Faculty of Power and Aeronautical Engineering
Department of Robotics

# Group Project Report

**Adhvaith Ramkumar,**
**Ahmad Masood,**
**Jafar Jafar,**
**Pratama Aji Nur Rochman**

supervised by
mgr Inż. Maksymilian Szumowski

# Contents

# List of Figures

## List of Tables

# 1 Manipulator description and structure

## 1.1 Robot manipulator

The mechanical structure of the robot manipulator is composed of arms (links) that are interconnected through joints (joints). The manipulator is characterized by an arm that ensures mobility and a wrist that is given speed and the end-effector performs the tasks assigned to the robotic manipulator.

A connection between two links can be formed from either joint. Each joint provides a structure with one degree of freedom (DOF).

DOF is the number of joints in determining the DOF of the manipulator. In particular, the manipulator has at least 6-DOF, three for position and three for orientation. If it is less than 6-DOF then the arm cannot reach every point with the desired orientation.

The manipulator workspace states that the end-effector manipulator environment is reachable. The working area is limited by the geometry of the manipulator as well as mechanical constraints on the joint manipulator robot. An example of a robot manipulator used in this project can be seen in Figure 1.

Figure 1: Robot manipulator

The principle of movement of the robot manipulator which has 6-DOF is divided into two parts, namely the movement to determine the position of the robot and determine the orientation of the end of the robot manipulator. The position of the robot is determined based on the movement of the first 3 joints, while determining the orientation is based on the next 3 joints. The entire joint of the robot manipulator is a revolute joint, which means that the most likely joint movement is rotation.

## 1.2 Parts of robot manipulator

The robot manipulator consists of two core parts, namely the wrist to move positions and the end-effector to do work

### 1.2.1 Wrist

The wrist of a robotic manipulator refers to the joint in kinematics model between the arm and hand. The wrist joint almost always revolutes. The wrist that is commonly used is the Spherical Wrist, which means the joint axes intersect at one point, the construction of which can be seen in Figure 2.

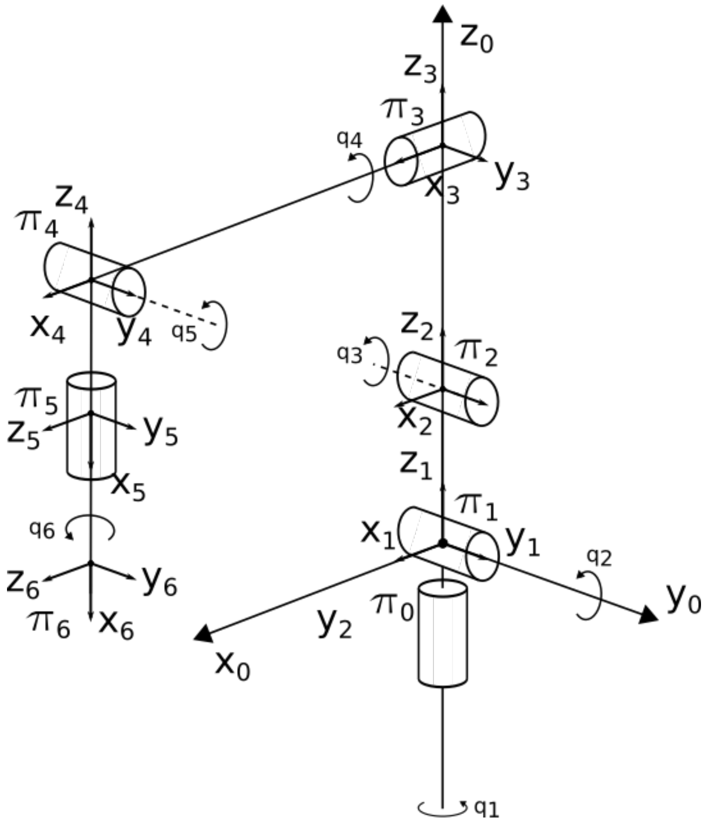


Figure 2: Kinematics model

Spherical wrist is very simple in kinematic analysis because it allows to separate analysis of position and orientation. The robotic arm and wrist are used to determine the position of the end-effector and the tool it carries.

### 1.2.2 End of Effector

End-effector is part of the robot manipulator that does work (holding, drilling, welding). The simplest type of end-effector is the gripper. As shown in Figure 3.

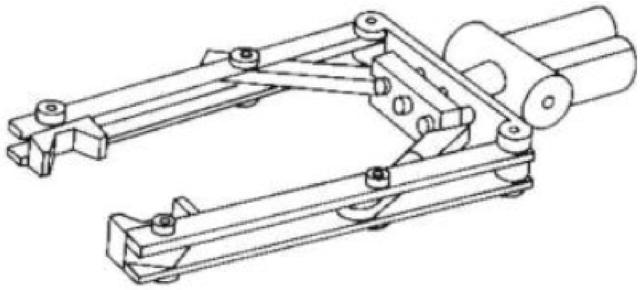

Figure 3: Gripper

# 2 Kinematics

Kinematics is the description of motion without regard to the forces that cause it. The kinematics solutions of any robot manipulator are divided into two solutions, the first one is the solution of Forward kinematics, and the second one is the inverse kinematics solution. Forward kinematics will determine where the robot's manipulator hand will be if all joints are known. Where the inverse kinematics will calculate what each joint variable must be if the desired position and orientation of end-effector is determined. Hence, Forward kinematics is defined as transformation from joint space to Cartesian space whereas Inverse kinematics is defined as transformation from Cartesian space to joint space [3]. The relationship between forward kinematics and inverse kinematics is shown in Figure 4.

