جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية الهندسة / قسم الهندسة الميكانيكية

نمذجة عملية ومحاكاة لتوليد مسار مستوي باستخدام روبوت مناور ذو أربع درجات حرية.

رسالة مقدمة الى جامعة بابل/ كلية الهندسة / قسم الهندسة الميكانيكية كجزء من متطلبات دراسة ماجستير في الهندسة الميكانيكية

(ميكانيك تطبيقي)

من قبل الطالبة

هاجر عبد الستار علي
بكالوريوس هندسة ميكانيك

بإشراف الدكتور
حاتم هادي عبيد

1444 ***2023***

Republic of Iraq
Ministry of Higher Education
and Scientific Research
University of Babylon
College of Engineering
Mechanical Engineering Department

Simulation and Experimental Modeling of Planner Trajectory Generation Using Four Degree of Freedom Robot Manipulator

A thesis

Submitted to the College of Engineering of the University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master of Science in Engineering/Mechanical Engineering /Applied Mechanics

Prepared by

HAJAR ABD AL-SATTAR ALI

B.SC. Mechanical Engineering

Supervised by

Prof. Dr. Hatem Hadi Obeid

2023 ***1444***

Abstract

Robots are one of the most significant scientific innovations, with applications in all industrial, medical, military, and other fields. The importance of robots lies in their ability to complete tasks with great accuracy, as well as to repeat tasks tirelessly and to the fullest. In this work (4 DOF) robot arm manipulator is modelled for the purpose of performing kinematics and kinetics analysis . This study was conducted using (SOLIDWORKS, ADAMS and LABVIEW) software's. Theoretical work deals with generation of two paths using the SOLIDWORKS program after completing a design that fully matches the desired arm movement according to the desired path. The results obtained were taken into account four times (2, 4, 6 and 8 seconds) and the parameters of (kinematics and kinetics) were studied, which are (Angular Displacement, Angular Velocity, Angular Acceleration, Torque, Power consumption and Moment of Inertia) to obtain the dynamic behavior of the robot at each time interval. The experimental work involved the implementation of the motion of the proposed two paths on the real robot at each time and with the help of LABVIEW program. According to the current study, the results obtained from time(2s) are incompatible with the behavior of the robot as in times (4, 6, and 8) seconds as illustrated in many figures of

theoretical and experimental study and for all the parameters used (speed, acceleration, torque, and power consumption), so we advise not to use it when studying the behavior of these parameters. The results of the remaining times (4, 6, and 8) were also found to be identical in terms of the robot's dynamic behavior. Where the highest error was found at time (4 seconds) and then decreases gradually at time (8 seconds) for all the parameters that studied.

الخلاصة

تعتبر الروبوتات من أهم الابتكارات العلمية ، ولها تطبيقات في جميع المجالات الصناعية والطبية والعسكرية وغيرها. تكمن أهمية الروبوتات في قدرتها على إكمال المهام بدقة كبيرة أيضًا. في هذا العمل تم تصميم روبوت مناور ذو اربع درجات حرية من اجل دراسة السلوك الديناميكي. تمت الدراسة عن طريق استخدام برامج (SOLIDWORKS, ADAMS and LABVIEW). تضمن الجانب النظري توليد مسارين باستخدام برنامج SOLIDWORKS بعد الانتهاء من تصميم يتطابق تمامًا مع حركة الذراع المطلوبة وفقًا للمسار المطلوب. تم أخذ النتائج التي تم الحصول عليها في الاعتبار لأربعة ازمان (٢ و ٤ و ٦ و ٨ ثوانٍ) وتم دراسة معايير (الحركة)، وهي (الإزاحة الزاوية والسرعة الزاوية والتسارع الزاوي وعزم الدوران واستهلاك الطاقة وعزم القصور الذاتي) للحصول على السلوك الديناميكي للروبوت في كل فترة زمنية. تضمن العمل التجريبي تنفيذ حركة المسارين المقترحين على الروبوت الحقيقي في كل مرة وبمساعدة برنامج LABVIEW. وفقًا للدراسة الحالية، فإن النتائج التي تم الحصول عليها من الزمن(s٢) لا تتوافق مع سلوك الروبوت كما هو الحال في الأوقات (٤ و ٦ و ٨) ثانية كما هو موضح في العديد من أرقام الدراسة النظرية والتجريبية ولجميع البارامترات المستخدمة (السرعة والتسارع وعزم الدوران واستهلاك الطاقة)، لذلك ننصح بعدم استخدامه عند دراسة سلوك هذه المعايير. كما تم العثور على نتائج الأوقات المتبقية (٤ و ٦ و ٨ ثوان) متطابقة من حيث السلوك الديناميكي للروبوت. حيث تم العثور على أعلى خطأ في الوقت المناسب (٤ ثوانٍ) ثم ينخفض تدريجيًا في الوقت المناسب (٨ ثوانٍ) لجميع المعلمات التي تمت دراستها.

List of content

List of Figures

List of Symbols

Symbol	Definition	unit
a_i	Link length	**mm**
d_i	Link twist	**mm**
θ_i	Joint angle	**Deg.**
T_i^{i-1}	Transformation Matrix	**-**
D_{rxm}	Direct Transmission Matrix	**-**
R_{x_i,α_i}	Rotation Matrix	**-**
g	Gravitational acceleration (9.81)	**(m/s2)**
I	Mass moment of inertia	**Kg.mm^2**
J	Jacobian matrix	**-**
J_o	Angular velocity Jacobian matrix	**-**
J_p	Linear velocity Jacobian matrix	**-**
K	Kinetic energy	**J**
U	potential energy	**J**
m	Mass	**Kg**
P	Position vector	**mm**
P_x	X – Coordinate of end effector position	**mm**
P_y	Y – Coordinate of end effector position	**mm**
P_z	Z – Coordinate of end	**mm**

	effector position	
α	Roll	**Deg.**
β	Pitch	**Deg.**
γ	Yaw	**Deg.**
τ_i	Torque Matrix	**-**
ω	Angular velocity	**Deg/sec**
$C(\theta, \dot{\theta})$	Dynamic system Coriolis and centrifugal Torque Matrix	**-**
I_z	The robotic manipulator total mass moment of inertia about Z axis	**Kg.mm^2**

List of Table

List of Abbreviations

DOF	Degree Of Freedom
TCP	Tool Center Point
TT	Tool Trajectory
JT	Joint Trajectory
PPP	Prismatic- Prismatic- Prismatic
RRR	Revolute – Revolute – Revolute
RRP	Revolute – Revolute – Prismatic
PTP	Point To Point
DH	Denavit-Hartenberg
FK	Forward Kinematic
IK	Inverse Kinematic
CAD	Computer Aided Design
3D	Three-Dimensional Graphics

Abstract

Robots are one of the most significant scientific innovations, with applications in all industrial, medical, military, and other fields. The importance of robots lies in their ability to complete tasks with great accuracy, as well as to repeat tasks tirelessly and to the fullest. In this work (4 DOF) robot arm manipulator is modelled for the purpose of performing kinematics and kinetics analysis . This study was conducted using (SOLIDWORKS, ADAMS and LABVIEW) software's. Theoretical work deals with generation of two paths using the SOLIDWORKS program after completing a design that fully matches the desired arm movement according to the desired path. The results obtained were taken into account four times (2, 4, 6 and 8 seconds) and the parameters of (kinematics and kinetics) were studied, which are (Angular Displacement, Angular Velocity, Angular Acceleration, Torque, Power consumption and Moment of Inertia) to obtain the dynamic behavior of the robot at each time interval. The experimental work involved the implementation of the motion of the proposed two paths on the real robot at each time and with the help of LABVIEW program. According to the current study, the results obtained from time(2s) are incompatible with the behavior of the robot as in times (4, 6, and 8) seconds as illustrated in many figures of

theoretical and experimental study and for all the parameters used (speed, acceleration, torque, and power consumption), so we advise not to use it when studying the behavior of these parameters. The results of the remaining times (4, 6, and 8) were also found to be identical in terms of the robot's dynamic behavior. Where the highest error was found at time (4 seconds) and then decreases gradually at time (8 seconds) for all the parameters that studied.

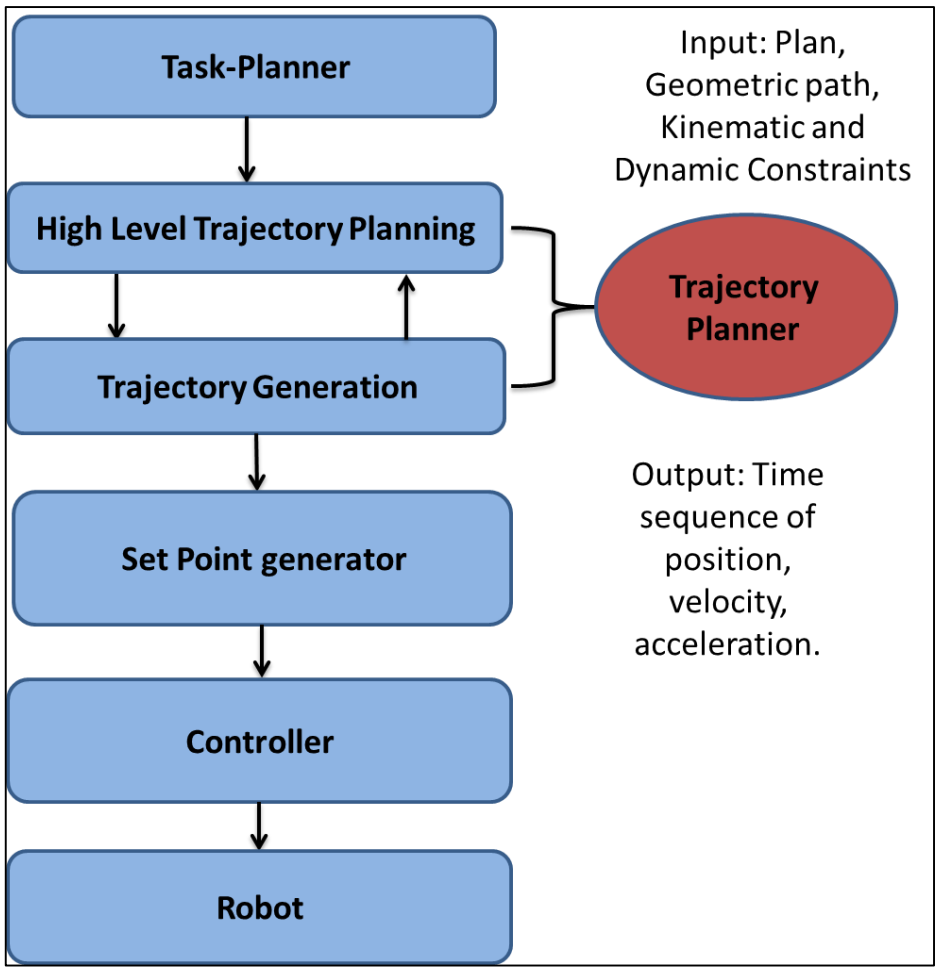
Chapter One

Introduction

1.1. General

Robotics is a multidisciplinary branch of engineering that aims to create intelligent machines that can replace humans, primarily in repetitive and precise tasks. Automated robotics include not only robots, but also other systems and devices that work in harmony with the robots. Robots have many applications and can only serve their owners by being carefully programmed and managed[1]. Robots have become increasingly popular in recent years due to their capacity to complete tasks more rapidly, accurately, and dependably than humans can. Robots are utilized in the majority of modern sectors, as they have been shown to improve productivity, precision, and overall product quality [2]. Robots used in manufacturing are often computer numerically controlled (CNC) machines. These machines are developed for use in industry to perform specialized material removal, and they are a component of a much larger collection of operating resources like (Milling, Coating, Welding or Spray Painting). Typically, a manufacturing robotic is an arm manipulator mounted to a stationary base and designed to perform a set of tasks within a specific work cell [3]. The challenge of planning trajectories is one of the most fundamental issues in robotics. It involves establishing a time motion law for a particular geometric path that fulfills specific restrictions imposed for the trajectory. The purpose of producing input for the manipulator's control system during trajectory planning is really to accomplish the goal of executing out a motion.

Figure (1) presents a schematic representation of the automation trajectory planning system's organizational components[4].

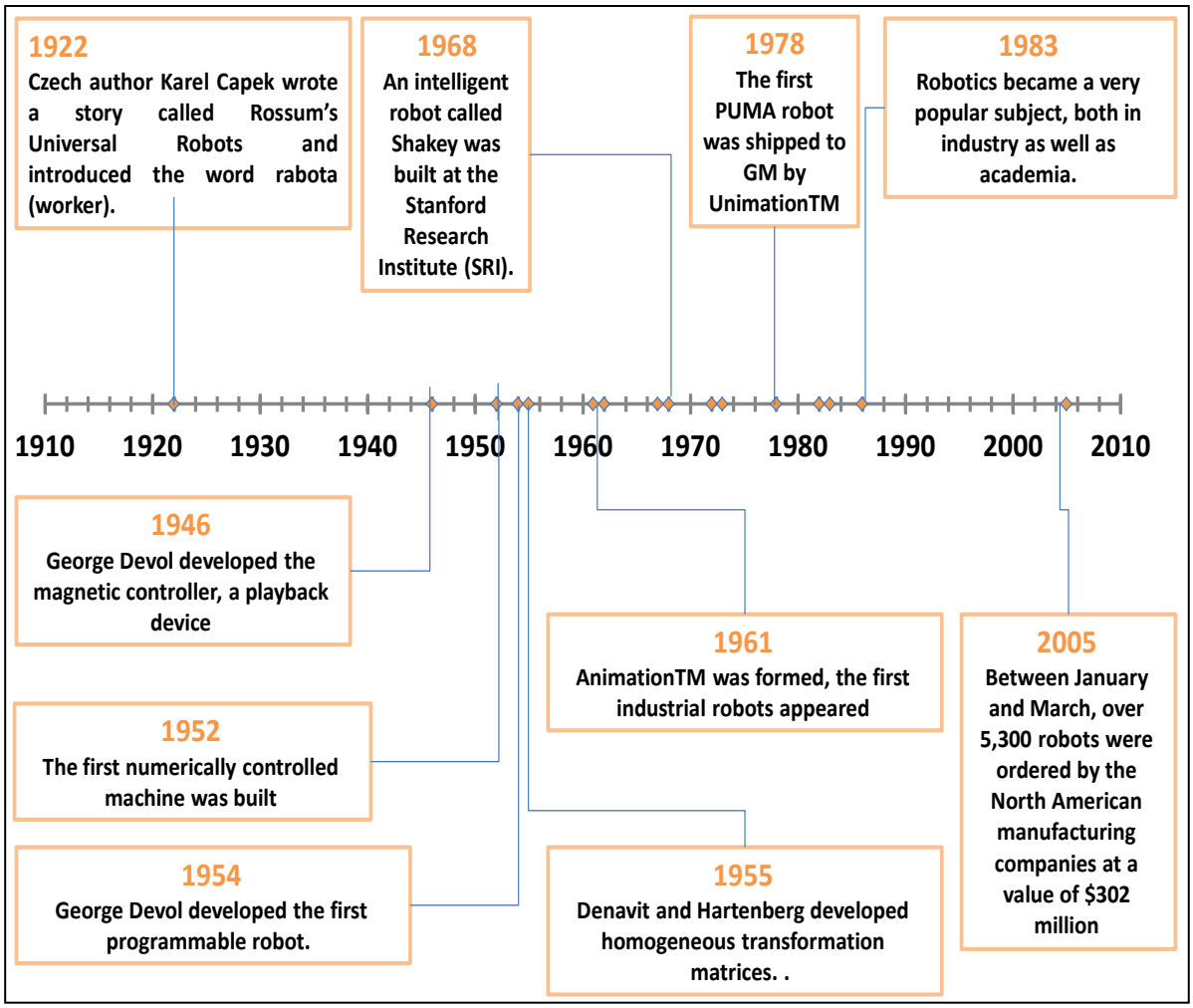


Figure(1.1) Structure of automated trajectory planning system.[4]

1.2. History of Robotics

Automatic machines were developed after World war II, to increase efficiencies, machine tool manufacturers created NC machines to help in factories make higher quality goods. Concurrently, nuclear material manipulation robots with several degrees of freedom have been created. The integration of the NC capacity of the machine tools and the

robot manipulators resulted in the creation of a straightforward robot. **Figure(1.2)** is a review of major events that have affected the industry's direction[1].



Figure(1.2) Robotics: A Brief History.[1]

1.3. Manipulators ' Common Kinematic Arrangements

Nowadays, Manipulators used in industry typically have high degrees of freedom. Many times, these manipulators are categorized according to the position of the first three joints in their arms, include wrist measurements

as a distinct field. Here's a brief overview of the most common arrangements.[5]

1.3.1.Manipulator with Articulation (RRR)

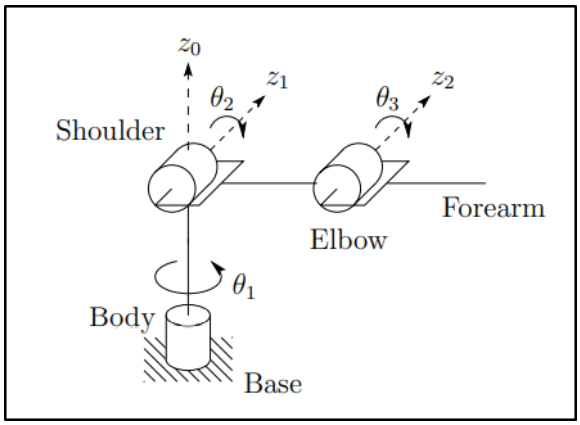
Articulated robot is a robot with rotary joint it is also referred as an anthropomorphic manipulator or a revolute. **Figure(1.3)** below depicts the ABB IRB1400 articulated arm.

Figure(1.3) Articulated manipulator (RRR).[5]

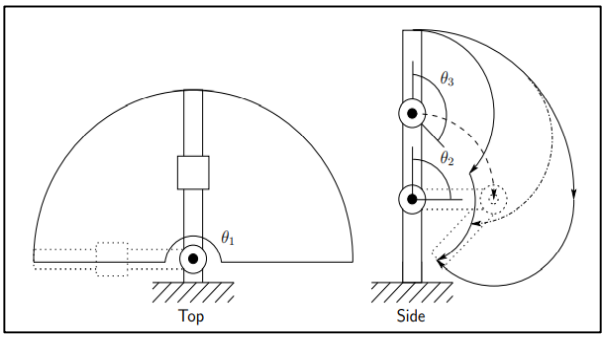
The parallelogram linkage, which can be seen in the Motoman SK16, is one example of a common type of revolute joint design shown in **figure(1.4)** .

Figure(1.4) The Motoman SK16 manipulator.[5]

It's called an elbow manipulator since it's used at the elbow. Elbow manipulator construction, terminology, and working space are depicted in **figures (1.5)** and **(1.6),** respectively[5].



Figure(1.5) Elbow manipulator's structural framework[5].

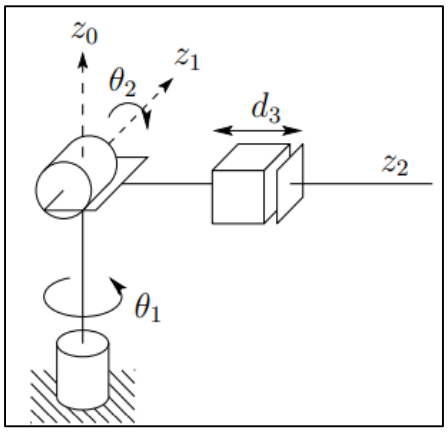


Figure(1.6) The operating area of the elbow manipulator[5].

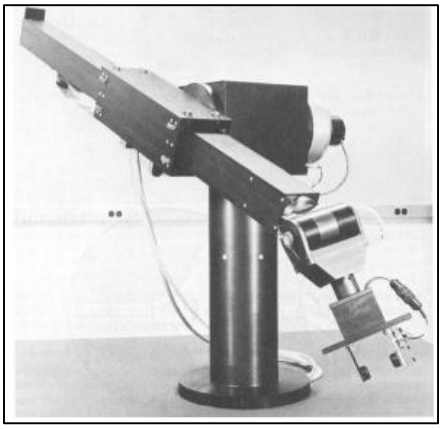
1.3.2. The Spherical-Manipulator (RRP)

A spherical manipulator is a manipulator in which all links perform spherical motion's about a common point it may be built by substituting the revolute robot manipulator's elbow joint with a prismatic joint, as shown in **figure (1.٧)**. **Figure (1.٨)** shows the Stanford Arm, which is

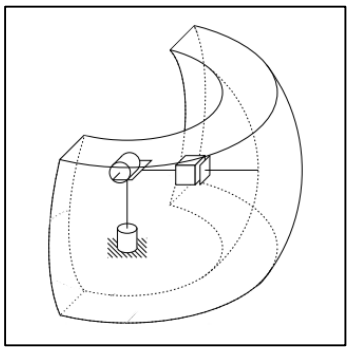
currently one of the most popular spherical robots. The working environment is also shown in **figure (1.٩).**



Figure(1.7)The spherical manipulator[5]



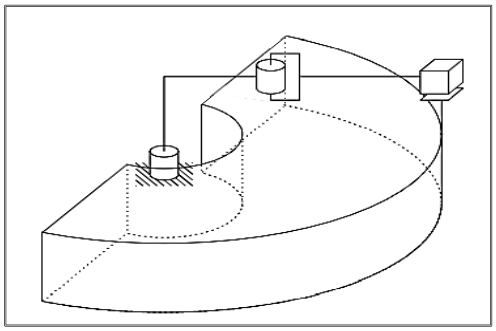
Figure(1.8) The Stanford Arm



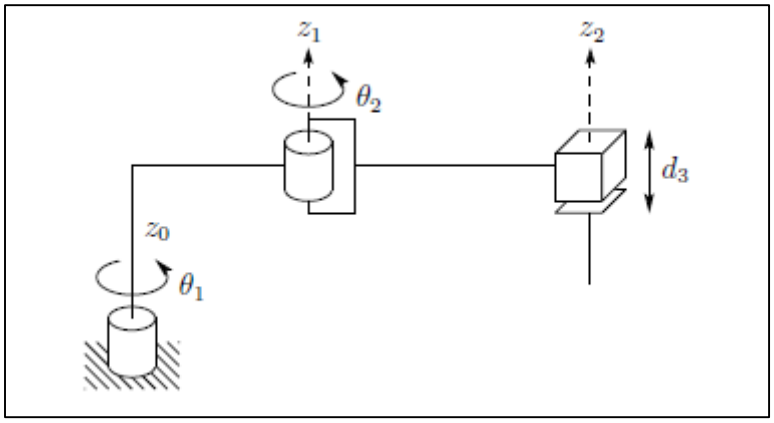
Figure(1.9) The spherical manipulator's workspace[5]

1.3.3. SCARA Manipulator (RRP)

The assisting equipment called the SCARA arm, which stands for for Selective Compliant Articulated Robot for Assembly, as well as the workspace it occupies are seen in **figure(1.10),(1.11)**.



Figure(1.10) The SCARA manipulator's workspace.



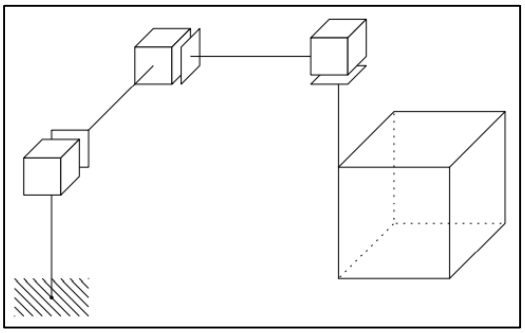
Fig(1.11) SCARA system (Selective and Compliant Articulated Robot for Assembly).[5]

1.3.4.Cartesian Manipulator (PPP)

The simplest kinematic description of a manipulator is that of a Cartesian manipulator, which is characterized by the prismatic nature of its first three joints as shown in **figure(1.12).**

Fig(1.12) Cartesian manipulator.[5]

Cartesian manipulator workspace is also illustrated in **figure(1.13)** below



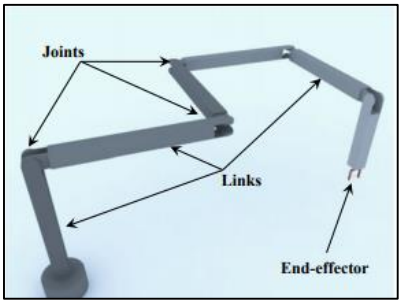
Figure(1.13) Workspace of the Cartesian manipulator.

1.4. Manipulator Component

Robot manipulators have mechanical linkages that are connected by joints. these parts serve to grasp and move objects and can be controlled by a computer[6].

1.4.1. Link

As shown in **figure(1.14)**, an item made of material, such as a beam, rod, or other link, attached to another by joints.

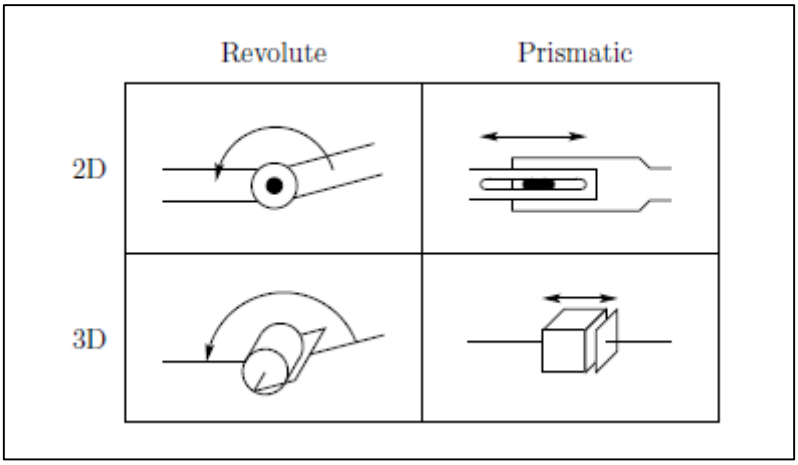


Figure(1.14) Model of a 7-DOF robot manipulator.[6]

1.4.2.Joint

The term "joint" refers to a physical connection between two links that enables them to move relative to one another**.** Both the revolute and prismatic joints are considered to be fundamental types of joints. Relative

rotation is possible at a revolute joint, but relative translation is possible at a prismatic junction. As illustrated in **figure(1.15)** below.



Figure(1.15) Joint types and representation[6].

1.4.3. End-Effector

Is the devise at the end of a robotic arm, designed to interact with the environment. The exact nature of the device depends on the application of the robot.

1.5. Basic Categories of Programming Languages

A robot programming language is used to program all robots. These programming languages are utilized to instruct the robot on where it should travel, how signals should be sent, and how inputs should be read. Currently, industrial robots can be programmed using one of two different approaches. In fact, both methodologies are frequently combined, resulting in a type of programming known as hybrid programming. The two major strategies are briefly outlined here[7].

1.5.1. Online-Programming

Online-programming refers to the process of developing a control program on the onboard computer of a robot by using a joystick or other manual control system, each desired position contributes to the code as a number of coordinates. The main drawbacks of this method are that it is time consuming and it has long production stops.

1.5.2. Offline-Programming

The process involves creating the control software on a separate device (PC, for example). There are two primary methods for accomplishing this goal: modifying code manually with a text editor, or automatically generating code from a modeling environment. An advantage of this technology is that reprogrammed robots can continue manufacturing immediately.

1.6. Robotic Motion Planning

Motion control is simply one of many features needed to enable the use of autonomous systems. Motion planning allows the robot to navigate its environment and complete a task securely. The initial step in creating robot motion is to define the path's geometry. In Cartesian space, a path is a curve (or in joint space). When designing robot mobility, there are three primary challenges that must be met [8].

1.6.1 Path planning

Robots having a high degree of autonomy for application in hazardous areas are considered advanced robotics. Path planning is one of the most exciting areas of study in robotics and automation and it is distinguished by the following characteristics: (Space, undersea, nuclear, military, and

so on).The challenge of path planning for a mobile system can be characterized as follows: finding a motion that is free from collisions between an initial configuration and a final configuration (goal). When the path is planned in an environment that is both static and known, the situation is at its simplest. However, the general problem is concerned with the generation of the motion of at least one robot arm that must adhere to kinematic and dynamic constraints while operating in an environment that is not known a priori.

1.6.2 Trajectory Planning

In order to ensure that the planned motion is really performed by the robot, trajectory planning must generate the standard inputs for the control system. The inputs of a trajectory tracking algorithm are the geometrical path, the dynamic and kinematic constraints of manipulator, and the outputs are the joint trajectory or end-effector trajectory, defined as a sequence of orientation, velocity, and acceleration.

1.6.3 Trajectory Tracking

This section's goal is to map out the control signal that will guarantee the accuracy of the performed trajectories.

1.7. Motion Planning Categorizations

Path planning refers to the process of specifying planned manipulator motion as a series of points in the space (the orientation and position of manipulator tool frame) through which the TCP tip must go, in addition to the space curve in which it traverses. The path is the space curve along which the manipulator hand travels from the starting position to the ending position. There are usually multiple paths that could connect the

two points. The manipulator might be directed to travel either along a straight line joining the two points or along a curved path with a specific radius and arc length. This is known as TCP space planning or Cartesian-space planning (TT). On the other hand, sometimes all that's needed is for the manipulator to be moved along a path where the joints smoothly follow a function trajectory that satisfies the location and orientation requirements of end-effecter at both ends. This type of planning is known as Joint space planning (JT) [9]. as can be seen in **figure (1.16).**

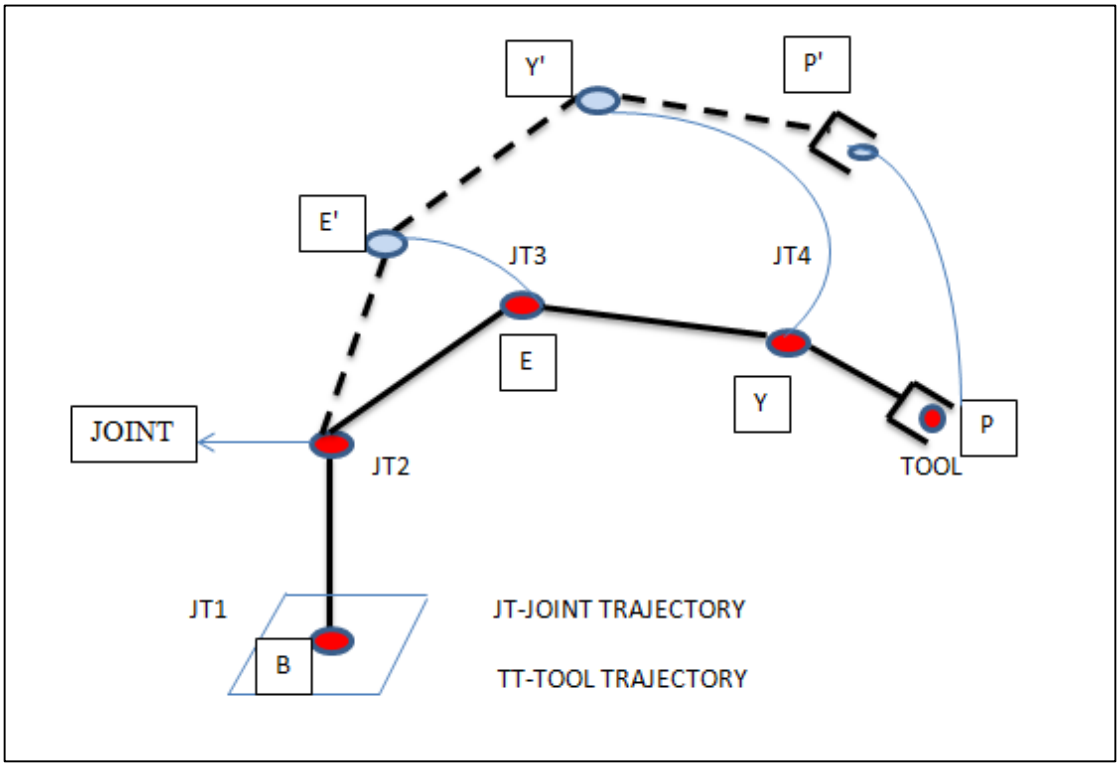


Figure (1.16): Tool and Joint Trajectories (TCP)[9].

1.8 Objectives of the work

The main aim of this thesis is to trace two paths using a four-degree freedom robot arm at four different time intervals and to study the kinetic and kinematic characteristics of the model used. The fundamental objectives of this research will be addressed through.

1- Creating a simulation of the four degree of freedom robot arm using SOLIDWORK program.
2- Construction a prototype of a four degree of freedom robot arm.
3- Writing a LABVIEW code to control the motion of the four degrees of freedom prototype.
4- Studying and analyzing the comparison between the simulation and experimental motion results.

Chapter Two

Literature Review

2.1 Introduction

A literature review was conducted throughout the entire work to acquire the knowledge and skills required to finish it. The chapter's goals included examining, comparing, and explaining the viewpoint and methodology used in earlier studies in addition to determining how those studies and theories might be connected to the project. Understanding theories is crucial because it serves as a guide for all research..

Gouasmi et al., [2012][10]. Described the modeling of the a 2R-robot type. The main duties were to compare two robot configurations that followed the same path and remained in place for the same amount of time, as well as to develop a computational technique to determine the variables of kinematics and dynamics. The theories were verified with SolidWorks, and the robot manipulator simulation was carried out in MATLAB/Simulink. Researcher figured out that the amount of energy lost during the motion of the robot in the first position (elbow down) was less than that of the second posture (elbow up).

Sam Rosidah et al., [2012][11]. Used SOLIDWORKS Soft motion software to design a pick-and-place robotics system. This software was used to create an articulated industrial robotic arm as well as the Robotic Cartesian with interchangeable grippers. The use of SOLIDWORKS 3D CAD software accelerated the design process and improved the quality of the final robot product. SOLIDWORKS 3D CAD software consists of four key components: manual drawing, the part module, the assembly module, and the drawing module. SOLIDWORKS was chosen for the analysis and testing of the pick-and-place industrial robot arm. By

presenting the outcomes of a motion analysis and simulation Xpress analysis of the modeled articulated robot arm part and assembly, the pick and place robotics system was demonstrated. With the results of this research in hand, it was clear that persons working under time constraints can successfully use SOLIDWORKS to draft the specifications for a robotic system.

Iqbal Jamshed et al., [2012][12]. Studied the kinematic models of a 6-DOF robotic arm. The forward kinematic model was constructed utilizing the method of Denavit and Hartenberg (DH) for Robotic arm resting place allocation. Assuming intended a robot's end-effector's location and orientation, the developed model based on inverse kinematics can determine the necessary joints angles. Utilizing MATLAB's Robotics Toolbox, the forward kinematic model was validated. Additionally, an actual robotic arm was used to test the inverse kinematics model. The constructed model allows the robotic arm's end-effector to point within (0.5cm) of the target coordinates, as evidenced by the experimental findings.

Chaudhary Himanshu et al., [٢٠١٢][13]. Were discussed two approaches to generating such trajectories: straight lines in joint space and straight lines in Cartesian space. Among the most prevalent needs in robotics in order to gently transporting the end-effector from the starting point to the destination point. Two user-defined methods for combined trajectory tracking in joint and Cartesian spaces were developed .The algorithm has been put through its paces in a simulation program.

Patel Y. D. et al., [2013][14]. The mathematical description of comprehensive the kinematics and kinetics of a two-degree-of-freedom robotic arm was described. Kinematics and dynamics were simulated using Pro/Engineer software, and the results were compared to analytical

results. The performance measurement results shown using condition number and manipubality index. The condition number value was incorrect both at the moment of actuation and when it reaches the end of the joint limit. The work space limit reflects the limits of actuations as well as the space for work part placement. Friction in a robot joint was known to be related to temperature, force/torque level, position, velocity, acceleration, and lubricant characteristics. When compared to the Lagrangian dynamics technique utilizing MATLAB and Pro/engineer software, the average variance in torques derived by Newton's Euler at both joint axes was roughly 3% for all conceivable positions.

Nabeel, H. et al., [2013][15] Robotic Arm with 3-DOF, Which Can doing standard operations performed in factories, such as spray painting, car assembly, handling of parts, etc. was successfully designed. Because lightening the load on the structure was a priority, aluminum grade 6061-T6 was used to fabricate the mechanical assembly designed with SOLIDWORK. Using inverse kinematics to establish overall angles of joint, the equations were input through ingenious (ATMEGA16) microcontroller, which performs all the calculations required to make an approximation of the joint angles based on the given coordinates so that the joints can be actuated by motorized control. LabVIEW-based graphical user interface was intended to give human-machine interface.

Mohammed, A. et al., [2014][16] Presented a method for creating an optimization module in MATLAB® to reduce the energy consumed by the robot's movement. The technique begins with calculating the reverse kinematics of a robot using Denavit-Hartenberg notation to discover a set of joint configurations necessary to execute the job. Determining the inverse kinematics was typically a difficult stage that involves in-depth analysis of the robot. The module finally solved the robot's inverse

dynamics to examine the torques and forces imparted to each joint and link. In addition, for each arrangement, the energy usage was calculated. The procedure culminates with the optimization of the computed configurations by selecting the one with the minimum power consumption and sending the results to the controller. Three cases were utilized to assess the module's performance. The experimental findings show that the created module may be used to plan energy-efficient robot paths. The results were further analyzed by comparing them to those generated by commercial simulation software. The case studies indicate that optimizing the placement for the goal path may significantly reduce energy consumption.

Hpande, V et al., [2014][17] Modeled the forward and backward kinematics of a 5-degrees-of-freedom (DOF) robotic arm for a pick-and-place task. A generic D-H representation of forward and inverse matrices was obtained. To evaluate the arm's motion through points in space to one another, mathematically-proven steps toward a both kinematics of a five degree of freedom (DOF) robotic arm in both forward and reverse was offered. 5-DOF mechanical arm consisting of five independently revolute joints, was a vertically oriented robot. It's an instructional robotic system that was dependable and secure.

Liu, X., & Xie, Z., [2015][18] Used the D-H matrix approach to determine the final mechanical arm position function and zero-validate its accuracy, therefore establishing the kinematics equations for a plane-armored robot. Planning and optimizing a robot's path requires a thorough kinematic analysis, which can be performed with the use of simulation software like ADAMS. This allows for a better intuitive understanding of the robot's motion, which was essential for real-time control and future optimization.

Mohammed, A. et al. [2015][19] Described the kinematics modeling of the RA- 02 4-degrees-of-freedom (DOF) robotic arm. The straightforward kinematics was solved by combining the Convention of Denavit and Hartenberg and the screw-theory-based product of exponential formula. Comparing the outcomes of both methodologies reveals that they yield equal manipulator kinematics solutions. In addition, an algebraic solution based on trigonometric formulas was offered for the inverse kinematics problem. Lastly, simulation results for kinematics model based on the DH convention using the MATLAB program were provided. Since the two methods were equal, the product of exponential formula should provide identical simulation results for the investigated robotic arm.

Singh, E. et al., [2015][20] The kinematics of motion forward and backward of a six-joint robot manipulator were examined. various techniques were described for locating many solutions for a specific end-effector position (algebraic, geometrical FABRIC, etc., as in the example). All approaches display the position of the end effector, but only for systems with 2 or 3 degrees of freedom. However, forward and backwards kinematics of robot in conjunction with the robotics toolbox in Matlab can determine the end effector positions of a robot system with a large number of degrees of freedom. This method also allow to determine the location of intermediate link position vectors and joint angles at any given robot position. Creation a three-dimensional perspective of the robot and control its progress from its initial location to its final position by using this strategy.

Ramish et al., [2016][21] Presented the design of a serial robotic arm with three degrees of freedom. A computer-aided design program was used to create its mechanical framework. The dynamic characteristics of the robotic arm have been analyzed. The robotic arm's complicated

equation of motion has been derived using the Euler-Lagrange method. RoboAnalyzer software simulations have been compared with the analytical data. The findings were mostly consistent with one another.

Kasera Shery et al., [2017][22] The performance of a sliding mode controller for a Robotic manipulator with three degrees of freedom in terms of trajectory tracking was examined. Industrial robot with three degrees of freedom (DOF) system dynamics were discussed. Sliding mode controllers were used primarily to give the connection some angular motion. MATLAB-Simulink was used to model an industrial manipulator. Sliding mode controller successfully applied to the dynamics of the robotic manipulator, producing excellent trajectory tracking performance.Consideration of the Coriolis, centrifugal, and gravitational matrices was incorporated into the modeling of a 3-degrees-of-freedom robotic manipulator. Due to its durability and stability, the sliding mode controller able to successfully regulate the dynamics of the robot manipulator.

Kütük, M et al., [2017][23] An analytical solution as it relates to the kinematics, both forward and backward, laboratory-accessible Denso 6-axis robot was the primary focus of this research. The resulting equations were presented. The pre-built features on the robotics toolbox have made learning about robot kinematics much easier for researchers. Traditional mathematical calculations, however, were necessary for robot control and for laying the groundwork for future research. Next, the analytical results were checked using a toolbox. The end outcome was the same either way. Interfaces were powerful methods to display numerous works in a small space. For this reason, the operation of Guide User Interface was carried out. Altering the motors' angular displacements one at a time allows for

simultaneous transformation matrix, orientation and position of the robot's hand.

Serrezuela, R et al., [2017][24] Designed and implemented model of a manipulator arm's kinematics types with four-degrees- of-freedom by using the coordinate frames determine the proposed matrices, hence the result using the Method of Denavit and Hartenberg to calculate the robot joints vector of angle both forward and inverse kinematics were covered. The coordinate system defined by the final effector location was used to establish the position and orientation of the robot in the space. The findings were incorporated into a MATLAB application that not only calculates quickly, but also verifies the theory and serves as a tool to streamline analysis and education thanks to its user-friendly interface, which shows off the robotic arm AL5A's virtual motions.

Tarun P et al., [2017][25] Simulated the motion of a robotic manipulator with five and six degrees of freedom in both directions. While forward kinematics can be designed using modern tools, inverse kinematics requires the use of time-tested techniques (iterative, DH notation, transformation). Then they checked the deviation from the analytical answer to make sure it was tolerable. Both the FK and IK solutions for the robotic manipulator's aspect have been modeled satisfactorily.

Xiao, J. et al., [2017][26] Presented a method for determining the manipulator's parameters and establishing its kinematics equation. MATLAB was used to explore how to determine the manipulator's working space, do analysis on that space, and finally obtain a range map of that space. The movement model of the manipulator was then simulated by exploring its positive kinematics, inverse kinematics, and trajectory planning using the Robotics Toolbox in the MATALAB software's working environment. Constructed and obtained time-

dependent angular velocity, angular acceleration, and angular displacement curve graphs for each joint. Because of this study's findings, manipulator motion spaces can be designed with greater ease and precision. In addition, the findings offer a theoretical foundation and point of reference for designing a manipulator's control system, conducting dynamic analysis, and plotting trajectories.

Aktaş, K et al., [2018][27] Were built robotic arm models that can successfully regulate the trajectory tracking for the given point. The MSC ADAMS®-MATLAB® interfaces was used to perform kinematic analysis of a 4- degrees of freedom robotic arm and trajectory tracking. The robotic arms solid model was generated using SOLIDWORKS® software, and motion analysis was performed using MSC ADAMS® & MATLAB® software. To derive a forward kinematics of robotic arm, homogenous transformations matrices were generated using Denavit-Hartenberg parameters. The joint variables were obtained using an inverse kinematic solution, and the kinematic equations were solved using an algebraic technique. As a consequence, the needed joint angles and motors torques for a certain reference trajectory were derived, and it was established that the robotic arm successfully tracked the trajectory.

Souza, L. et al., [2018][28] Presented a method for creating smooth point-to-point joint a trajectory created with radial basis functions for use by robot manipulators (RBFs). Before moving on to further interpolation methods, they first discussed Boundary-constrained interpolation using a gaussian RBF model. The proposed method was then contrasted with conventional planning systems that depend on polynomial & trigonometric formulas. Using the proposed RBF interpolation, the problem of designing trajectories with via points can also be solved. Algebraic and trigonometric splines were used to synthesize trajectories

and compared to the produced trajectories. At last, the six-jointed PUMA 560 robot was used to test the suggested method under two conditions (minimal time and minimum time jerk) and compared the outcomes to those of alternative planning strategies. The numerical results showed that the proposed method was beneficial, providing a practical answer to the problem of how to construct trajectories that were fast to execute and have a smooth profile.

Farman, M., et al., [2018][29] Demonstrated comparisons between three approaches to resolving the inverse kinematics of a three-degrees-of-freedom robotic manipulator. Algebraic methods using the Matlab® solution function, GAs, and ANNs are all part of the toolkit here (ANNs). Moreover, the manipulator's path planning was taken into account, as it allows the operator to direct the commanded motion in the joint space. Participants were compared polynomials of the third- fourth- and fifth orders in solving for the selected coordinates. The results demonstrated that, similar to the GA, the ANN technique achieved the best outcomes by defaulting to displaying only realistic joint values. An evaluation of trajectory planning yielded a fifth-order polynomial, indicating the best possible smoothness of the solution.

