YILDIZ TECHNICAL UNIVERSITY

Faculty of Mechanical Engineering
Department of Mechatronics Engineering

Artificial Intelligence & Image Processing Based Feeding Assistance Robot

Project Team

Oguzhan YARDIMCI / 1506A023

Mustafa UGUR / 1506A007

Cemil YILMAZ / 15067005

Haluk BASI / 1606A035

Advisor
Dr. Erhan Akdogan
Yildiz Technical University
Istanbul, Turkey

03.01.2020

SYMBOL LIST

θ_i	Joint angle [°*degree*]
d_i	Link offset [*mm*]
a_i	Link length [*mm*]
α_i	Twist angle [°*degree*]
η_{3x3}	Perspective vector [*0*]
δ	Scaler Vector [*1*]
$\dot{\theta}$	Velocity [rad/s]
$\ddot{\theta}$	Acceleration [rad/s^2]
τ	Torque [*kg.cm & N.m*]
F	Force
r	Distance
m	Mass [*gr*]
V	Voltage
μs	Microsecond
mA	Miliampere
mAh	Miliampere-hour
x,y,z	Coordinate Axes
Π	Multiplication Symbol
Σ	Summation Symbol [*Sigma*]
c_i	cosine *[°degree]*
s_i	sine *[°degree*]
c_{ij}	Addition of cosines *[°degree*]
s_{ij}	Addition of sines *[°degree*]
P_x	Translational Transform
T_i^{i-1}	Transformation Matrix
D_i^{i-1}	Rotation Matrix
ω	Angular Velocity
v	Linear Velocity
$K_E{}^{(R)}$	Rotational Kinetic Energy
$K_{E_1}^{(T)}$	Translational Kinetic Energy
P_E	Potential Energy
$\mathcal{L}$	Lagrangian Function
J	Moment of Inertia [$kg \cdot m^2$]
b	Motor viscous friction constant [$N \cdot m \cdot s$]
K_b	Electromotive Force Constant [$V/rad/s$]
K_t	Motor Torque Constant [$N \cdot m/Amp$]
R	Electrical Resistane [Ω]
L	Electrical Inductance [H]
K_p	Proportional Controller Gain
K_i	Integral Controller Gain
K_d	Derivative Controller Gain

ABBREVIATION LIST

EOG	Electrooculography
EMG	Electromyography
EEG	Electroencephalography
ALS	Amyotrophic Lateral Sclerosis
FPS	Frames per second
RGB	Red Green Blue
PLA	Polylactic Acid
ABS	Acrylonitrile Butadiene
DH	Denavit Hartenberg
FDM	Fused Deposition Modeling
DOF	Degrees of Freedom
SDK	Software Development Kit
TTL	Transistor to Transistor Logic
URDF	Unified Robot Description Format
PID	Proportional-Integrator-Derivative
Li-Po	Lithium Polymer
CAD	Computer Aided Design
USB	Universal Serial Bus
RAM	Random Access Memory
ISO	International Organization for Standardization
CE	Community Europe
IEC	International Electrotechnical Commission
R&D	Research & Development
INC	Incorporated
et. al.	et alia
mm	Millimeter
cm	Centimeter
rad	Radian
kg	Kilogram
s	Second

FIGURE LIST

TABLE LIST

ACKNOWLEDGMENT

*We offer our sincere thanks to **3LS R&D CONSULTING Industry and Trade INC.** for their help with the manufacturing in this project and our advisor is **Assoc. Prof. Dr. Erhan AKDOGAN.***

Abstract

This study explains the design of four degrees of freedom feeding assistant robot, which might be used by patients who have spinal cord injuries, amyotrophic lateral sclerosis (ALS) or Parkinson's disease or upper-limp amputees.

The design of the robot arm was investigated deeply in the following parts of the report. First, the mechanical design of the robot arm was given in detail. After that, strength analyses of each art of the design were investigated. Then, mathematical models of the robot arm were studied, in that part forward and inverse kinematics and dynamical equations of the robot arm were calculated. In addition to these, the servo motor and the camera which was chosen for the project were explored. Next, the simulation results of the robot arm for the trajectory following were given in detail. Finally, the image processing part of the design for the mouth tracking feature of the system was explained deeply.

This project has been developed in the scope of the Mechatronic System Design course, which is served in the Mechatronics Engineering Department at Yildiz Technical University, Istanbul, Turkey.

CONTENTS

1. Introduction

Robotic arms are one of the keystones of industrial production institutions. The robotic arm which is programmable characterize a part of the complex robot or entire of the mechanical structures. There are a lot of application areas of robotic arms like manufacturing, transportation, service or medical.

The average living time of people is increasing day by day in the world. Because of this the need for caretakers is also increasing. The use of medical robots and assistance robots to decrease these needs is becoming more and more common. In the past few years, many robots have been designed and sold in the market. Some of these designs are taking attention, especially robots that help people in their daily activities [1].

The most important group that needs care is people with physical disabilities. Even though we couldn't find an exact number, we can say that there are almost 9 million disabled people in Turkey [2]. Many of them are suffering while they are maintaining their daily activities and need help from others. Also, we couldn't reach the exact number of these kinds of people in Turkey. However, according to previous studies, there are 12.3 million people in the United States and 600,000 people in Japan who need assistance while doing their daily activities [3].

Eating is one of the essential activities that disabled people need to help. It is nearly impossible to do by a person who has any of the following diseases: spinal-cord injuries, strokes, upper-limb amputees or Parkinson. Because the need for a caretaker injures patient's feelings and there isn't enough caretaker in the market, people have been designing a few different products and selling them during the past few years.

1.1. The Scope and Motivation of the Project

In this study, a feeding assistance robot that has both eye-tracking and mouth-tracking methods will be performed. Users can select three different types of food without need any caretaker. Users can use this robot in hospitals and homes easily.

The feeding assistance robot can carry a spoon full of meals. The reachable area was thought 50 centimeters and created with respect to this restricted workspace. As the most important point, the robot will go to the user's mouth smoothly and consistently.

As in the scope of Mechatronic System Design course, the part of the project includes the design of the robotic arm, analyses of the robotic arm, tests of mouth tracking part of the software, results of simulations which allows to visualize the trajectory following of the robot and the selection of servo motors and the camera with respect to simulation and analyses results.

1.2. The Aim of the Project

Human-Robot interaction is the most important problem in those products which are still in the market. All those products need a teaching phase that needs an external person other than the user. Besides, users must use a button or a chin stick to transmit the required commands to the system. To deal with this problem, several methods have been tried to facilitate human-robot interaction. Control using electrooculography (EOG) signals, electromyography (EMG) signals and electroencephalography (EEG) signals can be given as examples of these methods.

The project that is going to be produced will take away the teaching phase. The feeding assistance robot that goes to the established point is going to the required position by using mouth tracking. Another problem of human-robot interaction will be removed by using the eye-tracking method to select the meals. To sum, the main purpose of the system to give back their independence to the users with a robust robot arm design and easy-to-use user interface.

1.3. Innovative Approaches to the Project

Several feeding assistance robots were produced in the previous time. Both in industrial designs and research projects, several different methods to improve the human-robot interaction were developed and two main methods have been tried so far are the mouth-tracking and eye-tracking methods. Both the eye-tracking method and mouth-tracking algorithms were studied in the previous research projects. However, these two methods have never been used together. Our design will be the first feeding assistant robot design that uses both these two methods at the same time.

1.4. Literature for Commercial Products and Academic Studies

There are several products that were produced like Obi, Meal Buddy, Bestic as commercial products. Also, researchers around the world working in their research projects to develop robust feeding assistant systems and the intention in this research are is increasing constantly with the increase of the developments in the computer science field.

1.4.1. Literature for Commercial Products

The first known product in the feeding assistant robot market is Obi. This six degrees of freedom robot still has the biggest percentage in the market. The first Obi prototype was developed in 2006 in the United States. Obi allows users to choose between four different meals and the user can switch the meals using a touch button. It was certified like the types of ISO, CE, IEC. Obi uses a Li-Po battery as a power source (48Wh / 3.2 Ah) and eating time per full charge is 4 hours. It has portable and rechargeable properties. The

weight of the Obi is $3.49\,kg$. Transport dimensions are $30.5 \times 43.2 \times 15.2\,cm$. The maximum and minimum food delivery heights are $38.1\,cm$ and $5\,cm$, respectively [4].

Figure 1: *Obi robotic feeder*

The second product of the feeding robot is Meal Buddy in commercial products. It has 4 DOF and 3 plates. Meal buddy has one-touch and push-button operation so it pauses anytime with a caretaker. It includes a robotic arm, a white base, three bowls, a spoon, a 2-1/2" diameter button switch and a carrying case. Bowls and base are dishwasher safe. A joystick is used to teach mouth position. Besides, joystick helps the eating program to start the feeding cycle. The caregiver can calibrate and control it easily [5].

Figure 2: *Meal Buddy robotic assist feeder*

The third one of the commercial products is Bestic. it has four degrees of freedom and a single plate. Bestic has an adjustable arm with a spoon on the end-effector. The motion of the arm can be controlled by the user through a button that could be pressed by hand, knee, foot, chin or shoulder. The height and distance of the end-effector to the mouth can be adjusted. The user has possibilities to change the user steers with different control

methods. The spoon does not enter the mouth for safety. The weight of Bestic is $2\ kg$ and the sizes are $3.4\ x\ 8.7\ x\ 7.9\ inches$. Eating time with batter is approximately 5 hours [6].

Figure 3: *Bestic feeding assistive device*

Table 1: *Comparison of Commercial Products*

	Robot Name / Features	Obi Robotic Feeder	Meal Buddy Robotic Assist Feeder	Bestic Feeding Assistive Device	Our Feeding Robot
Human-Robot Interface	Eye Tracking				✓
	Push Buttons	✓	✓	✓	✓
	Joystick		✓	✓	
	Head Control			✓	
	Chin Movements				
How to reach to the user?	Teaching Phase	✓	✓		
	Mouth Tracking				✓
Mechanical Design	4 - DOF		✓	✓	✓
	6 - DOF	✓			
Serving	Single Plate			✓	
	3 Plate		✓		✓
	4 Plate	✓			
Usability	Easy to Transport	✓			
	Cost	$5950	$3500	$3900	$650

1.4.2. Literature for Academic Studies

One of the academic works is the project that was developed by Tanaka et.al. Ultrasonic motors were used in this robot design to make it silent and to be less harmful to the electronic devices around the robot. In this study, the robot was designed to be controlled with an eye-tracking interface. The design does not include a multi-jointed arm structure. Therefore, the end-effector just moves in one direction [1].

Figure 4: *Tanaka et. al.*

The second one is four degrees of freedom robot that was produced by Perera et. al. This meal assistance robot that is controlled by using Electroencephalography (EEG) signals by users. And, it could track the position of the mouth and could detect whether the mouth is open or not. Moreover, the end-effector can reach to user's mouth because of the differences in seating positions using the mouth position tracking method [7].

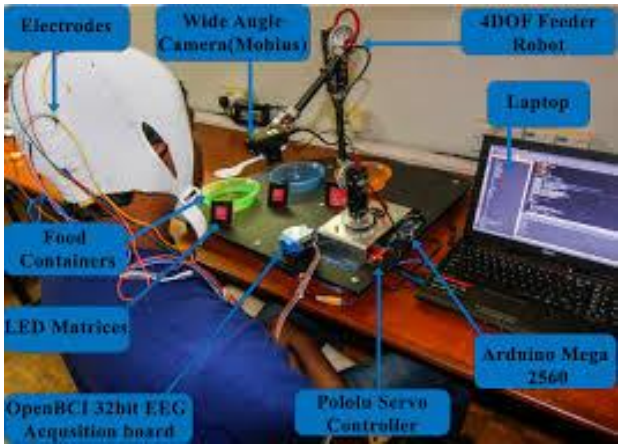

Figure 5: *Perera et. al.*

The third study was made by Bhattacharjee et. al. This design has six degrees of freedom robot arm and the movements of end-effector are done by finding the mouth position of the user. Detection of open and closed mouth is made to be understandable and there is a fork as an end-effector instead of a spoon [3].

Figure 6: *Bhattacharjee et. al.*

Table 2: *Comparison of academic studies*

	Robot Name / Features	Tanaka et. Al	Perera et. Al.	Bhattracharjee et. Al.	Our Feeding Robot
Human-Robot Interface	Eye Tracking	✓			✓
	Push Buttons			✓	✓
	EEG Signals		✓	✓	
How to reach to the user?	Teaching Phase				
	Mouth Tracking		✓	✓	✓
Mechanical Design	One direction	✓			
	4 - DOF		✓		✓
	6 - DOF			✓	
	End-Effector Type	Spoon	Spoon	Fork	Spoon
	Easy to transport				✓
Serving	No Plate	✓			
	Single Plate			✓	
	3 Plate		✓		✓

2. Theoretical Background

In this title of the report, theoretical backgrounds of analyses and image processing were explained. These topics were covered in an order that first strength analyses were covered briefly. Then forward and inverse kinematic analyses were given. Finally, the dynamical analysis topic was described briefly.

2.1. Analyses

This title is separated into three subtitles. First one of them is strength analysis using Solidworks and ANSYS. The second one is kinematic analyses of robot arms. The third one is the dynamical analysis of robot arms.

2.1.1. Strength Analysis

The mechanical components can be analyzed using Solidworks and ANSYS. The most important point of the analysis is to obtain the best mesh analysis because the process can take a long time. In the fine-tuning, the program can make an analysis of the system. In the coarse tuning, the program can make analysis faster than fine-tuning, but the results might not be sufficient. The relationships of overall components have to be defined and there should not be a limitation between parts.

2.1.2. Kinematic Analyses

The problem of kinematics is used to define the motion of the manipulator without considering of forces and torques. Denavit-Hartenberg representation, which is a general transformation between two joints, requires four parameters. These parameters, known as the Denavit-Hartenberg (DH) parameters which became a standard to describe robot kinematics. In serial robots, to calculate the forward kinematics is easier than inverse kinematics. In parallel robots, to calculate inverse kinematics is easier.

2.1.2.1. Forward Kinematic Analysis

The transformation of the coordinate of the end-effector point from the joint space to the world space is known as ***forward kinematic*** transformation. It is also called more clearly, forward kinematic which is to determine the position and orientation of the end effector given the values for the joint variables of the robots [9].

Table 3: *Denavit – Hartenberg Parameters Table*

Axis Number i	Link Length a_i	Twist Angle α_i	Link offset d_i	Joint angle θ_i
1	a_1	α_1	d_1	θ_1^*
2	a_2	α_2	d_2	θ_2^*

Denavit – Hartenberg (DH) method uses the four parameters including a_i,α_i, d_i and θ_i; which are the link length, twist angle, link offset and joint angle, respectively. $Joint\ Angle$ and $Link\ Offset$ are joint parameters. $Link\ Length$ and $Twist\ Angle$ are link parameters.

- θ_i , the joint angle is the angle from x_{i-1} to x_i measured around the z_{i-1}.
- d_i, the link offset is the distance from O_{i-1} to O_i measured along z_{i-1}.
- a_i, the link length is the distance from z_{i-1} to z_i measured along x_i.
- α_i, the twist angle is the angle from z_{i-1} to z_i measured along x_i

The matrix T_i^{i-1} is known as a D-H convention matrix given in this equation. In the matrix T_i^{i-1}, it means the homogenous transformation matrix, the quantities of a_{i-1},α_{i-1}, d_i are constant for a given link while the parameter θ_i for a revolute joint is variable.

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

$$A_i = T_i^{i-1} = P = \begin{bmatrix} & & p_x \\ & R_{3\times3} & p_y \\ & & p_z \\ \eta_{3\times3} & & 1 \end{bmatrix}$$

- $r_{3x3} =$ Rotational Transform: Rotational Transformation between the tool coordinate frame and the base coordinate frame.
- $p_{3x3} =$ Translational Transform: Translational Position of x,y,z axes of the tooltip with respect to the base.
- $\eta_{3x3} =$ Perspective Vector: Always 0.
- $\delta =$ Scaler Vector: Always 1.

2.1.2.2. Inverse Kinematic Analysis

The transformation of coordinates from world space to joint space is known as ***inverse kinematics*** transformation. The inverse kinematics solution uses the position and orientation (P_x, P_y, P_z) of the robot's end-effector to solve the joint angles ($\theta_1, \theta_2, \theta_3, \ldots$).

2.1.3. Dynamical Analysis

The dynamics equations are used to describe the motion of the manipulator with considering of forces and torques. The equations of motion are important to consider in the design of robots, in simulation and animation of robot motion and in the design of control algorithms.

Dynamical analysis can be made using two methods. The first one of them is Euler Lagrange equations, it means Lagrangian Formulation, describes the evolution of a

mechanical system subjected to holonomic constraints. In order to determine the Euler-Lagrange equations in a specific situation, one has to form the Lagrangian of the system, which is a difference between the kinetic energy and potential energy. The second one is Newton-Euler Formulation which is a recursive formulation of the dynamic equations that is often used for numerical calculation [8].

$$Forward\ Dynamics: Given\ \theta, \dot{\theta}, \tau\ and\ find\ \ddot{\theta}$$
$$Inverse\ Dynamics: Given\ \theta, \dot{\theta}, \ddot{\theta}\ and\ find\ \tau$$

The Forward dynamics problem is to calculate the joint accelerations $\ddot{\theta}$, given the current joint positions θ, the joint velocities $\dot{\theta}$, and the forces and torques τ applied at each joint. The forward dynamics is useful for simulation. The inverse dynamics problem is to find the joint forces and torques τ needed to create the acceleration $\ddot{\theta}$ for the given joint positions and velocities that are $\theta, \dot{\theta}$ respectively. The inverse dynamics are useful in the control of robots. Robot dynamics is necessary not just for simulation and control but also for the analysis of robot motion planners and controller.

In this study, Lagrangian Formulation was selected to make dynamical analysis because it is easier than Newton-Euler Formulation. In the Lagrangian Formulation that is based on the differentiation energy terms with respect to the system's variables and time, the Lagrangian Function is founded to calculate the torque values of each joint.

$$\boldsymbol{\mathcal{L}}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right) = \boldsymbol{K}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right) - \boldsymbol{P}(\boldsymbol{\theta})$$

which is $K\left(\theta, \dot{\theta}\right)$ is Kinetic Energy of the System and $P(\theta)$ is Potential Energy of the System.

The vector of joint forces and torques can be demonstrated as follows:

$$\boldsymbol{F_i} = \frac{\boldsymbol{d}}{\boldsymbol{dt}}\left(\frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial \dot{x}_i}}\right) - \frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial x_i}} \qquad , \qquad \boldsymbol{\tau_i} = \frac{\boldsymbol{d}}{\boldsymbol{dt}}\left(\frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial \dot{\theta}_i}}\right) - \frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial \theta_i}}$$

which is the meaning of $\mathcal{L}$ is the Lagrange function of joint variables in the Euler Lagrange equation.

These equations are rather complicated even for simple 2R robots. Some terms are linear in the joint acceleration $\ddot{\theta}$. Some terms do not depend on the joint acceleration but instead depend on a product of joint velocities like $\dot{\theta}_3^2$ or $\dot{\theta}_3^2$. Some terms have no dependence on the joint velocities or accelerations. With this observation, the vector equation of motion can be written as follow:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{c}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right) + \boldsymbol{g}(\boldsymbol{\theta})$$

which is M matrix is called the mass matrix. C vector includes velocity-product term since it composed of terms with $\dot{\theta}_i^2$ in it. Vector g is the gravity vector term since it depends on gravity. It is called a gravity term under the assumption that the potential energy comes

only from gravity but if there are springs at the robot joints, those springs would also contribute to the potential energy and therefore $g(\theta)$.

Overall these equations look like $f = ma + gravity$ which is $ma\ is\ equal\ to\ \ the\ M(\theta)\ddot{\theta} + c(\theta, \dot{\theta})$ and $gravity\ is\ equal\ to\ the\ g(\theta)$ except that the accelerations of the masses depend on the joint accelerations but also products of the joint velocities. These velocity-product terms are appeared because of the joint coordinates are not inertial coordinates.

$$\boldsymbol{\tau = M(\theta)\ddot{\theta} + c(\theta, \dot{\theta}) + g(\theta) + J^T(\theta)F_{tip}}$$

There is one more term added to the right-hand side which is $J^T(\theta)\ is\ Jacobian\ tranpose$ and $F_{tip}\ is\ wrench\ that\ the\ end\ effector$ applies to the environment.

When we focus on the velocity-product terms, it means “vector c”, terms that have the square of a single joint velocity are called **Centripetal terms**$\left(\ \dot{\theta}_i^2\right)$ and terms that have a product of two different joint velocities are **Coriolis terms**$\left(\ \dot{\theta}_i\ \dot{\theta}_j\right)$**.**

2.2. Image Processing

First of all, the frame rate which is one of the most important for image processing is selected. The sensor speed is the speed at which a camera takes the image and delivers it to the system. It should not be considered as focusing and zooming speeds. The sensor speed is expressed by the number of frames and exposure time it produces per second(FPS). The more frames a camera shoots in a second, the more detail it can turn into a signal. [19]

When the sensor wants to convert the movement of an object relatively faster to the number of frames it can produce, any point of the object creates a color value on any point on the sensor before the processing of that mouse falls on the neighbors of that sensor point, causing the image to be blurred. It is important to be able to shoot frames very quickly, but the key point is whether all frames produced in one second have the correct color values.

The movements in our system will not be very fast and 30 FPS will suffice for our system as observed in other projects for such purposes.

The logic of the gray filter is to assign a single value to the pixels by averaging the 3 basic color values which are red, green blue (RGB, [0-255 0-255 0-255]) found in the pixels. Gray filter was applied by using the OpenCV library.

HaarCascade is a system identified by a scientist named Alfred Haar. A method used to locate objects on an image. This method is called Haarlike features. The explanation of the Haarlike is given as follow:

Edge Feature: Indicates an edge property if a certain area on the image consists of a dark area and a certain area consists of light colors.

Line feature: If the image consists of open, closed, light colors respectively, there is a line feature.

Four-Square Feature: If there are diagonally dark and light tones diagonally as crosswise, it refers to the four-square feature.

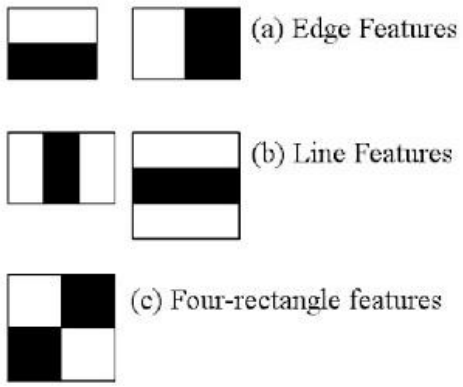

Figure 7: *Haarlike Features*

Using Haarlike features that are shown in Figure 7, many objects can be identified in the image such as lines, edges, faces, eyes, vehicles, etc. HaarCascade is applied to a human face as is shown in Figure 8. The HaarCascade method has been trained many times in advance and knows how the face shape is. When introducing a flower recognition or plate recognition process, the plate is defined many times. For example, in Figure 8, a system that seeks a face-first search for two eyes. If there is an eye, it will look for a nose. The nose is the structure that gives the desired results by checking whether there is an eyebrow.

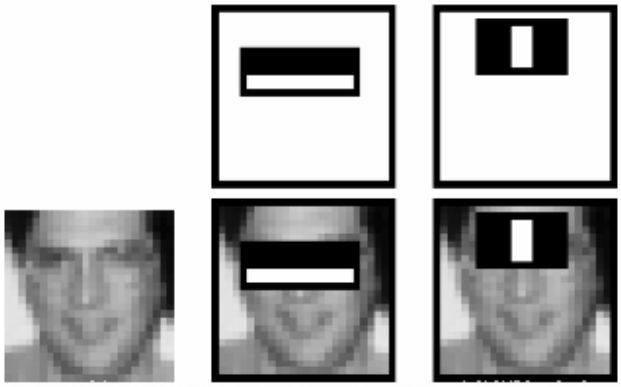

Figure 8: *Haarlike in face*

The item can be defined and searched for objects in the image using HaarCascade, which means HaarCascade can be made independently.

3. Materials and Methods

In this study, a 4 DOF artificial intelligence and image processing based feeding assistance robotic arm for disabled people was developed which is shown schematically in 8. The presented robotic arm is composed of the serial link which is affixed to each other with revolute joints from the base frame to the end-effector. All components were designed, assembled and analyzed by using Solidworks. Forward and inverse kinematic analysis were developed by using Denavit-Hartenberg parameters. Dynamic components were analyzed by using the standard mechanical formula and Solidworks. The feeding assistance robot will be attached to people.

As a material selection, all components were manufactured from PLA (Polylactic Acid) due to the properties of this organic biopolymer and thermoplastic material. It is not harmful to people's health. When we compare with ABS, PLA has a sleekier surface. 3D Printer using FDM technology can print PLA properly. The specifications of PLA are as follow:

- Hard structure. Therefore, PLA is durable and resistant to impacts.
- Slight flexibility but some brittle.
- No problems with plastic deformation during the cooling process.
- Resistance to temperature.
- Difficult to dissolve with acetone.
- Very easy to print according to ABS.

Figure 9: *The assembled model of the feeding assistance robot*

3.1. Description of the Feeding Assistance Robot Workspace

The workspace of a manipulator is the total volume swept out by the end effector as the manipulator executes all possible motions. The workspace is constrained by the geometry of the manipulator as well as mechanical constraints on the joints. For example, a revolute joint may be limited to less than a full 360° of motion. The workspace is often broken down into a ***reachable workspace*** and a ***dexterous workspace***. The reachable workspace is the entire set of points reachable by the manipulator, whereas the dexterous workspace consists of those points that the manipulator can reach with an arbitrary orientation of the end effector. Obviously, the dexterous workspace is a subset of the reachable workspace. In shortly, the robot workspace or reachable spaces consist of all the points in the Cartesian space that the end effector of the robotic arm can access. The workspace and quick access to a certain point in all robotic arms are strongly dependent on linkage properties, joint properties (length, angles, angular velocity, and torque), degree of freedom, angle/translation limitations and robot configurations.

Consider the feeding assistance robot in Figure 10. The right side shows the complete work envelope of the robot from the side view. The left side of Figure 10 shows the whole workspace from the top view. The maximum longitudinal distance reached by the arm is 45.826 cm. All dimensions have a tolerance in the range of 2–4 mm. The maximum height of the accessible point is 50.176 cm. According to the reachable points, the sizes of the feeding assistance robot is sufficient to be used in medical applications. In shortly, the workspace of the feeding assistance robot has a suitable reach to the user's mouth and bowls easily [9].