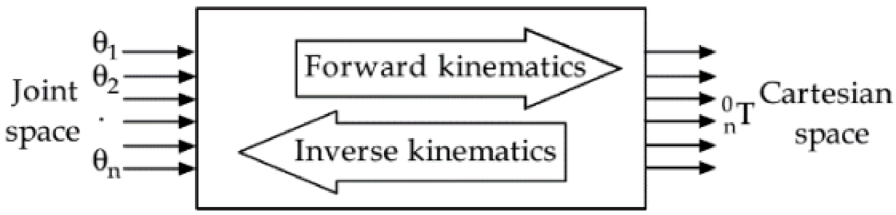


Figure 4: Relationship between forward and inverse kinematics

## 2.1 Forward Kinematics

Forward kinematics/direct kinematics of the robot is used to find out the position and orientation of the end-effector for a known joint angles and link parameters. Manipulator consists of so many parts and the positions can be calculated with respect to the different reference frames. An analysis of the links at different position is methodically calculated.

### 2.1.1 DH Parameters

DH parameter is used to describe the link relationship of the robot where the link is assumed to be a rigid body. There are four parameters in determining the link relationship, including those, which describe the link parameters, $\theta_i$ and $d_i$ which describe the relationship between links. The four parameters $a_i, \alpha_i, d_i, \theta_i$ are generally named link length, link twist, link offset, and joint angle.

In performing kinematics analysis, standardization is needed to determine the coordinates of a frame on each link. The standardization used in this final project is the DH (Denavit-Hertenberg) approach. In the DH approach, each homogeneous transformation is represented by the product of the four basic transformations as shown in this equation.

$$T_i^{i-1} = Rot_{x_{i-1},\alpha_{i-1}} Trans_{x_{i-1},\alpha_{i-1}} Trans_{z_i,d_i} Rot_{z_i,\theta_i} \tag{1}$$

$$T_i^{i-1} = \begin{bmatrix} c_{\theta i} & -s_{\theta i} & 0 & \alpha_{i-1} \\ s_{\theta i} c_{\alpha i-1} & c_{\theta i} c_{\alpha i-1} & -s_{\theta i-1} & -d_i s_{\theta i-1} \\ s_{\theta i} s_{\alpha i-1} & c_{\theta i} s_{\alpha i-1} & c_{\theta i-1} & d_i c_{\theta i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Thus, the DH parameters for the kinematic model as depicted in figure 2 will be presented in the table 1.

| i | $a_{i-1}$ | $\alpha_{i-1}$ | $d_i$ | $\theta_i$ |
|---|---|---|---|---|
| 1 | 0 | $3\pi/2$ | 0 | $\theta_1$ |
| 2 | $a_2$ | 0 | 0 | $\theta_2$ |
| 3 | $a_3$ | $3\pi/2$ | 0 | $\theta_3$ |
| 4 | 0 | $\pi/2$ | $d_4$ | $\theta_4$ |
| 5 | 0 | $3\pi/2$ | 0 | $\theta_5$ |
| 6 | 0 | 0 | $d_6$ | $\theta_6$ |

Table 1: DH Parameters

By substituting the DH parameters, individual transformation matrices obtained as follows

$$T_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ -s_1 & -c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & a_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^2 = \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ 0 & 0 & 1 & 0 \\ -s_3 & -c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^3 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ 0 & 0 & -1 & -d_4 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_5^4 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_3 & -c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6^5 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where $c_{\theta i}$ and $s_{\theta i}$ are the short hands for $cos_{\theta i}$ and $sin_{\theta i}$ respectively. The Transformation of end effector frame to be base frame (reference frame) the matrix product of each transformation matrices, Hence.

$$T_i^0 = T_1^0 \ . \ T_2^1 \ . \ T_3^2 \ . \ T_i^{i-1} \tag{2}$$

Composition of matrices

$T_2^0 = T_1^0 \ . \ T_2^1$

$$T_2^0 = \begin{bmatrix} c1c2 - s1s2 & -c1s2 - c2s1 & 0 & c1a1 \\ 0 & 0 & 1 & -d1 \\ -c1s2 - c2s1 & -c1c2 + s1s2 & 0 & -s1a1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$T_3^0 = T_1^0 \ . \ T_2^1 \ . \ T_3^2$

$$T_3^0 = \begin{bmatrix} (c1c2 - s1s2)c3 & -(c1c2 - s1s2)s3 & -c1s2 - c2s1 & (c1c2 - s1s2)a2 + c1a1 \\ -s13 & -c3 & 0 & -d1 \\ (-c1s2 - c2s1)c3 & -(-c1s2 - c2s1)s3 & -c1c2 + s1s2 & (-c1s2 - c2s1)a2 - s1a1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$T_4^0 = T_1^0 \ . \ T_2^1 \ . \ T_3^2 \ . \ T_4^3$