Benotsmane, R. et al., [2018][30] Outlined the fundamental steps involved in creating a working version of a 6-axis robotic arm and then using an algorithm, polynomial calculation, with MATLAB as a simulation software to plan and generate trajectories. The authors introduced the subject of trajectory planning and briefly discussed the various approaches that can be taken. Then talked about its basic application, which involves calculating parameters, and showed how it may be used in joint space to plan movements between two places with arbitrary trajectories in between. Finally, an accurate real-world example

of a procedure needed to construct the path of a 6-axis robotic arm using MATLAB simulation, error estimation, and enhancement of the calculating procedure.

Derelİ, S.et al., [2018][31] Were utilized a particle swarm optimization (PSO) in its two different forms to determine Kinematics in reverse refers to the study of robot arm with 7 movable joints. Getting this robot arm from A to B with few mistakes in the working area is a top priority. The results of the standard PSO were compared with those obtained using Random Inertia Weight and Global Local Best Inertia Weight. Furthermore, graphs indicated the course of the robot arm as determined using cubic trajectory planning. Computer simulation results showed that PSO was effective for inverse kinematics problem solving. Although the normal PSO end effector position was used, the PSO variables were far more effective when solving the inverse kinematics.

Consolini, L., et al., [2019][32] Were introduced a novel method for solving a sub problem in motion planning known as the production of velocity profiles. Time-optimal motion planning was a topic of current study in the academic community. The development of a profile can be reduced to a convex optimization problem if the restrictions on doing so were made simpler. But the convexity was destroyed by various real-world limitations (like torque that varies with velocity and viscous friction). The suggested method solves the non -convex optimization problem and finds its global optimum. The experimental outcomes with a three-DOF robot manipulator were also provided .

Ali, F. [2020][33] Described the use of MATLAB and the robotic-tool to perform a kinematic analysis of the medical robotic arm with four degrees of freedom; the arm itself was built with the help of a solid-works simulation program. The paper also provided specifics on the arm's actual

design's implementation of Arduino Mega 2560 for its control. In this investigation, then moved the arm to the specified location without injecting into the muscle, which would necessitate further research, and a medical sensor can be integrated into the interface design to determine the depth to which the needle must penetrate the muscle in accordance with the criteria established by the specialist.

Afrisal, H., et al., [2020][34] Were offered a trajectory planning approach combined with a 3 degree of freedom parallel link robotic arm's ability to avoid obstacles. When used in a practical scenario, like when moving materials throughout a hospital, for example, the robotic arm will inevitably encounter certain snags along the way. Therefore, an approach to trajectory planning that takes obstacles into account was necessary to preserve a smooth actuation and a safer manipulation point-to-point from a starting location to an end position. Using a method based on cubic polynomials, distances and build trajectories that successfully navigate around obstacles was calculated. The experimental data demonstrated the efficacy of proposed strategy in study environment. Experiments with an impediment result in an average position error of 0.43 cm for trajec tory tracking, whereas those without an obstacle result in an error of 0.35 cm.

Benotsmane, R., et al., [2020][35] Used SOLIDWORK software to model two distinct types of industrial robot arms. Models were converted from SOLIDWORK into MATLAB as URDF and XML files for visualization and optimization of industrial robot arm motion. The paper's main contribution was to describe of a case study taken from actual world involving the process of coordinating the actions of two separate software programs. By coordinating the efforts of two programs, manage, optimize, and visualize the movement of the robot arms can be enhanced. As a result, production businesses enjoy more profitability and

competitiveness to improve productivity brought about by the optimal operation of industrial robots.

Kim, J., et al., [2020][36] Described a method for determining starting and final points, as well as intermediate and final velocities and times, so as to satisfy the velocity criteria. A time-scaling method was provided as a means of optimizing the motion of the robot. The goal of this method was to reduce the margin that exists between the actual physical maximum values of velocity and acceleration in real robots and the intended trajectory. After this, a simulation was carried out to establish whether or not the suggested approach is capable of planning the trajectory for continuously moving multiple points, as well as to investigate the effects of varying the control settings. In light of the findings, it has become clear that the proposed approaches can be used for trajectory planning, and in comparison to the cubic spline method, they call for significantly less computation. In addition to this, the robot moves along the path that was designed for it, and the maximum values of velocity and acceleration are not exceeded by its motion at any point. Glue was dispensed onto the sole of the shoe throughout the production process, and an experiment is carried out in order to demonstrate that the proposed method is capable of being used in actual robotic jobs. The findings of this experiment indicated that the robot can use the proposed strategy to follow the curved contour in three dimensions while maintaining a constant speed.

Cesar, J., et al., [2020][37] Were controlled a SCARA manipulator in real time with the virtual interface was possible by creating a program based on the control blocks in LabVIEW, which simplifies programming and does away with hardware interfaces that can be destroyed over time. The adaptability of working with virtual interfaces was demonstrated by

the adaptation of a sub actuated and adaptable final effector. In addition, as shown in the initiation of each SCARA manipulator joint, the above permits simulation of real-time and off-line movement of all articulations and linkages of the manipulator. LabVIEW's user interface is helpful for simulating the operation of a not-yet-fabricated SolidWorks model. Finally, it was learned that such a type of interface reduces manufacturing costs because it does not require any physical controls, instead acting on input from the front panel, and that all required in this case was a simple circuit to read and manipulate the movements, the hardware to run the simulation, and an external power source for the motors and potentiometers. It was helpful for conventionally-programmed handlers looking to save cost while increasing productivity.

Zhang, X., et al., [2020][38] Was controlled and planned of the manipulator's trajectory. The use of the Denavit-Hartenberg (D-H) method for building the robot arm's joint coordinate system; followed by the solution model for forward and reverse kinematics. A fifth-order polynomial interpolation algorithm was used to get the motion trajectory and each rotation of the mechanical claws' ends in the space rectangular coordinate system. MATLAB was used for modeling and simulation to make sure that the robot's structure was well thought out. The joint's motion, velocity, and acceleration curves were all tested experimentally. Finally, theoretical framework parameters , and trajectories were implemented by having a single-chip microcontroller drive the manipulator. Results from real-world applications demonstrate that the overall precision and stability of the manipulator was improved by the control that followed trajectory planning, and that the losses of the manipulator decreased.

Salman, H et al., [2021][39] Demonstrated the kinematic model of an articulated robot arm with three degrees of freedom, built specifically for choosing and placing a task using a hand gripper. The forward kinematics model was introduced to identify the end effector positions according to the Denavit-Hartenberg (DH) standard. For a three degree of freedom (DOF) modular manipulator with a shoulder offset, an algebraic solution based on trigonometric expressions was paired with a geometric approach to determine the inverse kinematics. The work can be used as a resource for learning the actual interface design approach of other robot manipulators, leading to an optimal outcome in terms of the motional characteristics they exhibited.

Ohri, J. et al., [2021][40] Used a sliding mode controller to solve the challenge of tracking trajectories using a three degree of freedom robotic manipulator. The mathematical models of robotic manipulators were presented in this study. Using the mathematical model of the manipulator as input to the MATLAB simulation environment, many variants of a sliding mode controller (SMC) will be implemented in order to reduce tracking error and increase control torque.

Julio Cesar Cordero Muñoz et al., [2021][41] decoupled robot control simplifies the production of position, velocity, and acceleration trajectories, in addition to the combining of series of joint motion. The purpose of this study was to create a motion control application by incorporating a digital model into a 4 degree of freedom SCARA manipulator. First, the workspace for the manipulator was defined. Next, the control diagram was created in LabVIEW. Finally, the Arduino hardware was configured and communicated with. SolidWorks interface configuration; Simultaneous movement interface development; SolidWorks interface communication configuration. In order to develop

such an integration, special hardware features were required, such as those for supporting simulation and communication environments between the various software mentioned, for acquiring and processing analog signals used in the control algorithm applied to the model, and for modeling Lagrange polynomials and direct and inverse kinematics using the Denavit-Hartenberg approach.

Yu, X., et al., [2022][42] Devised a method for constructing trajectories that minimized time spent doing for the manipulator by conducting a simultaneous search for the ideal path. When the exact coordinates of a manipulator joint's motion were unknown, proposed method for approximating the curve's equation is an iterative cubic uniform B-spline interpolation. Next utilized a genetic algorithm to simultaneously optimize length of time it takes for the manipulator to move and its path point in order to determine motion trajectory of the manipulator that minimizes total time spent. Furthermore, provided experimental proof that the proposed algorithm was equivalent to other algorithms with known paths.

2.2 closing remark

A summary of previous research on the kinematics of multi-arm robotic arm with varying degrees of freedom is provided.

1- The kinematic problem can be divided into two sub problems: forward kinematics equations, which can be generated using the standard DH convention, and inverse kinematics equations, which can be solved using a close form approach, a numerical approach, or both. Dynamically, Euler Lagrange analysis provides a systematic way to drive the equations of motion; most researchers used a simplified dynamic model and concentrated on the system response under several

controllers. To validate and indicate the modeling results, the MATLAB Simulink software provides a powerful tool for dynamic simulation as well as an assistance robotics toolbox.

2- Some researchers studied the kinetic performance of the robot by using various computer programs, for example MATLAB, Solid-work, Auto-CAD…etc, in addition to forming the graphical interfaces GUI .
3- In addition to the Euler-Lagrange and the Denavit -Hardenberg techniques, various control algorithms were utilized to compute displacement, velocity, acceleration, torque, and forces.
4- Some of researches perform the motion analysis with help of ADAM program software built in SOLIDWORKS program.

We think the current study considered on earlier research. Two types of paths are tested at four times periods, the kinematics and kinetics parameters of 4-dof robot arm are calculated powered by SolidWorks and LabVIEW software.

Chapter Three

Mathematical Representation

3.1 Introduction

In this chapter, the forward kinematics equations are modeled using the DH convention, an appropriate solution for the arm inverse kinematics is obtained using geometrical and numerical methods, and the wrist inverse kinematics is solved using the analytical Euler angles method. The geometrical representation of velocity is used to create the Jacobian matrix, which is then used to determine the singular configurations of robots. The motion equations are derived with the use of Euler-Lagrange analysis, and motion planning is included the generation of trajectories (via tools like quantic polynomials). Also, we will develop an algorithm for tracking trajectories that uses nonlinear computed torque.

3.2 Robot Kinematics Modelling

When it comes to designing, analyzing, controlling, and simulating robot mechanisms, kinematics is regarded as crucial [43]. The model analysis is split into two main sections: forward/direct and inverse kinematics.

3.2.1. Forward Kinematics of Robots

Kinematics in the forward direction is the ability to figure out the position and orientation of the (i^{th}) link of the robot in the workspace with respect to the (i^{th}) coordinate frame**.** This is accomplished by affixing

the coordinate frame to every joint and employing the rigid motion approach. [45] to figure out its position and orientation in the neighboring frames. **Figure (3.1)** exhibits how the i^{th} link is connected to the $(i-1)^{st}$ link.

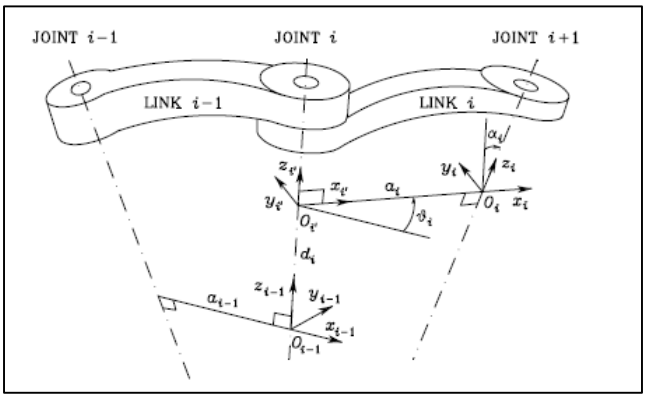


Figure 3.1: Two joined links in a serial chain [43].

Danavit-Hartenberg (DH) notation is used to make homogeneous transformation matrices. Simplifying the procedure by aligning the (z_i) axis with joint axis then establishing (x_i) along the normal between two objects. Since the (z_{i-1}) and (z_i) axes both point in the direction of the z_i axis, we can determine its value using the right hand rule. The four parameters (a_i, α_i, d_i, and θ_i) constitute a D-H coordinate system are illustrates in **table(3.1)** [44].

Table(3.1) Symbolic notation used to describe the DH parameters with its description.

Symbol	Name	Description
a_i	Link length	Distance between z_{i-1} and z_i (along x_i).
α_i	Link twist	Angle between z_{i-1} and z_i (measured around x_i)
d_i	Link offset	Distance between o_{i-1} and intersection of z_{i-1} and x_i (along z_{i-1})
θ_i	Joint angle	Angle between xi−1 and xi (measured around zi−1)

According to D-H convention, transition between two neighboring frames of coordinates is represented by the matrix (T_i^{i-1}) as the product of four basic transformations [45]. In order to derive the position kinematics equations for a given point in an open chain manipulator system, we need to know the transformation of that point from the (n^{th}) coordinate frame into the base coordinate frame.

$$T_i^{i-1} = D_{z_{i-1}}, di * R_{Z^{i-1}}, \theta i * Dxi, ai * Rxi, \alpha i \quad \ldots(1)$$

$$T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \ldots(2)$$

Where:

$$D_{z_{i-1}}, di = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ (Translation along the z_{i-1} axis by d_i)

$$R_{Z^{i-1}}, \theta i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ (Rotation about the z_{i-1} axis by θ_i)

$$Dxi, ai = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ (Translation along the (x_i) axis by (a_i)) .

$$Rxi, \alpha i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ (Rotation about the (x_i) axis by (α_i)).

One can decompose the transformation matrix into its component parts the rotation matrix $R_{(3x3)}$, which determines the orientation of the (i^{th}) frame, and the translation vector $d_{(3x1)}$, which determines the position of the (i^{th}) frame's origin.

$$T_i^{i-1} = \begin{bmatrix} R_i^{i-1} & d_i^{i-1} \\ 0 & 1 \end{bmatrix} \quad \ldots(3)$$

Where:

$$ {}_i^{i-1}R = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i c\alpha_i \\ 0 & s\alpha_i & c\alpha_i \end{bmatrix}, \quad \ldots(4)$$

$$ {}_i^{i-1}d = \begin{bmatrix} a_i c\theta_i \\ a_i s\theta_i \\ d_i \end{bmatrix}$$

The inverse of the transformation $\left({}_i^{i-1}T\right)^{-1}$ equals to $\left({}_{i-1}^{i}T\right)$, which is: [45]

$$\left({}_i^{i-1}T\right)^{-1} = \begin{bmatrix} i - {}_i^iR^T & -{}_i^{i-1}R^{Ti-1} \\ 0 & 1 \end{bmatrix} \quad \ldots(5)$$

$$\left({}_i^{i-1}T\right)^{-1} = \begin{bmatrix} c\theta_i & s\theta_i & 0 & -a_i \\ -s\theta_i c\alpha_i & c\theta_i c\alpha_i & s\alpha_i & -d_i s\alpha_i \\ s\theta_i s\alpha_i & -c\theta_i s\alpha_i & c\alpha_i & -d_i c\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \ldots(6)$$

The following formula is used to figure out how a point in such an open chain robot arm manipulator structure moves from the (n^{th})coordinates system to the reference coordinate frame:[45]

$$P^0 = T_n^0 * P^n \qquad n=1,2,..,4 \quad \ldots(7)$$

Where:

$$P^n = [Px\ Py\ Pz\ 1]_n^T \quad \ldots(8)$$

$${}_n^oT(q_1, \ldots, q_n) = {}_1^0T(q_1)\ {}_2^1T(q_2) \ldots\ldots {}_n^{n-1}T(q_n) \quad \ldots(9)$$

Parameters for the robotic arm manipulator's DH are shown in **table (3.2)** below, according to the assigned frames shown in **figure(3.2)**. Fixed base have axis ($X_0Y_0Z_0$). Link-1 joint represents the waist and its axis of rotation is Z_1. This joint provides a rotational (θ_1) angular motion around Z_1 axis X_1Y_1 plane.Link-2 joint is identified as the shoulder and its axis is perpendicular to Joint 1 axis. It provides a rotational (θ_2) angular motion around Z_2 axis in X_2Y_2 plane. Z_3 axis of Link-3 joint and Z_4 of Link-4 joint are parallel to Z_1 axis of Link-2 joint, they provide θ_3 and θ_4 angular motions in X_3Y_3 and X_4Y_4 planes respectively.

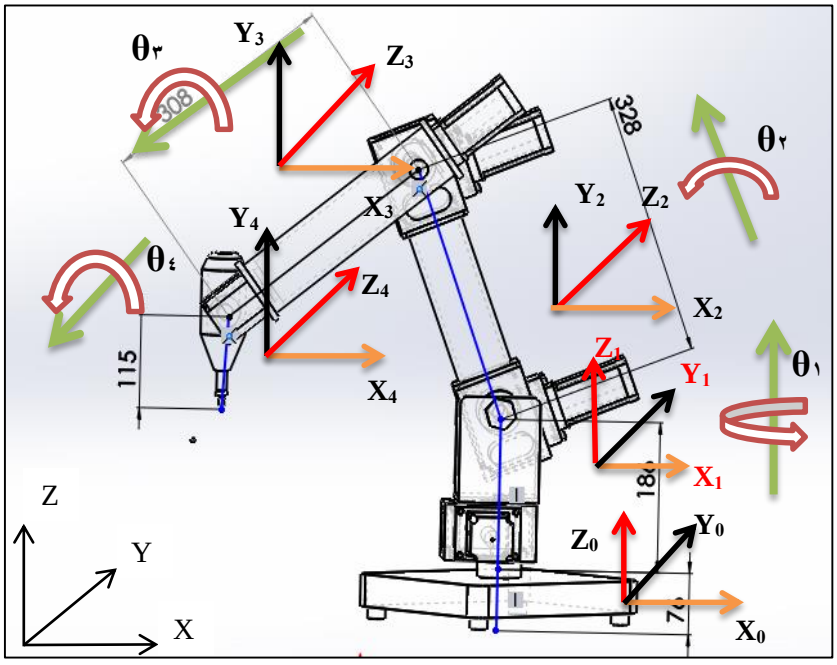


Figure 3.2: The Attached Coordinate Frame Systems.

Table(3.2) DH-Parameters.

Link No.	a_i(mm)	α_i(deg.)	d_i(mm)	θ_i
Link-1	L_1	90	d_1	(θ_1)
Link-2	L_2	0	0	(θ_2)
Link-3	L_3	0	0	(θ_3)
Link-4	L_4	0	0	(θ_4)

The end effector-to-base coordinate frame system transformation can be determined from:

$$ {}^{0}_{4}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T \qquad \ldots(10) $$

$$ {}^{0}_{1}T = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \ldots(11) $$

$$ {}^{1}_{2}T = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \ldots(12) $$

$$ {}^{2}_{3}T = \begin{bmatrix} c_3 & 0 & s_3 & 0 \\ s_3 & 0 & -c_3 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \ldots(13) $$

$$ {}^{3}_{4}T = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \ldots(14) $$

$$ {}^{0}_{4}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \ldots(15) $$

Where:

$$ \left.\begin{aligned} r_{11} &= c_1c_{23}\,c_4 + s_1s_4 \\ r_{12} &= -c_1s_{23} \\ r_{13} &= (c_1c_{23}s_4) + s_1c_4 \\ r_{21} &= s_1c_{23}c_4 - c_1s_4 \\ r_{23} &= s_1c_{23}s_4 - c_1c_4 \\ r_{31} &= s_{23}c_4 \\ r_{32} &= c_{23} \end{aligned}\right\} \qquad \ldots(١٦) $$

.

$$
\left.
\begin{aligned}
r_{33} &= s_{23}s_4 \\
P_x &= c_1(L_4 s_{23} + a_2 c_2) \\
P_y &= s_1(L_4 s_{23} + a_2 c_2) \\
P_z &= L_1 - (L_4 c_{23} + a_2 s_2)
\end{aligned}
\right\} \quad ..(١٧)
$$

.

The forward kinematics of the 3DOF Robot Arm has been determined dependent on **equation(17)** accomplished by SOLIDWORKS programs. Chapter four shows all physical parameters use in simulation based on software tool Simulation in SOLIDWORKS program.

3.2.2 Robot Inverse Kinematic

The computation of joints parameters for a given position of the robot is referred to as inverse kinematics. Different methodologies, such as analytical geometrical and numerical ones, can be used to solve inverse kinematics problems. Inverse kinematics is a subset of inverse dynamics. The main difficulty of inverse problems is the existence of multiple solutions, of which the robot must choose the best based on planning paths, actuator forces, response time, and other constraints. The most popular option is the configuration closest to the origin. When feasible, the geometrical approach is more effective because it is related to structure kinematics, solves equations quickly, and yields closed form solutions [46]. The control should compare and select the nearest one, whereas the numerical approach is preferred for that purpose in decision making because it provides the closest solution directly because it is dependent on an initial guess, which is always the last position [48]. Based on the end effector's Cartesian positions (P_x, P_y, P_z) as well as orientations (roll, pitch, and yaw), the rotation matrix R_4^0 can be made., as depicted in **figure (3.3)** [45]:

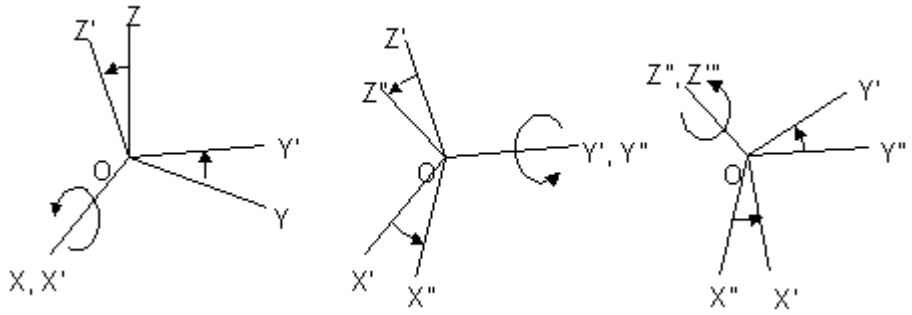


Figure 3.3: X-Y-Z Euler Rotation [45]

$$R_{(\alpha,\beta,\gamma)} = \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix}$$

$$R_{(\alpha,\beta,\gamma)} = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \quad \ldots(18)$$

And, the inverse solution for a given rotation matrix for $\beta \in [-\pi/2 \quad \pi/2]$ is [48]:

$$\begin{aligned} \alpha &= \text{Atan}\,2(r_{21}, r_{11}) \\ \beta &= \text{Atan}\,2\left(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}\right) \\ \gamma &= \text{Atan}\,2(r_{32}, r_{33}) \end{aligned} \quad \ldots(19)$$

And, for $\beta \in [\pi/2 \quad 3\pi/2]$:

$$\begin{aligned} \alpha &= \text{Atan}\,2(-r_{21}, -r_{11}) \\ \beta &= \text{Atan}\,2\left(-r_{31}, -\sqrt{r_{32}^2 + r_{33}^2}\right) \\ \gamma &= \text{Atan}\,2(-r_{32}, -r_{33}) \end{aligned} \quad \ldots(20)$$

The overall transformation matrix ${}^0_4\text{T}$ can be written as:

$${}^0_4T = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\cos\beta c\gamma + s\alpha s\gamma & x \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma & y \\ -s\beta & c\beta s\gamma & c\beta c\gamma & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \ldots(21)$$

3.3 Robot Manipulator Jacobian

Differential of kinematics connects the joint velocities to the proper end effector, exactly like the forward as well as inverse kinematics accomplish for the work area and joint space, correspondingly. Jacobian matrix represents the relationship between the linear and angular velocities.. Jacobian is one of the most effective tools for characterizing robot manipulators [46]. It may be utilized to detect singularities, discover inverse algorithms, determine the relationship between joint torques with applied forces, construct the equations of motion, and develop operational control systems. The 4-dof arm will have the following Jacobian matrix:

$$\begin{bmatrix} v_e \\ \omega_e \end{bmatrix}_{(4\times1)} = \begin{bmatrix} J_p \\ J_o \end{bmatrix}_{(4\times4)} \{\dot{q}\}_{(4\times1)} \quad \ldots(22)$$

Where:

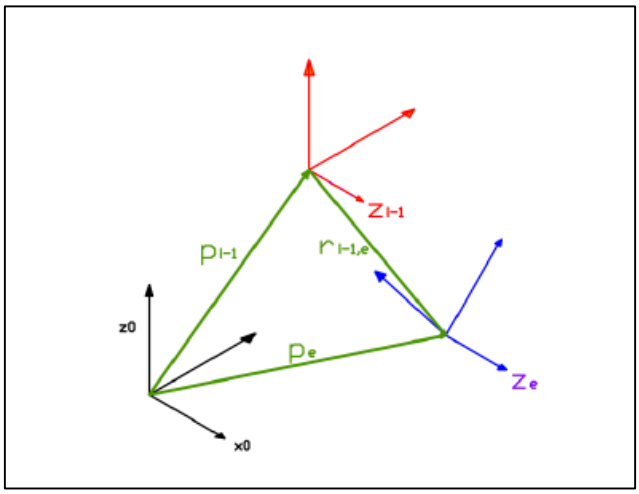
$$J_p = [J_{p1} \quad J_{p2} \quad \cdots \quad J_{p4}]_{3\times4}$$
$$J_o = [J_{o1} \quad J_{o2} \quad \cdots \quad J_{o4}]_{3\times4}$$

Due to rotations of frame i relative to frame i-1, as seen in **figure (3.9)**, the angular and linear velocities of frame i's origin in frame (i-1) may be expressed as vectors:

$$\omega_i^{i-1} = (\dot{\theta}_i . z_{i-1}) \quad \ldots(23)$$

$$v_i^{i-1} = {}^{i-1}\omega_i \times {}^{i-1}r_i \quad \ldots(24)$$

And the expressions for the angular and linear velocities of the origin of (i^{th}) frame as expressed in the base frame are:



Figure(3.4) Vectors Corresponding to the Revolute Joint, Base, and End Effector.

$$^{0}\omega_i = {}^{0}\omega_{i-1} + \dot{q}_i z_{i-1} \quad \ldots(25)$$

$$^{0}v_i = {}^{0}v_{i-1} + \omega_i \times {}^{i-1}r_i \quad \ldots(26)$$

The j[th] column of Jacobian matrix for revolute joint is:[44]

$$\begin{bmatrix} J_{p_i} \\ J_{o_i} \end{bmatrix} = \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} \quad \ldots(27)$$

The rotation matrix's third column ${}^{0}_{i-1}R$ gives z_{i-1}, which can be calculated as follows:

$$z_{i-1} = {}^{0}_{1}R \,..\, {}^{i-2}_{i-1}R \, z_0 \quad \ldots(28)$$

The first three components of the matrix's fourth column of the transformation matrix ${}^{0}_{4}T$ defined the parameter p_e, which can be calculated as:

$$p_e = {}^{0}_{1}T \,... \, {}^{n-1}T \, p_0 \quad \ldots(29)$$

The first three components of the matrix transformation's fourth column ${}^{i-1}_{i}T$ defined the value of p_{i-1}, which may be calculated as:

$$p_{i-1} = {}^{0}_{1}T. \, {}^{n-2}_{n-1}T p_0 \quad \ldots(30)$$

Where,

$$z_0 = [0 \quad 0 \quad 1]^T$$
$$p_0 = [0 \quad 0 \quad 0 \quad 1]^T$$

Then, the Jacobian of the robot end effector is:

$$J_e = \underset{z_0}{z_0 \times (p_e - p_0)} \quad \underset{z_1}{z_1 \times (p_e - p_1)} \quad \underset{z_2}{z_2 \times (p_e - p_2)} \quad \underset{z_3}{z_3 \times (p_e - p_3)} \qquad ...(31)$$

Where:

$$z_0 = \begin{bmatrix}0\\0\\1\end{bmatrix},\ z_1 = \begin{bmatrix}s_1\\-c_1\\0\end{bmatrix},\ z_2 = \begin{bmatrix}s_1\\-c_1\\0\end{bmatrix},\ z_3 = \begin{bmatrix}c_1 s_{23}\\s_1 s_{23}\\-c_{23}\end{bmatrix},$$

$$p_0 = \begin{bmatrix}0\\0\\0\end{bmatrix}, p_1 = \begin{bmatrix}0\\0\\d_1\end{bmatrix},\ p_2 = \begin{bmatrix}a_2 c_1 c_2 + d_2 s_1\\a_2 s_1 c_2 - d_2 c_1\\d_1 + a_2 s_2\end{bmatrix},\ p_3 = \begin{bmatrix}a_2 c_1 c_2\\a_2 s_1 c_2\\d_1 + a_2 s_2\end{bmatrix},$$

And $, p_e = \begin{bmatrix}p_x\\p_y\\p_z\end{bmatrix}$.

3.4 The Derivation of the Robot Manipulator's Dynamics

The link between the controller torques at joints and the structural motion of the manipulator is described by this dynamic system of the manipulator. Using the Euler-Lagrange formula, it is possible to get the equations of motion in a methodical manner, with the Lagrangian being defined as follows:

$$L = K - U \qquad \ldots(32)$$

Where:

L: represents Lagrangian.

K: stands for total kinetic energy.

U: denotes total potential energy.

To begin, an expression for a manipulator's kinetic energy was created. The (i^{th}) link's kinetic energy can be represented as

$$k_i = \frac{1}{2} m_i v_{C_i}^T v_{C_i} + \frac{1}{2}{}^{i}\omega_i^T C_i I_i {}^{i}\omega_i, \qquad ...(33)$$

where the first term is kinetic energy due to the linear velocity of the link's center of mass as well as the second term is kinetic energy due to the link's angular velocity. The overall kinetic energy of the robot is the summation of the kinetic energy in the separate links that is,

$$k = \sum_{i=1}^{n} k_i \quad \ldots(34)$$

Because the linear and angular velocity in **equation(35)** are functions of θ and $\dot{\theta}$, we can conclude that the kinetic energy of a manipulator can be characterized by a scalar expression as a function of joint position and velocity, k. (θ, $\dot{\theta}$). In actuality, a manipulator's kinetic energy is provided by:

$$k(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^{T} M(\theta)\dot{\theta} \quad \ldots(35)$$

M(θ) denotes the n x n manipulator mass matrix. **Equation (35)** is similar to the well-known equation for the kinetic of the point mass.

$$k = \frac{1}{2}mv^2 \quad \ldots(36)$$

Furthermore, the motion equations may be rewritten as follows:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} = \tau_i \qquad i = 1, \ldots, 4 \quad \ldots(37)$$

The kinetic energy of i_{th} link and of i_{th} motor is:

$$K_{i(\ell)} = \frac{1}{2} v_{i(L)}^{T} m_{i(L)} v_{i(L)} + \frac{1}{2} \omega_{i(L)}^{T} I_{i(L)} \omega_{i(L)}$$
$$K_{i(m)} = \frac{1}{2} v_{i(M)}^{T} m_{i(M)} v_{i(M)} + \frac{1}{2} \omega_{i(M)}^{T} I_{i(M)} \omega_{i(M)} \quad \ldots(38)$$

From equation(35):

$$^{0}v_i = J_p \dot{\theta}_i$$
$$^{0}\omega_i = J_o \dot{\theta}_i \quad \ldots(39)$$

Furthermore, the rotation of the principal inertia axes of the i"th " link with regard to the global frames is [49]:

$$^{0}I_i = {}_{i}^{0}R \; {}^{i} I_i \; {}_{i}R^{T} \quad \ldots(40)$$

So then, the supplied total kinetic energy of the robot manipulator is

$$K = \sum_{i=1}^{3} K_{i((L)} + \sum_{i=2}^{4} K_{i(M)} \quad \ldots(41)$$

Substituting equations (38) into (41) gives

$$K = \frac{1}{2}\sum_{i=1}^{3} \left(J_{p(i)}\dot{\theta}_i\right)^T m_{i(L)}\left(J_{p(i)}\dot{\theta}_i\right) + \frac{1}{2}\sum_{i=1}^{3} \left(J_{o(i)}\dot{\theta}_i\right)^T \left({}_i^0R^i I_{i(L)}\, {}_iR^T\right)\left(J_{o(i)}\dot{\theta}_i\right)$$
$$+\frac{1}{2}\sum_{i=2}^{4} \left(J_{p(i)}\dot{\theta}_i\right)^T m_{i(M)}\left(J_{p(i)}\dot{\theta}_i\right) + \frac{1}{2}\sum_{i=2}^{4} \left(J_{o(i)}\dot{\theta}_i\right)^T \left({}_iR^i I_{i(M)}\, {}_iR^T\right)\left(J_{o(i)}\dot{\theta}_i\right)$$
$$K = \frac{1}{2}\dot{\theta}^T B(\theta)\dot{\theta} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}(\theta)\dot{\theta}_i\dot{\theta}_j, \qquad n = 4$$

...(42)

Where:

$$B(\theta) = \sum_{i=1}^{3} \left(J_{p(i)}^T m_i J_{p(i)} + J_{o\iota(i)i}^T\, {}_i^0R^i I_{((i)}\, {}_iR^T J_{o(i)}\right) + \sum_{i=2}^{4} \left(J_{pm(i)}^T m_i J_{pm(i)} + J_{om(i)}^T\, {}_i^0R^i I_{m(i)}\, {}_i^0R^T J_{om(i)}\right) \quad \ldots(43)$$

The gravitational potential energy of the i[th] connection and i[th] actuator is also given by:

$$U_{i(L)} = -m_{i(L)}\, {}^0g^T {}^0r_{ci(L)}$$
$$U_{i(M)} = -m_{i(M)}\, {}^0g^T {}^0r_{ci(M)} \quad \ldots(44)$$

Where:

0_{g^T} : The vector of gravitational acceleration in the reference frame is:

$[0\ 0\ 9.81]^T$

r_{ci}^0 : Location of the mass center of the i[th] link or actuator in the global coordinates frame i.e:

$${}^0r_{ci} = {}_i^0T^t r_{ci}$$

And the whole amount of the robot's potential energy equals

$$U = \sum_1^3 U_{u(j)} + \sum_2^4 U_{i(M)} \quad \ldots \quad (45)$$

According to eq.s (42) and (43),

$$U(q) = -\left(\sum_{i=1}^{3} m_{i(L)}\, {}^0g^T\, {}^0r_{ci(L)} + \sum_{i=2}^{4} m_{i(M)}\, {}^0g^T\, {}^0r_{ci(M)}\right) \quad \ldots(46)$$

By recognizing that U does not rely on $\dot{\theta}$, **then equation (35)** may be represented as [48] :

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}_i}\right) - \frac{\partial K}{\partial \theta_i} + \frac{\partial U}{\partial \theta_i} = \tau_i \quad \ldots(48)$$

By inserting equations (42) and (46) into equation (48), finding that:

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}_i}\right) = \sum_{j=1}^{n} b_{ij}(\theta)\ddot{\theta}_j + \sum_{j=1}^{n} \frac{db_{ij}(\theta)}{dt}\dot{\theta}_j \quad \ldots(49)$$

or

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \theta_i}\right) = \sum_{j=1}^{n} b_{ij}(\theta)\ddot{\theta}_j + \sum_{j=1}^{n} \quad {}_{k=1}\sum^{n} \frac{\partial b_{ij}(\theta)}{\partial \theta_k}\dot{\theta}_k\dot{\theta}_j \quad \ldots(50)$$

Also,

$$\frac{\partial K}{\partial \theta_i} = \frac{1}{2}\sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial b_{jk}(\theta)}{\partial \theta_i}\dot{\theta}_k\dot{\theta}_j \quad \ldots(51)$$

And,

$$\frac{\partial U}{\partial \theta_i} = -\left(\sum_{i=1}^{3} m_{i(i)}\, {}^{0}g^{T} \frac{\partial {}^{0}r_{ci(L)}}{\partial \theta_i} + \sum_{i=2}^{4} m_{i(M)}\, {}^{0}g^{T} \frac{\partial {}^{0}r_{ci(M)}}{\partial \theta_i}\right) \quad \ldots(52)$$

$$\frac{\partial U}{\partial \theta_i} = -\left(\sum_{i=1}^{3} m_{i((L)}\, {}^{0}g^{T} J_{p((i)}(\theta) + \sum_{i=2}^{4} m_{i(M)}\, {}^{0}g^{T} J_{p(mi)}(\theta)\right) = G_i(\theta) \quad \ldots(53)$$

This leads to the following motion equations:

$$\sum_{j=1}^{4} b_{ij}(\theta)\ddot{\theta}_j + \sum_{j=1}^{4} \sum_{k=1}^{4} h_{ijk}(\theta)\dot{\theta}_k\dot{\theta}_j + G_i(\theta) = \tau_i \qquad i = 1,2,\ldots,4 \quad \ldots(54)$$

Where:

$$h_{ijk} = \frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial b_{jk}}{\partial q_i} \quad \ldots(55)$$

Equation (54) represents the Christoffel symbol first type [49].

$$h_{ijk} = \frac{1}{2}\left(\frac{\partial b_{ij}}{\partial \theta_k} + \frac{\partial b_{ik}}{\partial \theta_j} - \frac{\partial b_{jk}}{\partial \theta_i}\right) \quad \ldots(56)$$
$$h_{ijk} = h_{ikj}$$

And finally, the matrix form of the motion equation is [48]:

$$B(\theta)\{\ddot{\theta}\} + C(\theta,\dot{\theta})\{\dot{\theta}\} + G^{T}(\theta) = \tau \quad \ldots(57)$$

Where:

$$c_{ij} = \sum_{i=1}^{k} h_{ijk}\dot{\theta}_k \quad \ldots(58)$$

3.5 Trajectory Generation

In order to guarantee that the manipulator follows the predetermined path, the generated trajectory must be used as a reference by the control system for motion. By specifying the geometric designs of two end effector positions in operational space, a quantic polynomial that can meet the position, velocity, and acceleration constraints at the starting and ending points can be used to figure out how the joints behave over time.

$$q(\mathrm{t}) = a_0 + a_1(t - t_o) + a_2(t - t_o)^2 + a_3(t - t_o)^3 \quad \ldots(59)$$

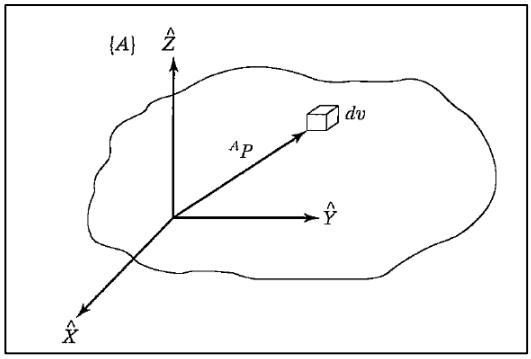
$$\dot{q}(\mathrm{t}) = a_1 + 2a_2(t - t_o) + 3a_3(t - t_o)^2 \quad \ldots(60)$$

$$\ddot{q}(\mathrm{t}) = 2a_2 + 6a_3(t - t_o) \quad \ldots(61)$$

Most of the time, each joint moves and stops moving at the same time. To make the planned path of the manipulator more accurate, it can be broken up into a number of segments. Each of these intermediate points is then transformed into a suitable set of joint angles by using inverse kinematics.

3.6 Inertia Tensor

The inertia tensor generalizes the moment of inertia defined on a flat plane **figure(3.5).** Its measurement system is in terms of mass multiplied by distance square (kg-m2)[44].



Figure(3.5) Inertia tensor describes the object's mass distribution.

The vector P^A here locates the rigid body's differentially volume element, dv. The moment of inertia tensor, expressed at a specific location (A) of the rigid body relative to frame (A) is:

$$I^A = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \quad \ldots(62)$$

Where I_{xx} ,I_{yy}, I_{zz} represent the mass moment of inertia having the value of :

$$I_{xx} = \iiint_V (y^2 + z^2)\rho dv \quad I_{yy} = \iiint_V (x^2 + z^2)\rho dv$$

$$I_{zz} = \iiint_V (x^2 + y^2)\rho dv \quad \ldots(63)$$

And I_{xy}, I_{xz}, I_{yz}, represent the mass products of ineria having the value of:

$$I_{xy} = \iiint_v xy\rho dv\,, I_{xz} = \iiint_v xz\rho dv\,, \; I_{yz} = \iiint_V yz\rho dv \quad \ldots(64)$$

3.7 Instantaneous moment of inertia (I_z)

Before starting with the mathematical formula of the robotic manipulator I_{zz}, it must first be evaluated. As the robotic manipulator spins around the Z_G-axis, its own mass moment of inertia (I_{zz}) changes in value as θ_o changes at each time step.

-The mass inertia moment for link No.1 with regard to the Z_G-axis (I_{1z}) may be calculated as follows:

$dI_{1z} = r^2 dm$; where $dm = \frac{m_1}{L_1} dl$ and $r = y$. This gives $dI_{1z} = y^2 \cdot \frac{m_1}{L_1} dl$.

Since $z = y\tan\theta_1$, then:

$$dl = \sqrt{1 + \left(\frac{dz}{dy}\right)^2}\, dy, \text{ or } dl = \frac{1}{\cos\theta_1} dy.$$
$$\therefore dI_{1z} = y^2 \frac{m_1}{L_1} \frac{1}{\cos\theta_1} dy \quad \ldots(65)$$

And for Link-2:

After simplification

$$I_{2z} = \frac{1}{3} m_2 L_2^2 \cos^2(\theta_1 + \theta_2) + m_2 L_1^2 \cos^2\theta_1 + m_2 L_1 L_2 \cos\theta_1 \cos(\theta_1 + \theta_2) \quad \ldots(66)$$

For Link-3

$$I_{3z} = \frac{1}{12} m_3 L_3^2 \cos^2(\theta_1 + \theta_2 + \theta_3) + m_3 \left[\frac{L_3}{2} \cos(\theta_1 + \theta_2 + \theta_3) + L_2 \cos(\theta_1 + \theta_2) + L_1 \cos\theta_1\right]^2 \quad \ldots(67)$$

For Link-4

$$I_{4z} = \frac{1}{12} m_4 L_4^2 \cos^2(\theta_1 + \theta_2 + \theta_3 + \theta_4) + m_4 \left[\frac{L_4}{2} \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) + L_2 \cos(\theta_1 + \theta_2) + L_1 \cos\theta_1\right]^2 \quad \ldots(68)$$

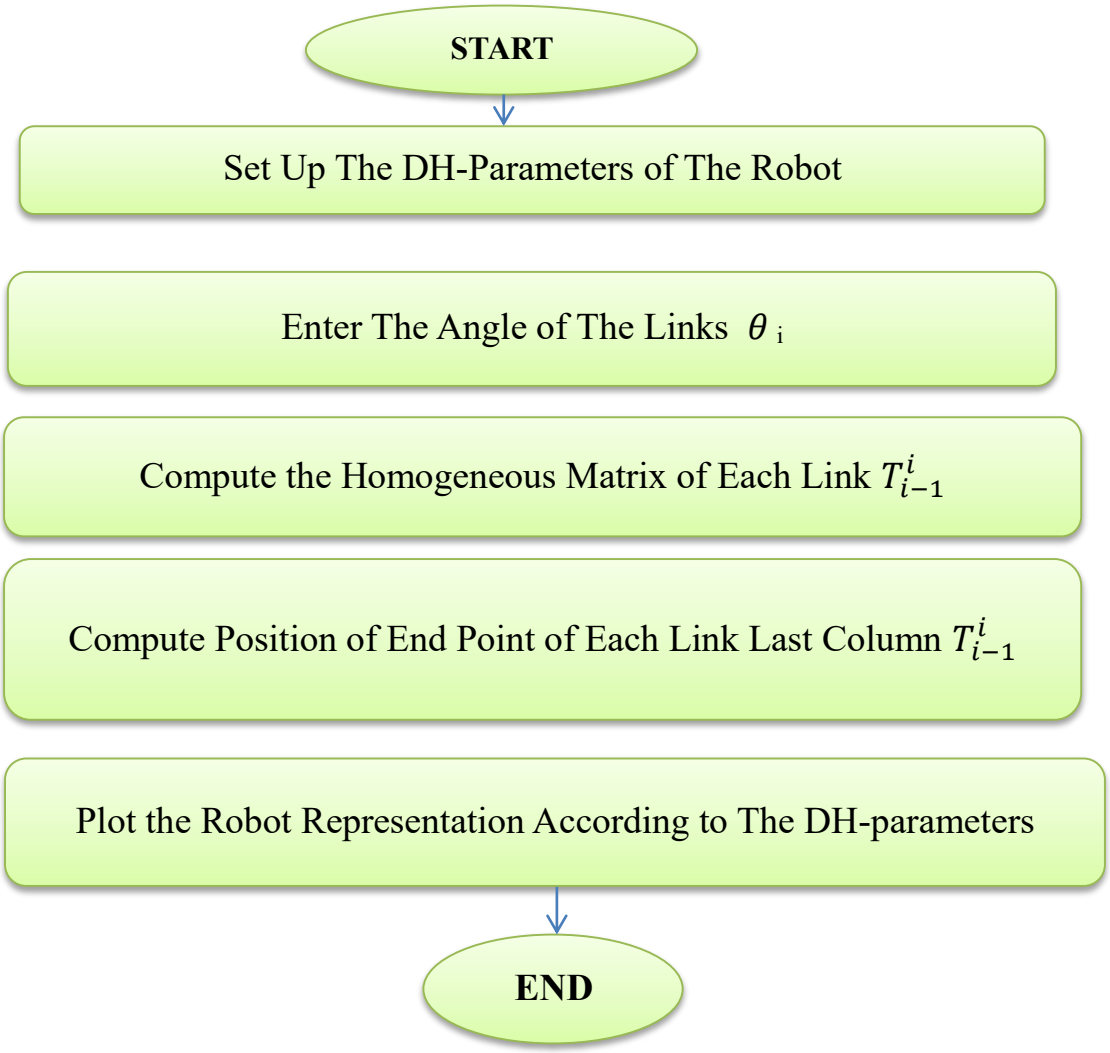
All masses moment of inertia are computed according to equations (62-68) and its value clears in chapter four with help of SOLIDWORKS program.