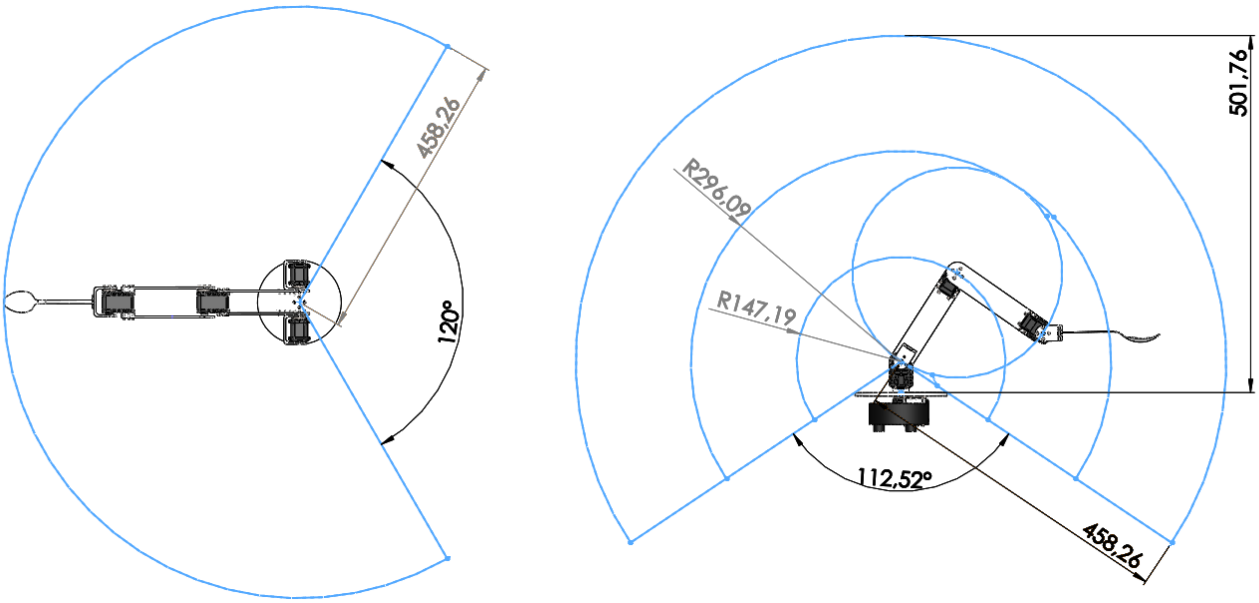

Figure 10: *Workspace of the feeding assistance robot*

3.2. Mechanical Design of the Feeding Assistance Robot

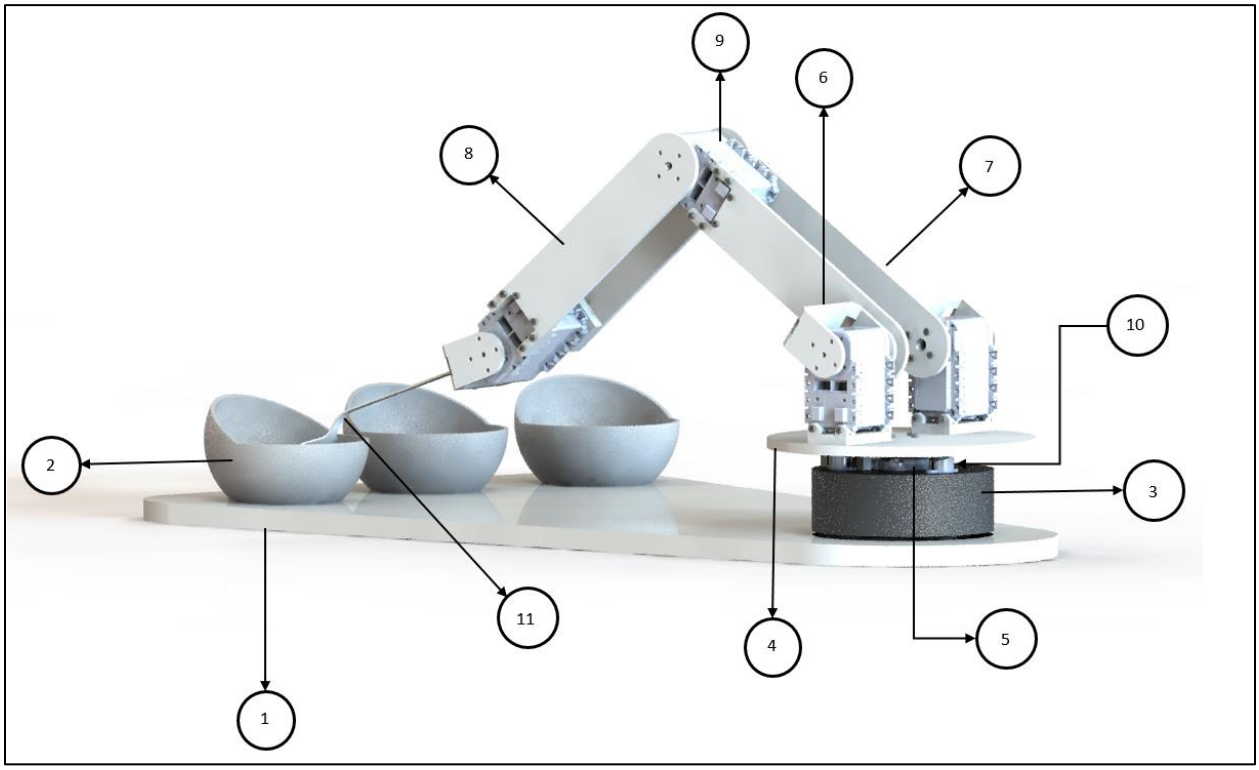

Figure 11: *The Parts of the Feeding Assistance Robot*

The robot arm parts could be seen in figure 11 and the name of each part is given below:

1. Base Plate
2. Bowl
3. Base Retainer
4. Bottom Plate
5. Connecting Holder
6. Connecting Kit
7. First Arm Links
8. Second Arm Links
9. Servo Motor
10. Caster Wheel
11. Spoon End-Effector

In this section, how to design a robot arm and robot arm parts' specifications subjects are worked.

The mechanical design of the robot arm is based on a robot manipulator with similar functions to a human arm. Robotic arm system often consists of links, joints, actuators, sensors, and controllers. The links are connected by joints to form an open kinematic chain. One end of the chain is attached to the robot base plate which has two servo

motors connection points, and another end is equipped with a spoon manipulator. The links of the manipulator are considered to form a kinematic chain. In a robotic system, the number of degrees of freedom is determined by the number of independent joints. Complexity and cost are increased as a number of degrees of freedom. In our robotic arm design with only four degrees of freedom is designed because it is enough for most of the necessary movement. Specification of each joint is given below:

Table 4: *Specific DOF of the servo motors that were used*

Degree Of Freedom	Rotation Degree	Servo Motor
1	300°	Dynamixel AX-18A
2	180°	2×Dynamixel AX-12A
3	180°	Dynamixel AX-18A
4	90°	Dynamixel AX-12A

In the second joint, two Dynamixel AX-12A servo motors are used to increase torque value and to decrease the cost.

The area that the spoon end-effector can reach is called robot arms workspace. The degree of freedoms, arm link lengths, translation limitations, and robot configuration affect the robot arm workspace. So, keeping all these things in mind, the assistant robot arm is designed considering these configurations based on the previous design.

In this process steps that are given below are carried out:

- The number of degrees of freedom is determined.
- The workspace of our robot arm is determined similar to a human arm and it could be seen in figure 10.
- The arm link lengths and the spoon manipulator length are determined based on motor-torque calculations and academicals design and these parts could be seen in figures 18,19 and 22.
- The length of other parts of our design is determined based on the specification of the servo motors. These are:
    - Base Retainer
    - Bottom Plate
    - Connecting Holder
    - Connecting Kit also these parts could be seen in figures 14,15,16 and 17.
- The base plate is designed based on the workspace and this part could be seen in figure 12.
- The bowl is designed to operate the robot arm effectively and to gain an aesthetic appearance to our system and this part could be seen in figure 13.
- The assembly of our design is completed to realize the static analysis and simulation of our robot arm.