$T_4^0(1,1) = [c1c2 - s1s2)c3c4 + (-c1s2 - c2s1)s4]$

$T_4^0(1,2) = [-(c1c2 - s1s2)c3s4 + (-c1s2 - c2s1)c4]$

$T_4^0(1,3) = [(c1c2 - s1s2)s3]$

$T_4^0(1,4) = [(c1c2 - s1s2)s3d4 + (c1c2 - s1s2)a2 + c1a1]$

$T_4^0(2,1) = [-s13c4]$

$T_4^0(2,2) = [s13s4]$

$T_4^0(2,3) = [c3]$

$T_4^0(2,4) = [c3d4 - d1]$

$T_4^0(3,1) = [(-c1s2 - c2s1)c3c4 + (-c1c2 + s1s2)s4]$

$T_4^0(3,2) = [-(-c1s2 - c2s1)c3s4 + (-c1c2 + s1s2)c4]$

$T_4^0(3,3) = [(-c1s2 - c2s1)s3]$

$T_4^0(3,4) = [(-c1s2 - c2s1)s3d4 + (-c1s2 - c2s1)a2 - s1a1]$

$T_4^0(4) = [0 \ 0 \ 0 \ 1]$

$T_5^0 = T_1^0 \ . \ T_2^1 \ . \ T_3^2 \ . \ T_4^3 \ . \ T_5^4$

$T_5^0(1,1) = [((c1c2 - s1s2)c3c4 + (-c1s2 - c2s1)s4)c4 + (c1c2 - s1s2)s3s4]$

$T_5^0(1,2) = [-((c1c2 - s1s2)c3c4 + (-c1s2 - c2s1)s4)s4 + (c1c2 - s1s2)s3c4]$

$T_5^0(1,3) = [(c1c2 - s1s2)c3s4 - (-c1s2 - c2s1)c4]$

$T_5^0(1,4) = [-(-(c1c2 - s1s2)c3s4 + (-c1s2 - c2s1)c4)d4 + (c1c2 - s1s2)s3d4 + (c1c2 - s1s2)a2 + c1a1]$

$T_5^0(2,1) = [-s13c4^2 + c3s4]$

$T_5^0(2,2) = [s13c4s4 + c3c4]$

$T_5^0(2,3) = [-s13s4]$

$T_5^0(2,4) = [-s13s4d4 + c3d4 - d1]$

$T_5^0(3,1) = [((-c1s2 - c2s1)c3c4 + (-c1c2 + s1s2)s4)c4 + (-c1s2 - c2s1)s3s4]$

$T_5^0(3,2) = [-((-c1s2 - c2s1)c3c4 + (-c1c2 + s1s2)s4)s4 + (-c1s2 - c2s1)s3c4]$

$T_5^0(3,3) = [(-c1s2 - c2s1)c3s4 - (-c1c2 + s1s2)]$

$T_5^0(3,4) = [-(-(-c1s2 - c2s1)c3s4 + (-c1c2 + s1s2)c4)d4 + (-c1s2 - c2s1)s3d4 + (-c1s2 - c2s1)a2 - s1a1]$

$T_5^0(4) = [0 \ 0 \ 0 \ 1]$

$T_6^0 = T_1^0 \ . \ T_2^1 \ . \ T_3^2 \ . \ T_4^3 \ . \ T_5^4 \ . \ T_6^5$

$T_6^0(1,1) = [(((c1c2 - s1s2)c3c4 + (-c1s2 - c2s1)s4)c4 + (c1c2 - s1s2)s3s4)c5 - ((c1c2 - s1s2)c3s4 - (-c1s2 - c2s1)c4)s5]$

$T_6^0(1,2) = [-(((c1c2 - s1s2)c3c4 + (-c1s2 - c2s1)s4)c4 + (c1c2 - s1s2)s3s4)s5 - ((c1c2 - s1s2)c3s4 - (-c1s2 - c2s1)c4)c5]$

$T_6^0(1,3) = [-((c1c2 - s1s2)c3c4 + (-c1s2 - c2s1)s4)s4 + (c1c2 - s1s2)s3c4]$

$T_6^0(1,4) = [-(-(c1c2 - s1s2)c3s4 + (-c1s2 - c2s1)c4)d4 + (c1c2 - s1s2)s3d4 + (c1c2 - s1s2)a2 + c1a1]$

$T_6^0(2,1) = [(-c4^2s13 + c3s4)c5 + s13s4s5]$

$T_6^0(2,2) = [-(-c4^2s13 + c3s4)s5 + s13s4c5]$

$T_6^0(2,3) = [c4s13s4 + c3c4]$

$T_6^0(2,4) = [-d4s13s4 + c3d4 - d1]$

$T_6^0(3,1) = [(((-c1s2 - c2s1)c3c4 + (-c1c2 + s1s2)s4)c4 + (-c1s2 - c2s1)s3s4)c5 - ((-c1s2 - c2s1)c3s4 - (-c1c2 + s1s2)c4)s5]$

$T_6^0(3,2) = [-(((-c1s2 - c2s1)c3c4 + (-c1c2 + s1s2)s4)c4 + (-c1s2 - c2s1)s3s4)s5 - ((-c1s2 - c2s1)c3s4 - (-c1c2 + s1s2)c4)c5]$

$T_6^0(3,3) = [-((-c1s2 - c2s1)c3c4 + (-c1c2 + s1s2)s4)s4 + (-c1s2 - c2s1)s3c4]$

$T_6^0(3,4) = [-(-(-c1s2 - c2s1)c3s4 + (-c1c2 + s1s2)c4)d4 + (-c1s2 - c2s1)s3d4 + (-c1s2 - c2s1)a2 - s1a1]$

