T.C.

AKSARAY UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

DEPARTMENT OF ELECTRICAL-ELECTRONIC AND COMPUTER ENGINEERING

PATH PLANNING FOR 3 DEGREE OF FREEDOM (RRR) ROBOT AND ITS APPLICATION

Ph.D. THESIS

Hasan DEMİR

SUPERVISOR

Prof. Dr. MEHMET REŞİT TOLUN

AKSARAY, 2020

Aksaray University, Graduate School of Natural and Applied Sciences Approval, Hasan DEMİR, Doctor of Philosophy with thesis, Student No. 162353302 successfully presented the Doctor of Philosophy Thesis titled "(**PATH PLANNING FOR 3 DEGREE OF FREEDOM (RRR) ROBOT AND ITS APPLICATION**)", prepared after satisfying all the requirements identified in the concerned bylaws, before the jury whose signatures are affixed below.

Thesis Advisor: Prof. Dr. Mehmet Reşit TOLUN
Konya Food and Agricultural University
I approve that this thesis is a Doctor of Philosophy in scope and quality ....

Member: Assoc. Prof. Dr. Yunus UZUN
Aksaray University
I approve that this thesis is a Doctor of Philosophy in scope and quality ....

Member: Asst. Prof. Dr. Abdül Kadir GÖRÜR
Çankaya University
I approve that this thesis is a Doctor of Philosophy in scope and quality ....

Member: Assoc. Prof. Dr. İsmail BAYRAKLI
Aksaray University
I approve that this thesis is a Doctor of Philosophy in scope and quality ....

Member: Asst. Prof. Dr. Abdülsamed TABAK
Necmettin Erbakan University
I approve that this thesis is a Doctor of Philosophy in scope and quality ....

Date of Defence: 25/06/2020

I agree that this thesis which is accepted by the jury, has fulfilled the requirements for being a Doctor of Philosophy Thesis.

....

Assoc. Prof. Dr. Mehmet Ali HINIS
Director of Natural and Applied Science

PLAGIARISM PAGE

I hereby declare that all information in this thesis has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work; otherwise I accept all legal responsibility.

Hasan DEMİR

ACKNOWLEDGMENTS

First of all, I would like to express my deepest appreciation to my supervisor, Prof. Dr. Mehmet Reşit TOLUN, who gave me this great research opportunity. I am heartily thankful to him for his tireless guidance, extraordinary patience and constant encouragement during my academic journey. Without his persistent help, this dissertation would not have been possible.

I would like to thank my committee members, Assoc.Prof. Yunus UZUN and Assist. Prof. Dr. Abdul Kadir GÖRÜR who examine my dissertation and provide me so many valuable suggestions.

Finally, I am grateful to all my family members, especially my wife Seval DEMİR, my son Ahmet Efe DEMİR and my father Hasan Hüseyin DEMİR for always believing in me and encouraging me to do my best. All of their love, support, and sacrifices over these past years gave me the strength and determination to make my dreams come true.

Hasan DEMİR
AKSARAY, 2020

TABLE OF CONTENTS

DOKTORA TEZİ

3 SERBESTLİK DERECESİNE SAHİP (RRR) ROBOTUN YÖRÜNGE PLANLAMASI ve UYGULAMASI

Hasan DEMİR

Aksaray Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Mehmet Reşit TOLUN

ÖZET

Bu çalışmada, 3 serbestlik derecesine sahip RRR robotun yörünge planlaması yapılmış ve 3d yazıcı ile üretilen model robot kol üzerinde uygulanmıştır. Yörünge planlaması yapılırken genetik algoritmalar kullanılarak zaman optimizasyonu yapılmıştır. Model olarak ilk üç eklemin Kartezyen uzayda konumu etkilemesi nedeniyle üç eklemi de döner eklem olan RRR yapıdaki robot kol kullanılmıştır. Robot kolun öncelikle ileri kinematik analizi yapılmıştır. İleri kinematik analiz yöntemi olarak en çok kullanılan analiz yöntemi olan Denativ-Hartenberg yöntemi seçilmiştir. İleri kinematik analiz sonucunda elde edilen bilgiler ile ters kinematik analiz yapılmıştır. Ters kinematik analiz sonucunda model robotun Kartezyen uzayda bir noktaya ulaşması için gerekli olan eklem değişkenlerini elde edilmiştir. Kinematik analizler robotun eklem uzayında yörünge planlamasını yapmak üzere gerçekleştirilmiştir.

Robot kolun yörünge planlaması yapılırken hareketlerini en kısa sürede tamamlaması için genetik algoritmalar kullanılarak zaman optimizasyonu yapılmıştır. Eklem uzayında yörünge planlama sonucunda her bir eklem için elde edilen hız ve ivme denklemleri optimizasyonda amaç fonksiyonları olarak kullanılmıştır. Amaç fonksiyonunun sınırlarını robot kolun eklemlerinde kullanılan servo motorların hız ve ivme değerleri oluşturmuştur. Optimizasyon sonucunda her bir eklemin hareketini en kısa zamanda tamamlayabileceği süreler bulunmuştur. Model robot kol 3d yazıcıda üretilmiş ve deney seti oluşturulmuştur. Optimizasyon sonucunda elde edilen veriler deneysel çalışmalardan elde edilen veriler ile karşılaştırılmıştır. Bu çalışmanın sonuncunda genetik algoritmalar kullanılarak zaman optimizasyonu yapılmış ve her hangi bir robota uygulanabilecek hareketi en kısa sürede tamamlayan bir model oluşturulmuştur.

Anahtar Kelimeler: İleri kinematik, Ters kinematik, Yörünge planlama, Genetik Algoritmalar, Zaman optimizasyonu, 3d yazıcı, RRR robot kol.

Haziran, 2020; 70 sayfa

Ph.D. THESIS

PATH PLANNING FOR 3 DEGREE OF FREEDOM (RRR) ROBOT AND ITS APPLICATION

Hasan DEMİR

Aksaray University
Graduate School of Natural and Applied Sciences
Department of Electrical Electronic and Computer Engineering

Supervisor: Prof. Dr. Mehmet Reşit TOLUN

ABSTRACT

In this study, the path planning of the RRR robot with 3 degrees of freedom has been made and has been applied on the model robot arm produced with 3d printer. Time optimization has been made by using Genetic Algorithms during path planning. Since the first three joints affect the position in the Cartesian space, the robot arm with the RRR structure which has the three rotating joints, was used as the model. Firstly, the forward kinematic analysis of the robot arm was performed. Denativ-Hartenberg method, which is the most widely used analysis method, was chosen as the forward kinematic analysis method. Inverse kinematic analysis was performed with the information obtained from the forward kinematic analysis. As a result of inverse kinematic analysis, the joint variables required for the model robot to reach a point in Cartesian space were obtained. Kinematic analyzes were performed to planning path in the joint space of the robot.

Time optimization was performed by using Genetic Algorithms in order to complete the movements of the robot arm as soon as possible while planning path. The velocity and acceleration equations obtained for each joint as a result of path planning in joint space were used as objective functions in optimization. The limits of the objective function are the velocity and acceleration values of the servo motors used in the joints of the robot arm. As a result of optimization, it was found that each joint can complete the movement as soon as possible. The model robot arm was produced with the 3d printer and the experimental set was created. The data obtained as the result of optimization were compared with the data obtained from experimental studies. At the end of this study, time optimization was made by using Genetic Algorithms and a model which completed the movement in the shortest time was created that could be applied to any robot.

Keywords: Forward kinematics, Inverse kinematics, Path planning, Genetic Algorithms, Time optimization, 3D printer, RRR robot arm.

June, 2020; 70 pages

LIST OF FIGURES

LIST OF TABLES

ICONS AND ABBREVIATIONS

ABS	Acrylonitrile Butadiene Styrene
CAD	Computer Aided Design
CNC	Computer Numerical Control
D-H	Denativ-Hartenberg
DOF	Degrees Of Freedom
FFF	The Fused Filament Fabrication
GA	Genetic Algorithm
GUI	Graphical User Interface
KLR	Robot Programming Language
LbD	Learning by Demonstration
MAF	Moving average filter
PLA	Polylactic Acid
PSO	Particle Swarm Optimization
PUMA	Programmable Universal Machine for Assembly
RRR	Three Rotary Joints
T	Transformation Matrix
TCP	Tool Center Point
TP-GMM	Task Parametrized Gaussian Mixture Model
$\mathbf{a_{i-1}}$	The Bond Length Between Two Axes
$\mathbf{d_i}$	The Joint Misalignment Between Overlapping Bonds
$\boldsymbol{\alpha_{i-1}}$	The Bond Angle Between Axes (i-1) and i
$\boldsymbol{\theta_i}$	The Joint Angle Between Two Bonds
$\boldsymbol{\theta'(t)}$	Velocity
$\boldsymbol{\theta''(t)}$	Acceleration

1. INTRODUCTION

Due to increased product and product diversity, productivity needs to be increased in order to protect competitiveness. For this reason, the use of robots in automation is a suitable solution. In the 1930's, industrial robots were first made to paint the walls with spray paint. In the 1960s, the first industrial robots were used for transport operations. Today, industrial robots are frequently used in many industrial sectors. To control an industrial robot, its configuration, the lengths of links and its joints must be well known. Using these known variables, Denativ-Hartenberg (D-H) parameters are found and written in tables. The forward kinematic solution is developed to define the relationship between the first and last joints. Then the inverse kinematic solution is used to find the equations of the joint variables. Forward and inverse kinematic equations are used in the path planning equations that are needed to move the robot in Cartesian space. There are many different studies in the literature about this subject. Information on these studies was given in this section.

A robot manipulator is consist of serial links which connected to each other revolute or prismatic joints. Each joint location is defined to relative previous joint. The relation is described by homogeneous transformation matrix. The homogeneous transformation matrix have orientation and position data. The number of its remarks the degrees of freedom (DOF) of robots. Each homogeneous transformation matrix multiplied with the others, called as forward kinematics. The forward kinematics equations are given Cartesian space position using set of joint angles. The inverse kinematics equations are represented set of joint angles using Cartesian space position. Therefore, there is always a forward kinematic solution of a robot. However, the inverse kinematic doesn't always give a solution of a robot. In Figure 1.1, it is illustrated relationship between joint space and Cartesian space [1].

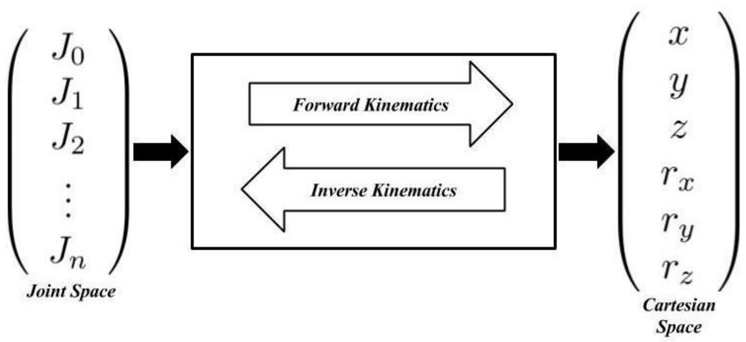

Figure 1.1. The schematic representation of forward and inverse kinematic [1].

The relationship between end-effector and base for a robot manipulator is given in Equation 1.1 as a function of position and orientation.

$$ {}_{end-effector}^{base}T = \begin{bmatrix} r_{11} & r_{21} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.1} $$

r_{ij} indicates rotational elements and the position vector is represented by p_x, p_y and p_z. At first transformation matrix must be multiplied with each other for forward kinematic.

$$ {}_6^0T = {}_1^0T(q_1)\,{}_2^1T(q_2)\,{}_3^2T(q_3)\,{}_4^3T(q_4)\,{}_5^4T(q_5)\,{}_6^5T(q_6) \tag{1.2} $$

Where q_i is the revolute or prismatic joint variable for joint i. To find joint variable q_1, the link transformation inverses are multiplied as follows.

$$ \begin{gathered} [{}_1^0T(q_1)]^{-1}\,{}_6^0T = [{}_1^0T(q_1)]^{-1}\,{}_1^0T(q_1)\,{}_2^1T(q_2)\,{}_3^2T(q_3)\,{}_4^3T(q_4)\,{}_5^4T(q_5)\,{}_6^5T(q_6) \\ [{}_1^0T(q_1)]^{-1}\,{}_6^0T = {}_2^1T(q_2)\,{}_3^2T(q_3)\,{}_4^3T(q_4)\,{}_5^4T(q_5)\,{}_6^5T(q_6) \end{gathered} \tag{1.3} $$

The following equations to find the other joint variable are obtained in the same way,

$$ [{}_1^0T(q_1)\,{}_2^1T(q_2)]^{-1}\,{}_6^0T = {}_3^2T(q_3)\,{}_4^3T(q_4)\,{}_5^4T(q_5)\,{}_6^5T(q_6) \tag{1.4} $$

$$ [{}_1^0T(q_1)\,{}_2^1T(q_2)\,{}_3^2T(q_3)]^{-1}\,{}_6^0T = {}_4^3T(q_4)\,{}_5^4T(q_5)\,{}_6^5T(q_6) \tag{1.5} $$

$$ [{}_1^0T(q_1)\,{}_2^1T(q_2)\,{}_3^2T(q_3)\,{}_4^3T(q_4)]^{-1}\,{}_6^0T = {}_5^4T(q_5)\,{}_6^5T(q_6) \tag{1.6} $$

$$ [{}_1^0T(q_1)\,{}_2^1T(q_2)\,{}_3^2T(q_3)\,{}_4^3T(q_4)\,{}_5^4T(q_5)]^{-1}\,{}_6^0T = {}_6^5T(q_6) \tag{1.7} $$

In Equation 1.4, q_1 can solved as functions of r_{11}, r_{12}………r_{33}, p_x, p_y, p_z and fixed link parameters. q_1 is the only unknown. When q_1 is solved, other joint variable are found same way [1]. It is necessary to classify the robots before the solution starts. The robots are classified and their characteristic features appear in solution.

To classify a robot configuration, two-letter code is developed by B. Huang and V. Milenkovic [2]. First joint and it's relationship to the second joint is identified by the first letter. The second letter characterizes the third joint and third joint's relationship to the second joint. The letter meanings are:

S : Slider

C : Rotary parallel to slider

N : Rotary vertical to rotary

R : Rotary vertical to rotary, or rotary parallel to rotary

The robot manipulators can be separated to four types. In the first type shown in Figure 1.2a, which have three joints are vertical each other, there are five orthogonal robot manipulator (SS, SN, NS, NN and RR). In the second type shown in Figure 1.2b, the first two joints are parallel to each other (CS, CR, RN and CN). In the third type shown in Figure 1.2c, the last two joints are parallel to each other (SC, NR, RC and NC). In the fourth type shown in Figure 1.2d, three joints are parallel to each other (CC, RS and SR). An inverse kinematic drawing is given belong to SN, CS, NR, CC robots of the four groups as an example [1].

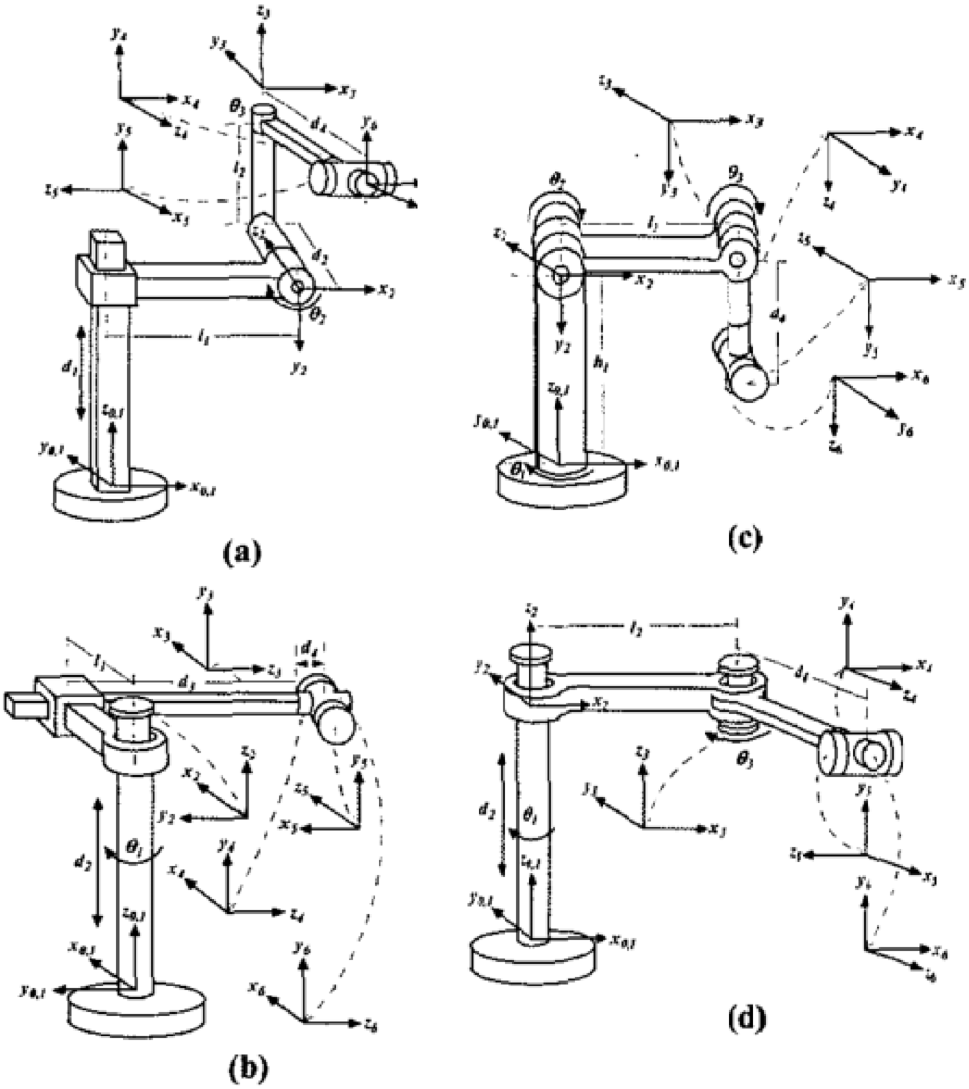

Figure 1.2. a) SN robot manipulator, b) CS robot manipulator, c) NR robot manipulator, d) CC robot manipulator [1].

The inverse kinematics of SN, CS, NR and CC robot were solved. The joint variables for SN robot are given by,

$$d_1 = p_z - \sqrt{d_4^2 + l_2^2 - 2d_2 d_4 c\theta_3 - p_x^2 - l_1^2 + 2p_x l_1 - p_y^2} \tag{1.8}$$

$$\theta_2 = A\tan 2(d_4 s\theta_3, l_2) \pm A\tan 2(\sqrt{l_2^2 + d_4^2 s^2\theta_3 - (p_z - d_1)^2}, p_z - d_1) \tag{1.9}$$

$$\theta_3 = A\tan 2(\pm\sqrt{1-(\frac{d_2 - p_y}{d_4})^2}, \frac{d_2 - p_y}{d_4}) \tag{1.10}$$

$$\theta_4 = A\tan 2([-s\theta_2 r_{13} + c\theta_2 r_{33}]/s\theta_5, [c\theta_2 c\theta_3 r_{13} + s\theta_3 r_{23} + s\theta_2 c\theta_3 r_{33}]/s\theta_5) \tag{1.11}$$

$$\begin{gathered}\theta_5 = A\tan 2(\pm\sqrt{1-(c\theta_2 s\theta_3 r_{13} - c\theta_3 r_{23} + s\theta_2 s\theta_3 r_{33})^2}, c\theta_2 s\theta_3 r_{13} - c\theta_3 r_{23} + \ldots \\ s\theta_2 s\theta_3 r_{33})\end{gathered} \tag{1.12}$$

$$\begin{gathered}\theta_6 = A\tan 2([-c\theta_2 s\theta_3 r_{12} + c\theta_3 r_{22} - s\theta_2 s\theta_3 r_{32}]/s\theta_5, \\ [c\theta_2 s\theta_3 r_{11} - c\theta_3 r_{21} + s\theta_2 s\theta_3 r_{31}]/s\theta_5)\end{gathered} \tag{1.13}$$

CS robot's joint variables are given by,

$$\theta_1 = A\tan 2(p_y, p_x) \pm A\tan 2(\sqrt{p_x^2 + p_y^2 - l_1^2}, l_1) \tag{1.14}$$

$$d_2 = p_z \tag{1.15}$$

$$d_3 = s\theta_1 p_x - c\theta_1 p_y - d_4 \tag{1.16}$$

$$\begin{gathered}\theta_4 = A\tan 2(-r_{33}, -c\theta_1 r_{13} - s\theta_1 r_{23}) \quad \text{or} \\ \theta_4 = A\tan 2(r_{33}, c\theta_1 r_{13} + s\theta_1 r_{23})\end{gathered} \tag{1.17}$$

$$\theta_5 = A\tan 2(c\theta_1 c\theta_4 r_{13} + s\theta_2 c\theta_4 r_{23} + s\theta_4 r_{33}, s\theta_1 r_{13} - c\theta_1 r_{23}) \tag{1.18}$$

$$\begin{gathered}\theta_6 = A\tan 2(-c\theta_1 s\theta_4 r_{11} - s\theta_1 s\theta_4 r_{21} + c\theta_4 r_{33}, \\ -c\theta_1 s\theta_4 r_{12} - s\theta_1 s\theta_4 r_{22} + c\theta_4 r_{32})\end{gathered} \tag{1.19}$$

NR robot's joint variables are given by,

$$\begin{gathered}\theta_1 = A\tan 2(-p_y, -p_x) \quad \text{or} \\ \theta_1 = A\tan 2(p_y, p_x)\end{gathered} \tag{1.20}$$

$$\begin{gathered}\theta_2 = A\tan 2(d_4 c\theta_3, d_4 s\theta_3 + l_2) \\ \pm A\tan 2(\sqrt{d_4^2 + l_2^2 + 2l_2 d_4 s\theta_3 - (c\theta_1 p_x + s\theta_1 p_y)^2}, c\theta_1 p_x + s\theta_1 p_y)\end{gathered} \tag{1.21}$$

$$\theta_3 = A\tan 2(\frac{(c\theta_1 p_x + s\theta_1 p_y)^2 + (p_z - h_1)^2 - d_4^2 - l_2^2}{2d_4 l_2}, \pm\sqrt{1-(\frac{(c\theta_1 p_x + s\theta_1 p_y)^2 + (p_z - h_1)^2 - d_4^2 - l_2^2}{2d_4 l_2})^2}) \tag{1.22}$$

$$\theta_4 = A\tan 2([s\theta_1 r_{13} - c\theta_1 r_{23}]/s\theta_5, [c\theta_1 s\theta_{23} r_{13} + s\theta_1 c\theta_{23} r_{23} + s\theta_{23} r_{33}]/s\theta_5) \tag{1.23}$$

$$\theta_5 = \pm A\tan 2(\sqrt{1-(c\theta_1 s\theta_{23} r_{13} + s\theta_1 s\theta_{23} r_{23} - c\theta_{23} r_{33})^2}, c\theta_1 s\theta_{23} r_{13} + s\theta_1 s\theta_{23} r_{23} - c\theta_{23} r_{33}) \tag{1.24}$$

$$\theta_6 = A\tan 2([c\theta_1 s\theta_{23} r_{12} + s\theta_1 s\theta_{23} r_{22} - c\theta_{23} r_{32}]/s\theta_5, [-c\theta_1 s\theta_{23} r_{11} - s\theta_1 s\theta_{23} r_{21} + c\theta_{23} r_{31}]/s\theta_5) \tag{1.25}$$

CC robot’s joint variables are given by,

$$\theta_1 = A\tan 2(p_y, p_x) \pm A\tan 2(\sqrt{p_x^2 + p_y^2 - (d_4 s\theta_3 + l_2)^2}, d_4 s\theta_3 + l_2) \tag{1.26}$$

$$d_2 = p_z \tag{1.27}$$

$$\theta_3 = A\tan 2(\frac{p_x^2 + p_y^2 - d_4^2 - l_2^2}{2l_2 d_4}, \pm\sqrt{1-(\frac{p_x^2 + p_y^2 - d_4^2 - l_2^2}{2l_2 d_4})^2}) \tag{1.28}$$

$$\theta_4 = A\tan 2(-r_{33}, -c\theta_{13} r_{13} - s\theta_{13} r_{23}) \quad \text{or} \quad \theta_4 = A\tan 2(r_{33}, c\theta_{13} r_{13} + s\theta_{13} r_{23}) \tag{1.29}$$

$$\theta_5 = A\tan 2(-c\theta_4 c\theta_{13} r_{13} - s\theta_{13} c\theta_4 r_{23} - s\theta_4 r_{33}, s\theta_{13} r_{13} - c\theta_{13} r_{23}) \tag{1.30}$$

$$\theta_6 = A\tan 2(-c\theta_{13} s\theta_4 r_{11} - s\theta_{13} s\theta_4 r_{21} + c\theta_4 r_{31}, -c\theta_{13} s\theta_4 r_{12} - s\theta_{13} s\theta_4 r_{22} + c\theta_4 r_{32}) \tag{1.31}$$

The inverse kinematic solutions were defined in closed form. The inverse kinematics equations are executed in the microprocessor [1].

In 2014, Jian Fang, Tao Mei, Jian Chen and Jianghai Zhao conducted a study. It was focused on the inverse kinematics of the redundancy robot. In Figure 1.3, the structure of the redundancy robot was shown. D-H method, the general method, was used for kinematic analysis. D-H parameter was shown in the Table 1.1 [3].