3.2.1. Base Plate

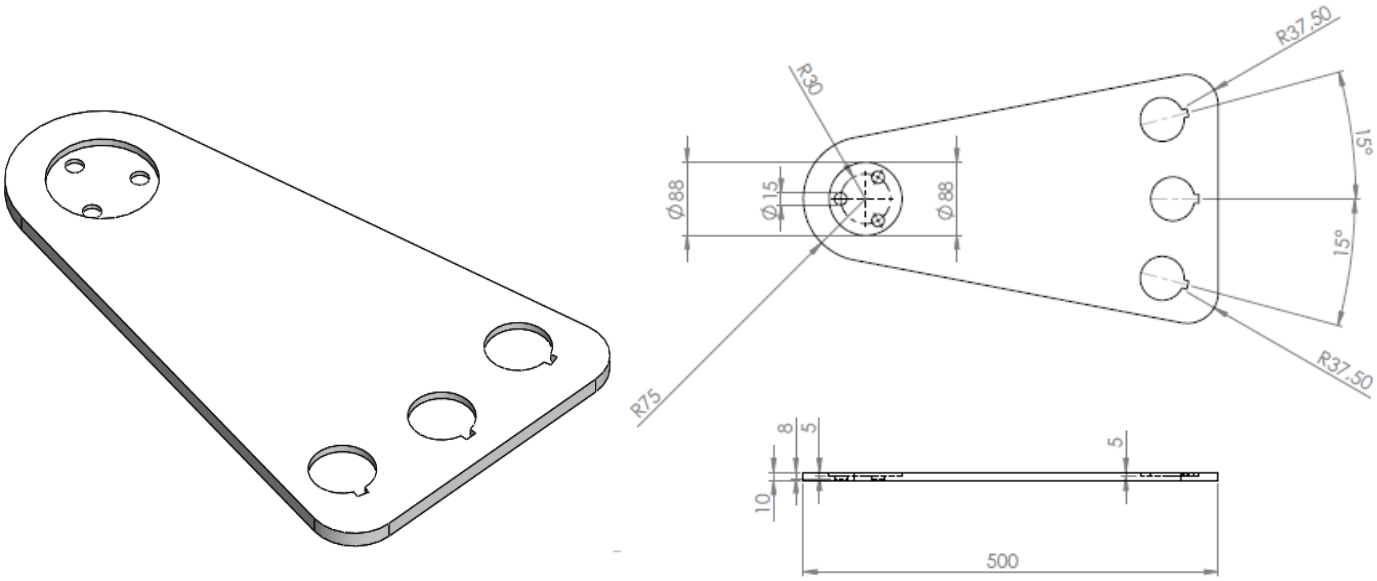

***Figure 12**: Technical Drawing and Trimetric Views of the Base Plate*

3.2.2. Bowl

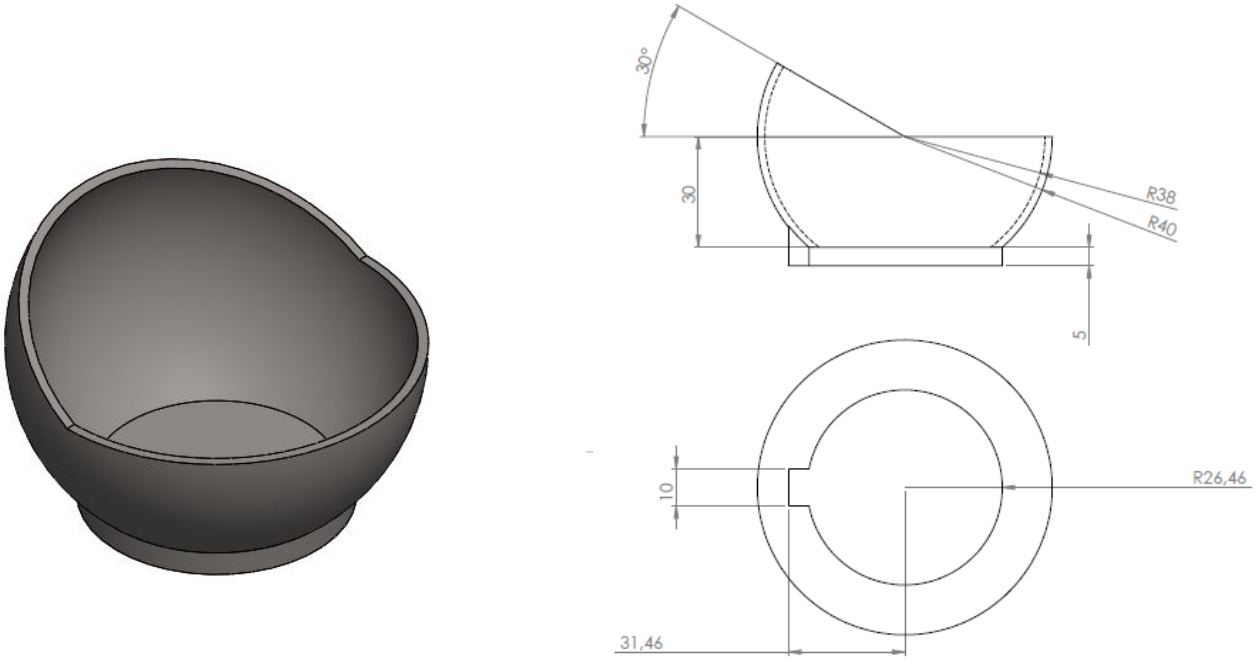

***Figure 13:** Technical Drawing and Trimetric Views of the Bowl*

3.2.3. Base Retainer

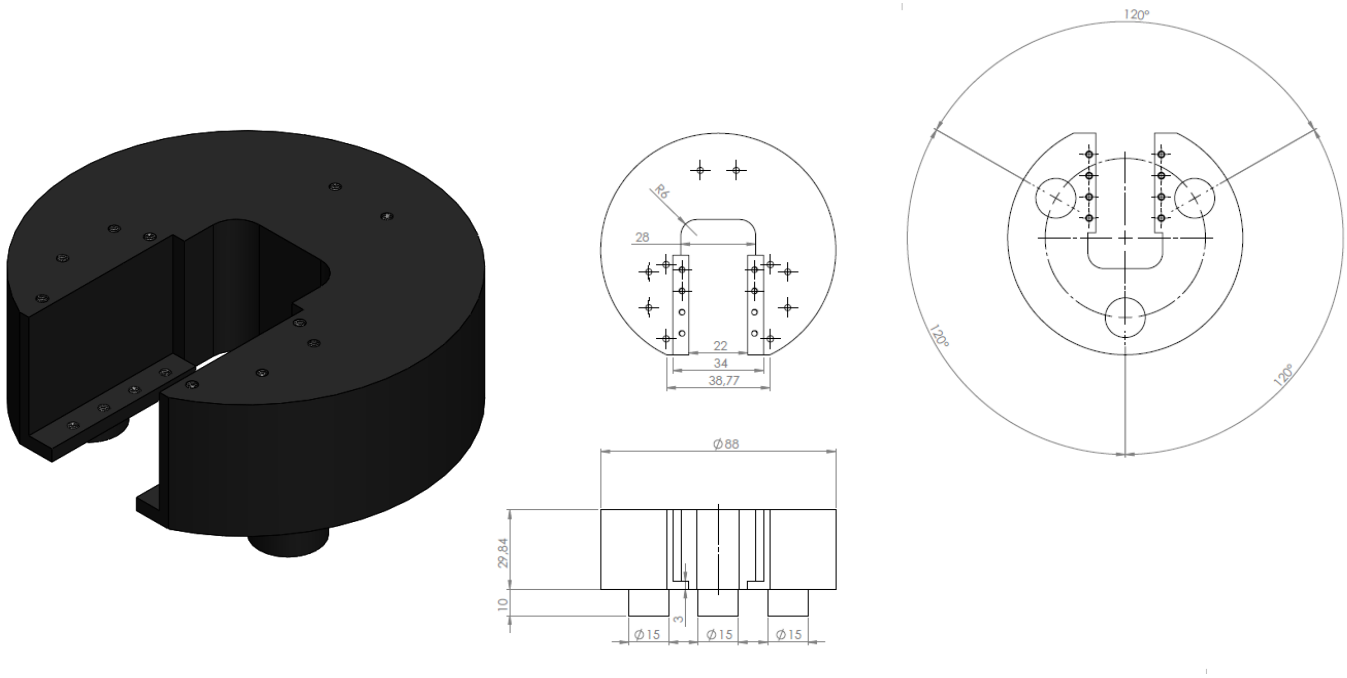

Figure 14: *Technical Drawing and Trimetric Views of the Base Retainer*

3.2.4. Bottom Plate

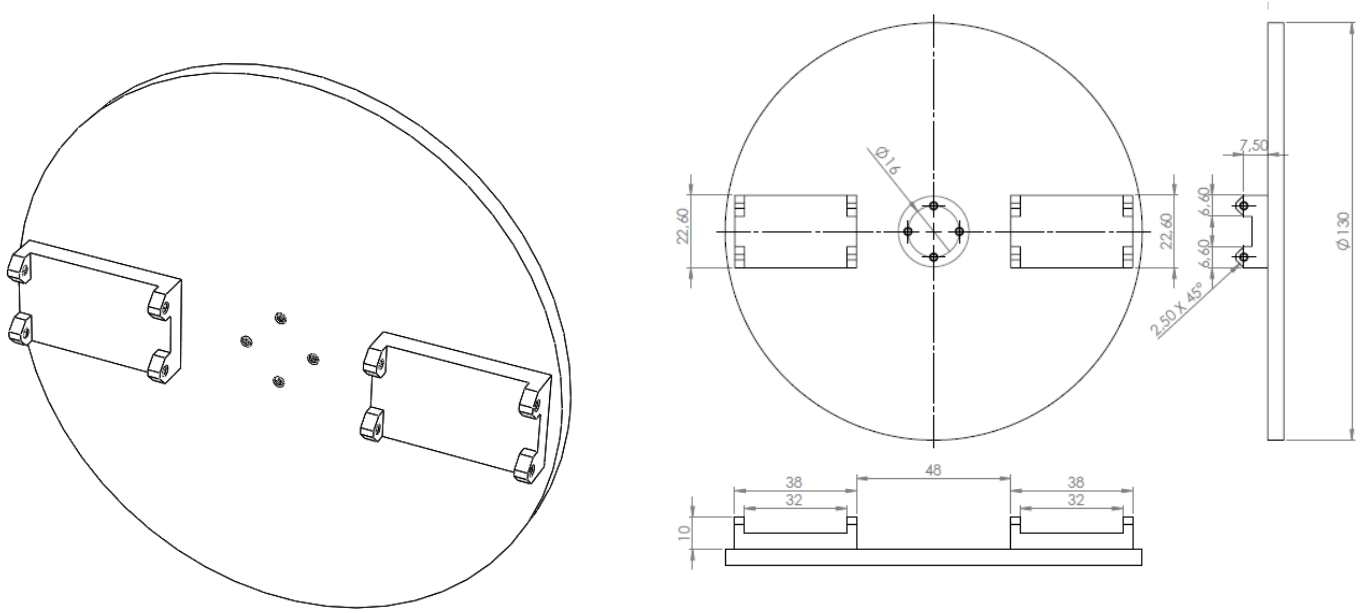

Figure 15: *Technical Drawing and Trimetric Views of the Bottom Plate*

3.2.5. Connecting Holder

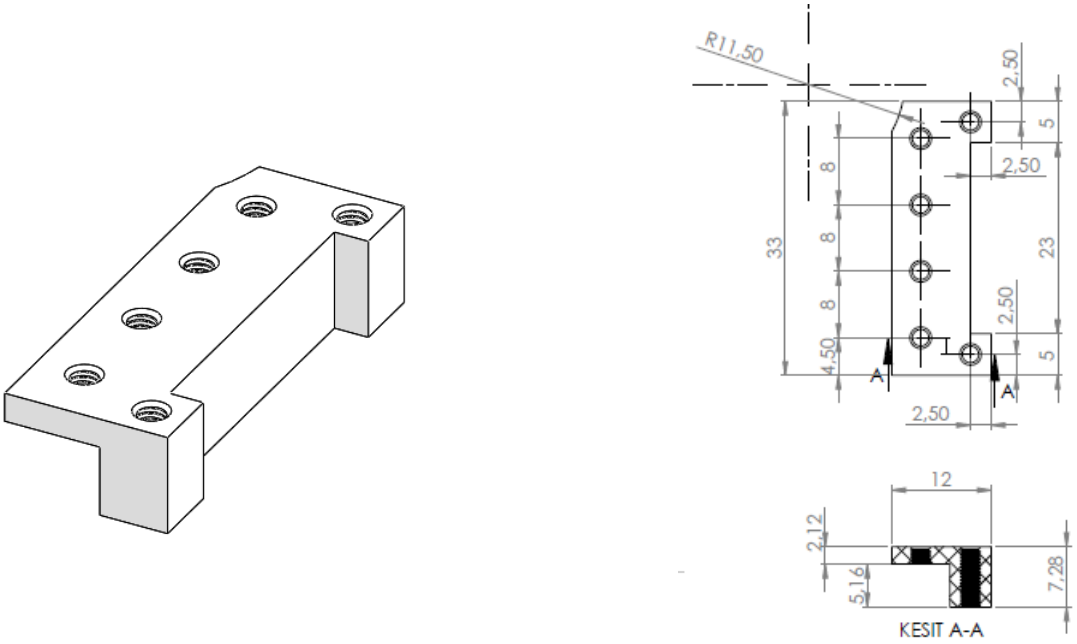

Figure 16: *Technical Drawing and Trimetric Views of the Connecting Holder*

3.2.6. Connecting Kit

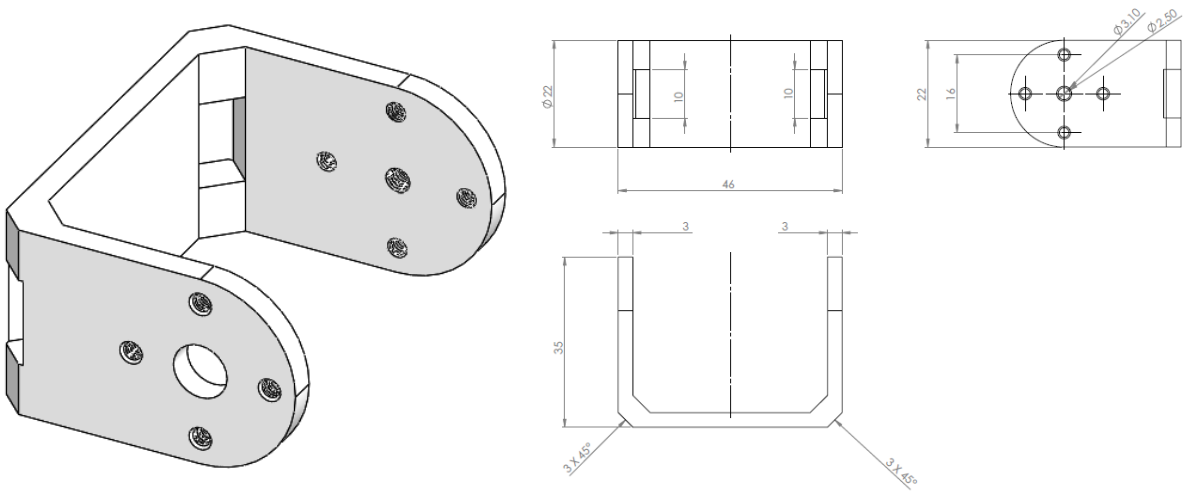

Figure 17: *Technical Drawing and Trimetric Views of the Connecting Kit*

3.2.7. First Arm Link

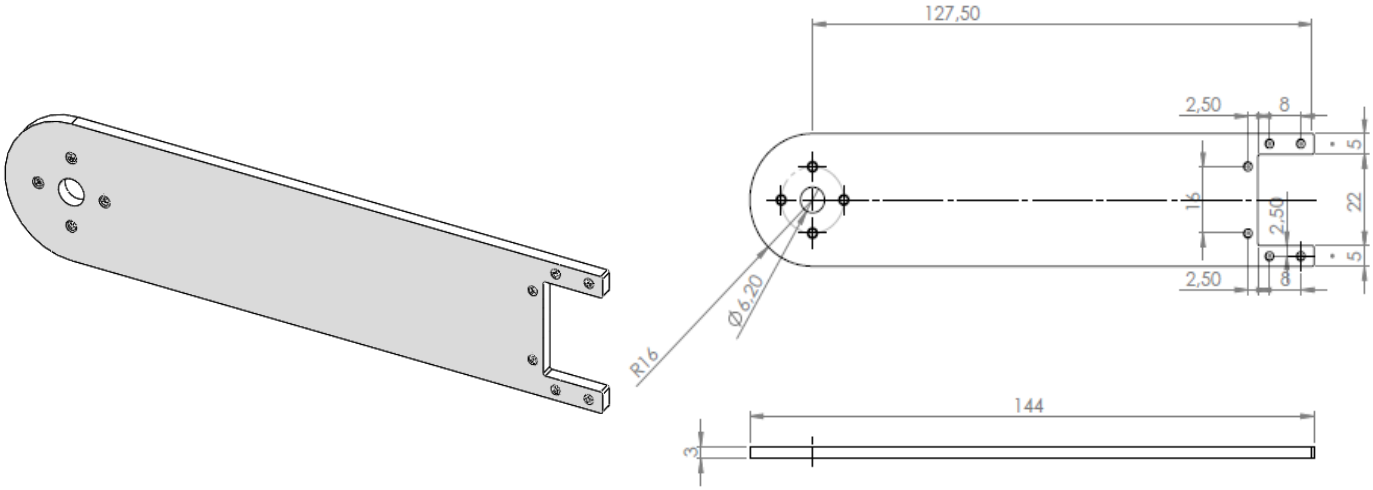

Figure 18: *Technical Drawing and Trimetric Views of the First Arm Link*

3.2.8. Second Arm Link

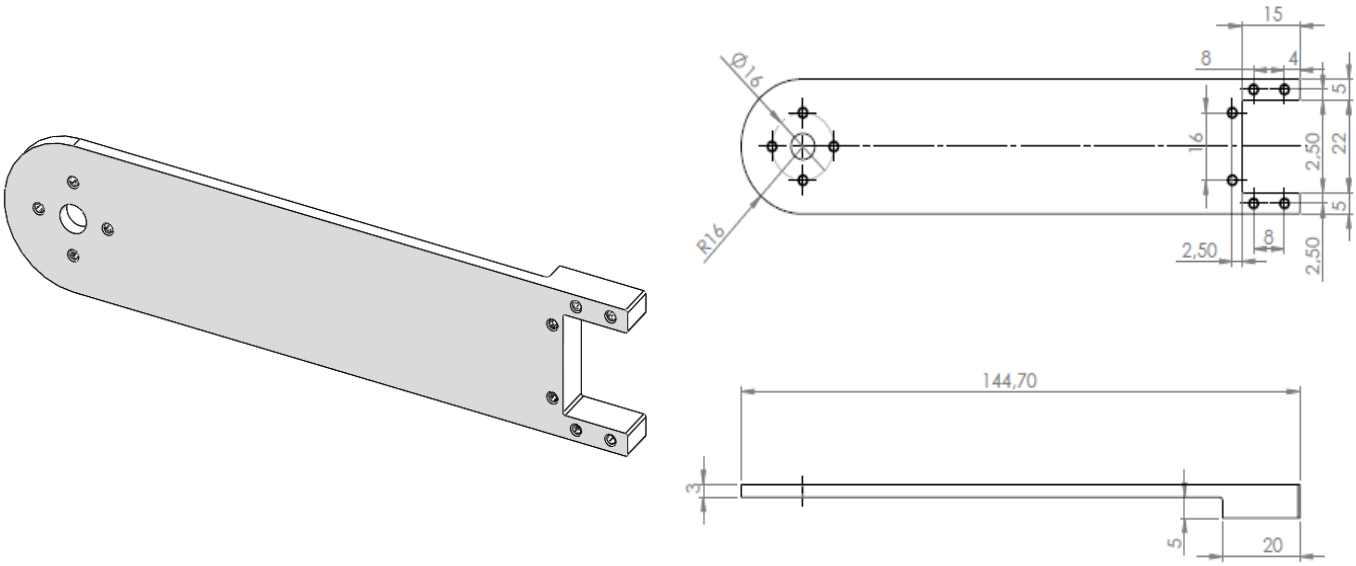

Figure 19: *Technical Drawing and Trimetric Views of the Second Arm Link*

3.2.9. Servo Motor

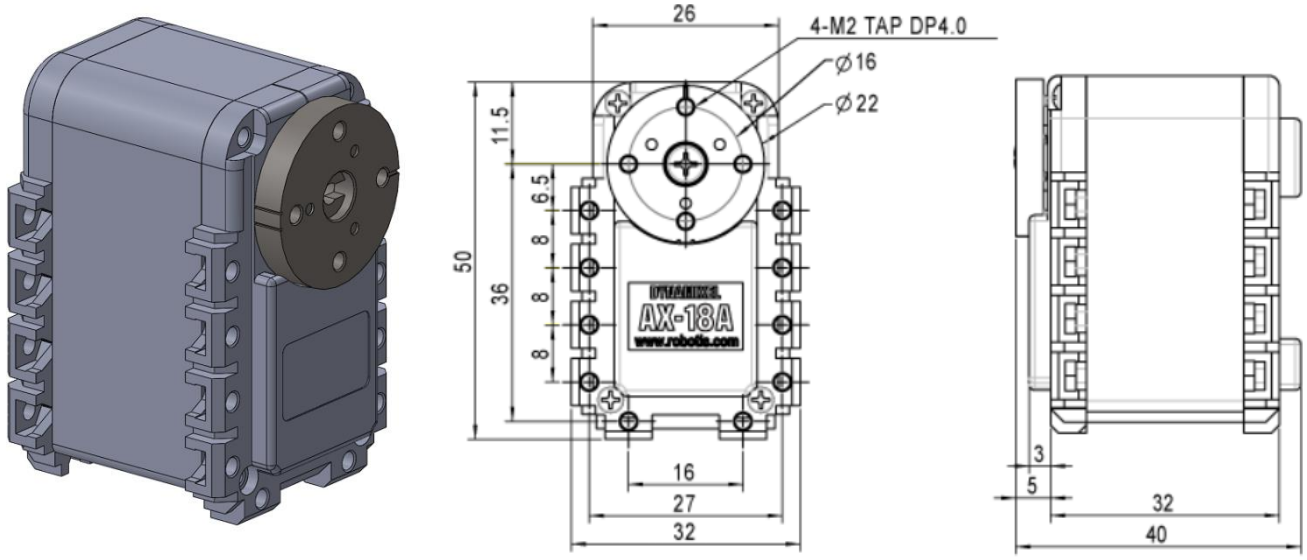

Figure 20: *Technical Drawing and Trimetric Views of the Servo Motor*

3.2.10. Wheel

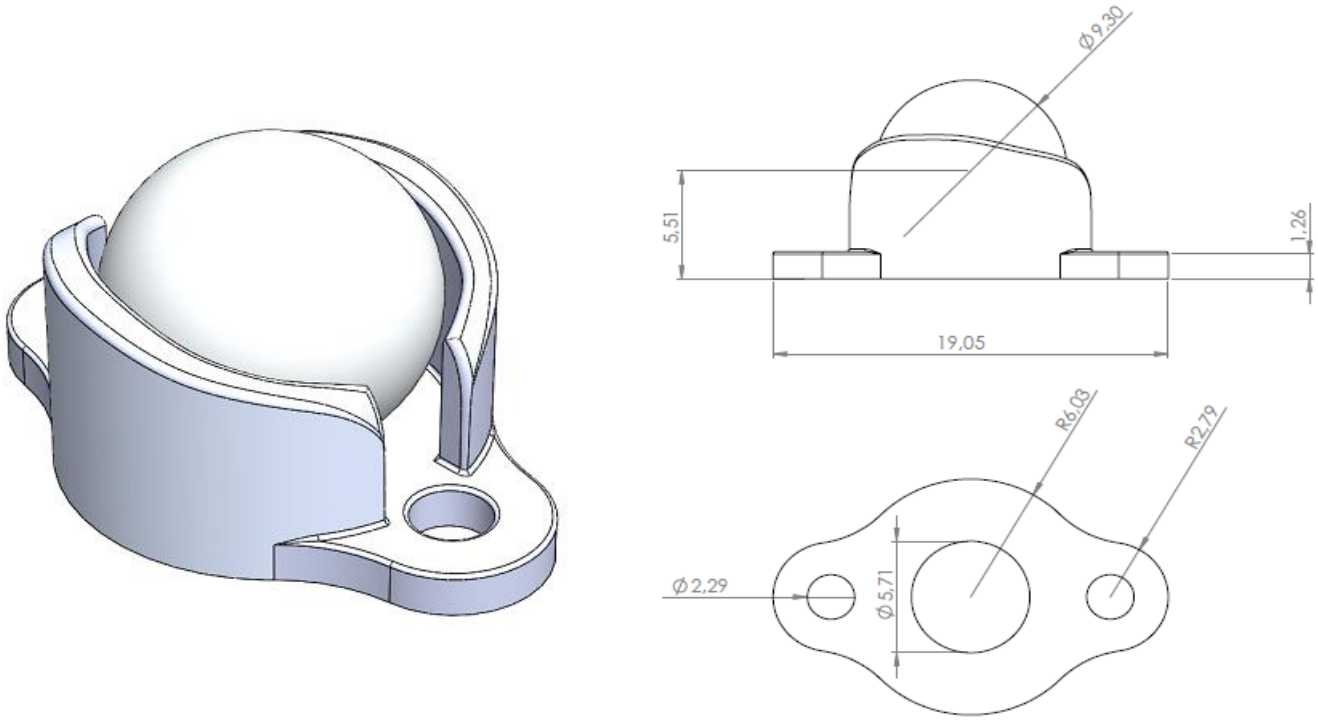

Figure 21: *Technical Drawing and Trimetric Views of the Wheel*

3.2.11. Spoon End-Effector

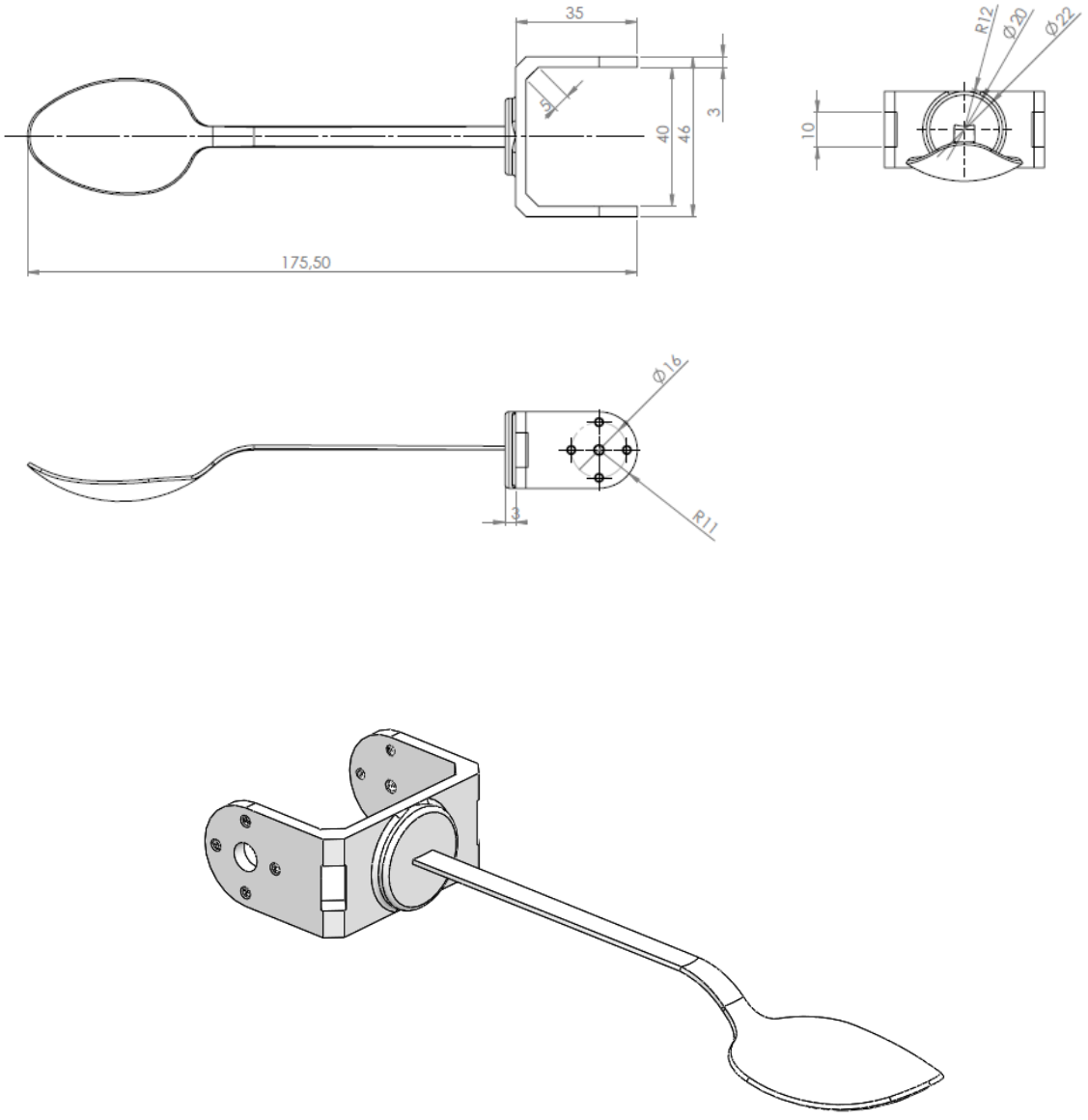

Figure 22: *Technical Drawing and Trimetric Views of the Spoon End-Effector*

3.3. Electromechanical Design of the Feeding Assistance Robot

In the selection of servo motors, some technical requirements were considered. In the following of this part of the report, these requirements and detailed explanations about how the selected servo motor satisfies them will be given. Also, there are some factors other than technical requirements, and these are cost, reliability, availability, and difficulty of the control of the motor. This factor also will be explored in the following document.

The first technical requirement that our system must satisfy that our system should be able to move smoothly, which guarantees that the movement of the spoon (which is also our end-effector) without spilling out the food on it. In order to meet this requirement, the servo motor must have a small resolution. It could be verified from the values that were reached in simulations. For example, the mean value of the change in the second joint during the trajectory following from the home position to the bowl in the middle was $0.0108\,rad$ which is equal to $0.62\,degrees$. From this evaluation, it could be easily seen that our servo motor in the second joint must have a smaller resolution than $0.62\,degrees$. To provide this smooth trajectory following the servo motors also have to have a small response time. It could be confirmed from the simulation, the system works properly when $0.02\,s$ was chosen as the interval time between two different trajectory points. So servo motors must have a smaller response time than $0.02\,s$. Another thing that our servo motors have to maintain was that these motors should be able to feedback some values to our system. For example, shaft position, speed, and load. These values will provide to our robot arm to keep a convenient movement around its workspace.

Secondly, our system must sustain the user's confidence and maintain a safe operation. In order to ensure these, our system has to follow trajectories smoothly without any sudden movement, this requirement was talked in the above immensely, also it should have collision detection. This collision detection feature could be added to our system with servo motors that could feedback the load value.

Other than these technical requirements our servo motors must maintain required torque values both in static conditions and dynamic conditions. The static condition of the robot arm and required torque values are stated in Figure 24. The dynamic conditions were explored in the next chapters in this report.

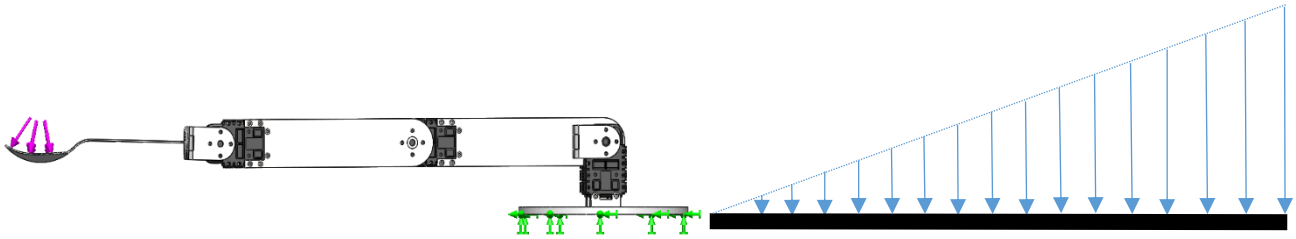

Figure 23: *(a) Position of the maximum torque of the robotic arm (b) Representation of torque distribution on the robotic arm*

As shown in Figure 23, the orientation of the robot must be like that to obtain maximum torque in each link and joints because of the minimize of the cosine angle of joints. Maximum Torque will be in the farthest point because of the torque formula which is $\tau = Fr$.

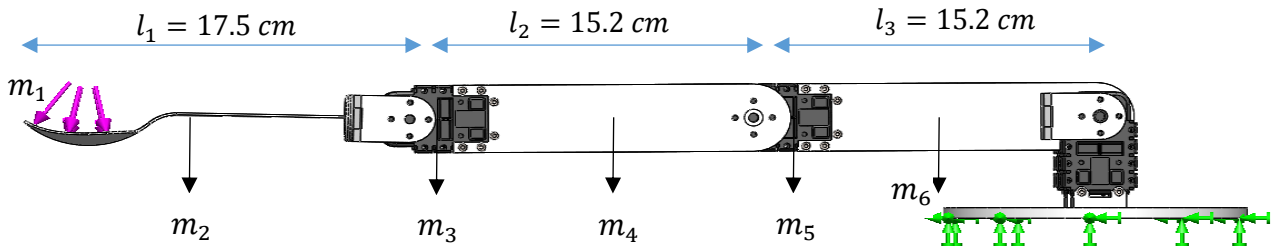

Figure 24: *Position of the maximum torque of the robotic arm in static condition*

$m_1 = 0.06\ gram$ $m_2 = 9.59\ gram$ $m_3 = 58\ gram$

$m_4 = 14.24x2\ gram$ $m_5 = 58\ gram$ $m_6 = 12.59x2\ gram$

While finding the maximum torque value of the last joint, the load being lifted and the mass of the end-effector are taken into account. Mass of the end-effector is equal to 9.59 grams and the load being lifted is equal to 6 grams. Otherwise, the length of the end-effector is equal to 17.5 cm.

$$\tau_1 = m_1 l_1 + m_2 \frac{l_1}{2} = 6x10^{-3}x17.5 + 9.59x10^{-3}x\frac{17.5}{2} = 0.1889125\ kgcm$$

In the second joint, the mass of link 2 is equal to 14.24 grams and the mass of the servo motor is equal to 58 grams. And, the length of the second link is equal to 15.2 cm.

$$\tau_2 = m_1(l_1 + l_2) + m_2\left(\frac{l_1}{2} + l_2\right) + m_3 l_2 + m_4 \frac{l_2}{2}$$

$$6x10^{-3}x(17.5 + 15.2) + 9.59x10^{-3}x\left(\frac{17.5}{2} + 15.2\right) + 58x10^{-3}x15.2 + 14.24x10^{-3}x\frac{15.2}{2} = 1.524\ kg\ cm$$

In the first joint, the mass of link 3 is equal to 12.59 grams and the mass of the servo motor is equal to 58 gr. And, the length of the first link is equal to 15.2 cm.

$$\tau_3 = m_1(l_1 + l_2 + l_3) + m_2\left(\frac{l_1}{2} + l_2 + l_3\right) + m_3(l_2 + l_3) + m_4\left(\frac{l_2}{2} + l_3\right) + m_5 l_3 + m_6 \frac{l_3}{2} = 4.148\ kgcm$$

where τ_1, τ_2, τ_3 are torque values required for those joints. In this analysis, the torque value required for the first joint wasn't calculated because there is no torque affects the rotation of this joint.

In addition to these technical requirements, our servo motors must be user-friendly. This means that the user should be able to control the motor easily. Some servo motors exist

in the market were investigated. In sum, it was decided that a servo motor that has its own Software Development Kit (SDK) in which the user controls it with any programming language must be used in our robot arm.

In the above, technical requirements and features that our servo motors must have to satisfy these technical requirements were explained in detail. In the following parts, the servo motor that we used in our project will be investigated.

The Dynamixel AX18-A servo motor was chosen for our project. It provides feedback on shaft position, speed, temperature, voltage and load values. It has its own SDK which enables us to control the robot. This SDK is compatible with many of the programming languages like C++, Python, and Matlab, and one could control this servo motor just a couple of lines of Python code. One other advantage of this motor is that it does not need an extra circuit like a servo controller, it just needs a USB to TTL converter and 11.1 V power supply. Finally, as could be seen in figure 20, it also has many connection points and a nice case.

Figure 25: *Dynamixel AX 18-A servo motor*

As we mentioned in the above, the servo motor has to be chosen such that it has a resolution lower than $0.62\,degrees$. However, the servo motor has a resolution $0.29\,degrees$ which is pretty enough for our system.

Also, the response time requirement was talked about. And, it was decided that the motor must have a response time lower than $0.02\,s$. In our case, the servo motor's response time is $0.254\,\mu s$. It means that our servo motor could perform different actions in $0.254\,\mu s$ and this is pretty enough for our system.

Other than these specifications it has a compatible size and weight with our robot arm design. The size of the motor is $32 \times 50 \times 40\,mm$ while the weight is $54.5\,gram$. Especially, the lightweight specification of the servo motor made our joint designs more efficient.

The final feature of the servo motor that it could be easily connected to the computer as it could be seen in figure 26. Which enables us to control the motor easily. In addition to

this, there is no need for a voltage regulator, and it could be used with connecting the power cable from the power source as it is seen from the figure 27, in our case the power source was 11.1 V 3S Lipo battery from ProFuse brand.

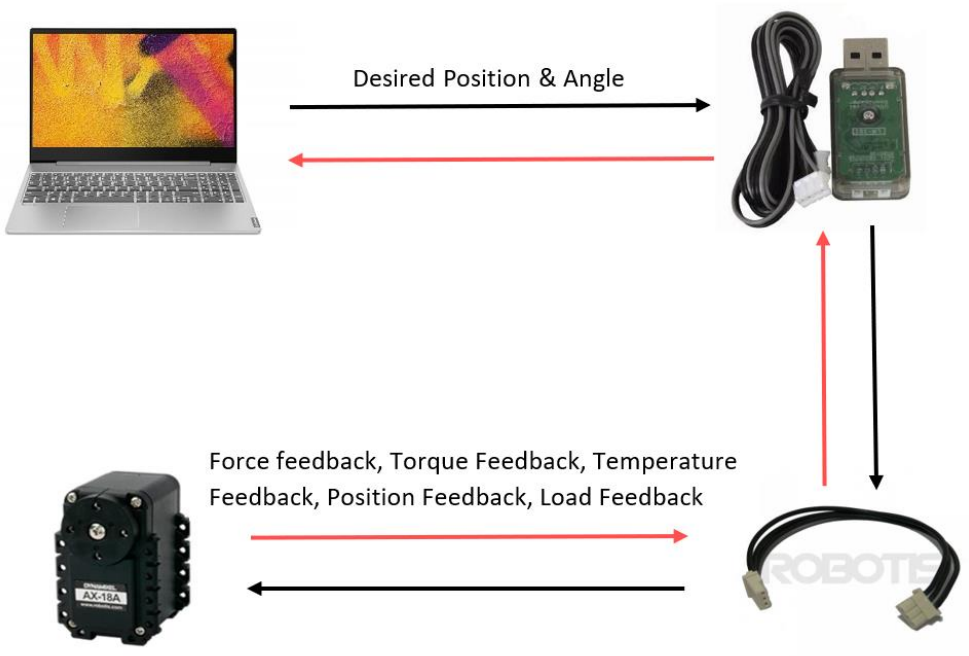

Figure 26: *General Diagram for the control of the servo motor*

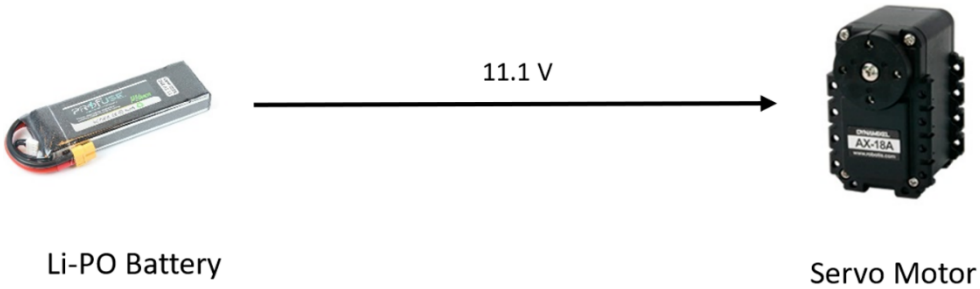

Figure 27: *Electrical connection of the servo motor*

Our servo motor's maximum current is $2200\,mA$ and none of the servo motors in our system work with above the half of this value. This means that our servo motors use a maximum of $1100\,mA$ with their current radial speeds and loads. According to these assumptions working time could be determined from the following equations:

$$Battery\ time(hours) = \frac{Battery\ Capacity(mAh)}{Current(mA)} \times 0.707$$

$$= \frac{2800\ mAh}{4400\ mAh} \times 0.0707 = 0.44\ hours$$

To sum, our servo motors are quite suitable for our project with its high technical specifications, its SDK, and its plugin connection type. All of these give us a huge advantage of the testing of the robot arm to make our system more robust.

3.4. Analyses of the Feeding Assistance Robot

In this title of the report, Analyses of the Feeding Assistance Robot will be mentioned. The background information about the analyses was given in the previous section.

3.4.1. Kinematic Analysis of the Robotic Arm

The kinematic equations of the robot that has 4 DOF were founded. In forward kinematics, Denavit – Hartenberg Parameters were used. In inverse kinematics, the standard trigonometric formula was used.