$T_6^0(4) = [0\ 0\ 0\ 1]$

## 2.2 Inverse Kinematics

Inverse kinematics is used to find joint variables obtained from the position and orientation of the end-effector. In general, calculations for inverse kinematics are more difficult than for forward kinematics. Because the problem in inverse kinematics is to find the angle of each joint based on the desired final position of a robot, inverse kinematics is said to be a solution that has a unique final result. It is said to be unique because of the results of the angle for each joint there are many kinds of solutions depending on the number of joints used. Therefore, there are several methods to find inverse kinematics solutions, geometrical approach, iterative approach, pseudo inverse and matrix algebraic approach

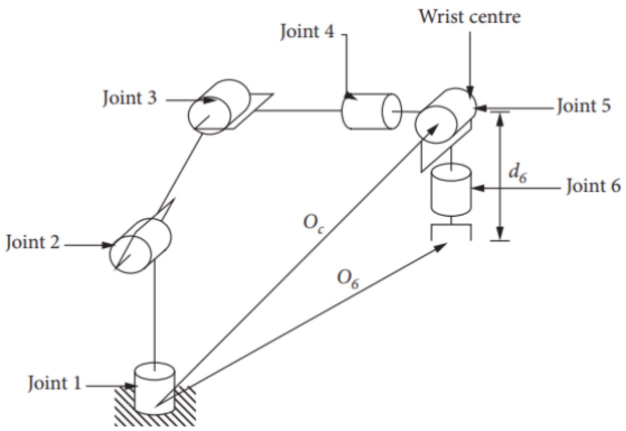


Figure 5: Kinematics decoupling

We can observe in the diagram that the motion of the angles $\theta_4$, $\theta_5$, $\theta_6$ does not affect the position of the point denoted as $O_c$ so we can analyze the angles $\theta_1$, $\theta_2$, $\theta_3$ in order to find the position of the manipulator.

So we can temporarily remove the links after the joint 5 and focused in this $O_c$ point, to obtain the coordinates of $O_c$ we can observe that is the $0_6$ point – $d_6$ multiplying for the rotations, this means:

$$P^0_{xc} = P^0_{x6} \ - \ d_6 \ . \ r_{13} \tag{3}$$

$$P^0_{yc} = P^0_{y6} \ - \ d_6 \ . \ r_{23} \tag{4}$$

$$P^0_{zc} = P^0_{z6} \ - \ d_6 \ . \ r_{33} \tag{5}$$

Now to solve $\theta_1$ is enough to take a look at the configuration from the top.

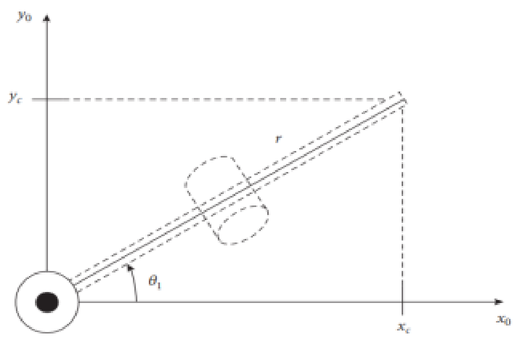


Figure 6: Projection of wrist center onto $\mathrm{x}_0, y_0 plane(Spong, vidyasagar, 2006)$

From here we can see that the angle $\theta_1$ is dependent of the coordinates of $x_c$ and $y_c$, to this angle we can use the arctg ($x_c$, $y_c$). This means:

$$\theta_{11} = arctg(\frac{P^0_{yc}}{P^0_{xc}}) \tag{6}$$

If the wrist is rotated, then it will result in the following equation:

$$\theta_{12} = arctg(\frac{P^0_{yc}}{P^0_{xc}}) + \pi \tag{7}$$

Now we can take a projection from the axis Z and X and we can observe how the rest of the links behave.

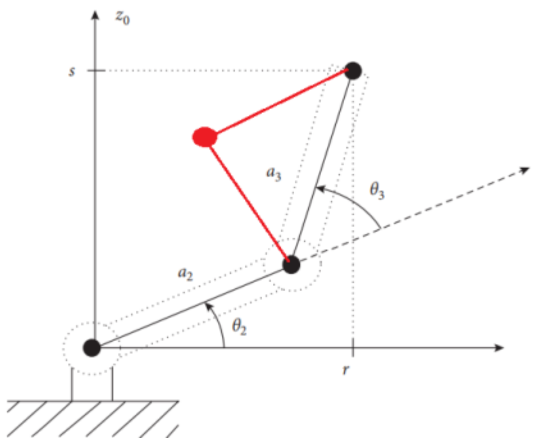


Figure 7: Link of DH parameter

We can observe that because the angle $\theta_4$ not rotate in the same axes as $\theta_3$ This distance will be always the same and between $\theta_3$ and $\theta_5$ we have this constant measure that we can call $l_3$ and it is equal to $l_3$= $\sqrt{a_4^2 \ . \ d_4^2}$ this line has an angle that following the cosine law we can define as:

$$cos\theta_3 = \frac{(xc^2 + yc^2) + (zc - d1)^2 - a_3^2 - l_3^2}{2 \ . \ a_3^l \ . \ 3} \tag{8}$$

And applying trigonometry rules we can solve $\theta_3$ :

$$c\theta_3 = \frac{\pi}{2} - arctg(\frac{\pm\sqrt{1 - cos\theta_3^2}}{cos\theta_3}) \tag{9}$$

Once again, we will obtain 2 values multiplying the 2 values obtained for $\theta_1$ We have already 4 possible solutions.