Iteration method was used to calculate the inverse solution. The method’s idea was approximate the target by iteration based on the error between current and target result. Initial parameters for the iteration method which are maximum iterations, iterative error and initial value of joint angle were set. The error between the current pose and target pose was computed. The value of error was compared a given value. When the value of error was greater than a given value, the correction joint angle was competed. An effective iteration method of redundancy robot had been presented. It can be used on the real-time control. Because of short computation time [3].

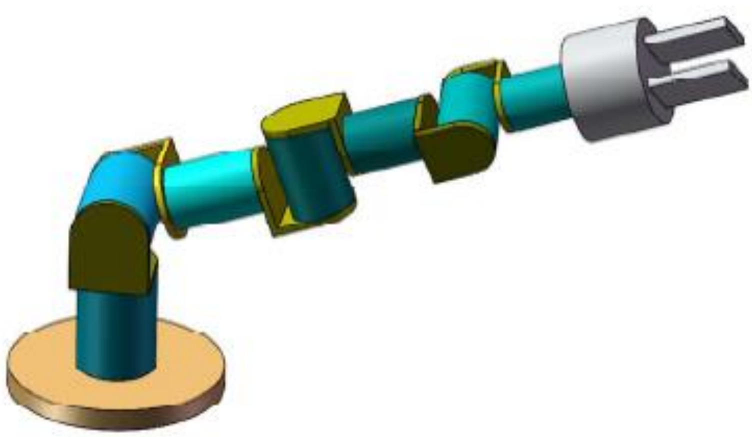

Figure 1.3. The structure of the redundancy robot [3].

Table 1.1. D-H parameters [3].

Link	θ_i	d_i	a_i	α_i
1	θ_1	0	0	$\pi/2$
2	θ_2	0	0	$-\pi/2$
3	θ_3	d_3	0	$\pi/2$
4	θ_4	0	0	$-\pi/2$
5	θ_5	d_5	0	$\pi/2$
6	θ_6	0	0	$-\pi/2$
7	θ_7	d_7	0	0

Vivek Deshpande and P. M. George presented a study in 2014 called as Kinematic Modelling and Analysis of 5 DOF Robotic Arm. The study purposes were modelling a 5 DOF Robotic Arm kinematics which was shown in Figure 1.4 [4].

Table 1.2. D-H parameters of 5 DOF Robotic arm [4].

Joint i	α_i (deg)	a_i (mm)	d_i (mm)	θ_i (deg)
1	0	0	86	θ_1
2	90	0	0	θ_2
3	0	96	0	θ_3
4	0	96	0	θ_4
5	90	0	59.5	θ_5

Figure 1.4. 5 DOF Robotic Arm [4].

D-H method was used to model of robot. This method was found the position and orientation of end-effector. D-H parameters were shown in Table 1.2 [4].

The transformation matrix based on the D-H Table was given by,

$$^{0}T_{1} = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.32}$$

$$^{1}T_{2} = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.33}$$

$$^{2}T_{3} = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 * c_3 \\ s_3 & c_3 & 0 & a_3 * s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.34}$$

$$^{3}T_{4} = \begin{bmatrix} c_4 & -s_4 & 0 & a_4 * c_4 \\ s_4 & c_4 & 0 & a_4 * s_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.35}$$

$$^{4}T_{5} = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.36}$$

where, $s_i=\sin\theta_i$, $c_i=\cos\theta_i$.

Using the forward kinematic matrix, inverse kinematic analysis was solved. The set of joints angles were obtain and were used in Matlab. For the set of joints angles of $\theta_1=30^o$, $\theta_2=50^o$, $\theta_3=45^o$, $\theta_4=25^o$ and $\theta_5=0^o$ result was found $P_x=76.0852$, $P_y=88.8540$ and $P_z=244.0927$ [4].

In this study, the mathematical model for 5 DOF robotic arm was prepared and the position and the orientation parameters were found by in Matlab. The result of forward kinematics were compared with the analytical method of kinematic model and the forward kinematic were verified. 5 DOF robotic arm was ensured as an educational tool [4].

In 2015, Martin Svejda, Tomas Cechura conducted a study. It was focused on the problem trajectory planning of the manipulator end-effector for arc welding and machining and inspecting, etc. There were many algorithms for trajectory planning of manipulator. They were presumed for direct computation of joint coordinates rely on the given initial and finish point. Two interpolation methods which were line interpolation and the new cubic spline interpolation were compared [5].

In the line interpolation, firstly one directional coordinate p was assumed. p^k was desired coincident point and $\dot{p}^k_{\max}$ was maximal velocity and $\ddot{p}^k_{\max}$ was acceleration in the each segment. Each polynomial blending was given as:

$$p^k_a = p_k - T_k \dot{p}^k_{\max} \tag{1.37}$$

In Figure 1.5 line interpolation with polynomial blending was shown. The k-th segment interpolation was given as:

$$\begin{aligned} &p(t) = (t - t_k - T_k)\dot{p}^k_{\max} + p^k \; fort \in \langle t_k + 2T_k, t_{k+1} \rangle \\ &\dot{p}(t) = \dot{p}^k_{\max}, \ddot{p}(t) = 0 \end{aligned} \tag{1.38}$$

For the one dimensional line interpolation with polynomial blending the problem was extended to multidimensional case. The orientation and position of the manipulators were given x, y, z, α, β, γ where x, y, z were position coordinates and α, β, γ were orientation coordinates given by the Euler angles. The result of the trajectory planning algorithm with line interpolation was shown in Figure 1.6 [5].

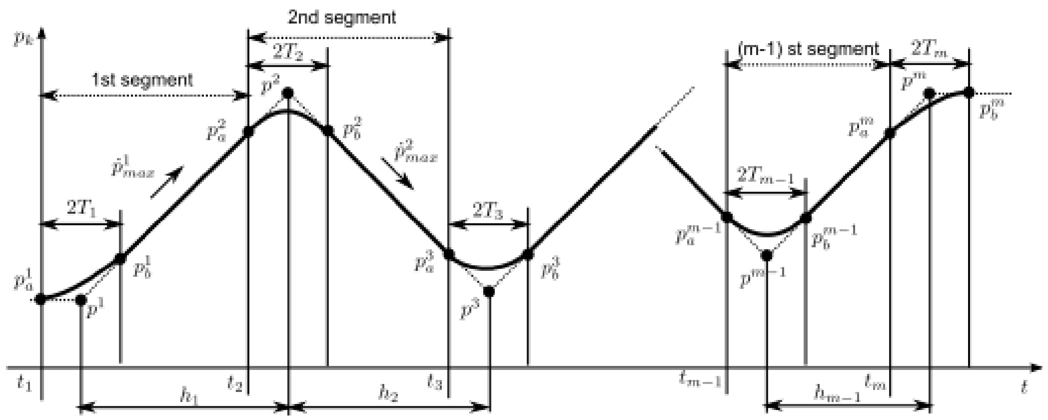

Figure 1.5. Line interpolation with polynomial blending [5].

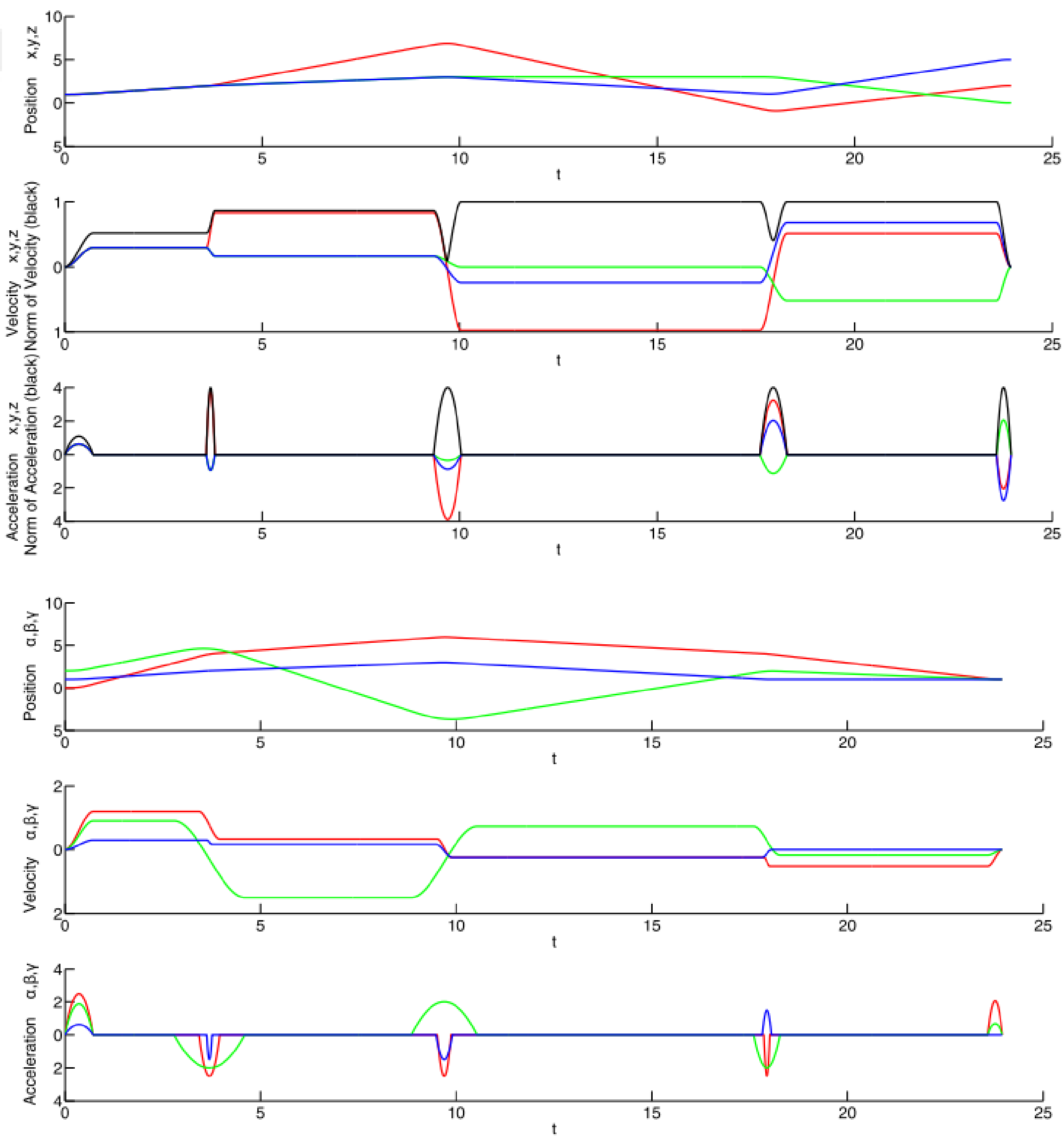

Figure 1.6. The result of the trajectory planning algorithm with line interpolation (x,α - red, y,β - green, z,γ - blue) [5].

The cubic spline interpolation was not suitable for interpolation. There were two problem. First one was that there were peaks in the acceleration and second one, the velocity of Euler angles was not static. For the solution, there were a lot of coincident points. There were many possibilities for choosing the coincident points. But the best choice was only the continuity of position, velocity and acceleration. In Figure 1.7, the cubic interpolation curves for the same coincident points were shown [5].

The result of study, they evaluated experimental measurement and real prototype. The line interpolation with polynomial blending was combined the coincident points. The trajectory was predictable as a result. The cubic spline interpolation was satisfy demanded profile. This method was more suitable for pick and place applications with a few coincident points [5].

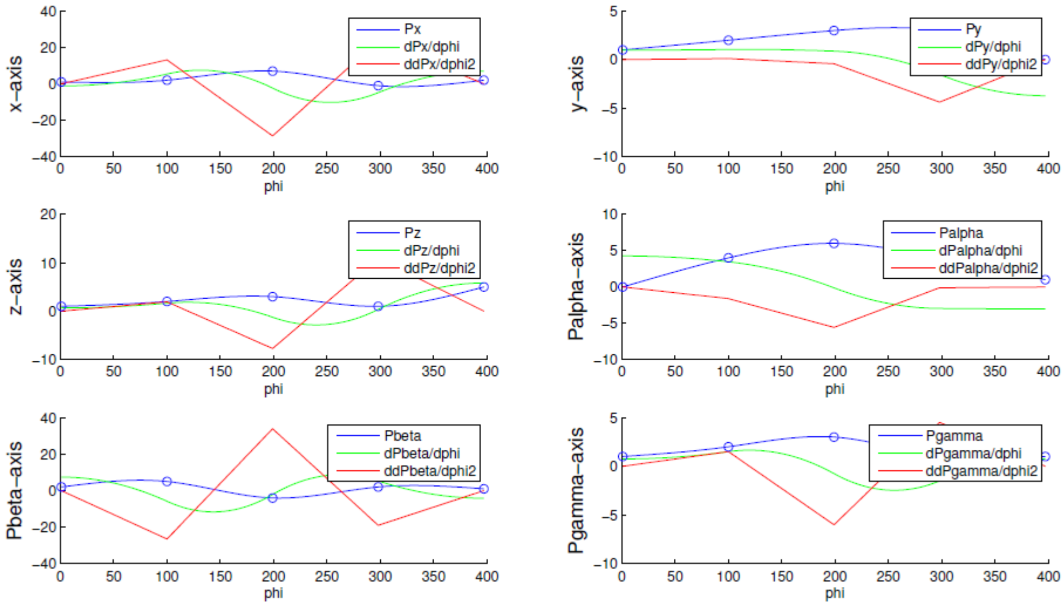

Figure 1.7. The cubic interpolation curves for the same coincident points [5].

Kai Wu, Carsten Krewet, Bernd Kuhlenkötter presented a study called and focused the problem of industrial robot limitations. One of two main solutions was to change the structure of the robot. The other solution was to develop the control system. In this study, the CNC's control system was approached for a robot motion. In Figure 1.8, the general motion control system was shown. The program was generated the planned path X^p. According to inverse mathematical model the control signals Φ_m^p were generated by the trajectory planning. Φ_m^r was measurement for real motion of the motor. The robot's arms rotated and X^r which was the real path was obtained. The robot's performance depended on the joint friction, backlash and elasticity [6].

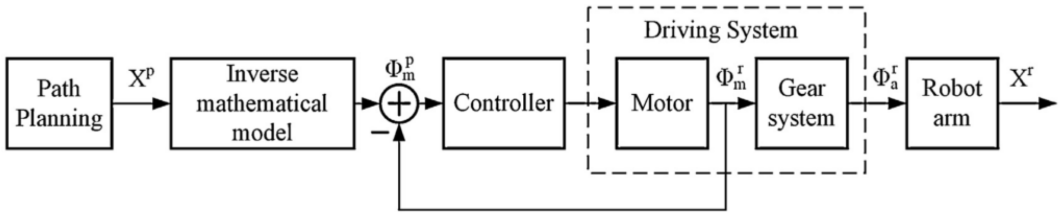

Figure 1.8. General motion control system [6].

The three kinds of acceleration profiles for CNC were step-shaped, trapezoidal and square-sinusoidal. In the acceleration profile, there were four phases which were increase in accelerating, decrease in accelerating, increase in braking and decrease in braking. The corner path in the CNC program was describe as a polynomial. The parameters in the corner path are shown Figure 1.9. a was the distance of start to corner. b was the distance of end to corner. e was deflection of corner. Both of a and b could change according to program [6].

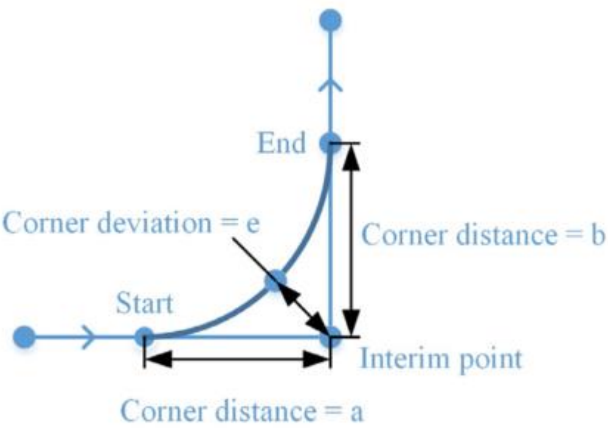

Figure 1.9. The corner path [6].

In this study, two differences program KLR and CNC were comprised. After the experimental studies, three paths are programmed at low and high velocity. In Figure 1.10, the position distribution in the X-Y plane was given. It was shown that the KLR has a bigger corner deviation than CNC. Two types of paths were Profile 0 (step-shaped) and Profile 1 (trapezoidal). While the velocity increased from 30 mm/s to 200 mm/s, the corner deviation of the KLR path raised 0.85 mm. But the corner deviation of CNC increased by 0.28 mm. The velocity has a little influence in the CNC compared to KLR on the corner deviation [6].

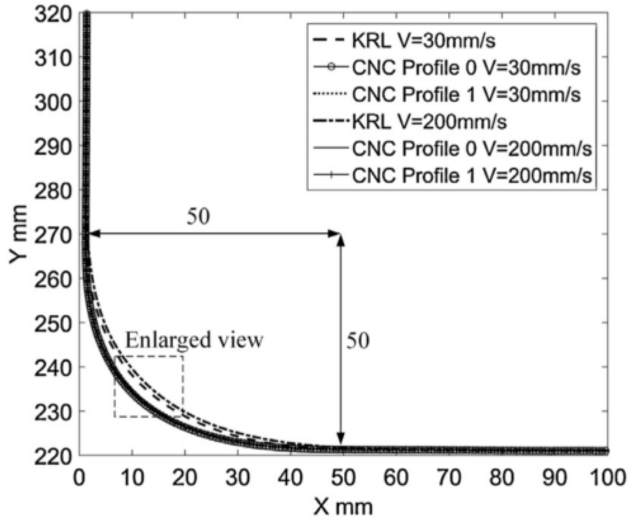

Figure 1.10. Distribution of X-Y [6].

In Figure 1.11, acceleration distribution of corner path was shown. The KLR path has much higher acceleration value than the CNC path. According to two types of velocity, comparing the acceleration the value at velocity 200 mm/s nearly has similar the value at a velocity of 30 mm/s. But the absolute value of KRL path was much higher than CNC path [6].

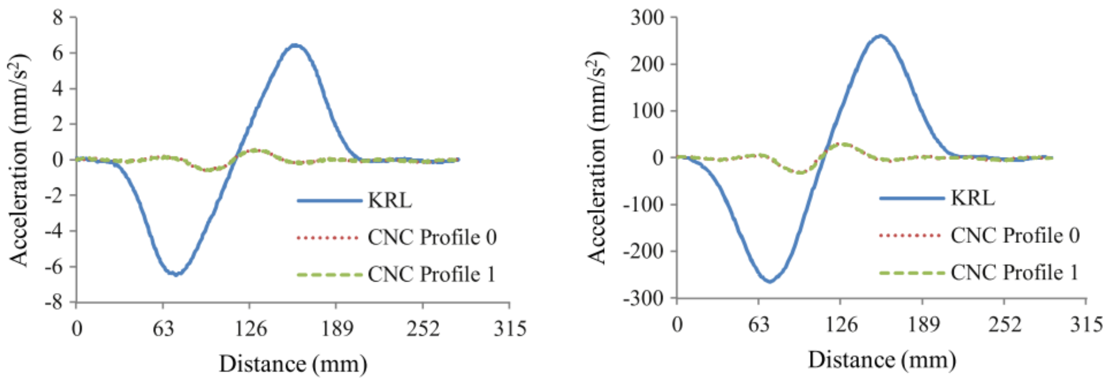

Figure 1.11. Acceleration distribution of corner path [6].

In 2016, Xing Jin, Junfeng Kang, Jingjing Zhang and Xiang Yang performed a study called Trajectory Planning of a Six-DOF Robot Based on a Hybrid Optimization Algorithm. Robot’s trajectory planning affected the accuracy and stability. There were two trajectory planning methods: Cartesian space planning and joint space planning. The joint space planning was used for tasks without path requirements. However, Cartesian space planning was predictable and visual [7].

The ant colony algorithm, particle swarm optimization (PSO) and Genetic Algorithm (GA) was used for trajectory planning. In this study, a hybrid algorithm which was a combination PSO, the ant colony algorithm and GA was used. The PUMA 560 spot welding robot was used to create trajectory planning in the shortest path in Cartesian space for experimental study [7].

The ant colony algorithm was an optimization algorithm that behaves of the smart ant colony. A positive feedback mechanism was an advantage for the ant colony algorithm. The slow solving speed, the lack of an initial pheromone and high risk of falling into the local optimal solution were disadvantage of it. GA was a coincidentally optimization algorithm. GA exposed exact parallel search performance and high global impact. Meanwhile, it had slow search speed for the late period. PSO was a swarm intelligence algorithm. The speed convergence, easy parameter regulation and easy combination with another algorithm were the advantages of PSO. Its disadvantages were bad local search ability and unsuccess to fully utilized feedback [7].

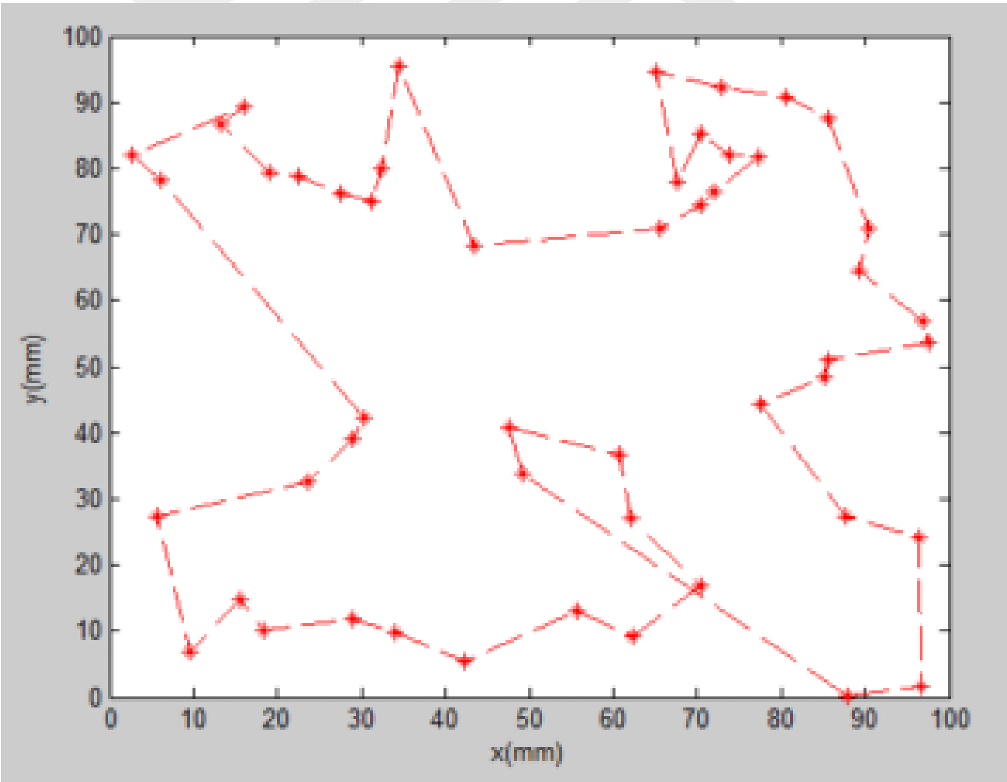

Figure 1.12. Trajectory planning for optimal path with the ant colony algorithm [7].

In the experimental study, the ant colony algorithm which was include PSO and GA factors was used. The advantages of PSO, which was speed convergence for early period, and the advantages of GA, which was good global optimization impact, were combined into the ant colony algorithm. To obtain the shortest path, new hybrid algorithm was used with a six-DOF spot welding robot [7].

In Figure 1.12, it was shown the trajectory planning which was exist crosses for optimal path with the ant colony algorithm. This algorithm remarked better solution. But the hybrid algorithm which was shown in Figure 1.13 was better than the ant colony algorithm [7].

To prevent from occasionality, both of algorithms executed 1000 times each. The global best values obtained for 1000 times [7].

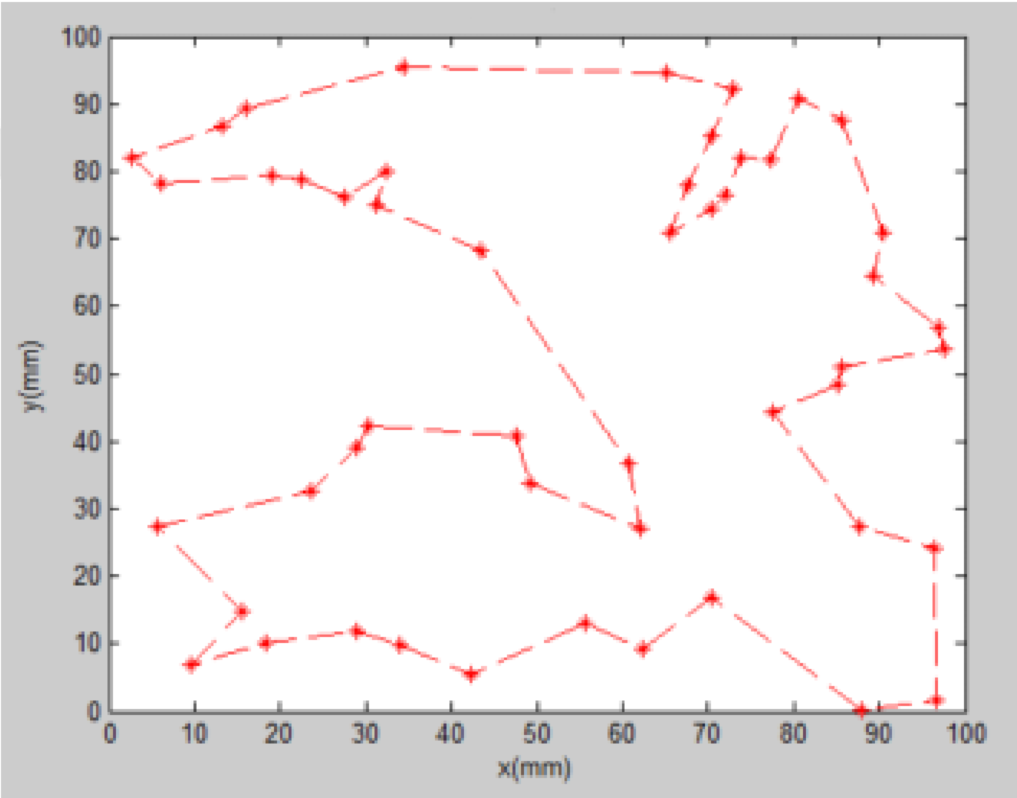

Figure 1.13. Trajectory planning for the hybrid algorithm [7].