3.4.1.1. Forward Kinematic Analysis of the Robotic Arm

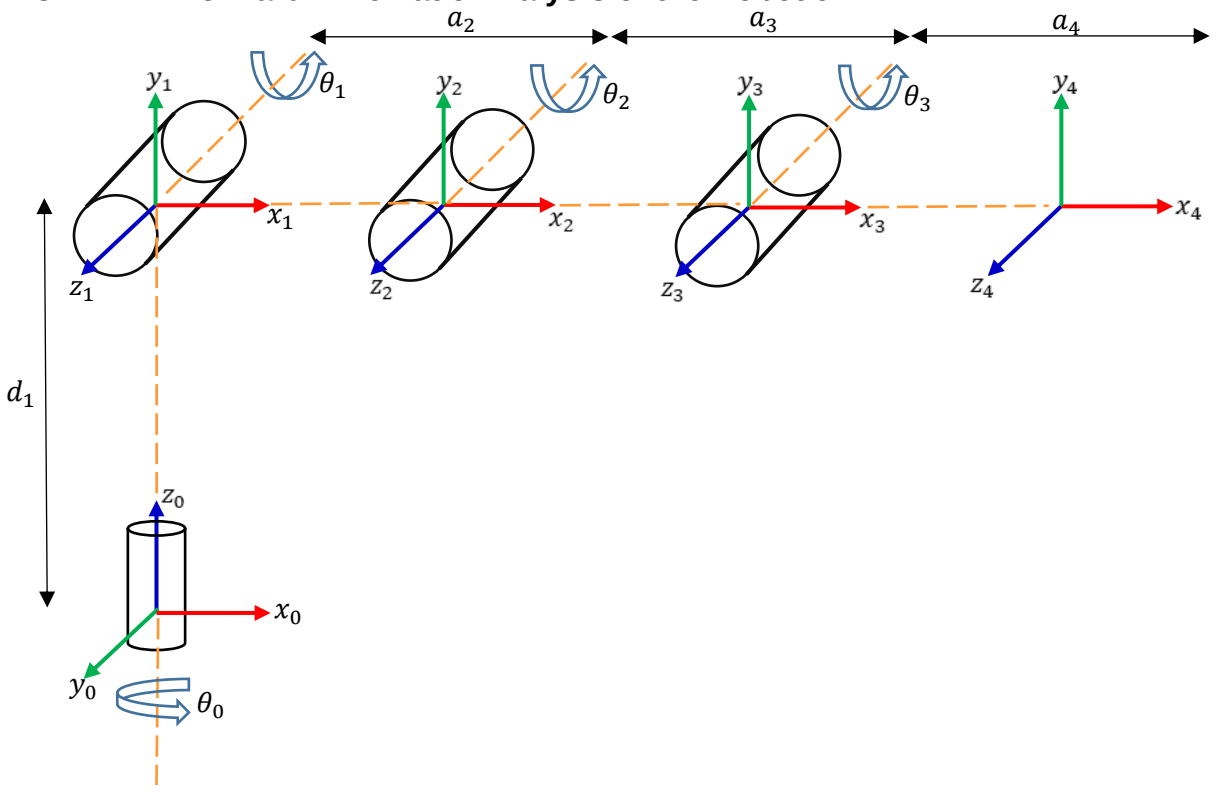

Figure 28: *Coordinate frames attached to elbow manipulator*

In Figure 28, coordinate frames are demonstrated. z_i axes are selected because of the motion axis which is revolute joints. x_i axes are selected to be perpendicular to z_{i-1} and z_i. Hereby, the Denavit – Hartenberg (D-H) Parameters were obtained as follow:

Table 5: *Denavit – Hartenberg Parameters Table of the Feeding Assistance Robot*

Axis Number i	Link Length a_i	Twist Angle α_i	Link offset d_i	Joint angle θ_i
1	0	$\pi/2$	d_1	θ_1^*
2	a_2	0	0	θ_2^*
3	a_3	0	0	θ_3^*
4	a_4	0	0	θ_4^*

By using D-H Parameters, the homogenous transformation matrices are obtained.

$$A_1 = T_1^0 = \begin{bmatrix} cos\theta_1 & 0 & sin\theta_1 & 0 \\ sin\theta_1 & 0 & -cos\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = T_2^1 = \begin{bmatrix} cos\theta_2 & -sin\theta_2 & 0 & a_2\, cos\theta_2 \\ sin\theta_2 & cos\theta_2 & 0 & a_2 sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = T_3^2 = \begin{bmatrix} cos\theta_3 & -sin\theta_3 & 0 & a_3\, cos\theta_3 \\ sin\theta_3 & cos\theta_3 & 0 & a_3 sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = T_4^3 = \begin{bmatrix} cos\theta_4 & -sin\theta_4 & 0 & a_4\, cos\theta_4 \\ sin\theta_4 & cos\theta_4 & 0 & a_4 sin\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, the arm matrix is created by multiplying each matrix. T_4^0 matrix defined as follows,

$$T_4^0 = \prod_{i=1}^{4} T_i^{i-1} = T_1^0\, T_2^1\, T_3^2\, T_4^3 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix} = A_1 A_2 A_3 A_4$$

$r_{11} = c_1c_4c_{23} - c_1s_4s_{23} = \cos(\theta_2 + \theta_3 + \theta_4)\cos(\theta_1)$
$r_{12} = -c_1s_4c_{23} - c_1c_4s_{23} = -\sin(\theta_2 + \theta_3 + \theta_4)\cos(\theta_1)$
$r_{13} = s_1$
$r_{14} = a_2c_1c_2 + a_4c_1c_4c_{23} - a_4c_1s_4s_{23} + a_3c_1c_{23}$
$r_{21} = c_4s_1c_{23} - s_1s_4s_{23} = \cos(\theta_2 + \theta_3 + \theta_4)\sin(\theta_1)$
$r_{22} = -s_1s_4c_{23} - c_4s_1s_{23} = -\sin(\theta_2 + \theta_3 + \theta_4)\sin(\theta_1)$
$r_{23} = -c_1$
$r_{24} = a_2c_2s_1 + a_4c_4s_1c_{23} - a_4s_1s_4s_{23} + a_3s_1c_{23}$
$r_{31} = c_4s_{23} + s_4c_{23} = \sin(\theta_2 + \theta_3 + \theta_4)$
$r_{32} = c_4c_{23} - s_4s_{23} = \cos(\theta_2 + \theta_3 + \theta_4)$
$r_{33} = 0$
$r_{34} = d_1 + a_2s_2 + a_4c_4s_{23} + a_4s_4c_{23} + a_3s_{23}$
$r_{41} = 0$
$r_{42} = 0$
$r_{43} = 0$
$r_{44} = 1$

In the expressions of these elements of the arm matrix, the variables are defined as follow:

$$c_i = cos\theta_i \quad , s_i = sin\theta_i \quad , c_{ij} = cos(\theta_i + \theta_j) \ , s_{ij} = sin(\theta_i + \theta_j)$$

And arm matrix (T_4^0) consist of as indicated:

$$\Rightarrow \begin{bmatrix} c_{234}c_1 & -s_{234}c_1 & s_1 & a_2c_1c_2 + a_4c_1c_4c_{23} - a_4c_1s_4s_{23} + a_3c_1c_{23} \\ c_{234}s_1 & -s_{234}s_1 & -c_1 & a_2c_2s_1 + a_4c_4s_1c_{23} - a_4s_1s_4s_{23} + a_3s_1c_{23} \\ s_{234} & c_{234} & 0 & d_1 + a_2s_2 + a_4c_4s_{23} + a_4s_4c_{23} + a_3s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It is possible to define the values of the translational transform (P_x, P_y, P_z) with respect to the base coordinate system by using arm matrix.

$$P_x = a_2cos\theta_1cos\theta_2 - a_4cos\theta_1cos\theta_4(sin\theta_2sin\theta_3 - cos\theta_2cos\theta_3)$$
$$-a_4cos\theta_1sin\theta_4(cos\theta_2sin\theta_3 + cos\theta_3sin\theta_2) + a_3cos\theta_1(cos\theta_2cos\theta_3 - sin\theta_2sin\theta_3)$$

$$P_y = a_2cos\theta_2sin\theta_1 - a_4cos\theta_4sin\theta_1(sin\theta_2sin\theta_3 - cos\theta_2cos\theta_3)$$
$$-a_4sin\theta_1sin\theta_4(cos\theta_2sin\theta_3 + cos\theta_3sin\theta_2) + a_3sin\theta_1(cos\theta_2cos\theta_3 - sin\theta_2sin\theta_3)$$

$$P_z = d_1 + a_2sin\theta_2 + a_4cos\theta_4(cos\theta_2sin\theta_3 + cos\theta_3sin\theta_2) + a_4sin\theta_4(cos\theta_2cos\theta_3 - sin\theta_2sin\theta_3) + a_3(cos\theta_2sin\theta_3 + cos\theta_3sin\theta_2)$$

Also, the arm matrix can be demonstrated more clearly as follow:

$$\begin{bmatrix} c_{234}c_1 & -s_{234}c_1 & s_1 & c_1(a_3c_{23} + a_2c_2 + a_4c_{234}) \\ c_{234}s_1 & -s_{234}s_1 & -c_1 & s_1(a_3c_{23} + a_2c_2 + a_4c_{234}) \\ s_{234} & c_{234} & 0 & d_1 + a_3s_{23} + a_2s_2 + a_4s_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.4.1.2. Inverse Kinematic Analysis of the Robotic Arm

The elements of translational transform (P_x, P_y, P_z) can be demonstrated in shorter notation by using simplify() command on the command window in MATLAB.

$$P_x = a_2cos\theta_1cos\theta_2 + a_4cos\theta_1cos\theta_4\, cos(\theta_2 + \theta_3) - a_4cos\theta_1sin\theta_4\, sin(\theta_2 + \theta_3) + a_3cos\theta_1cos\,(\theta_2 + \theta_3)$$

- $P_x \Rightarrow cos\theta_1\big(a_3\, cos(\theta_2 + \theta_3) + a_2cos\theta_2 + a_4cos(\theta_2 + \theta_3 + \theta_4)\big)$

$$P_y = a_2cos\theta_2sin\theta_1 + a_4cos\theta_4sin\theta_1\, cos(\theta_2 + \theta_3) - a_4sin\theta_1sin\theta_4\, sin(\theta_2 + \theta_3) + a_3sin\theta_1cos\,(\theta_2 + \theta_3)$$

- $P_y \Rightarrow sin\theta_1\big(a_3\, cos(\theta_2 + \theta_3) + a_2cos\theta_2 + a_4cos(\theta_2 + \theta_3 + \theta_4)\big)$

$$P_z = d_1 + a_2sin\theta_2 + a_4cos\theta_4sin\,(\theta_2 + \theta_3) + a_4sin\theta_4cos\,(\theta_2 + \theta_3) + a_3sin\,(\theta_2 + \theta_3)$$

- $P_z \Rightarrow d_1 + a_3\sin\,(\theta_2 + \theta_3) + a_2sin\theta_2 + a_4sin(\theta_2 + \theta_3 + \theta_4)$

$$\frac{P_y}{P_x} = \frac{sin\theta_1}{cos\theta_1} = tan\theta_1 \Rightarrow \theta_1 = tan^{-1}\left(\frac{P_y}{P_x}\right)\ or\ \theta_1 = Atan2(P_y, P_x)$$

If P_x is multiplied by c_1 and P_y is multiplied by s_1, A can be obtained explicitly.

$$P_x.\, c_1 + P_y.\, s_1 = a_3\cos(\theta_2 + \theta_3) + a_2\cos\theta_2 + a_4\cos(\theta_2 + \theta_3 + \theta_4) = A$$

$$c_{23} = \frac{\big(P_x.\, c_1 + P_y.\, s_1\big) - a_2\cos\theta_2 + a_4\cos(\theta_2 + \theta_3 + \theta_4)}{a_3}$$

And, also we obtained s_{23} by solving P_z.

$$s_{23} = \frac{P_z - d_1 - a_2 sin\theta_2 - a_2 sin(\theta_2 + \theta_3 + \theta_4)}{a_3}$$

Substituting last two equations into the $c_{23}^2 + s_{23}^2 = 1$

$$\Big(\big(P_x.c_1 + P_y.s_1\big) - a_2\cos\theta_2 + a_4\cos(\theta_2 + \theta_3 + \theta_4)\Big)^2 + \big(P_z - d_1 - a_2 sin\theta_2 - a_2 sin(\theta_2 + \theta_3 + \theta_4)\big)^2 = a_3^2$$

$$\Big(\big(P_x.c_1 + P_y.s_1 + a_4\cos(\theta_2 + \theta_3 + \theta_4)\big) - a_2\cos\theta_2)\Big)^2 + \Big(\big(P_z - d_1 - a_2 sin(\theta_2 + \theta_3 + \theta_4)\big) - a_2 sin\theta_2\Big)^2 = a_3^2$$

$$\big(P_x.c_1 + P_y.s_1 + a_4\cos(\theta_2 + \theta_3 + \theta_4)\big)^2 - 2\big(P_x.c_1 + P_y.s_1 + a_4\cos(\theta_2 + \theta_3 + \theta_4)\big)a_2\cos\theta_2) + a_2^2c_2^2 + (P_z - d_1 - a_2 sin(\theta_2 + \theta_3 + \theta_4)^2 - 2(P_z - d_1 - a_2 sin(\theta_2 + \theta_3 + \theta_4)a_2 sin\theta_2 + a_2^2 s_2^2 \quad = a_3^2$$

The equation can be regulated more comprehensible:

$$f = \big(P_z - d_1 - a_2 sin(\theta_2 + \theta_3 + \theta_4)\big)$$

$$\big(P_x.c_1 + P_y.s_1 + a_4\cos(\theta_2 + \theta_3 + \theta_4)\big)\cos\theta_2 + \big(P_z - d_1 - a_2 sin(\theta_2 + \theta_3 + \theta_4)\big)sin\theta_2 =$$

$$\Rightarrow \frac{\big(P_x.c_1 + P_y.s_1 + a_4\cos(\theta_2 + \theta_3 + \theta_4)\big)^2 + a_2^2 + \big(P_z - d_1 - a_2 sin(\theta_2 + \theta_3 + \theta_4)\big)^2 - a_3^2}{2a_2} = A$$

$$g = P_x.c_1 + P_y.s_1 + a_4\cos(\theta_2 + \theta_3 + \theta_4)$$

$$h = \frac{\big(P_x.c_1 + P_y.s_1 + a_4\cos(\theta_2 + \theta_3 + \theta_4)\big)^2 + a_2^2 + \big(P_z - d_1 - a_2 sin(\theta_2 + \theta_3 + \theta_4)\big)^2 - a_3^2}{2a_2}$$

- $f\ sin\theta_2 + g\cos\theta_2 = h$

If the approximations are considered,

$$g+h\neq 0, \qquad f\sqrt{f^2+g^2-h^2}-f^2-g^2-gh\neq 0\rightarrow\theta_2$$

$$\approx 2.\left(3.14159n+tan^{-1}\left(\frac{f-\sqrt{f^2+g^2-h^2}}{f+g}\right)\right),n\in Z$$

$$g+h\neq 0, \qquad f\sqrt{f^2+g^2-h^2}+f^2+g^2+gh\neq 0\rightarrow\theta_2$$

$$\approx 2.\left(3.14159n+tan^{-1}\left(\frac{f+\sqrt{f^2+g^2-h^2}}{f+g}\right)\right),n\in Z$$

$$f\neq 0\,,\quad f^2+g^2\neq 0,\quad h\approx -f\rightarrow\theta_2\approx 2.\left(3.14159n+tan^{-1}\left(\frac{g}{f}\right)\right),n\in Z$$

$$g=-f\rightarrow\theta_2=2\pi n+\pi,n\in Z$$

And if $g=-f\,,\ x=2\pi n+\pi$ it is possible to obtain as follows:

$$\theta_2=Atan2\left(\frac{gh-\sqrt{f^4+f^2g^2-f^2h^2}}{f^2+g^2},\frac{1}{f}\left(\frac{g\sqrt{-f^2(-f^2-g^2+h^2)}-g^2h}{f^2+g^2}+h\right)\right)$$

$$\theta_2=Atan2\left(\frac{gh+\sqrt{f^4+f^2g^2-f^2h^2}}{f^2+g^2},\frac{1}{f}\left(\frac{-g\sqrt{-f^2(-f^2-g^2+h^2)}-g^2h}{f^2+g^2}+h\right)\right)$$

If we consider c_{23} and s_{23} from the previous equations to obtain $tan(\theta_2+\theta_3)$

$$tan(\theta_2+\theta_3)=\frac{P_z-d_1-a_2sin\theta_2-a_2sin(\theta_2+\theta_3+\theta_4)}{\left(P_x.c_1+P_y.s_1\right)-a_2\cos\theta_2+a_4\cos(\theta_2+\theta_3+\theta_4)}$$

Now, θ_3 can be obtained easily.

$$\theta_3=Atan2\big(P_z-d_1-a_2sin\theta_2-a_2sin(\theta_2+\theta_3+\theta_4),\left(P_x.c_1+P_y.s_1\right)-a_2\cos\theta_2+a_4\cos(\theta_2+\theta_3+\theta_4)\big)-\theta_2$$

$$\frac{P_{y_4^3}}{P_{x_4^3}}=\frac{s_4}{c_4}=\theta_4=Atan2\left(P_{y_4^3},P_{x_4^3}\right)=\theta_{234}-\theta_2-\theta_3$$

3.4.2. Dynamical Analysis of the Robotic Arm

In short definition, Lagrangian Formulation, a variational approach based on the kinetic and potential energy of the robot. Dynamical Analysis of the Robotic Arm was made using Lagrangian Formulation [10].

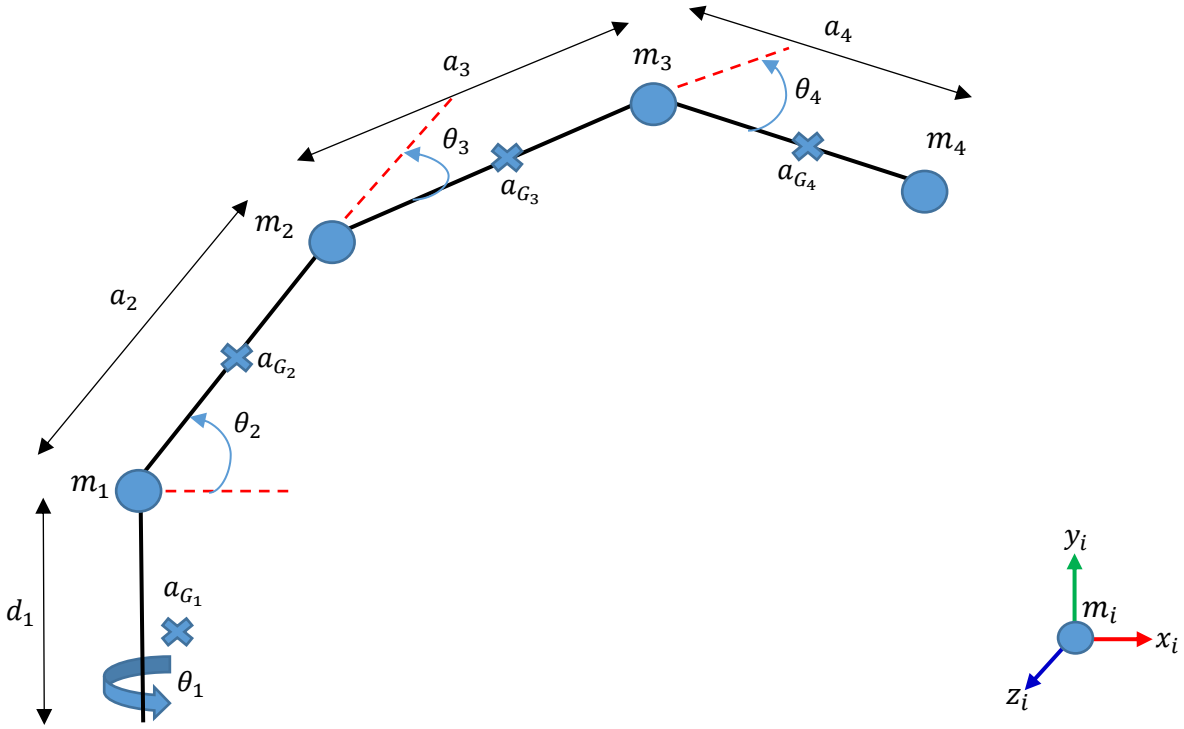

Figure 29: *Positions of the center of masses and link*

3.4.2.1. Defining and calculating the rotation matrices of the system

$$D_0^1 = \begin{bmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad D_1^2 = \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D_2^3 = \begin{bmatrix} c_3 & s_3 & 0 \\ -s_3 & c_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad D_3^4 = \begin{bmatrix} c_4 & s_4 & 0 \\ -s_4 & c_4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D_0^2 = \begin{bmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad D_0^3 = \begin{bmatrix} c_{123} & s_{123} & 0 \\ -s_{123} & c_{123} & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad D_0^4 = \begin{bmatrix} c_{1234} & s_{1234} & 0 \\ -s_{1234} & c_{1234} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3.4.2.2. Defining the angular velocities of the sequential links

$$\omega_0^0(R_0) = \begin{bmatrix}0\\0\\0\end{bmatrix} \qquad \omega_1^0(R_0) = \begin{bmatrix}0\\0\\\dot{\theta}_1\end{bmatrix} \qquad \omega_2^0(R_1) = \begin{bmatrix}0\\0\\\dot{\theta}_2\end{bmatrix}$$

$$\omega_3^0(R_2) = \begin{bmatrix}0\\0\\\dot{\theta}_3\end{bmatrix} \qquad \omega_4^0(R_3) = \begin{bmatrix}0\\0\\\dot{\theta}_4\end{bmatrix}$$

3.4.2.3. Calculating the angular velocities of the overall links

$$\omega_1^0(R_1) = \omega_1^0(R_0) + D_0^1\,\omega_0^0(R_0) = \begin{bmatrix}0\\0\\\dot{\theta}_1\end{bmatrix} + \begin{bmatrix}c_1 & s_1 & 0\\-s_1 & c_1 & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\0\\0\end{bmatrix} = \begin{bmatrix}0\\0\\\dot{\theta}_1\end{bmatrix}$$

$$\omega_2^0(R_2) = \omega_2^1(R_1) + D_1^2\,\omega_1^0(R_1) = \begin{bmatrix}0\\0\\\dot{\theta}_2\end{bmatrix} + \begin{bmatrix}c_2 & s_2 & 0\\-s_2 & c_2 & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\0\\\dot{\theta}_1\end{bmatrix} = \begin{bmatrix}0\\0\\\dot{\theta}_1+\dot{\theta}_2\end{bmatrix}$$

$$\omega_3^0(R_3) = \omega_3^2(R_2) + D_2^3\,\omega_2^0(R_2) = \begin{bmatrix}0\\0\\\dot{\theta}_3\end{bmatrix} + \begin{bmatrix}c_3 & s_3 & 0\\-s_3 & c_3 & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\0\\\dot{\theta}_1+\dot{\theta}_2\end{bmatrix} = \begin{bmatrix}0\\0\\\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3\end{bmatrix}$$

$$\omega_4^0(R_4) = \omega_4^3(R_3) + D_3^4\,\omega_3^0(R_3) = \begin{bmatrix}0\\0\\\dot{\theta}_4\end{bmatrix} + \begin{bmatrix}c_4 & s_4 & 0\\-s_4 & c_4 & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\0\\\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3\end{bmatrix}$$

$$= \begin{bmatrix}0\\0\\\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3+\dot{\theta}_4\end{bmatrix}$$

There are 2 types of the kinetic energy of the system and one of them is translational kinetic energy and the other is rotational kinetic energy.

3.4.2.4. Calculating of the Rotational Kinetic Energy

$$K_{E_1}^{(R)} = \frac{1}{2}[\omega_1^0(R_1)]^T\, I^1\, [\omega_1^0(R_1)] = \frac{1}{2}[0 \quad 0 \quad \dot{\theta}_1].\begin{bmatrix} I_{xx}^1 & I_{xy}^1 & I_{xz}^1 \\ I_{yx}^1 & I_{yy}^1 & I_{yz}^1 \\ I_{zx}^1 & I_{zy}^1 & I_{zz}^1 \end{bmatrix}.\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} = \frac{1}{2} I_{zz}^1 \left(\dot{\theta}_1\right)^2$$

$$K_{E_2}^{(R)} = \frac{1}{2}[\omega_2^0(R_2)]^T\, I^2\, [\omega_2^0(R_2)] = \frac{1}{2}[0 \quad 0 \quad \dot{\theta}_1 + \dot{\theta}_2].\begin{bmatrix} I_{xx}^2 & I_{xy}^2 & I_{xz}^2 \\ I_{yx}^2 & I_{yy}^2 & I_{yz}^2 \\ I_{zx}^2 & I_{zy}^2 & I_{zz}^2 \end{bmatrix}.\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix}$$

$$= \frac{1}{2} I_{zz}^2 \left(\dot{\theta}_1 + \dot{\theta}_2\right)^2 = \frac{1}{2} I_{zz}^2 \left(\dot{\theta}_1\right)^2 + \frac{1}{2} I_{zz}^2 \left(\dot{\theta}_2\right)^2 + \frac{1}{2} I_{zz}^2 \left(\dot{\theta}_1 \dot{\theta}_2\right)$$

$$K_{E_3}^{(R)} = \frac{1}{2}[\omega_3^0(R_3)]^T\, I^3\, [\omega_3^0(R_3)] = \frac{1}{2}[0 \quad 0 \quad \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3].\begin{bmatrix} I_{xx}^3 & I_{xy}^3 & I_{xz}^3 \\ I_{yx}^3 & I_{yy}^3 & I_{yz}^3 \\ I_{zx}^3 & I_{zy}^3 & I_{zz}^3 \end{bmatrix}.\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix}$$

$$= \frac{1}{2} I_{zz}^3 \left(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3\right)^2 = \frac{1}{2} I_{zz}^3 \left(\dot{\theta}_1\right)^2 + \frac{1}{2} I_{zz}^3 \left(\dot{\theta}_2\right)^2 + \frac{1}{2} I_{zz}^3 \left(\dot{\theta}_3\right)^2 + \frac{1}{2} I_{zz}^3 \left(\dot{\theta}_1 \dot{\theta}_2\right) + \frac{1}{2} I_{zz}^3 \left(\dot{\theta}_1 \dot{\theta}_3\right) + \frac{1}{2} I_{zz}^3 \left(\dot{\theta}_2 \dot{\theta}_3\right)$$

$$K_{E_4}^{(R)} = \frac{1}{2}[\omega_4^0(R_4)]^T\, I^4\, [\omega_4^0(R_4)] = \frac{1}{2}[0 \quad 0 \quad \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4].\begin{bmatrix} I_{xx}^4 & I_{xy}^4 & I_{xz}^4 \\ I_{yx}^4 & I_{yy}^4 & I_{yz}^4 \\ I_{zx}^4 & I_{zy}^4 & I_{zz}^4 \end{bmatrix}.\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4 \end{bmatrix}$$

$$= \frac{1}{2} I_{zz}^4 \left(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4\right)^2$$

which I_{zz}^n is equal to the moment of inertia of n^{th} link. The spoon, it means manipulator, is 4^{th} link on the system.