Paths are Tracking for 4-DOF Robot Arm Manipulator

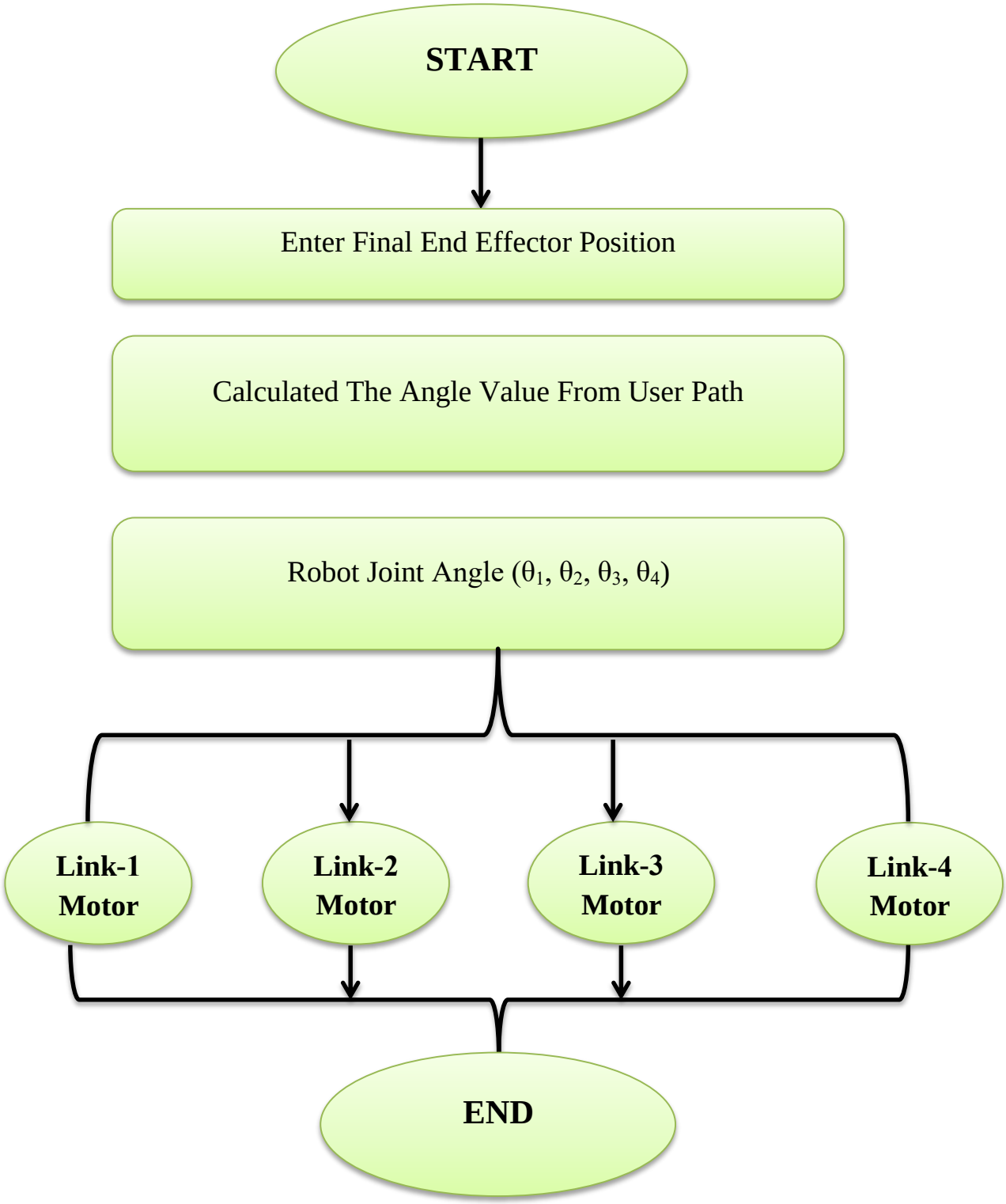
Two paths are tracking in this study as follows:

1- **Vertical Path.**
2- **Horizontal Path.**

Figure(3.6,3.7) represent flowchart of the forward and inverse kinematics of 4-DOF robot arm manipulator, also **figure(3.8)** represents the flowchart system of robot . The mathematical formulation of Forward and Inverse that are illustrated in chapter three have been invested to develop algorithms and then to build computer programs to solve these problems, at the same time the developed program can simulate the instantaneous position of the robot and end-effector while it moves from one point to another. Algorithms will be suggested and developed for each forward analysis inverse analysis and simulation so as solving robot kinematics and evaluation.



Figure(3.6): Flowchart of the Forward Kinematics of 4DOF robot arm.



Figure(3.7): Flowchart of the Inverse Kinematics of 4DOF robot arm.

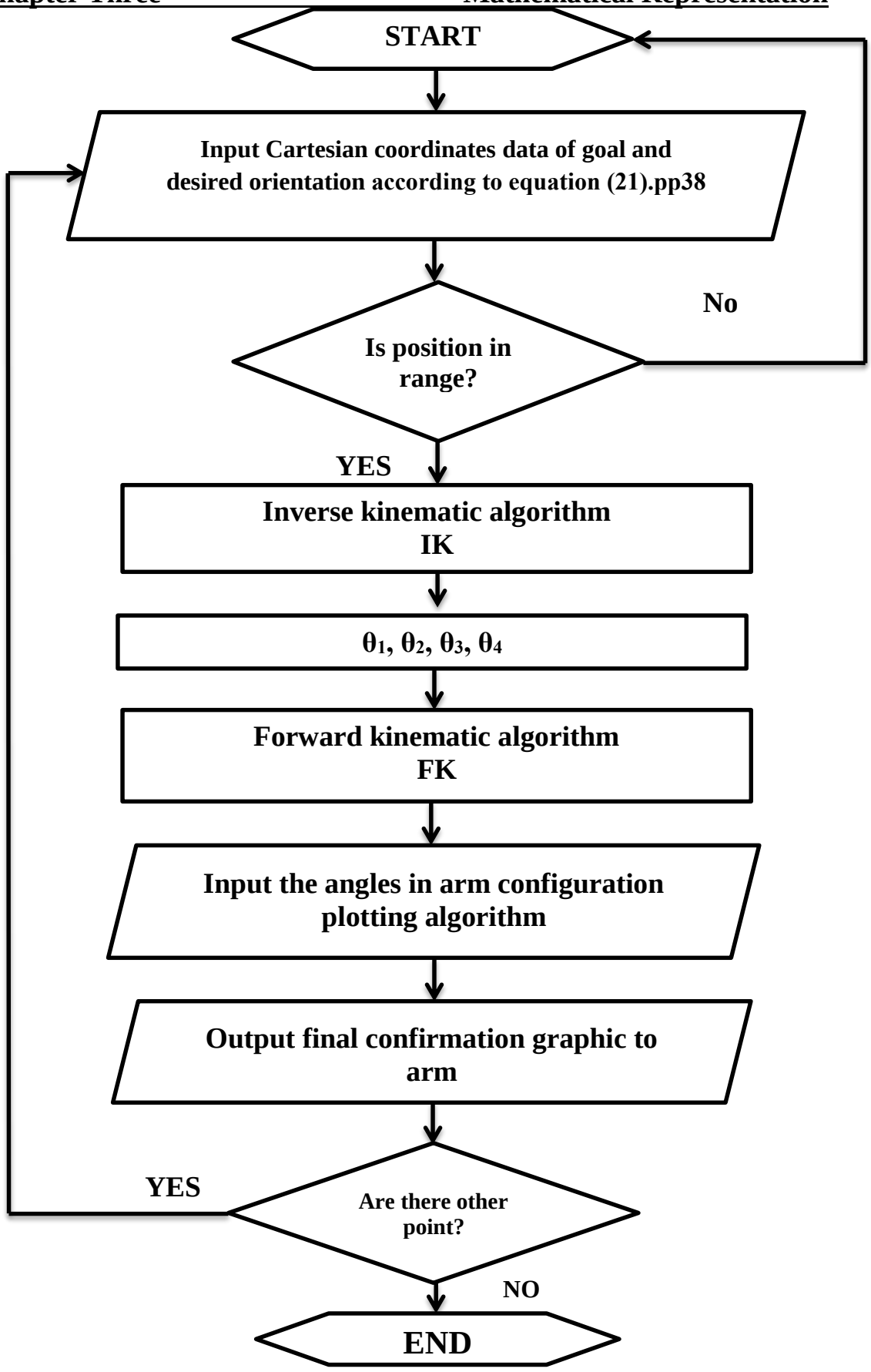


Figure (3.7): Flowchart to Simulate the Robot Motion.

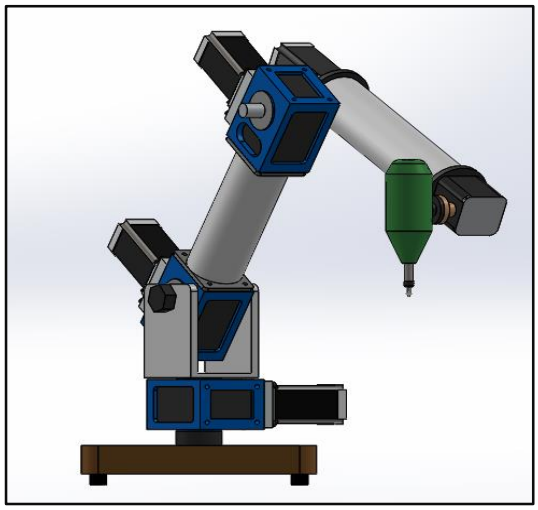
Chapter Four
Simulation and Experimental Work

4.1 Introduction

In the software portion, multiple preparations can be made to achieve a global simulation platform for manipulator motion control; additionally, a fundamental understanding of embedded system planning is necessary to manage the information and data transmission from the computer unit to the robot actuators and vice versa. Four stepper motors were utilized to operate the robot manipulator's hardware, since stepper motors are regarded an acceptable actuator for accurate positioning. Each stepper motor is equipped with a reduction gearbox to boost the precision resolution and maximize the given torques. This chapter explained the programs and subroutines that were used, as well as all the hardware parts, besides all the theoretical and experimental steps to accomplish this work.

4.2 Modelling Manipulators in Computer-Aided Design

To reduce the amount of time needed to develop the robot and to increase its speed and design quality, SOLIDWORKS 3D CAD software was used to create the robot arm manipulator. Assembly of a 4 degree of freedom robot arm manipulator is shown in **figure(4.1).**



Figure(4.1) Assembly of 4DOF robot arm manipulator.

The parts in the assembly are shown in the figures below, **Figure (4.2) (a)** represents the fixed base which attached to Link-1. **Figure (4.2) (b)** illustrates Link-1, While **Figure (4.2) (c)** clears Link-2 , **Figure(4.2) (d)** represents Link-3, **Figure(4.2) (e)** shows stepper motor, finally **Figure (4.2) (f)** shows Link-4 which is the end effector of the robot.

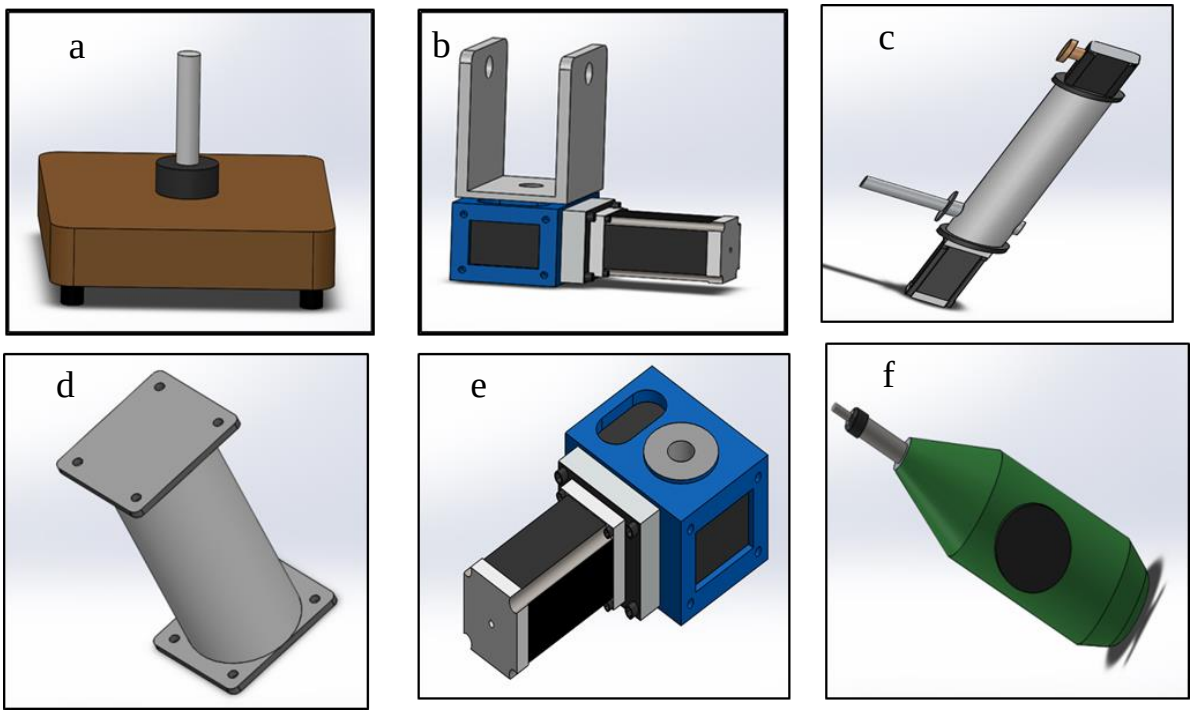


Figure (4.2) (a) (b) (c) (d) (e) (f) Parts of the robot manipulator.

The mechanical characteristics of each component of the robot are illustrated and represented below.

1- Base

Base is a solid box made from (Steel material). **Figure (4.3)** exhibits the design of the 4-DOF Robot Arm's base.

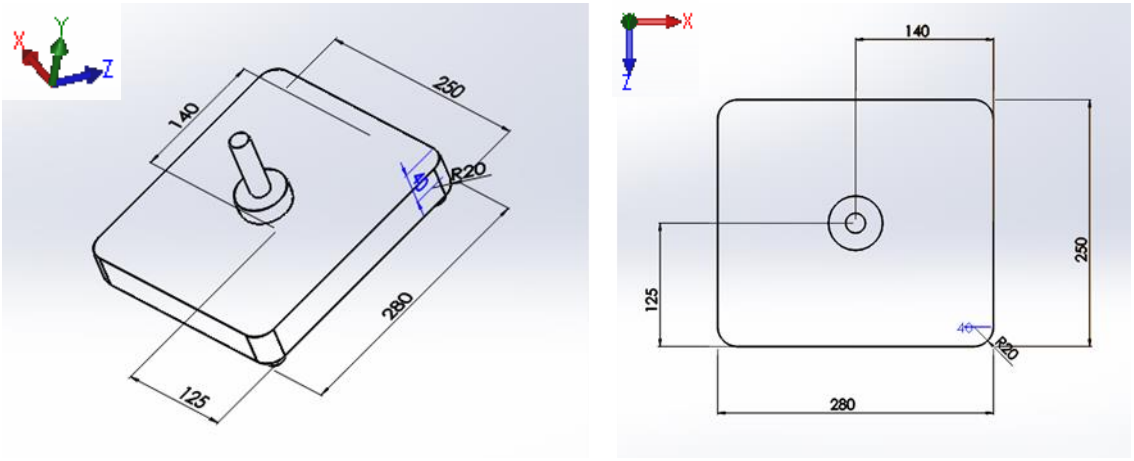


Figure (4.3) Base of the robot arm.

The mass properties of the base clearly illustrated in **table(4.1).**

Table(4.1) Mass properties of Base.

Mass (kg)	density (kg.m3)	Surface area (m^2)	Center of mass (m)
21.57	7500	0.19	X=0 Y=0.02 Z=0

The following table outlines the primary moments of inertia as well as the principal axis of inertia: (in $Kg.m^2$) Measured at the mass's center .

Table(4.2) Moment of Inertia of Base.

Principal axis of inertia ($Kg.m^2$)	Principal moment inertia ($Kg.m^2$)
Ix = (1.00, 0.00, 0.00)	Px = 0.11
Iy = (0.00, 0.00, -1.00)	Py = 0.14
Iz = (0.00, 1.00, 0.00)	Pz = 0.25

Moment of inertia is measured at the center of mass and connected to the output coordinate systems, as shown in **table (4.3).**

Table(4.3) principal axis of inertia

Moment of Inertia($Kg.mm^2$)		
I_{xx} = 0.11	I_{xy} = 0.00	I_{xz} = 0.00
I_{yx} = 0.00	I_{yy} = 0.25	I_{yz} = 0.00
I_{zx} = 0.00	I_{zy} = 0.00	I_{zz} = 0.14

Moments of inertia: (kilograms * square meters) taken at the output coordinate system, is shown in **table(4.4).**

Table(4.4) Moment of inertia at the output coordinate

Moment of Inertia($Kg.m^2$)		
I_{xx} = 1.51	I_{xy} = 0.18	I_{xz} = 0.33
I_{yx} = 0.18	I_{yy} = 1.41	I_{yz} = 0.60
I_{zx} = 0.33	I_{zy} = 0.60	I_{zz} = 0.58

2- Link_1

Link-1 is a hollow shaft made from (Steel material). The design of Link-1 of robot arm manipulator attached to the first stepper motor is shown in **figure(4.4).**

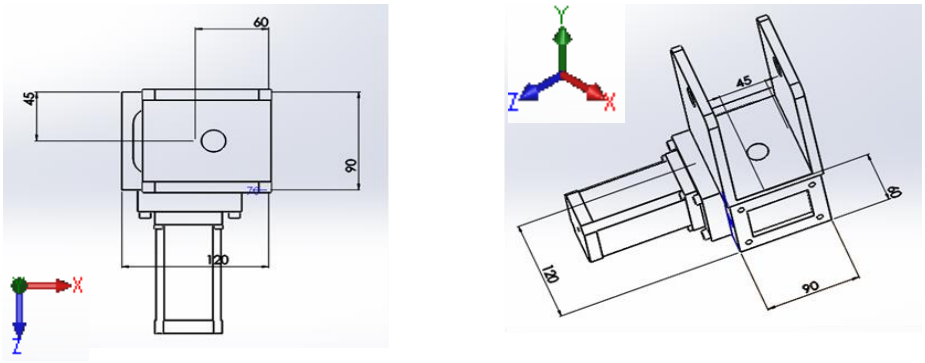


Figure (4.4) Link-1 of the robot arm.

The following **table(4.5)** details the Link-1's mass characteristics.

Table (4.5) Mass properties of Link-1.

Mass (kg)	Density (kg.m3)	Surface area (m^2)	Center of mass (m)
9.80	7500	0.19	X = 0.00 Y = 0.06 Z = 0.03

Table(4.6) provides an illustration of the principal axis as well as the moments of inertia that result from the location of the center of mass.

Table(4.6) Principle axis of Link-1.

Principal axis of inertia ($Kg.m^2$)	Principal moment inertia ($Kg.m^2$)
I_x = (0.72, -0.42, 0.55)	P_x = 0.03
I_y = (0.40, 0.90, 0.16)	P_y = 0.04

I_z = (-0.57, 0.11, 0.82)	P_z = 0.06

Table(4.7)below displays the moment of inertia, measured at the mass's center and transformed into the coordinate system of the output.

Table(4.7) Moment of Inertia of Link-1.

Moment of Inertia(Kg.mm^2)		
I_{xx} = 0.04	I_{xy} = 0.00	I_{xz} = 0.01
I_{yx} = 0.00	I_{yy} = 0.04	I_{yz} = 0.00
I_{zx} = 0.01	I_{zy} = 0.00	I_{zz} = 0.05

The moment of inertia that is gotten from the output coordinate system also is shown in **table(4.8).**

Table(4.8) Moment of Inertia of Link-1.

Moment of Inertia(Kg.mm^2)		
I_{xx} = 1.13	I_{xy} = 0.18	I_{xz} = 0.22
I_{yx} = 0.18	I_{yy} = 0.72	I_{yz} = 0.54
I_{zx} = 0.22	I_{zy} = 0.54	I_{zz} = 0.61

3- Link-2

Link-2 is a hollow shaft made from (Steel material). Design of Link-2 of the robot arm manipulator, that attached to the second stepper motor is shown in **figure(4.5).**

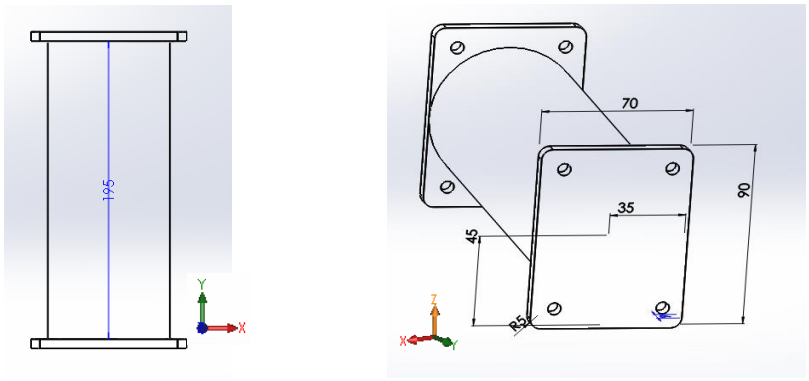


Figure (4.5) Link-2 of the robot arm.

The Link-2 mass characteristics are listed in the **table(4.9)** below:

Table(4.9) Mass properties of Link-2.

Mass (Kg)	**Density ($Kg.m^3$)**	**Surface area (m^2)**	**Center of mass**
15.28	7500.00	0.34	X=0.03 Y = 0.49 Z = 0.28

The center of mass is represented by the principal axis and the moments of inertia is detailed in **table(4.10)** below.

Table(4.10) Principle axis of Link-2.

Principal axis of inertia ($Kg.m^2$)	**Principal moment inertia ($Kg.m^2$)**
I_x = (-0.32, 0.80, 0.51)	P_x = 0.06
I_y = (-0.42, -0.60, 0.68)	P_y = 0.40

I_z = (0.85, 0.00, 0.53)	P_z = 0.44

Moment of inertia measured at the mass center and matched with the output coordinate, is given in **table (4.11).**

Table(4.11) Moment of Inertia of Link-2.

Moment of Inertia(Kg.mm^2)		
I_{xx} = 0.40	I_{xy} = -0.09	I_{xz} = -0.08
I_{yx} = -0.09	I_{yy} = 0.18	I_{yz} = 0.14
I_{zx} = -0.08	I_{zy} = 0.14	I_{zz} = 0.32

The moment of inertia that is got from the output coordinate system is also demonstrated in **table(4.12).**

Table(4.12) Moment of Inertia of Link-2.

Moment of Inertia(Kg.mm^2)		
I_{xx} = 5.23	I_{xy} = 0.17	I_{xz} = 0.07
I_{yx} = 0.17	I_{yy} = 1.38	I_{yz} = 2.21
I_{zx} = 0.07	I_{zy} = 2.21	I_{zz} = 3.99

4- Link_3

Link-3 is a hollow shaft made from (Steel material). **Figure(4.6)** clearly shows the design of Link-3 of the robot arm manipulator, which is coupled to the third stepper motor.

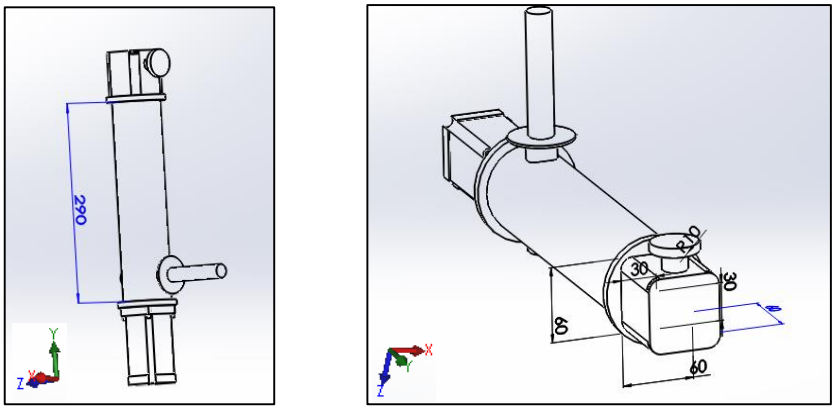


Figure (4.6) Link-3 of the robot arm.

The mass specifications of the Link-3 are shown in **table (4.13)** below:

Table(4.13) Mass properties of Link-3

Mass (Kg)	Density ($Kg.m^3$)	Surface area (m^2)	Center of mass
Mass = 6.04	7500.00	0.27	X = 0.00 Y = 0.58 Z = 0.49

Table(4.14) shows the principle axis and moments of inertia, it is taken form the center of mass.

Table(4.14) Principle axis of Inertia.

Principal axis of inertia (Kg.m^2)	Principal moment inertia (Kg.m^2)
I_x = (-0.49, -0.45, 0.74)	P_x = 0.01
I_y = (0.84, -0.01, 0.55)	P_y = 0.16
I_z = (-0.24, 0.89, 0.39)	P_z = 0.17

Calculating moment of inertia with the mass's center of gravity aligned with the coordinate system of interest is shown in **table(4.15).**

Table(4.15) Moment of Inertia

Moment of Inertia(Kg.mm^2)		
I_{xx} = 0.13	I_{xy} = 0.04	I_{xz} = -0.06
I_{yx} = 0.04	I_{yy} = 0.13	I_{yz} = -0.05
I_{zx} = -0.06	I_{zy} = -0.05	I_{zz} = 0.08

The moment of inertia that is get from the output coordinate system is presented in **Table(4.16)**.

Table(4.16) Moment of Inertia.

Moment of Inertia(Kg.mm^2)		
I_{xx} = 3.60	I_{xy} = 0.03	I_{xz} = -0.06
I_{yx} = 0.03	I_{yy} = 1.58	I_{yz} = 1.66
I_{zx} = -0.06	I_{zy} = 1.66	I_{zz} = 2.11

5- Link-4

Figure (4.7) depicts the design of Link-3, the manipulator attached to the third stepper motor on the robot arm.

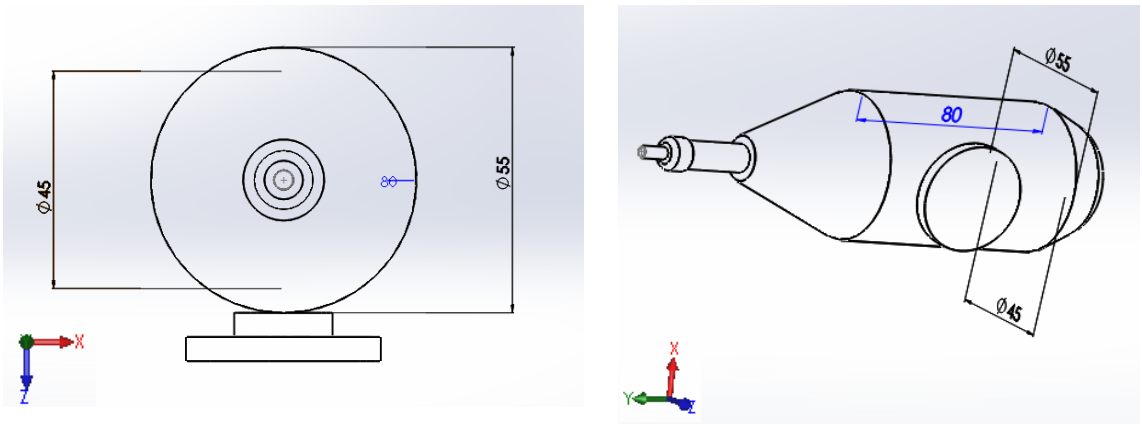


Figure (4.7) Link-4 of the robot arm(end effector).

The mass parameters of the base are shown in **table (4.17).**

Table(4.17) Mass properties.

Mass (Kg)	Density ($Kg.m^3$)	Surface area (m^2)	Center of mass
0.25	1600.00	0.04	X = -0.18 Y = 0.47 Z = 0.63

Table(4.18) containing the moments of inertia and direction of the major axis of the rotation around the mass center.

Table(4.18) Principle moment of Inertia.

Principal axis of inertia (Kg.m^2)	Principal moment inertia (Kg.m^2)
I_x = (0.00, 1.00, 0.00)	P_x = 0.00
I_y = (-0.85, 0.00, -0.52)	P_y = 0.00
I_z = (-0.52, 0.00, 0.85)	P_z = 0.00

The moment of inertia is calculated with the mass's center of gravity aligned with the coordinate component as in **table(4.19).**

Table(4.19) Moment of Inertia.

Moment of Inertia(Kg.mm^2)		
I_{xx} = 0.00	I_{xy} = 0.00	I_{xz} = 0.00
I_{yx} = 0.00	I_{yy} = 0.00	I_{yz} = 0.00
I_{zx} = 0.00	I_{zy} = 0.00	I_{zz} = 0.00

Moment of inertia of coordinate system of the output is cleared in **table(4.20).**

Table(4.20) Moment of Inertia.

Moment of Inertia(Kg.mm^2)		
I_{xx} = 0.15	I_{xy} = -0.02	I_{xz} = -0.03
I_{yx} = -0.02	I_{yy} = 0.11	I_{yz} = 0.07
I_{zx} = -0.03	I_{zy} = 0.07	I_{zz} = 0.06

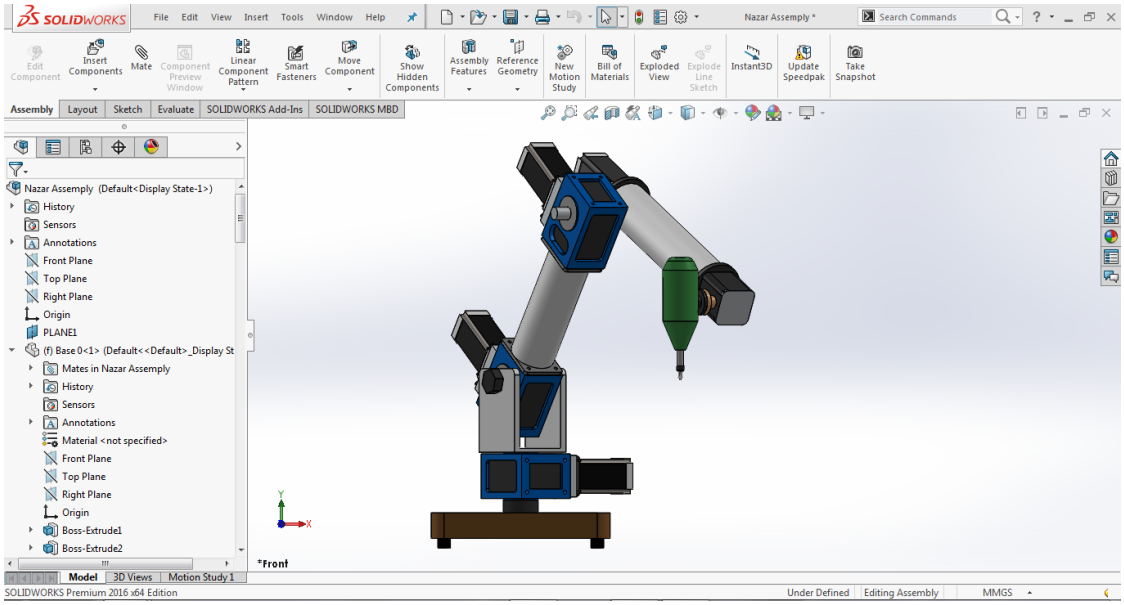
4.3 Simulation Procedures.

1- The lengths of the manipulator's arms and the total number of its movable parts were measured.

2- The program (SOLIDWORKS) drew the components of the manipulator in a manner that was both detailed and accurate, making them similar to the device in its original form.

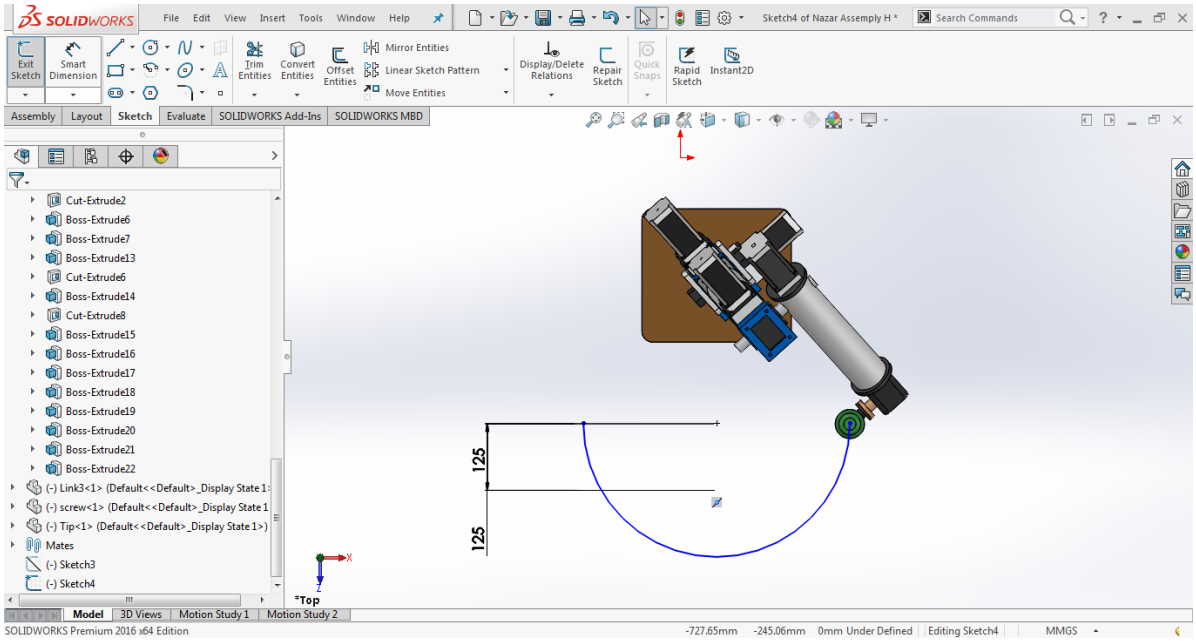
3- After taking into account the tolerable range of movement and the connection of the parts with one another in a manner that is

identical to reality, the components of the device were drawn by (SOLIDWORK 2016) and combined to create an integrated system that was comparable to the device that was first created by following the instructions given by (ASSEMBLY). The assembly clearly shown in **figure(4.8).**



Figure(4.8) Assembly of the robot arm manipulator.

4- It has been proposed to draw the movement path twice (once in the vertical plane and once in the horizontal plane), so that it can be studied and applied in practice; this would make the path a semicircle with a diameter of (250 mm) away from the vertical axis (350 mm), once in each plane. The starting point of the drawing or the point of origin would be the same as the starting point of the movement. The device is located near the top of the vertical axis, creating a stable working area (WORKSPACE). The PATHMATE with model shown in **figure(4.9).**



Figure(4.9) PATHMATE assembly with Model.

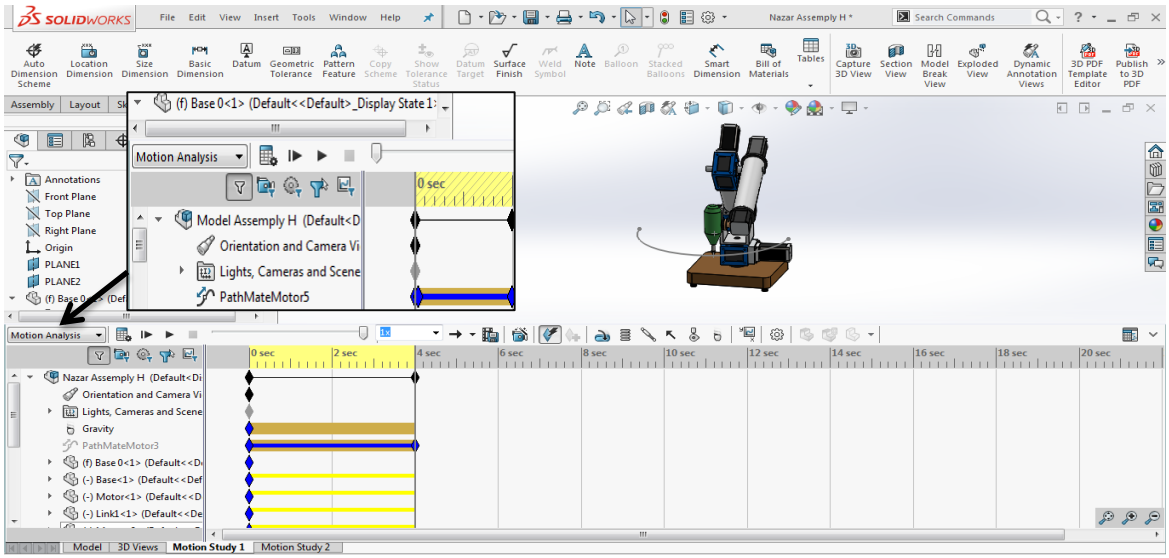
5- The terminal point of the last arm was connected to the path by making use of a directive known as a (pathmate), then moving the system along this path while opting for it to be a logical progression and to be in accordance with reality.

6- The beginning and stopping speeds were calculated, and various times periods were chosen to examine the fourth arm's terminal point (motion reference point).

7- Each component's motion was measured relative to the system's (motion reference point), and the results of angular displacement for each link were exported as (Axel data) in order to study the analysis of motion according to the obtained angular displacement.

8- A total time of 2-4-6-8 seconds was chosen in order to study and analyze the movement, calculate the power required to implement this speed, extract the moments, and compile all of the data for the study and analysis. The analysis was repeated many times with varying starting speeds, and the data was compiled by drawing all

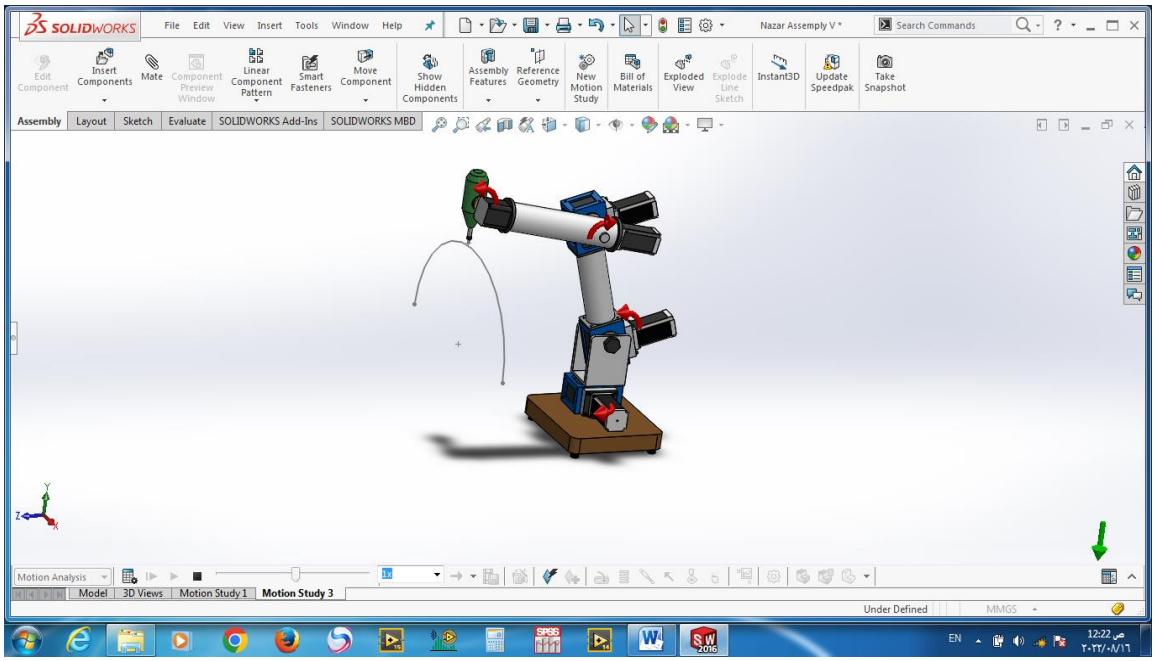
of the moments and drawing all of the data for the study and analysis. The movement analysis process was done by using the (Motion Analysis) button: The most realistic simulation, taking into account all available types of motion objects and providing accurate, numerical results, in the program[50], As shown in **figure(4.10).**



Figure(4.10) Motion study type selection.

9- Next, specify a motion for each of the links. Link-1 must be rotated to 64 degree in the vertical path and 84 degree in the horizontal path. While Link-2 should be rotated 20 degree in the vertical path and 24 degree in horizontal path. Link-3 moved 18 degree in vertical path and 20 degree in horizontal path. Link -4 rotated 40 degree in vertical path while the horizontal path required 16 degree for rotation. To do this, we'll apply a rotational motion to each link at the concentric mate point, selected Rotary Motor under Motor Type. Motion Analysis the most advanced motion analytical tool, incorporating all necessary analysis characteristics such as inertial properties, dynamic circumstances, contacts, mate friction, and so on. The SolidWorks Motion Manager's top time line controls how

long the motion simulation will last. **Figure(4.11)** illustrate the movement of four motors to accomplish the horizontal and vertical path through the proposed time intervals.

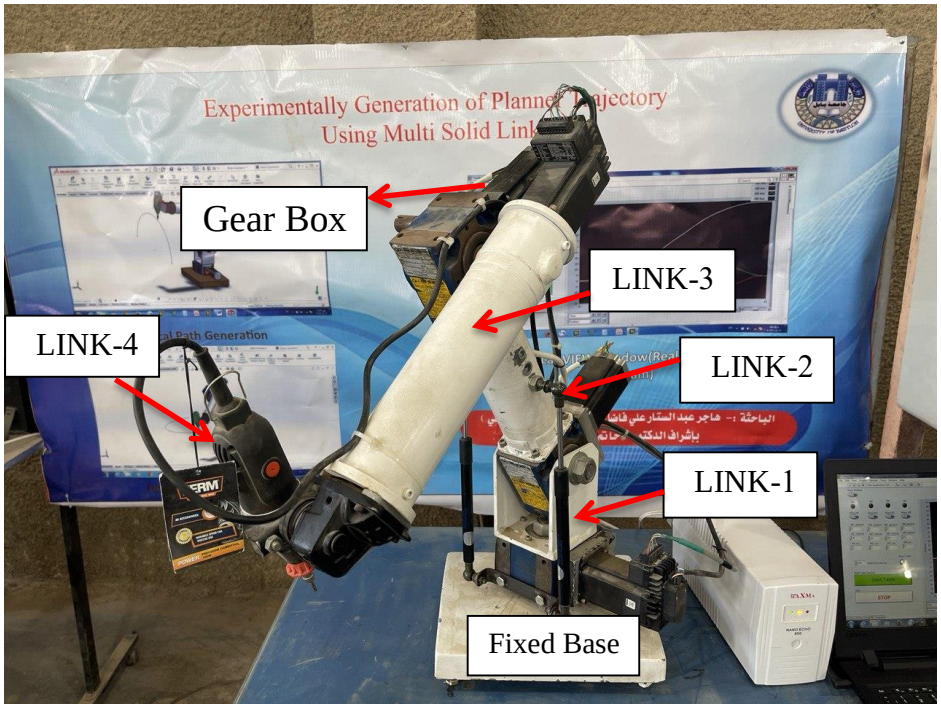


Figure(4.11) Four motors location at the Joints.

4.4 Construction Of Four Degree Of Freedom Real Robot Arm.

In order to carry out the analysis of the research and put into practice a controlled motion, an articulated robot manipulator with four links was constructed , as illustrated in **figure(4.12).** The robot manipulator composed of the base that fixed and cannot be moved. It is always horizontal and is attached to the ground in real life. four links, each of which has four revolute joints R⊥ R∥R (wrist, shoulder, elbow) that rotate in the Cartesian work space , In addition to a wrist that is spherical R⊥R that are accountable for the alignment of the end effector tool. Four stepper motors was used to accomplish the motion of the arm, these motors was connected to gear boxing to transfer the movement from the motors to the arm. Some of Characteristics should be taken into

consideration in modeling like gear ratio, gear inertia, motor efficiency and gear efficiency. The practical side has been accomplished at the University of Babylon, Faculty of Engineering, Manufacturing Process Laboratory of the Department of Mechanical Engineering.



Figure(4.12) 4DOF Robot Arm Manipulator.