The global best mean value of the hybrid algorithm as shown in Figure 1.14 was littler than that of the ant colony algorithm. This means that the hybrid algorithm's performance was better than the ant colony algorithm's performance [7].

In the result, the initial solving rapid and the optimal solution quality of the hybrid algorithm were better than the standard ant colony algorithm's. So, this study was proved the efficiency of the hybrid algorithm [7].

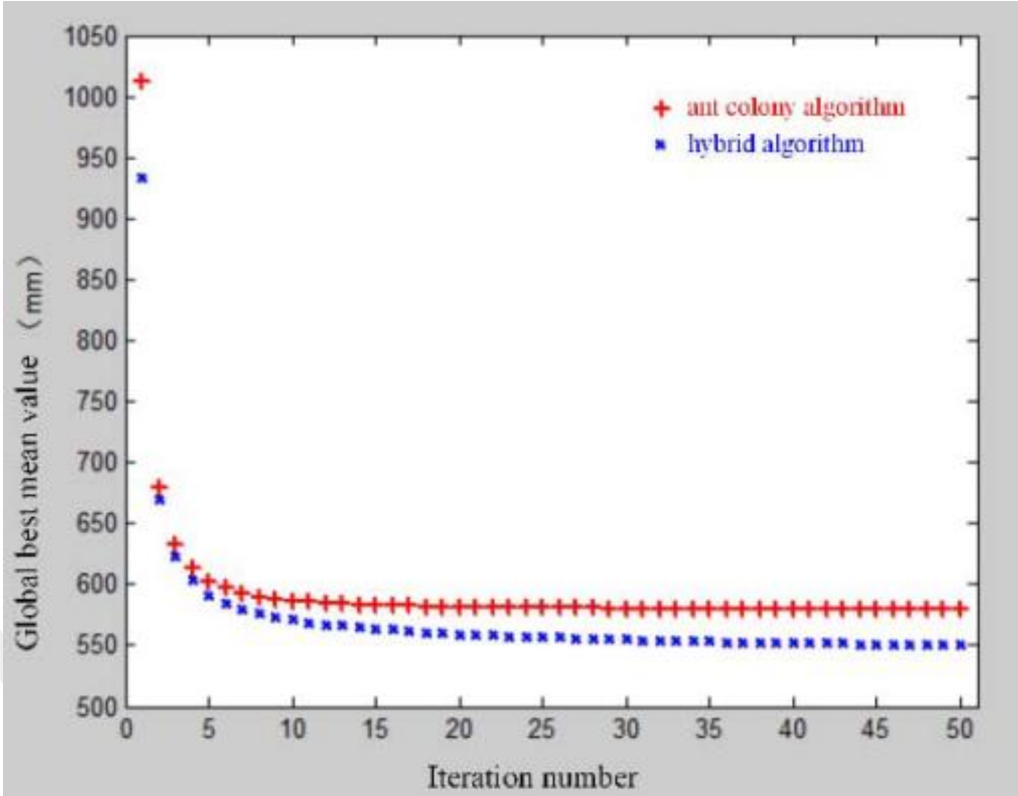

Figure 1.14. The global best mean value of both of algorithm [7].

In 2019, Liu Yan, Liu Ya, Tian Xincheng focused on trajectory and velocity planning method for robot to machine a spherical single Y-groove. The quality of the welding depends on the preparations before welding and the surfaces to be welded. The grooves commonly used when surface welding are V-groove, J-groove, Y-groove and so on. The best surface treatment technology for the sphere-pipe intersection joints is the single Y-groove. In Figure 1.15, it was shown the model of sphere-pipe intersection [8].

The parameter equations of intersecting curve were defined and mathematical model was introduced. An industrial robot arm was used as a cutting machine by mounting a cutting torch, and the trajectory of the robot was defined according to the coordinate system of the cutting torch. In cartesian space, position and orientation information is included in trajectory planning. There are multiple methods to define the robot trajectory: these are the position + rotation angle, the homogeneous transformation matrix and so on. Using the homogeneous transformation method R_x, R_y and R_z rotation angles of the robot were found [8].

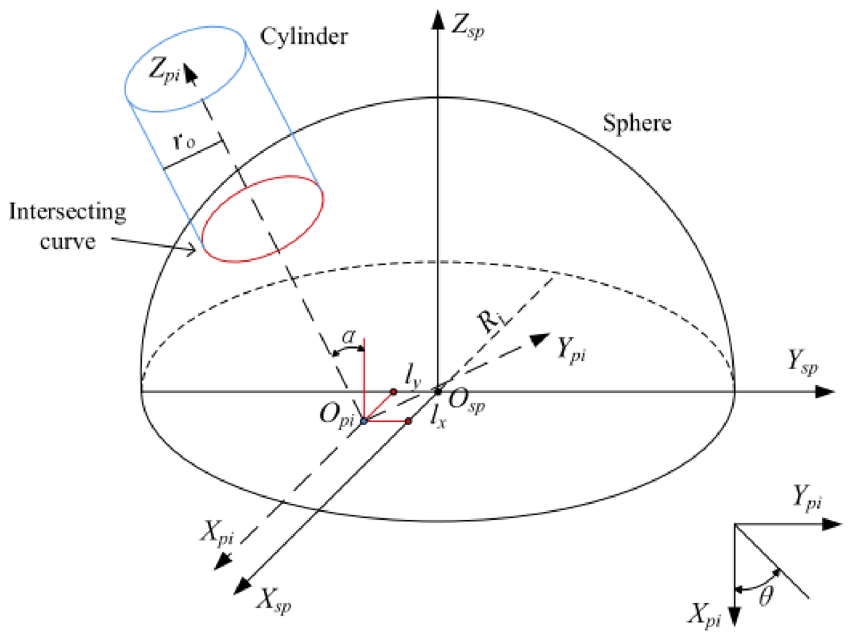

Figure 1.15. The model of sphere-pipe intersection [8].

The fluctuations in the cutting speed affect stability and controllability. It that cause significant losses in the mechanical tools disrupt the accuracy of the single Y-groove. The robot controller was interpolated using two speed planning methods. The first is the interpolation speed of the robot controller and the second is the joint speed of robot [8].

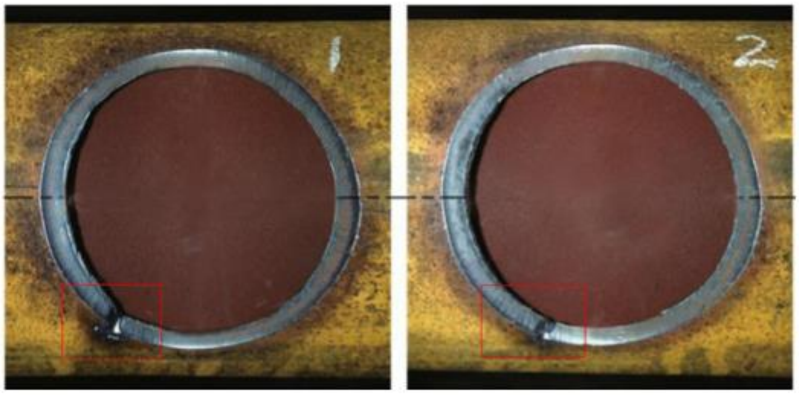

Figure 1.16. The plasma cutting faults [8].

The plasma processing is the one of the methods of single Y-groove. The faults of plasma cutting were shown in Figure 1.16. The plasma cutting has advantages: cutting speed is faster; thermal deformation is minor; the production efficiency is upgrade. Nevertheless, the arc voltage of the plasma head at the start of the arc is unstable and often causes a cutting gap [8].

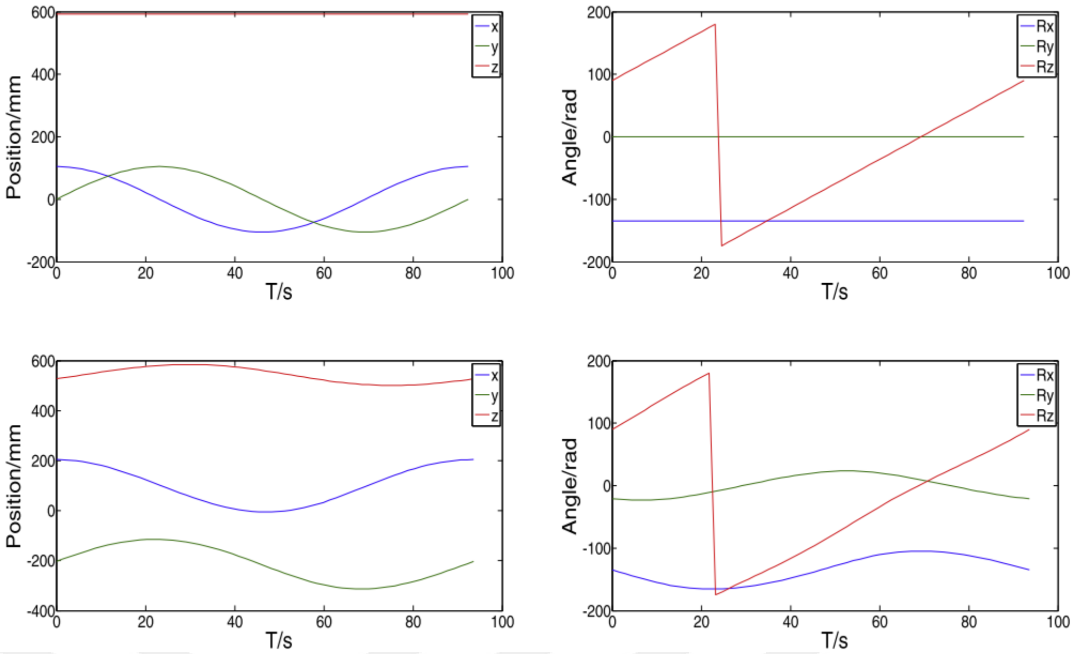

Figure 1.17. The curves of the trajectory in simulation [8].

The position and angle graphs of the defined trajectory are given in Figure 1.17. As a result of the simulations, it was observed that the proposed sphere-pipe intersection model coincides with the structures commonly used in real operations and consolidates the main characteristic of the groove features. The mentioned model was analyzed using geometric coordinate systems. The relationship between coordinate systems is expressed by homogeneous transformation matrices. With the application, it was possible to achieve functional modularity. The final homogeneous transformation matrix obtained from the study can be applied to many industrial robots and their program rules. Interpolation was performed using two different Tool Center Point (TCP) speeds and the stability of the speed was shown by simulation. In the plasma cutting process, initially the arc voltage is unstable and therefore the cutting gap is formed. As a result, the initial adjustment algorithm for the groove cutting model increases accuracy and prevents cutting gaps [8].

David A. Duque, Flavio A. Prieto and Jose G. Hoyos conducted a study that called trajectory generation for robotic assembly operations using learning by demonstration in 2019. The purpose of the study is to make trajectory planning for the robots used in assembly processes using different programming techniques. Learning by Demonstration (LbD) technique based on Task Parametrized Gaussian Mixture Model (TP-GMM) and Petri nets models was used in the execution of the robot assembly processes. The trajectory production consists of 3 stages. In the first stage, the trajectory coordinates (x, y, z) must be shown by a teacher. In the second stage, the training of the TP-GMM model was carried out. In the third stage, the trajectory was produced depending on the location of the objects [9].

Each demonstration was pre-treated with Moving average filter (MAF) and Data elimination to ensure uniformity and smoothness of movement. Hand occlusions occur due to assembly operation and lighting. Moving average filter reduces the diversity in the data consisting of hand occlusions. As a result of the filtering process, satisfactory smoothness was obtained. In Data elimination, while obtaining the t point, more than 20 percent of the points obtained from the previous point are discarded. This is because the image segmentation is not correct and shows the position of the hand differently from the actual position. As a result of the data elimination process, a more constant speed was obtained [9].

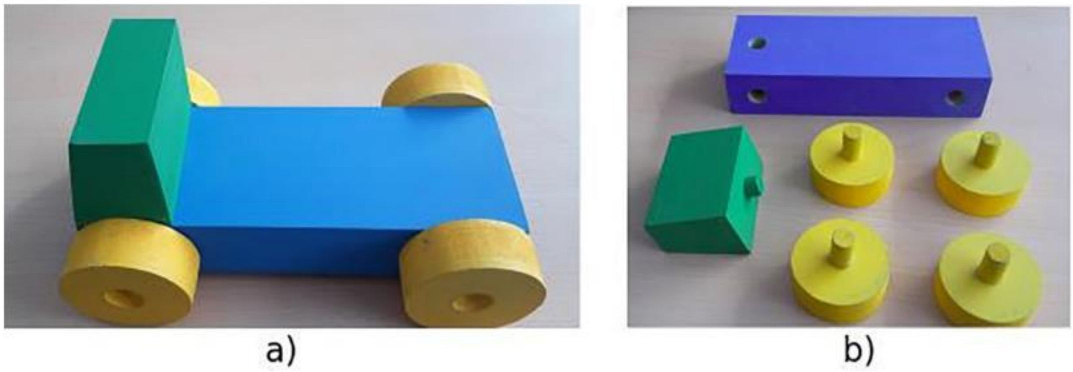

Figure 1.18. a) Car to be assembled. b) Parts to be assembled [9].

A demonstration was carried out using a Kinetic motion sensor. The assembly of the model vehicle given in Figure 1.18 was made by the methodology described. Assembly process was carried out by one person and detected by Kinetic motion sensor. The demonstration was held in a laboratory environment where conditions such as light were controlled. Cubic interpolation was made in trajectory generation. The filtered trajectory obtained in Figure 1.19 is shown. The trajectory acquired is shown in orange, while the trajectory filtered is shown in blue. It was used to train the TP-GMM model given from each demonstration [9].

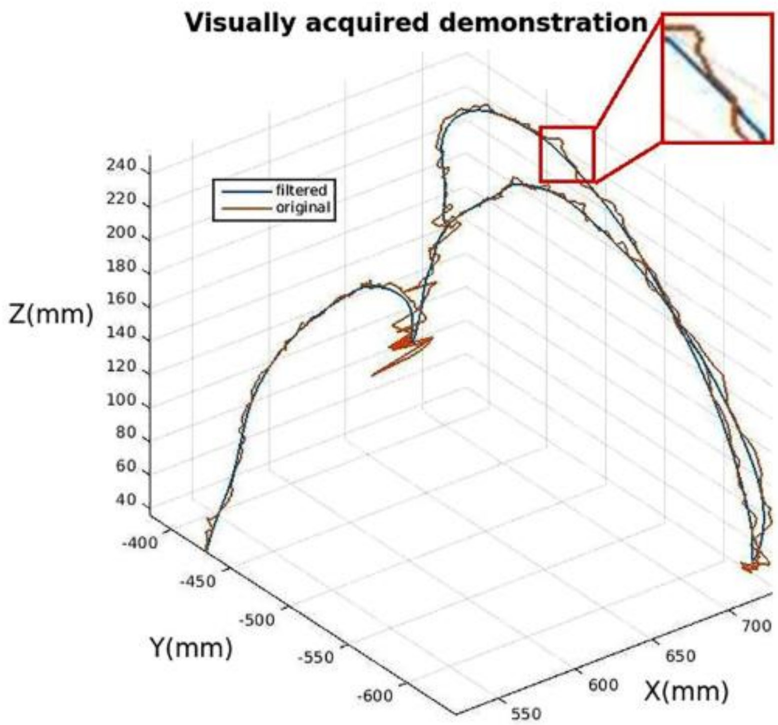

Figure 1.19. Trajectory filtering [9].

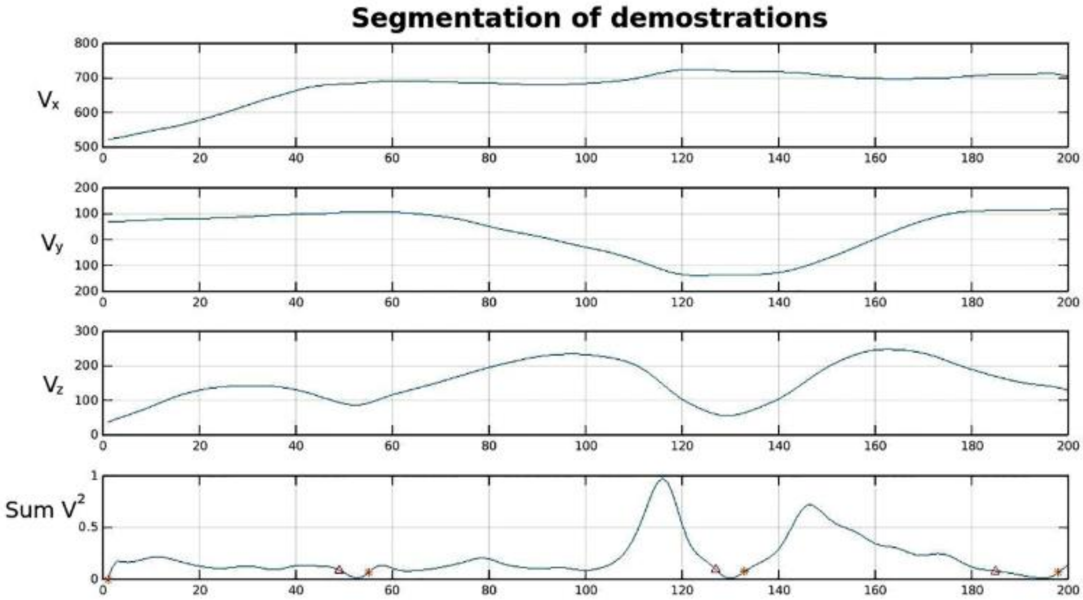

Figure 1.20. The vector components of the velocity [9].

By evaluating the filtered trajectories, a well-defined task for segmentation was determined and moments when the speed was close to zero were selected. Figure 1.20 shows the vector components of the velocity and the total velocity graph for one of the demonstration [9].

As a result of this study, a machine vision supported trajectory generation system based on learning from GMM and TP-GMM models was developed. The algorithm, trained on the position of objects, successfully generates trajectory when there is no significant change in positions. The errors that occur while producing new trajectories offer important values for training the algorithm. It is possible to move the simulation software produced as a result of the study to ABB robots [9].

In 2018, in the study named "Reverse-driving Trajectory Planning and Simulation of Joint Robot", the researchers made trajectory planning of a joint robot with the reverse-driving method. In conventional trajectory planning method, inverse kinematics solutions should be made after forward kinematics. These kinematic analyses consist of difficult-to-solve kinematic equations, and sometimes there is more than one solution, while sometimes no solution can be found. In this study, kinematics and dynamical simulations were performed with the reverse-driving method for the robot whose virtual prototype model was shown in Figure 1.21 [10].

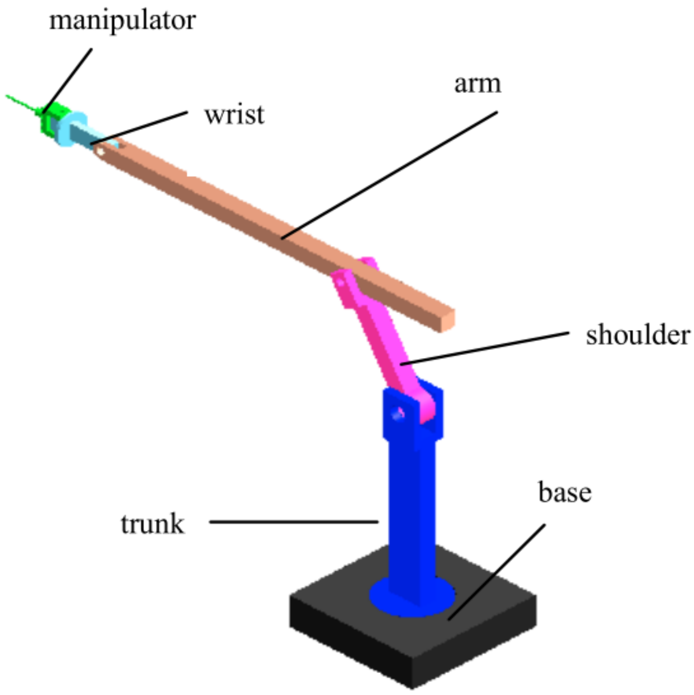

Figure 1.21. The virtual prototype model of joint robot [10].

The trajectory determined for trajectory planning is given in Figure 1.22. Displacement data and time of three directions data has been extracted as data point from the curve in the figure. It was then transferred to the ADAMS program and the robot was moved in the specified orbit [10].

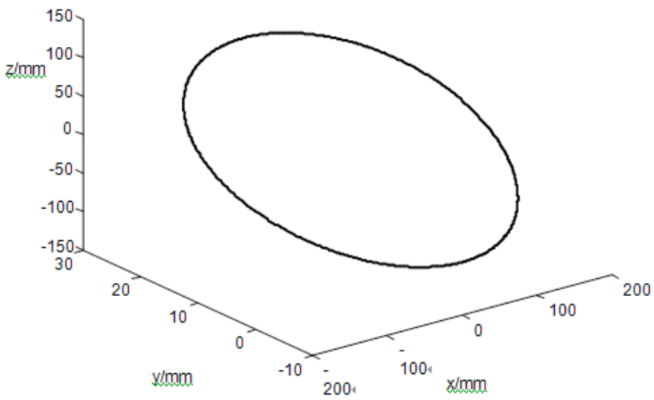

Figure 1.22. The desired trajectory [10].

In Figure 1.23, angular displacement curves are given for each joint of the robot driven using the Akima spline. The curves in the graphs appear continuous and smooth. With the method presented in this study, each joint can be driven quickly and realizes the desired trajectory correctly [10].

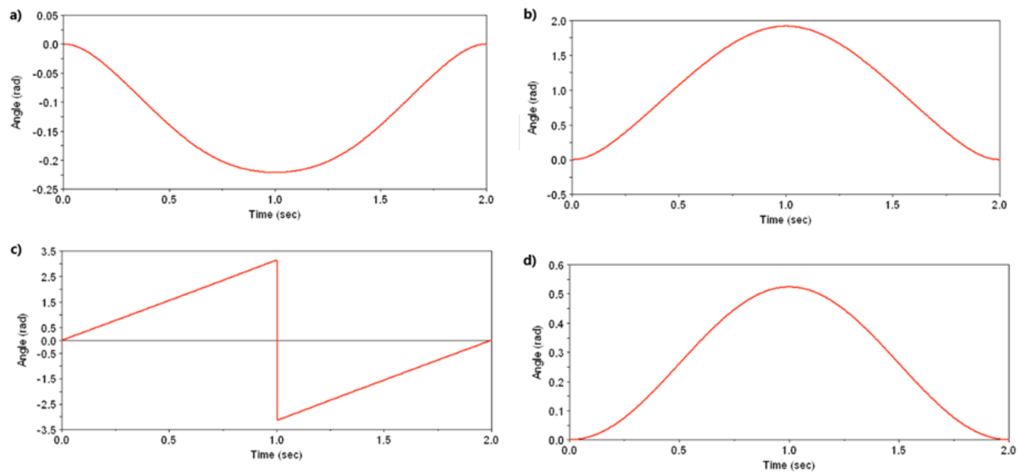

Figure 1.23. a) Trunk and arm joint angular displacement, b) Manipulator and arm joint angular displacement, c) Trunk and base joint angular displacement, d) Manipulator and wrist joint angular displacement [10].

2. NUMERICAL ANALYSIS

Today, it is preferred to use robots with six DOF which are fast and flexible in the industry. In industrial robots with six joints, the last three joints are identical, but the first three joints are different. The joint properties of the first three joints are important to classify the robots according to DOF. The reason for this is that the last three joints of the robots used in the industry are the Euler wrist structure [11]. The Euler wrist structure resembles the structure of the human wrist. It consists of three joints and joint shift and link length. Since the kinematic solution of the Euler wrist structures is the same, the characteristics of the first three joints are taken into account when performing kinematic analysis in industrial robots. In this study, the first three joint structures that characterize industrial robots will be examined. Joints are rotary joints as in industrial rotary robots. Euler wrist structures can be added the robot which has three joints to create industrial robots with six DOF. In the path planning, the speed and position data of each joint will be obtained. Forward and inverse kinematic analyzes are carried out before starting the path planning. The data is processed and presented with an interface in Matlab.