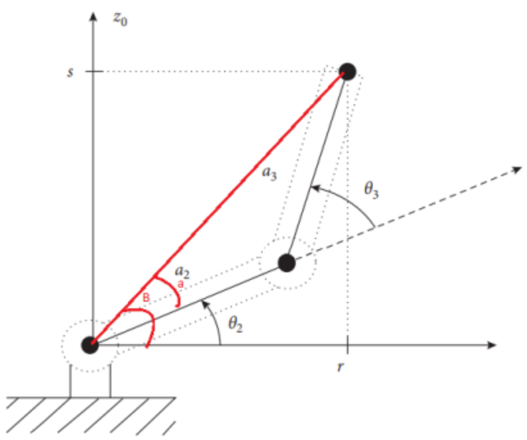


Figure 8: Projection onto $l_2$ and $l_3$ plane to find $\theta_2$. (Spong , vidyasagar, 2006)

Now we can see that $\theta_2$ is equal to $\beta$ (the angle between the X axis and the hypotenuse created between $a_3$ and $l_3$) minus $\alpha$ (the angle between the hypotenuse and $a_3$).

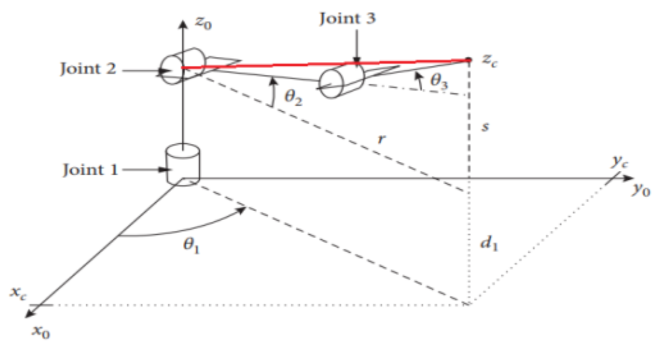


Figure 9: Projection onto $l_2$ and $l_3$ plane to find $\beta$. (Spong , vidyasagar, 2006)

From figure 9 we can see that $\beta$ (the angle between the X axis and the hypotenuse created between $a_3$ and $l_3$) is the angle between the points given by r and $z_c - d_1$ and from here we can assume that:

$$\beta = arctg(\frac{\pm\sqrt{x_c^2 + y_c^2}}{z_c - d_1}) \tag{10}$$

To fin $\alpha$ we can observe the next image.

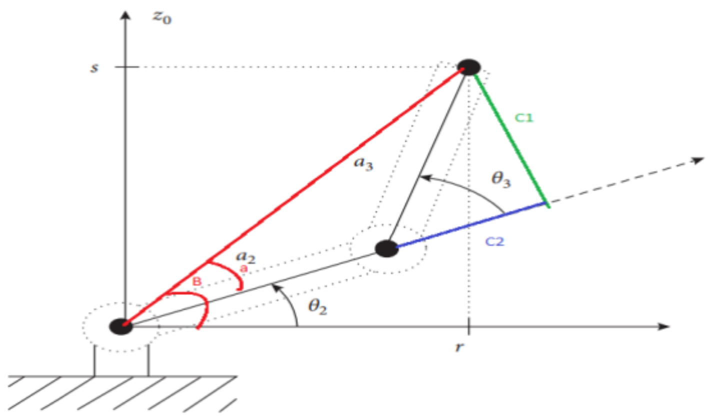


Figure 10: Projection onto $l_2$ and $l_3$ plane to find $\alpha$. (Spong , vidyasagar, 2006)

From this image we know that $c_1 = \sin\theta_3 \ . \ l_3$ and $c_1 = \cos\theta_3 \ . \ l_3$ now $\alpha$ is given by:

$$\alpha = arctg(\frac{sin\theta_3 \ . \ l_3}{cos\theta_3 \ . \ l_3 + a_3}) \tag{11}$$

then $\theta_2$ is given by:

$$\theta_2 = \beta - \alpha \tag{12}$$

#### 2.2.1 Inverse Kinematics Solution

From the direct Kinematics, we now have an expression for the matrix $T_6^0$ in terms of the joint angles.

However, in order to direct the manipulator's end effector to a given point, it is necessary that we have a method to determine joint angles that would give us a desired position and orientation. It should also be noted that often multiple joint co-ordinates result in the same end effector position and orientation.

For the given manipulator, we have: $T_6^0 = R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Where the values $r_{ij}$, $i, j \in 1, 2, 3$ and $p_x$,$p_y$,$p_z$ are given.

Based on the expressions for the direct kinematics, we can derive the following equations for the joint angles.

