_3\left(c_1s_{02} + c_2s_{01}\right) - c_0s_3 & s_4\left(c_{03} + s_3\left(c_1s_{02} + c_2s_{01}\right)\right) - c_4\left(s_{012} - c_{12}s_0\right) \\ 0 & s_0 & s_0 & -cos\left(\theta_1 + \theta_2\right)c_0 & -s_{03} - c_3\left(c_{01}s_2 + c_{02}s_1\right) & s_4\left(c_3s_0 - s_3\left(c_{01}s_2 + c_{02}s_1\right)\right) + c_4\left(c_0s_{12} - c_{012}\right) \\ 1 & 0 & 0 & -sin\left(\theta_1 + \theta_2\right) & cos\left(\theta_1 + \theta_2\right)c_3 & cos\left(\theta_1 + \theta_2\right)s_{34} - sin\left(\theta_1 + \theta_2\right)c_4 \end{bmatrix}$$

Figure 12: Determined Angular Velocity Jacobian