2.1. The Robot Configuration

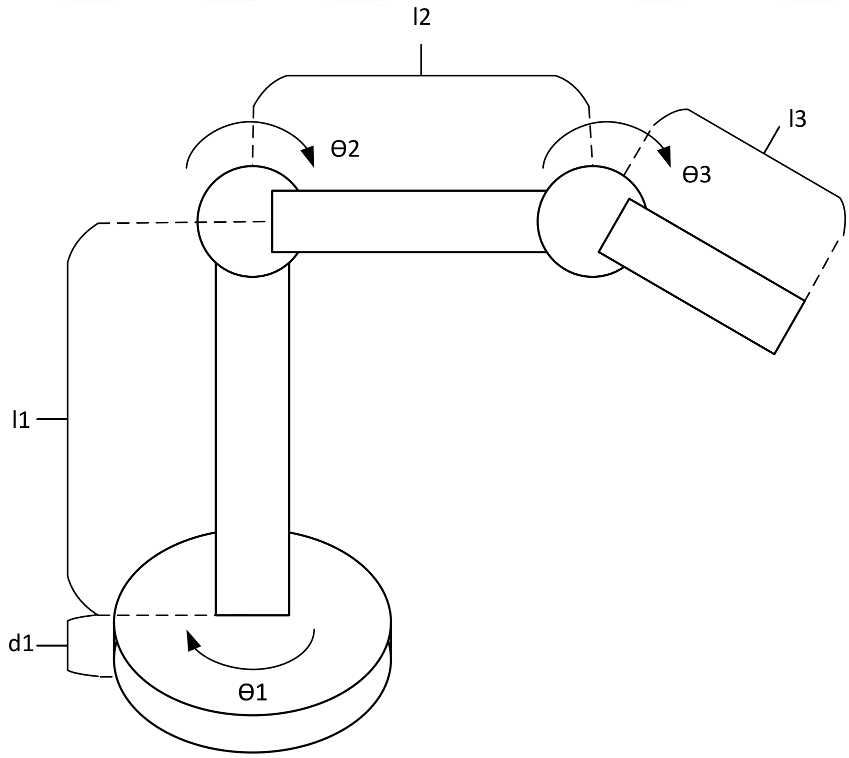

Figure 2.1. The robot configuration.

The robot configuration in this study, three rotary joints are used. This configuration is the basis of most industrial robots. Because these robots have a huge working space and they are very flexible and fast. But they can not reach every point in the working space and they have very complicated kinematic equations. The three rotary joint is preferred in the robot configuration which is shown in Figure 2.1.

2.2. The Forward Kinematic

Forward kinematics deals with the relationship of the positions, velocities, and accelerations between of robot ties. A robot consists of serial links pinned to each other by prismatic or rotary joints from the base frame to the tool frame. If the coordinate system insert in each joint, the relationship between the two adjacent joints is represented by a ${}^{i-1}T_i$ which called transformation matrix. The relationship between the main frame and the tool frame is defined by multiplying the transformation matrices. This association is called forward kinematic and expresses the orientation and position of the tool frame relative to the base frame. For the forward kinematic solution, joint variables should be determined and the Denavit-Hartenberg method is usually used for this [11].

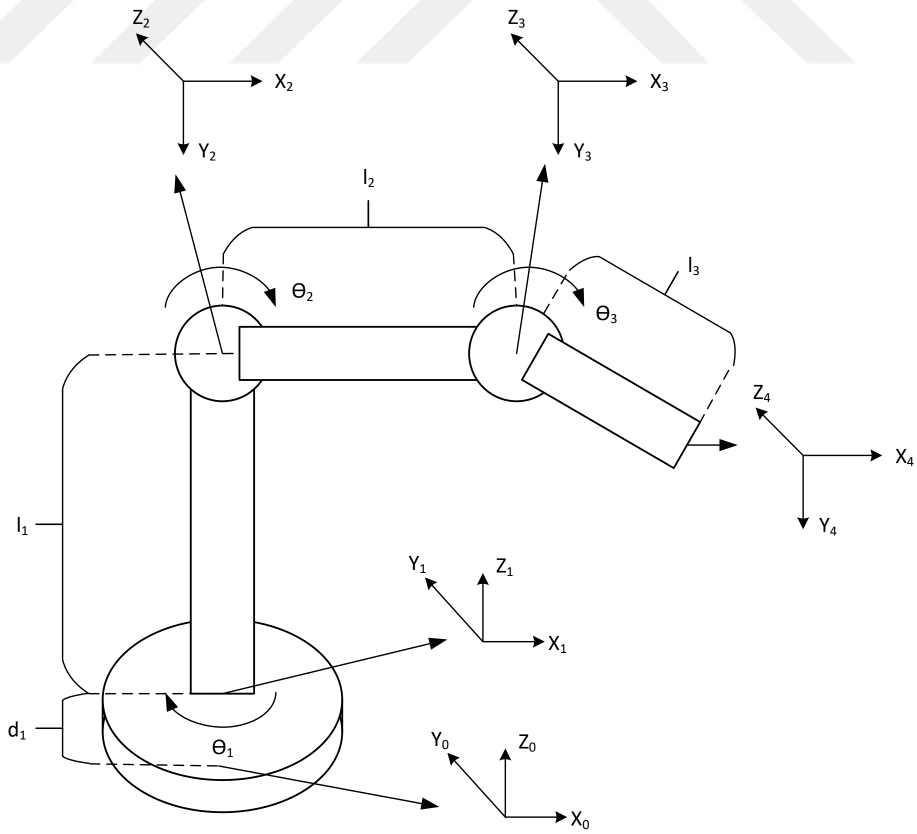

Figure 2.2. The coordinate system on each joint.

D-H method uses four main variables. These variables are the bond length between two axes (a_{i-1}), the bond angle between axes (i-1) and i (α_{i-1}), the joint misalignment between overlapping bonds (d_i), and the joint angle between two bonds θ_i . The robot, which is analyzed and configured in section 2.1, is shown in Figure 2.2 where the coordinate system is placed on each joint. The D-H variables of the robot are shown in Table 2.1.

Table 2.1. D-H variables for the robot.

Number of axes	D-H variables				Joint variables
i	α_{i-1}	a_{i-1}	d_i	θ_i	d_i or θ_i
1	0	0	d_1+ l_1	θ_1	θ_1
2	-90	0	0	θ_2	θ_2
3	0	l_2	0	θ_3	θ_3
4	0	l_3	0	0	0

The transformation matrix of a joint of the robot is obtained with the joint variables by using Equation 2.1.

$$^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

The transformation matrices for each joint which obtained using D-H variables are given below.

$$^{0}T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

$$^{1}T_2 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

$$^{2}T_{3}=\begin{bmatrix} c_3 & -s_3 & 0 & l_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

$$^{3}T_{4}=\begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

By multiplying the transformation matrices of each joint, a general transformation matrix is obtained.

$$\begin{gathered} {}^{0}_{4}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T \\ {}^{0}_{4}T=\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{gathered} \tag{2.6}$$

In the general transformation matrix there are 9 rotation (r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}) and 3 position (p_x, p_y, p_z) specifying elements.

2.3. The Inverse Kinematic

When the joint angles are given, finding the position and orientation of the end effector of a robot manipulator in Cartesian space is called a forward kinematic problem [12]. There is always a solution for the forward kinematic problem which is very easy compared to the inverse kinematic problem. The inverse kinematic problem is defined as the existence of joint variables with the help of position and orientation data of the end effector given to the base frame in Cartesian space [13]. The characteristics of inverse kinematic problems are given below [11].

- It contains complex and nonlinear equations.
- As the number of rotational joints increases, the solution becomes increasingly difficult.
- The mathematical solution does not always happen physically.
- There are more than one solution to reach a point.

The inverse kinematic problem can be solved completely analytically or solved using numerical methods where analytical solution is not possible. Due to the slower performance of numerical analysis than analytical solutions in computer, robot designers usually focus on designs where analytical solution is possible. Today, industrial robots are often produced with simple structures that can be solved analytically [11].

By using analytical solution approach, the inverse kinematic solution of the robot used in the study has been realized. The general transformation matrix is symbolically written as in Equation 2.6. In Equation 2.6, both sides of the equation are multiplied by $\left[{}^{0}_{1}T\right]^{-1}$.

$$\begin{aligned} {}^{0}_{4}T\left[{}^{0}_{1}T\right]^{-1} &= \left[{}^{0}_{1}T\right]^{-1}\,{}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T \\ {}^{0}_{4}T\left[{}^{0}_{1}T\right]^{-1} &= {}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T \end{aligned} \tag{2.7}$$

When you perform the operation on the left side of the equation:

$$\begin{aligned} {}^{0}_{4}T\left[{}^{0}_{1}T\right]^{-1} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & -d_1-l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \dots \\ \begin{bmatrix} . & c_1 * p_x + s_1 * p_y \\ . & -s_1 * p_x + c_1 * p_y \\ . & p_z - d_1 - l_1 \\ . & 1 \end{bmatrix} \end{aligned} \tag{2.8}$$

When you perform the operation on the right side of the equation:

$${}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T = \begin{bmatrix} . & . & . & l_2 * c_{23} + l_2 * c_2 \\ . & . & . & 0 \\ . & . & . & -l_3 * s_{23} - l_2 * s_2 \\ . & . & . & 1 \end{bmatrix} \tag{2.9}$$

Where c_{23} is $c_2 * c_3 - s_2 * s_3$ and s_{23} is $s_2 * c_3 + s_3 * c_2$.

Equation 2.10 is obtained when Equation 2.7 is rewritten.

$$
{}_4^0T\left[{}_1^0T\right]^{-1} = {}_2^1T\,{}_3^2T\,{}_4^3T
$$

$$
\begin{bmatrix} . & . & . & c_1 * p_x + s_1 * p_y \\ . & . & . & -s_1 * p_x + c_1 * p_y \\ . & . & . & p_z - d_1 - l_1 \\ . & . & . & 1 \end{bmatrix} = \begin{bmatrix} . & . & . & l_2 * c_{23} + l_2 * c_2 \\ . & . & . & 0 \\ . & . & . & -l_3 * s_{23} - l_2 * s_2 \\ . & . & . & 1 \end{bmatrix} \tag{2.10}
$$

Equations 2.11, 2.12, 2.13 are obtained if the matrix elements are equalized in equation 2.10.

$$
c_1 * p_x + s_1 * p_y = l_2 * c_{23} + l_2 * c_2 \tag{2.11}
$$

$$
-s_1 * p_x + c_1 * p_y = 0 \tag{2.12}
$$

$$
p_z - d_1 - l_1 = -l_3 * s_{23} - l_2 * s_2 \tag{2.13}
$$

The following trigonometric equations which is used an inverse kinematic solution is given below.

$$
\cos(\theta) = a \Rightarrow \theta = A\tan 2(\mp\sqrt{1-a^2}, a) \tag{2.14}
$$

$$
a\sin(\theta) + b\cos(\theta) = c \Rightarrow \theta = A\tan 2(a,b) + A\tan 2(\mp\sqrt{a^2+b^2-c^2}, c) \tag{2.15}
$$

If Equation 2.12 is similar to Equation 2.15, θ_1 which is first joint variable is obtained.

$$
\theta_1 = A\tan 2(-p_x, p_y) + A\tan 2(\mp\sqrt{p_x^2 + p_y^2}, 0) \tag{2.16}
$$

If we take the squares of Equations 2.11, 2.12 and 2.13 together and make the necessary simplifications, Equation 2.17 is obtained.

$$
\begin{gathered}
(p_x^2 + p_y^2)(s_1^2 + c_1^2) + (p_z - d_1 - l_1)^2 = l_3^2(c_{23}^2 + s_{23}^2) + l_2^2(c_2^2 + s_2^2) + 2l_3l_2(c_2c_{23} + s_2s_{23}) \\
\cos(\theta_3) = \frac{p_x^2 + p_y^2 + (p_z - d_1 - l_1)^2 - l_3^2 - l_2^2}{2l_3l_2}
\end{gathered} \tag{2.17}
$$

Using Equation 2.14, Equation 2.17 is solved and Equation 2.18 is obtained. And θ_3 is third joint variable.

$$
\begin{gathered}
\theta_3 = A\tan 2(\mp\sqrt{1-\left(\frac{p_x^2 + p_y^2 + (p_z - d_1 - l_1)^2 - l_3^2 - l_2^2}{2l_3l_2}\right)^2}, \ldots \\
\frac{p_x^2 + p_y^2 + (p_z - d_1 - l_1)^2 - l_3^2 - l_2^2}{2l_3l_2})
\end{gathered} \tag{2.18}
$$

When Equation 2.13 is solved, the second joint variable is found.

$$\theta_2 = A\tan 2(-l_3\cos(\theta_3) - l_2, -l_3\sin(\theta_3)) \mp \ldots$$
$$A\tan 2(\sqrt{l_3^2 + l_2^2 + 2l_3l_2\cos(\theta_3) - (p_z - d_1 - l_1)^2}, p_z - d_1 - l_1) \quad (2.19)$$

After finding the joint variables, it appears that there are eight solution clusters. One or more than of the solution clusters may be true or none may be true. To test the correctness of solution sets, joint variables can be substituted in the forward kinematic position vector. The forward kinematics position vector gives in Equation 2.20.

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} l_3\,c_{23}\,c_1 + l_2\,c_2\,c_1 \\ l_3\,c_{23}\,s_1 + l_2\,c_2\,s_1 \\ -l_3s_{23} - l_2s_2 + d_1 + l_1 \end{bmatrix} \quad (2.20)$$

2.4. Path Planning in the Joint Space

Three- or higher-order polynomials are used when trajectory planning is performed in joint space. The initial and target positions of the end effector are found by applying inverse kinematics. If the end effectors velocities at the start and end is assumed to be zero, then the conditions necessary to create a third order polynomial which is shown in Equation 2.22 are obtained. The initial conditions are given in Equation 2.21.

$$\begin{aligned} \theta(0) &= \theta_0 \\ \theta(t_f) &= \theta_f \\ \dot{\theta}(0) &= 0 \\ \dot{\theta}(t_f) &= 0 \end{aligned} \quad (2.21)$$

$$\theta(t) = s_0 + s_1t + s_2t^2 + s_3t^3 \quad (2.22)$$

The joint velocities and accelerations are obtained with taking the first and second derivatives of the path polynomial. And they are shown in Equation 2.23 and 2.24.

$$\dot{\theta}(t) = s_1 + 2s_2t + 3s_3t^2 \quad (2.23)$$

$$\ddot{\theta}(t) = 2s_2 + 6s_3t \quad (2.24)$$

The initial conditions are replaced by position, velocity, and acceleration polynomials to determine the coefficients of the path polynomial. The equations for the coefficients of the path polynomial are given below.

$$s_0 = \theta_0 \quad (2.25)$$

$$s_1 = 0 \tag{2.26}$$

$$s_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) \tag{2.27}$$

$$s_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) \tag{2.28}$$

The path planning for the robot which shown in Figure 2.1 made in Matlab. The joint angles was found required to move the robot to the desired point in the Cartesian space with the inverse kinematic analysis. As an example, a movement has been performed and position graphics given in joint space. The start and target positions is given with joint variables in Table 2.2.

Table 2.2. The start and target position and joint variables.

Position	X (cm)	Y (cm)	Z (cm)	θ_1 (deg)	θ_2 (deg)	θ_3 (deg)
I	6	9	7	34.6066	-30.1607	47.8226
II	6	9	10	34.6066	-30.5239	13.7391

The path equation for the motion of the first joint is given below.

$$\theta(t) = 34.6066 + 0t + 0t^2 + 0t^3 \tag{2.29}$$

The path polynomial of the second joint is given in Equation 2.30.

$$\theta(t) = -30.1607 + 0t - 0.2724t^2 + 0.0908t^3 \tag{2.30}$$

Finally, the path polynomial of the third joint is given in Equation 2.31.

$$\theta(t) = 47.8226 + 0t - 25.5701t^2 + 8.5234t^3 \tag{2.31}$$

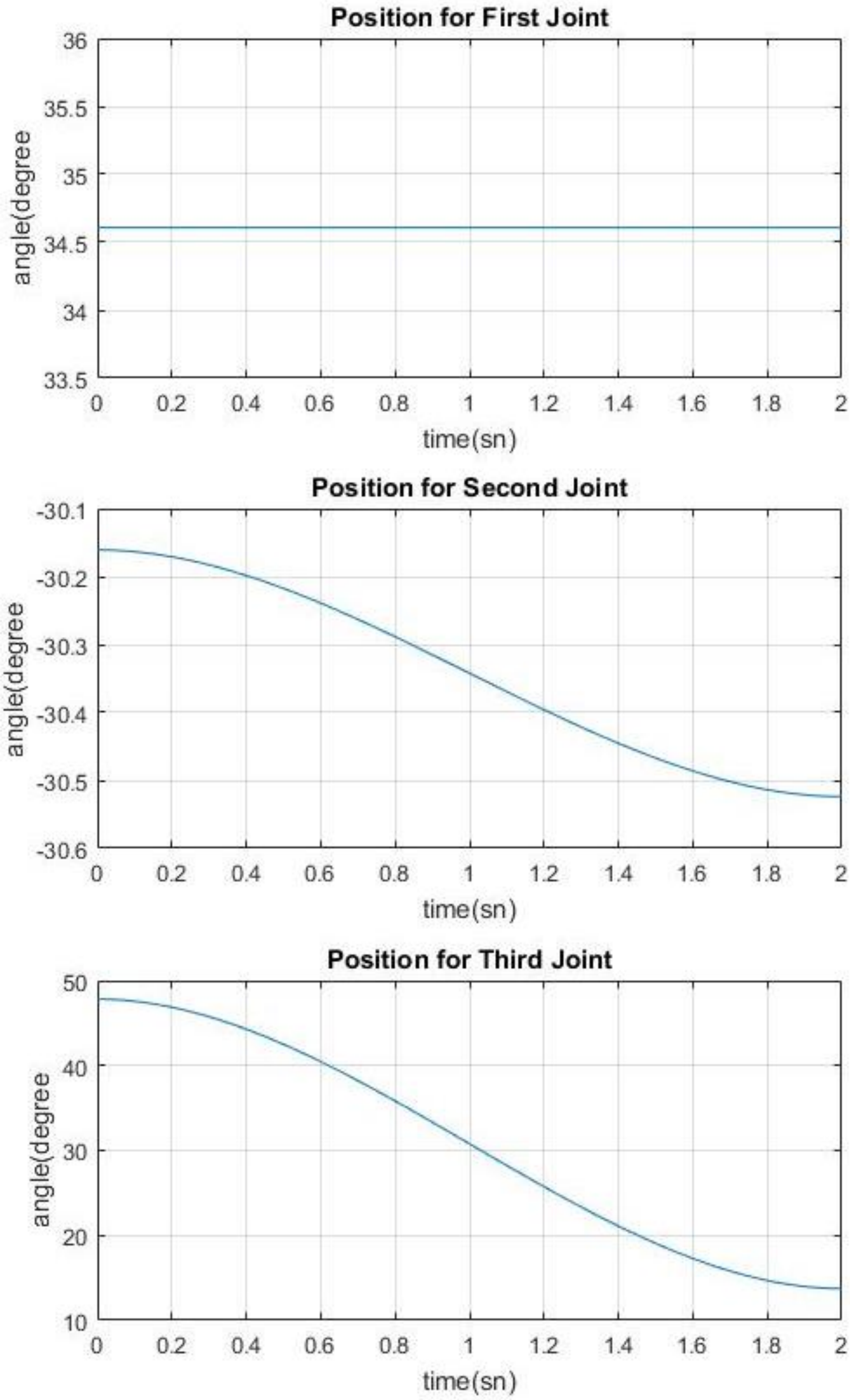

Figure 2.3. Joints position of the joint space.

After finding the polynomial of each joint position, velocity and acceleration polynomials can be obtained from positional derivatives. In the numerical analysis, when the movement of the robot is considered to be performed for 2 seconds, the position graphs in the joint space of each joint are given in Figure 2.3.

Path planning in the joint space allows the end effector to move from the start point to the desired end point at a certain time interval. If there are no intermediate points between the start point and the end point, it is possible to form the path equation with a third-order cubic equation. If starting and ending velocity and angular position are known, the coefficients of a third-order cubic path equation can be found [14].

When planning a path in the Cartesian space, in some cases it is desirable for the end effector to pass through certain intermediate points. In this case, the velocity of the end effector differs from zero when passing through the intermediate points. Path equation should be quantic (5-degree) for smoothing the path curve, which is given in Equation 2.32 [15].

$$\theta(t) = s_0 + s_1 t + s_2 t^2 + s_3 t^3 + s_4 t^4 + s_5 t^5 \tag{2.32}$$

The velocity, acceleration and displacement limits at the start and end point are given in Equations 2.33, 2.34 and 2.35 respectively.

$$\begin{aligned} \theta(0) &= \theta_0 \\ \theta(f) &= \theta_f \end{aligned} \tag{2.33}$$

$$\begin{aligned} \theta'(0) &= \theta'_0 \\ \theta'(f) &= \theta'_f \end{aligned} \tag{2.34}$$

$$\begin{aligned} \theta''(0) &= \theta''_0 \\ \theta''(f) &= \theta''_f \end{aligned} \tag{2.35}$$

The derivation of Equation 2.32 is obtained the velocity given in Equation 2.36.

$$\theta'(t) = s_1 + 2s_2 t + 3s_3 t^2 + 4s_4 t^3 + 5s_5 t^4 \tag{2.36}$$

It is possible to find the acceleration equation given in Equation 2.37 by taking the second derivation of the path equation.

$$\theta''(t) = 2s_2 + 6s_3 t + 12s_4 t^2 + 20s_5 t^3 \tag{2.37}$$

The coefficients of the angular position equation can be determined using the determined displacement, velocity and acceleration limits.

$$
\begin{aligned}
s_0 &= \theta_0 \\
s_1 &= \theta_0' \\
s_2 &= \frac{\theta_0''}{2} \\
s_3 &= \frac{20(\theta_f - \theta_0) - (8\theta_f' + 12\theta_0')t_f - (\theta_f'' - 3\theta_0'')t_f^2}{2t_f^3} \\
s_4 &= \frac{30(\theta_f - \theta_0) - (14\theta_f' + 16\theta_0')t_f + (3\theta_0'' - 2\theta_f'')t_f^2}{2t_f^4} \\
s_5 &= \frac{12(\theta_f - \theta_0) - 6(\theta_f' + \theta_0')t_f - (\theta_0'' - \theta_f'')t_f^2}{2t_f^5}
\end{aligned}
\tag{2.38}
$$

3. GENETIC ALGORITHM

Holland described Genetic Algorithms (GA) as a search and optimization process based on the principles of nature. In other words, GA is software that performs function optimization. More than 40 years Genetic algorithms based on of genetics are used in solving difficult problems. The measurement of the quality of the solution is defined by the fitness function, and this measurement determines which chromosomes to process. The fitness function can be a mathematical model or a computer simulation [16].

GA is one of the evolutionary algorithms and allows the selection of the most suitable individuals to produce an optimized solution. Candidate solutions are a coding of the problem similar to the chromosomes of biological systems. The biological chromosome and the computational chromosome were shown in Figure 3.1. Chromosomes come together to form populations. Each chromosome is a solution for objective function. The most appropriate chromosomes are selected according to their level of suitability for the problem and used to generate new generations. This leads to evolution between generations more appropriate to the problem. Evolution takes place using natural genetic mechanisms such as selection, crossover and mutation [17]–[20].

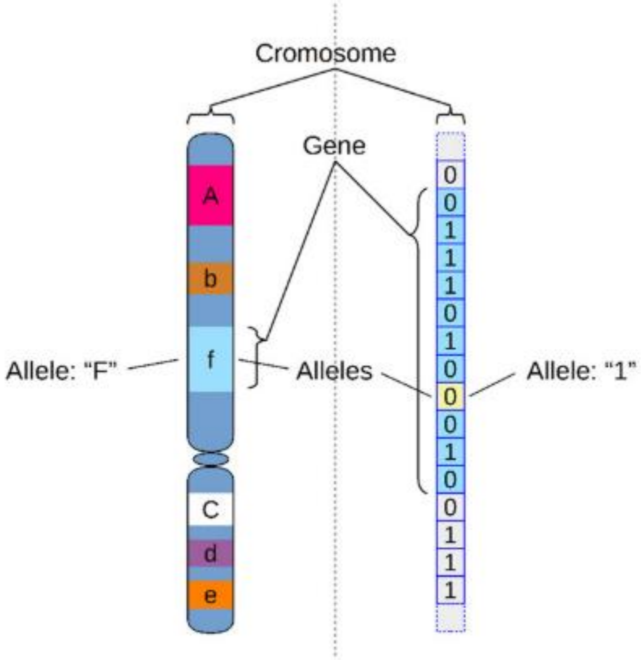

Figure 3.1. (left) Biological chromosomes; (right) Computational chromosomes [21].