3.4.2.5. Calculating the linear velocity vectors of the center of masses

$$v_{G_0}^0(R_0) = 0$$

$$v_{G_1}^0(R_1) = v_1^0(R_1) + v_{G_1}^1(R_1) + \omega_1^0(R_1) x O_1 G_1(R_1) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} x \begin{bmatrix} a_{G_1} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ a_{G_1} \dot{\theta}_1 \\ 0 \end{bmatrix}$$

which is the meaning of a_{G_1} is the length from G_1 to O_1 on the second link that its length is a_2.

$$v_2^0(R_1) = v_1^0(R_1) + v_2^1(R_1) + \omega_1^0(R_1) x O_1O_2(R_1) = \begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\\dot{\theta}_1\end{bmatrix} x \begin{bmatrix}a_2\\0\\0\end{bmatrix} = \begin{bmatrix}0\\a_2\dot{\theta}_1\\0\end{bmatrix}$$

which is the definition of a_2 is the length of the second link.

$$v_2^0(R_2) = D_1^2\, v_2^0(R_1) = \begin{bmatrix} c_2 & s_2 & 0\\ -s_2 & c_2 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\a_2\dot{\theta}_1\\0\end{bmatrix} = \begin{bmatrix}a_2\dot{\theta}_1 s_2\\a_2\dot{\theta}_1 c_2\\0\end{bmatrix}$$

$$v_{G_2}^0(R_2) = v_2^0(R_2) + v_{G_2}^2(R_2) + \omega_2^0(R_2) x O_2G_2(R_2) = \begin{bmatrix}a_2\dot{\theta}_1 s_2\\a_2\dot{\theta}_1 c_2\\0\end{bmatrix} + \begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\\dot{\theta}_1+\dot{\theta}_2\end{bmatrix} x \begin{bmatrix}a_{G_2}\\0\\0\end{bmatrix} =$$

$$= \begin{bmatrix}a_2\dot{\theta}_1 s_2\\a_{G_2}(\dot{\theta}_1+\dot{\theta}_2) + a_2\dot{\theta}_1 c_2\\0\end{bmatrix}$$

which is the meaning of a_{G_2} is the length from G_2 to O_2 on the third link that its length is a_3.

$$v_3^0(R_2) = v_2^0(R_2) + v_3^2(R_2) + \omega_2^0(R_2) x O_2O_3(R_2) = \begin{bmatrix}a_2\dot{\theta}_1 s_2\\a_2\dot{\theta}_1 c_2\\0\end{bmatrix} + \begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\\dot{\theta}_1+\dot{\theta}_2\end{bmatrix} x \begin{bmatrix}a_3\\0\\0\end{bmatrix} =$$

$$= \begin{bmatrix}a_2\dot{\theta}_1 s_2\\a_3(\dot{\theta}_1+\dot{\theta}_2) + a_2\dot{\theta}_1 c_2\\0\end{bmatrix}$$

which is the definition of a_3 is the length of the third link.

$$v_3^0(R_3) = D_2^3\ v_3^0(R_2) = \begin{bmatrix} c_3 & s_3 & 0 \\ -s_3 & c_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_2\dot{\theta}_1 s_2 \\ a_3\left(\dot{\theta}_1 + \dot{\theta}_2\right) + a_2\dot{\theta}_1 c_2 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} (a_2 s_{23} + a_3 s_3)\dot{\theta}_1 + a_3 s_3 \dot{\theta}_2 \\ (a_2 c_{23} + a_3 c_3)\dot{\theta}_1 + a_3 c_3 \dot{\theta}_2 \\ 0 \end{bmatrix}$$

$$v_{G_3}^0(R_3) = v_3^0(R_3) + v_{G_3}^3(R_3) + \omega_3^0(R_3) x O_3 G_3(R_3) =$$

$$= \begin{bmatrix} (a_2 s_{23} + a_3 s_3)\dot{\theta}_1 + a_3 s_3 \dot{\theta}_2 \\ (a_2 c_{23} + a_3 c_3)\dot{\theta}_1 + a_3 c_3 \dot{\theta}_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix} x \begin{bmatrix} a_{G_3} \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} (a_3 s_3 + a_2 s_{23})\dot{\theta}_1 + a_3 s_3 \dot{\theta}_2 \\ \left(a_{G_3} + a_3 c_3 + a_2 c_{23}\right)\dot{\theta}_1 + \left(a_{G_3} + a_3 c_3\right)\dot{\theta}_2 + a_{G_3}\dot{\theta}_3 \\ 0 \end{bmatrix}$$

$$v_4^0(R_3) = v_3^0(R_3) + v_4^3(R_3) + \omega_3^0(R_3) x O_3 O_4(R_3) =$$

$$= \begin{bmatrix} (a_3 s_3 + a_2 s_{23})\dot{\theta}_1 + a_3 s_3 \dot{\theta}_2 \\ \left(a_{G_3} + a_3 c_3 + a_2 c_{23}\right)\dot{\theta}_1 + \left(a_{G_3} + a_3 c_3\right)\dot{\theta}_2 + a_{G_3}\dot{\theta}_3 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix} x \begin{bmatrix} a_4 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} (a_3 s_3 + a_2 s_{23})\dot{\theta}_1 + a_3 s_3 \dot{\theta}_2 \\ \left(a_{G_3} + a_4 + a_3 c_3 + a_2 c_{23}\right)\dot{\theta}_1 + \left(a_{G_3} + a_4 + a_3 c_3\right)\dot{\theta}_2 + \left(a_{G_3} + a_4\right)\dot{\theta}_3 \\ 0 \end{bmatrix}$$

$$v_4^0(R_4) = D_3^4\ v_4^0(R_3) =$$

$$= \begin{bmatrix} c_4 & s_4 & 0 \\ -s_4 & c_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (a_3 s_3 + a_2 s_{23})\dot{\theta}_1 + a_3 s_3 \dot{\theta}_2 \\ \left(a_{G_3} + a_4 + a_3 c_3 + a_2 c_{23}\right)\dot{\theta}_1 + \left(a_{G_3} + a_4 + a_3 c_3\right)\dot{\theta}_2 + \left(a_{G_3} + a_4\right)\dot{\theta}_3 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \left(\left(a_{G_3} + a_4 + a_3 c_3 + a_2 c_{23}\right)s_4 + (a_3 s_3 + a_2 s_{23})c_4\right)\dot{\theta}_1 + \left(s_4\left(a_{G_3} + a_4 + a_3 c_3\right) + a_3 s_3 c_4\right)\dot{\theta}_2 + \left(a_{G_3} + a_4\right)s_4\dot{\theta}_3 \\ \left(\left(a_{G_3} + a_4 + a_3 c_3 + a_2 c_{23}\right)c_4 - (a_3 s_3 + a_2 s_{23})s_4\right)\dot{\theta}_1 + \left(c_4\left(a_{G_3} + a_4 + a_3 c_3\right) - a_3 s_3 s_4\right)\dot{\theta}_2 + \left(a_{G_3} + a_4\right)c_4\dot{\theta}_3 \\ 0 \end{bmatrix}$$

$$v_{G_4}^0(R_4) = v_4^0(R_4) + v_{G_4}^4(R_4) + \omega_4^0(R_4) x O_4 G_4(R_4) =$$

$$= \begin{bmatrix} \left((a_{G_3} + a_4 + a_3c_3 + a_2c_{23})s_4 + (a_3s_3 + a_2s_{23})c_4\right)\dot{\theta}_1 + \left(s_4(a_{G_3} + a_4 + a_3c_3) + a_3s_3c_4\right)\dot{\theta}_2 + (a_{G_3} + a_4)s_4\dot{\theta}_3 \\ \left((a_{G_3} + a_4 + a_3c_3 + a_2c_{23})c_4 - (a_3s_3 + a_2s_{23})s_4\right)\dot{\theta}_1 + \left(c_4(a_{G_3} + a_4 + a_3c_3) - a_3s_3s_4\right)\dot{\theta}_2 + (a_{G_3} + a_4)c_4\dot{\theta}_3 \\ 0 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4 \end{bmatrix} x \begin{bmatrix} a_4 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \left((a_{G_3} + a_4 + a_3c_3 + a_2c_{23})s_4 + (a_3s_3 + a_2s_{23})c_4\right)\dot{\theta}_1 + \left(s_4(a_{G_3} + a_4 + a_3c_3) + a_3s_3c_4\right)\dot{\theta}_2 + (a_{G_3} + a_4)s_4\dot{\theta}_3 \\ \left((a_{G_3} + a_4 + a_3c_3 + a_2c_{23})c_4 - (a_3s_3 + a_2s_{23})s_4 + a_4\right)\dot{\theta}_1 + \left(c_4(a_{G_3} + a_4 + a_3c_3) - a_3s_3s_4 + a_4\right)\dot{\theta}_2 + \left(c_4(a_{G_3} + a_4) + a_4\right)\dot{\theta}_3 + a_4\dot{\theta}_4 \\ 0 \end{bmatrix}$$

3.4.2.6. Calculating the translational kinetic energies of the system

$$K_{E_1}^{(T)} = \frac{1}{2}m_1\left[V_{G_1}^0(R_1)\right]^T\left[V_{G_1}^0(R_1)\right] = \frac{1}{2}m_1\begin{bmatrix}0 & a_{G_1}\dot{\theta}_1 & 0\end{bmatrix}\begin{bmatrix}0 \\ a_{G_1}\dot{\theta}_1 \\ 0\end{bmatrix} = \frac{1}{2}m_1\left(a_{G_1}\dot{\theta}_1\right)^2$$

$$K_{E_2}^{(T)} = \frac{1}{2}m_2\left[V_{G_2}^0(R_2)\right]^T\left[V_{G_2}^0(R_2)\right] = \frac{1}{2}m_2\begin{bmatrix}a_2s_2\dot{\theta}_1 & (a_{G_2} + a_2c_2)\dot{\theta}_1 + a_{G_2}\dot{\theta}_2 & 0\end{bmatrix}\begin{bmatrix}a_2s_2\dot{\theta}_1 \\ (a_{G_2} + a_2c_2)\dot{\theta}_1 + a_{G_2}\dot{\theta}_2 \\ 0\end{bmatrix}$$

$$\Rightarrow \left(\frac{1}{2}m_2\left(a_2^2 + a_{G_2}^2\right) + m_2a_{G_2}a_2c_2\right)\left(\dot{\theta}_1\right)^2 + \frac{1}{2}m_2a_{G_2}^2\left(\dot{\theta}_2\right)^2 + m_2\left(a_{G_2} + a_2c_2\right)a_{G_2}\dot{\theta}_1\dot{\theta}_2$$

$$K_{E_3}^{(T)} = \frac{1}{2}m_3\left[V_{G_3}^0(R_3)\right]^T\left[V_{G_3}^0(R_3)\right] \Rightarrow$$

$$= \frac{1}{2}m_3\begin{bmatrix}(a_3s_3 + a_2s_{23})\dot{\theta}_1 + a_3s_3\dot{\theta}_2 & (a_{G_3} + a_3c_3 + a_2c_{23})\dot{\theta}_1 + (a_{G_3} + a_3c_3)\dot{\theta}_2 + a_{G_3}\dot{\theta}_3 & 0\end{bmatrix}\begin{bmatrix}(a_3s_3 + a_2s_{23})\dot{\theta}_1 + a_3s_3\dot{\theta}_2 \\ (a_{G_3} + a_3c_3 + a_2c_{23})\dot{\theta}_1 + (a_{G_3} + a_3c_3)\dot{\theta}_2 + a_{G_3}\dot{\theta}_3 \\ 0\end{bmatrix}$$

$$= \frac{1}{2}m_3\left(a_2^2 + a_3^2 + 2a_3a_2c_2 + 2a_{G_3}(a_3c_3 + a_2c_{23})\right)\left(\dot{\theta}_1\right)^2 + \frac{1}{2}m_3\left(a_3^2 + a_{G_3}^2 + 2a_{G_3}a_3c_3\right)\left(\dot{\theta}_2\right)^2 + m_3a_{G_3}^2\left(\dot{\theta}_3\right)^2 + \frac{1}{2}m_3\left(a_3^2s_3^2 + a_2a_3s_3s_{23} + 2\left(a_{G_3}^2 + 2a_{G_3}a_3c_3 + a_3^2c_3^2 + a_{G_3}a_2c_{23} + a_2a_3c_3c_{23}\right)\right)\dot{\theta}_1\dot{\theta}_2 + m_3\left(a_{G_3}^2 + a_{G_3}a_3c_3 + a_{G_3}a_2c_{23}\right)\dot{\theta}_1\dot{\theta}_3 + m_3\left(a_{G_3} + a_3c_3\right)\dot{\theta}_2\dot{\theta}_3$$

$K_{E_4}^{(T)} = \frac{1}{2}m_4\left[V_{G_4}^0(R_4)\right]^T\left[V_{G_4}^0(R_4)\right] = \frac{1}{2}m_4(2a_4 + a_{G_3} + a_2c_{23} + a_3c_3 + a_4c_4 + a_{G_3}c_4 + a_2c_{23}c_4 + a_3c_3c_4 - a_2s_{23}s_4 - a_3s_3s_4 - a_2s_{23}c_4 - a_3s_3c_4)(\dot{\theta}_1)^2 + \frac{1}{2}m_4(a_4c_4 + a_{G_3}c_4 + a_3c_3c_4 - a_3s_3s_4 - a_3s_3c_4 + a_4s_4 + a_{G_3}s_4 + a_3c_3s_4 + a_4)(\dot{\theta}_2)^2 + \frac{1}{2}m_4(a_4c_4 + a_{G_3}c_4 + a_4 + a_{G_3} + a_4)(\dot{\theta}_3)^2 + \frac{1}{2}m_4a_4(\dot{\theta}_4)^2 + 2\dot{\theta}_1\dot{\theta}_2 + 2\dot{\theta}_1\dot{\theta}_3 + 2\dot{\theta}_1\dot{\theta}_4 + 2\dot{\theta}_2\dot{\theta}_3 + 2\dot{\theta}_2\dot{\theta}_4 + 2\dot{\theta}_3\dot{\theta}_4$

From the calculated kinetic energies, the total kinetic will be as the following;

$$K_E = K_E^{(R)} + K_E^{(T)}$$

$\Rightarrow \frac{1}{2}\left\{I_{zz}^1 + I_{zz}^2 + I_{zz}^3 + I_{zz}^4 + m_1a_{G_1}^2 + m_2\left(a_2^2 + a_{G_2}^2 + 2a_{G_2}a_2c_2\right) + m_3\left(a_2^2 + a_3^2 + 2a_3a_2c_2 + 2a_{G_3}(a_3c_3 + a_2c_{23})\right) + m_4\left(a_4 + a_{G_3} + a_2c_{23} + a_3c_3 + a_4c_4 + a_{G_3}c_4 + a_2c_{23}c_4 + a_3c_3c_4 - a_2s_{23}s_4 - a_3s_3s_4 - a_2s_{23}c_4 - a_3s_3c_4 + a_4\right)\right\}(\dot{\theta}_1)^2 + \frac{1}{2}\left\{I_{zz}^2 + I_{zz}^3 + I_{zz}^4 + m_2a_{G_2}^2 + m_3\left(a_3^2 + a_{G_3}^2 + 2a_{G_3}a_3c_3\right) + m_4\left(a_4c_4 + a_{G_3}c_4 + a_3c_3c_4 - a_3s_3s_4 - a_3s_3c_4 + a_4s_4 + a_{G_3}s_4 + a_3c_3s_4 + a_4\right)\right\}(\dot{\theta}_2)^2 + \frac{1}{2}\left\{I_{zz}^3 + I_{zz}^4 + 2m_3a_{G_3}^2 + m_4\left(a_4c_4 + a_{G_3}c_4 + a_4 + a_{G_3} + a_4\right)\right\}(\dot{\theta}_3)^2 + \frac{1}{2}\{I_{zz}^4 + m_4a_4\}(\dot{\theta}_4)^2 + \left\{\frac{1}{2}I_{zz}^2 + \frac{1}{2}I_{zz}^3 + \frac{1}{2}I_{zz}^4 + m_2\left(a_{G_2}^2 + a_2c_2a_{G_2}\right) + \frac{1}{2}m_3\left(a_3^2s_3^2 + a_2a_3s_3s_{23} + 2\left(a_{G_3}^2 + 2a_{G_3}a_3c_3 + a_3^2c_3^2 + a_{G_3}a_2c_{23} + a_2a_3c_3c_{23}\right)\right) + 2\right\}\dot{\theta}_1\dot{\theta}_2 + \left\{\frac{1}{2}I_{zz}^3 + \frac{1}{2}I_{zz}^4 + m_3\left(a_{G_3}^2 + a_{G_3}a_3c_3 + a_{G_3}a_2c_{23}\right) + 2\right\}\dot{\theta}_1\dot{\theta}_3 + \left\{\frac{1}{2}I_{zz}^3 + \frac{1}{2}I_{zz}^4 + m_3\left(a_{G_3} + a_3c_3\right) + 2\right\}\dot{\theta}_2\dot{\theta}_3 + \{I_{zz}^4 + 2\}\dot{\theta}_1\dot{\theta}_4 + \{I_{zz}^4 + 2\}\dot{\theta}_2\dot{\theta}_4 + \{I_{zz}^4 + 2\}\dot{\theta}_3\dot{\theta}_4$

Now, some parameters must be defined to take derivatives easily.

$k_1 = I_{zz}^1 + I_{zz}^2 + I_{zz}^3 + I_{zz}^4 + m_1a_{G_1}^2 + m_2a_2^2 + m_2a_{G_2}^2$

$k_2 = 2m_2a_{G_2}a_2$

$k_3 = m_3(a_2^2 + a_3^2)$

$k_4 = 2m_3a_3a_2$

$k_5 = 2m_3a_{G_3}a_3$

$k_6 = 2m_3a_{G_3}a_2$

$$k_7 = m_4\left(2a_4 + a_{G_3}\right)$$

$$k_8 = m_4 a_2$$

$$k_9 = m_4 a_3$$

$$k_{10} = m_4 a_4$$

$$k_{11} = m_4 a_{G_3}$$

$$k_{12} = I_{zz}^2 + I_{zz}^3 + I_{zz}^4 + m_2 a_{G_2}^2 + m_3\left(a_3^2 + a_{G_3}^2\right)$$

$$k_{13} = m_4 a_4$$

$$k_{14} = I_{zz}^3 + I_{zz}^4 + 2m_3 a_{G_3}^2 + m_4\left(2a_4 + a_{G_3}\right)$$

$$k_{15} = I_{zz}^4 + m_4 a_4$$

$$k_{16} = \frac{1}{2} I_{zz}^2 + \frac{1}{2} I_{zz}^3 + \frac{1}{2} I_{zz}^4 + m_2 a_{G_2}^2$$

$$k_{17} = m_3 a_{G_3}^2$$

$$k_{18} = m_3 a_3^2$$

$$k_{19} = \frac{1}{2} I_{zz}^3 + \frac{1}{2} I_{zz}^4$$

$$k_{20} = m_3 a_{G_3}$$

$$k_{21} = m_3 a_3$$

$$k_{22} = I_{zz}^4 + 2$$

The total kinetic energy can be defined as the following:

$$K_E = \tfrac{1}{2}(k_1 + k_2 c_2 + k_3 + k_4 c_2 + k_5 c_3 + k_6 c_{23} + k_7 + k_8 c_{23} + k_9 c_3 + k_{10} c_4 + k_{11} c_4 + k_8 c_{23} c_4 + k_9 c_3 c_4 - k_8 s_{23} s_4 - k_9 s_3 s_4 - k_8 s_{23} c_4 - k_9 s_3 c_4)\left(\dot{\theta}_1\right)^2 + \tfrac{1}{2}(k_{12} + k_5 c_3 + k_{13} + k_{13} c_4 + k_{11} c_4 + k_9 c_3 c_4 - k_9 s_3 s_4 - k_9 s_3 c_4 + k_{10} s_4 + k_{11} s_4 + k_9 c_3 s_4)\left(\dot{\theta}_2\right)^2 + \tfrac{1}{2}(k_{14} + k_{10} c_4 + k_{11} c_4)\left(\dot{\theta}_3\right)^2 + \tfrac{1}{2} k_{15}\left(\dot{\theta}_4\right)^2 + \left(k_{16} + \tfrac{k_2}{2} c_2 + \tfrac{k_{18}}{2} s_3^2 + \tfrac{k_4}{2} s_3 s_{23} + k_{17} + k_5 + k_{18} c_3^2 + \tfrac{k_6}{2} c_{23} + \tfrac{k_4}{2} c_3 c_{23} + 2\right)\dot{\theta}_1 \dot{\theta}_2 + \left(k_{19} + k_{17} + \tfrac{k_5}{2} c_3 + \tfrac{k_6}{2} c_{23} + 2\right)\dot{\theta}_1 \dot{\theta}_3 + (k_{19} + k_{20} + k_{21} c_3 + 2)\dot{\theta}_2 \dot{\theta}_3 + k_{22}\dot{\theta}_1 \dot{\theta}_4 + k_{22}\dot{\theta}_2 \dot{\theta}_4 + k_{22}\dot{\theta}_3 \dot{\theta}_4$$

3.4.2.7. Calculating the potential energies of the system

$$P_1 = m_1.g.y_1 = m_1.g.d_1$$

$$P_2 = m_2.g.y_2 = m_2.g.(d_1 + a_2 sin\theta_2)$$

$$P_3 = m_3.g.y_3 = m_3.g.(d_1 + a_2 sin\theta_2 + a_3 sin\theta_3)$$

$$P_4 = m_4.g.y_4 = m_4.g.(d_1 + a_2 sin\theta_2 + a_3 sin\theta_3 + a_4 sin\theta_4)$$

$$P_E = P_1 + P_2 + P_3 + P_4$$

$$\Rightarrow (m_1 + m_2 + m_3 + m_4)g.d_1 + (m_2 + m_3 + m_4)ga_2 sin\theta_2 + (m_3 + m_4)ga_3 sin\theta_3 + m_4.ga_4 sin\theta_4$$

which is P_E is equal to the total potential energy of the system.

3.4.2.8. Calculating of Lagrangian Function

$$\mathcal{L}(\theta, \dot{\theta}) = \sum_{i=1}^{4} (K_i - P_i) \qquad and \qquad \tau_i = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i}\right) - \frac{\partial \mathcal{L}}{\partial \theta_i} \qquad i = 1,2,3,4$$

In order to determine the Euler-Lagrange equations in a specific situation, one has to form the Lagrangian of the system which is the difference between the kinetic energy and potential energy.

$$\mathcal{L}(\theta, \dot{\theta}) = \sum_{i=1}^{4} (K_i - P_i) = (K_1 + K_2 + K_3 + K_4) - (P_1 - P_2 - P_3 - P_4) = K_E - P_E$$

$$\Rightarrow \frac{1}{2}(k_1 + k_2 c_2 + k_3 + k_4 c_2 + k_5 c_3 + k_6 c_{23} + k_7 + k_8 c_{23} + k_9 c_3 + k_{10} c_4 + k_{11} c_4 + k_8 c_{23} c_4 + k_9 c_3 c_4 - k_8 s_{23} s_4 - k_9 s_3 s_4 - k_8 s_{23} c_4 - k_9 s_3 c_4)(\dot{\theta}_1)^2 + \frac{1}{2}(k_{12} + k_5 c_3 + k_{13} + k_{13} c_4 + k_{11} c_4 + k_9 c_3 c_4 - k_9 s_3 s_4 - k_9 s_3 c_4 + k_{10} s_4 + k_{11} s_4 + k_9 c_3 s_4)(\dot{\theta}_2)^2 + \frac{1}{2}(k_{14} + k_{10} c_4 + k_{11} c_4)(\dot{\theta}_3)^2 + \frac{1}{2} k_{15}(\dot{\theta}_4)^2 + \left(k_{16} + \frac{k_2}{2} c_2 + \frac{k_4}{2} s_3 s_{23} + k_{17} + k_5 + k_{18} + \frac{k_6}{2} c_{23} + \frac{k_4}{2} c_3 c_{23} + 2\right)\dot{\theta}_1 \dot{\theta}_2 + \left(k_{19} + k_{17} + \frac{k_5}{2} c_3 + \frac{k_6}{2} c_{23} + 2\right)\dot{\theta}_1 \dot{\theta}_3 + (k_{19} + k_{20} + k_{21} c_3 + 2)\dot{\theta}_2 \dot{\theta}_3 + k_{22} \dot{\theta}_1 \dot{\theta}_4 + k_{22} \dot{\theta}_2 \dot{\theta}_4 + k_{22} \dot{\theta}_3 \dot{\theta}_4 - (m_1 + m_2 + m_3 + m_4)g.d_1 - (m_2 + m_3 + m_4)ga_2 sin\theta_2 - (m_3 + m_4)ga_3 sin\theta_3 - m_4.ga_4 sin\theta_4$$

3.4.2.9. Calculating the Partial Derivatives of the Lagrangian Function

$$\frac{\partial \mathcal{L}}{\partial \theta_1} = 0$$

$$\frac{\partial \mathcal{L}}{\partial \theta_2} = -\frac{1}{2}(k_4 + k_6 s_{23} + k_8 s_{23} + k_8 s_{23} c_4 + k_8 c_{23} s_4 + k_8 c_{23} c_4)\left(\dot{\theta}_1\right)^2 - \left(\frac{k_2}{2} s_2 - \frac{k_4}{2} s_3 c_{23} + \frac{k_6}{2} s_{23} + \frac{k_4}{2} c_3 s_{23}\right)\dot{\theta}_1\dot{\theta}_2 - \frac{k_6}{2} s_{23}\dot{\theta}_1\dot{\theta}_3 + (m_2 + m_3 + m_4) g a_2 c_2$$

$$\frac{\partial \mathcal{L}}{\partial \theta_3} = -\frac{1}{2}(k_5 s_3 + k_6 s_{23} + k_8 s_{23} + k_9 s_3 + k_8 s_{23} c_4 + k_9 s_3 c_4 + k_8 c_{23} s_4 + k_9 c_3 s_4 + k_8 c_{23} c_4 + k_9 c_3 c_4)\left(\dot{\theta}_1\right)^2 - \frac{1}{2}(k_5 s_3 + k_9 s_3 c_4 + k_9 c_3 s_4 + k_9 c_3 c_4)\left(\dot{\theta}_2\right)^2 + \left(\frac{k_4}{2} s_3 c_{23} - \frac{k_6}{2} s_{23} + \frac{k_4}{2} s_2\right)\dot{\theta}_1\dot{\theta}_2 - \left(\frac{k_5}{2} s_3 + \frac{k_6}{2} s_{23}\right)\dot{\theta}_1\dot{\theta}_3 - k_{21} s_3 \dot{\theta}_2\dot{\theta}_3 - (m_3 + m_4) g a_3 c_3$$

$$\frac{\partial \mathcal{L}}{\partial \theta_4} = -\frac{1}{2}(k_{10} s_4 + k_{11} s_4 + k_8 c_{23} s_4 - k_9 c_3 s_4 + k_8 s_{23} c_4 - k_9 s_3 c_4 + k_8 s_{23} s_4 + k_9 s_3 s_4)\left(\dot{\theta}_1\right)^2 + \frac{1}{2}(-k_{13} c_4 - k_{11} c_4 - k_9 c_3 c_4 - k_9 s_3 c_4 + k_9 s_3 s_4 + k_{10} c_4 + k_{11} c_4 + k_9 c_3 c_4)\left(\dot{\theta}_2\right)^2 - \frac{1}{2}(k_{10} s_4 + k_{11} s_4)\left(\dot{\theta}_3\right)^2 - m_4 . g a_4 c_4$$

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1} = (k_1 + k_2 c_2 + k_3 + k_4 c_2 + k_5 c_3 + k_6 c_{23} + k_7 + k_8 c_{23} + k_9 c_3 + k_{10} c_4 + k_{11} c_4 + k_8 c_{23} c_4 + k_9 c_3 c_4 - k_8 s_{23} s_4 - k_9 s_3 s_4 - k_8 s_{23} c_4 - k_9 s_3 c_4)\dot{\theta}_1 + \left(k_{16} + \frac{k_2}{2} c_2 + k_{18} + \frac{k_4}{2} s_3 s_{23} + k_{17} + k_5 + \frac{k_6}{2} c_{23} + \frac{k_4}{2} c_3 c_{23} + 2\right)\dot{\theta}_2 + \left(k_{19} + k_{17} + \frac{k_5}{2} c_3 + \frac{k_6}{2} c_{23} + 2\right)\dot{\theta}_3$$

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}_2} = \left(k_{16} + \frac{k_2}{2} c_2 + k_{18} + \frac{k_4}{2} s_3 s_{23} + k_{17} + k_5 + \frac{k_6}{2} c_{23} + \frac{k_4}{2} c_3 c_{23} + 2\right)\dot{\theta}_1 + (k_{12} + k_5 c_3 + k_{13} + k_{13} c_4 + k_{11} c_4 + k_9 c_3 c_4 - k_9 s_3 s_4 - k_9 s_3 c_4 + k_{10} s_4 + k_{11} s_4 + k_9 c_3 s_4)\dot{\theta}_2 + (k_{19} + k_{20} + k_{21} c_3 + 2)\dot{\theta}_3 + k_{22}\dot{\theta}_4$$