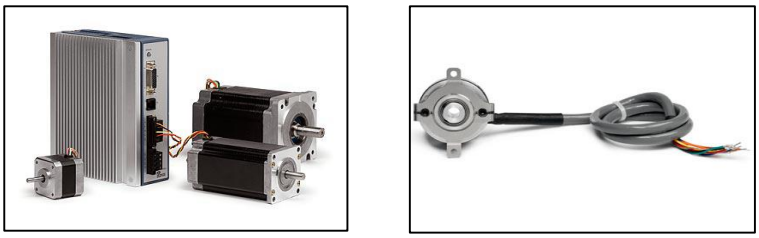
4.5 Stepper Motor Specifications of Robot Manipulator.[51]

The four degree of freedom robotic manipulator's joints are all revolute-type joints, and each is powered by a stepper motor with particular specifications in terms of operating voltage, size, current and, most critically, motor holding torque. National Instruments provides a simple-to-use stepper motion control package, which includes stepper motors, drives, controllers, and software. High torque, accuracy, and simplicity in connecting to stepper motor drivers are all features of NI's selection of stepper motors. Stepper motors are a great option for applications including machine control, manufacturing test, semiconductor placement, and others because of their low cost, simple design, and lack of costly feedback requirements. Stepper motors offer exceptionally accurate and

cost-efficient motion control, small, accurate, and requiring no maintenance, the 2-phase brushless motors rotate by 1.8 degrees at 200 steps each rotation. Regulating stepping motion is straightforward and does not necessitate expensive or complicated feedback systems. For applications that need absolute position accuracy, National Instruments offers encoders that may be connected to motors. Three different frame widths and single or the double shafts are available from the National - Electrical -Manufacturers in Northern- Ireland, Association (NEMA). With NI's P7000 stepper drivers, these motors reach their full potential in both performance and ease of setup. The joints of the four links robot arm are described in the following points:

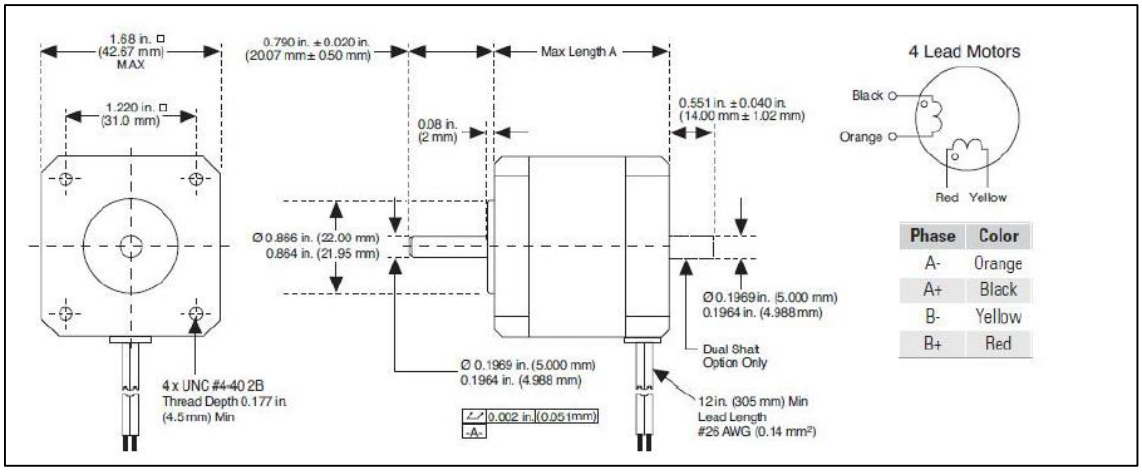
1- Link-1 joint is powered by a NEMA stepper motor 17 via a gear box with a speed ratio of 80:1.
2- Link-2 joint, because it carries all of the succeeding linkages and motors, the shoulder joint is severely loaded as the robotic manipulator moves. A gear box with a 80:1 speed ratio drives the shoulder joint using NEMA 23 stepper motor.
3- Link-3 joint is powered by a NEMA stepper motor 17 via a gear box with a speed ratio of 80:1.
4- Link-4 joint because this joint carries the smallest amount of load, it may be driven by a NEMA 23 stepper motor through a gear box with a 33:1 speed ratio.

Figure(4.13) below illustrate the stepper motors and encoders that were used in this work.



Figure(4.1٣) Stepper motors and Encoders.

Figure(4.14) represents the Dimensions and Wiring of the used stepper motors. While the Software and Electronics specifications and NEMA 23 Motor specification are shown in **table(4.21)** and **(4.22)** respectively.



Figure(4.14) Dimensions and Wiring of Stepper motor.[51]

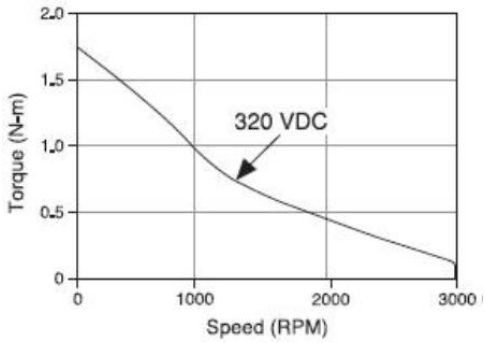
Table(4.21) Software and Electronics of stepper motors and encoders[51]

Stepper-Motors	Encoders
NEMA (17_ 23_and 34) are the frame sizes	1000 counts/revolution resolution
Maximum holding torque of 1710 oz-in (12.1 N m)	NEMA 23 and 34 motor compatibility
3000 rpm max speed	Design with a 1 in (25.4 mm) low profile and simple installation
1.8_ degree step angle	Industrial construction
Designed to work well with P7000 drives	----------

Table(4.22) NEMA 23 Motor specification[51]

Electrical	. Step angle 1.8 deg. . Steps per revolution 200. . Angular accuracy ±3%. . Phases 2
Industry Standards	.Industrial standards CE, cUR, UR. . RoHS Compliance Yes.
Physical	.Operating temperature -20 to 40 °C. .Rated ambient temperature 40 °C. .Shaft load (20,000 hours at 1,500 rpm) .Radial 20 lb (9.1 kg) at shaft center .Axial push 6 lb (2.7 kg) .Axial pull 50 lb (22.7 kg) .Recommended heat sink size 10 x 10 x 1/4 in. .aluminum plate .Recommended encoder 780251-01

The Torque versus speed diagram at 3 A. required to rotate the stepper motor also illustrated in **Figure(4.15).**



Figure(4.15)Torque vs. Speed at 3.0 A. [51]

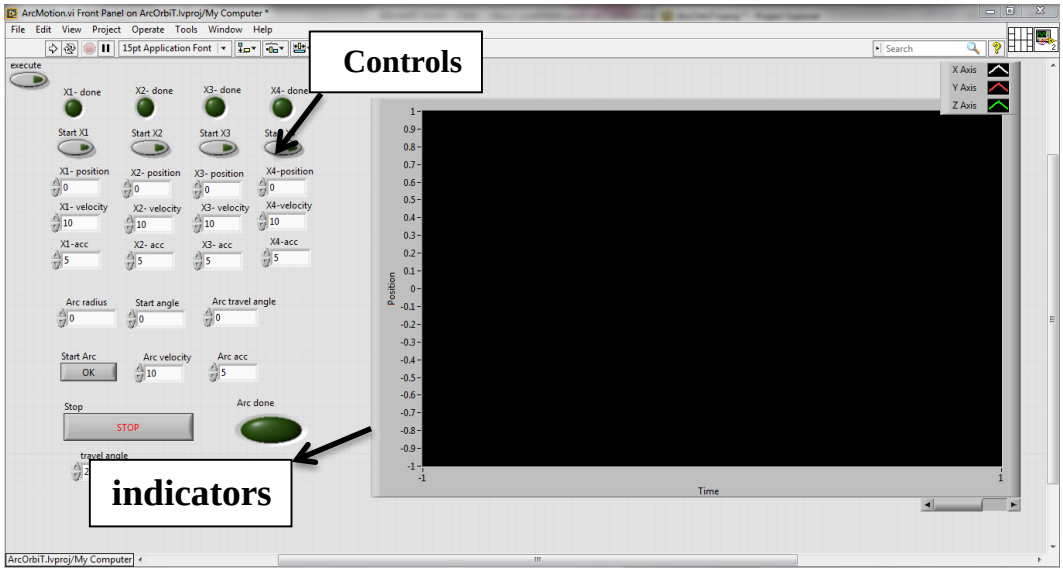
4.6 Experimental steps:

The name of the LabVIEW code developed as (Virtual Tools) VI they are files with extension iv.

VIs are made up of three fundamental parts: The front panel, The Block Diagram, The Icon and Connector Pane.

1- Front Panel: It is the program's interface. The Front Panel was built with Controls and Indicators. Controls are the input units in VI, such as knobs, push buttons, dials, and others. They are analogous

to the input units in real electronic devices, while indicators represents the output or display units in VI, such as graphs, leds, and others. It is similar to the output and display units in real electronic devices [52], as shown in **figure(4.16).**

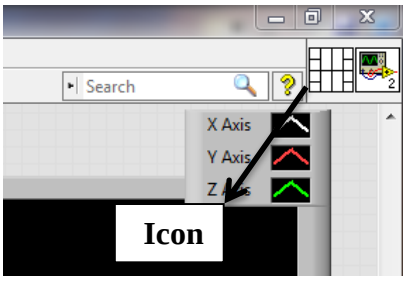


Figure(4.16) LABVIEW Front Panel.

Additionally, the software transfers data from the Controls to the Block Diagram so that it may be finished. The computer code is run on it to see the results, which are shown as a block diagram and indicators.

2- Block Diagram: It is the computer program code that is created by graphics. For each Control or Indicator located in the Front Panel, there is a corresponding terminal in the Block Diagram. This terminal is automatically placed in the Block Diagram as soon as Control is set indicator in Front Panel. When deleting any Control or Indicator from the Front Panel, the private terminal is deleted automatically. That terminal Can only be deleted if the corresponding Control or Indicator is also deleted. In addition to terminals, The Block Diagram contains Sub VIs and functions, constants, structures, and wires that are data path.

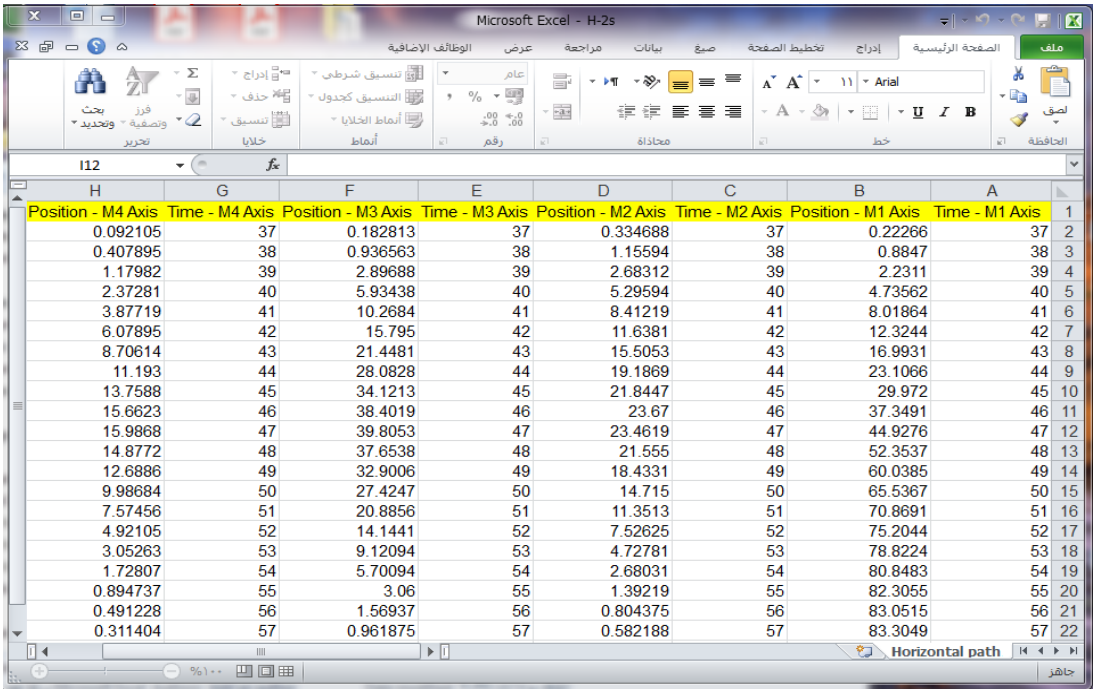
3- Icon and connector pan: For each VI there is an icon that appears at the top right of the window **figure(4.17).** This Icon can contain a drawing, writing, or both



Figure(4.17) Icon in LABVIEW.

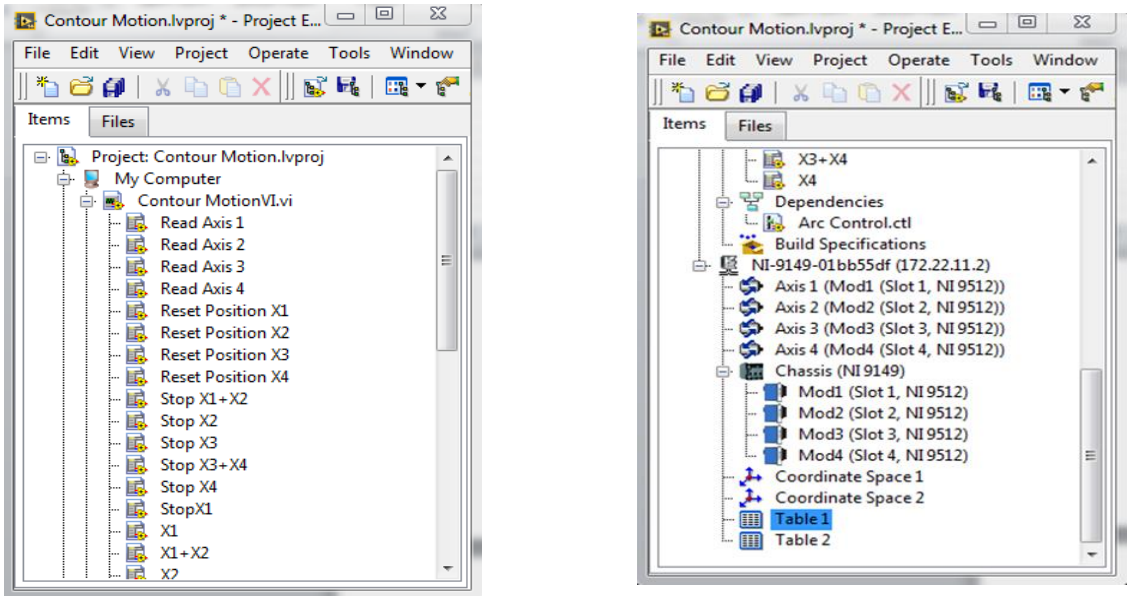
The (LABVIEW) program that is responsible for operating the robotic arm and controlling the movement of the four motors was linked after the final theoretical analysis process finished. This was done in order to implement the movement in real time within the work environment, as this requires feeding the LABVIEW program with the movement and speed of each motor simultaneously in order to produce the movement Basic program (SOLIDWORK motion analysis) within the specified parameters. After redesigning the route and movement to take place in a horizontal direction, the previously outlined stages were carried out once more, and the movement was then carried out on the ground. For each motor, there is a set of readings representing the angular displacement. These readings were obtained using the SolidWorks program. The figure below represents the sample readings that were exported to LABVIEW program in the table field by making copy-paste process for each link in order to Perform the movement for each path. The above paragraph represents the first step in the experimental work . After that by clicking on the project which bears the extension (*contour Motion .Ivproj) then clicking on (NI-9149-01bb55df) that contains the four axis of the four motors, as illustrated in **figure(4.18).** When accomplish all the above

steps, the next step involved moving to the program window and clicking on the button (START TABLE), the robotic arm moved according to the readings fed into the table.



H	G	F	E	D	C	B	A	
Position - M4 Axis	Time - M4 Axis	Position - M3 Axis	Time - M3 Axis	Position - M2 Axis	Time - M2 Axis	Position - M1 Axis	Time - M1 Axis	1
0.092105	37	0.182813	37	0.334688	37	0.22266	37	2
0.407895	38	0.936563	38	1.15594	38	0.8847	38	3
1.17982	39	2.89688	39	2.68312	39	2.2311	39	4
2.37281	40	5.93438	40	5.29594	40	4.73562	40	5
3.87719	41	10.2684	41	8.41219	41	8.01864	41	6
6.07895	42	15.795	42	11.6381	42	12.3244	42	7
8.70614	43	21.4481	43	15.5053	43	16.9931	43	8
11.193	44	28.0828	44	19.1869	44	23.1066	44	9
13.7588	45	34.1213	45	21.8447	45	29.972	45	10
15.6623	46	38.4019	46	23.67	46	37.3491	46	11
15.9868	47	39.8053	47	23.4619	47	44.9276	47	12
14.8772	48	37.6538	48	21.555	48	52.3537	48	13
12.6886	49	32.9006	49	18.4331	49	60.0385	49	14
9.98684	50	27.4247	50	14.715	50	65.5367	50	15
7.57456	51	20.8856	51	11.3513	51	70.8691	51	16
4.92105	52	14.1441	52	7.52625	52	75.2044	52	17
3.05263	53	9.12094	53	4.72781	53	78.8224	53	18
1.72807	54	5.70094	54	2.68031	54	80.8483	54	19
0.894737	55	3.06	55	1.39219	55	82.3055	55	20
0.491228	56	1.56937	56	0.804375	56	83.0515	56	21
0.311404	57	0.961875	57	0.582188	57	83.3049	57	22

Figure(4.18) Sample Data.



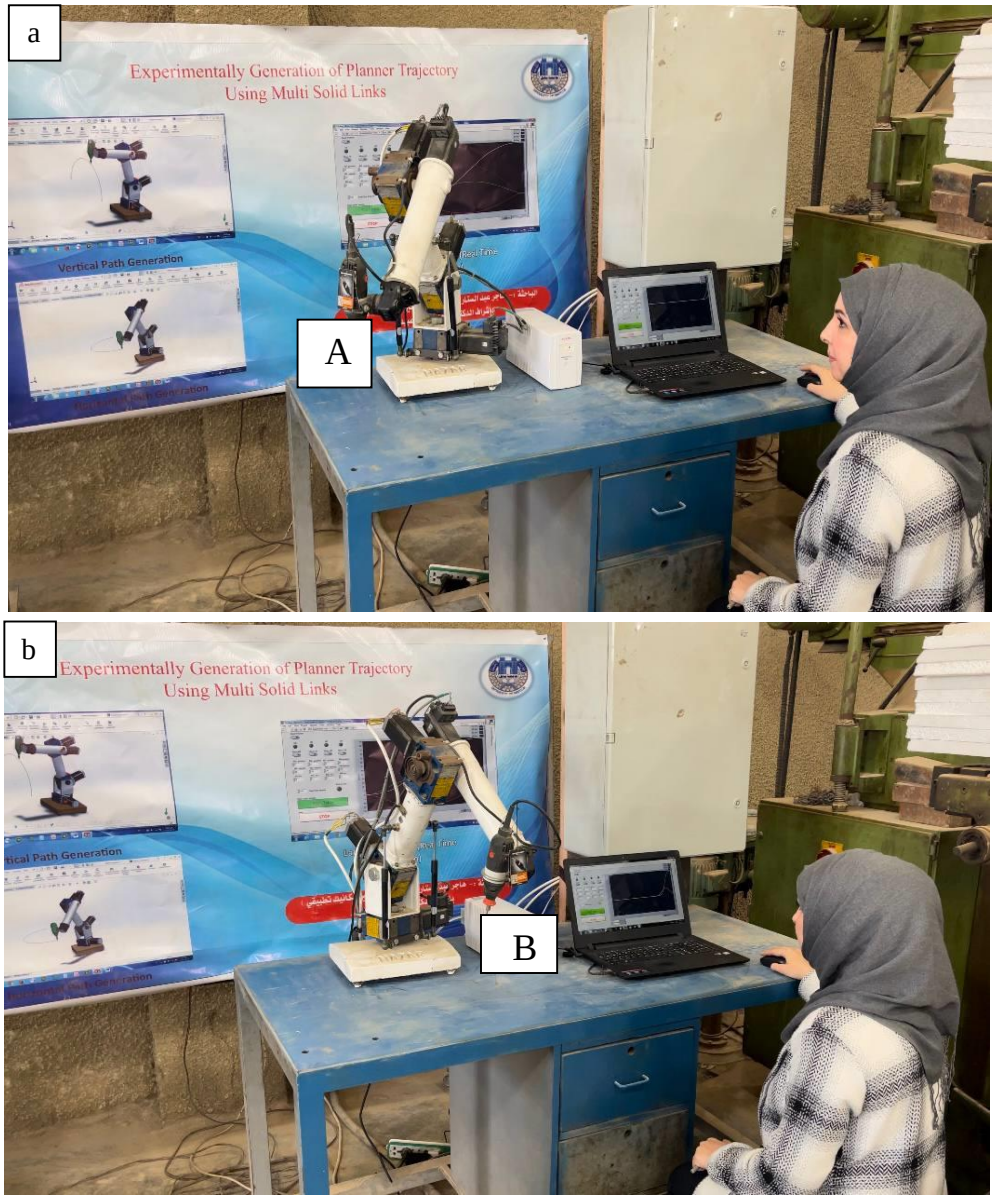
Figure(4.19) Step before operating the robot.

After that the electric power is supplied to the controller board, see **figure(4.20)** which is connected to real robot arm manipulator. The robot was then linked to the laptop so that the LabVIEW software could control it and accomplish the required path **figure(4.21).**

Figure(4.20) Controller Board

Figure(4.21) Robot arm connect with Laptop.

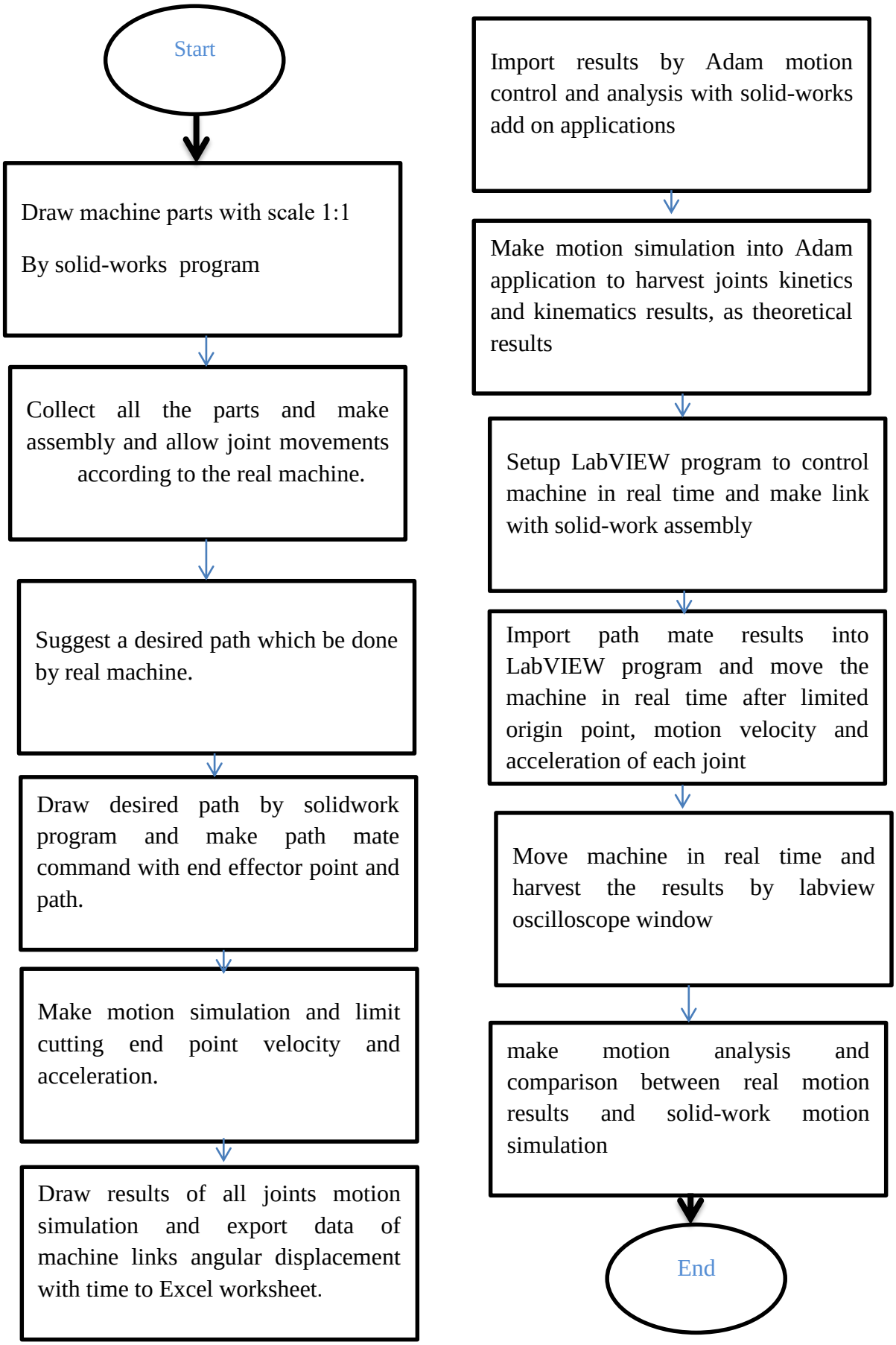
The robot was moved from position (A), which represents the starting point, to point (B), which represents the goal point, after inserting the necessary data for the LabVIEW software and changing the point of origin and the movement of the motors as cleared in **figure(4.22) a, and (4.22) b**



Figure(4.22) a, b Start and Final position of the robot arm.

4.7 Flow Chart of Simulation and Experimental Work.

The Flow chart below represents all the practical and theoretical steps that were followed in order to accomplish this work **figure(4.23)**



Figure(4.23) Flow Chart of Simulation and Experimental Work.

Chapter Five

Results And Discussion

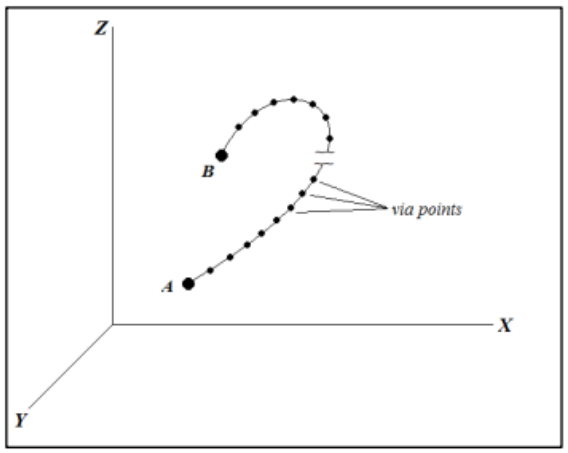
5.1 Introduction

The findings of both theoretical and experimental results are presented in this chapter, including a number of experiments to test the mathematical model and assess the robot's accuracy and repeatability. Mathematical model was created by using (SOLIDWORKS 2016) program, and simulation process was done by (ADAM program) which was built in (SOLIDWORK) .Real model was calibration and tested by using (LABVIEW 2014) program. Four times periods were tested with two types of paths: Vertical curved path and Horizontal curved path to evaluate the robot feasibility in real time trajectory tracking. Firstly the description of the two paths have been shown and conducted, then the trajectory generation geometry for the four times period also shown and discussed for each periods. Finally all results related to the vertical and horizontal path were shown and explained in detail and for each suggested time period. Then, the percentage errors for the angular velocity, acceleration, torques and power consumptions were explained in tables for each the vertical and horizontal path.

5.2 Path Generation

A path is a curve that joins two points in three-dimensional space, defined by a series of intermediate through points that are typically (but not always) spaced equally apart from the two ends. A typical path connecting two endpoints (A) and (B) through a series of points is shown in **figure(5.1)**. The type of task, the robot must complete, greatly influences the path that is selected. For example, a robotic arm designated

to paint must take a completely different path while that same robotic arm is welding.



Figure(5.1). 3D Space Path

The end-effector of the robotic manipulator moves according to the path chosen **figure (5.2)** represents the vertical path trajectory of end effector while **figure (5.3)** illustrates the horizontal path trajectory of end effector which represents the position of the path with respect to (X,Y,Z) coordinates measured by millimeters it was extracted by using MATLAB PROGRAM. Also, the figures below demonstrates the implementation of the vertical and horizontal paths tasks.

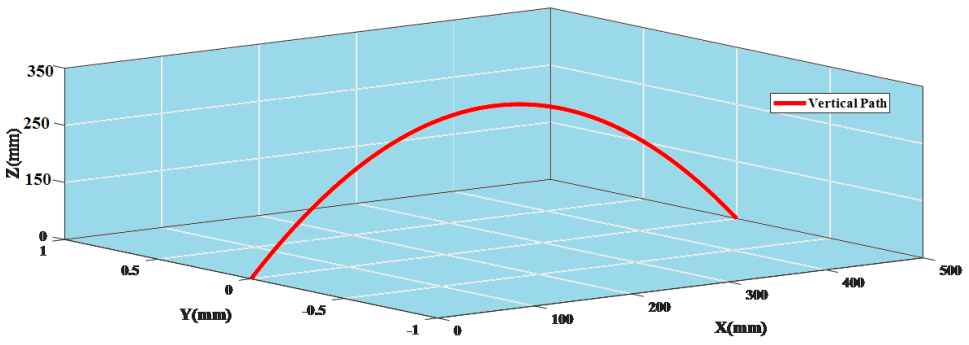


Figure (5.2) vertical path projection on each coordinate plane

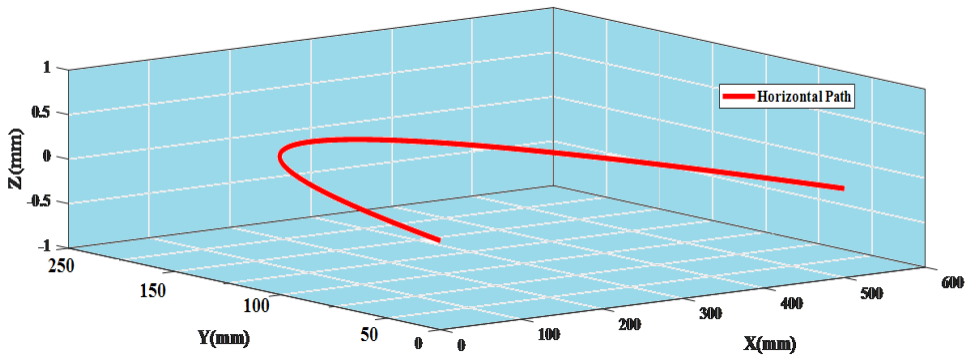


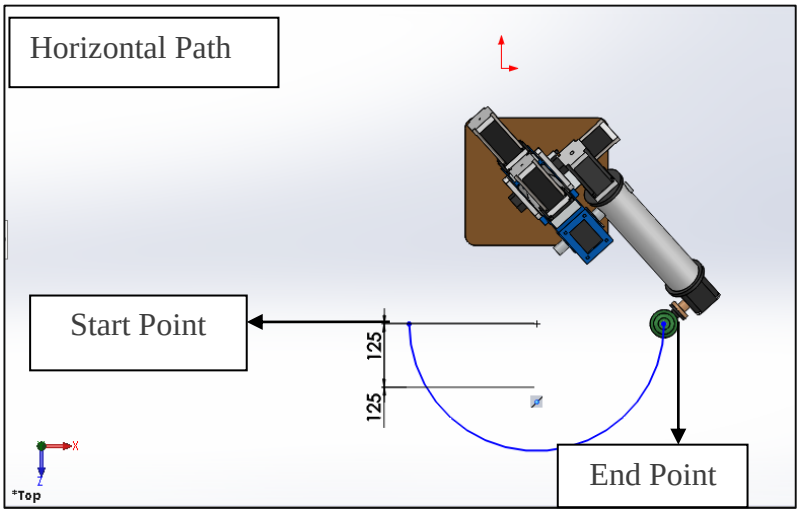
Figure (5.3) Horizontal path projection on each coordinate plane.

Each point on the path (which is represented in Cartesian 3D space) must be represented by the set of joint angles of the robotic arm for the robot to follow the path. Every point on the path must have a solution to the inverse kinematics problem in order to convert the points in the Cartesian space to those in the joint space. In this work, two types of paths are proposed and chosen in the SOLIDWORK program, with the aim of achieving it on the real time using the LABVIEW program. As for velocity and acceleration profile, they were found from the normalization of motion equations and the concept of derivative, where velocity is the derivative of displacement and acceleration is the derivative of velocity. The specifications of the horizontal and vertical path are illustrated in **table (5.1)** below which represents the robot at time zero in initial position for start point while the middle point represents trajectory inflection point to reach the final destination at the end point of the path.

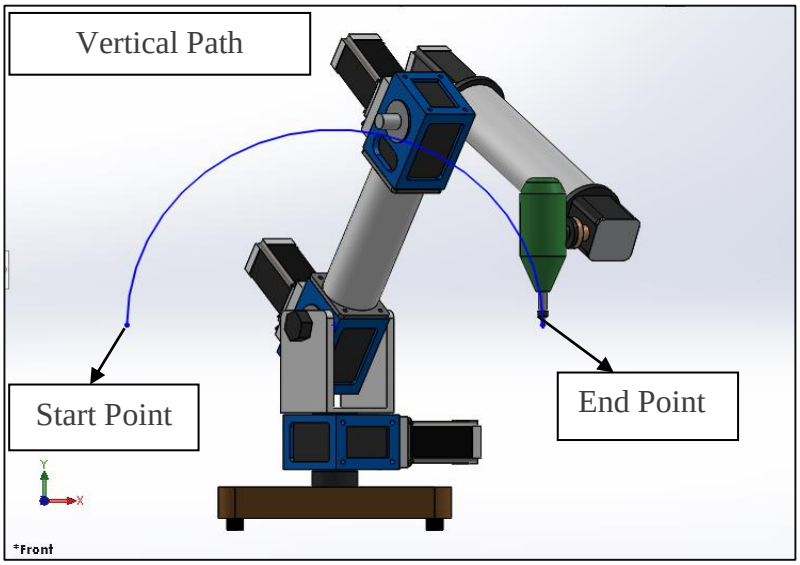
Table(5.1) Specification of Horizontal and Vertical Path.

Path	type	Radius (mm)	Plane	Coordinate axis (mm)
Vertical	Arc	250 mm	XZ plane	Start point- P1=(0,0,0) Middle point- P2=(250,0,250) End point- P3=(500,0,0)
Horizontal	Arc	250 mm	XY plane	Start point- P1=(0,0,0) Middle point- P2=(250,250,0) End point- P3=(500,0,0)

Figures Below show the horizontal and vertical paths from (SOLIDWORK PROGRAM) window.



Figure(5.4) Horizontal Path.

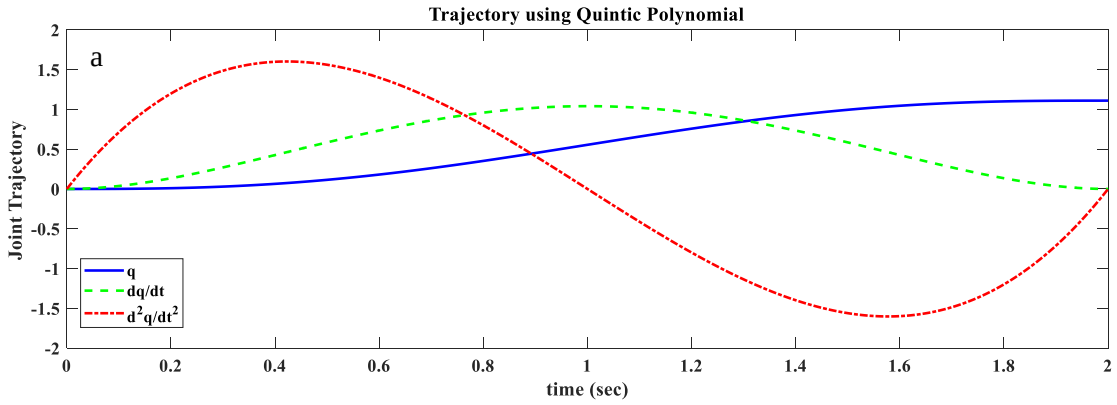


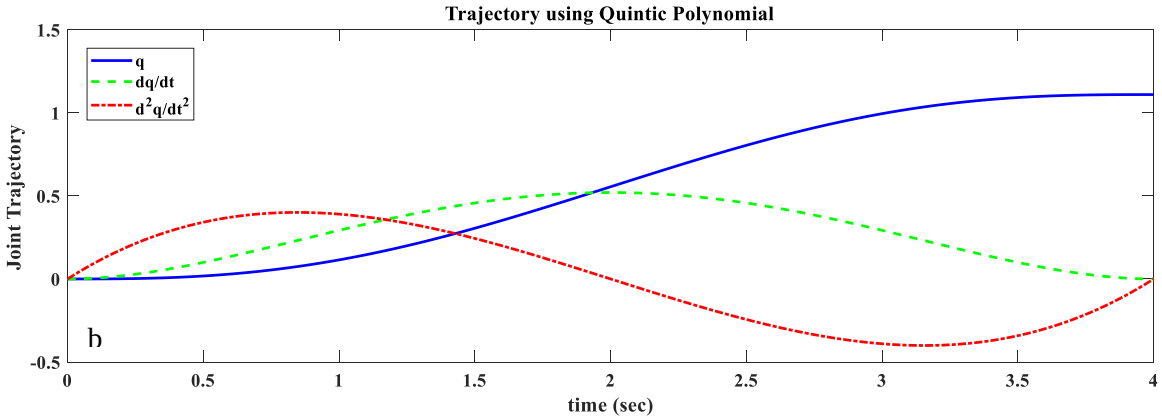
Figure(5.5) Vertical Path.

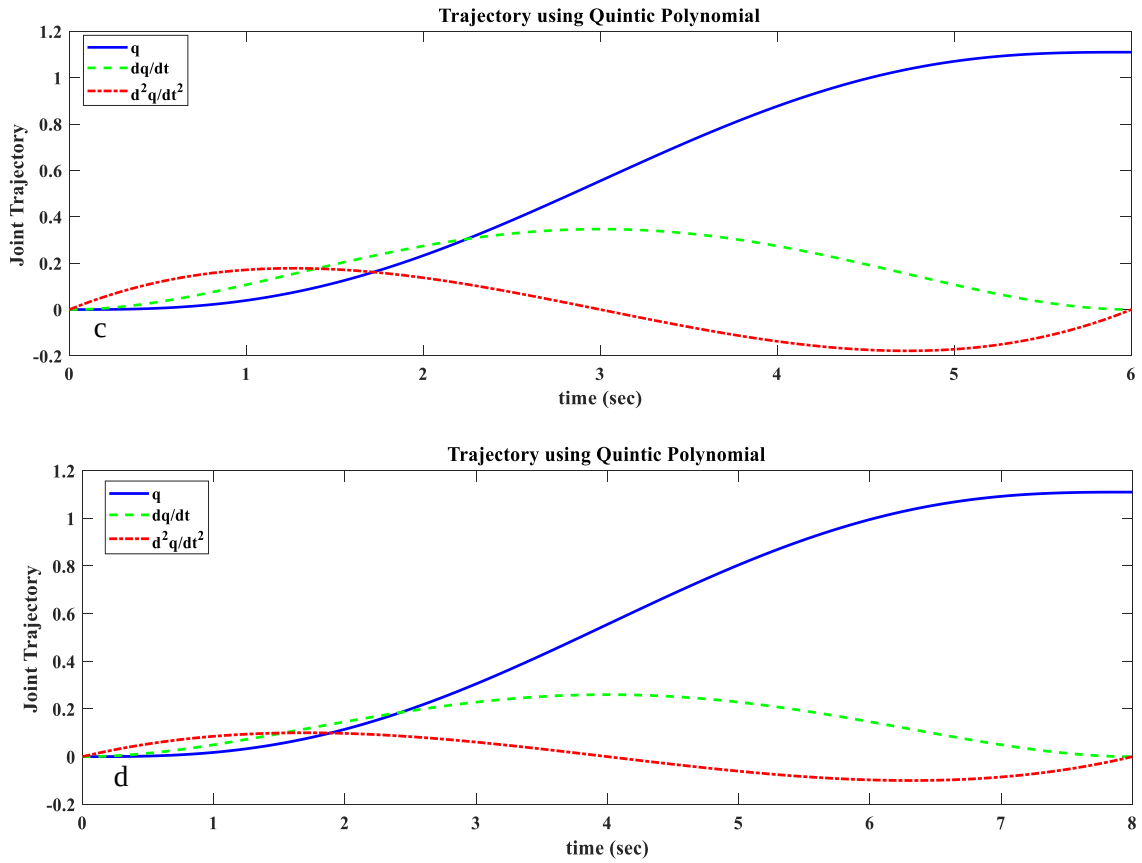
5.3 Trajectory Planning Geometry

When a path is made for a robotic arm, just space-time information is taken into account. The dynamics of the robotic arm are not taken into account. When the dynamics of a robotic arm is considered, we have to take into account the acceleration and speed of each joint. This is called

"trajectory following". When following the path, the user defines how fast the end-effector should move between through points. The end-Cartesian effector's motion was translated into a joint space motion using inverse kinematics, and the resulting mapping is known as the position profile. The position profile shows how each joint in the robotic arm changes over time. By quantitatively differentiating the position profile once and twice, respectively, it is simple to obtain information on the motion's velocity and acceleration. A quantic polynomial that can meet the position, velocity, and acceleration constraints at the initial and final points can be used to define the geometric configurations of two end effector points in operational space and describe the time behavior of joints variables between them. Figure(5.6) (a,b,c,d) represents the displacement, velocity acceleration of the path at (two, four ,six, eight) seconds.







Figure(5.6) a,b,c,d Position Velocity Acceleration profile at four time periods of the path.

5. 4 Theoretical Results

The research project's software program built and modeled a robot arm manipulator with four degrees of freedom, which allowed the robots to follow a variety of different trajectories. This section presents and discusses the kinematic properties, and it also covers the dynamic properties, which include the torques, power consumption, and moment of inertia.

5.4.1 Vertical Path Results

The kinematics and dynamics results of vertical curved path are described and discussed. The curved path Simulation from the start point where the (X,Y,Z) coordinates axes are(0,0,0) middle point (250mm, 0,

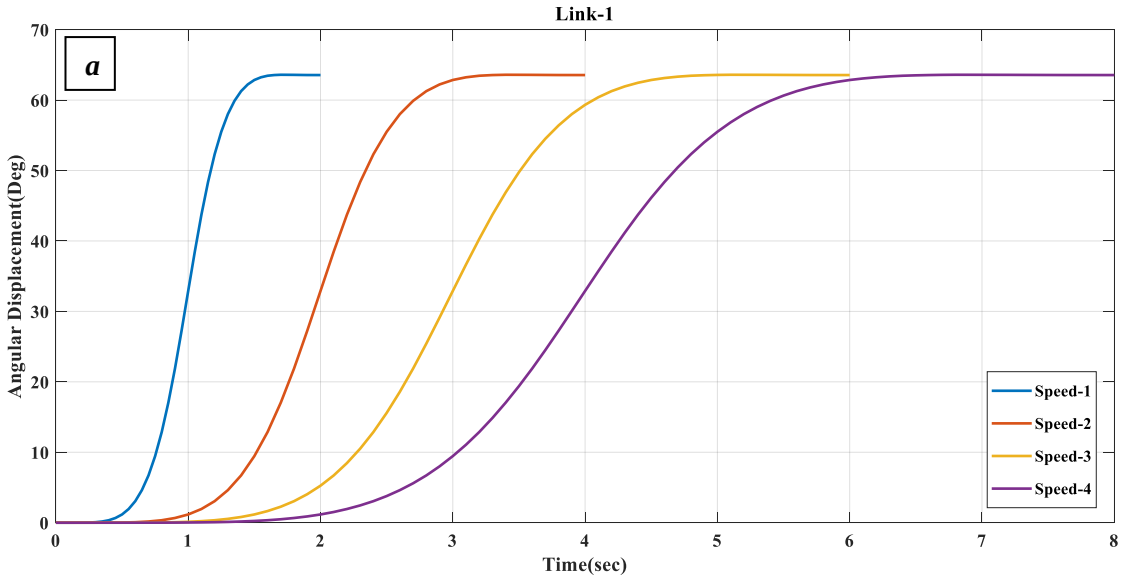
250mm) to the end point which has the coordinate axes(500mm,0 ,0).The 4DOF Robot manipulator result analysis is separated into two sections, as follows:1

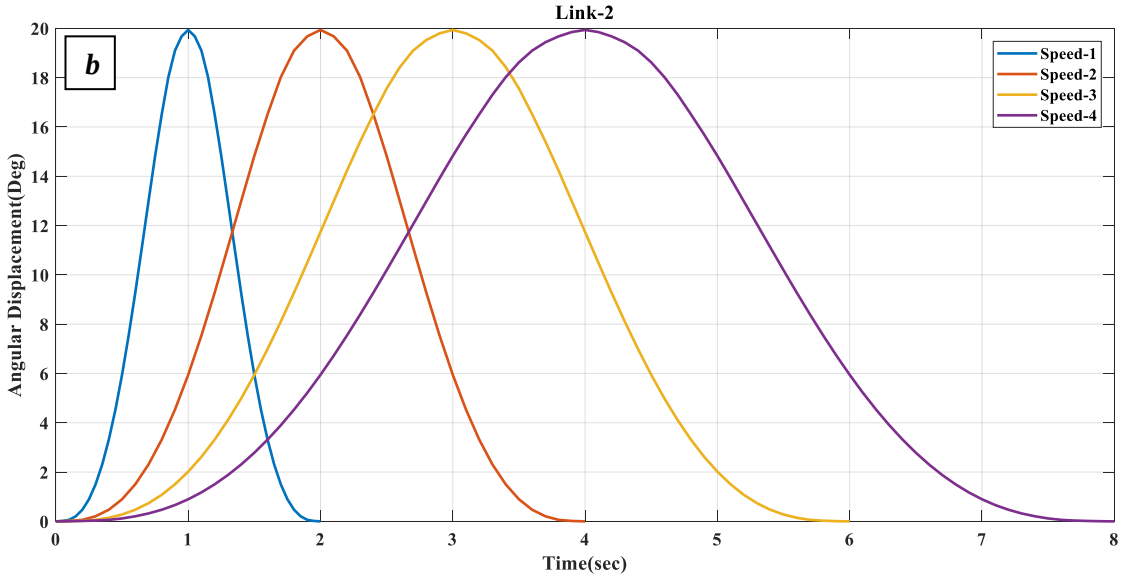
1- Kinematic Analysis of 4-dof robot arm.
2- Dynamic Analysis of 4-dof robot arm.