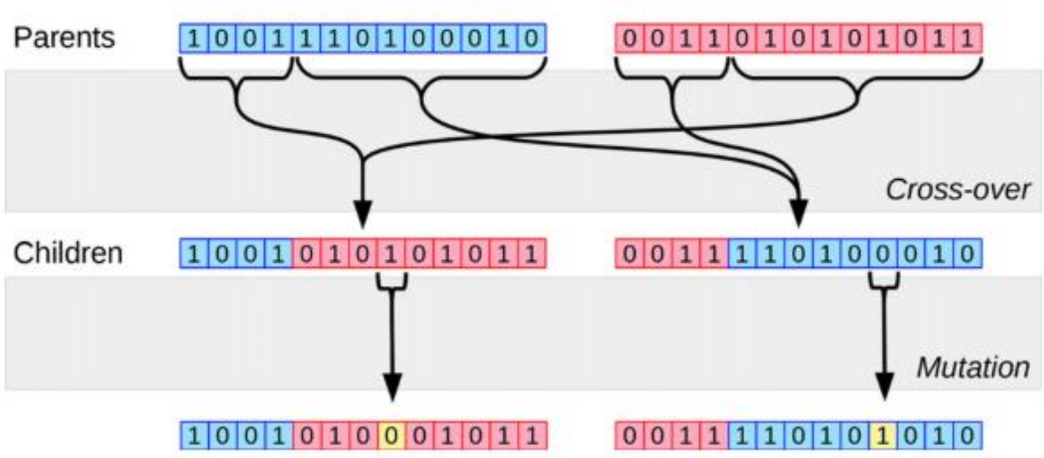

Figure 3.2. The parents and children [21].

The process of selection determines which individuals to match and how many new individuals will be created. It increases the chances of better individuals being selected and parents. The search area is reduced by throwing out bad individuals. The process of creating children from parents was shown in Figure 3.2. The aim of the selection is to ensure that the newly formed offspring give better fitness results by using filtered individuals. The selection needs to be balanced by crossover and mutation. This is due to the very strong selection that the population of overly good individuals takes over and the diversity is reduced for further progress. Otherwise, a weak choice causes a very slow evolution. There are many selection mechanisms in the genetic algorithm literature. One of the obvious problems of genetic algorithms is that there is no explanation of which method should be used for which problems [22]. There are three basic selection mechanisms: Roulette Wheel Selection, Tournament Selection, Rank-based Roulette Wheel Selection [23], [24].

The method of selection in proportion to the suitability of chromosome individuals is called the Roulette Wheel. This method increases the probability that the most suitable element will move to the next generation. The total suitability of the population of a wheel-like graphic constitutes the size of the area. As seen in Figure 3.3, the size of each section is the suitability value of the chromosomes. The population of individuals giving more favourable results is larger [25].

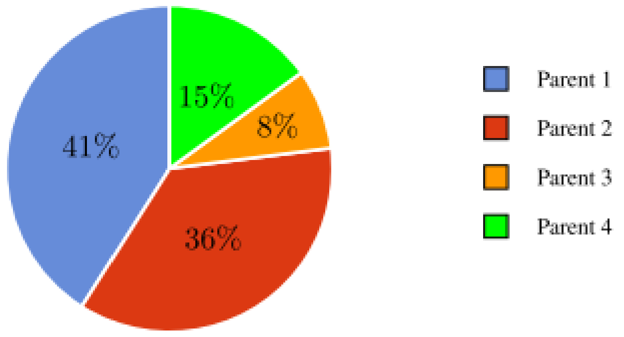

Figure 3.3. Roulette Wheel Selection [25].

Tournament selection method is another selection method that also takes place in genetic algorithms. The tournament selection mechanism randomly selects 2 or more chromosomes and selects the best fitness value chromosome and discards the other chromosome. As seen in Figure 3.4, the chromosome determined as a result of the selection is pushed into the matching pool and subjected to genetic algorithm operations. There is no possibility to take the parent with the smallest fitness value in tournament selection. In the selection of Roulette Wheels, the probability of being placed in the match pool of each parent is relative. This is the main difference between tournament selection and Roulette Wheel selection. The tournament selection chooses 2 individuals each time from the population size. The selection continues until the number of individuals in the new generation is equal to that of the old generation. Referring to Figure 3.4, it is seen that P_2 will be selected and taken into the mating pool since the suitability value of P_2 is greater than P_1. Likewise, since the suitability value of P_5 is greater than P_3, P_5 will be taken into the mating pool. The tournament selection will select randomly by taking at least 2 chromosomes from the old population [25].

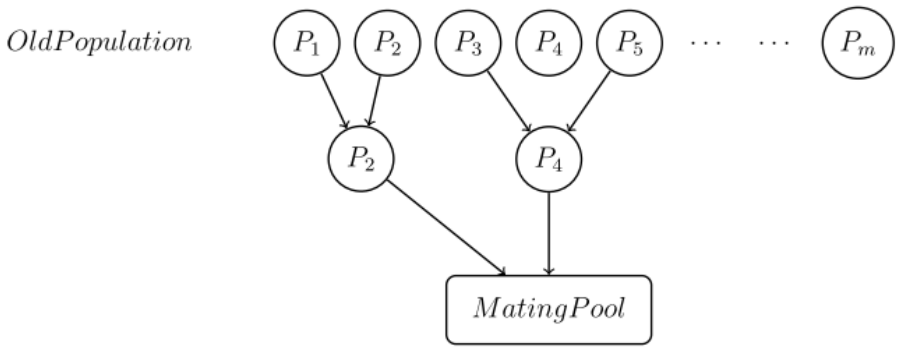

Figure 3.4. Tournament Selection [25].

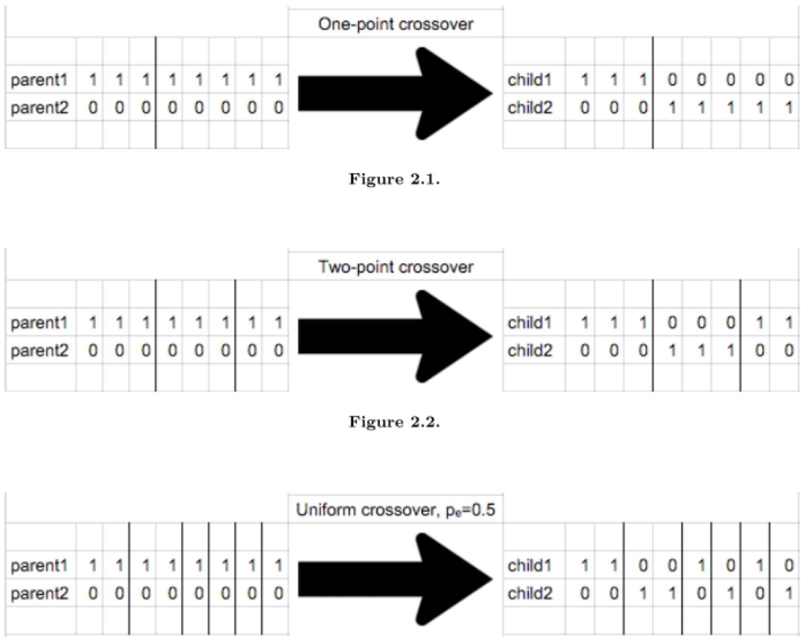

Figure 3.5. Crossover.

Offspring should be created after the parents are selected and can be done in a variety of ways. Crossover is used to create better generations than the previous generation using the gene pool. Two chromosomes from the new generation are randomly selected and crossed mutually. The one point crossover method is the simplest way and both chromosomes must have the same gene length. The one point crossover selects a random point on the chromosomes and replaces all items from this point on the array to create new individuals. In the two point crossover, two points are selected on the chromosome and the items between or outside the two elements are replaced. Instead of one and two point crossover, chromosomes that can be divided by n points in random cross between sections, and this is called uniform crossover. Figure 3.5 are shown one and two point and uniform crossover [26].

The mutation is the mechanism by which new chromosomes are produced from existing chromosome families. Crossover leads to loss of genetic material and mutation prevents this loss. Mutation is applied to increase the diversity of chromosomes. The mutation is applied to a certain percentage of individuals who are the result of crossover. This rate is called the mutation rate [27], [28].

4. SOLID MODEL DESIGN FOR RRR ROBOT AND SIMULATION

An industrial robot is a programmable manipulator with a tool for performing an operation at the end. Industrial robotic systems consist of arm lengths connected to each other in parallel or serial [29]. Arm lengths determine the configuration of robots as they have great effects on their weight. Reducing the weights of the arms will reduce the energy consumption and extend the life of the motors used for the movement. The weight of the arms can be reduced by decreasing the materials used in the construction or by using different types of materials [30]. The designs made with 3d printers are a solution that can reduce the weight of the robots due to their low weight. In recent years, 3d printers have had an increasing popularity. The fused filament fabrication (FFF) technique is use 3d printers. The plastic derivatives which are polylactic acid (PLA) and acrylonitrile butadiene styrene (ABS) is used as filament [31]. PLA and ABS materials can be used in the production of robot parts due to their lower specific gravity. However, mechanical strength of PLA and ABS is not good because it is mechanically about 10 times weaker than the casting materials forming the bodies of industrial robots. Therefore, we cannot use these materials in robots designed to lift loads. It is possible to make handles and other parts of robots that do not have load lifting feature but can work very fast as delta robots. Delta robots can operate very fast due to their structure and therefore they are used in the construction of electronic circuits and quality control processes, etc. In Figure 4.1, a delta robot made some parts with 3d printer is given.

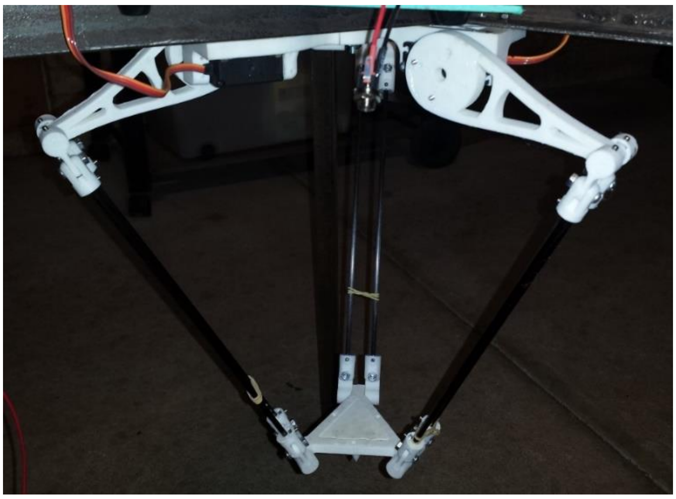

Figure 4.1. A delta robot made some parts with 3d printer [32].

In this study, an industrial robot which three rotary joints (RRR) are used was designed using Solidworks 2016. The robot configuration consists of 3 joints and 6 parts. While the first 3 joints in industrial robots determine the position in Cartesian space, the last 3 joints are generally the Euler wrist structure and determine the orientation. Therefore it is sufficient to examine the first three joints to determine the position. A small industrial robot model was designed for research and assembled in Solidworks. In Figure 4.2 represent the 3D CAD models of the RRR robot.

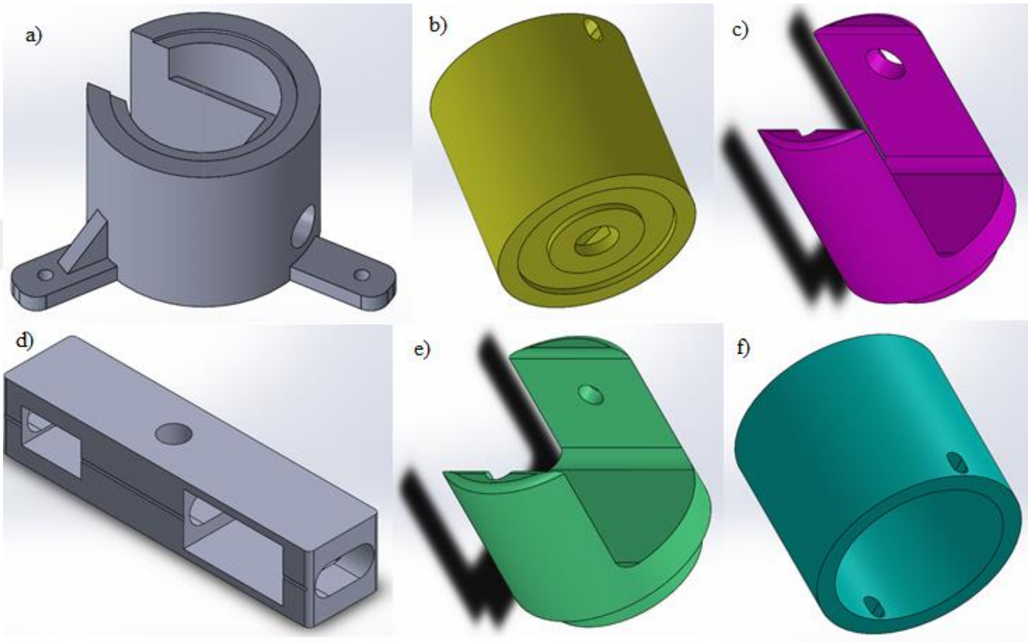

Figure 4.2. CAD models: a) Base and first joint, b) Joint offset, c) Second joint, d) Arm, e) Third joint, f) Arm.

The model formed by combining the parts and the robot configuration is given as isometric view in Figure 4.3. The lengths of the robot are given in Table 4.1. There is a joint offset between the first joint and second joint of the robot. There is also a length which is commonly seen in industrial robots between the reference coordinate system and the first joint.

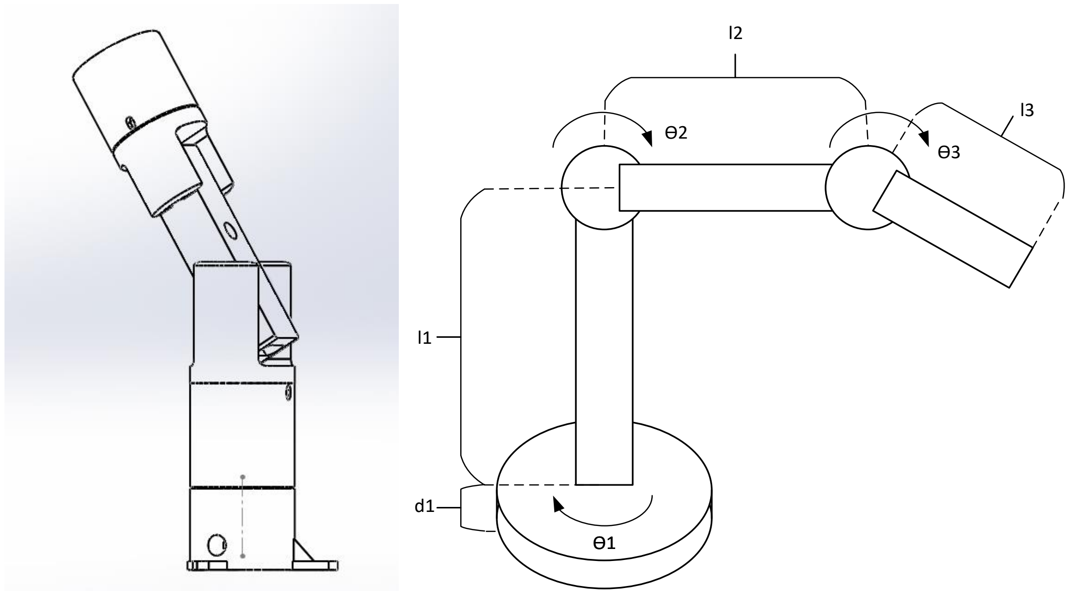

Figure 4.3. An isometric view RRR robot and the robot configuration.

Table 4.1. The lengths of the RRR robot.

The lengths of robot	
d1	115,6 mm
l1	46,6 mm
l2	71,8 mm
l3	83,5 mm

The solid model of the RRR robot was simulated in Solidworks. Parts that are not compatible with each other and parts colliding with each other are detected and the solid model has been changed.

4.1. Workspace of the Robot

The workspace is defined as the sum of points that the robot arm can reach in Cartesian space. It is of great importance by determining the boundaries of the working space [33]–[35]. The boundaries vary depending on the mechanical structure of the robot arm and the ability of the joints to rotate. Generally, a robot arm consists of two main parts, namely the regional structure and orientation structure [36]. The first three joints form the regional structure by determining the position, while the last three joints determine the orientation and form the orientation structure. Looking at the literature, there are many algorithms or methods for calculating workspace. Genetic algorithms are used to determine the working area of the Delta robot, which consists of parallel arms. Determining the working space is important for the robots to show their capacities [37]–[40]. The shape or width of the workspace depends on the degree of freedom of a robot arm and the location of the coordinates of the first three joints forming the robot arm. Usually the first three joints create a cylindrical, rotating, spherical and prismatic working space [11]. The arm length of an industrial robot arm is one of the most important factors affecting the size of the workspace. In addition, the structure and degree of rotation of the joints determine the shape of the working space.

The working space of the RRR robot used in this study was shown in Figure 4.4.

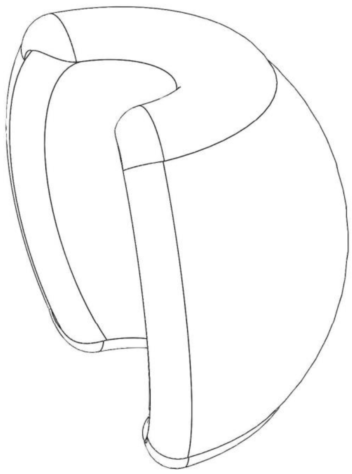

Figure 4.4. The workspace of RRR robot in this study.

The workspace of the RRR robot used in this study is classified as spherical workspace. Due to the rotating structure of all joints, a spherical workspace is formed. The rotation angles of the joints prevent the workspace from becoming a full sphere.

5. PRINTING AND ASSEMBLY OF SOLID MODEL WITH 3D PRINTER

3D printing technology has been the additive manufacturing process that has emerged in recent years. 3D printing method constructs the parts by layer by layer. 3D printers are of great importance especially in prototyping due to the rapid production of parts and updating. Additive manufacturing enables high precision production at low cost [41]. The first prototype of the designed RRR robot structure is produced by 3D printing method. 3D prints of all parts which were given the solid model were given in Figure 5.1.

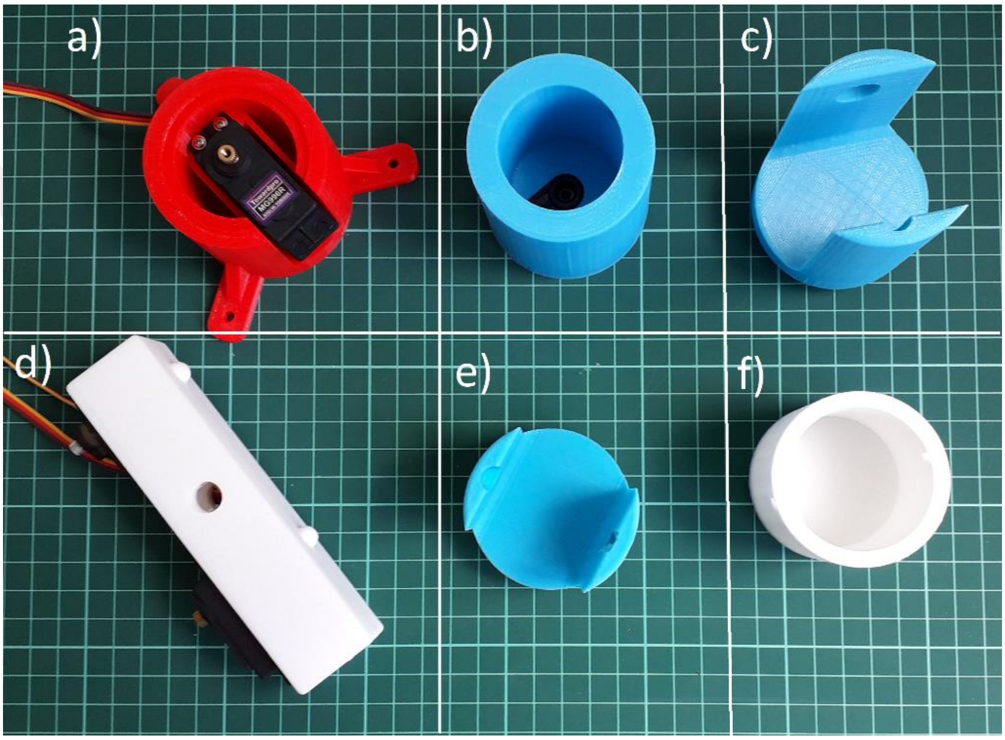

Figure 5.1. 3d prints of all parts.

Figure 5.2. z-axis ball bearing.

The robot structure was made of PLA material with a 3D printer and a lightweight robot body was created to reduce stresses on the motors. However, since the first joint carries the weight of the whole body of the robot, a z-axis ball bearing was used to facilitate the joint rotation with the servo motor. Figure 5.2 shows the image of the bearing used.

All parts were assembled and it was seen that the parts worked in harmony when simple movements were made. The assembled of the RRR robot was given in Figure 5.3. The robot was fixed to the floor from the base and prevented moving.

Figure 5.3. The assembled of the RRR robot.

6. ESTABLISHMENT OF THE EXPERIMENTAL SET AND ELECTRONICS CIRCUITS

An Arduino mega-control card with Matlab control is planned to use for running the RRR robot. In order to apply the numerical analysis results in Matlab to the controller, the controller is intended to work with Matlab. Servo motors were used to move the arms of the robot. 2 types of servo motors were used: MG966R and SG90 micro servo motor. MG966R servo motors with metal gear were used because of the biggest forces in the first and second joints. The movement of the third joint was provided by the SG90 micro servo motor. The first motor was embedded in the base part and the second and third motors were embedded in the arm. The motors will draw more current when they work together. If the servo motors are fed through the control board, the current will be low. In this case the external power supply was used to prevent the current from falling. The connection between the motors, the control board and the external power shall provide the electronic circuit in Figure 6.1.

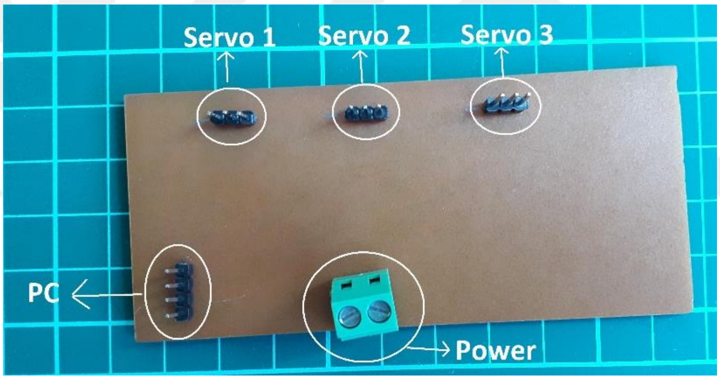

Figure 6.1. Electronic circuit.

The commands sent by Matlab will be processed on the control card and sent to the motors in the joints. The data obtained in numerical analysis of RRR robot will be applied to robot structure with control card. The data obtained from the experimental set will be recorded and compared with the numerical analysis. The reasons of the cases will be investigated and the results will be examined.

7. EXPERIMENTAL STUDY

Robot kinematics and path planning in the joint space are described in the previous section. Path planning is a combination of points in time space and depends on the technical properties of the motors used in joints. Consequently, genetic algorithms have been used to achieve optimal planning of motion time and to make the angular position, velocity and acceleration curves of each joint softer.

As the robot moves, each arm and end effector passes through n points. Inverse kinematic solution is made for each point to find the joint variable. All joints reach their target angular positions through certain points. In order for the robot's end effector to reach the target point, each joint has a target point that it must reach. Therefore, the motion of each joint is examined separately. If each joint completes its movement at time t_i, the total movement time is determined as T, given in Equation 7.1.

$$T = \sum t_i \tag{7.1}$$

Each joint is expected to complete its movement in a short time and reach the target point. In this study, the motion time of the robot has been optimized by using the evolutionary algorithms GA. The experimental study used the robot arm is aimed to complete a determined movement as soon as possible. For this purpose, the objective function is given in Equation 7.2.

$$\text{Objective Function: } T = \sum t_i \tag{7.2}$$

The limits of the objective function are determined by the servo motors used in the joints. The velocity and acceleration values of the joint directly affect the movement time. The velocity and acceleration values of the servo motors constitute the constraint conditions of the objective function and these values are given in Table 7.1.

Table 7.1. The constraint conditions.

Joint	Velocity ($\theta'(t)$)	Acceleration ($\theta''(t)$)
1	$300 \le \theta''(t) \le 600$	$50 \le \theta''(t) \le 200$
2	$300 \le \theta''(t) \le 600$	$50 \le \theta''(t) \le 200$
3	$400 \le \theta''(t) \le 750$	$40 \le \theta''(t) \le 300$

Optimization is done with GA toolbox of MATLAB. The objective of the optimization is to minimize the total time under given constraints. Selection of GA's natural genetic mechanisms has determined scholastic uniformly. The population size was entered as 50 and it was observed that this size was sufficient in the simulation results. The algorithm's stop criterion is 100*numberOfVariables, which is the default value. The genetic algorithm was run for each joint at different crossover fraction and mutation rates. Mutation rates were determined as 0.01 and 0.02. The crossover fraction was selected as 0.2, 0.5 and 0.8. Finally, mutation function was determined as Uniform and simulation was performed.

A quintic path polynomial was created for three joints and time optimized. The robot, which specified in Table 4.1, was moved from $P_0(\,\mathrm{X}_0 = 155.28,\ \mathrm{Y}_0 = 0,\ \mathrm{Z}_0 = 164.14)$ to $P_1(\,\mathrm{X}_1 = 135.65,\ \mathrm{Y}_1 = 23.92,\ \mathrm{Z}_1 = 99.12)$ in Cartesian space. In the P_0 position, the robot joints were $\theta_1 = 0,\ \theta_2 = 0,\ \theta_3 = 0$ in the zero position and in P_1 the position was $\theta_1 = 10,\ \theta_2 = 15,\ \theta_3 = 20$. As a result of the optimization, the movement time of each joint was obtained. Angular position values of three joints are given in Figure 7.1 and velocity values are given in Figure 7.2.