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}_3} = \left(k_{19} + k_{17} + \frac{k_5}{2}c_3 + \frac{k_6}{2}c_{23} + 2\right)\dot{\theta}_1 + (k_{19} + k_{20} + k_{21}c_3 + 2)\dot{\theta}_2 + (k_{14} + k_{10}c_4 + k_{11}c_4)\dot{\theta}_3 + k_{22}\dot{\theta}_4$$

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}_4} = k_{22}\dot{\theta}_1 + k_{22}\dot{\theta}_2 + k_{22}\dot{\theta}_3 + k_{15}\dot{\theta}_4$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1}\right) = (k_1 + k_2c_2 + k_3 + k_4c_2 + k_5c_3 + k_6c_{23} + k_7 + k_8c_{23} + k_9c_3 + k_{10}c_4 + k_{11}c_4 + k_8c_{23}c_4 + k_9c_3c_4 - k_8s_{23}s_4 - k_9s_3s_4 - k_8s_{23}c_4 - k_9s_3c_4)\ddot{\theta}_1 + \left(k_{16} + \frac{k_2}{2}c_2 + k_{18} + \frac{k_4}{2}s_3s_{23} + k_{17} + k_5 + \frac{k_6}{2}c_{23} + \frac{k_4}{2}c_3c_{23} + 2\right)\ddot{\theta}_2 + \left(k_{19} + k_{17} + \frac{k_5}{2}c_3 + \frac{k_6}{2}c_{23} + 2\right)\ddot{\theta}_3 + (-k_2s_2 - k_4s_2 - k_6s_{23} - k_8s_{23} - k_8s_{23}c_4 - k_8s_{23}s_4 - k_8c_{23}c_4)\left(\dot{\theta}_1\dot{\theta}_2\right) + (-k_5s_3 - k_6s_{23} - k_8s_{23} - k_8s_{23}c_4 - k_9s_3c_4 - k_8c_{23}s_4 - k_9c_3s_4 - k_8c_{23}c_4)\left(\dot{\theta}_1\dot{\theta}_3\right) + (-k_{10}s_4 - k_{11}s_4 - k_8c_{23}s_4 - k_9c_3s_4 - k_8s_{23}c_4 - k_9s_3c_4 + k_8s_{23}s_4 + k_9s_3s_4)\left(\dot{\theta}_1\dot{\theta}_4\right) + \left(-\frac{k_2}{2}s_2 + \frac{k_4}{2}s_3c_{23} - \frac{k_6}{2}s_{23} - \frac{k_4}{2}c_3s_{23}\right)\dot{\theta}_2^2 + \left(\frac{k_4}{2}s_3c_{23} - k_6s_{23} - \frac{k_4}{2}s_3c_{23} - \frac{k_4}{2}c_3s_{23}\right)\left(\dot{\theta}_2\dot{\theta}_3\right) + \left(-\frac{k_5}{2}s_3 - \frac{k_6}{2}s_{23}\right)\dot{\theta}_3^2$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_2}\right) = \left(k_{16} + \frac{k_2}{2}c_2 + k_{18} + \frac{k_4}{2}s_3s_{23} + k_{17} + k_5 + \frac{k_6}{2}c_{23} + \frac{k_4}{2}c_3c_{23} + 2\right)\ddot{\theta}_1 + (k_{12} + k_5c_3 + k_{13} + k_{13}c_4 + k_{11}c_4 + k_9c_3c_4 - k_9s_3s_4 - k_9s_3c_4 + k_{10}s_4 + k_{11}s_4 + k_9c_3s_4)\ddot{\theta}_2 + (k_{19} + k_{20} + k_{21}c_3 + 2)\ddot{\theta}_3 + k_{22}\ddot{\theta}_4 + \left(-\frac{k_2}{2}s_2 + \frac{k_4}{2}s_3c_{23} - \frac{k_6}{2}s_{23} - \frac{k_4}{2}c_3s_{23}\right)\left(\dot{\theta}_1\dot{\theta}_2\right) + \left(\frac{k_4}{2}c_3s_{23} - \frac{k_6}{2}s_{23}\right)\left(\dot{\theta}_1\dot{\theta}_3\right) + (-k_5s_3 - k_9s_3c_4 - k_9c_3s_4 - k_9s_3s_4)\left(\dot{\theta}_2\dot{\theta}_3\right) + (-k_{13}s_4 - k_{11}s_4 - k_9c_3s_4 - k_9s_3c_4 + k_9c_3c_4 + k_{10}c_4 + k_{11}c_4)\left(\dot{\theta}_2\dot{\theta}_4\right) - k_{21}s_3\dot{\theta}_3^2$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_3}\right) = \left(k_{19} + k_{17} + \frac{k_5}{2}c_3 + \frac{k_6}{2}c_{23} + 2\right)\ddot{\theta}_1 + (k_{19} + k_{20} + k_{21}c_3 + 2)\ddot{\theta}_2 + (k_{14} + k_{10}c_4 + k_{11}c_4)\ddot{\theta}_3 + k_{22}\ddot{\theta}_4 - \frac{k_6}{2}s_{23}\left(\dot{\theta}_1\dot{\theta}_2\right) + \left(-\frac{k_5}{2}s_3 - \frac{k_6}{2}s_{23}\right)\left(\dot{\theta}_1\dot{\theta}_3\right) - k_{21}s_3\left(\dot{\theta}_2\dot{\theta}_3\right) + (-k_{10}s_4 + k_{11}s_4)\left(\dot{\theta}_3\dot{\theta}_4\right)$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_4}\right) = k_{22}\ddot{\theta}_1 + k_{22}\ddot{\theta}_2 + k_{22}\ddot{\theta}_3 + k_{15}\ddot{\theta}_4$$

3.4.2.10. Calculating the Torque Values of all joints

$$\tau_i = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i}\right) - \frac{\partial \mathcal{L}}{\partial \theta_i} \qquad i = 1,2,3,4$$

$$\tau_1 = (k_1 + k_2c_2 + k_3 + k_4c_2 + k_5c_3 + k_6c_{23} + k_7 + k_8c_{23} + k_9c_3 + k_{10}c_4 + k_{11}c_4 + k_8c_{23}c_4 + k_9c_3c_4 - k_8s_{23}s_4 - k_9s_3s_4 - k_8s_{23}c_4 - k_9s_3c_4)\ddot{\theta}_1 + \left(k_{16} + \frac{k_2}{2}c_2 + k_{18} + \frac{k_4}{2}s_3s_{23} + k_{17} + k_5 + \frac{k_6}{2}c_{23} + \frac{k_4}{2}c_3c_{23} + 2\right)\ddot{\theta}_2 + \left(k_{19} + k_{17} + \frac{k_5}{2}c_3 + \frac{k_6}{2}c_{23} + 2\right)\ddot{\theta}_3 + (-k_2s_2 - k_4s_2 - k_6s_{23} - k_8s_{23} - k_8s_{23}c_4 - k_8s_{23}s_4 - k_8c_{23}c_4)\left(\dot{\theta}_1\dot{\theta}_2\right) + (-k_5s_3 - k_6s_{23} - k_8s_{23} - k_8s_{23}c_4 - k_9s_3c_4 - k_8c_{23}s_4 - k_9c_3s_4 - k_8c_{23}c_4)\left(\dot{\theta}_1\dot{\theta}_3\right) + (-k_{10}s_4 - k_{11}s_4 - k_8c_{23}s_4 - k_9c_3s_4 - k_8s_{23}c_4 - k_9s_3c_4 + k_8s_{23}s_4 + k_9s_3s_4)\left(\dot{\theta}_1\dot{\theta}_4\right) + \left(-\frac{k_2}{2}s_2 + \frac{k_4}{2}s_3c_{23} - \frac{k_6}{2}s_{23} - \frac{k_4}{2}c_3s_{23}\right)\dot{\theta}_2^2 + \left(\frac{k_4}{2}s_3c_{23} - k_6s_{23} - \frac{k_4}{2}s_3c_{23} - \frac{k_4}{2}c_3s_{23}\right)\left(\dot{\theta}_2\dot{\theta}_3\right) + \left(-\frac{k_5}{2}s_3 - \frac{k_6}{2}s_{23}\right)\dot{\theta}_3^2$$

$$\tau_2 = \left(k_{16} + \frac{k_2}{2}c_2 + k_{18} + \frac{k_4}{2}s_3s_{23} + k_{17} + k_5 + \frac{k_6}{2}c_{23} + \frac{k_4}{2}c_3c_{23} + 2\right)\ddot{\theta}_1 + (k_{12} + k_5c_3 + k_{13} + k_{13}c_4 + k_{11}c_4 + k_9c_3c_4 - k_9s_3s_4 - k_9s_3c_4 + k_{10}s_4 + k_{11}s_4 + k_9c_3s_4)\ddot{\theta}_2 + (k_{19} + k_{20} + k_{21}c_3 + 2)\ddot{\theta}_3 + k_{22}\ddot{\theta}_4 + \left(-\frac{k_2}{2}s_2 + \frac{k_4}{2}s_3c_{23} - \frac{k_6}{2}s_{23} - \frac{k_4}{2}c_3s_{23}\right)\left(\dot{\theta}_1\dot{\theta}_2\right) + \left(\frac{k_4}{2}c_3s_{23} - \frac{k_6}{2}s_{23}\right)\left(\dot{\theta}_1\dot{\theta}_3\right) + (-k_5s_3 - k_9s_3c_4 - k_9c_3s_4 - k_9s_3s_4)\left(\dot{\theta}_2\dot{\theta}_3\right) + (-k_{13}s_4 - k_{11}s_4 - k_9c_3s_4 - k_9s_3c_4 + k_9c_3c_4 + k_{10}c_4 + k_{11}c_4)\left(\dot{\theta}_2\dot{\theta}_4\right) - k_{21}s_3\dot{\theta}_3^2 - \frac{1}{2}(k_4 + k_6s_{23} + k_8s_{23} + k_8s_{23}c_4 + k_8c_{23}s_4 + k_8c_{23}c_4)\left(\dot{\theta}_1\right)^2 - \left(\frac{k_2}{2}s_2 - \frac{k_4}{2}s_3c_{23} + \frac{k_6}{2}s_{23} + \frac{k_4}{2}c_3s_{23}\right)\dot{\theta}_1\dot{\theta}_2 - \frac{k_6}{2}s_{23}\dot{\theta}_1\dot{\theta}_3 + (m_2 + m_3 + m_4)ga_2c_2$$

$$\tau_3 = \left(k_{19} + k_{17} + \frac{k_5}{2}c_3 + \frac{k_6}{2}c_{23} + 2\right)\ddot{\theta}_1 + (k_{19} + k_{20} + k_{21}c_3 + 2)\ddot{\theta}_2 + (k_{14} + k_{10}c_4 + k_{11}c_4)\ddot{\theta}_3 + k_{22}\ddot{\theta}_4 - \frac{k_6}{2}s_{23}(\dot{\theta}_1\dot{\theta}_2) + \left(-\frac{k_5}{2}s_3 - \frac{k_6}{2}s_{23}\right)(\dot{\theta}_1\dot{\theta}_3) - k_{21}s_3(\dot{\theta}_2\dot{\theta}_3) + (-k_{10}s_4 + k_{11}s_4)(\dot{\theta}_3\dot{\theta}_4) - \frac{1}{2}(k_5s_3 + k_6s_{23} + k_8s_{23} + k_9s_3 + k_8s_{23}c_4 + k_9s_3c_4 + k_8c_{23}s_4 + k_9c_3s_4 + k_8c_{23}c_4 + k_9c_3c_4)(\dot{\theta}_1)^2 - \frac{1}{2}(k_5s_3 + k_9s_3c_4 + k_9c_3s_4 + k_9c_3c_4)(\dot{\theta}_2)^2 + \left(\frac{k_4}{2}s_3c_{23} - \frac{k_6}{2}s_{23} + \frac{k_4}{2}s_2\right)\dot{\theta}_1\dot{\theta}_2 - \left(\frac{k_5}{2}s_3 + \frac{k_6}{2}s_{23}\right)\dot{\theta}_1\dot{\theta}_3 - k_{21}s_3\dot{\theta}_2\dot{\theta}_3 + (m_3 + m_4)ga_3c_3$$

$$\tau_4 = k_{22}\ddot{\theta}_1 + k_{22}\ddot{\theta}_2 + k_{22}\ddot{\theta}_3 + k_{15}\ddot{\theta}_4 - \frac{1}{2}(k_{10}s_4 + k_{11}s_4 + k_8c_{23}s_4 - k_9c_3s_4 + k_8s_{23}c_4 - k_9s_3c_4 + k_8s_{23}s_4 + k_9s_3s_4)(\dot{\theta}_1)^2 + \frac{1}{2}(-k_{13}c_4 - k_{11}c_4 - k_9c_3c_4 - k_9s_3c_4 + k_9s_3s_4 + k_{10}c_4 + k_{11}c_4 + k_9c_3c_4)(\dot{\theta}_2)^2 - \frac{1}{2}(k_{10}s_4 + k_{11}s_4)(\dot{\theta}_3)^2 + m_4.ga_4c_4$$

3.4.2.11. Finding Mass Matrix, C Vector and Gravity Vector

$$\tau = M(\theta)\ddot{\theta} + c(\theta, \dot{\theta}) + g(\theta)$$

- $M(\theta) = \begin{bmatrix} n_{11} & n_{12} & n_{13} & n_{14} \\ n_{21} & n_{22} & n_{23} & n_{24} \\ n_{31} & n_{32} & n_{33} & n_{34} \\ n_{41} & n_{42} & n_{43} & n_{44} \end{bmatrix}$

$$n_{11} = k_1 + k_2c_2 + k_3 + k_4c_2 + k_5c_3 + k_6c_{23} + k_7 + k_8c_{23} + k_9c_3 + k_{10}c_4 + k_{11}c_4 + k_8c_{23}c_4 + k_9c_3c_4 - k_8s_{23}s_4 - k_9s_3s_4 - k_8s_{23}c_4 - k_9s_3c_4$$

$$n_{12} = k_{16} + \frac{k_2}{2}c_2 + k_{18} + \frac{k_4}{2}s_3s_{23} + k_{17} + k_5 + \frac{k_6}{2}c_{23} + \frac{k_4}{2}c_3c_{23} + 2$$

$$n_{13} = k_{19} + k_{17} + \frac{k_5}{2}c_3 + \frac{k_6}{2}c_{23} + 2$$

$$n_{14} = 0$$

$$n_{21} = k_{16} + \frac{k_2}{2}c_2 + k_{18} + \frac{k_4}{2}s_3 s_{23} + k_{17} + k_5 + \frac{k_6}{2}c_{23} + \frac{k_4}{2}c_3 c_{23} + 2$$

$$n_{22} = k_{12} + k_5 c_3 + k_{13} + k_{13}c_4 + k_{11}c_4 + k_9 c_3 c_4 - k_9 s_3 s_4 - k_9 s_3 c_4 + k_{10}s_4 + k_{11}s_4 + k_9 c_3 s_4$$

$$n_{23} = k_{19} + k_{20} + k_{21}c_3 + 2$$

$$n_{24} = k_{22}$$

$$n_{31} = k_{19} + k_{17} + \frac{k_5}{2}c_3 + \frac{k_6}{2}c_{23} + 2$$

$$n_{32} = k_{19} + k_{20} + k_{21}c_3 + 2$$

$$n_{33} = k_{14} + k_{10}c_4 + k_{11}c_4$$

$$n_{34} = k_{22}$$

$$n_{41} = k_{22}$$

$$n_{42} = k_{22}$$

$$n_{43} = k_{22}$$

$$n_{44} = k_{22}$$

- $c(\theta, \dot{\theta}) = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}$

$$p_{11} = (-k_2 s_2 - k_4 s_2 - k_6 s_{23} - k_8 s_{23} - k_8 s_{23} c_4 - k_8 s_{23} s_4 - k_8 c_{23} c_4)(\dot{\theta}_2) + (-k_5 s_3 - k_6 s_{23} - k_8 s_{23} - k_8 s_{23} c_4 - k_9 s_3 c_4 - k_8 c_{23} s_4 - k_9 c_3 s_4 - k_8 c_{23} c_4)(\dot{\theta}_3) + (-k_{10} s_4 - k_{11} s_4 - k_8 c_{23} s_4 - k_9 c_3 s_4 - k_8 s_{23} c_4 - k_9 s_3 c_4 + k_8 s_{23} s_4 + k_9 s_3 s_4)(\dot{\theta}_4)$$

$$p_{12} = \left(-\frac{k_2}{2}s_2 + \frac{k_4}{2}s_3 c_{23} - \frac{k_6}{2}s_{23} - \frac{k_4}{2}c_3 s_{23}\right)\dot{\theta}_2 + \left(\frac{k_4}{2}s_3 c_{23} - k_6 s_{23} - \frac{k_4}{2}s_3 c_{23} - \frac{k_4}{2}c_3 s_{23}\right)(\dot{\theta}_3)$$

$$p_{13} = \left(-\frac{k_5}{2}s_3 - \frac{k_6}{2}s_{23}\right)\dot{\theta}_3$$

$$p_{14} = 0$$

$$p_{21} = -\frac{1}{2}(k_4 + k_6 s_{23} + k_8 s_{23} + k_8 s_{23} c_4 + k_8 c_{23} s_4 + k_8 c_{23} c_4)(\dot{\theta}_1) + (-k_2 s_2 + k_4 s_3 c_{23} - k_6 s_{23} - k_4 c_3 s_{23})(\dot{\theta}_2) + \left(\frac{k_4}{2} c_3 s_{23} - k_6 s_{23}\right)(\dot{\theta}_3)$$

$$p_{22} = (-k_5 s_3 - k_9 s_3 c_4 - k_9 c_3 s_4 - k_9 s_3 s_4)(\dot{\theta}_3) + (-k_{13} s_4 - k_{11} s_4 - k_9 c_3 s_4 - k_9 s_3 c_4 + k_9 c_3 c_4 + k_{10} c_4 + k_{11} c_4)(\dot{\theta}_4)$$

$$p_{23} = -k_{21} s_3 \dot{\theta}_3$$

$$p_{24} = 0$$

$$p_{31} = \left(\frac{k_6}{2} s_{23} + \frac{k_4}{2} s_3 c_{23} - \frac{k_6}{2} s_{23} + \frac{k_4}{2} s_2\right)(\dot{\theta}_2) + (-k_5 s_3 - k_6 s_{23})(\dot{\theta}_3)$$

$$p_{32} = -\frac{1}{2}(k_5 s_3 + k_9 s_3 c_4 + k_9 c_3 s_4 + k_9 c_3 c_4)(\dot{\theta}_2) - k_{21} s_3 \dot{\theta}_3 - k_{21} s_3 \dot{\theta}_3$$

$$p_{33} = (-k_{10} s_4 + k_{11} s_4)(\dot{\theta}_4)$$

$$p_{34} = 0$$

$$p_{41} = -\frac{1}{2}(k_{10} s_4 + k_{11} s_4 + k_8 c_{23} s_4 - k_9 c_3 s_4 + k_8 s_{23} c_4 - k_9 s_3 c_4 + k_8 s_{23} s_4 + k_9 s_3 s_4)(\dot{\theta}_1)$$

$$p_{42} = \frac{1}{2}(-k_{13} c_4 - k_{11} c_4 - k_9 c_3 c_4 - k_9 s_3 c_4 + k_9 s_3 s_4 + k_{10} c_4 + k_{11} c_4 + k_9 c_3 c_4)(\dot{\theta}_2)$$

$$p_{43} = -\frac{1}{2}(k_{10} s_4 + k_{11} s_4)(\dot{\theta}_3)$$

$$p_{44} = 0$$

$$g(\theta) = \begin{bmatrix} 0 \\ (m_2 + m_3 + m_4) g a_2 cos\theta_2 + (m_3 + m_4) g a_3 cos(\theta_2 + \theta_3) + m_4 g a_4 cos(\theta_2 + \theta_3 + \theta_4) \\ (m_3 + m_4) g a_3 cos(\theta_2 + \theta_3) + m_4 g a_4 cos(\theta_2 + \theta_3 + \theta_4) \\ m_4 g a_4 cos(\theta_2 + \theta_3 + \theta_4) \end{bmatrix}$$

3.5. Image Processing

The food taken from the plate as a result of eye-tracking is brought to the pre-fixed position and the mouth tracking algorithm is started.

In the case of mouth scanning, the frame is taken from the camera 30 frames per second for each second which corresponds to 30 fps. Then, these frames are gray filtered and the image is grayed out to perform facial detection with 'frontal face.xml' which is one of the open-source HaarCascade classifiers [11]. After filtering, the Frontal Face HaarCascade algorithm was applied and face detection was performed.

After the face is found, a new pixel set is defined to process the pixels on which the face is found. HaarCascade classifiers are used again to find the mouth in the resulting cluster and this time 'mcs_mouth.xml' is used [12].

The basis of the HaarCascade classifier is that artificial intelligence trains the given data and learn what the mouth looks like. Images taken from datasets are divided into two, positive and negative images. While positive images contain the object we are looking for, negative images should not contain or even resemble this object. In general, the number of negative images is greater than the number of positive images. It should be noted that the more data, the better the HaarCascade classifier works. The tiring part is the selection of individual objects from positive images. Software is available on the Internet to simplify this process but still takes time. After the selection of the objects is finished, these data are trained with artificial intelligence and the HaarCascade algorithm is created. Even if the open-source HaarCascade classifiers do not work well, own cascade can be created, but the results are sufficient.

After the mouth is found, cluster is defined from the pixels where the mouth is located in order to narrow our working area further and masking process is performed in our defined color ranges in order to select the red tones in this cluster.

The location of the mouth creates two inputs. The first is the distance of the spoon on the horizontal axis with the user and the second is the height adjustment of the spoon. When the robot reaches the fixed position with the food, the mouth tracking application starts working as described above and the inputs are necessary for reaching the spoon to the mouth.

The first input, i.e. the distance difference in the horizontal axis, is obtained from the difference of the horizontal distance from the center of the frame to the point where the mouth is located in the fixed position of the camera shown in Figure 30 on the left side. The calculation is as follows:

$$\Delta X = C_X - \left(x + mx + \frac{mw}{2}\right) \ , \ x + mx = X \ , \ \frac{mw}{2} = \frac{W}{2}$$

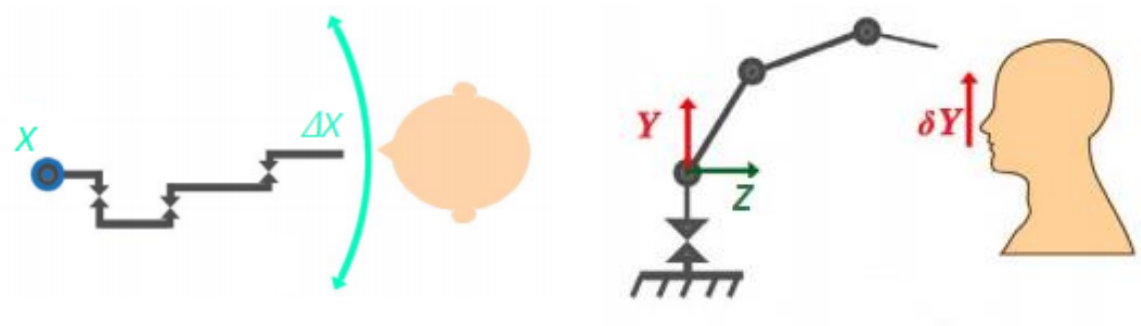

Figure 30: *Rotation of the robot in x-axis(Left) , lifting in y-axis(Right)*

Algorithm 1: Panning motion (X direction) adjustment algorithm

if $\Delta X < -50$ pixels **then**

$Panservo_{new} = Panservo_{old} - k;$

if $\Delta X > 50$ pixels **then**

$Panservo_{new} = Panservo_{old} + k;$

The second input is the difference in distance between the spoon and the mouth, which is the difference between the vertical distance in the middle of the frame and the vertical distance of the mouth. Vertical movements of the camera are represented as in Figure 30 on the right side.

$$\Delta Y = C_Y - \left(y + my + \frac{mh}{2}\right) \ , \ y + my = Y \ , \ \frac{mh}{2} = \frac{H}{2}$$

Algorithm 2: Height (Y direction) adjustment algorithm

if $\Delta Y < -200$ pixels **then**

$A_y = A_y - k;$

if $\Delta Y > -100$ pixels **then**

$A_y = A_y + k;$

Algorithm 2 is realized whether to increase or decrease the height of the spoon or end effector considering ΔY. Experimentally it is found out that keeping the identified mouth between -200 to -100 pixels gives the proper height needed to feed the food properly. Inverse kinematics equations of a 4 DOF planer robot are used to calculate the new joint angles according to the output by Algorithm 2.