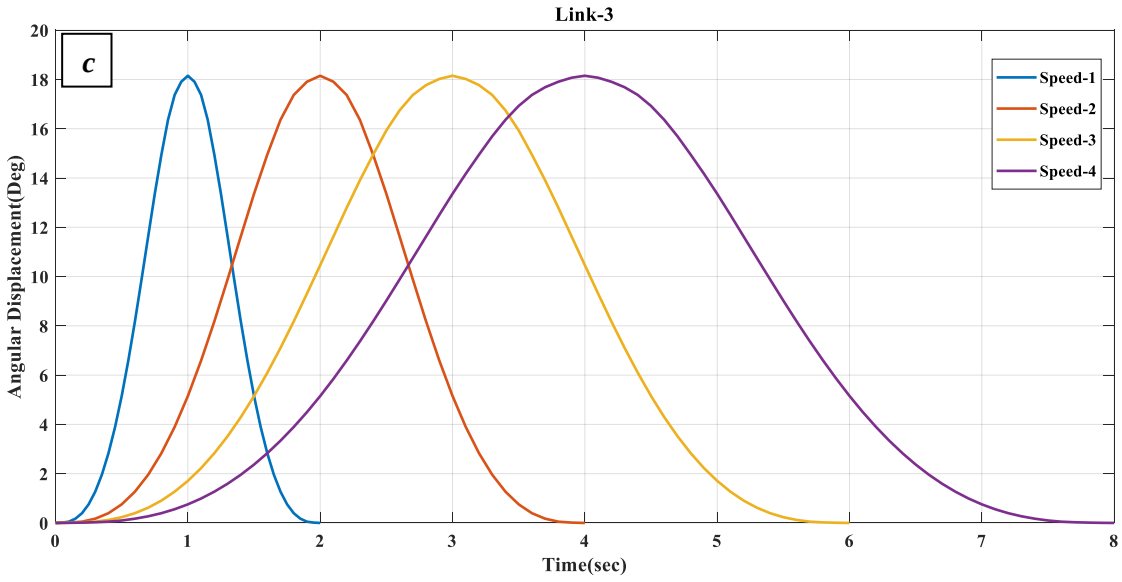
5.4.1.1 Kinematics And Kinetics of Links -1-2-3-4 for Vertical Curved Path.

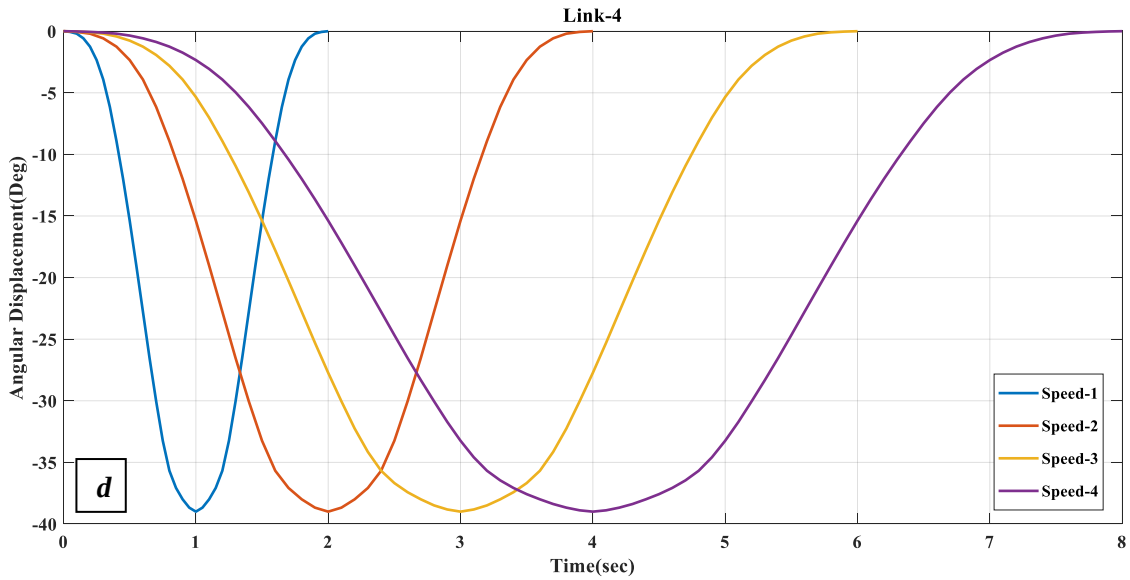
a-Angular Displacement Profile

Figure(5.7) (a) shows that the displacement of the Link -1 differs along the path and the direction in (counter clockwise). Link-2 also track the path anticlockwise direction as illustrated in **figure(5.7) (b)** . The same applies to the third link, as it traversed the path in a counterclockwise direction as illustrated in **figure(5.7) (c).** Link-4 reverse its direction taking clockwise direction the anticlockwise direction along the path for the different time periods as shown in **figure(5.7) (d).** Blue color curve represents the angular displacement at two-second, Red color curve represents the angular displacement at four-second, yellow color curve represents the angular displacement at six-second while the purple color curve represents the angular displacement at eight-seconds.


Link-1
a
Angular Displacement(Deg)
Time(sec)
Speed-1
Speed-2
Speed-3
Speed-4


Link-2
b
Angular Displacement(Deg)
Time(sec)
Speed-1
Speed-2
Speed-3
Speed-4


Link-3
c
Angular Displacement(Deg)
Time(sec)
Speed-1
Speed-2
Speed-3
Speed-4

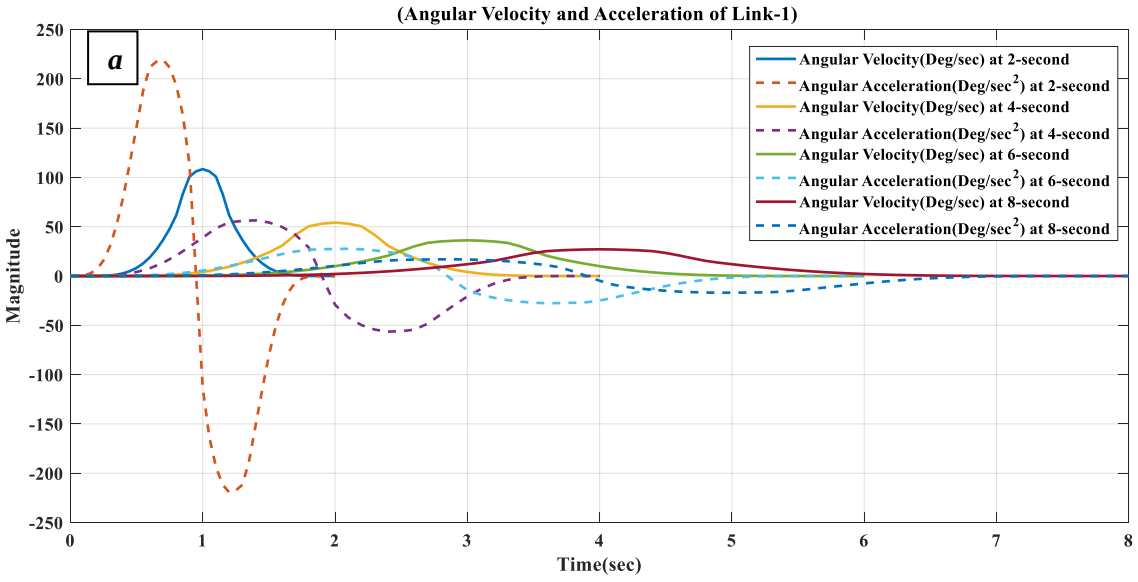


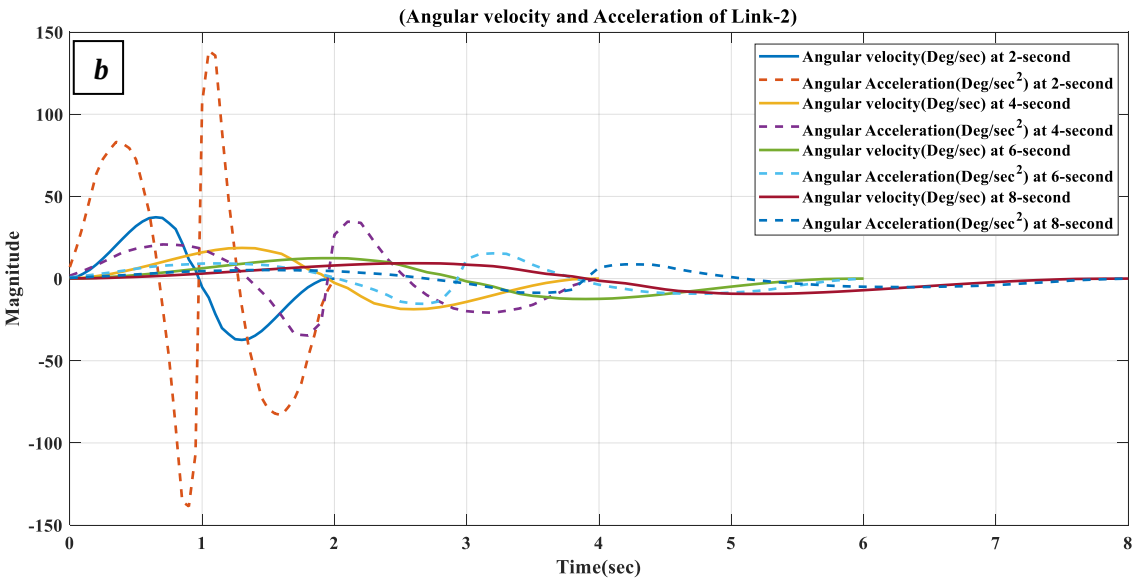
Figure(5.7) a,b,c,d Vertical Curved path Angular Displacement of Links 1-2-3-4 for (2,4,6,8) times periods.

b-Angular Velocity and Acceleration profile.

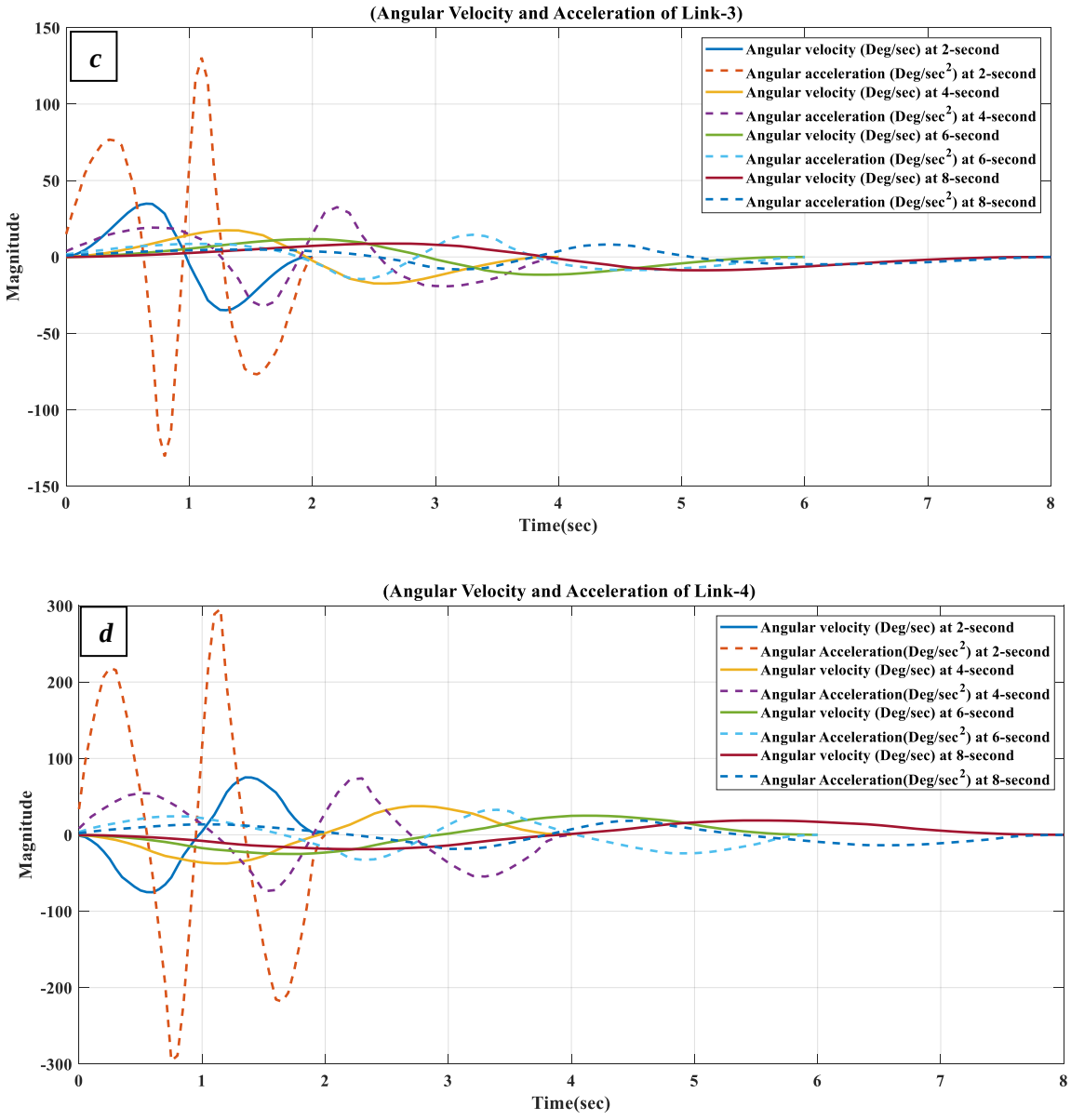
Figure(5.8)a shows the velocity and acceleration are taken by Link-1 to be displaced (sixty-three degrees) angular displacement at different times. The behavior indicated that the path displaced at two seconds required higher velocity and acceleration than others. Such that, the required velocity and acceleration inversed proportional with the period was taken. The maximum reached velocity was recorded as (109.386, 55.193, 36.701 and 27.896 degrees per seconds) for the periods (two, four, six and eight second). The same figure illustrates the variation in acceleration with time for each period for Link-1, as can note that the movement in the path with a period of two seconds requires higher acceleration than the movement in the same path in higher periods, where the maximum acceleration in the path of the period two second(215.256) degrees per second square and decreases for higher periods as (56.431, 27.694, 16.966) degrees per second square. As we noted that the acceleration has a sharp gradient when the velocity reaches the maximum limit in order

for the speed to be decreased symmetrically. **Figure(5.8)b** Explain the velocities and accelerations that Link-2 needs to move in order to achieve (Twenty degrees) of angular displacement at various periods. It was reported that the maximum velocity obtained was (37.408, 18.554, 12.318, and 9.247 degrees per second) during the durations of (two, four, sixth, and eighth seconds) respectively. The time-dependent acceleration for each period for Link-2, as we see that the maximum accelerate in the path of the period two second(140.097)degrees per second square and then decelerates for higher periods as the period increases (37.646, 16.125, 8.961) degrees per second square. In order for the speed to decay symmetrically, we observe that the acceleration has a high gradient at the maximum velocity.





Figure(5.8)c Clarifies the velocities and accelerations at which Link-3 needs to move at different times to get (Eighteenth degrees) of angular displacement. Maximum angular velocities of (33.854, 16.927, 11.733 and 8.637 degrees per second) were observed for (two, four, six, and eight seconds) of exposure, respectively. At the same time maximum angular acceleration for Link-3 recorded at two-second to be (130.351 Deg./sec^2) decreasing to (32.588, 14.841, 8.479 Deg./sec^2) for other times periods. **Figure(5.8)d** Illustrates the velocities at which Link-4 must move at various moments in order to achieve an (Negative fourteen) degree angular displacement. Maximum angular velocities of (74.152, 37.476, 25.015, and 18.789)degrees per second were observed, respectively. While maximum angular acceleration reaches (291.635,78.958, 32.878, 18.801 Deg./sec^2) For (two, four, six, and eight) seconds.



Figure(5.8) a,b,c,d Vertical Curved path Angular Velocity and Acceleration of Links 1-2-3-4 for (2,4,6,8) times periods.

By observing the curve's behavior of angular displacements, velocity and acceleration, we can see the following:

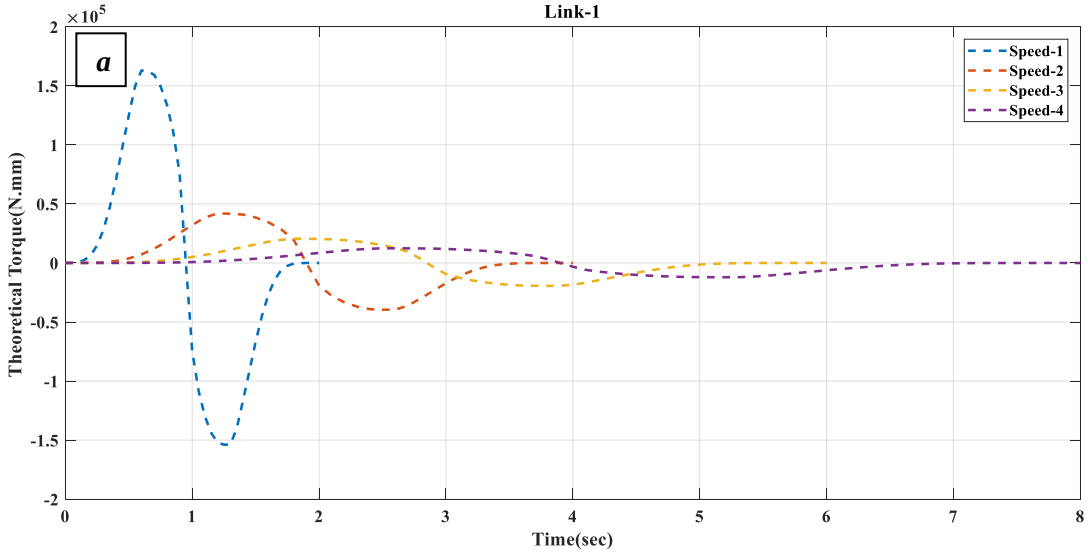
1- The velocity increases until its maximum value and then decreases, but the displacement still increases because it should reach to the final point of the path and there is no significant change in the acceleration.

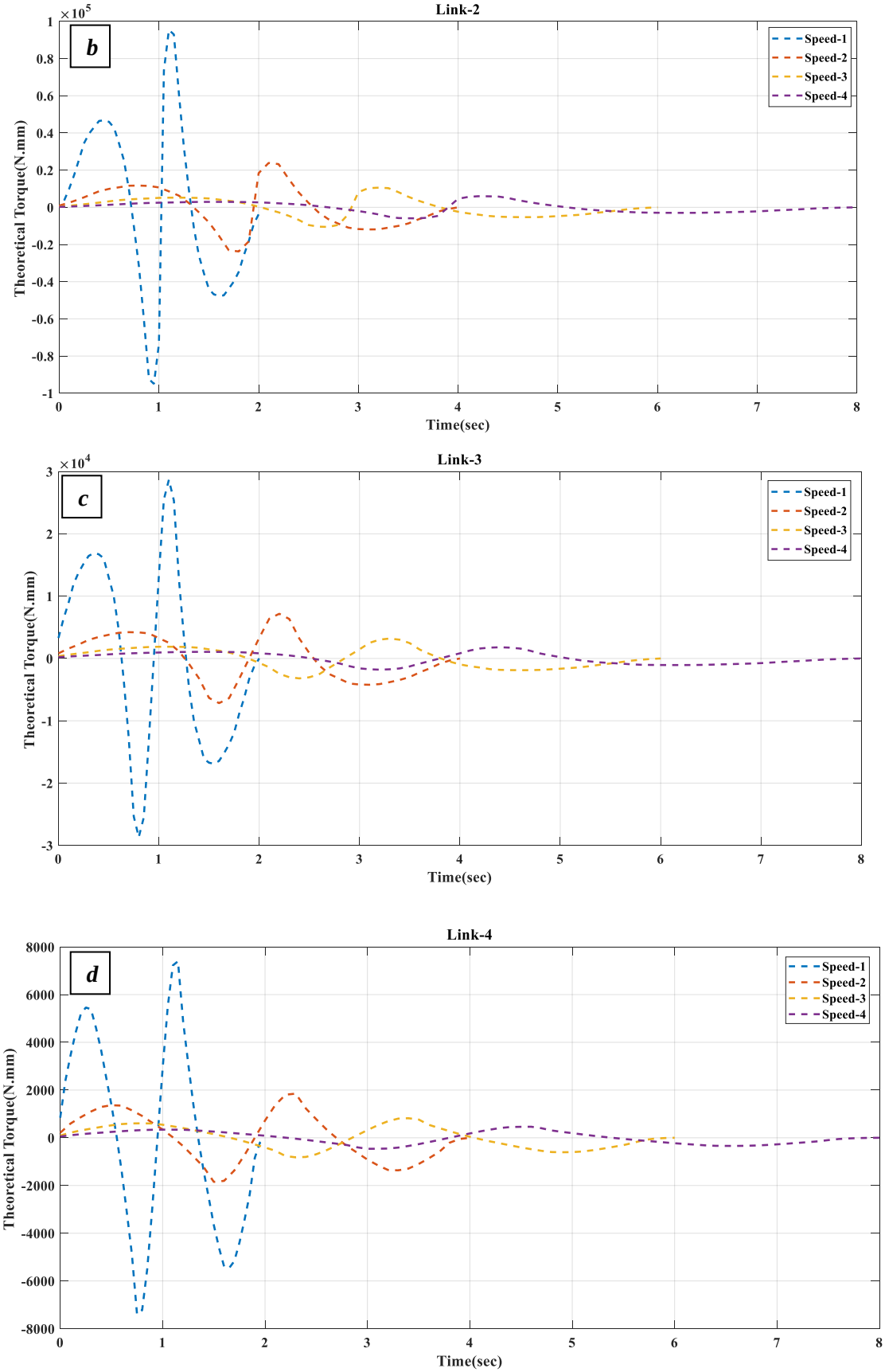
2- The maximum value of acceleration at the middle of periods was decreased because velocity decreases at this point. There is accelerate and decelerate in the robot speed according to the shape of the path.

3- Different behavior between four links attributed to the difference in the axis of rotation of the arms with respect to Global coordinates system.

c-Torque Profile

The torques operating on the four Links of the robotic arm, depicted in **figures (5.9a), (5.9b), (5.9c),** and **(5.9d),** are calculated with respect to time (t). During the motion, these forces are produced as a result of the continual variations in the acceleration that each link experiences. The magnitude of the torque exerted by a link is mostly determined by the link's mass; the greater the link's mass, the greater the inertia force that is generated by the connection. Additionally, there is a correlation between a high amount of acceleration at a certain time step and a high amount of torques at that time step.





Figure(5.9) a,b,d,c Vertical Curved path Torques of Links 1-2-3-4 for (2,4,6,8) times periods.

By watching how the torques change along the curve, we can observe the following:

1. Link-1 having the maximum value of torque rather than the other three links of the robot manipulator, because it is the only link on the track that is responsible for supporting the weight of all three arms , it's reached the higher value of (1.91E+05 N.mm) at two-second. After that the value of torque continues to decrease gradually until reaching (1.29E+04 N.mm) at eight-second. While link-4 having minimum value of (7995.879 N.mm) torque at 2-second then decreased to (462.288 N.mm) at time eight second.
2. The torque-time graphs include many peaks because of unexpected shifts in the direction of movement of the robotic arm brought on by shifts in angular acceleration, which in turn impact the torque value.

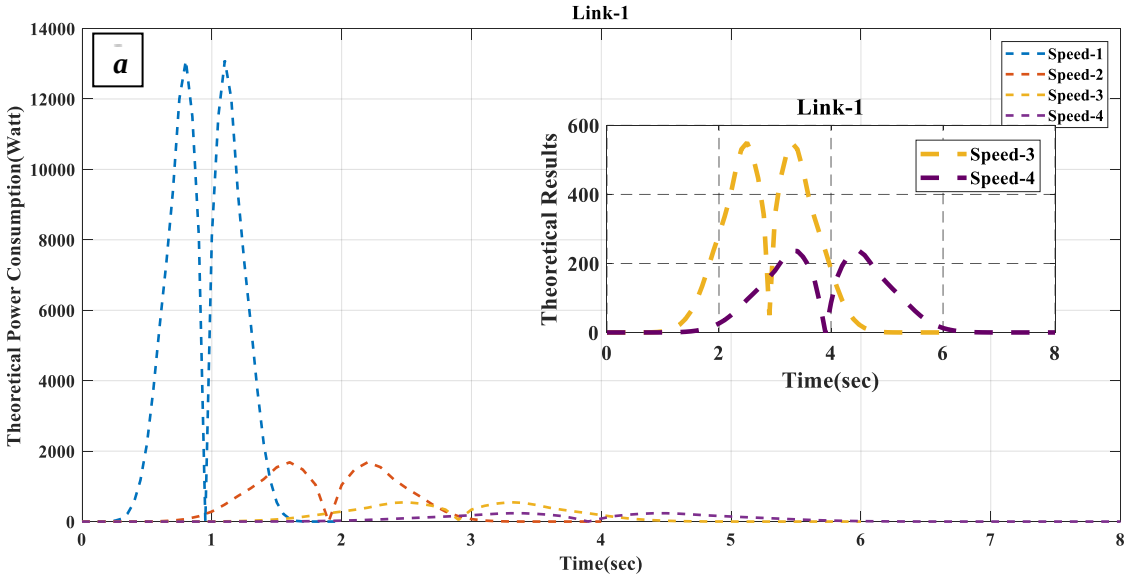
d-Power Consumption

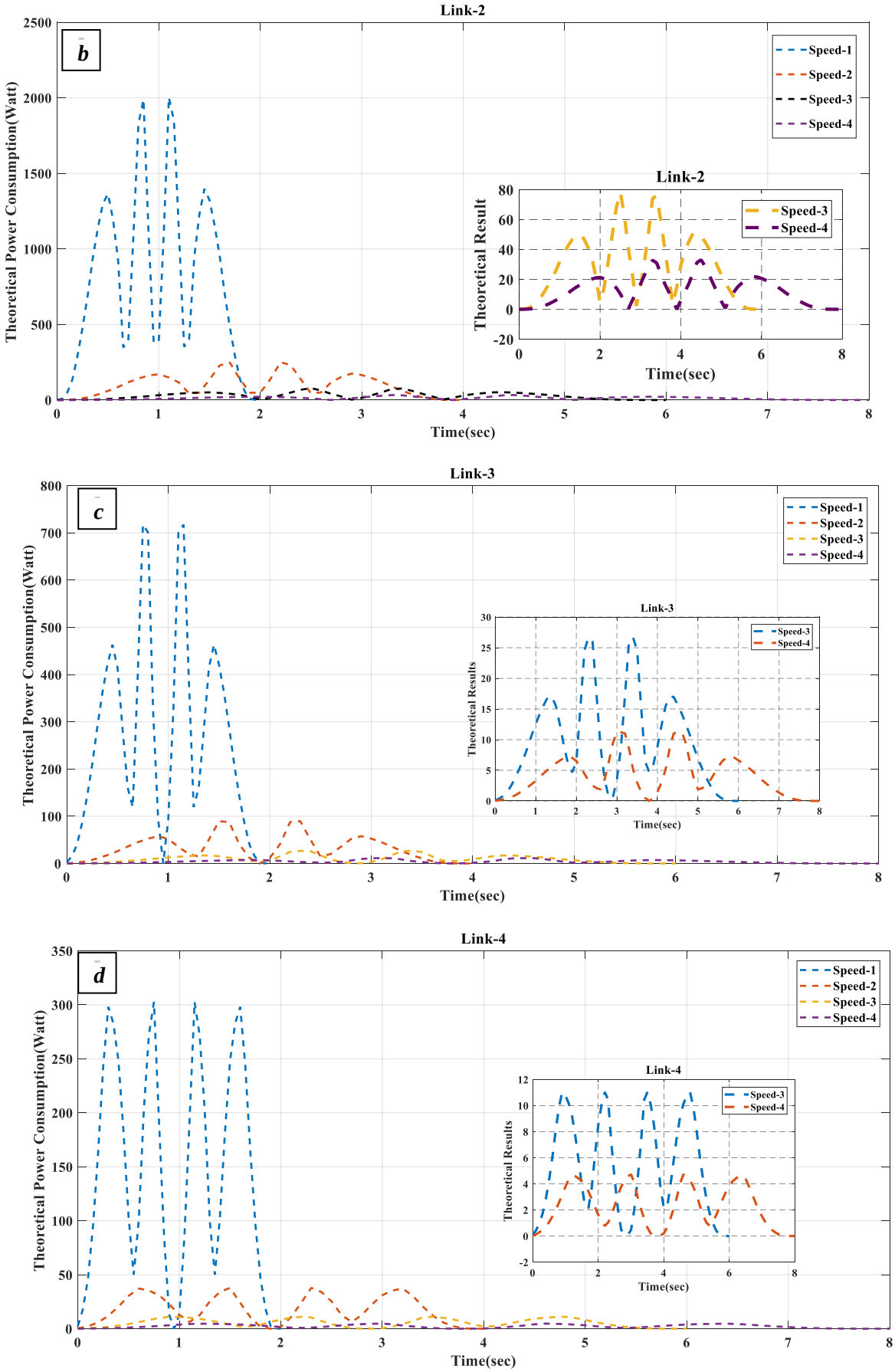
Figure(5.10)a to figure(5.10)d display a profile of the power consumption that was performed by links in order to complete the path from the starting point to the destination location. The highest rates of power consumption were seen at the briefest of time intervals, as shown by the curves below. The first link of the robot used up the most power because of its rapid speed, which is directly related to the amount of power that was used, where the maximum value reached (13089 watt at 2-second) then decreased to (332.660 watt at 2-second) for the last link which represents Link-4.

By observing the progression of the power as it moves along the curve, we may make the following observations:

1- Since the quantity of power consumption is a function of both the robot's torque and its angular velocity, the velocity had a significant impact on the total amount of power consumption. This can be seen quite well in the curve of link-1, which has the highest possible torque and angular velocity.

2- A slower angular acceleration was guided to longer operating times, which resulted in higher power consumption. However, there are some circumstances in which a higher level of productivity is required, and the robot in question must be able to perform at the highest possible level while carrying the greatest possible payload.





Figure(5.10)a,b,c,d Vertical Curved path Power Consumption of Links 1-2-3-4 for (2,4,6,8) times Periods .

e-Moment of Inertia

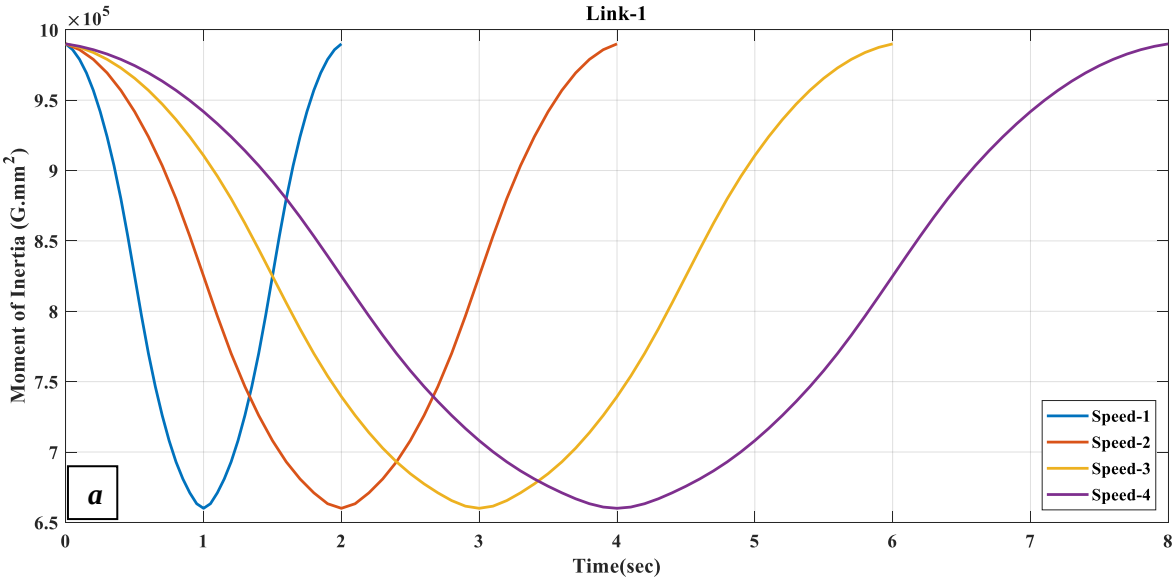
Among the parameters of rigid-body dynamics, inertia is the most challenging to detect and verify. In **Figures(5.11) (a) ,(5.11) (b) , (5.11) (c) ,(5.11) (d)** the curved path's moment of inertia was calculated, as illustrated in the figures. The moment of inertia on the first and second links is comparably variable because it is dependent on the contours of the mass and how it is distributed along the axis of rotation, and the shape of the curve varies depending on the geometry of the required trajectory. While the moment of inertia on the third and fourth linkages remains constant over time. As a result, the drawing has the appearance of a straight line parallel to the x-axis.

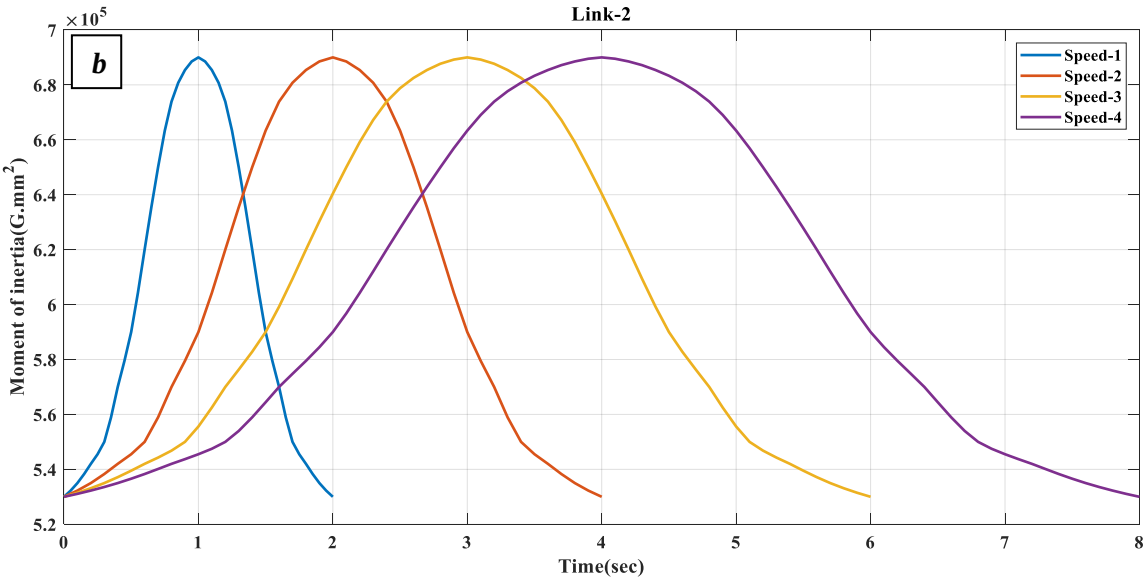
By tracking the inertial moment's evolution along the curve, we may discover the following:

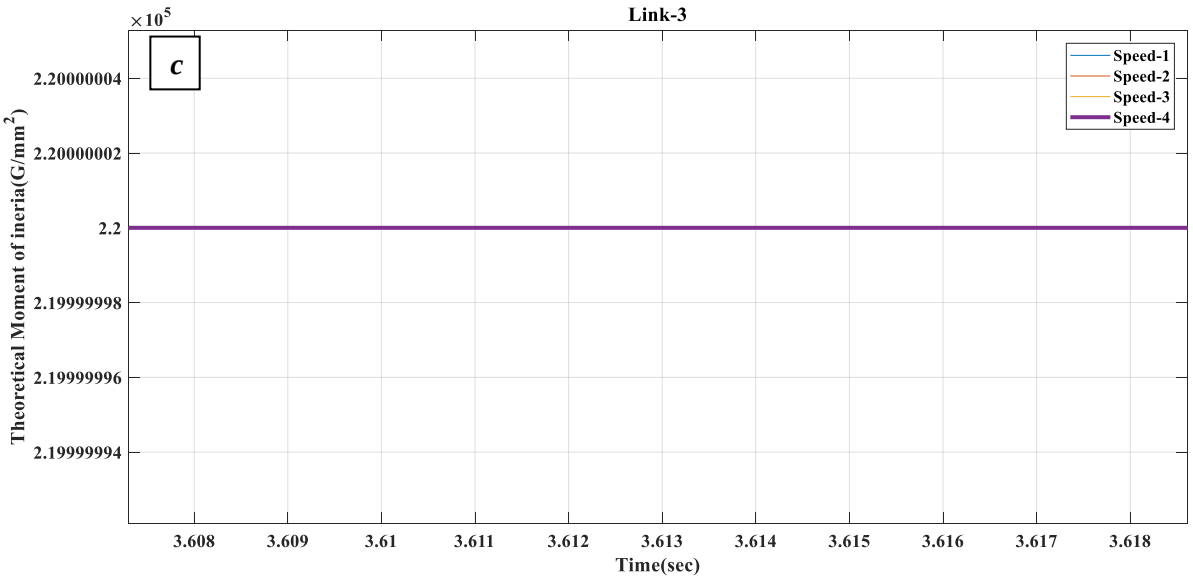
1- The value of the moment of inertia for Link-1 along the path remains the same throughout all four time intervals at the start point, which has a value of (9.90E+05 g.mm^2) However, at the middle of the time interval, which represents the center of the path, the value of the moment of inertia decreases to ($6.5*10^5$ g.mm^2) this is because the link is closest to the global axis of rotation at this point. Link-4 has the smallest and constant moment of inertia along the path at the value of (2.5*10^4 g.mm^2) for all the time intervals.

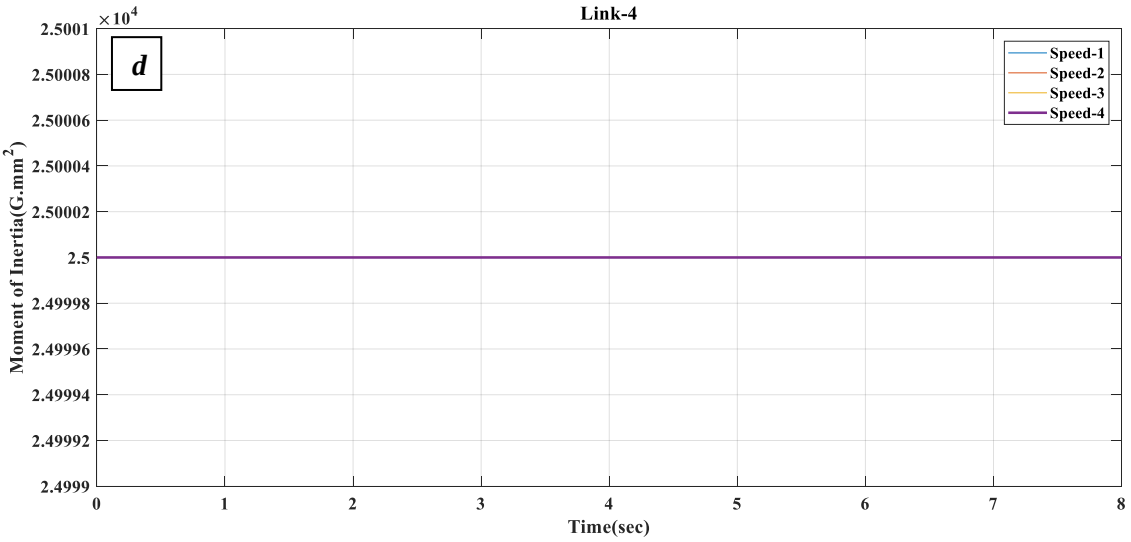
2- Moments and products of inertia may vary in value when the orientation of the axis relative to the body changes.

3- The inertia of Link-2-3-4 depends on how far away the center of all links is from the axis of the first joint.


Link-1
×10^5
Moment of Inertia (G.mm^2)
Time(sec)
a
Speed-1
Speed-2
Speed-3
Speed-4


Link-2
×10^5
Moment of inertia(G.mm^2)
Time(sec)
b
Speed-1
Speed-2
Speed-3
Speed-4


Link-3
×10^5
Theoretical Moment of ineria(G/mm^2)
Time(sec)
c
Speed-1
Speed-2
Speed-3
Speed-4



Figure(5.11) a,b,c,d Horizontal Curved path Moment of Inertia of Links 1-2-3-4 for (2,4,6,8) times Periods.

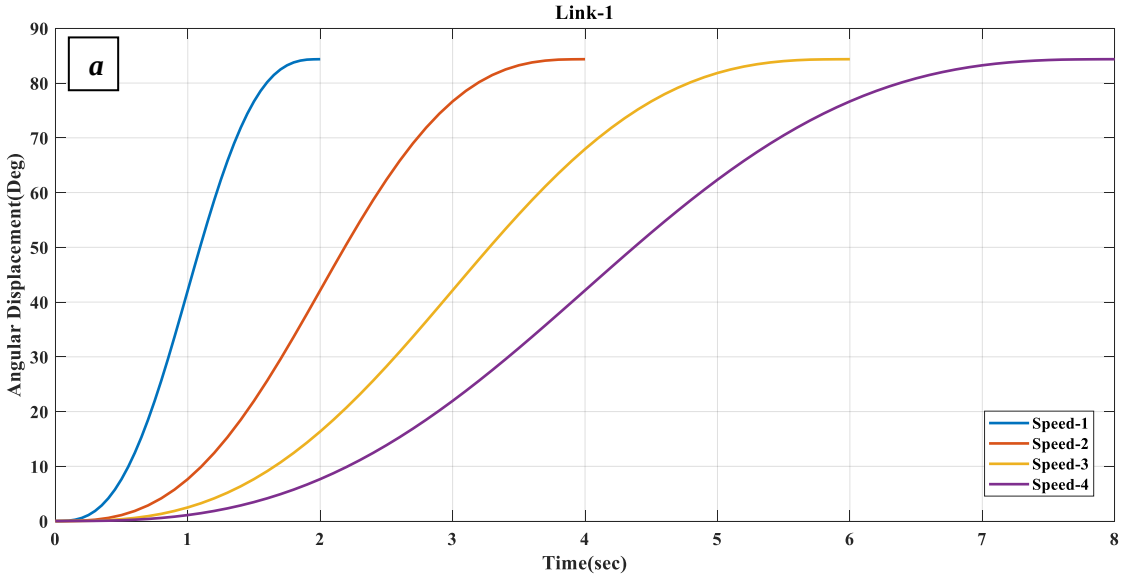
5.4.2 Horizontal Path Results

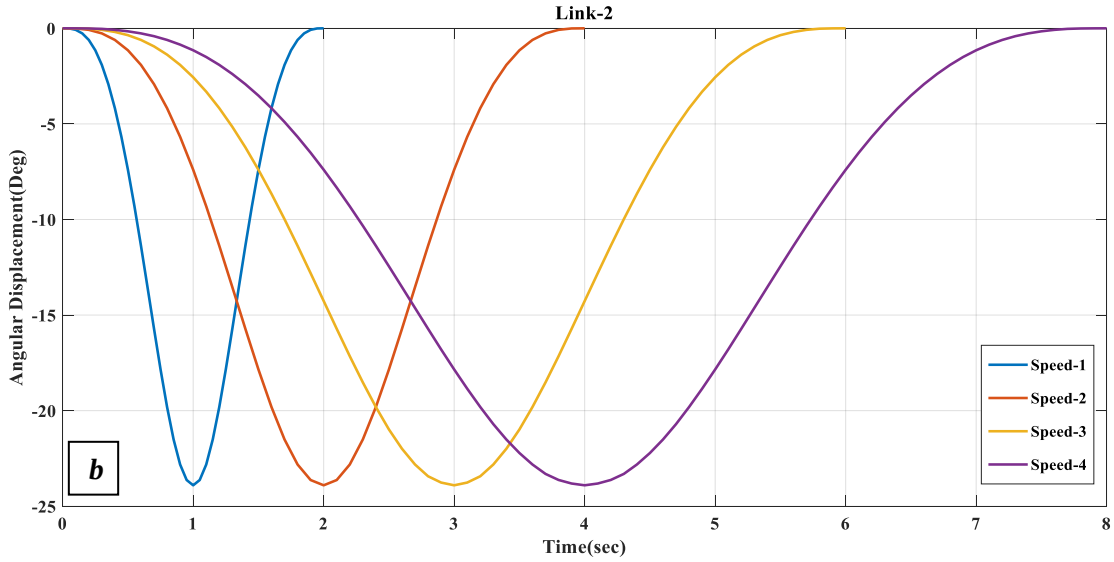
The results of the kinematics and dynamics of a horizontally curved path are explained. The curving path Simulation from the starting point, where the X, Y, and Z axes are (0mm, 0mm, 0mm), to the middle point, where the axes are (250mm, 250mm, 0mm), to the end point, where the axes are (500mm,0 ,0).