In order to show smoothness, the speed graph obtained as a result of an optimization is given. The smoothness of the graphics prevents stress accumulation during movement and prevents mechanical damage to the robot arm.

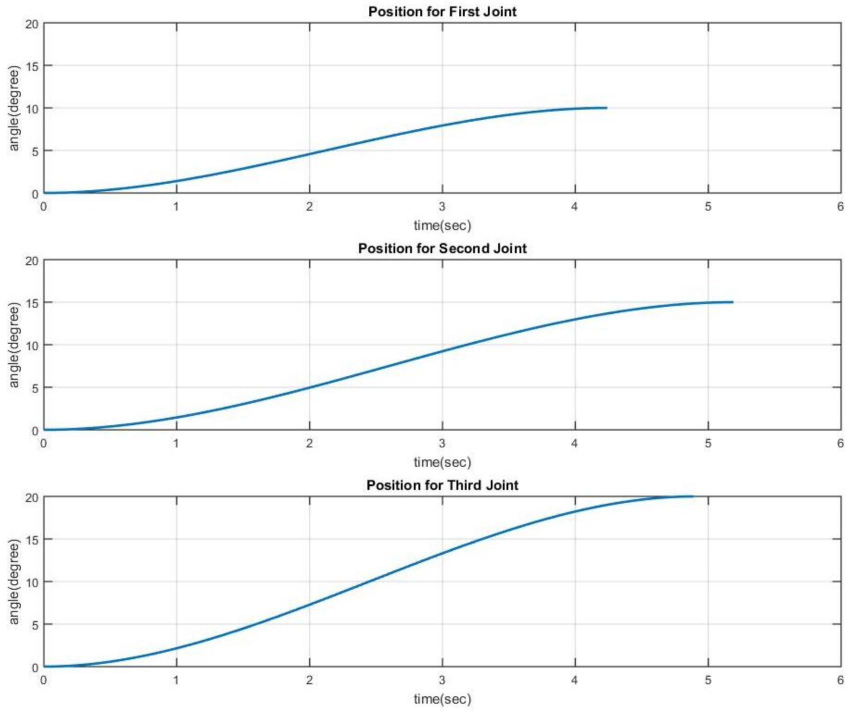

Figure 7.1. Angular position graphics of three joints.

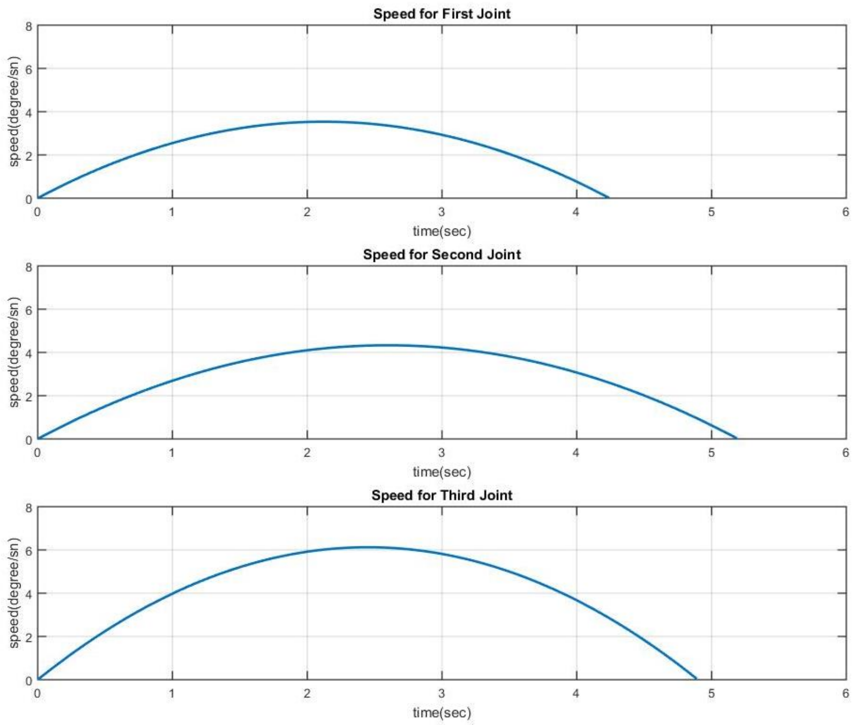

Figure 7.2. Velocity graphics of three joints.

The optimization results with 0.01 mutation rate are given for the first joint. Studies with crossover fractions of 0.2, 0.5 and 0.8 are given respectively.

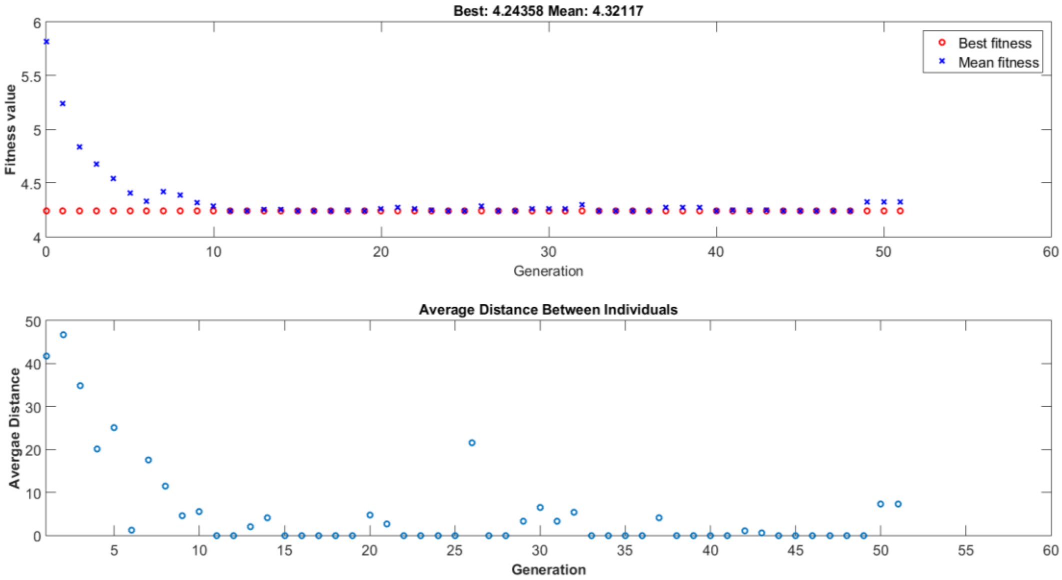

Figure 7.3. The first joint's optimization result (crossover = 0.2 and mutation rate = 0.01).

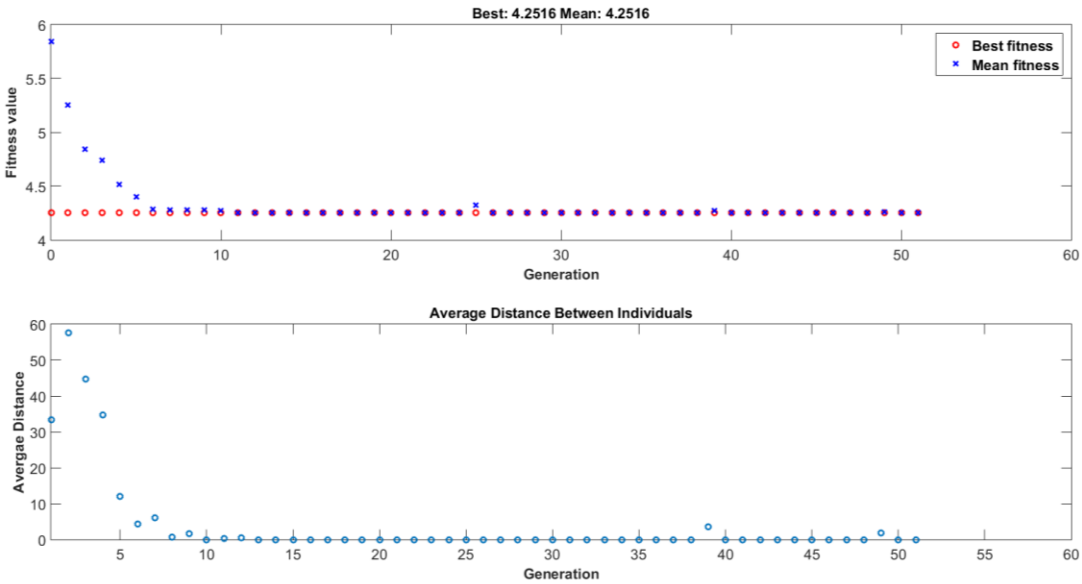

Figure 7.4. The first joint's optimization result (crossover = 0.5 and mutation rate = 0.01).

In Figure 7.3, the result of the algorithm operated with 0.2 crossover fraction and 0.01 mutation rate for the first joint is given. It has been observed that optimization ends in the 51st generation. The reason for this is that the best result has been achieved after the 11th generation and there is no big difference between the results obtained later. As a result of the algorithm, it completes the first joint movement in 4.24358 seconds with 0.01 mutation rate and 0.2 crossover fraction.

The result of the optimization with the 0.5 crossover fraction and the 0.01 mutation rate for the first joint is given in Figure 7.4. When the figure is examined, it is seen that the algorithm ends in 11th generations. It is seen that there is less difference between generations compared with optimization with 0.2 crossover fraction. As a result of optimization with 0.5 crossover fraction and 0.01 mutation rate, it was calculated that the first joint completed its movement in 4.2516 seconds.

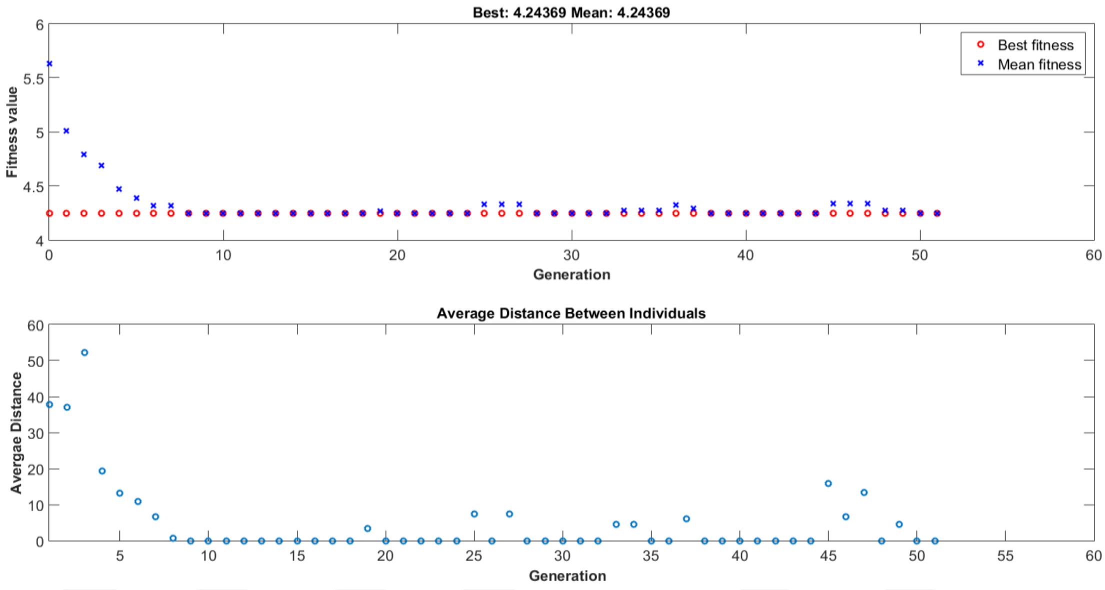

Figure 7.5. The first joint’s optimization result (crossover = 0.8 and mutation rate = 0.01).

In Figure 7.5, the result of the algorithm run with 0.01 mutation rate and 0.8 crossover fraction is given. The algorithm is finished in the 51st generation. After the 8th generation, the difference between best fitness and mean fitness has closed, and this difference will be very little in later generations. The algorithm, which is operated with 0.8 crossover fraction and 0.01 mutation rate, gives 4.24369 seconds for the first joint.

The algorithm for the first joint was performed using crossover fraction of 0.2, 0.5 and 0.8 at the 0.01 mutation rate. The results obtained are given in Table 7.2. When the results are examined, it is seen that the shortest period is obtained as a result of 0.2 crossover fraction.

Table 7.2. First joint algorithm results with 0.01 mutation rate.

Crossover Fraction	Mutation Rate (0.01)
0.2	4.24358 sec
0.5	4.2516 sec
0.8	4.24369 sec

After the optimization of the first joint at the 0.01 mutation rate is finished, the optimization of the second joint at the same mutation rate was made. First, the algorithm was run for the second joint (objective function and the constraint conditions are different) with a 0.2 crossover fraction and 0.01 mutation rate.

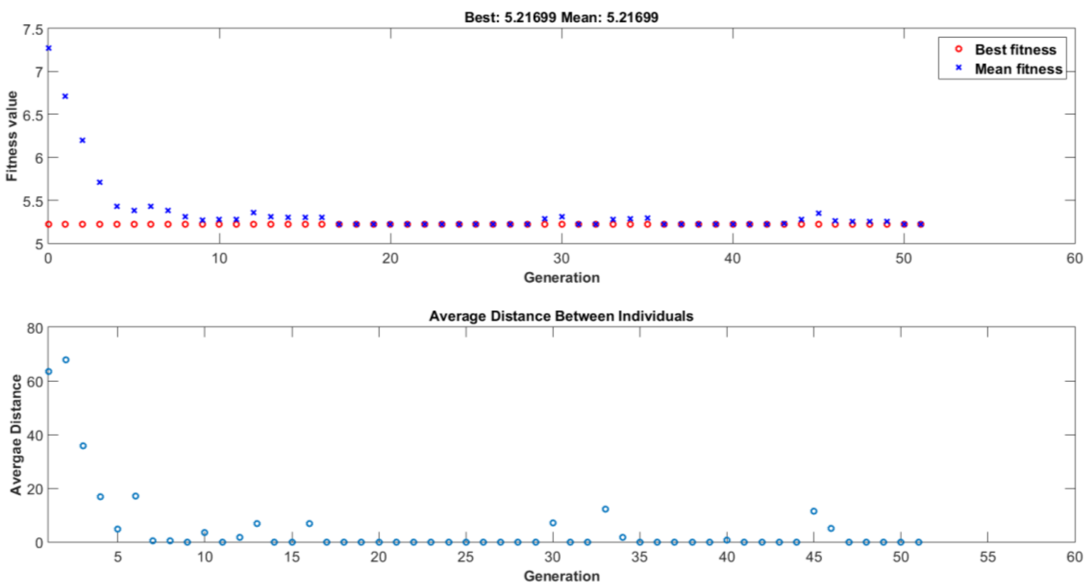

Figure 7.6. The second joint's optimization result (crossover = 0.2 and mutation rate = 0.01).

The algorithm for the second joint was run at the rate of 0.2 crossover fraction and 0.01 mutation rate, and the result is given in Figure 7.6. The algorithm achieved the best result in the 17th generation and resulted in the 51st generation. In the chart of average distance between individuals, it is seen that there are small differences between generations. The algorithm, working with 0.01 mutation rate and 0.2 crossover fraction, gives 5.21699 seconds for the second joint.

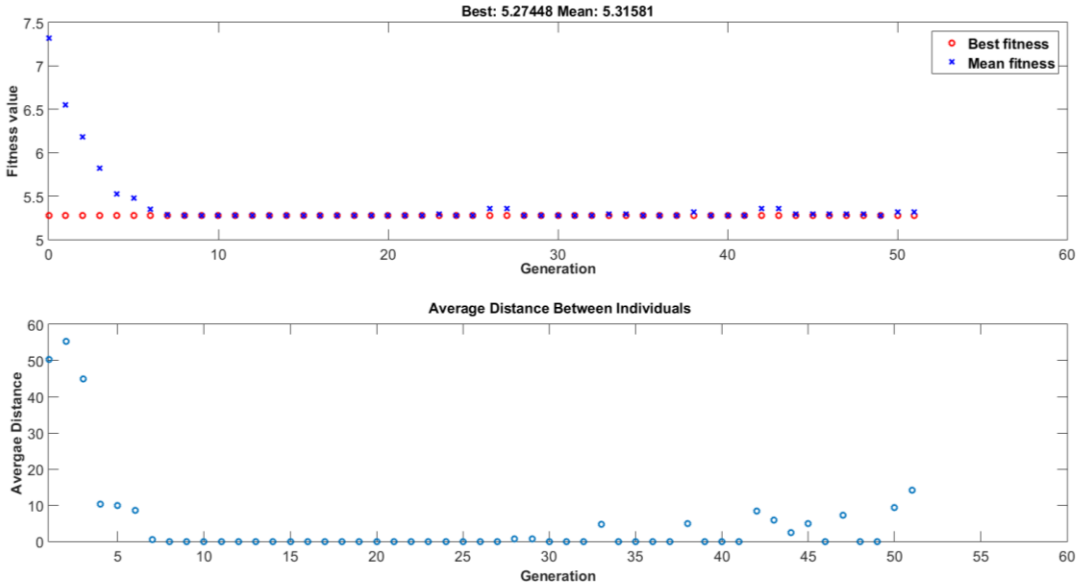

Figure 7.7. The second joint’s optimization result (crossover = 0.5 and mutation rate = 0.01).

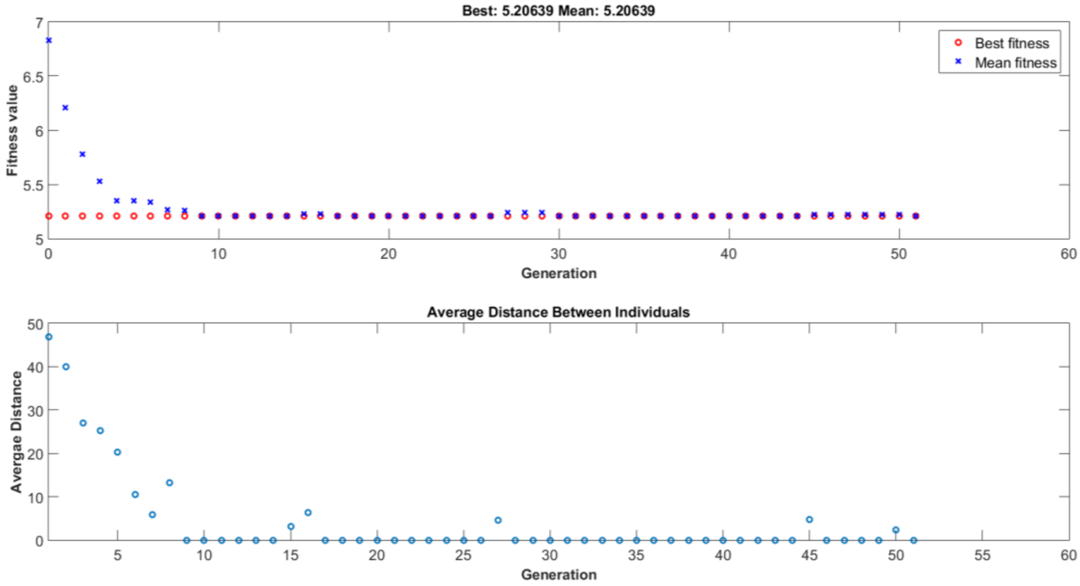

Figure 7.8. The second joint’s optimization result (crossover = 0.8 and mutation rate = 0.01).

In Figure 7.7, optimization was done with a 0.5 cross ratio for the second joint. The optimization has reached the result at the 7th generation, and the difference between best fitness and mean fitness in the next generations is minimal. When looking at the average distance between individuals, a difference is observed between generations. The optimization with 0.5 crossover fraction and 0.01 mutation rate resulted in 5.27448 seconds.

For the second joint, the algorithm was run with 0.8 crossover fraction and 0.01 mutation rate and given in Figure 7.8. When the figure is examined, it is seen that the result is reached in the 9th generation. Compared to the optimization of the second joint with 0.5 crossover fraction, it is seen that the average distance between individuals is less. However, optimization with 0.5 crossover fraction reaches the result in a shorter time. As a result of algorithm operated with 0.8 crossover fraction and 0.01 mutation rate, the result was found as 5.20639 seconds.

The results of the optimizations performed at different crossover fraction for the second joint are given in Table 7.3. As a result of the algorithms run, it is seen that the best value in the 0.01 mutation rate is obtained at the 0.8 crossover fraction.

Table 7.3. The second joint algorithm results with 0.01 mutation rate.

Crossover Fraction	**Mutation Rate (0.01)**
0.2	5.21699 sec
0.5	5.27448 sec
0.8	5.20639 sec

The algorithm for different crossover fraction at 0.01 mutation rate in the third joint was run and the results were shown. Firstly, optimization was made at 0.2 crossover fraction.

For the third joint, the algorithm, which was run at the rate of 0.2 crossover fraction and 0.01 mutation rate, gave 4.90852 seconds. Figure 7.9 shows the graphic obtained as a result of the algorithm. The algorithm approached the best result in the 7th generation and finished in the 51st generation. When the average distance between individuals is examined, it is seen that there is fluctuation between generations. The difference between the best fitness and mean fitness is seen as less between generations.

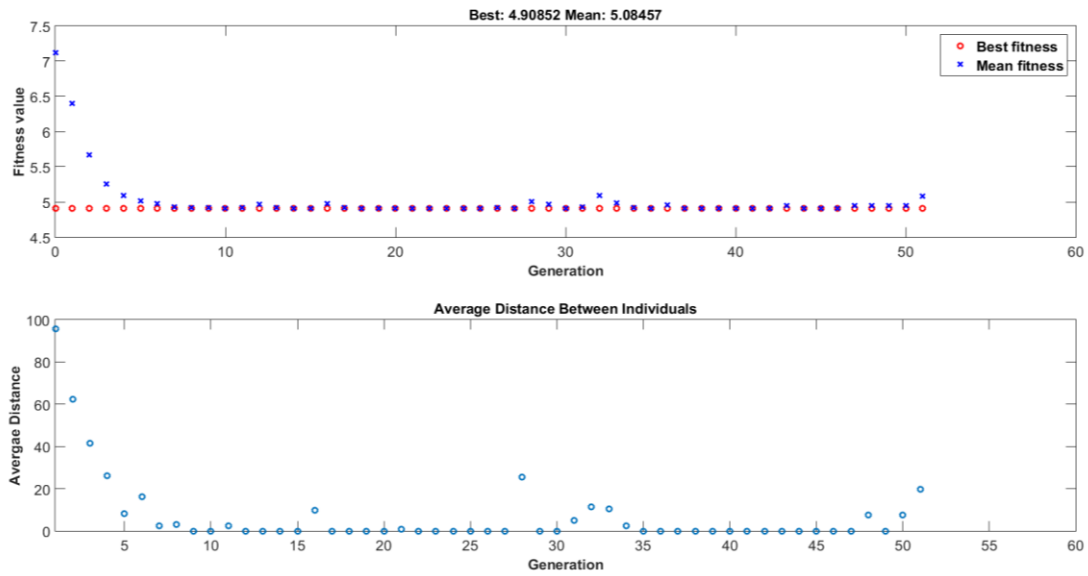

Figure 7.9. The third joint's optimization result (crossover = 0.2 and mutation rate = 0.01).

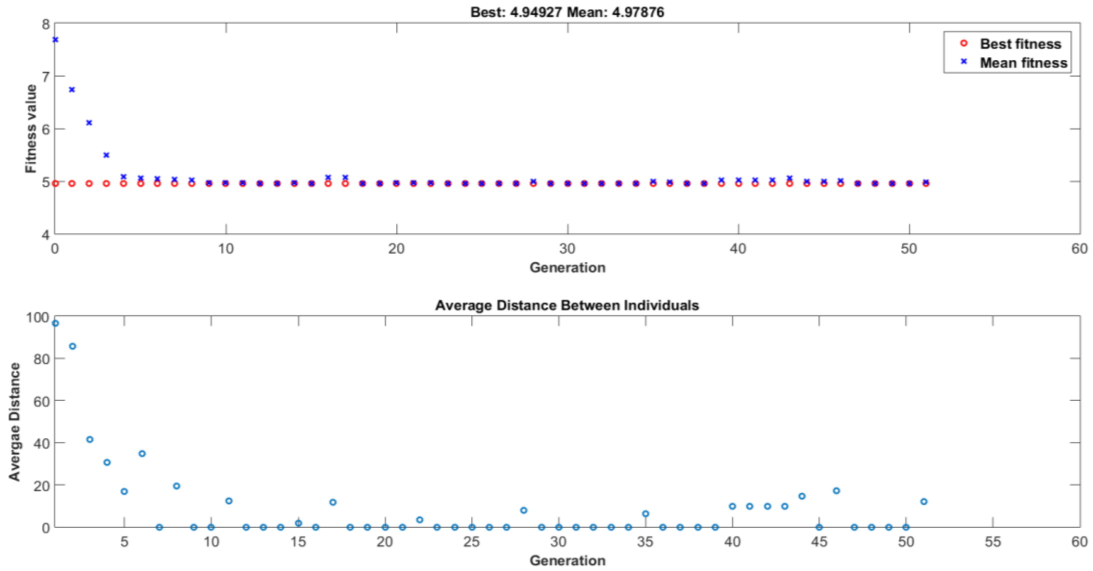

Figure 7.10. The third joint's optimization result (crossover = 0.5 and mutation rate = 0.01).

Figure 7.10 shows the result of optimization with 0.5 crossing and 0.01 mutation rate for the third joint. Although the optimization approached the result in the 5th generation, it was able to reach the result in the 9th generation. As a result of optimization, the result for the third joint was found as 4.94927 seconds.