3.5.1. Flowchart of the Algorithm

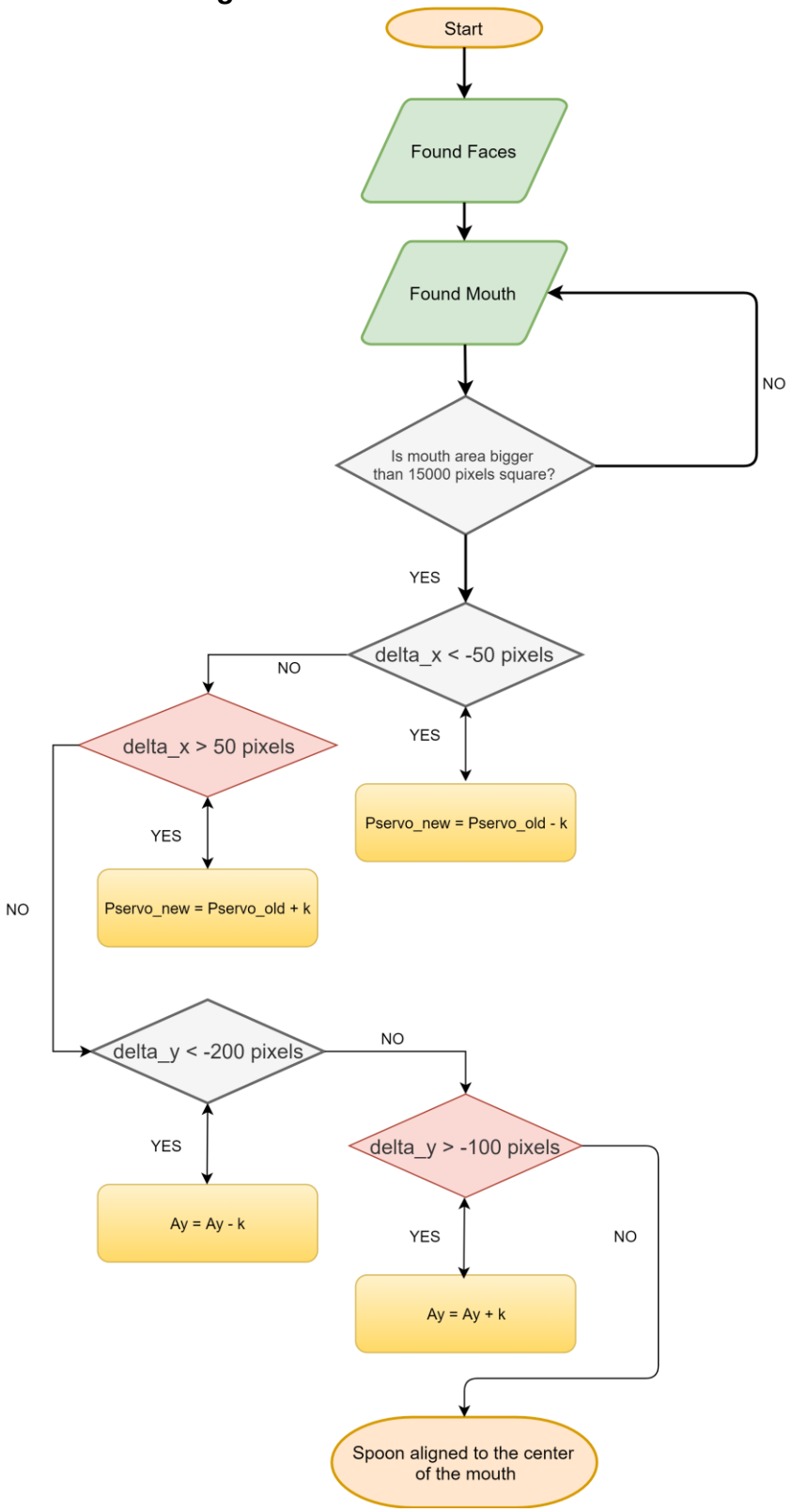

Figure 31: *Flowchart of the Algorithm*

4. Results

4.1. Results of the Strength Analysis

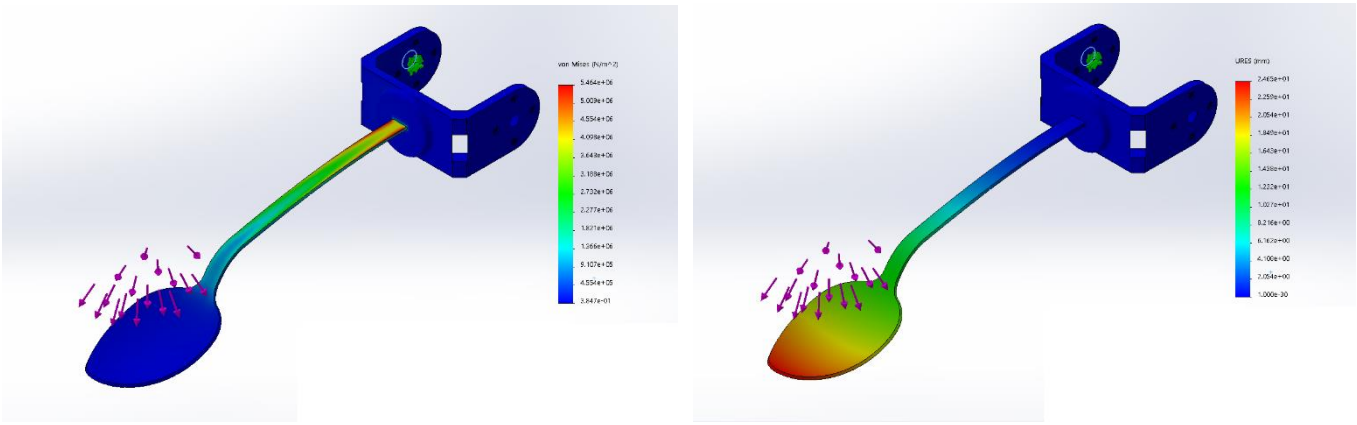

Figure 32: *(Left) Statically Stress Analysis of the end-effector of the robot. (Right) Statically displacement Analysis of the end-effector of the robot*

Figure 32:

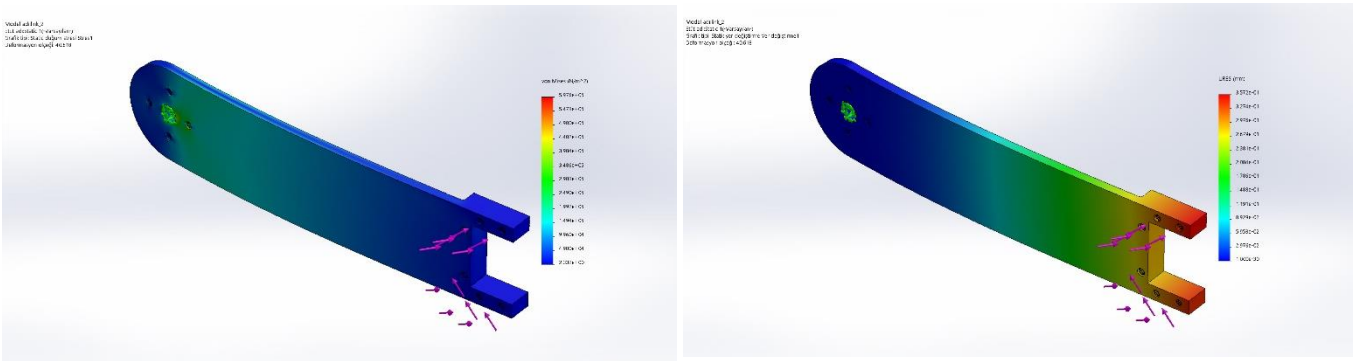

Figure 33: *(Left) Statically Stress Analysis of the second link of the robot. (Right) Statically displacement Analysis of the second link of the robot*

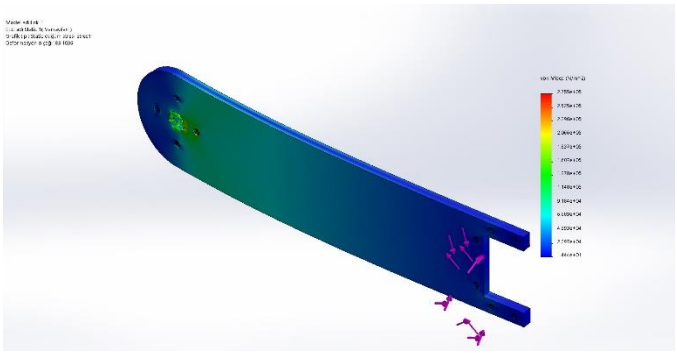

Figure 34: *Statically Stress Analysis of the first link of the robot*

4.2. Results of the Image Processing

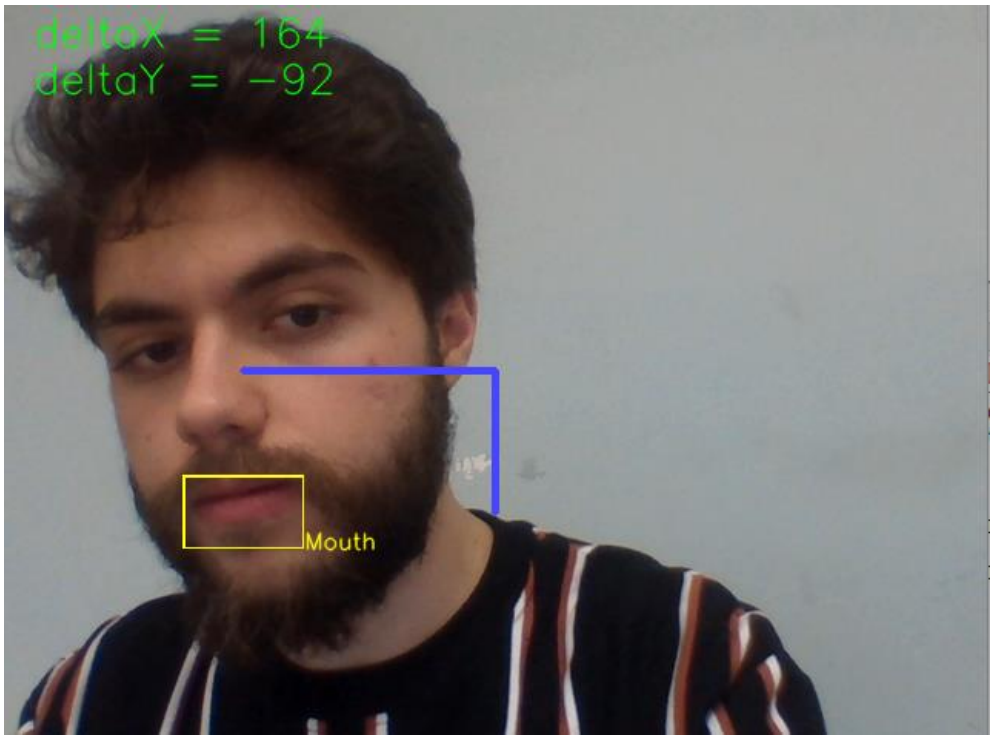

Figure 35: *Mouth Position Determined and Shown*

As you can see in Figure 35, the program output is demonstrated. The horizontal and vertical distance differences and the location of the opening are clearly indicated.

When the robot arm reaches the desired position range on the x and y axes, the program output is changed in Figure 36. The location of the mouth was found successfully and the position information results were obtained.

Figure 36: *Output changes when the position of the mouth is appropriate*

4.3. Results of the Simulation of the Feeding Assistance Robot

In the previous parts of this document inverse and forward kinematics were investigated. These subjects are also called as the geometry of the robot arm. With these analyses, the mathematical model of the robot arm was reached. These solutions are completely related to the intrinsic of the robot arm.

However, in this part of the project, the main focus was to show that the robot arm can work properly in its workspace. In order to get a properly working robot arm, the robot arm should be able to move from one to another in a smooth way, firstly. In other words, this smooth move problem could be named as trajectory planning. Here trajectory means that the position, velocity and acceleration values of each joint are known at that exact same time. This will provide the robot arm a smooth movement and will make us sure that the system maintains our requirements.

In this part of the project, all of the specified movements were visualized in our simulation environment and relative outputs such as position, velocity, acceleration comparison of solutions that were made in different spaces are recorded. Furthermore, the source codes, mechanical design and assembly files, video record of the simulations and detailed explanations about the steps of implementation of a robot arm in a simulation environment were shared in our team's GitHub repository [13] so anyone who wants to simulate a robot arm could find this source helpful.

The first thing that was done to create a URDF file from the robot arm's assembly file which created in SOLIDWORKS, an extension called SolidWorks to URDF Exporter [14] was used to create this URDF file. In this part, choosing joint axes and origins for these joint axes were important. After done with these, relations between links must be assigned with the extension's easy to use interface. The interface of the SolidWorks to URDF Exporter could be seen in figure 37.

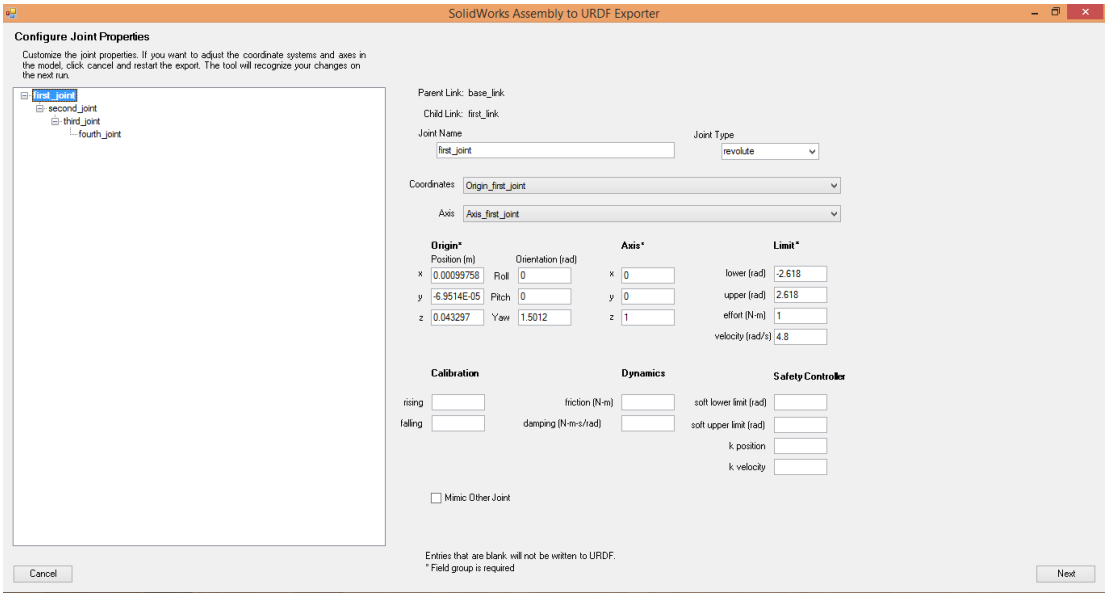

Figure 37: *The interface of the Solidworks to URDF Exporter*

After we got the URDF file from Solidworks, the project was ready to be worked on it. For the simulation part, MATLAB and Simulink were used. Especially, a toolbox called Robotics System Toolbox made this simulation available. Robotics System Toolbox provides tools and algorithms for designing, simulating, and testing manipulators, mobile robots, and humanoid robots. For manipulators and humanoid robots, the toolbox includes algorithms for collision checking, trajectory generation, forward and inverse kinematics, and dynamics using a rigid body tree representation [15].
The first step in MATLAB was to show the robot arm in the simulation environment. In figure 38 below you can see the robot arm and the second joint colored in yellow and its positive revolute direction also.

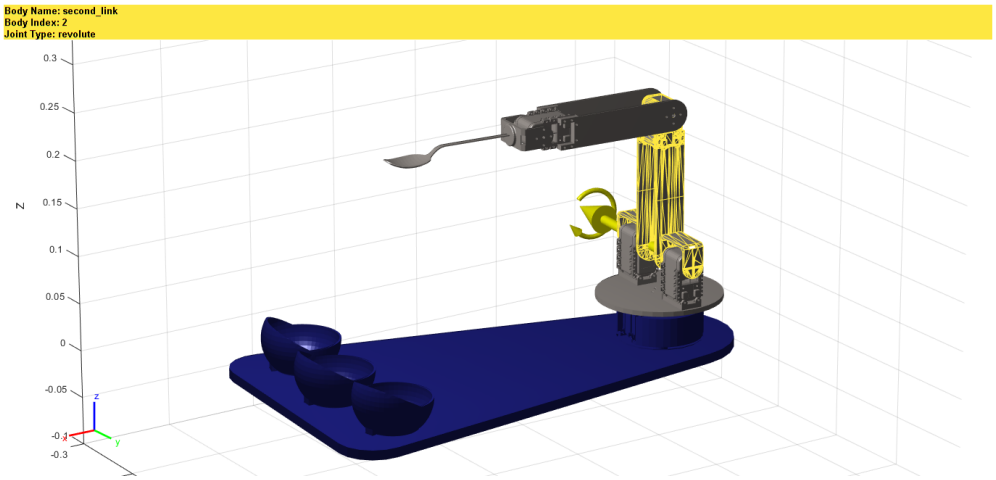

Figure 38: *The robot arm in the simulation environment*

After the import operation was done the path planning could be done. There are two spaces that we could use in trajectory generation and these are joint space and task spaces. Both of these solutions have pros and cons, for instance, while the joint space provides a faster solution, the task space, on the other hand, provides better handling of obstacles and collisions [16].

To show the difference between these two solution methods, two trajectories from the home position to the middle bowl was investigated. These trajectories were followed by the end-effector of the robot arm. The home position in this situation was just the position as seen in figure 39. Also, the robot moved to a point on top of the bowl. In figure 39 trajectories produced with joint space and task space methods are shown. While the blue line states the task space solution the dotted red line states the joint space solution.

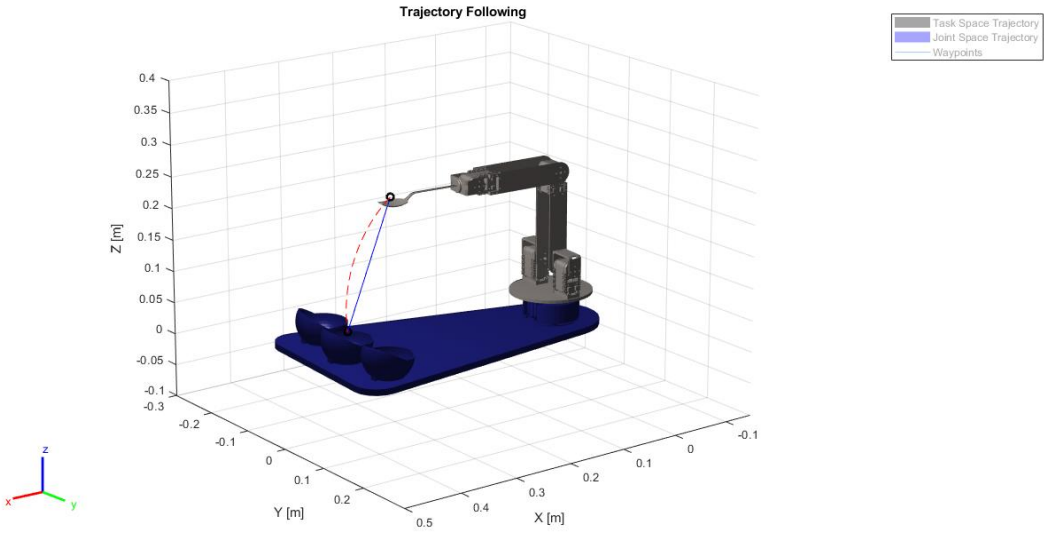

Figure 39: *The comparison between joint space and task space solutions*

The main difference between these two solutions was the execution time. In the case of task space solution, inverse kinematics of the robot arm must be solved in every point on the trajectory. However, in the joint space solution, the inverse kinematics of the robot arm has to be solved just in the beginning and endpoints. This trajectory shown in the above figure was completed in four seconds and this movement was divided into 201 timepieces. So our program has to solve inverse kinematics of the robot arm for 201 times. The clear execution time difference could be seen from the results of our system. After the simulation was run in our own computer (core i5, 8GB RAM), the program completed in 14.0772 s for the joint space solution. On the other hand, it took 1192.8127 s (which is nearly 20 minutes) to calculate for task space solution.

The execution time depends on the number which the system has to solve. In the above case, the step time was chosen as 0.02 s which means that inverse kinematics must be solved in every 0.02 s between 0s and 4s. However, it is still possible to get a smooth motion with bigger intervals. For example, when 0.2 s is taken as step time the trajectory was calculated in 112.5419 s with task space solution. However, there wasn't any clear difference in joint space solution and it took 13.5797 s to solve it. Because our system will always do same motion while reaching the bowls and to the home position and these trajectory planning solutions could be saved and be used in every time task space solutions will be used in our trajectories.

One other reason for choosing the task space solution was that its position plot is differentiable and guarantees a smooth motion. Furthermore, there was no clear difference between task space and joint space as it could be seen in figure 40. This figure illustrates joint positions solved in the task and joint space for all four joints.

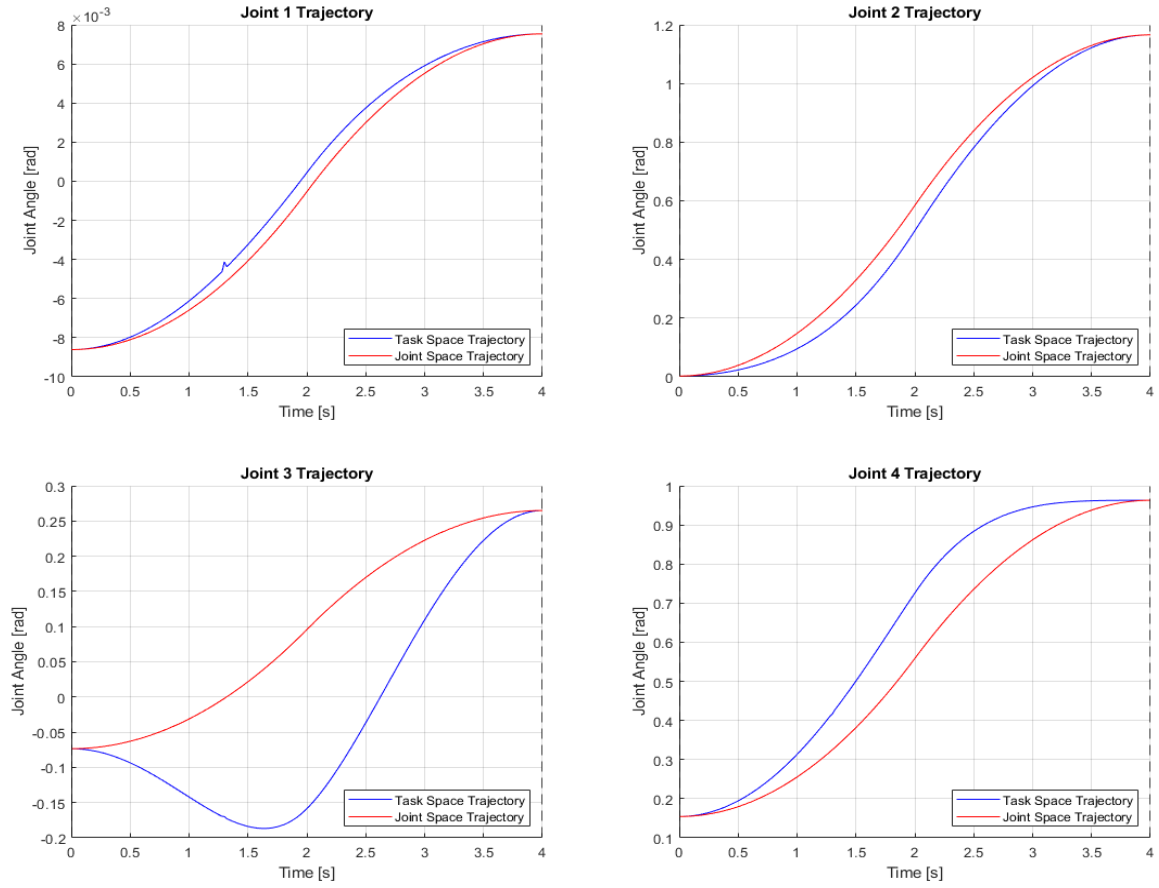

Figure 40: *The comparison of the joint positions produced with joint space and task space for all four joints*

The simulation also proved that our servo motors are capable to work properly. As it is seen from figure 41 the maximum angular velocity that the system reached was $0.6\,rad/s$ in the second joint. In this angular velocity, our servo motors can provide $1.69\,N \cdot m$ torque which is obviously higher than our required torque value for that joint which is $0.40\,N \cdot m$ in static conditions. Furthermore, other joints' angular velocity plots with respect to time, is seen in figure 41.

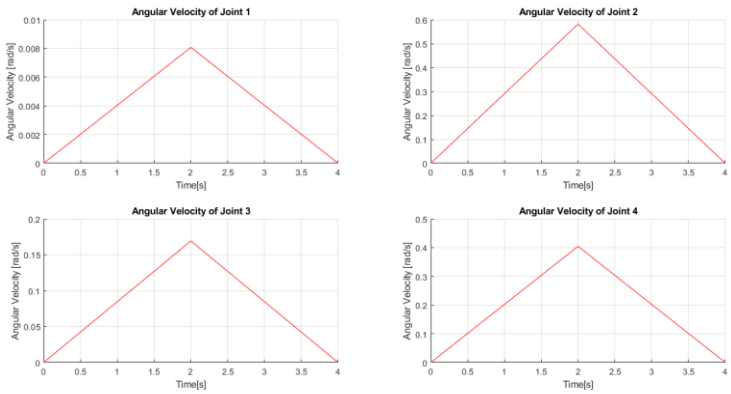

Figure 41: *Angular velocity plots for each joint*

Moreover, the angular acceleration of each joint was investigated, it is seen in figure 42 when these acceleration values put into the dynamics analyses that mentioned earlier, it could be proven that our servo motors could provide these torque values. Also, it is seen from these figures that there are no sudden acceleration changes which could affect the smoothness of the motion. According to these plots, the maximum sudden change in the acceleration value is $0.6\,rad/s^2$.

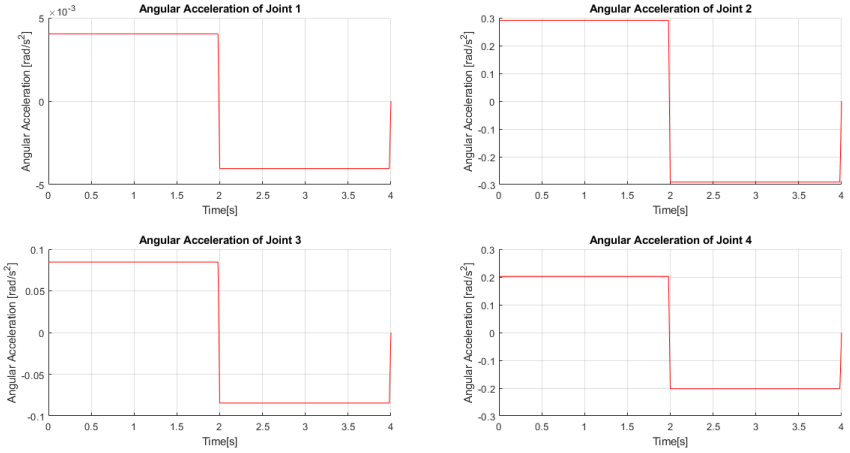

Figure 42: *Angular acceleration plots for each joint*

After the trajectories were explored from the home position to the bowl in the middle, movements from the home position to the bowls in the right and left side of the base plane were explored. However, the same code used for the simulation of previous trajectories we talked about in the above couldn't be used because the end-effector also has to have an orientation in these new trajectories. In order to provide this orientation at the end of this path, our system has to calculate the orientation matrices in addition to the position vector and must solve the inverse kinematics with respect to the whole 4×4 transformation matrix. It is seen in the equation (1) that a transformation matrix without any rotation and in the equation (2) that a transformation matrix with a rotation matrix.

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

$$T = \begin{bmatrix} & R_{3\times3} & & p_x \\ & & & p_y \\ & & & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

As it can be seen from figure 43 the end-effector must have a 15 degrees rotation in the z-axis and must have no rotation in other axes. Also, it could be observed from the figure clearly that the system solved the rotations and positions at the same time for each step.