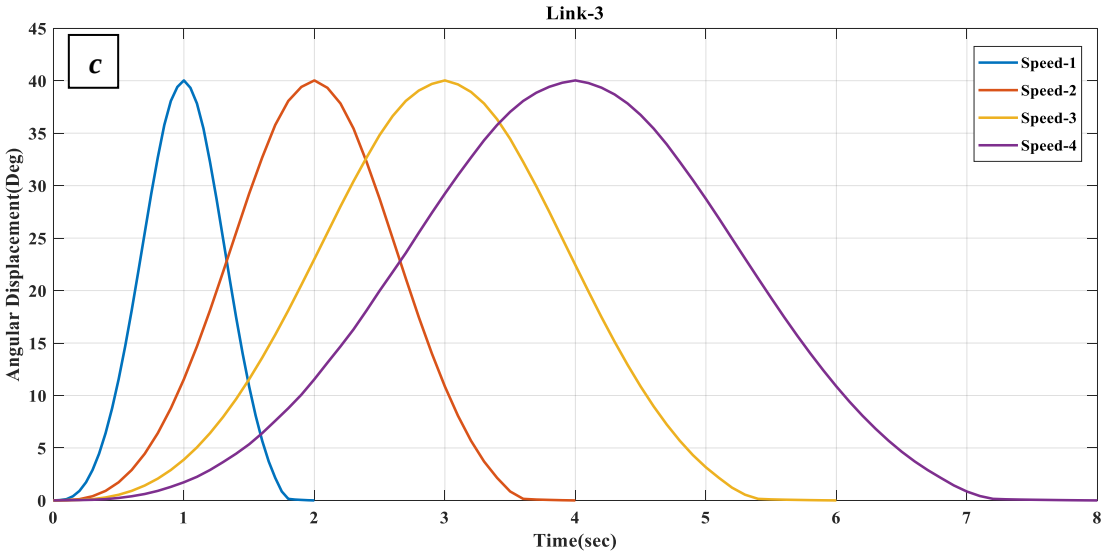
5.4.2.1 Kinematics and Kinetics of Links-1-2-3-4 for Horizontal Curved Path

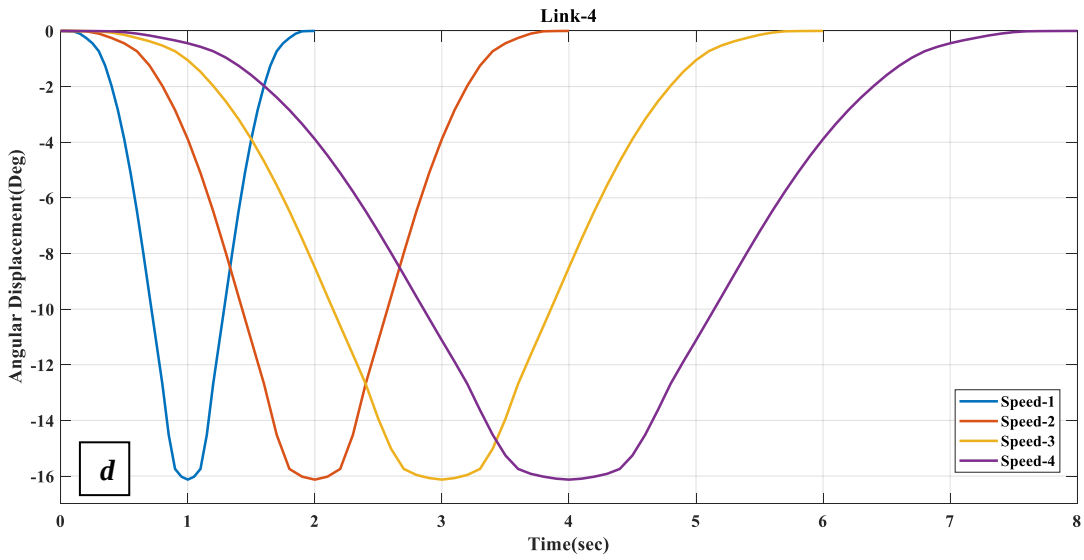
a-Displacement Profile

It can be seen in **figure(5.12) (a)** that the Link- 1's displacement varies both in direction along the path (counter clockwise). **Figure (5.12) (b)** depicts Link-2's track reversal and crosses in a clockwise orientation. As seen in **figure(5.12)(c),** Link-3 may turn around and go across the path in a counterclockwise manner. In **figure(5.12)(d),** the varied time periods require Link -4 to go clockwise along the path is clarified**.**


Link-1
a
Angular Displacement(Deg)
Time(sec)
Speed-1
Speed-2
Speed-3
Speed-4


Link-2
b
Angular Displacement(Deg)
Time(sec)
Speed-1
Speed-2
Speed-3
Speed-4


Link-3
c
Angular Displacement(Deg)
Time(sec)
Speed-1
Speed-2
Speed-3
Speed-4

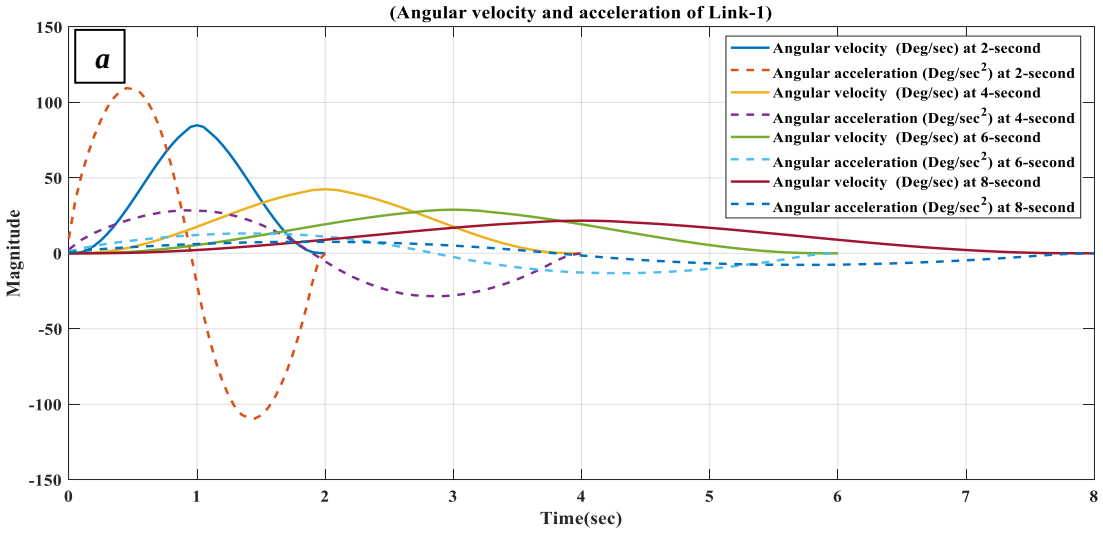


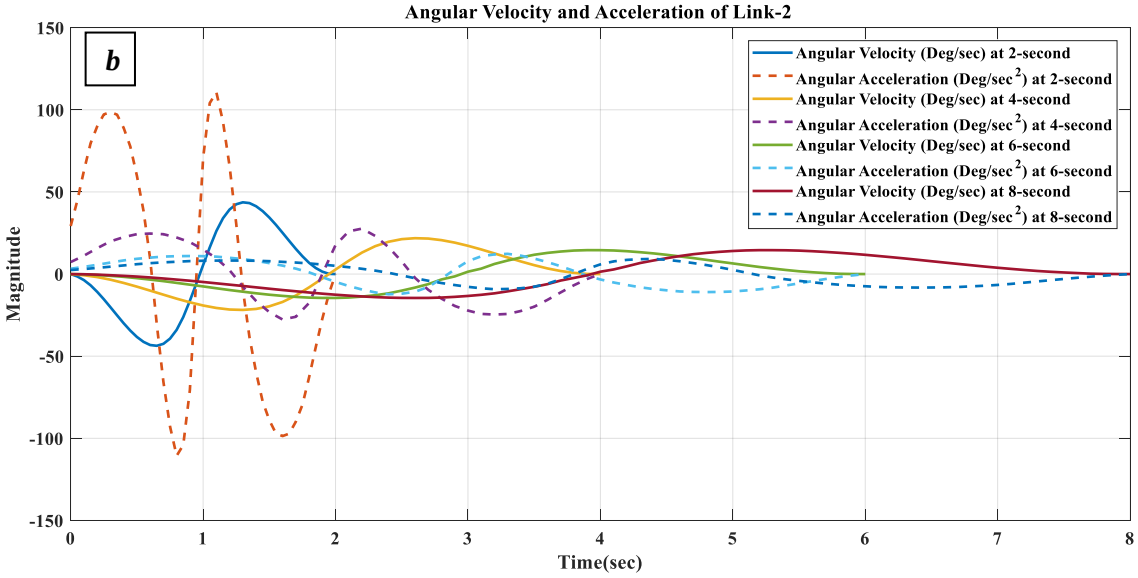
Figure(5.12)a,b,c,d Horizontal Curved path Angular Displacement of Links 1-2-3-4 for (2,4,6,8) times periods

b-Angular Velocity and Acceleration profile

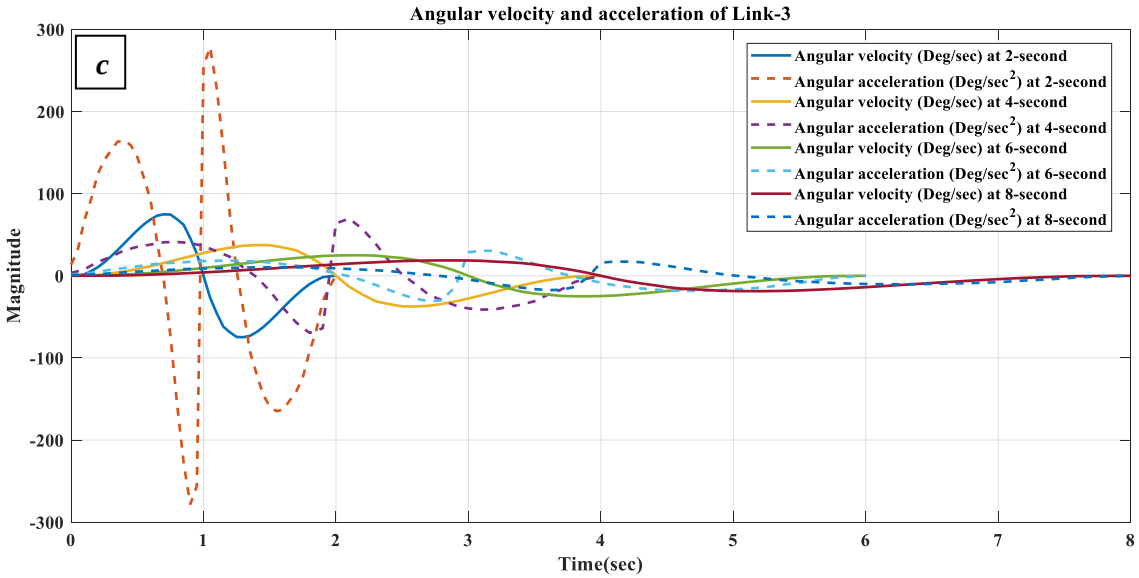
In **figure (5.13)a,** we can see that the varying velocities and accelerations Link-1 requires to rotate by (eighty-four) degrees at various times. A higher velocity needed to go along the path that was displaced after two seconds compared to the other paths, where the needed velocity inversely proportional to the time period. For times (2, 4, 6, and 8) seconds, the highest observed velocities were (84.758, 42.731, 9.529, and 5.367 degrees per second), respectively. The acceleration needed to move along the path with a period of two seconds is greater than the acceleration needed to move along the same path with higher periods, with the maximum speed in the path of the period two second(108.780 degrees per second square) and decreases for higher periods as the period increases (29.826, 4.322, 1.982) degrees per second square. When the speed approaches the maximum limit, the acceleration has a strong gradient that declines in speed symmetrically. **Figure(5.13)b** explains the angular velocity at which Link-2 must travel at different times in order to achieve (negative twenty-four degrees) of rotational displacement.

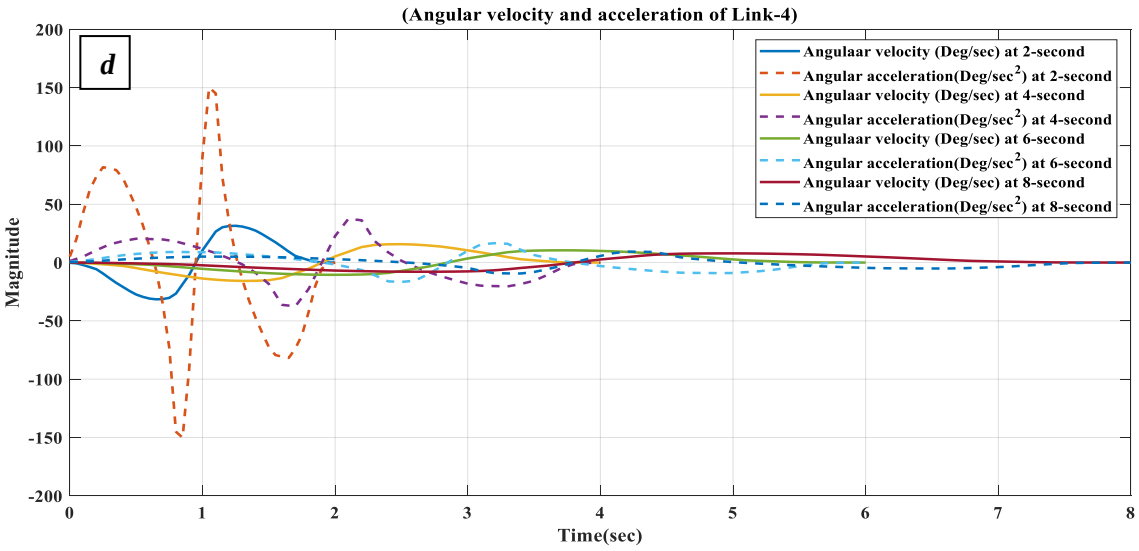
Maximum angular velocities of (44.991, 21.834, 14.951 and 11.314 degrees per second) were reported for (2, 4, 6, and 8 seconds) respectively. Link-2's time-dependent acceleration is also shown in **figure(5.13)b**, where we can see that the maximum speed along the path reached at a period of two seconds (110.472) degrees per second squared, and then decreases for longer time periods (27.968, 10.627, 8.227) degrees per second squared. We see that the greatest acceleration has a steep gradient if the speed is to decline symmetrically from that point.





Figure(5.13)c Clarifies the timing and speeds at which Link-3 must travel, so that a (forty degree) angular displacement can be achieved. There was a maximum angular velocity of (73.178) degrees per second for a duration of two- seconds, followed by (37.210) degrees for four-seconds, (24.79) degrees per second for six second, and (18.742) degrees per second for eight seconds. Similarly to what is seen in **figure(5.13)c**, Link-3 had the greatest angular acceleration reaches the value of (278.306) degree per seconds square at time two seconds, for others times the values of maximum angular accelerations are recorded as (63.333, 30.110, 17.435 Deg./sec^2). In **figure(5.13)d**, the velocities and accelerations that Link-4 must travel at to achieve a (negative sixteen) degree angular displacement are shown. The maximum angular velocities were (31.321, 15.766, 10.476, and 7.859 degrees per second) for 2, 4, 6, and 8 seconds, respectively. While the greatest angular accelerations observed at the four time periods are (156.130, 37.932, 16.614, and 9.384 Deg./sec^2).





Figure(5.13)a,b,c,d Horizontal Curved path Angular Velocity and acceleration of Links 1-2-3-4 for (2,4,6,8) times periods

c-Torque Profile

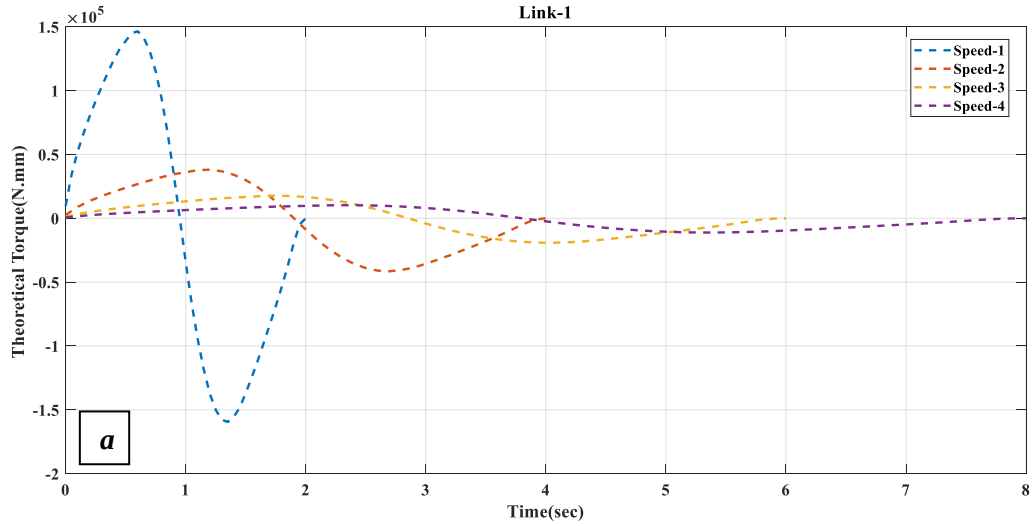
Figure(5.14)(a),(5.14)b,(5.14)c,(5.14)d show how the torques on the four links of the robotic arm are calculated with time. During the movement, these forces are caused by the constant changes in acceleration that each link goes through. The mass of a link is the main factor that determines how much torque it can produce. The more mass a link has, the more inertia force it creates. Also, a high amount of acceleration at a certain time step and a high amount of torques at that time step go together.

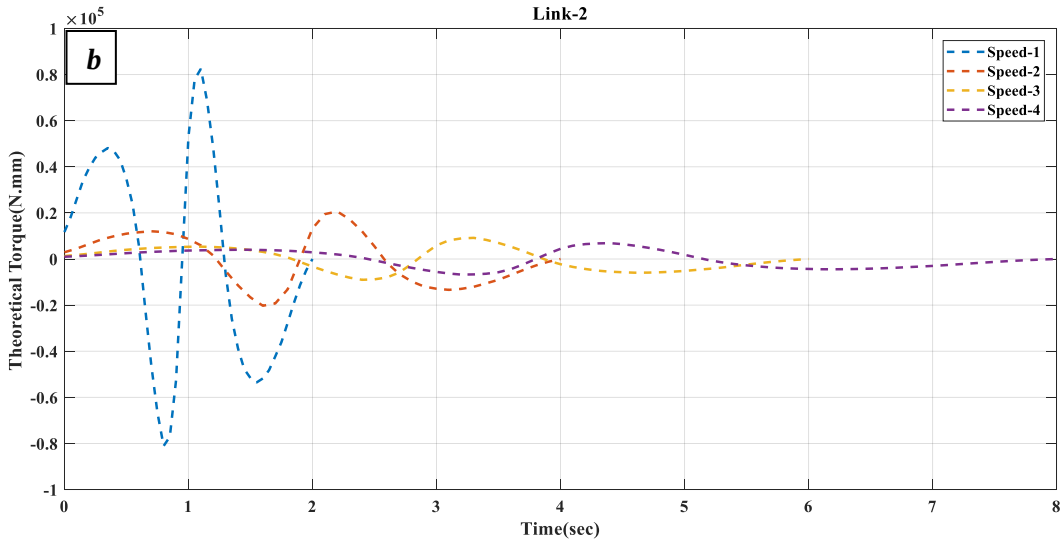
By observing the progression of the torques along the curve, we are able to make the following observations:

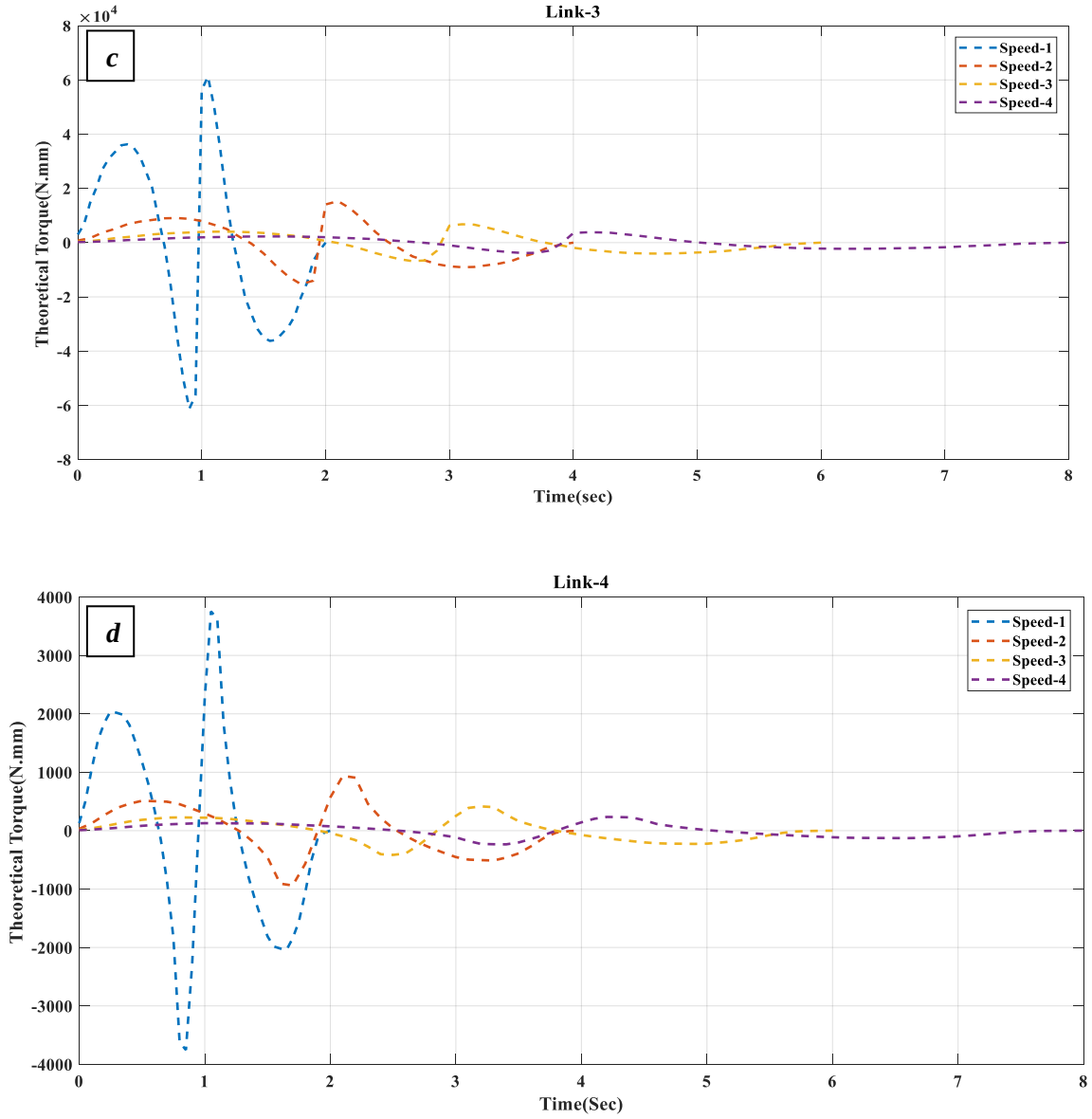
1. Link-1 because it is the exclusive link on the track that is responsible for supporting the weight of all three arms, it has achieved the higher value of (1.48E+05 N.mm) at two seconds, giving it the highest value of torque compared to the other three

links of the robot manipulator. After this point, the torque value may continue to progressively drop until it reaches the eight-second point.

2. The torque-time graphs include several peaks because of unanticipated changes in the direction of movement of the robotic arm. These changes were caused by changes in the angular acceleration, which in turn had an effect on the torque value.







Figure(5.14)a,b,c,d Horizontal Curved path Torques of Links 1-2-3-4 for (2,4,6,8) times periods

d-Power Consumption

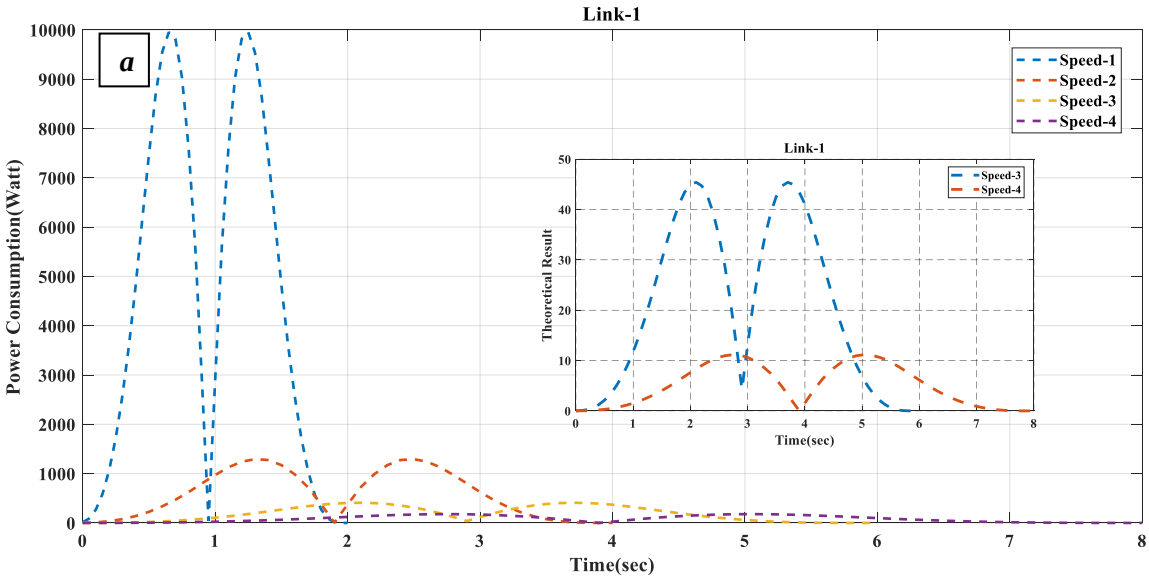
A profile of the power consumption that was carried out by links in order to finish the path from the beginning point to the location of the destination is displayed in **figure (5.15)a** through **figure (5.15)d** respectively. According to the curves that are shown below, the rates of power consumption that were the greatest during the shortest time intervals. The first link of the robot used up the most power due to its high speed, which is directly related to the amount of the used power,

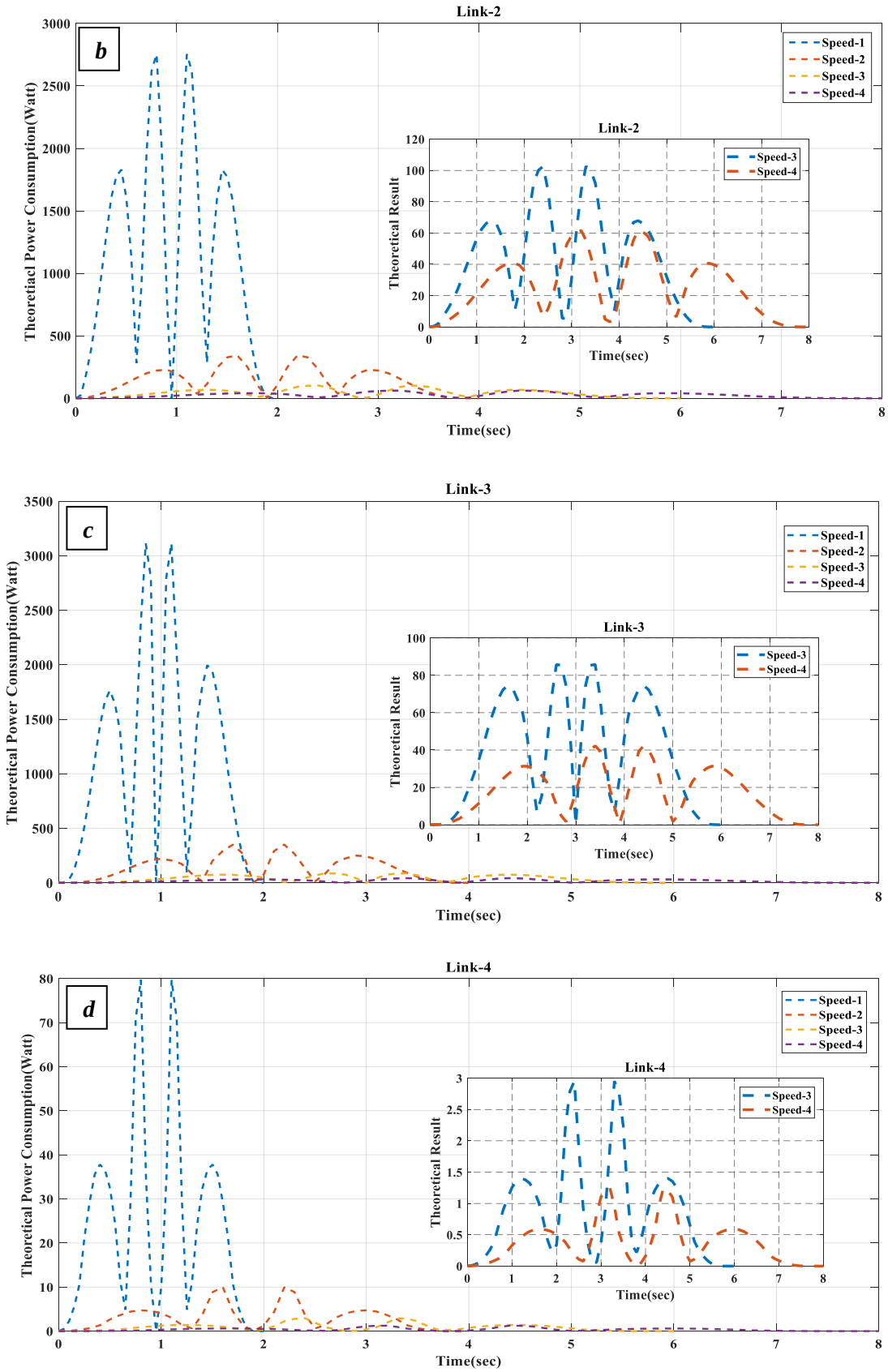
where the maximum value reached (9932.417 watt at 2-second), then decreased to (79.988 watt at 2-second) for the last link, which represents Link-4. This is due to the amount of the power used.

The following may be determined by tracking the power as it is applied at different points along the curve:

1- Power consumption is a function of the robot's torque and its angular velocity, hence increasing the robot's speed significantly increased its power consumption. This is clearly seen in the link-1 curve, which exhibits the maximum torque and angular velocity.

2- A lower angular velocity may result during the longer amount of time spent operating, which led to an increase in the amount of power consumption. However, there are other situations in which a higher degree of productivity is necessary. In these cases, the robot in concern needs to be able to function at the highest possible level while also being able to carry the most possible payload. As a direct consequence of this, it is essential to carry out an optimization of the functioning parameters of the robot.



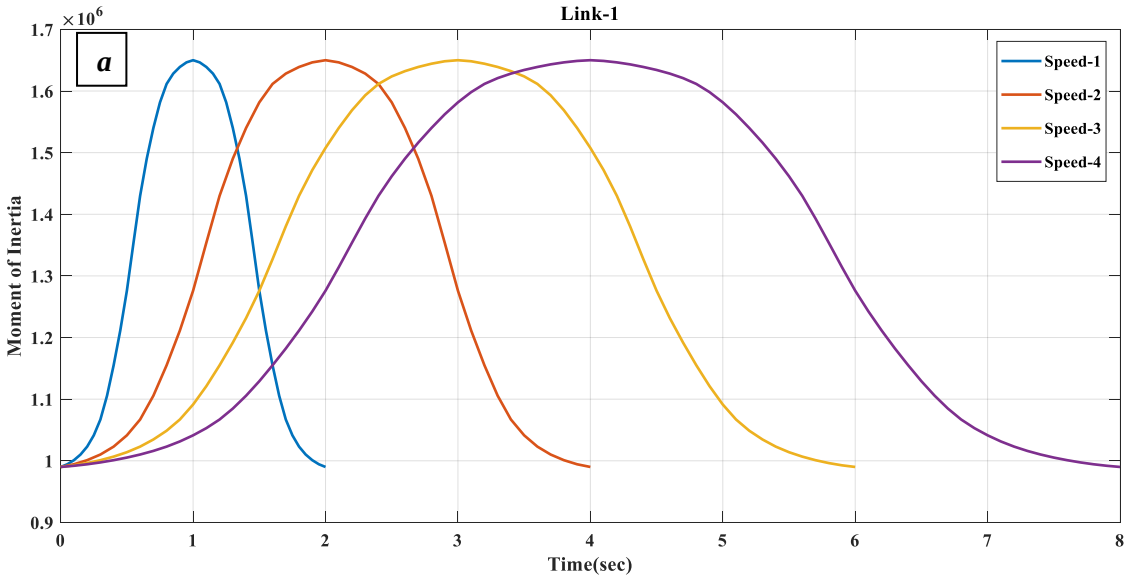


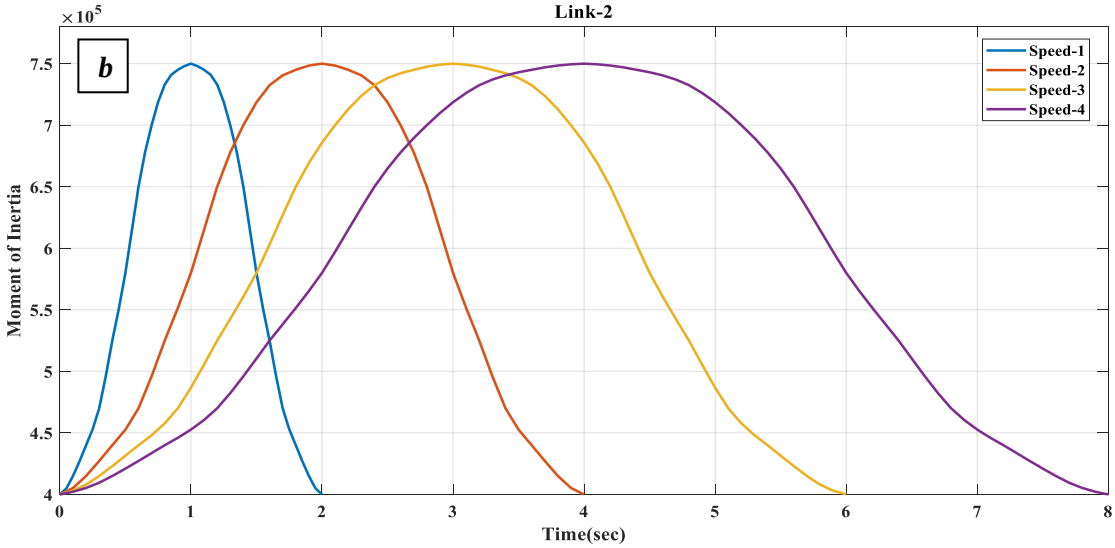
Figure(5.15)a,b,c,d Horizontal Curved path Power Consumption of Links 1-2-3-4 for (2,4,6,8) times periods.

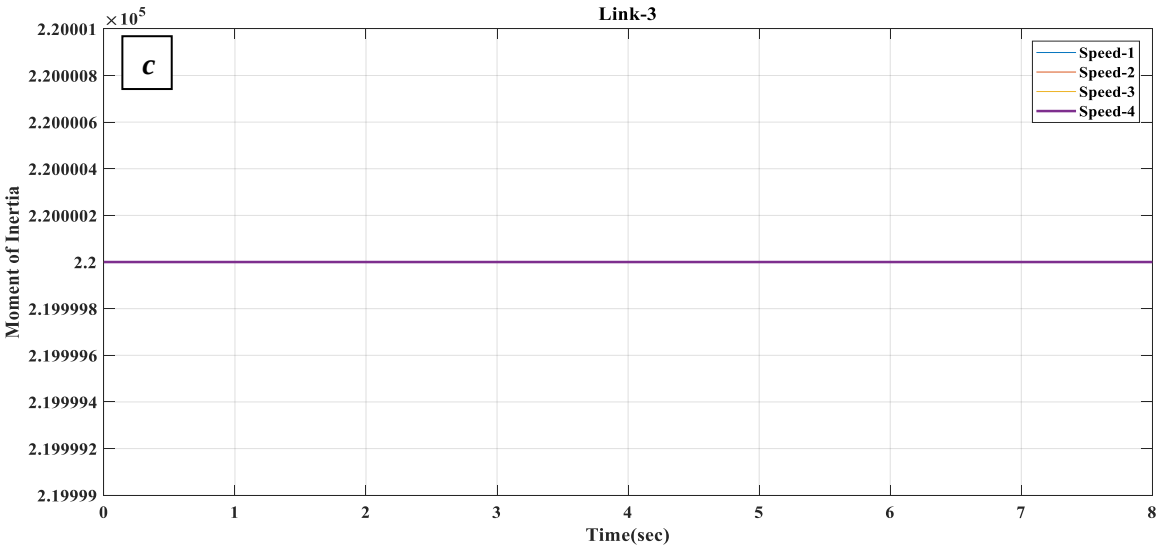
e-Moment of Inertia.

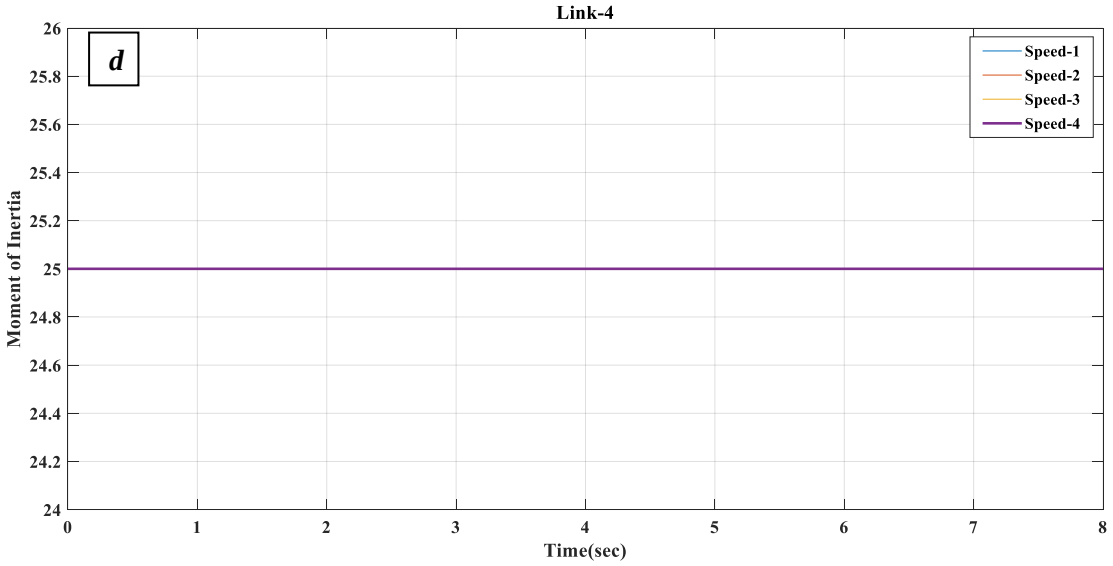
Inertia is the parameter in rigid-body dynamics that presents the greatest challenge in terms of determining and verifying its value. **Figures(5.16) (a) ,(5.16) (b) , (5.16) (c) ,(5.16) (d)** show the moment of inertia of the curved path, where the results are depicted. Because it is reliant on the contours of the mass and how it is distributed along the axis of rotation, the moment of inertia on the first and second links is relatively changeable. Additionally, the form of the curve changes depending on the geometry of the needed trajectory, but the moments of inertia on the third and fourth links do not change during the course of time. As a consequence of this, the line drawing seems like it is running in a straight line that is parallel to the x-axis. The following may be gained by observing the inertial moment as it changes along the curve:

1- At the initial location, the moment of inertia for Link-1 along the path has maximum value of (1.65E+06 g.mm2), and this value is constant for all four time periods. However, the moment of inertia of link-2 reduces to (6.5*10^5 g.mm2) at the time interval's center, which reflects the path center of mass. This is due to the fact that the axis of rotation of the base coincides with the location of the connection. The moment of inertia of Link-4 is the least at (25 $g.mm^2$) and remains relatively consistent along the course and over all time intervals.
2- When the axis's direction changes relative to the body, consequently the value of the moment and product of inertia decreased.
3- Distance from the first joint's axis determines how distant the center of mass of Links 2-4 is from the rest of the chain.


Link-1
×10^6
a
Speed-1
Speed-2
Speed-3
Speed-4
Moment of Inertia
Time(sec)


Link-2
×10^5
b
Speed-1
Speed-2
Speed-3
Speed-4
Moment of Inertia
Time(sec)


Link-3
×10^5
c
Speed-1
Speed-2
Speed-3
Speed-4
Moment of Inertia
Time(sec)



Figure(5.16) a,b,c,d Horizontal Curved path Moment of Inertia of Links 1-2-3-4 for (2,4,6,8) times periods.

5. 5 Experimental Results

The purpose of the laboratory tests was to demonstrate that the Robot Manipulator could be controlled by LABVIEW and made to follow a specific path accurately. The communication strategy between the SOLIDWORKS and LabVIEW may be thought of as a Master-Slave relationship due to the nature of their connection. LabVIEW functions as the Master since LabVIEW is the one doing the task, possesses general control over the system and may perform actions such as initiating communication between two programs, terminating a simulation, receiving inputs, and so on. Both SOLIDWORKS and LabVIEW are able to provide users with the ability to retrieve outputs. The findings are presented in the same approach, with a brief summary and visual representations of the important variables.

5.5.1 Experiment 1:Vertical Path Real Time Test

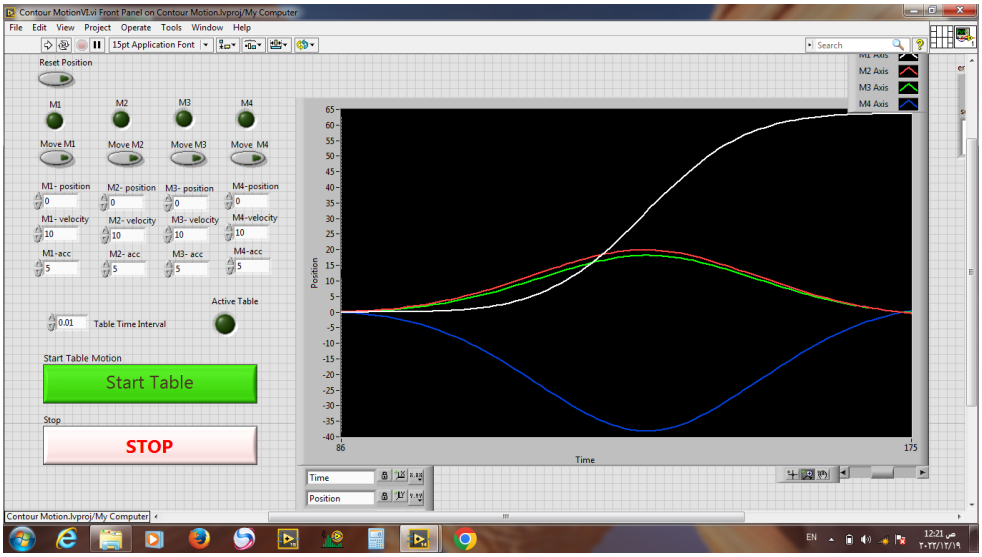
The first experiment aims to use the LabVIEW program to instruct the robot to follow the vertical path. Due to the effects of clearance in mechanical elements such as the gearbox and friction between the parts,

which were not included in the theoretical part, a little gap will be indicated between the practical and theoretical findings in the comparison section.