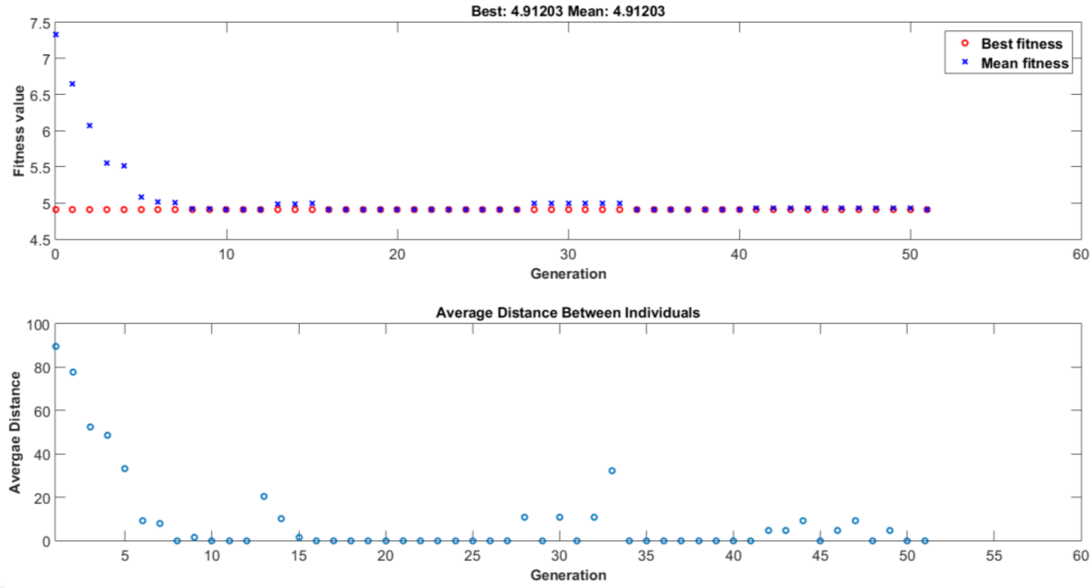

Figure 7.11. The third joint's optimization result (crossover = 0.8 and mutation rate = 0.01).

The final optimization of the third joint at the 0.01 mutation rate was done with a 0.8 crossover fraction and was shown in Figure 7.11. Although the algorithm approached the result in the 5th generation, it reached the result in the 8th generation. There are differences between the best fitness and mean fitness between generations. The average distance between individuals varies between generations. With 0.01 mutation rate and 0.8 crossover fraction operators, the best value of the algorithm is 4.91203 seconds.

The results of the studies performed using different genetic algorithm operators of the third joint are given in Table 7.4. The best result was obtained with 0.01 mutation rate and 0.2 crossover fraction. However, when considering other results, there is no big difference between the results.

Table 7.4. The third joint algorithm results with 0.01 mutation rate.

Crossover Fraction	**Mutation Rate (0.01)**
0.2	4.90852 sec
0.5	4.94927 sec
0.8	4.91203 sec

The algorithm was run in three joints with different crossover fraction at the 0.01 mutation rate, and the results are presented in tables. The optimization was done with the constraint conditions determined for each joint. The results obtained are for the robot to move from point P_0 to point P_1.

Optimization was run with 0.02 mutation rate and 0.2 and 0.5 crossing rates to see the results of the algorithm at different mutation rates.

The result of the algorithm which is run at the rate of 0.2 crossover fraction and 0.02 mutation rate for the first joint is given in Figure 7.12. It is observed that the algorithm ends in the 51st generation. The algorithm approached the result in 7th generation, but it is noteworthy that the number of generations with the difference between best fitness and mean fitness is high. The differences average distance between individuals are observed between generations. The algorithm ultimately gave 4.24654 seconds for the first joint.

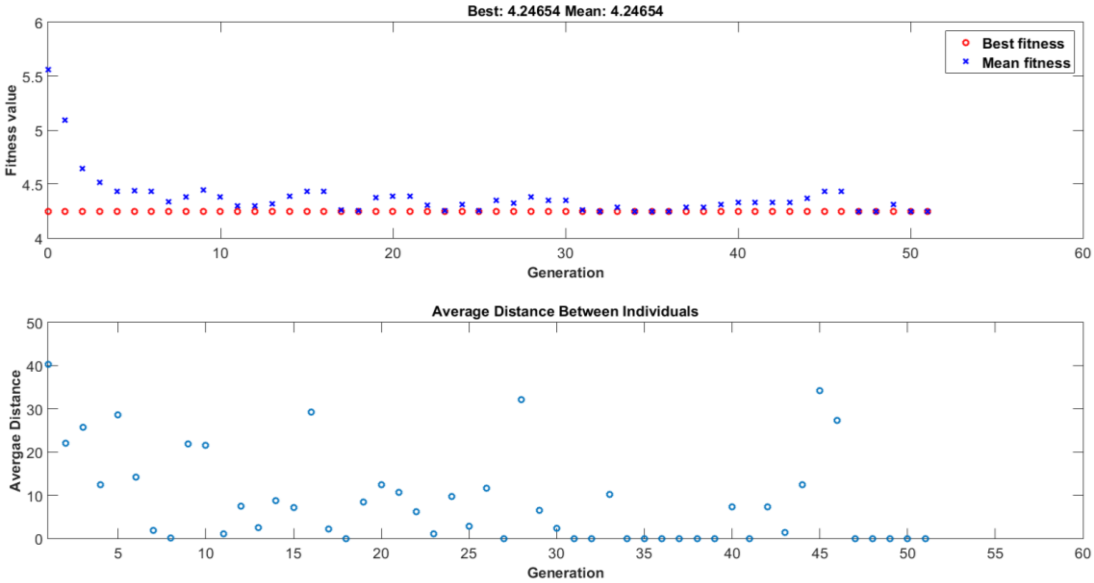

Figure 7.12. The first joint’s optimization result (crossover = 0.2 and mutation rate = 0.02).

In Figure 7.13, the result of the algorithm run for the first joint with 0.5 crossover fraction and 0.02 mutation rate is given. The result was achieved in the 6th generation and the result between generations does not differ much. Compared to the study with 0.2 crossover fraction and 0.02 mutation rate, it is observed that there is less difference between generations and between individuals. The result for the first joint with 0.5 crossover fraction and a rate of 0.02 mutation is 4.28907 seconds.

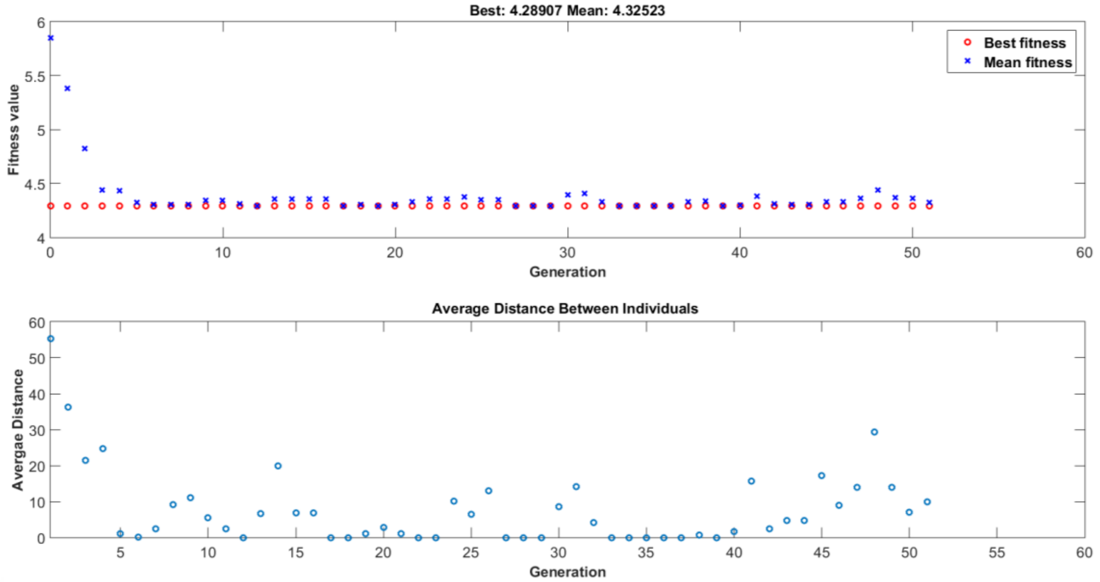

Figure 7.13. The first joint's optimization result (crossover = 0.5 and mutation rate = 0.02).

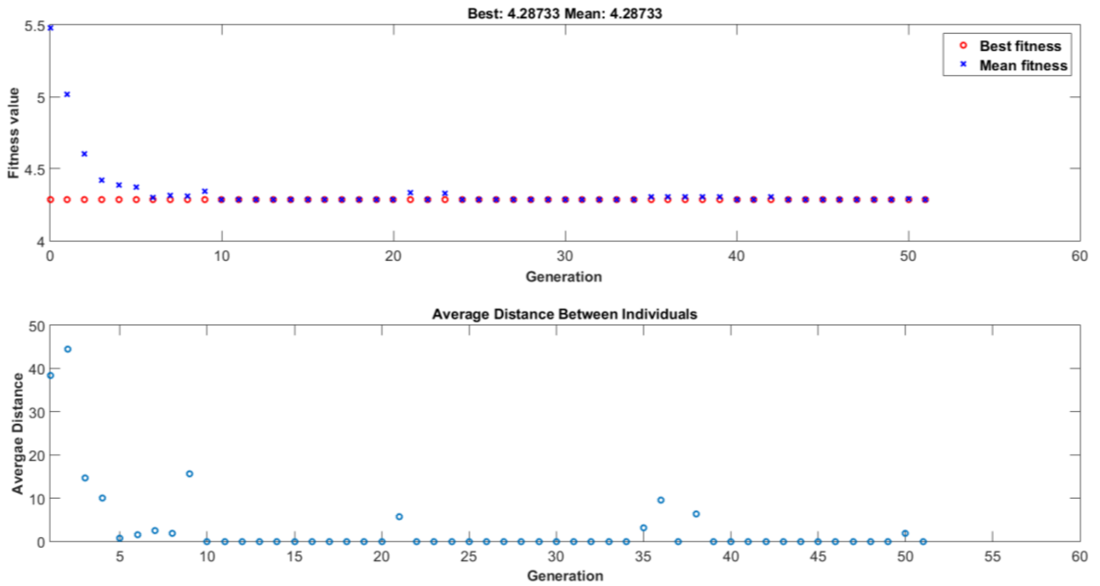

Figure 7.14. The first joint's optimization result (crossover = 0.8 and mutation rate = 0.02).

The result of the algorithm with 0.8 crossover fraction and 0.02 mutation rate is given in Figure 7.14. When the figure is examined, it was observed that the result was obtained in the 10th generation and the algorithm was terminated in the 51st generation. There is very little difference between best fitness and mean fitness in between generations. The algorithm working with 0.8 crossover fraction and 0.02 mutation rate calculated the result as 4.28733 seconds.

The algorithm for the first joint was run for different crossover fraction at the rate of 0.02 mutation and the results obtained are given in Table 7.5. Almost the same result was obtained at crossover fraction of 0.5 and 0.8, but the best result was obtained with crossover fraction of 0.2.

Table 7.5. The first joint algorithm results with 0.02 mutation rate.

Crossover Fraction	Mutation Rate (0.02)
0.2	4.24654 sec
0.5	4.28907 sec
0.8	4.28733 sec

After the optimization of the first joint with the 0.02 mutation rate is finished, the optimization of the second joint with the 0.02 mutation rate and different crossover fraction is performed.

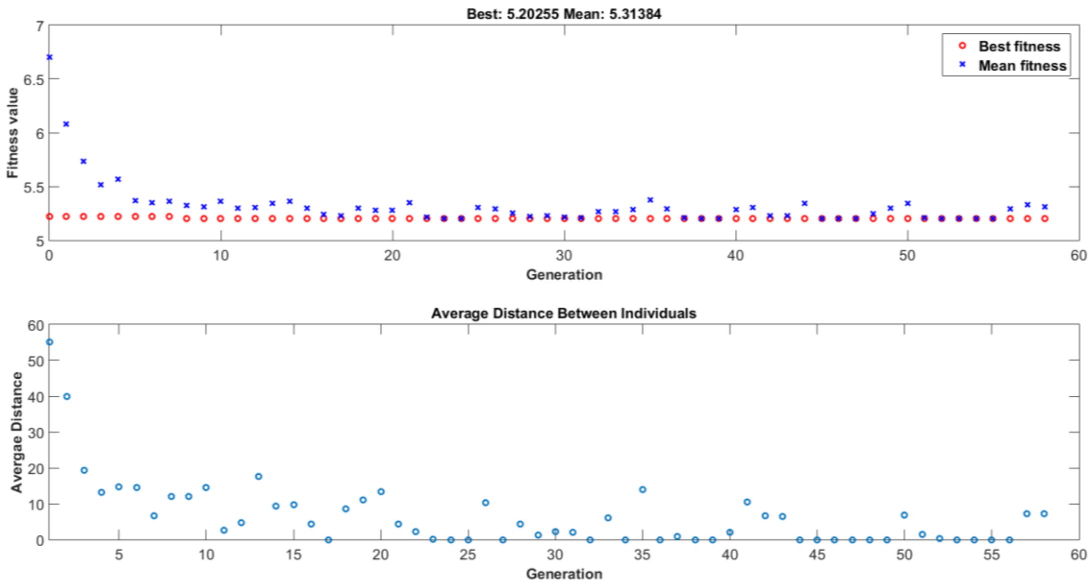

Figure 7.15. The second joint's optimization result (crossover = 0.2 and mutation rate = 0.02).

The algorithm for the second joint was performed with 0.2 crossover fraction and 0.02 mutation rate, and the result is given in Figure 7.15. When the fitness value graph is examined, it is seen that there is a great difference between the best fitness and mean fitness between generations. The algorithm, which operates with 0.2 crossover fraction and 0.02 mutation rate, gives 5.20255 seconds result.

For the second joint, the algorithm was run with 0.5 crossover fraction and 0.02 mutation rate and given in Figure 7.16. The algorithm achieved the best value in the 9th generation and ended in the 51st generation. There is little difference between the best fitness and mean fitness between generations. Compared to the optimization with 0.2 crossover fraction for the second joint, it is seen that the difference between generations is less. As a result of optimization with 0.5 crossover fraction and 0.02 mutation rate, 5.19656 seconds were found.

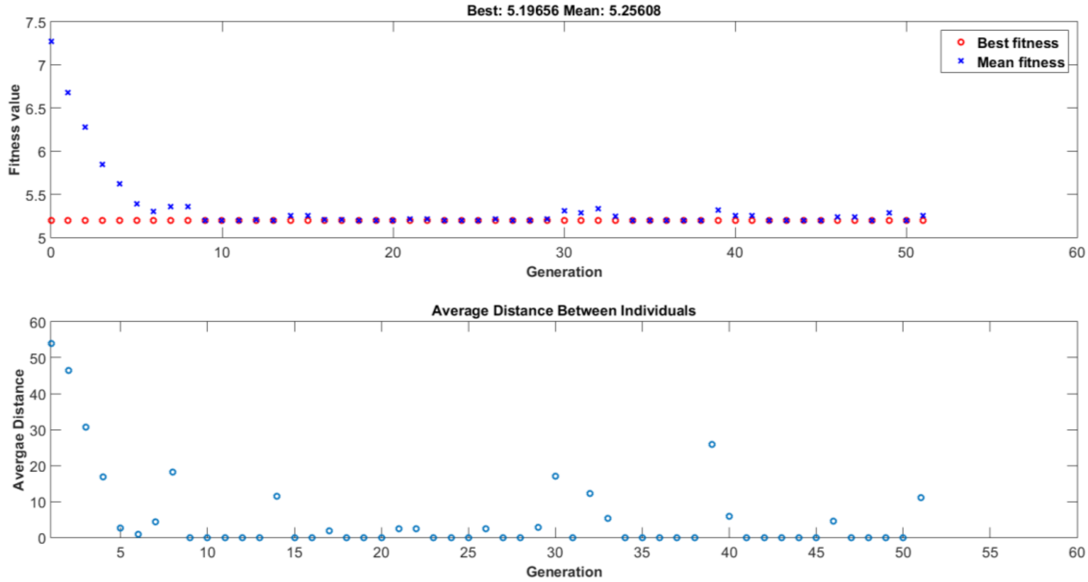

Figure 7.16. The second joint's optimization result (crossover = 0.5 and mutation rate = 0.02).

Figure 7.17 is shown the result of optimization with 0.01 mutation rate and 0.8 crossover fraction for the second joint. The result of optimization is 5.19776 seconds. The algorithm approached the result in the 5th generation and reached the result in the 11th generation. The average distance between individuals varies slightly between generations.

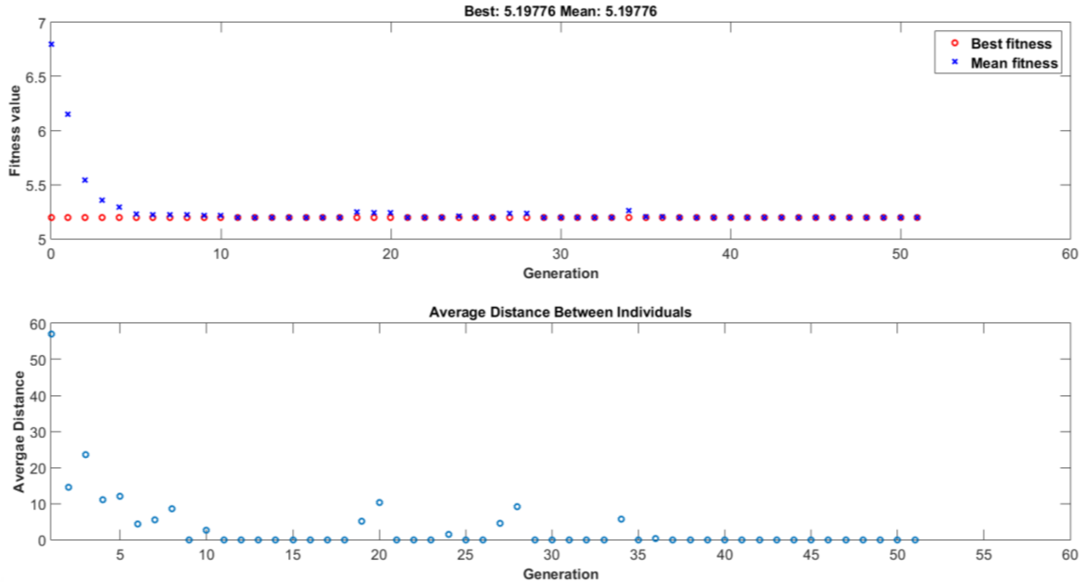

Figure 7.17. The second joint's optimization result (crossover = 0.8 and mutation rate = 0.02).

The algorithm was run for the second joint with 0.02 mutation rate and different crossover fraction, and the results obtained are given in Table 7.6. The second joint gave almost the same result at all crossover fraction.

Table 7.6. The second joint algorithm results with 0.02 mutation rate.

Crossover Fraction	Mutation Rate (0.02)
0.2	5.20255 sec
0.5	5.19656 sec
0.8	5.19776 sec

The optimization studies were carried out for the third joint with a 0.02 mutation rate and different crossover fraction. The results obtained are given below.

The algorithm was run for the third joint with 0.2 crossover fraction and 0.02 mutation rate. When Figure 7.18 is examined, the result is reached in the 20th generation, although the result is approached in the 7th generation. The average distance between individuals, a difference is observed between generations. The algorithm eventually found 4.92415 seconds.

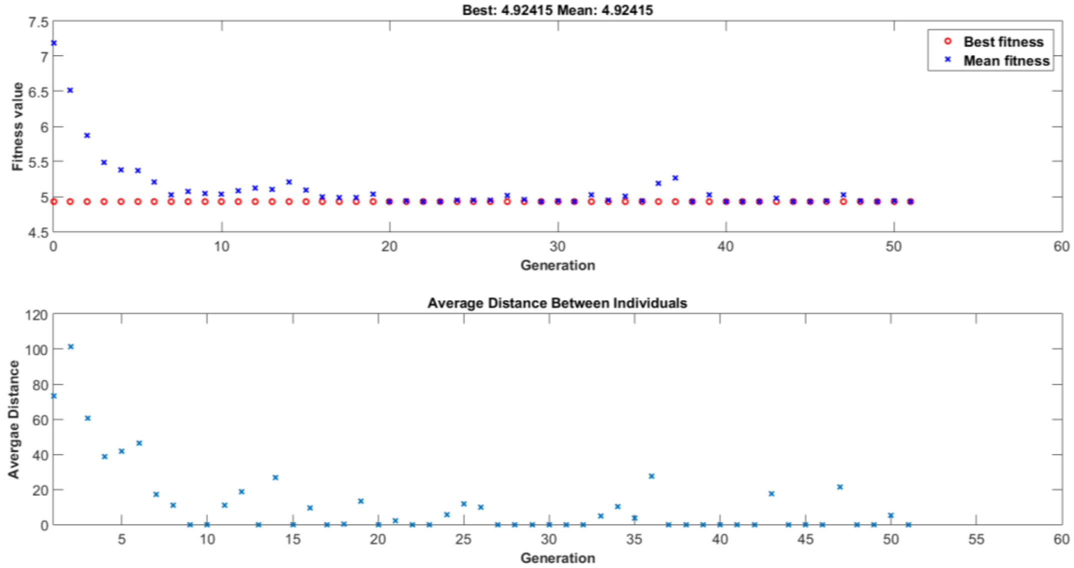

Figure 7.18. The third joint’s optimization result (crossover = 0.2 and mutation rate = 0.02).

In Figure 7.19, the result of the algorithm operated at 0.5 crossover fraction and 0.02 mutation rate for the third joint given. When the figure is examined, there are fluctuations between best fitness and mean fitness. It differs the distance between individuals in between generations. The algorithm executed for the third joint with a 0.5 crossover fraction and 0.02 mutation rate calculated the result of 4.94372 seconds.

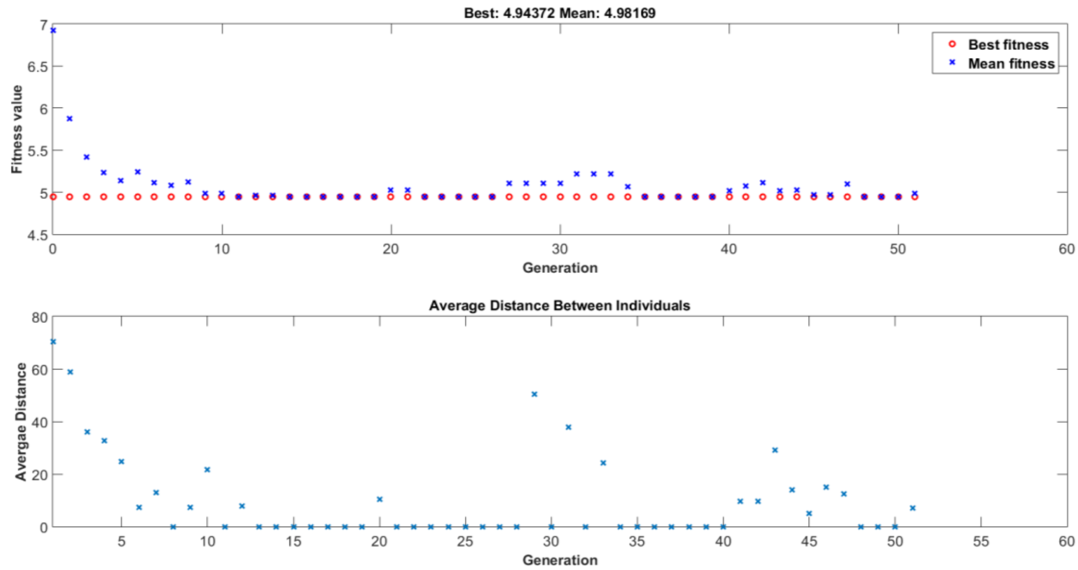

Figure 7.19. The third joint’s optimization result (crossover = 0.5 and mutation rate = 0.02).

The result of the last optimization with 0.8 crossover fraction and 0.02 mutation rate of the third joint is given in Figure 7.20. When the figure is examined, it is seen that the result was caught in the 6th generation, but there are fluctuations in the next generations. As a result of optimization, 4.94527 seconds were found.

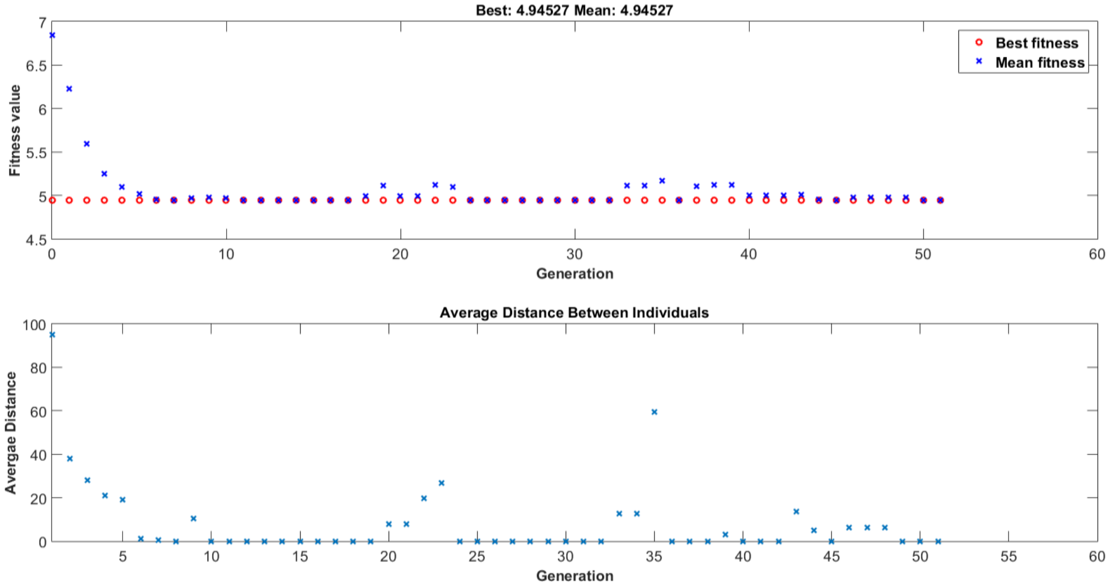

Figure 7.20. The third joint's optimization result (crossover = 0.8 and mutation rate = 0.02).