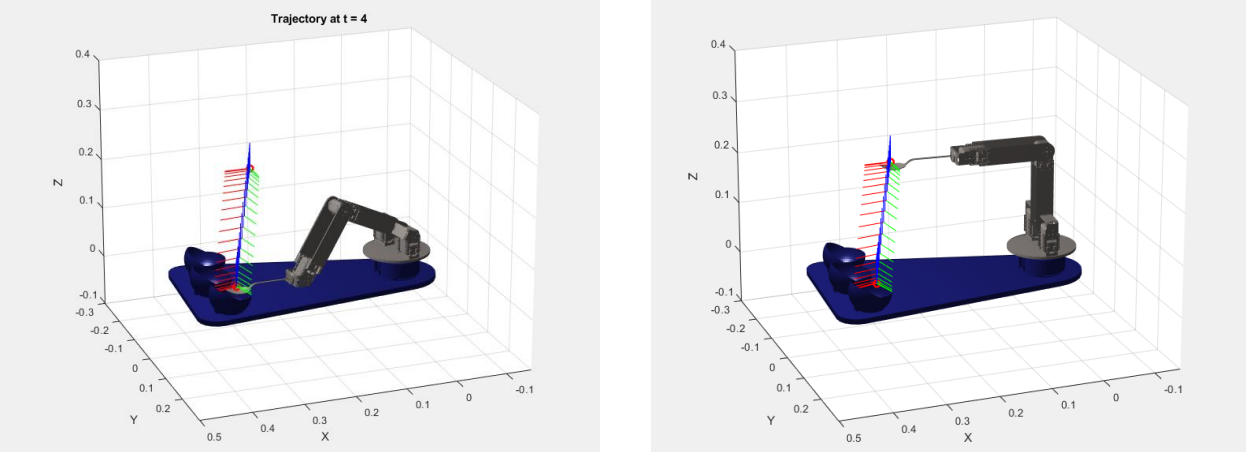

Figure 43: *End-effector with rotation on the endpoint(left), end-effector in the home position*

According to our settings, the end-effector moves from home position to the bowl in the right in four seconds. And, the time interval was assigned as 0.2 s which means there were just 21 points that our system has to solve the inverse kinematics in these points. Because we can save the data produced by the solution of inverse kinematics the execution time wasn't considered as a constraint. For that reason, just the task space solution was used to creation of the trajectory. In the case 0.2 s was used as time interval the system was able to create the trajectory in 130.785 s which is reasonable execution time. Also, 0.1 s was tested for the comparison and in that case, our system was able to create the trajectory in 237.19 s.

In addition to these trajectories, the trajectory from home position to the bowl in the left was simulated. As it could be seen from figure 44 it is just the mirrored pair of the previous trajectory. Furthermore, the execution times were nearly the same for both of these trajectories. The only difference in this motion was that the orientation of the end-effector at the endpoint was -15 degrees.

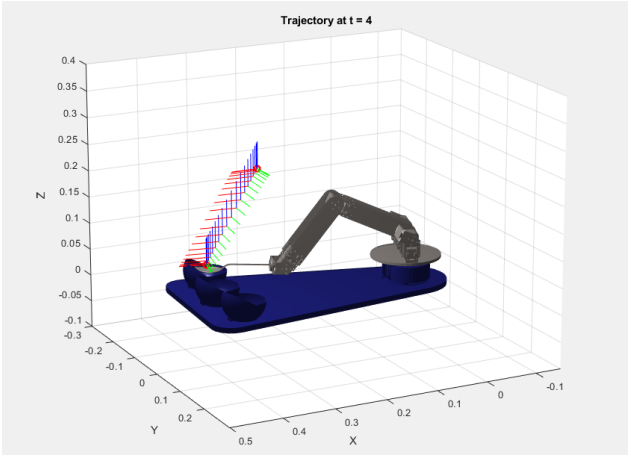

Figure 44: *End-effector with rotation on top of the bowl in the right*

Totally, six different trajectories for the robot arm were created and simulated in Matlab. To be able to evaluate these, firstly, the assembly file of the robot arm was exported to a URDF file. After that, some arrangements were made in the Matlab environment to control the robot using functionalities of the Robotics System Toolbox. Then, waypoints that the end-effector of the robot must be followed were given to our system and also rotations in those waypoints were given to the system. The following was telling the system the trajectory planning method. In our case trapezoidal trajectory planning was chosen and tested, other trajectory planning methods could be chosen. After the system was run and created trajectories the final step was to run the simulation part of the code.

After simulation results were obtained, the controller design was investigated. In our project, servo motors were used as a rotary actuator. In that case, there is no need to design a controller because servo motors used in our project contains its own controller circuit inside of it. If somebody needs to control these motors, s/he just needs to install its software development kit [17] and control them with couple lines of codes. However, in the scope of the Mechatronic System Design course design of a controller is a must. For this reason, a basic PID controller was developed in the Simulink environment to control the position output as seen in figure 45.

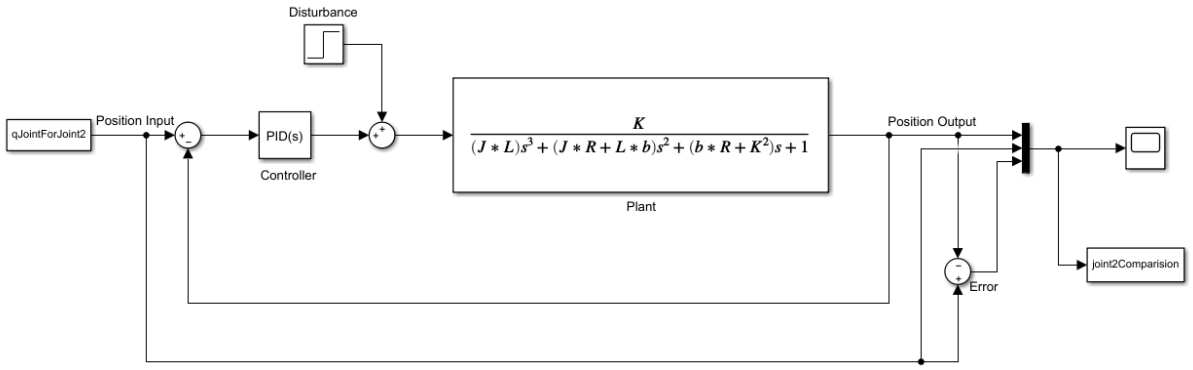

Figure 45: *Simulink blocks for position control*

In this controller design, some assumptions were made. As it has known a servo motor is comprised from a DC motor, gears and a controller circuit. And, to design the Plant part of the controller some physical parameters (like the moment of inertia of the rotor, motor torque constant e.g.) must be known exactly. However, there was no information on the producer side about any of this parameter. Even though there are some written programs in Simulink to find these parameters' values, we couldn't reach those because we haven't been able to buy the connection part for the servo motor.

Instead, some constants were used to simulate the servo motor These values were derived by experiment from an actual motor in Carnegie Mellon University's undergraduate controls lab [18].

$$(J)\ Moment\ of\ inertia\ of\ the\ rotor\ \ 3.2284 \cdot 10^{-6}\ kg \cdot m^2$$

$$(b)\ Motor\ viscous\ friction\ constant\ \ 3.5077 \cdot 10^{-6} N \cdot m \cdot s$$

$$(K_b)\ Electromotive\ force\ constant\ \ 0.0274\ V/rad/s$$

$$(K_t)\ Motor\ torque\ constant\ \ 0.0274\ \ N \cdot m/Amp$$

$$(R)\ Elecrical\ resistance\ \ 4\ Ohm$$

$$(L)\ Electrical\ Inductance\ \ 2.75 \cdot 10^{-6} H$$

The transfer function for the PID controller was like that:

$$C(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

This PID controller's parameter was chosen with the help of Simulink's PID Tuner App, and the best parameters for our system were given like these:

$$K_p = 324, K_i = 32186, K_d = 0.57$$

As it has seen from the above figure, the position input, the output position and the error between these two signals were plotted in the same scope. This scope could be seen in figure 46. As seen from this figure the controller worked accurately and produced an output signal as exactly the same with the input signal. Figure 46 illustrates signals for the third joint, in this case, the blue line represents the output while the red line represents the input signal. The error was about 0.01 rad at the beginning of the movement.

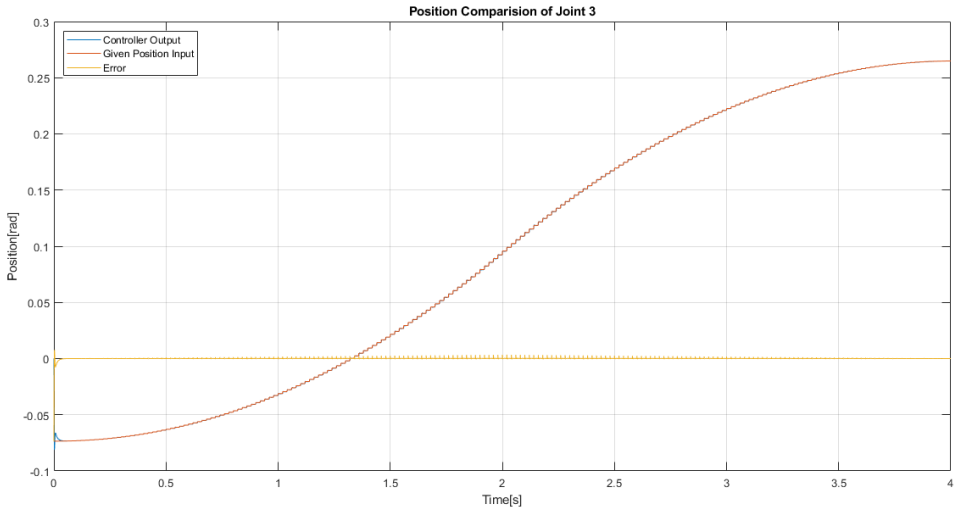

Figure 46: *Comparison of input, output signals and the error between two of them for third joint*

Also, the controller characteristic without auto-tuning was observed, in this case, PID parameters were investigated by plots of the output for a given step input. With the help of many loops which enabled us to test many cases, our controller had a sufficient step response. And, the application of this controller for the second joint could be seen in figure 47.

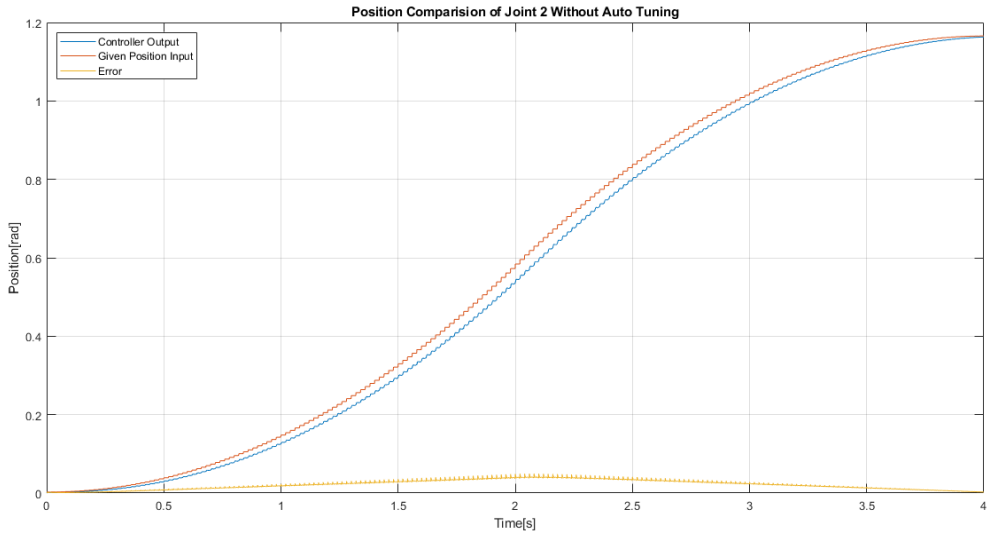

Figure 47: *Comparison of input, output signals and the error between two of them for second joint*

Also, one other simulation environment was created for the test of the controllers developed for the motors. In order to evaluate this simulation, the assembly file of the robot arm was converted to a Simulink structure by using the Simscape Multibody Link add-on. After this conversation, the robot arm was seen in the Simulink environment as in figure 48.

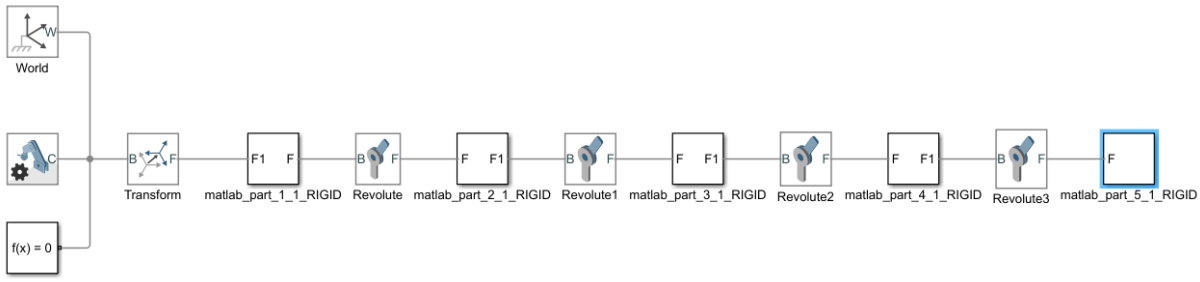

Figure 48: *The structure in the Simulink environment, which represents the robot arm.*

After this structure was obtained, the next step was to convert this structure to a subsystem for simplicity. This subsystem was created with determining its input and output ports. In this subsystem, the position input signals which were provided by a PID controller were the input signals. The position output signals and the torque values were the output values of the system as it would be seen in figure 49.

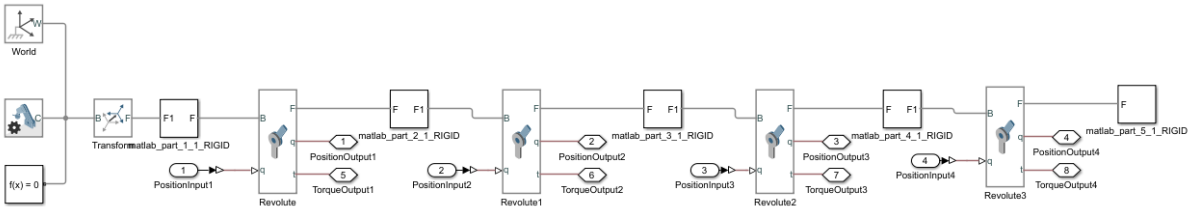

Figure 49: *The structure of the subsystem*

Then all four joints must be controlled by a controller. In our case, the PID controller was chosen because it provides proportional (P), integral (I) and derivate (D) control terms. All of the PID controllers were added before the input terminal of the subsystem as it could be seen in figure 3. The same parameters were used in all four PID controller because it was assumed that all joints were actuated by the same servo motor. And, these are the parameters obtained by the PID Tuner App in MATLAB:

$$K_p = 2, K_i = 14200, K_d = 6.50 \times 10^{-5}$$

Also, the comparison plots were obtained which shows the difference between the input signal and output signal. Other than these plots one plot was added to the system to show the joints torque values as it could be seen in figure 50.

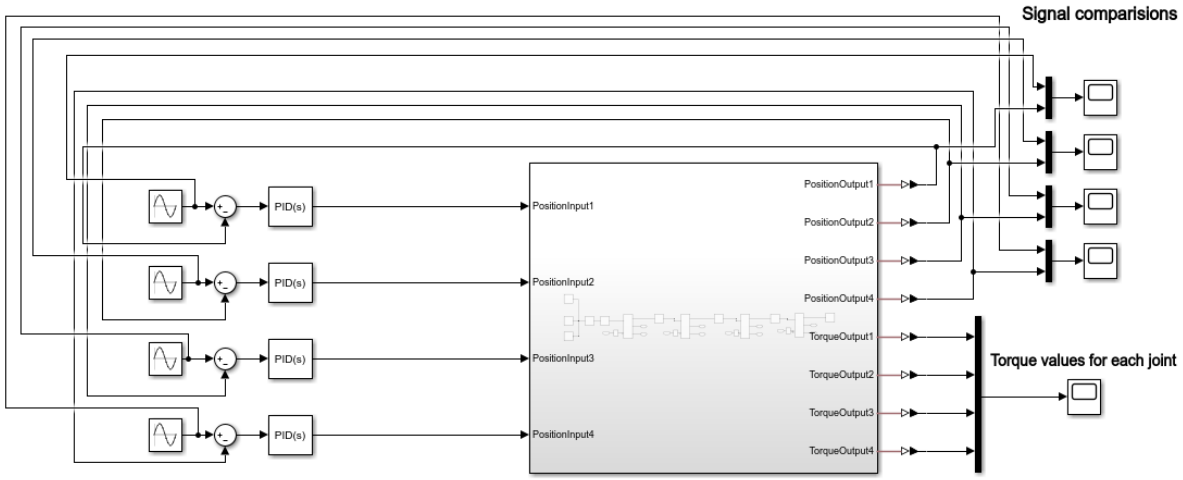

Figure 50: *The overall system to control the robot arm.*

From this simulation, it could be said that our PID controller was pretty sufficient to meet the system's requirements to be controlled. This result is also clearly seen in figure 51. This figure represents the comparison of the reference (input) and output position signals and as it is seen from that these two signals are exactly the same.

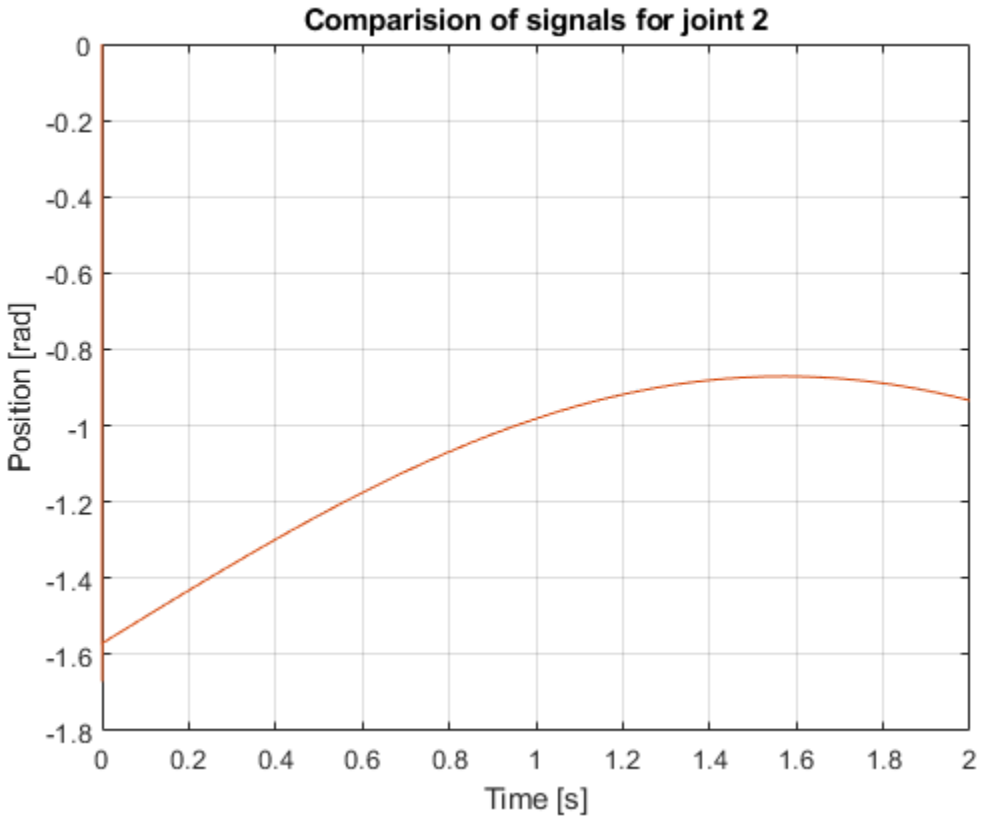

Figure 51: *The comparison of reference and output signals for the second joint.*

Moreover, this simulation gives the torque values which are required to move the desired joint position. And, the maximum torque was required in the second joint and it was around the $0.22\ N \cdot m$ which is clearly lower than the torque values that our servo motors could supply.

5. Cost and Product Backlog Items Tables

5.1. Cost Table

Table 6: *Cost Table of the Feeding Assistance Robot*

Product Name	Brand/Model	Quantity	Cost (₺)
Servo Motors	Dynamixel AX18-A	2 pieces	1541,82
Servo Motors	Dynamixel AX12-A	3 pieces	1094,34
USB to TTL Converter	USB2Dynamixel	1 piece	267,17
TTL Cables	Robotis 3P 140mm	10 pieces	767,6
Camera	Logitech C310 HD Webcam	2 pieces	541,78
Lipo Battery	ProFuse 11.1V 3S 2800 mAh	1 piece	200,82
3D Printed Parts	colorFabb	1 piece	227,99
Bolts	Robotis M2	72 pieces	28,74
Bolts	Robotis M3	6 pieces	28.74
Caster Wheel	Robotistan	3 pieces	42,12
		Total Cost:	**4712,38**

5.2. Product Backlog Items Table

Table 6: *Product Backlog Items Table of the Feeding Assistance Robot*

Sprint Number	Work in Progress(WIP)	Tasks
1	Upper Side Manufacturing of the robotic arm	Manufacturing of the Base Retainer
		Manufacturing of the First Link
		Manufacturing of the Second Link
		Manufacturing of the Bottom Plate
		Manufacturing of the Connecting Kit
	Redesign of the end-effector	Design of the Spoon
		Check the Dynamic Analysis
2	Integration of the servo motors	Control of the servo motors
		Communications with computer
		Connection with robot
	Tested of the servo motors	Testing the movement
3	Manufacturing of other parts of the robotic arm	Manufacturing of the Base Plate
		Manufacturing of the Bowl
		Manufacturing of the Connecting Holder
		Manufacturing of the Spoon
	General integration of the system	Integrations of the overall system
4	Integration of the Cameras	To locate the camera to suitable position without change the center of gravity to obtain more accurate, smooth motion
		Connection with computer
		Testing the camera
	Mouth-Tracking integration to robotic arm	Output of the mouth-tracking algorithm used in the servo motor controls to obtain motions on x,y axes.
		Creating the open-closed mouth detection and testing feeding operation
5	Design of User-Interface	Design of the GUI(Graphical User Interface)
		Coding of each actions for each buttons
6	Integration of eye-tracking	Testing of eye-tracking libraries
		Integrations of eye-tracking with our GUI
7	Testing of prototype	Test of the system with development team
		Test of the system with patients

Note: Each Sprint will take two weeks.

6. Conclusion

For people who have upper limb disabilities, this feeding robot design will be helpful so much. It won't only give back their independence back to them, it also will reduce the need for a caregiver. In conclusion, in the scope of this project, the mechanical design of a 4-DOF robotic arm was done to serve as a feeding assistant robot, the forward and inverse kinematics of it were developed. Then dynamical analyses of this robot were done. After these, each part of the robot arm design was subjected to strength analysis. A simulation of the robot arm to show that it is capable of working in its workspace was developed. Finally, the mouth tracking feature of the system was written, and it was tested successfully.

These works are the things that have been done during the fall semester for the scope of the Mechatronics System Design course. In the next semester, this design will be realized in the scope of the Senior Design course. First, the servo motors will be tested for trajectory following. Then, each part of the design will be manufactured with 3D printers. After that, mechanical and electrical integration will be held. Next, each different trajectory for each of the bowls will be tested. Finally, the eye-tracking and mouth tracking software will be integrated with the trajectory following software.

7. References

[1] Tanaka, Kanya et al. “Meal-Assistance Robot Using Ultrasonic Motor with Eye Interface.” IJAT 8 (2014): 186-192.

[2] 2019. Engelsiz Yaşama Derneği. (2019). https://ey-der.com/ana-sayfa/turkiye-ve-dunyada-engelliler/, [Online; Retrieved on 4th December, 2019]

[3] T. Bhattacharjee, M. E. Cabrera, A. Caspi, M. Cakmak, and S.S. Srinivasa. A Community-Centered Design Framework for Robot-Assisted Feeding Systems. In International ACM SIGACCESS Conference on Computers and Accessibility. 2019.

[4] 2019 Obi Robot. https://meetobi.com/, [Online Retrieved on 24^{th} December , 2019]

[5] Meal Buddy Robotic Assist Feeder https://www.performancehealth.com/meal-buddy-systems , [Online, Retrieved on 24^{th} December , 2019]

[6] Bestic, https://www.camanio.com/us/products/bestic/ , [Online, Retrieved on 24^{th} December , 2019]

[7] C. J. Perera, T. D. Lalitharatne, K. Kiguchi. “EEG- Controlled Meal Assistance Robot with Camera-Based Automatic Mouth Position Tracking and Mouth Open Detection” IEEE International Conference on Robotics and Automation (ICRA). 2017

[8] Mark W. Spong, Seth Hutchinson & M. Vidyasagar, Hoboken, NJ: John Wiley & Sons, 2006. Robot Modelling and Control (p. 239)

[9] A. Roshanianfard, N. Noguchi. “Kinematic Analysis and Simulation of a 5 DOF Articulated Robotic Arm Applied to Heavy Products Harvesting”

[10] Z.Y. Bayraktaroglu, 2018, Robot Teknolojisine Giris, Istanbul, Nobel Akademik Yayincilik

[11] Automatic camera selection mechanism based on facial feature detection. [Online]. Available: https://github.com/een450/MasterProject

[12] OpenCv Library. [Online]. Available: https://github.com/opencv/opencv/tree/master/data/haarcascades

[13] FeedingAssistantRobot. (2019, December). [Online]. Available: https://github.com/Ugurmustafa97/Feeding AssistantRobot

[14] SolidWorks to URDF Exporter. (2019, December). [Online]. Available: http://wiki.ros.org/sw_urdf_exporter

[15] Robotics System Toolbox. (2019, December). [Online]. Available: https://www.mathworks.com/products /robotics.html

[16] Trajectory Planning for Robot Manipulators. (2019, December). [Online]. Available: https://www.mathworks .com/videos/trajectory-planning-for-robot-manipulators-1556705635398.html

[17] Dynamixel SDK. (2019, December). [Online]. Available: http://emanual.robotis.com/docs/en/software /dynamixel/dynamixel_sdk/overview/#dynamixel-sdk

[18] DC Motor Position: System Modeling. (2019, December). [Online]. Available: http://ctms.engin.umich.edu /CTMS/index.php?example=MotorPosition&section=SystemModeling

[19] Camera Selection: http://infodif.com/blog/goruntu-isleme-amaciyla-optik-algilayici-kamera-seciminde-onemli-faktorler/