5.5.1.1 Experiment Kinematics And Kinetics of Links-1-2-3-4 for Vertical Curved Path

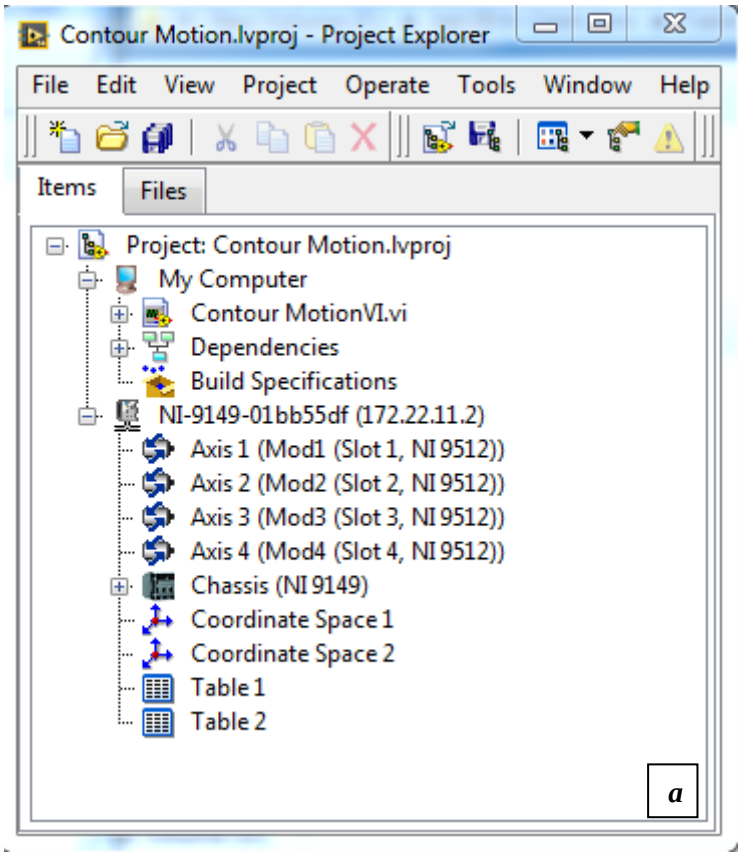
Figure(5.17) illustrates LabVIEW program window that shows the input angular displacement value that obtained from the simulation process in order to extract the curved path for the (four Links) of vertical curved path.

1- The white color line represents the displacement of motor-1 to move link-1 along path.
2- Red color line illustrates the displacement of motor-2 in order to move Link-2 from start point to distention point.
3- Green color line shows the displacement of motor-3 to make Link-3 tracks the desired path.
4- Blue color line explains displacement of motor-4 to forces Link-4 follow the curved path.

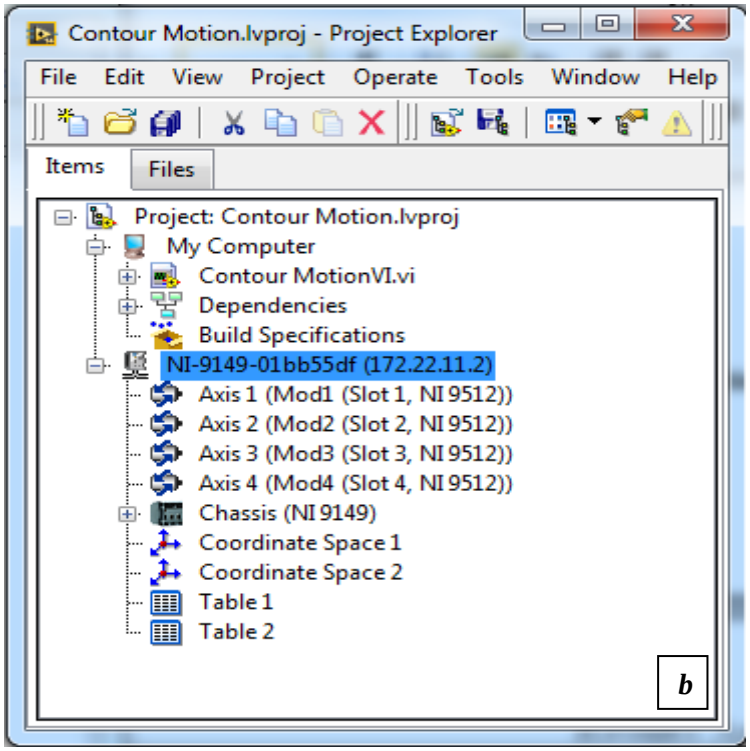


Figure(5.17) LabVIEW program window.

After that, the LabVIEW NI- SOFTMOTION axes that have been added to the project manager are the ones that activate the motors that have been described in the SOLIDWORKS Assembly[54]. If a specific motion is to be carried out concurrently by much more than one motor, then the LabVIEW project requires the addition of a coordinate space defined by the axes of each of the various motors. Both SOLIDWORKS and LabVIEW need to fulfill a number of conditions in order to achieve the goal of performing a seamless integration of the two applications. In this work, in order to perform the motion of robot along the curved path four axis four motors should be programed in (LABVIEW) as illustrated in **figure(5.18) a, figure(5.18) b.**



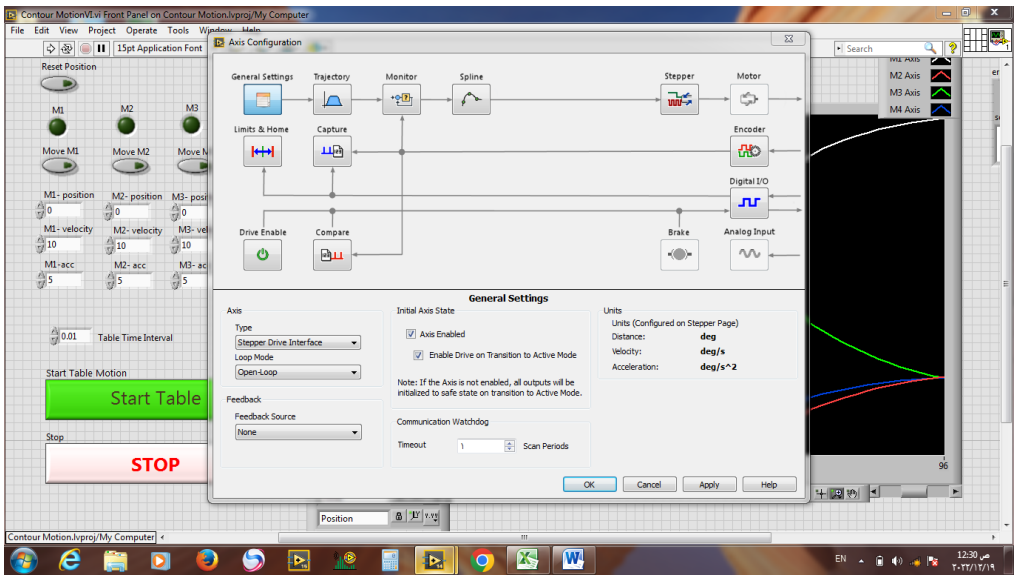
(a)



(b)

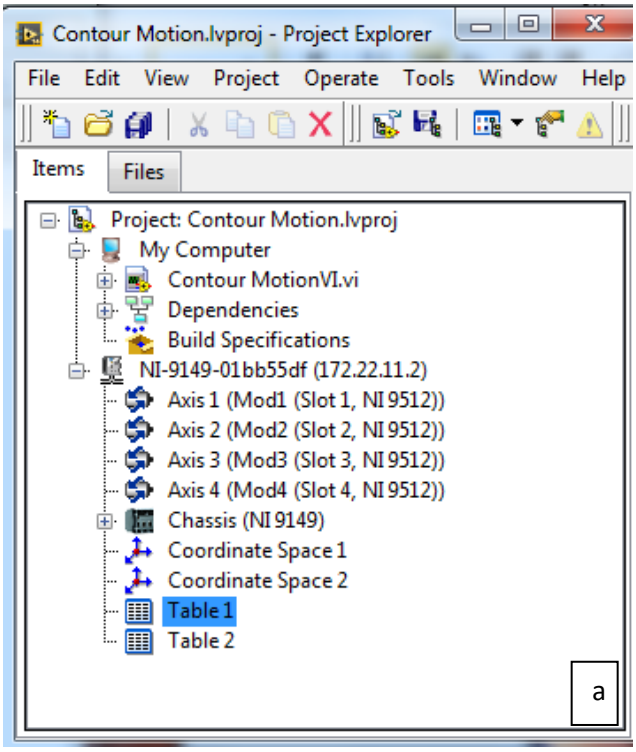
Figure(5.18) (a) (b) Project Explorer of contour motion

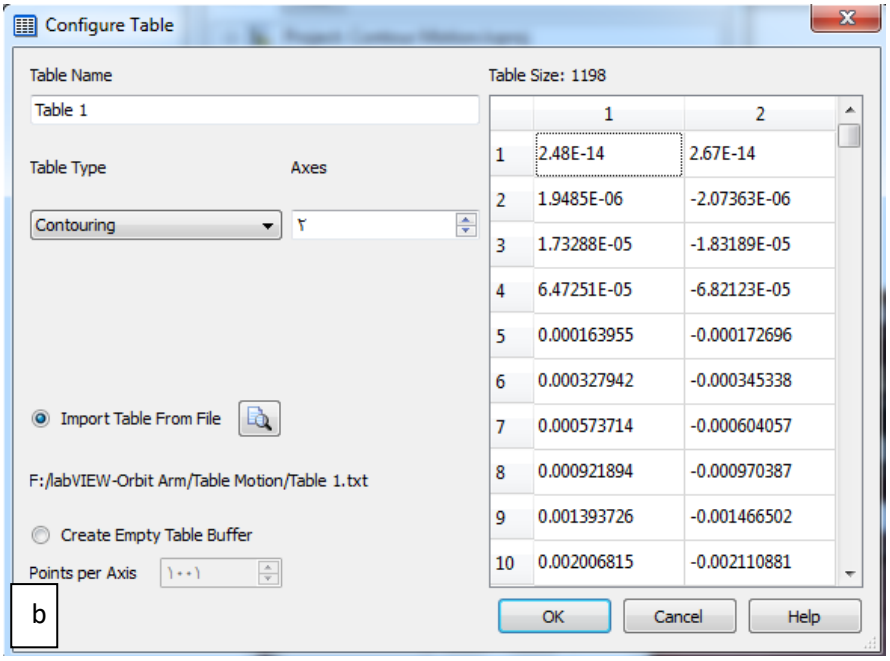
The interface for preparing and programming the motor, the conversion ratio, and the process of electronically controlling the movement of the motor all these are illustrated in **figure(5.19) .**



Figure(5.19) Axis configuration

After the process of controlling the movement of the motors the step of extracting the results, where exported as a table of data, can be seen in **figure(5.20) a and b.**



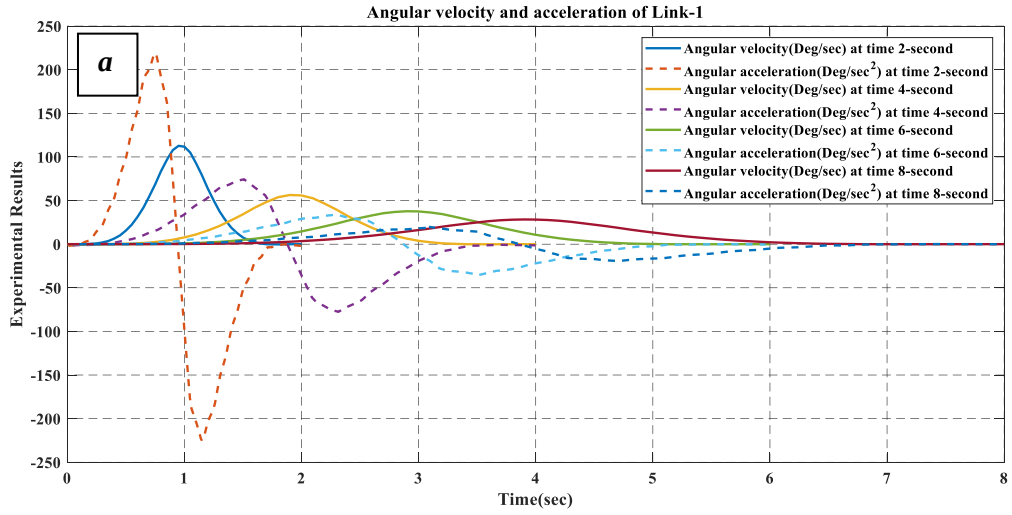


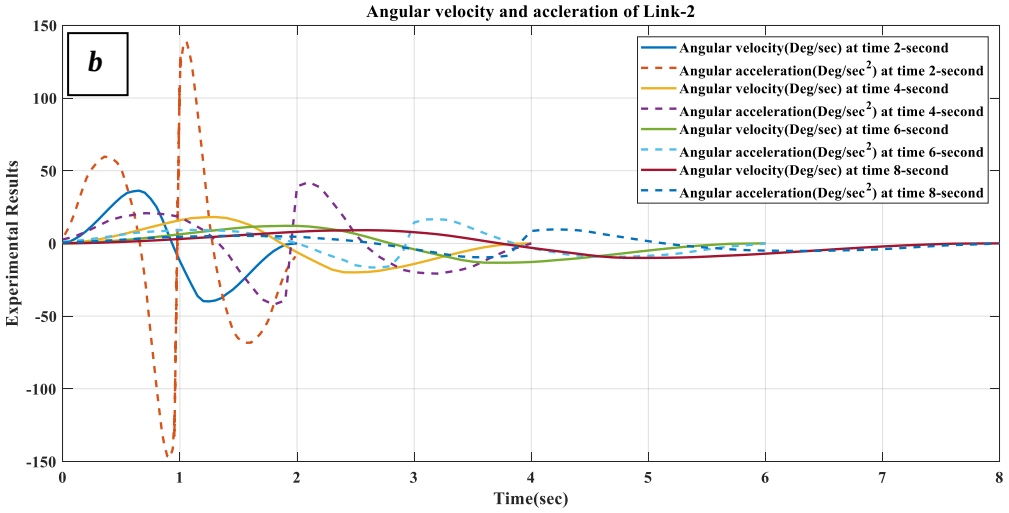
	1	2
1	2.48E-14	2.67E-14
2	1.9485E-06	-2.07363E-06
3	1.73288E-05	-1.83189E-05
4	6.47251E-05	-6.82123E-05
5	0.000163955	-0.000172696
6	0.000327942	-0.000345338
7	0.000573714	-0.000604057
8	0.000921894	-0.000970387
9	0.001393726	-0.001466502
10	0.002006815	-0.002110881

Figure(5.20)a,b Axis configuration.

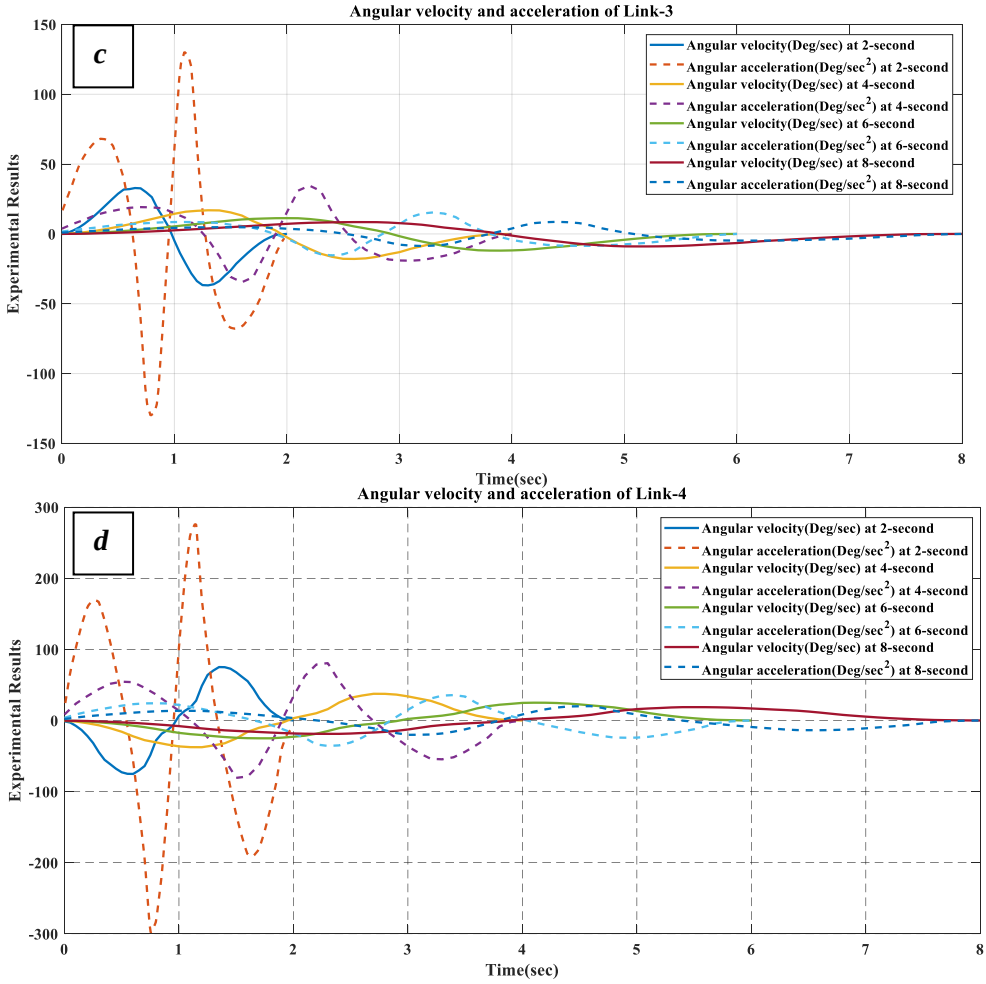
a-Experimental velocity and acceleration profile.

Figure(5.21) illustrates the velocity and acceleration profile for vertical path of Links -1-2-3-4 from LabVIEW program that displaced the robot (63 degree) at four time periods. Maximum value of angular velocity and acceleration for link-1 are (111.236 deg./sec and 222.185 deg./sec^2) respectively, While Link-2 having the maximum angular velocity and acceleration of (37.408 deg./sec and 146.097 deg./sec^2) as shown in **figure(5.21) (a)(b)**.





Figure(5.21) (c)(d) shows the angular velocity and acceleration curves for Link-3-4 respectively, the maximum of (32.954 deg./sec and 135.350 deg./sec^2) for Link-3, while Link-4 having maximum value of (75.952 deg./sec and 299.835 deg./sec^2).

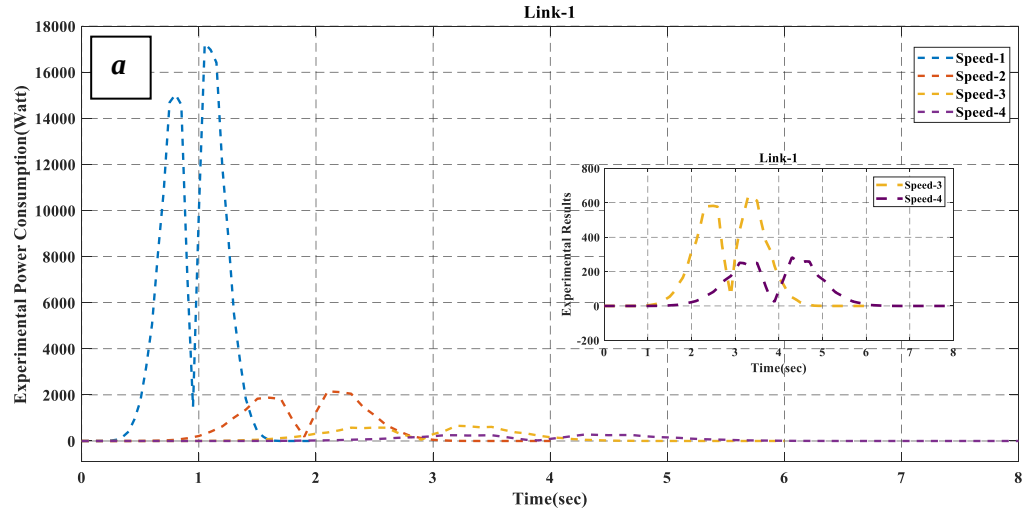


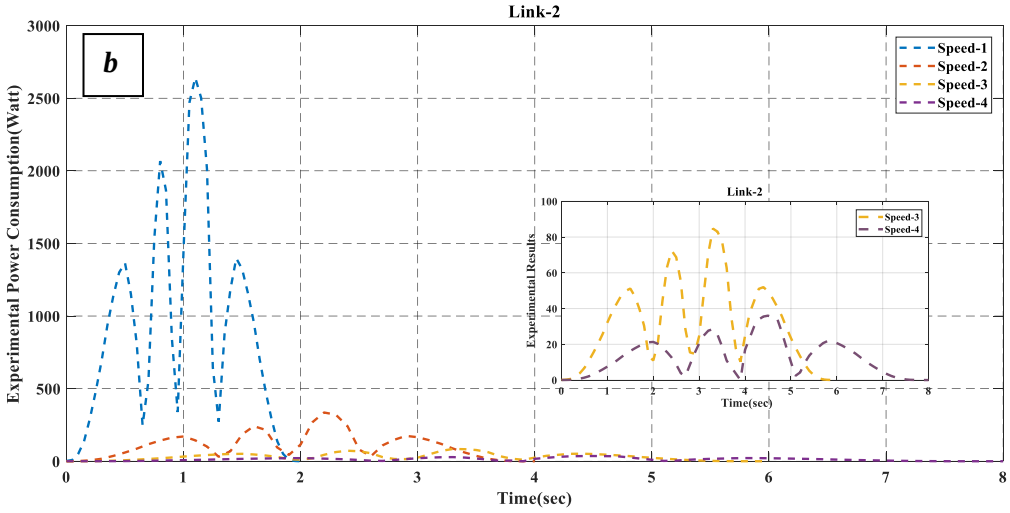
Figure(5.21) a,b,c,d Experimental Angular Velocity and acceleration curves of Link -1-2-3-4 at times(2,4,6,8) second

b-Power Consumption Profile

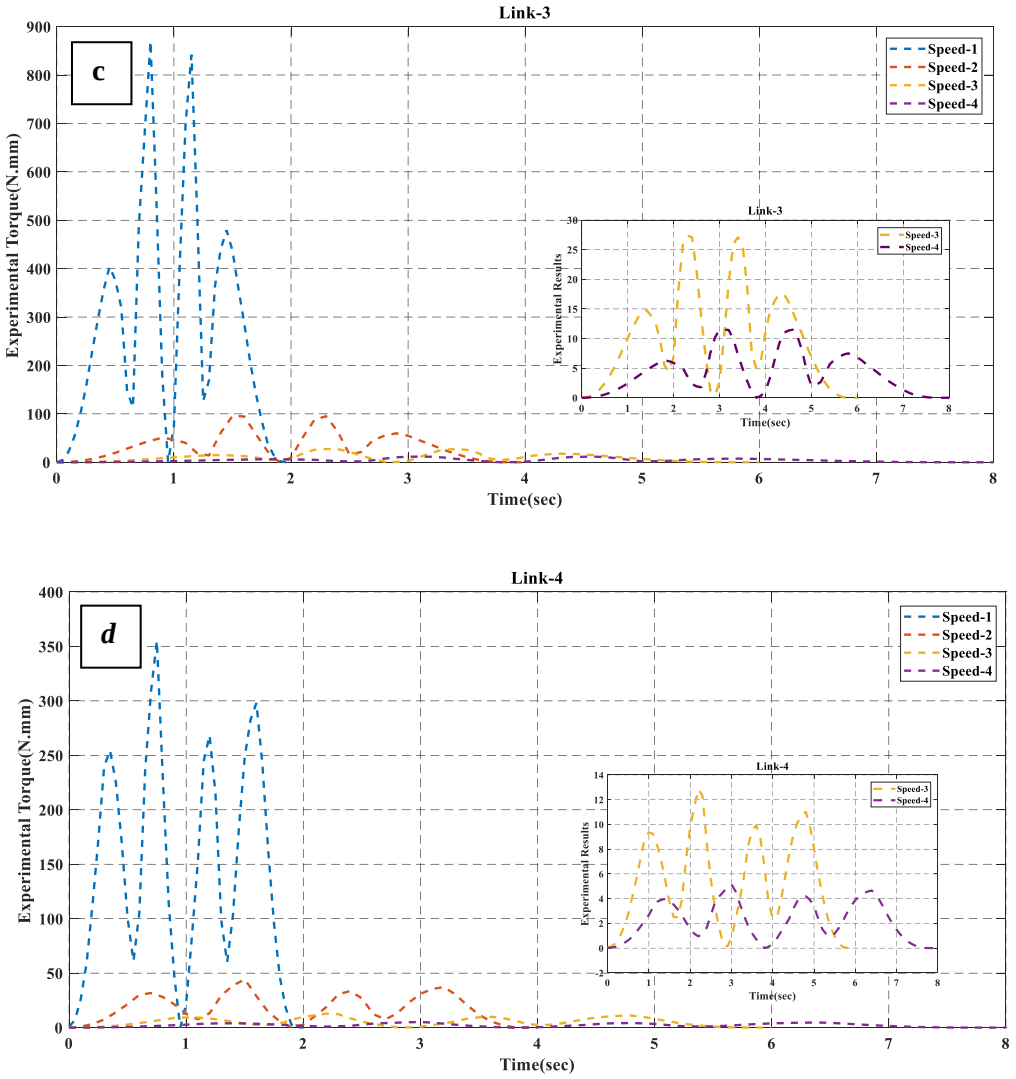
Figure(5.22) illustrates the power consumption profile for vertical path of Link-1-2-3-4 from LabVIEW program that displaced the robot (64 degree) at four time periods. It is obvious that the maximum value of the power consumption for link-1 reaches at 2-second which is the shortest periods having the value of (14029.371 Watt) then decreased at the other times periods as clear in **figure(5.22 a)**. As for Link-2, the value of

maximum power is less than Link-1that has the value of (2640.625 Watt) at time(2-second) as shown in **figure(5.22 b)**.





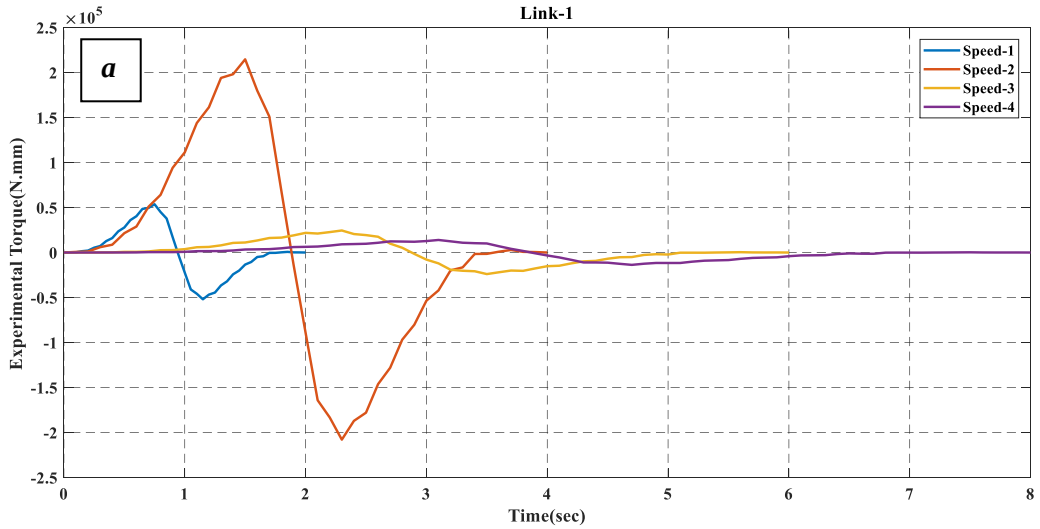
For **figure (5.42 c)**, the maximum power consumption reached the value of (885.190 Watt) for Link-3 at time 2-second, and then decreased at the other time periods. **Figure(5.42 d)** shows Link-4 as having maximum power consumption of (355.269 Watt) then decreased gradually to the end of the proposed periods.

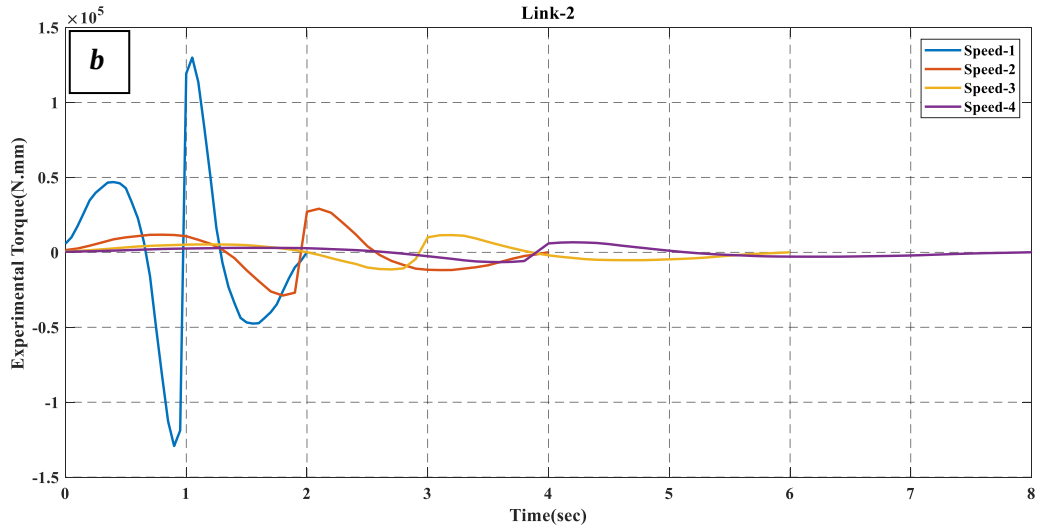


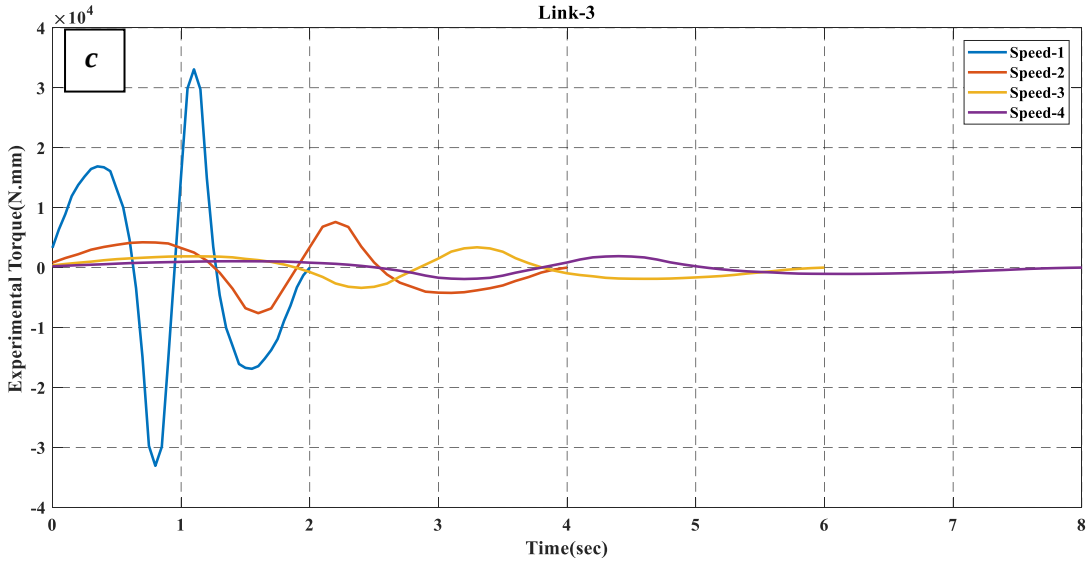
***Figure(5.22)a,b,c,d* Experimental Power consumption curves of Links 1-2-3-4 at times(2,4,6,8) seconds**

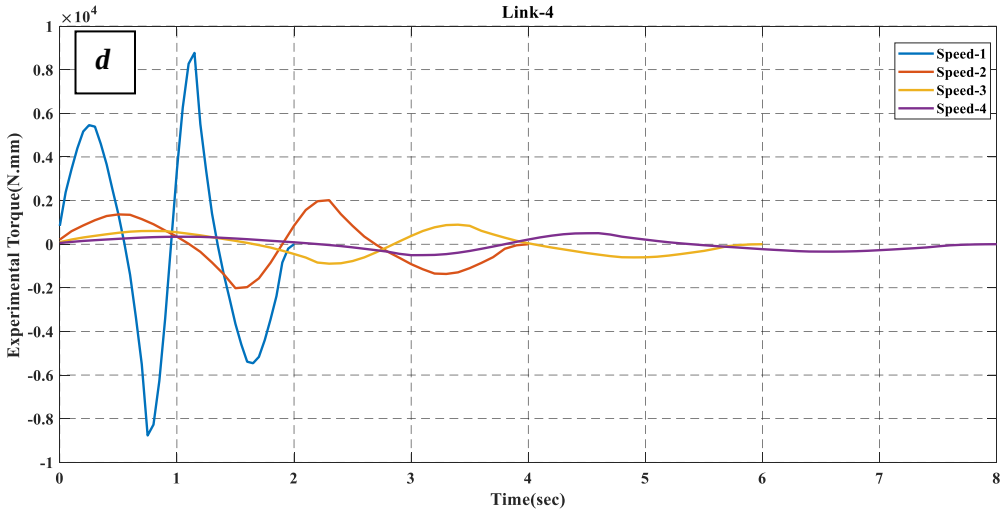
c-Torque Profile

Figure(5.23)a,b,c,d illustrate the torque profile for vertical path of Link-1-2-3-4 from LabVIEW program that displaced the robot (84 degree) at four time periods. From figures it was clear that Link-1 has the maximum values of torques (2.01 $*10^5$ N.mm)at the shortest time period while Link-4 has the minimum value of (8290.779 N.mm).


Link-1
a
×10^5
Experimental Torque(N.mm)
Time(sec)
Speed-1
Speed-2
Speed-3
Speed-4


Link-2
b
×10^5
Experimental Torque(N.mm)
Time(sec)
Speed-1
Speed-2
Speed-3
Speed-4


Link-3
c
×10^4
Experimental Torque(N.mm)
Time(sec)
Speed-1
Speed-2
Speed-3
Speed-4



Figure(5.23)a,b,c,d Experimental Torque curves of Links 1-2-3-4 at times(2,4,6,8) seconds.

5.5.2 Experiment 2:Horizontal Path Real Time Test

The second experiment aims to use the LabVIEW program to instruct the robot to follow the horizontal path. Because of the clearance in the mechanical parts such as the gearbox, as well as the friction between the parts, these factors were not taken into account in the theoretical part , this led to a small difference between the practical and theoretical results, which is shown in the comparison part.

5.5.2.1 Experiment Kinematics And Kinetics of Link-1-2-3-4 for Horizontal Curved Path

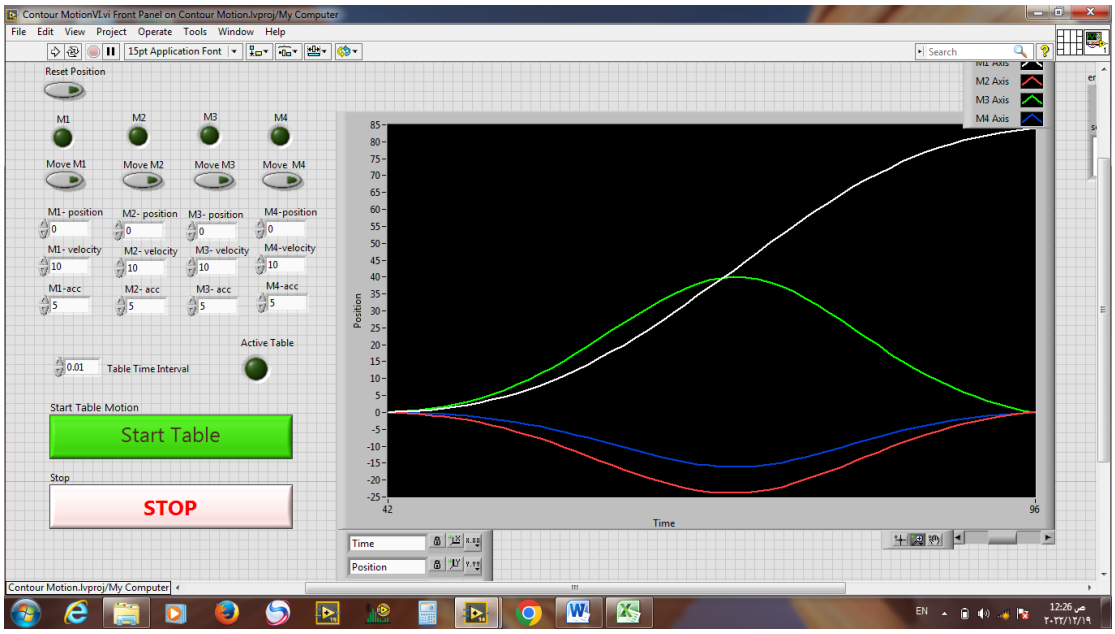
Figure (5.24) depicts the LabVIEW program window that displays the input angular displacement value received from simulation in order to extract the horizontal curved path for the (four Links). The following items illustrate and explain the LABVIEW window.

1- The white color line indicates the displacement of motor-1 in order to move link-1 along the path.

2- The red line represents the displacement of Motor-2 in order to move Link-2 from its starting position to its final destination.

3- The green color line indicates the displacement that Motor-3 is making in order to ensure that Link-3 follows the required path.

4- The blue color line described Motor-4 displacement to forces Follow the curving path, Link-4.



Figure(5.24) LabVIEW program window

Figure(5.25) shows the speed and acceleration profile for the 84-degree horizontal path of Links-1-2-3-4 from the LabVIEW program that moved the robot. Link 1 has a maximum angular velocity of (85.463 degrees per second) and an acceleration of (110.279 degrees per second squared). Link 2 has a maximum angular velocity of (45.698 degrees per second) and an acceleration of (112 degrees per second squared). This is clear in **figure (5.25) (a)(b).**

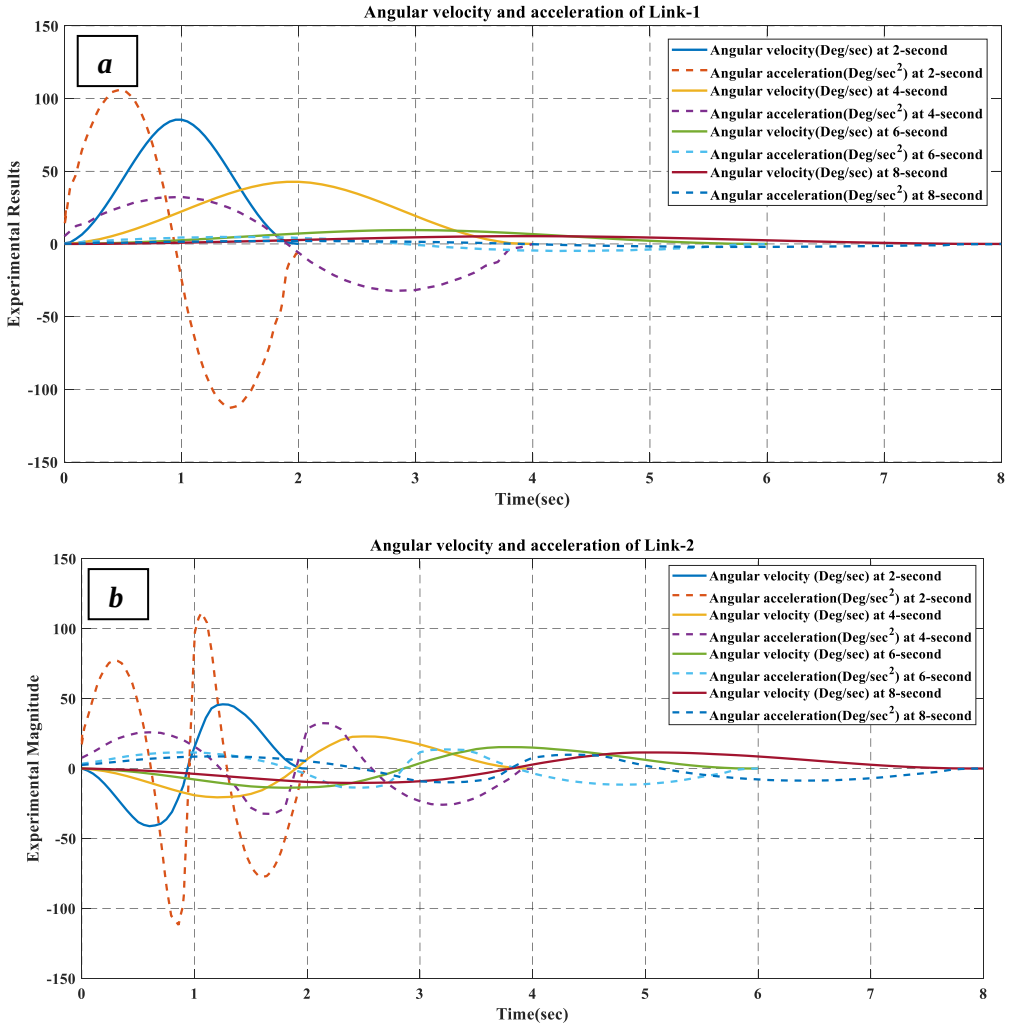
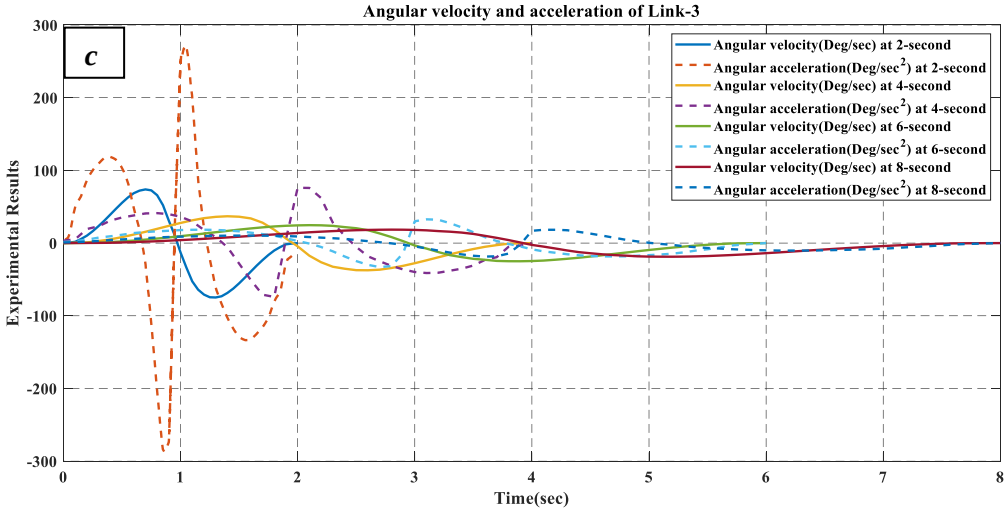
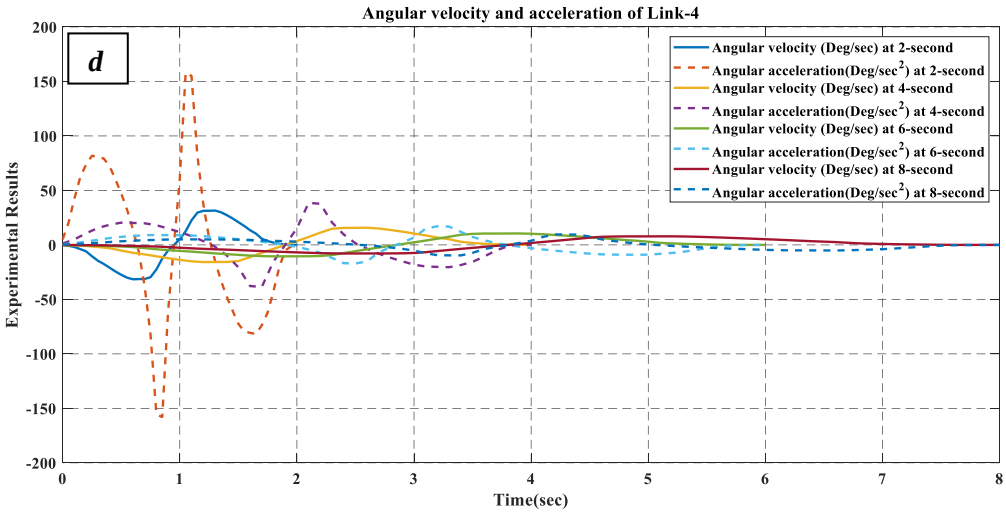


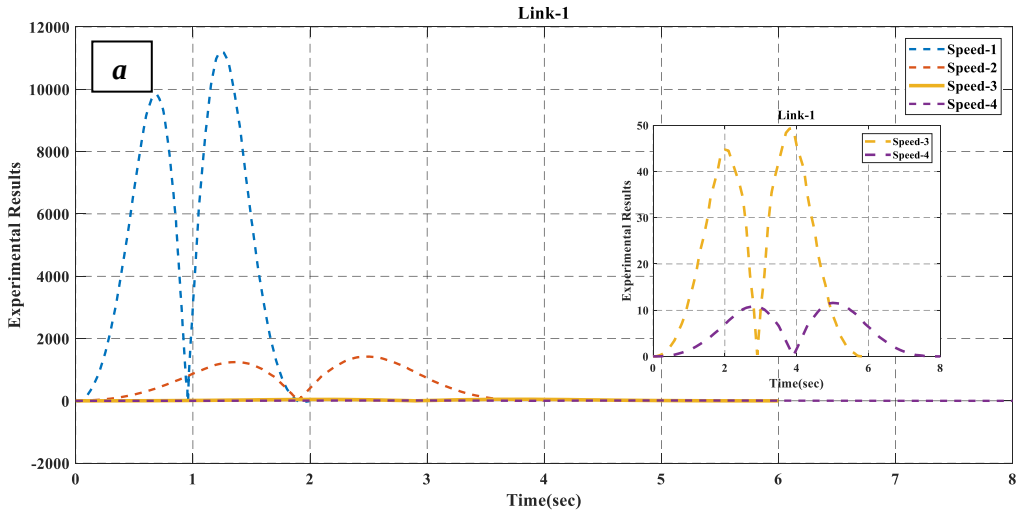
Figure (5.25) (c) and (d) show the angular velocity and acceleration curves for Link-3 and Link-4, with a maximum of (73.178 degrees per second) and (280.435 degrees per second squared) for Link-3 and (31.532 degrees per second) and (158.130 degrees per second squared) for Link-4.