Table 7.7 gives the results of the algorithm that was run at the rate of 0.02 mutation and different crossover fraction for the third joint. The results seem to be close to each other.

Table 7.7. The third joint algorithm results with 0.02 mutation rate.

Crossover Fraction	Mutation Rate (0.02)
0.2	4.92415 sec
0.5	4.94372 sec
0.8	4.94527 sec

The results obtained from the optimizations for each joint are given in Table 7.8. The results are obtained so that the robot can move from the previously determined point P_0 to P_1. When the results are examined, it is seen that the best result for the first joint is obtained with 0.01 mutation rate and 0.2 crossover fraction. However, 0.01 mutation rate, 0.8 crossover fraction and 0.02 mutation, 0.2 crossover fraction are very close to the best result in. When the results of the algorithm performed for the second joint are examined, it is seen that the best result is obtained with 0.02 mutation rate and 0.5 crossover fraction. It is seen that the results of the remaining optimizations are close to each other except the optimization performed with 0.01 mutation rate and 0.5 crossover fraction. The operators of the algorithm that give the best results in the studies for the third joint are 0.01 mutation rate and 0.2 crossover fraction.

Table 7.8. The result of optimizations with different mutation rates and different crossover fractions.

Crossover Fraction	**Mutation Rate of the First Joint**		**Mutation Rate of the Second Joint**		**Mutation Rate of the Third Joint**	
	0.01	**0.02**	**0.01**	**0.02**	**0.01**	**0.02**
0.2	4.24358 sec	4.24654 sec	5.21699 sec	5.20255 sec	4.90852 sec	4.92415 sec
0.5	4.2516 sec	4.28907 sec	5.27448 sec	5.19656 sec	4.94927 sec	4.94372 sec
0.8	4.24369 sec	4.28733 sec	5.20639 sec	5.19776 sec	4.91203 sec	4.94527 sec

As a result of optimization, if the joints are moved in sequence, the robot arm completes the movement in a total of 14,346 seconds, when the best result for each joint is considered. Total motion time is calculated under the constraint conditions of servo motors used in the model robot arm.

For experimental applications, the robot arm model produced in 3d printer is used. In experimental studies, the robot arm was moved at desired points and the position of the end effector was detected by ultrasonic sensors. The experiment set is given in Figure 7.21.

The robotic arm was controlled by Matlab using the Arduino mega 2560 control card. The data from the sensors were instantly transferred to the Matlab environment with another Arduino mega 2560 card. Three sensors were used to detect the position p_x, p_y and p_z. The schematic representation of the experimental setup is shown in Figure 7.22. Since the sensor data is read instantly from the serial port, a separate control card is used. The joints of the robot model were moved in sequence and their movement times and their position in the Cartesian space are shown in the interface prepared with the Matlab / GUI toolbox. The GUI is shown in Figure 7.23. As a result of experimental studies, the target point in Cartesian space was reached with small error rate.

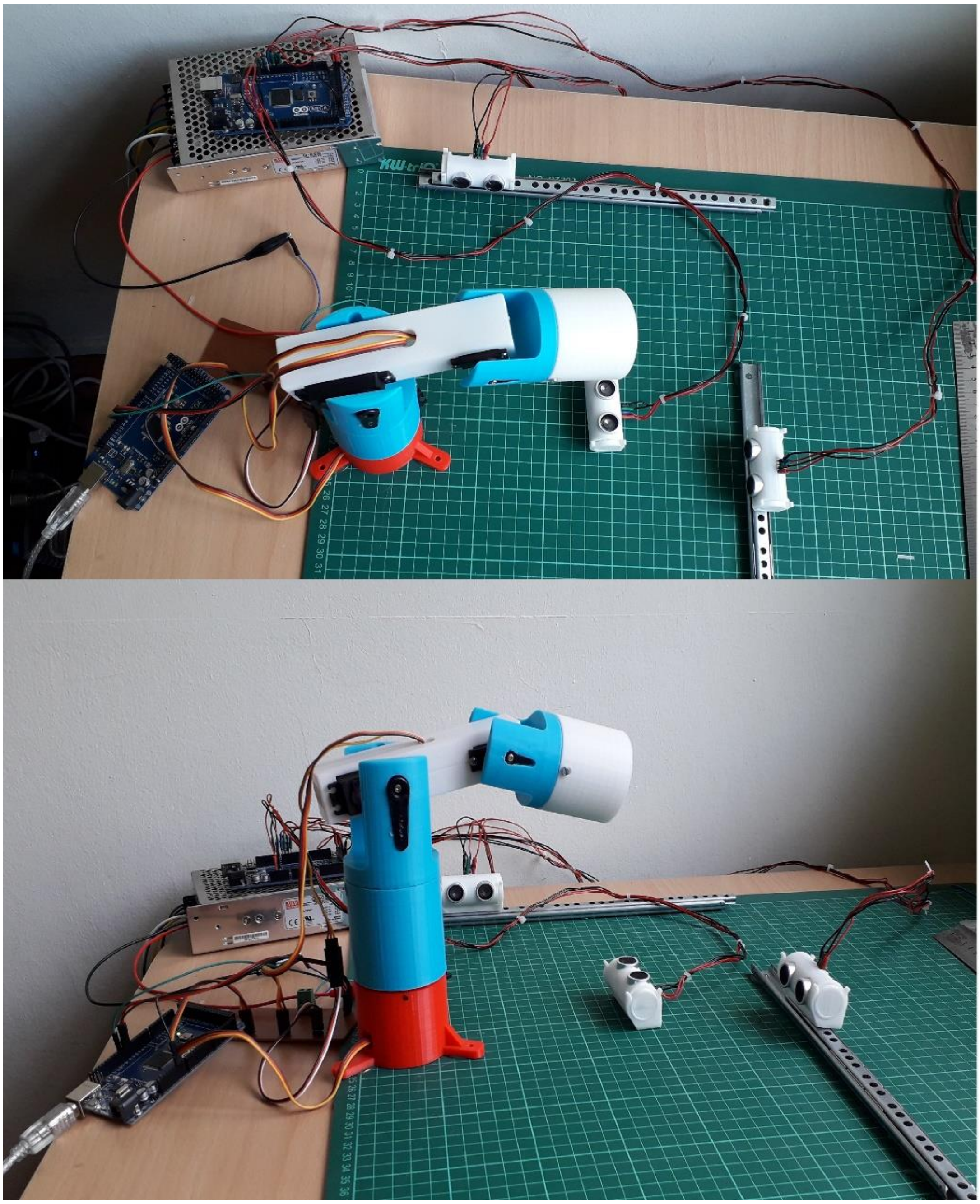

Figure 7.21. The experimental setup.

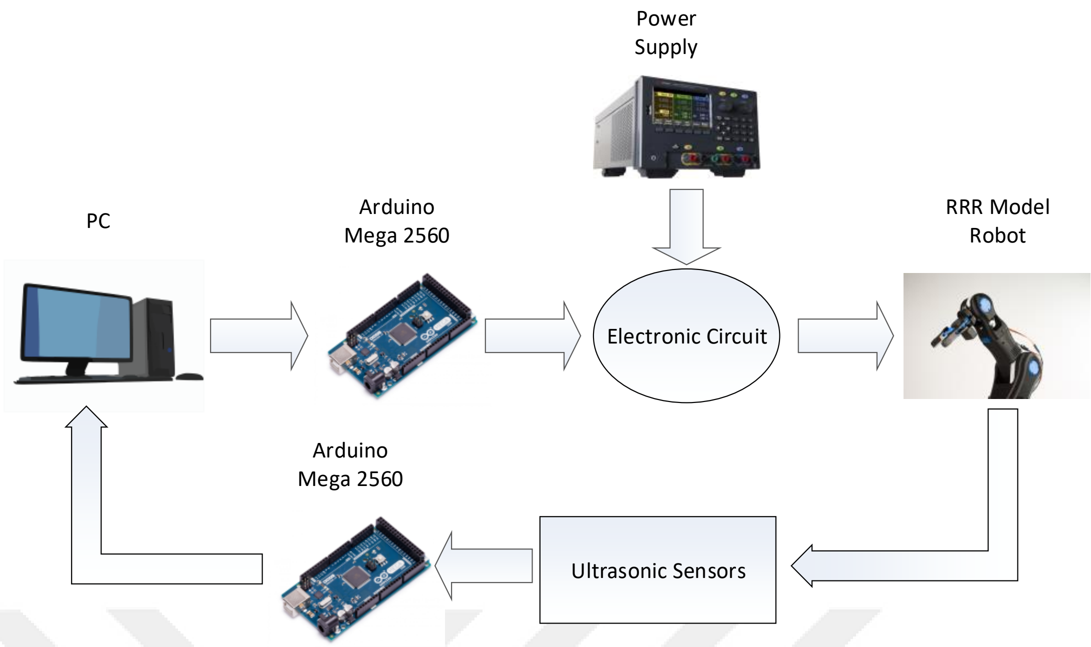

Figure 7.22. The schematic representation of the experimental setup.

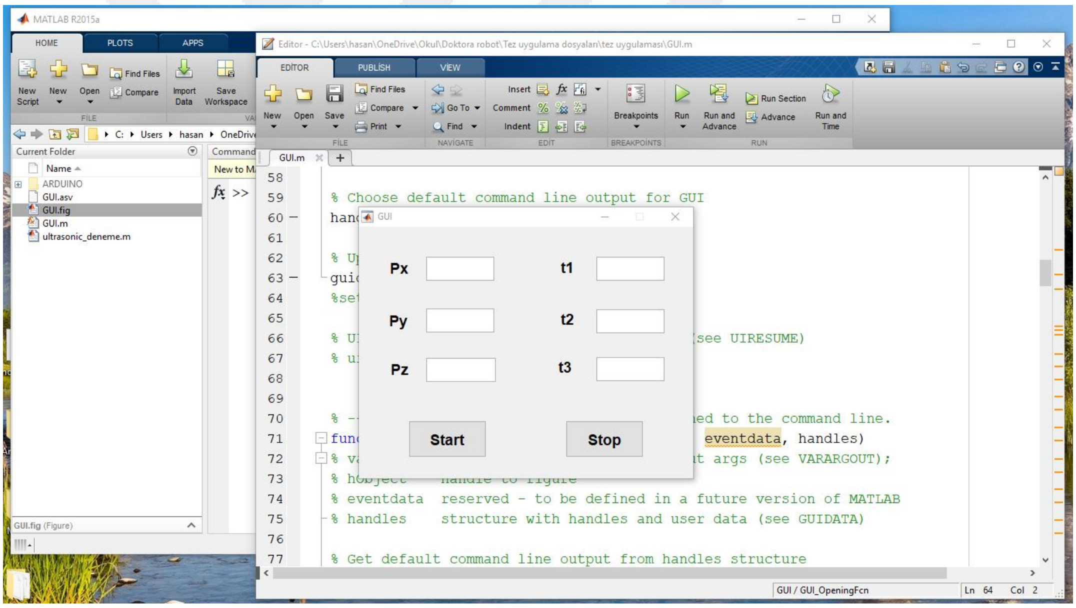

Figure 7.23. The interface prepared with the Matlab / GUI toolbox.

8. CONCLUSION

In this study, path planning of a robot model for RRR joint structure is made in joint space with Genetic Algorithm. The robot model consists of three rotating joints because the first three joints in the industrial robot arms determine the position in the Cartesian space and the last three joints determine the orientation. Since it is possible to plan a path with a certain position in the joint space, a robot model is used to represent the first three joints. Then, the quintic path polynomial was created for the points indicated in Cartesian space and time optimization was made with GA. The constraint conditions of the objective function were the velocity and acceleration values of the servo motors used in the joints. It has been found out how soon each joint will complete the given movement. The data obtained as a result of optimization was applied by experimental setup.

A result of optimization with 0.01 and 0.02 mutation ratio for the first joint, findings obtained in experimental studies using the constraint conditions obtained are given in Table 8.1. Except for minor differences, there are similarities between experimental studies and optimization results.

Table 8.1. The result for the first joint.

Crossover Fraction	**0.01 Mutation Rate of the First Joint**		**0.02 Mutation Rate of the First Joint**	
	Simulation result time (sec)	**Experimental result time (sec)**	**Simulation result time (sec)**	**Experimental result time (sec)**
0.2	4.24358	4.2505	4.24654	4.2556
0.5	4.2516	4.2597	4.28907	4.2954
0.8	4.24369	4.2502	4.28733	4.2982

Table 8.2 provides information on optimization and experimental studies for the second joint. The table contains the results obtained with different mutation rates and crossover fraction. There are small differences between the results. Although motors with the same characteristics are used in the first joint and the second joint, the time values are different because the second joint travels more according to the path.

Table 8.2. The result for the second joint.

Crossover Fraction	0.01 Mutation Rate of the Second Joint		0.02 Mutation Rate of the Second Joint	
	Simulation result time (sec)	Experimental result time (sec)	Simulation result time (sec)	Experimental result time (sec)
0.2	5.21699	5.2212	5.20255	5.2198
0.5	5.27448	5.2901	5.19656	5.2012
0.8	5.20639	5.2195	5.19776	5.2054

The results of the third joint are given in Table 8.3. A smaller servo motor was used in the third joint due to avoiding weight. The constraint conditions of the motor are different from other servo motors. When the results are examined, it is seen that the experimental studies and simulation results are almost the same.

Table 8.3. The result for the third joint.

Crossover Fraction	0.01 Mutation Rate of the Third Joint		0.02 Mutation Rate of the Third Joint	
	Simulation result time (sec)	Experimental result time (sec)	Simulation result time (sec)	Experimental result time (sec)
0.2	4.90852	4.9152	4.92415	4.9354
0.5	4.94927	4.9598	4.94372	4.9601
0.8	4.91203	4.9225	4.94527	4.9587

As a result of the experiments, except for small deviations, values close to the optimization result were found. The reasons for the small differences are the production accuracy of the 3d model and the measurement accuracy of the sensors, etc. As a result of optimization, minimum value of working time of each joint was found. Compared with several literature, in reference literature [7], the optimization are realized for the shortest path with GA, PSO and ant colony algorithm. Optimization is made by intervening the path of the robot arm. In the reference literature [12], time optimization was made using PUMA560 industrial robot arm's path equations. In this study, a model that can be applied to any robot arm is created by time optimization on a designed industrial robot arm. In conclusion, the shortest working time for each robot joint was found in this study and an optimization model was created for the shortest working time for a designed robot arm.

8.1. Future Work

This thesis includes the study of time optimization of the orbital planning of industrial robotic arms by genetic algorithms. A lot of academic research is being done on industrial robotic arms, which became even more important with the Industrial 4.0 revolution. In this study, it contains some insights and directions for future academic research. As a result of the information and findings obtained, some suggestions are made for future research.

First, this thesis was conducted on a model robot arm. A model including time optimization was developed. This model can be tested on a robotic arm used in the industry provided that sufficient technical facilities are provided. As a result of the test, the efficiency of the model can be investigated. The results of the research can be compared with the results of this thesis.

Second, the generated time optimization model was tested when the robot arm was under no-load conditions. New tests can be performed while the model is under load. Tests under load can be compared with unloaded model tests performed in this study.

Third, the tests can be repeated with different optimization methods. In this study, genetic algorithms were used for testing. Experiments can be performed on the model robot arm using different optimization methods such as PSO, ant colony, gray wolf and performance data can be obtained by comparing the results.

Finally, the subject of this thesis can be extended with future studies. Future studies may reveal more advanced time-dependent optimization models. Due to improvements in optimization methods, the movement times of the robot arm model can be shortened. The model created in this study can be developed by testing with future academic research.

REFERENCES

1. Küçük, S. and Bingül, Z., 2004. The Inverse Kinematics Solutions of Industrial Robot Manipulators, IEEE International Conference on Mechatronics, İstanbul, Proceedings of the IEEE International Conference on Mechatronics, 1, 274–279.

2. Milenkovic, V. and Huang, B., 1983. Kinematics of Major Robot Linkages, Robot. Int. SME, 2, 16–31.

3. Fang, J., Mei, T., Chen, J. and Zhao J., 2014. An Iteration Method for Inverse Kinematics of Redundancy Robot, IEEE International Conference on Mechatronics and Automation, Tianjin, China, Proceedings of 2014 IEEE, 1005–1010.

4. Deshpande, V. and George, P.M., 2014. Kinematic Modelling and Analysis of 5 DOF Robotic Arm, International Journal of Robotics Research and Development, 2, 4, 1–8.

5. Švejda, M. and Čechura, T., Interpolation method for robot trajectory planning, International Conference on Process Control, Štrbske Pleso, Slovakia, Proceedings Book, 1, 406–411.

6. Wu, K., Krewet, C., Bickendorf, J. and Kuhlenkoetter, B., 2017. Dynamic performance of industrial robot with CNC controller, Robotics and Computer-Integrated Manufacturing, 000, 1, 1–6.

7. Jin, X., Kang, J., Zhang, J. and Yang, X., 2016. Trajectory Planning of a Six-DOF Robot Based on a Hybrid Optimization Algorithm, 9th International Symposium on Computational Intelligence and Design, Hangzhou, China, Proceedings Book, 148-151.

8. Yan, L., Ya, L. and Xincheng T., 2019. Trajectory and velocity planning of the robot for sphere-pipe intersection hole cutting with single-Y welding groove, Robotics and Computer Integrated Manufacturing, 56, 244–253.

9. Duque, D. A., Prieto, F. A. and Hoyos, J. G., 2019. Trajectory generation for robotic assembly operations using learning by demonstration, Robotics and Computer Integrated Manufacturing, 57, 292–302.

10. Xuan, G. and Shao, Y., 2018. Reverse-driving Trajectory Planning and Simulation of Joint Robot, International Federation of Automatic Control, 17, 51, 384–388.

11. Bingül, Z. and Küçük, S., 2009. Robot Kinematiği, First Edition, Birsen Publisher, İstanbul, Turkey.

12. Craig, J. J., 2005. Introduction to Robotics Mechanics and Control, Third Edition, Peasron Prentice Hall, New Jersey, U.S.A.

13. Wang, L. and Chen, C., 1991. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators, IEEE Transactions on Robotics and Automation, 4, 7, 489–499.

14. Tan, G. Z. and Hu, S. Y., 2002. Real-time accurate hand path tracking and joint trajectory planning for industrial robots (II), Journal of Central South University of Technology (English Edition), 4, 9, 273–278.

15. Zhang, J., Meng, Q., Feng, X. and Shen, H., 2018. A 6-DOF robot-time optimal trajectory planning based on an improved genetic algorithm, Robotics and Biomimetics, 3, 5, 3–9.

16. Jonasson, J. and Norgren, E., 2016. Investigating a Genetic Algorithm - Simulated Annealing Hybrid Applied to University Course Timetabling Problem, Degree Project in Technology, Stockholm, Sweden.

17. Holland, J., 1975. Adaptation in Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, Michigan.

18. Fogel, D. B., 1995. Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. Wiley-IEEE Press.

19. Schmitt, L. M., Nehaniv, C. L. and Fujii, R. H., 1998. Linear analysis of genetic algorithms, Theoretical Computer Science, 200, 101–134.

20. Goldberg, D. E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, U.S.A.

21. Urso, A., Fiannaca, A., La Rosa, M., Ravì, V. and Rizzo, R., 2018. Data mining: Prediction methods, Encyclopedia Bioinformatics Computational Biology, 1, 413–430.

22. Mitchell, M., 1999. An Introduction to genetic algorithms, First MIT Press paperback edition, A Bradford Book The MIT Press, Cambridge, Massachusetts.

23. Pulat, M. and Deveci Kocakoç, I., 2016. Gezgin Satıcı Probleminin Genetik Algoritmalar Kullanarak Çözümünde Çaprazlama Operatörlerinin Örnek Olaylar Bazlı İncelenmesi, Yöneylem Araştırması Endüstri Mühendisliği 36. Ulusal Kongresi, Bornova.

24. Siriwardene, N. R. and Perera, B. J. C., 2006. Selection of genetic algorithm operators for urban drainage model parameter optimisation, Mathematical Computer Modelling, 44, 415–429.

25. Xu, H., 2015. Solver Tuning with Genetic Algorithms, Dosctor of Philosophy, University of Dundee.

26. Emel, G. G. and Taşkın, Ç., 2002. Genetik Algoritmalar ve Uygulama Alanları, Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 1, 21, 129–152.

27. İşçi, Ö. and Korukoğlu, S.,2003. Genetik Algoritma Yaklaşım ve Yöneylem Araştırmasında Bir Uygulama, Yönetim ve Ekonomi, 2, 10, 191–208.

28. Çalişkan, F., Yüksel, H. and Dayik, M., 2016. Genetik Algoritmaların Tasarım Sürecinde Kullanılması,” SDÜ Teknik Bilimler Dergisi, 6, 2, 21–27.

29. Jha, N. K. and Jha, P., 1996. Mathematical programming formulation and optimum design of robot control system, Applied Mathematical Modelling, 11, 20, 853–865.

30. Bugday, M. and Karali, M., 2019. Design optimization of industrial robot arm to minimize redundant weight, Engineering Science and Technology, an International Journal, 1, 22, 346–352.

31. Gu, J., Wensing, M., Uhde, E. and Salthammer, T., 2018. Characterization of particulate and gaseous pollutants emitted during operation of a desktop 3D printer, Environment International, 123, 476–485.

32. “Assembled.jpg (1024×748).” [Online]. Available: http://www.chrisherring.net /wp-content/uploads/2014/02/DeltaRobotResources/Assembled.jpg. [Accessed: 10-Mar-2019].

33. Hentz, G., Charpentier, I. and Renaud, P., 2016. Higher-order continuation for the determination of robot workspace boundaries, Comptes Rendus Mecanique, 2, 344, 95–101.

34. Abu-Dakka, F. J., Rubio, F., Valero, F. and Mata, V., 2013. Evolutionary indirect approach to solving trajectory planning problem for industrial robots operating in workspaces with obstacles, European Journal of Mechanics A/Solids, 42, 210–218.

35. Pusey, J., Fattah, A., Agrawal, S. and Messina, E., 2004. Design and workspace analysis of a 6-6 cable-suspended parallel robot, Mechanism and Machine Theory, 7, 39, 761–778.

36. Kucuk, S. and Bingul, Z., 2005. Robot workspace optimization based on a novel local and global performance indices, IEEE International Symposium on Industrial Electronics, Dubrovnik, Croatia, 1593–1598.

37. Laribi, M. A., Romdhane, L. and Zeghloul, S., 2007. Analysis and dimensional synthesis of the DELTA robot for a prescribed workspace, Mechanism and Machine Theory, 7, 42, 859–870.

38. Tsai, Y. C. and Soni, A. H., 1983. An algorithm forthe workspace of a general n-r robot, Journal of Mechasisms, Trasnmissions, and Automation in Design, 1, 105, 126–132.

39. Borst, C., Zacharias, F. and Hirzinger, G., 2007. Capturing robot workspace structure : Representing robot capabilities, IEEE/RSJ International Conference

on Intelligence Robots and Systems, San Diego, CA, U.S.A., Proceedings Book, 3229–3236.

40. Cao, Y., Lu, K., Li, X. and Zang, Y., 2011. Accurate Numerical Methods for Computing 2D and 3D Robot Workspace, International Journal of Advanced Robotic Systems, 6, 8, 1–13.

41. Yang, W. and Jian, R., 2019. Research on Intelligent Manufacturing of 3D Printing/Copying of Polymer, Advanced Industrial and Engineering Polymer Research.

CURRICULUM VITAE

Name and Surname : Hasan DEMİR

Address :Aksaray University Ortaköy Vocational School

Email :hasandemir@aksaray.edu.tr
hasanfe_88@hotmail.com

Education

Bachelor : Kocaeli University, 2006-2011
Anadolu University, 2007-2017

Master : Karabük University, 2012-2015

PROFESSIONAL EXPERIENCE AND AWARDS

1. Karabük University /Research Assistant (2012-2016)
2. Aksaray University / Lecturer (2016-)
3. First place, 10th International METU Robot Days, Sumo Robot Category, 2013.
4. Second Place, 10th International METU Robot Days, Mini Sumo Robot Category, 2013.
5. Third Place, 10th International METU Robot Days, Sumo Robot Category, 2013.
6. Best vehicle design award in TUBITAK Electromobile Category-2014

PUBLICATIONS, PRESENTATIONS AND PATENTS PRODUCED FROM THESIS

1. Demir, H., Sarı, F. and Tolun, M.R., 2019. Review for Path Planning for Robots And Its Application, Academic Studies In Engineering, Architecture, Planning And Design Sciences-2019/2 Ivpe Press Cetinje, Montenegro.