$cos(\theta_3) = \frac{p_y - d_6 r_{23}}{d_4} \implies \theta_3 = \arccos(\frac{p_y - d_6 r_{23}}{d_4})$

If $\theta_3 \neq 0, pi$ then $\theta_4 = \arcsin(\frac{r_{23}}{sin(\theta_3)})$

However if $|p_y - d_6 r_{23}| = |d_4|$ then we have $\theta_3 = 0$,$\pi$ and $\theta_4 = 0$,$\pi$

Let $A = cos\theta_3 sin\theta_4$, $B = -\cos\theta_4$, $\gamma = \theta_1 + \theta_2$ and $\phi = \arctan \frac{-cos\theta_4}{sin\theta_4 cos\theta_3}$. Then

$r_{33} = A \sin\gamma - B\cos\gamma$

$\implies \gamma = \arcsin(\frac{r_{33}}{\sqrt{A^2+B^2}}) - \phi)$

If on the other hand, $|p_y - d_6 r_{23}| = |d_4|$ then we have $\gamma = \arccos(\pm r_{33})$

We can now use the equation

$p_z = d_6 \sin(\theta_4)\cos(\theta_3)\sin(\gamma) - d_6\cos(\theta_4)\cos(\gamma) - a_2\sin(\gamma) - a_1\sin(\theta_1)$

to calculate that

$\sin(\theta_1) = (-p_z + d_6\sin(\theta_4)\cos(\theta_3)\sin(\gamma) - d_6\cos(\theta_4)\cos(\gamma) - a_2\sin(\gamma))$

And thus, we now have values for $\theta_1, \theta_2, \theta_3, \theta_4$.

Therefore, it is now possible to express the matrix $T_4^0$ in terms of the given values of $r_{ij}$ and $p_x$,$p_y$,$p_z$.

Let us consider the matrix $L$ given by

$$L = (T_4^0)^{-1} R \tag{13}$$

And, in particular $L = \begin{bmatrix} l_{11} & l_{12} & l_{13} & l_{14} \\ l_{21} & l_{22} & l_{23} & l_{24} \\ l_{31} & l_{32} & l_{33} & l_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$

With the $l_{ij}$ given in terms of $\theta_k$ for $k \in 1, 2, 3, 4$ which have been determined, and the matrix $R$.

We also know that $L = T_5^4 T_6^5$. Let $\psi = \theta_5 + \theta_6$

$$L = \begin{bmatrix} c(\psi) - s(\psi) & -c_5 s_6 - s_5 c_6 & 0 & -s_5 d_6 \\ 0 & 0 & 1 & 0 \\ -c_5 s_6 - s_5 c_6 & -c(\psi) + s(\psi) & 0 & -c_5 d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Therefore, we have $\cos(\psi) = l_{11}$ and $\sin(\psi) = -l_{12}$, which allows us to determine $\psi$.

We further know that $r_{31} = \cos(\theta_3)\sin(\theta_5 - \theta_6) - \cos\theta_4 \sin\theta_3 \cos(\psi)$

$(\theta_5 - \theta_6) = \arcsin(\frac{r_{31} + \cos\theta_4 \sin\theta_3 \cos\psi}{\cos(\theta_3)})$

Since we know $\theta_5 + \theta_6$ as well as $\theta_5 - \theta_6$, we therefore have all the joint co-ordinates, as required.

# 3 Visualisation of manipulator

The visualization of the manipulator using a function for LSPB trajectory and rendered using The Robotics Toolbox for MATLAB. For the purpose of proof this values a program was created in MATLAB, the program is taking all of the DH parameters and calculating all of the transformations, then given the angles 1, 2, 3, 4, 5, 6 by link.

```
>> L(1) = Link('d',0,'a',0,'alpha',3*pi/2);
L(2) = Link('d',0,'a',0.0715,'alpha',0);
L(3) = Link('d',0,'a',0.15018,'alpha',3*pi/2);
L(4) = Link('d',0.1312,'a',0,'alpha',pi/2);
L(5) = Link('d',0,'a',0,'alpha',3*pi/2);
L(6) = Link('d',0.076,'a',0,'alpha',0);
Robot = SerialLink(L)
Robot.name = 'robot';
```

Figure 11: Link of DH parameter

Before the program is run several parameters are entered to get the final position of the robot by using the formula Linear Splines with Parabolic Blends (LSPB) trajectory

```
>> q=invkin(Robot,0.1,0,0.1)

q =

    0.0000   -0.8332   -0.9896    0.0000   -1.4154    0.0000

>> q1=invkin(Robot,0.2,0,0.1)

q1 =

    0.0000    0.4641    0.2077    0.0000   -1.2749    0.0000

>> lspbt(q,q1,0.02*ones(1,6),0.05*ones(1,6),600)
```

Figure 12: Formula of LPSB

The command in figure 7 is executed to display the visualization

```
>> visual(Robot,ans)
```

Figure 13: Command for visualization

Figure 8 shows the results of the visualization

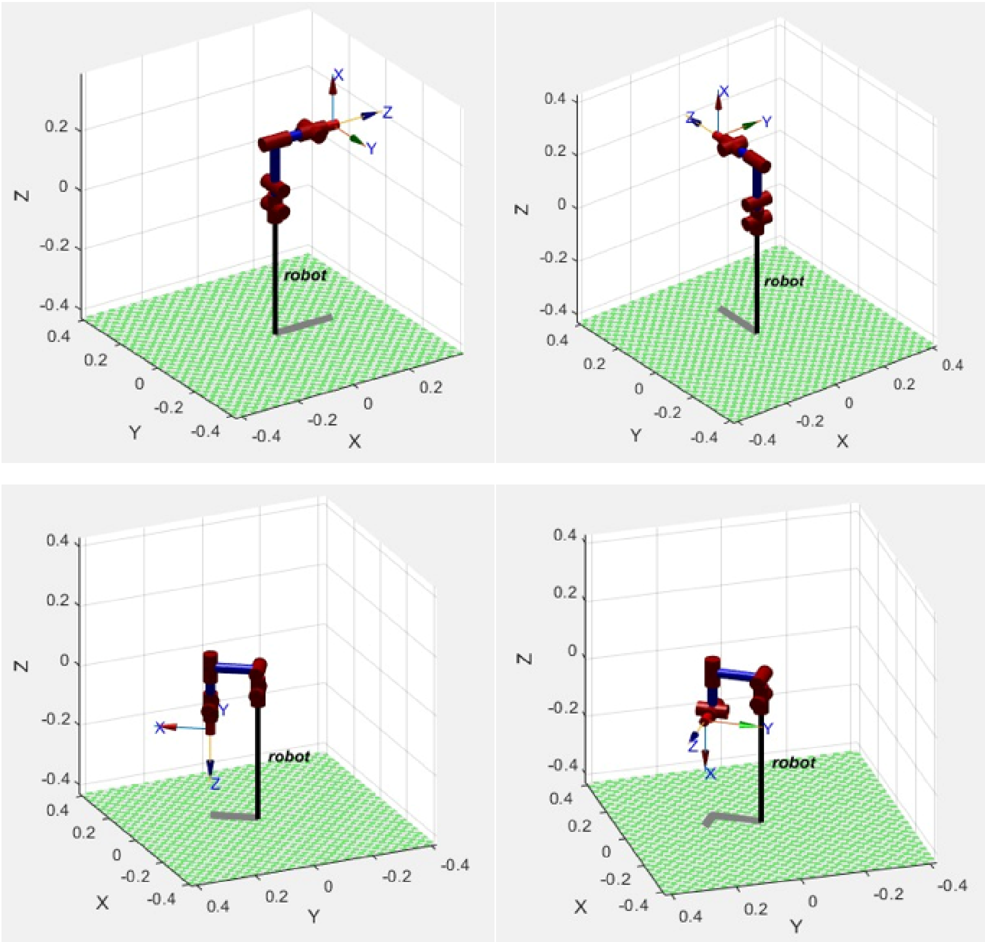


Figure 14: Command for visualization

# 4 Trajectory generation

The goal of the project is to solve the pick and place problem. Given an object (with fixed dimensions) on a surface, the aim is to pick it up from a fixed location and transfer it to a target location, where it is to be placed.

For the sake of simplicity, it is assumed that the orientation of the gripper while picking and placing the object is fixed. To this end, we choose the rotation matrix determining the orientation of the gripper to be the identity matrix. Hence, if we assign the values $p_x$,$p_y$ and $p_z$ to the location of the gripper, we have the transformation $T$ to be:

$$T = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that while the co-ordinates $p_x, p_y$ are available from the location on the board of the object $p_z$ is obtained using the height of the object.

## 4.1 Generation of trajectory

This section addresses the solution for the trajectory generation problem.

The trajectory generation for this problem is solved entirely in the joint space, that is the initial and final points are converted to the relevant co-ordinates in the joint space, and a trajectory is generated therein. The following sections address each of these in turn.

### 4.1.1 Conversion to Joint Space

Using the inverse kinematics, and the locations of the pick and place points, we are able to determine the joint co-ordinates corresponding to these points. That is, we can determine:

$$q_i = (\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i, \theta_5^i, \theta_6^i) \tag{14}$$

$$q_f = (\theta_1^f, \theta_2,^f \theta_3^f, \theta_4^f, \theta_5^f, \theta_6^f) \tag{15}$$

Where $q_i$ and $q_f$ correspond to the joint co-ordinates for the pick and the place points respectively. The trajectory between these two points can then be generated between these two points in the six-dimensional joint space.

### 4.1.2 Trajectory in the Joint Space

We consider the general problem of generating a trajectory in the joint space between two points $x$ and $y$ (from $x$ to $y$) given by

$$x = (x_1, x_2, x_3, x_4, x_5, x_6) \tag{16}$$

$$y = (y_1, y_2, y_3, y_4, y_5, y_6) \tag{17}$$

In the manipulator, we are primarily concerned with smooth transitions from one joint angle to the next for each joint. Consequently, a Linear Splines with Parabolic Blends (LSPB) trajectory is chosen to accomplish this, where the maximum speed and acceleration are set as follows ($v$ indicates the set of maximum velocities, and $a$ indicates the maximum acceleration)

$$v = (v_1, v_2, v_3, v_4, v_5, v_6) \tag{18}$$

$$a = (a_1, a_2, a_3, a_4, a_5, a_6) \tag{19}$$

The methodology of the trajectory generation algorithm is briefly outlined in what follows.

Preliminary steps:

- Firstly, for all $i \in 1, 2, 3, 4, 5, 6$ we set $sgn(v_i) = sgn(a_i) = sgn(y_i - x_i)$ to ensure that movement in the joints is occurring in the right direction
- Secondly, we estimate the total time for the trajectory and blending times (for each joint $i$), given by the equations
  $t_f^i = \frac{y_i - x_i}{v_i} + \frac{v_i}{a_i}$ and
  $t_b^i = \frac{v_i}{a_i}$
- Note that in order for the LSPB trajectory to be feasible, it is necessary that $t_f^i > 2t_b^i \ \forall i \in 1, 2 \ldots 6$, and thus we have the constraint
  $\frac{v_i}{a_i} < \frac{y_i - x_i}{v_i}$
  which aids us in choosing the speeds and accelerations involved.

After these preliminary steps, it is possible to generate the trajectory using the values obtained. However, during the course of experimentation with the manipulator it was observed that it is often necessary to ensure that $t_f^i = t_f^j \ \forall i, j \in 1, 2 \ldots 6$ in order to achieve smooth motion. Thus, we can use the largest $t_f^i$ achieved in the initial estimates, and duly adjust the maximum speeds to satisfy this additional constraint. The equation is (interpreted as an assignment of values):
$v_i = \frac{t_f a_i + \sqrt{(t_f a_i)^2 - 4a_i(y_i - x_i)}}{2}$

The points in the trajectory can now be generated. It is assumed that we wish to generate $n$ points in the trajectory given by $(x^0 = x, x^1, \ldots, x^{n-1}, x^n = y)$.

The points are determined as follows:

- For $k \leq \frac{nt_b^i}{t_f^i}$ we have $x_i^k = x_i + (a_i/2)((k(t_f^i)/n)^2)$
- For $\frac{nt_b^i}{t_f^i} \leq k \leq \frac{n(t_b^i - t_f)^i}{t_f^i}$ we have $x_i^k = x_i + (v_i t_b^i/2) + (v_i((kt_f^i/n) - (t_b^i)))$
- For $\frac{n(t_b^i - t_f)^i}{t_f^i} \leq k \leq n$ we have $x_i^k = y_i - (a_i/2)(\frac{kt_f^i}{n} - t_f^i)^2$

Thus, given any two points in the joint space, we have a method to obtain a trajectory in between them.

#### 4.1.3 Application to the Problem

For the purposes of this task, the trajectory generation method outlined above is applied using the following methodology:

- The points for the pick ($p^i = (p^i_x, p^i_y, p^i_z)$) and place ($p^f = (p^f_x, p^f_y, p^f_z)$) are converted to the joint co-ordinates $q_i$ and $q_f$ respectively
- In addition to these, we require two additional points $w_1 = (p^i_x, p^i_y, p^i_z + \delta)$ and $w_2 = (p^f_x, p^f_y, p^f_z + \delta)$. Since it was observed during the course of the project that generating a trajectory directly in between the two points often resulted in the manipulator dragging the object along the board, these points serve to ensure the object gains some clearance $\delta$ from the surface of the pick and place points
- The motion of the manipulator therefore is performed through the LSPB trajectory in each of the following segments
  1. From the initial position of the manipulator to $w_1$
  2. From $w_1$ to the point $p^i$
  3. From $p^i$ to $w_2$
  4. Finally, from $w_2$ to the point $p^f$

Note that, since the nature of the LSPB trajectory ensures that the manipulator is unlikely to attempt to pass through the board in any of the segments of motion outlined above, the reliance on accurate inverse kinematics is only in the retrieval of the joint co-ordinates of the manipulator for the pick and place points.

This solution is therefore amenable to allowing the user flexibility in input- the user can choose to input joint co-ordinates instead of initial and final positions as well.

## 5 Software

The major components of the software and various functionalities are elucidated in this section. ALl the code has been written using MATLAB.

### 5.1 Kinematics

While the direct and inverse kinematics have been analytically derived for the given manipulator, for the implementation the *fkine()* and *ikine()* functions from the Robotics Toolbox for MATLAB have been used for the direct and inverse kinematics respectively.This choice is motivated by two major factors:

- The Robotics Toolbox functions allow for convenient visualization of the manipulator in various poses, which allows for preliminary testing for concept trajectories
- The results of these functions are generally reliable on the basis of elementary tests for robustness. The results of the *fkine()* and *ikine()* functions for various joint angles are consistent, and reflect the behaviour of the manipulator accurately

### 5.2 Visualization

The visualization of the manipulator is accomplished by a software that leverages:

1. A function for LSPB trajectory generation based on the outline in Section 4.1
2. The Robotics Toolbox for MATLAB

The software shows the motion of the manipulator through the points generated by the trajectory.

### 5.3 Final Software

The major result of this project is a software that receives as input the pick and place points, as well as additional inputs (The clearance $\delta$ mentioned in Section 4.1.3 for example). The flow of processes in this software and the major features are outlined as follows:

- The software utilizes the bulk read/write function in order to communicate with the Dynamixel system used to actuate the robot
- Initially, the communication protocol is set and the torque to the various motors is enabled. A maximum is set on the torque used for the motors, in order to avoid excessively taxing the manipulator
- The current position of the manipulator is read and saved.
- Trajectories are generated between the various points according to the scheme outlined in Section 4.1.3
- These trajectories are then successively transmitted to the manipulator as a sequence of goal positions, where all the joints are given their goals simultaneously leveraging the bulk-write functionality

## References


[1] John J. Craig *Introduction to Robotics, Mechanics and Control, Third Edition.* Pearson Education, .

[2] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar *Robot Modeling and Control* Jhon Wiley, Sons, Inc, 2006.