Figure(5.25) a,b,c,d Experimental Angular Velocity and acceleration of Link -1-2-3-4 at (2,4,6,8) times periods.

The power consumption profile for the vertical path of Link-1-2-3-4 from the LabVIEW software that shifted the robot (84 degrees) at four time periods is shown in **figure(5.26).** Power consumption peaks for Link-1 at 2 seconds, the shortest time period, with a value of (10103.812 Watt) **figure(5.26)a,** and subsequently decreases at the remaining time intervals. Maximum power on Link-2 is smaller than that of Link-1 at the same time (2 seconds), as shown in **figure(5.26 b),** where the value of maximum power on Link-1 is (3016.723 watts).



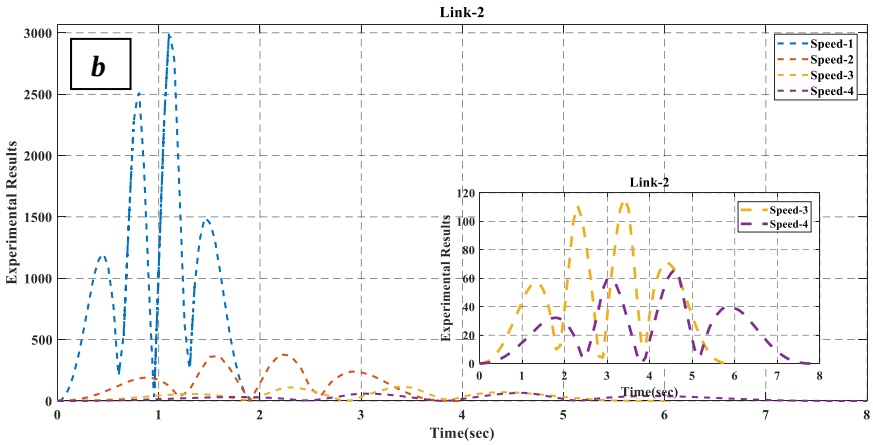
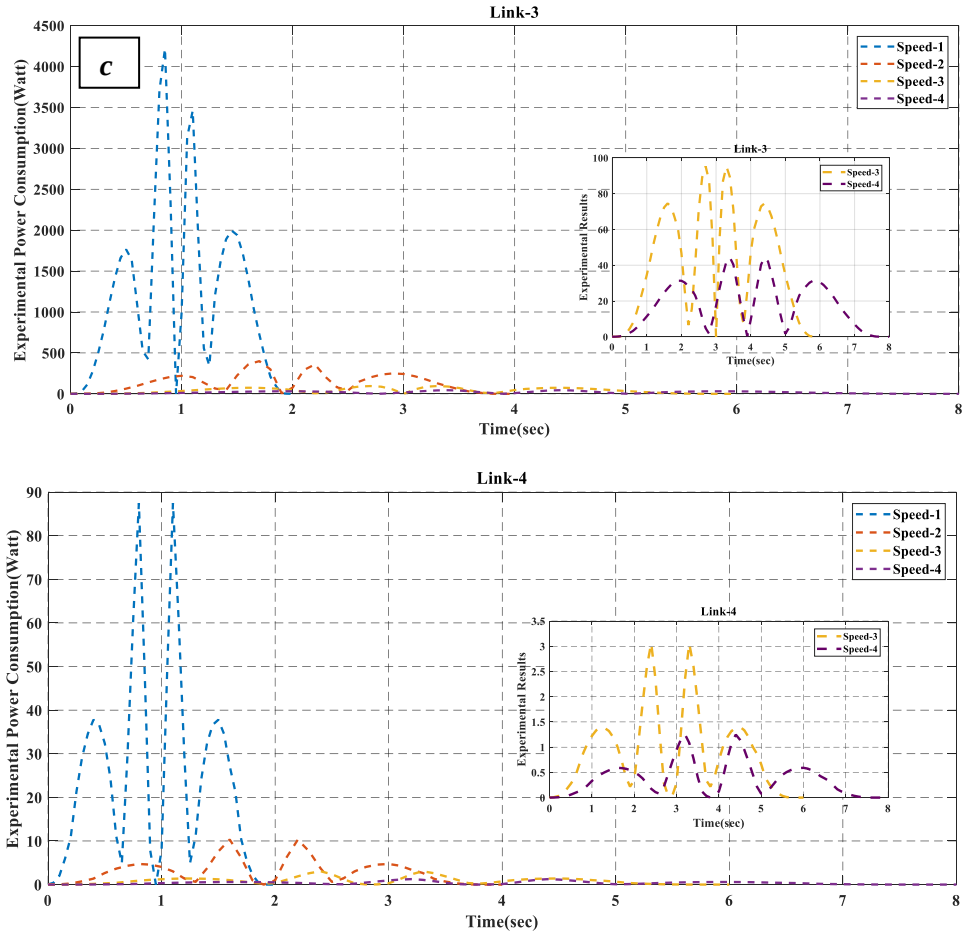
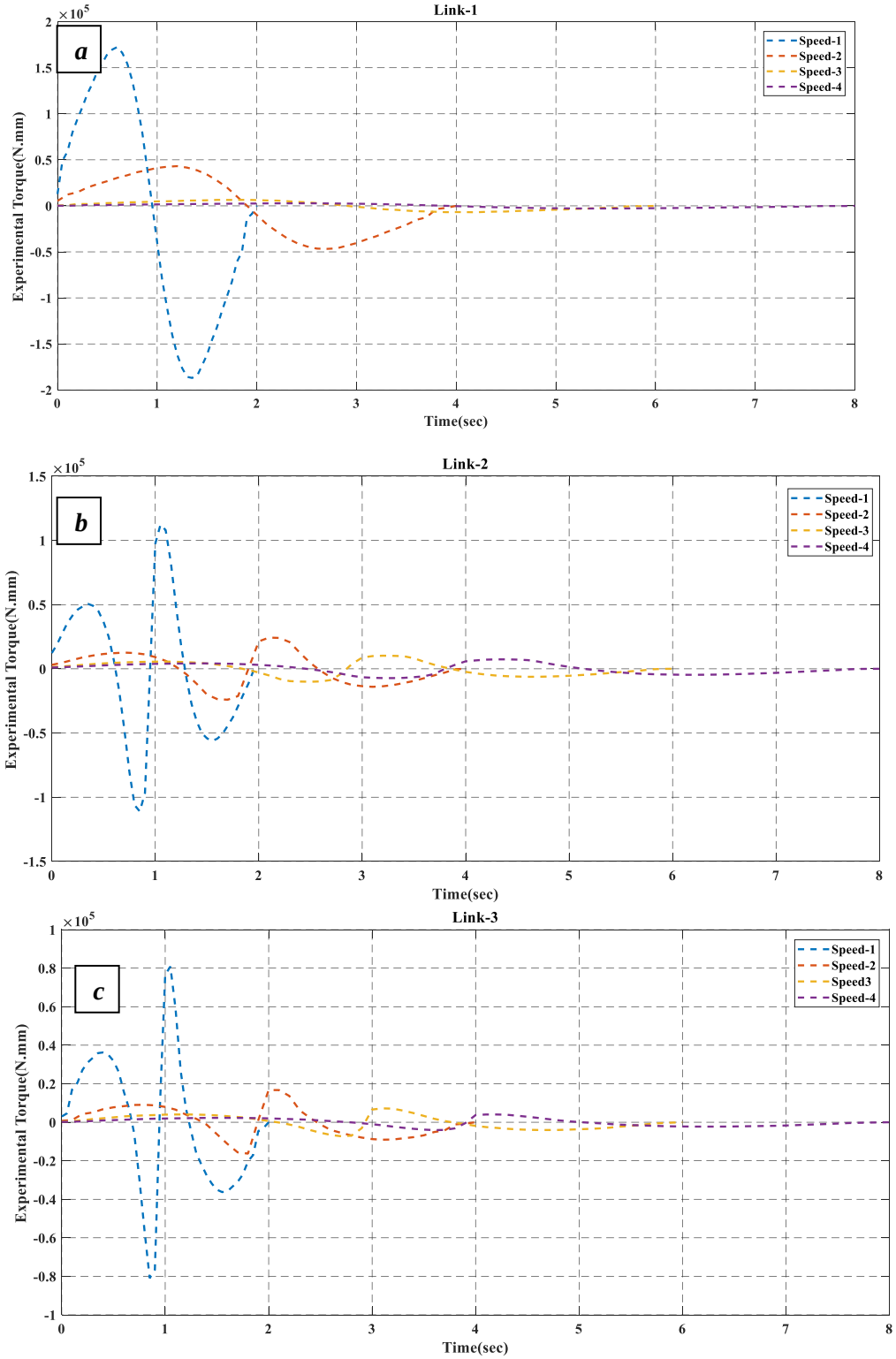


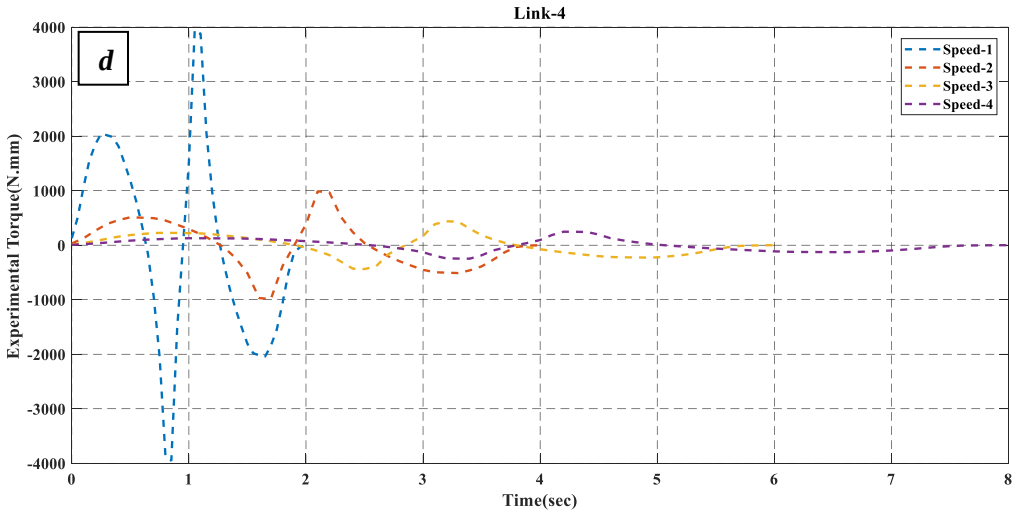
Figure (5.26 c) shows the maximum power consumption that reached the value of (3160.039 Watt) for Link-3 at time 2-second then decreased at the other time periods. In **Figure(5.26 d)**, Link-4 has a maximum power consumption of (81.442 Watt) then decreased gradually to the end of the proposed periods.



Figure(5.26) a,b,c,d Experimental Power consumption curves of Links 1-2-3-4 at times(2,4,6,8) seconds

Figure(5.27) illustrates the torque profile for vertical path of Link-1-2-3-4 from LabVIEW program that displaced the robot (84 degree) at four time periods. From the figures, it was clear that Link-1 has the maximum values of torques (1.51 *10^5 N.mm)at the shortest time period while Link-4 having the minimum value of (3853.250 N.mm).





Figure(5.27) a,b,c,d Experimental Torque curves of Links 1-2-3-4 at times(2,4,6,8) seconds.

5.6 Comparison Between Theoretical and experimental Analysis.

In order to validate and analyze the accuracy of the robot model a comparison between simulation and experimental data was used. A measure of the estimated difference between the observed or calculated value of a parameters and its true value its known as (error). The deviation that occurs in the kinematics and kinetics parameters for the vertical and horizontal path at four times periods will be discussed below. By observing the results of comparison we concluded that the horizontal path is the optimum path that the manipulator to track it, especially at four and six seconds because of the percentage error at the horizontal path is less than that for the vertical path. Percentage error was found by the formula in equation (5.1)[53]:

$$\text{Percentage error\%} = \frac{|\text{Approximate Value} - \text{Exact Value}|}{|\text{Exact Value}|} \times 100\% \quad \ldots(5.1)$$

5.5.1 Vertical Path

Comparison between the experimental and theoretical results and the percentage error for all the proposed time intervals for vertical path will be illustrated in **Table(5.2) (5.3) (5.4) (5.5)**, where they demonstrated the deviation that occur between the experimental and theoretical results for Link one, two, three and four at the proposed time periods. It is obvious that the percentage error at time two seconds is more than the other time intervals, also the difference between the actual and theoretical results is less and more stable at time four seconds and six seconds which considered as the optimal periods to accomplish the task .

Table(5.2) Comparison between actual and theoretical results for Link-1

Link No.	Parameter		٢ (sec.)	٤ (sec.)	٦ (sec.)	٨ (sec.)
Link-1	Velocity Deg./sec)(	Theoretical	109.386	55.193	36.701	27.896
		Experimental	111.236	56.368	37.152	28.315
		Percentage error	2.539%	2.084%	1.215%	1.478%
	Acceleration Deg./sec^2)(	Theoretical	٢١٥,٢٥٦	٥٦,٤٣١	٢٧,٦٩٤	١٦,٩٦٦
		Experimental	٢٢٢,١٨٥	٥٧,٧٩٦	٢٨,٢٤٢	١٧,٢٢٣
		Percentage error	3.118%	2.360%	١,٩٣٨%	%١,٤٩٣
	Torque (N.mm)	Theoretical	1.91E+05	4.85E+04	2.20E+04	1.29E+04
		Experimental	2.01E+05	5.03E+04	2.15E+04	1.31E+04
		Percentage error	4.975%	3.578%	2.325%	1.526%
	Power consumption (Watt)	Theoretical	13089.762	2035.969	629.737	274.329
		Experimental	14029.371	2154.302	660.995	282.838
		Percentage error	6.697%	5.492%	4.728	3.008%

Table(5.3) Comparison between actual and theoretical results for Link-2

Link No.	Parameter		٢ (sec.)	٤ (sec.)	٦ (sec.)	٨ (sec.)
Link-2	Velocity Deg./sec)(	Theoretical	37.408	18.554	12.318	9.247
		Experimental	36.308	18.154	12.073	9.077
		Percentage	3.029%	2.203%	2.022%	1.872%

		error				
	Acceleration Deg./sec^2)(	Theoretical	140.097	٣٧,٦٤٦	١٦,١٢٥	٨,٩٦١
		Experimental	146.087	٣٩,١٤٦	١٥,٧٥٩	٩,١٣٦
		Percentage error	4.1002%	3.831%	2.272%	1.915%
	Torque (N.mm)	Theoretical	9.84E+04	2.79E+04	1.08E+04	5.99E+03
		Experimental	1.03E+05	2.90E+04	1.11E+04	6.13E+03
		Percentage error	4.466%	3.793%	2.702%	2.283%
	Power consumption (Watt)	Theoretical	2472.910	321.963	81.586	34.999
		Experimental	2640.625	335.733	84.720	36.014
		Percentage error	6.351%	4.101%	3.698%	2.818%

Table(5.4) Comparison between actual and theoretical results for Link-3

Link No.	Parameter		٢ (sec.)	٤ (sec.)	٦ (sec.)	٨ (sec.)
Link-3	Velocity (Deg./sec)	Theoretical	33.854	16.927	11.733	8.637
		Experimental	32.954	17.327	11.482	8.467
		Percentage error	2.731%	2.363%	2.186%	2.008%
	Acceleration (Deg./sec^2)	Theoretical	130.351	٣٢,٥٨٨	١٤,٨٤١	٨,٤٧٩
		Experimental	135.350	٣٣,٥٨٧	١٥,١٨٧	٨,٦٥٥
		Percentage error	3.693%	2.974%	2.279%	٢,٠٣٣%
	Torque (N.mm)	Theoretical	29677.021	7369.258	3290.094	1904.217
		Experimental	31077.029	7609.257	3385.274	1855.490
		Percentage error	4.504%	3.154%	2.811%	2.626%
	Power consumption (Watt)	Theoretical	827.302	89.982	26.654	11.275
		Experimental	885.190	94.148	27.3520	11.552
		Percentage error	6.539%	4.424%	2.550%	2.397%

Table(5.5) Comparison between actual and theoretical results for Link-4

Link No.	Parameter		٢ (sec.)	٤ (sec.)	٦ (sec.)	٨ (sec.)
Link-4	Velocity (Deg./sec)	Theoretical	74.152	37.476	25.015	18.789
		Experimental	75.952	38.276	25.545	18.989
		Percentage	2.369%	2.090%	2.074%	1.053%

		error				
	Acceleration (Deg./sec^2)	Theoretical	291.635	٧٨,٩٥٨	٣٢,٨٧٨	١٨,٨٠١
		Experimental	299.835	٨٠,٩٠٨	٣٣,٧٦٢	١٩,١٧٩
		Percentage error	2.734%	2.410%	2.618%	١,٩٦٩%
	Torque (N.mm)	Theoretical	7995.879	1997.969	871.465	462.288
		Experimental	8290.779	2069.719	894.066	504.480
		Percentage error	3.556%	3.466%	2.527%	2.527%
	Power consumption (Watt)	Theoretical	332.660	41.832	11.940	4.929
		Experimental	355.269	44.408	12.568	5.151
		Percentage error	6.364%	5.801%	4.995%	4.311%

-Horizontal path

The comparison of practical and theoretical findings, as well as the percentage error for all specified horizontal path time intervals was demonstrated bellow. The percentage error at time two seconds is clearly greater than at the other time intervals, and the gap between the actual and theoretical findings is less and more steady at time four and six seconds. The percentage inaccuracy and comparison between experimental and theoretical findings for all proposed vertical path time intervals are shown below. **Tables (5.6) (5.7) (5.8) (5.9)** show the differences in experimental and theoretical findings for Links one, two, three, and four for the specified time periods. It is clear that the percentage deviation at time two seconds is more than at the other time intervals; Additionally, the discrepancy between the real and theoretical results is less and more steady at time four seconds and six seconds, which are regarded ideal times for completing the work.

Table(5.6) Comparison between actual and theoretical for Link-1

Link No.	Parameter		٢	٤	٦	٨
Link-١	Velocity	Theoretical	84.758	42.731	9.529	5.367
		Experimental	85.463	42.429	9.571	5.350
		Percentage error	0.824%	0.707%	0.439%	0.319%

	Acceleration	Theoretical	108.780	٢٩,٨٢٦	٤,٣٢٢	١,٩٨٢
		Experimental	110.279	٣٠,١٧٣	٤,٣٦٤	٢,٠٠١٧
		Percentage error	1.359%	1.149%	0.949%	0.983%
	Torque	Theoretical	1.48E+05	3.81E+04	5.97E+03	2.57E+03
		Experimental	1.51E+05	3.88E+04	6.05E+03	2.60E+03
		Percentage error	1.986%	1.804%	1.322%	1.153%
	Power consumption	Theoretical	9932.417	1290.915	46.121	11.112
		Experimental	10103.812	1306.851	45.602	11.239
		Percentage error	1.696%	1.219%	1.137%	1.129

Table(5.7) Comparison between actual and theoretical for Link-2

Link No.	Parameter		٢	٤	٦	٨
Link-٢	Velocity	Theoretical	44.991	21.834	14.951	11.314
		Experimental	45.698	22.144	15.124	11.437
		Percentage error	1.547%	1.4015%	1.144%	1.0787%
	Acceleration	Theoretical	110.472	٢٧,٩٦٨	١٠,٦٢٧	٨,٢٢٧
		Experimental	112	٢٨,٣	١٠,٥٠٧	٨,٢٧٤
		Percentage error	1.363%	1.172%	1.137%	0.568%
	Torque	Theoretical	4.81E+04	1.24E+04	5.36E+03	4.05E+03
		Experimental	4.88E+04	1.22E+04	5.42E+03	4.09E+03
		Percentage error	1.434%	1.639%	1.107%	0.977%
	Power consumption	Theoretical	2960.151	351.845	102.618	61.570
		Experimental	3016.723	346.268	104.286	62.479
		Percentage error	1.875%	1.610%	1.599%	1.453%

Table(5.8) Comparison between actual and theoretical for Link-3

Link No.	Parameter		٢	٤	٦	٨
Link-3	Velocity	Theoretical	٧٣,١٧٨	٣٧,٢١٠	٢٤,٧٩	١٨,٧٤٢
		Experimental	٧٣,٩٢١	٣٦,٩١٠	٢٤,٦٦	١٨,٥٩٥
		Percentage error	١,٠٠٤%	٠,٨١٢٧%	0.527%	٠,٧٩٣%
	Acceleration	Theoretical	٢٧٨,٣٠٦	٦٣,٣٣٣	٣٠,١١٠	١٧,٤٣٥

		Experimental	٢٨٠,٤٣٥	٦٢,٩٨٠	٣٠,٠٣	١٧,٤١١
		Percentage error	٠,٧٥٩%	٠,٥٦٠%	٠,٢٦٧%	٠,١٤٠%
	Torque	Theoretical	٥,٦٣E+04	١٥٨٠٦,٨٥	٦٣٦٥,٦	٣٧٨٧,٠٠٤
		Experimental	٥,٧١E+04	١٥٩٢٢,٩٦	٦٤٠١,٦	٣٧٧٦,٢١٢
		Percentage error	١,٤٠١%	٠,٧٢٩%	٠,٥٦٢%	٠,٢٨٥%
	Power consumption	Theoretical	٣١١٣,٧٤٦	٣٩٤,٥٥١	٨٧,٩١٩	٤٢,٧٨٠١
		Experimental	٣١٦٠,٠٣٩	٤٠٠,٠٠٢	٨٩,١١٩	٤٣,٢٨٠١
		Percentage error	1.464%	١,٣٦٢٧%	١,٣٤٦%	١,١٥٥%

Table(5.9) Comparison between actual and theoretical for Link-4

Link No.	Parameter		٢	٤	٦	٨
Link-٤	Velocity	Theoretical	٣١,٣٢١	١٥,٧٦٦	١٠,٤٧٦	٧,٨٥٩
		Experimental	٣١,٥٣٢	١٥,٦٦٠	١٠,٥١١	٧,٨٨٣
		Percentage error	0.667%	0.672%	0.328%	0.303%
	Acceleration	Theoretical	١٥٦,١٣٠	٣٧,٩٣٢	١٦,٦١٤	٩,٣٨٤
		Experimental	١٥٨,١٣٠	٣٨,٣٣٢	١٦,٧٨٨	٩,٤٤٣
		Percentage error	1.264%	1.0435%	1.033%	0.629%
	Torque	Theoretical	3753.2509	٩٠٦,٧٥٧	٤٠٥,٠٤٣	٢٣٤,٦٠١
		Experimental	3853.250	٩٢٠,٢٥٧	٤١٠,٣٠٢	٢٣٧,١٠٢
		Percentage error	1.830%	1.466%	1.281%	1.054%
	Power consumption	Theoretical	٧٩,٩٨٨	٩,٩٩١٠	٢,٩٩١٨	٠,٥٥٣
		Experimental	٨١,٤٤٢	١٠,١٠٢	٣,٠٢١٢	٠,٥٤٩
		Percentage error	1.785%	1.106%	0.973%	0.659%

Chapter Six

Conclusion and Suggestions for Future Work

6.1 Conclusion

1- From the current study, it has been observed that the results obtained from time(2s) are incompatible with the behavior of the robot as in times (4, 6, and 8) seconds as illustrated in many figures of theoretical and experimental study and for all the parameters used (speed, acceleration, torque, and power consumption) so we advise not to adopt it when studying the behavior of these parameters.

2- The dynamic behavior of the studied parameters in this research corresponds to time (4, 6, and 8) seconds and as illustrated and evident in all the figures in the chapter five gives support that the greater the time the accuracy of the results and the lower the error ratio.

3- When completing the horizontal path by the model, it was found that the energy consumption and torque needed to complete the task are much less than the energy, consumption, and torque required to complete the vertical path, as the robot’s arms in the case of the horizontal path.

4- The first and fourth motors have angular acceleration and a large angular velocity compared to the second and third motors, especially at the first speed when the time (2-seconds).

5- The value of the torque required to complete the vertical and horizontal paths reached its highest level at the first speed, represented by the time of 2 seconds, as the robot at this time is unstable. The first motor also consumed more torque than the rest of the three motors as a result of bearing the weights of the other three arms.

6- The percentage of deviation between theoretical and practical increased with the increase in the speed of the robot, especially at the time of 2 seconds. It was also noted that the percentage of deviation for the vertical path was higher than its counterpart for the horizontal path.

7- The moment of inertia of the first motor is the highest compared to the remaining three motors, and the value of the moment of inertia was equal for the arm at the four suggested times.

6.2 Suggestions for Future Work

1- A more complex helical path can be created rather than a circular path.

2- The work area can be increased when the robot is working.

3- Placing obstacles during path execution.

4- Considering payload and elastic constraints as external dynamic loads.

5- Using more than four links to completion the same vertical and horizontal path such us five or seven links.

[1] Niku, S. B., (2011),An Introduction to Robotics: analysis, control, applications, Second Edition, Wiley.

[2] Ashagrie, A., et al. (2021). Modeling and control of a 3-DOF articulated robotic manipulator using self-tuning fuzzy sliding mode controller. Cogent Engineering, Vol.8, No(1). pp.1-33.https://doi.org/10.1080/23311916.2021.1950105.

[3] Verl, A., Valente, et al. (2019). Manufacturing Technology Robots in machining. CIRP Annals - Manufacturing Technology.Vol. 68, No. (2), pp. 799-822. ISSN 0007-8506.

[4] Macfarlane, S., & Croft, E. A. (2003). Jerk-Bounded Manipulator Trajectory Planning : Design for Real-Time Applications. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION .Vol.19, No(1), pp.42–52.

[5] Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). Robot modeling and control. *IEEE Control Systems, Vol.26*, No(6), pp.113–115. https://doi.org/10.1109/MCS.2006.252815.

[6] Ghafil, H. N. (2020). *Optimization for Robot Modelling with MATLAB*.Springer.

[7] Mikael Johnsson (2002). Visual Programming. Master Thesis Department of Computer Science at Mälardalen University, Västerås, Sweden.pp.1-60.

[8] Gasparetto, A., et al. (2012). Trajectory Planning in Robotics. Mathematics in Computer Science.pp.1-11. https://doi.org/10.1007/s11786-012-0123-8

[9] Ratheesh Rajan (2001). Foundation Studies for an Alternate

Approach to Motion Planning of Dynamic Systems. Master thesis at University of Texas at Austin. pp.1-141

[10] Gouasmi, M., Ouali, M., Fernini, B., & Meghatria, M. (2012). Kinematic modelling and simulation of a 2-R robot using solidworks and verification by matlab/simulink. International Journal of Advanced Robotic Systems, *9*, pp.1–13. https://doi.org/10.5772/50203

[11] Sam, R., Arrifin, K., & Buniyamin, N. (2012). Simulation of Pick and Place Robotics System Using Solidworks Softmotion. International Conference on System Engineering and Technology *(ICSET).* pp. 1-6, doi: 10.1109/ICSEngT.2012.6339325..

[12] Iqbal, J., Islam, R., & Khan, H. (2012). Modeling and Analysis of a 6 DOF Robotic Arm Manipulator. Canadian Journal on Electrical and Electronics Engineering. Vol. 3,pp.300-306.

[13] Chaudhary, H. (2012). Trajectory Tracking Control Of Scorbot-ER V Plus Robot Manipulator Based On Kinematical Approach. International Journal of Engineering Science and Technology (IJEST). Vol. 4 No.03. pp.1174–1182.

[14] Patel, Y. D., & George, P. M. (2013). Performance Measurement And Dynamic Analysis Of Two DOF Robotic Arm Manipulator. International Journal of Research in Engineering and Technology. Volume: 02 Issue: 09.pp.77–84.

[15] Nabeel, H. M., Azher, A., Ali, S. M. U., & Wahab, A. (2013). Designing , Fabrication and Controlling Of Multipurpose3-DOF Robotic Arm Designing , Fabrication and Controlling Of

Multipurpose3- DOF Robotic Arm .IOP Conf.Series:MaterialsScienceandEngineering.pp.1-6. https://doi.org/10.1088/1757-899X/51/1/012023

[16] Mohammed, A., Schmidt, B., Wang, L., & Gao, L. (2014). Minimizing Energy Consumption for Robot Arm Movement. Procedia CIRP,vol. 25,pp. 400–405. https://doi.org/10.1016/j.procir.2014.10.055

[17] Hpande, V. D. E. S., & George, P. M. (2014). Kinematic Modelling And Analysis Of 5 DOF Robotic Arm. International Journal of Robotics Research and Development (IJRRD).Vol.4(2),pp. 17–24.

[18] Liu, X., & Xie, Z. (2015). Kinematic analysis and simulation for a kind of Planar Articulated robot based on ADAMS. Joint International Mechanical, Electronic and Information Technology Conference (JIMET) pp.1054–1059.

[19] Mohammed, A. A., & Sunar, M. (2015). Kinematics Modeling of a 4-DOF Robotic Arm. International Conference on Control, Automation and Robotics.pp.87–91.

[20] Singh, E. H., Dhillon, N., & Ansari, E. I. (2015). Forward and inverse Kinematics Solution for Six DOF with the help of Robotics tool box in matlab. International Journal of Application or Innovation in Engineering & Management (IJAIEM) .vol. 4(3), Issue 3, pp.17–22.

[21] Hussain, S. B. (2016). Design of a 3 DoF Robotic Arm. Sixth International Conference on Innovative Computing Technology.pp.145–149.

[22] Kasera, S., Kumar, A., & Prasad, L. B. (2017). Trajectory Tracking of 3-DOF Industrial Robot Manipulator by Sliding Mode Control. International Conference on Electrical, Computer and Electronics (UPCON).pp.364–369.

[23] Kütük, M. E., Daş, M. T., & Dülger, L. C. (2017). Forward and Inverse Kinematics Analysis of Denso Robot. Proceedings of the International Symposium of Mechanism and Machine Science .pp.71-78.

[24] Serrezuela, R. R., Fernando, A., Chavarro, C., Angel, M., Cardozo, T., Toquica, A. L., Fernando, L., & Martinez, O. (2017). Kinematic Modelling Of A Robotic Arm Manipulator Using MATLAB. ARPN Journal of Engineering and Applied Sciences .Vol.12, No(7), pp.2037–2045.

[25] Tarun P. et al. (2017).Forward and Inverse Kinematic Analysis of Robotic Manipulators. International Research Journal of Engineering and Technology (IRJET).Vol.04,Issue(02),pp.1459-1469.

[26] Xiao, J., Han, W., & Wang, A. (2017). Simulation Research of A Six Degrees of Freedom Manipulator Kinematics based On MATLAB Toolbox. International Conference on Advanced Mechatronic Systems. pp.376–380.

[27] Aktaş, K. G., Pehlivan, F., & Esen, İ. (2017). Kinematic Analysis of 4 Degrees of Freedom Robotic Arm and Simultaneous Trajectory Tracking Using ADAMS ® -MATLAB ® Software . INTERNATIONAL SCIENTIFIC AND VOCATIONAL STUDIES CONGRESS (BILMES), pp.4–9.

[28] Souza, L. B. De, Dalmedico,et al., K. (2018). Inverse Kinematics and Trajectory Planning Analysis of a Robotic Manipulator. International Journal of Advanced Engineering Research and Science (IJAERS). Vol.5, Issue.4, ISSN: 2349-6495,pp.207–214.

[29] Farman, M., Al-shaibah, M., Aoraiath, Z., & Jarrar, F. (2018). Design of a Three Degrees of Freedom Robotic Arm. International Journal of Computer Applications. Vol. 179 , No.37, pp.12–17.

[30] Benotsmane, R., Kacemi, S., & Benachenhou, M. R. (2018). Calculation methodology for trajectory planning of a 6 axis manipulator arm. ANNALS of Faculty Engineering Hunedoara International Journal of Engineering, Vol.3, pp.27-32.

[31] Derelİ, S., & Köker, R. (2018). IW-PSO Approach To The Inverse Kinematics Solution Of A 7-DOF Serial Robot Manipulator. Sigma Journal of Engineering and Natural Sciences .Vol.36(1), pp.77–85.

[32] Consolini, L., et al.(2019), "Optimal Time-Complexity Speed Planning for Robot Manipulators," in IEEE Transactions on Robotics, vol. 35, no. 3, pp. 790-797, doi: 10.1109/TRO.2019.2899212.

[333] Ali, F. (2020). The Kinematic Analysis Of Four Degrees Of Freedom For A Medical Robot And Control It By Labview And Arduino. Journal of Advances in Computer Engineering and Technology (JACET) .Vol.6, Issue(1), pp.33–38.

[34] Afrisal, H., Setiyono, B., et al. (2020). Trajectory Planning with Obstacle Avoidance of 3 DoF Robotic Arm for Test Tube Handling System. Int. Conf. on Information Tech., Computer, and Electrical Engineering (ICITACEE),pp.41–46.

[35] Benotsmane, R., et al. (2020). Simulation And Trajectory Optimization Of Collaborating Robots By Application Of SOLIDWORKS And MATLAB Software In Industry. Academic Journal Of Manufacturing Engineering .Vol.18, Issue(4), pp.191–197.

[36] Kim, J., Jin, M., Park, et al. (2020). Task Space Trajectory Planning for Robot Manipulators to Follow 3-D Curved Contours. Electronics Journal, Vol.9,Issue 9,No. 9, pp.1-20.

[37] Cesar, J., Muñoz, C., Yahaira, H.et al. (2021). Design of Control Interface for a SCARA Manipulator with Subactuated Final Effector. Científica journal.Vol. 25,pp. 71-82.

[38] Zhang, X., Kou, et al. (2021). Trajectory Planning and Control of Handling Robot Manipulator Trajectory Planning and Control of Handling Robot Manipulator. Journal of Physics: Conference Series,pp.1-11.https://doi.org/10.1088/1742-6596/1802/2/022067.

[39] Salman, H. D., Hamzah, et al. (2021). Kinematic Analysis And Implementation Of Three Degrees Of Freedom Robotic Arm. The Iraqi Journal For Mechanical And Material Engineering. Vo.2l, No.2.pp.118-129.

[40] Ohri, J. (2020). Trajectory Tracking in a 3-DOF Robotic Manipulator using Sliding Mode Controller, IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA) ,pp.9–14.

[41] Cesar J. et al.(2021). Design of Control Interface for a SCARA Manipulator with Subactuated Final Effector. Científica. Vol.25, pp.71-82.

[42] Yu, X., Dong, M., & Yin, W. (2022). Time-optimal trajectory planning of manipulator with simultaneously searching the optimal path. Computer Communications, Vol.181, pp.446–453.

https://doi.org/10.1016/j.com..10.005.

[٤3] Siciliano B. et al. (2016), "Springer Handbook of Robotics", Springer.

[٤4] Jazar R. (2010), "Theory of Applied Robotics: Kinematics, Dynamics, and Control", second edition., Springer Science & Business Media.

[٤5] Craig J. (2005), "Introduction to Robotics: Mechanics and Control", third edition, USA.

[46] Shabeeb A. and Mohammed L. , (2015). "Inverse Kinematics Analysis Using Close Form Solution Method for 5 DOF Robot Manipulate", Eng.& Tech. J., Vol. 33, No. 9, pp. 2094–2106.

[47] Park J., Kim J. et al. , (2012) "An Iterative Algorithm for Inverse Kinematics of 5-DOF Manipulator with Offset Wrist", World Acad. Sci. Eng. Technol., Vol.6, No.12, pp.150–155.

[48] Siciliano B., Sciavicco L. ,et al. (2012) "Modelling and Control of Robot Manipulators", Springer Science & Business Media.

[49] Murray R. (2017) , "A mathematical introduction to robotic manipulation", CRC press.

[50] Huei- Huang Lee,(2014) , Engineering Dynamics Labs with SolidWorks Motion, SDC Publications.

[51] Singh R., R. Sugandhi,et al. (2018) "Development of digital control system in LabVIEW for stepper motor drives," International Conference

on Circuits and Systems in Digital Enterprise Technology (ICCSDET), Kottayam, India , pp. 1-6, doi: 10.1109/ICCSDET. 8821122.

[52] Fundamentals, Labview(2005). "National Instruments." Austin, Texas, USA.

[53] Green K. and Tashman L. (2009). Percentage Error: What Denominator?. Foresight: The International Journal of Applied Forecasting. Vol. 12. pp.36-40.

References

[1] Niku, S. B., (2011),An Introduction to Robotics: analysis, control, applications, Second Edition, Wiley.

[2] Ashagrie, A., et al. (2021). Modeling and control of a 3-DOF articulated robotic manipulator using self-tuning fuzzy sliding mode controller. Cogent Engineering, Vol.8, No(1). pp.1-33.https://doi.org/10.1080/23311916.2021.1950105.

[3] Verl, A., Valente, et al. (2019). Manufacturing Technology Robots in machining. CIRP Annals - Manufacturing Technology.Vol. 68, No. (2), pp. 799-822. ISSN 0007-8506.

[4] Macfarlane, S., & Croft, E. A. (2003). Jerk-Bounded Manipulator Trajectory Planning : Design for Real-Time Applications. IEEE Transcation On Robotics And Automation .Vol.19, No(1), pp.42–52.

[5] Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). Robot modeling and control. *IEEE Control Systems*, *Vol.26*, No(6), pp.113–115. https://doi.org/10.1109/MCS.2006.252815.

[6] Ghafil, H. N. (2020). *Optimization for Robot Modelling with MATLAB*.Springer.

[7] Mikael Johnsson (2002). Visual Programming. Master Thesis Department of Computer Science at Mälardalen University, Västerås, Sweden.pp.1-60.

[8] Gasparetto, A., et al. (2012). Trajectory Planning in Robotics. Mathematics in Computer Science.pp.1-11. https://doi.org/10.1007/s11786-012-0123-8

[9] Ratheesh Rajan (2001). Foundation Studies for an Alternate Approach to Motion Planning of Dynamic Systems. Master thesis at University of Texas at Austin. pp.1-141

[10] Gouasmi, M., Ouali, M., Fernini, B., & Meghatria, M. (2012). Kinematic modelling and simulation of a 2-R robot using solidworks and verification by matlab/simulink. International Journal of Advanced Robotic Systems, *9*, pp.1–13. https://doi.org/10.5772/50203

[11] Sam, R., Arrifin, K., & Buniyamin, N. (2012). Simulation of Pick and Place Robotics System Using Solidworks Softmotion. International Conference on System Engineering and Technology *(ICSET).* pp. 1-6, doi: 10.1109/ICSEngT.2012.6339325..

[12] Iqbal, J., Islam, R., & Khan, H. (2012). Modeling and Analysis of a 6 DOF Robotic Arm Manipulator. Canadian Journal on Electrical and Electronics Engineering. Vol. 3,pp.300-306.

[13] Chaudhary, H. (2012). Trajectory Tracking Control Of Scorbot-ER V Plus Robot Manipulator Based On Kinematical Approach. International Journal of Engineering Science and Technology (IJEST). Vol. 4 No.03. pp.1174–1182.

[14] Patel, Y. D., & George, P. M. (2013). Performance Measurement And Dynamic Analysis Of Two DOF Robotic Arm Manipulator. International Journal of Research in Engineering and Technology. Volume: 02 Issue: 09.pp.77–84.

[15] Nabeel, H. M., Azher, A., Ali, S. M. U., & Wahab, A. (2013). Designing , Fabrication and Controlling Of Multipurpose3-DOF Robotic Arm Designing , Fabrication and Controlling Of

Multipurpose3- DOF Robotic Arm .IOP Conf.Series:MaterialsScienceandEngineering.pp.1-6. https://doi.org/10.1088/1757-899X/51/1/012023

[16] Mohammed, A., Schmidt, B., Wang, L., & Gao, L. (2014). Minimizing Energy Consumption for Robot Arm Movement. Procedia CIRP,vol. 25,pp. 400–405. https://doi.org/10.1016/j.procir.2014.10.055

[17] Hpande, V. D. E. S., & George, P. M. (2014). Kinematic Modelling And Analysis Of 5 DOF Robotic Arm. International Journal of Robotics Research and Development (IJRRD).Vol.4(2),pp. 17–24.

[18] Liu, X., & Xie, Z. (2015). Kinematic analysis and simulation for a kind of Planar Articulated robot based on ADAMS. Joint International Mechanical, Electronic and Information Technology Conference (JIMET) pp.1054–1059.

[19] Mohammed, A. A., & Sunar, M. (2015). Kinematics Modeling of a 4-DOF Robotic Arm. International Conference on Control, Automation and Robotics.pp.87–91.

[20] Singh, E. H., Dhillon, N., & Ansari, E. I. (2015). Forward and inverse Kinematics Solution for Six DOF with the help of Robotics tool box in matlab. International Journal of Application or Innovation in Engineering & Management (IJAIEM) .vol. 4(3), Issue 3, pp.17–22.

[21] Hussain, S. B. (2016). Design of a 3 DoF Robotic Arm. Sixth International Conference on Innovative Computing Technology.pp.145–149.

[22] Kasera, S., Kumar, A., & Prasad, L. B. (2017). Trajectory Tracking of 3-DOF Industrial Robot Manipulator by Sliding Mode Control.

International Conference on Electrical, Computer and Electronics (UPCON).pp.364–369.

[23] Kütük, M. E., Daş, M. T., & Dülger, L. C. (2017). Forward and Inverse Kinematics Analysis of Denso Robot. Proceedings of the International Symposium of Mechanism and Machine Science .pp.71-78.

[24] Serrezuela, R. R., Fernando, A., Chavarro, C., Angel, M., Cardozo, T., Toquica, A. L., Fernando, L., & Martinez, O. (2017). Kinematic Modelling Of A Robotic Arm Manipulator Using MATLAB. ARPN Journal of Engineering and Applied Sciences .Vol.12, No(7), pp.2037–2045.

[25] Tarun P. et al. (2017).Forward and Inverse Kinematic Analysis of Robotic Manipulators. International Research Journal of Engineering and Technology (IRJET).Vol.04,Issue(02),pp.1459-1469.

[26] Xiao, J., Han, W., & Wang, A. (2017). Simulation Research of A Six Degrees of Freedom Manipulator Kinematics based On MATLAB Toolbox. International Conference on Advanced Mechatronic Systems. pp.376–380.

[27] Aktaş, K. G., Pehlivan, F., & Esen, İ. (2017). Kinematic Analysis of 4 Degrees of Freedom Robotic Arm and Simultaneous Trajectory Tracking Using ADAMS ® -MATLAB ® Software . International Scientific And Vocational Studies Congress (BILMES), pp.4–9.

[28] Souza, L. B. De, Dalmedico,et al., K. (2018). Inverse Kinematics and Trajectory Planning Analysis of a Robotic Manipulator. International Journal of Advanced Engineering Research and Science (IJAERS). Vol.5, Issue.4, ISSN: 2349-6495,pp.207–214.

[29] Farman, M., Al-shaibah, M., Aoraiath, Z., & Jarrar, F. (2018). Design of a Three Degrees of Freedom Robotic Arm. International Journal of Computer Applications. Vol. 179 , No.37, pp.12–17.

[30] Benotsmane, R., Kacemi, S., & Benachenhou, M. R. (2018). Calculation methodology for trajectory planning of a 6 axis manipulator arm. ANNALS of Faculty Engineering Hunedoara International Journal of Engineering, Vol.3, pp.27-32.

[31] Derelİ, S., & Köker, R. (2018). IW-PSO Approach To The Inverse Kinematics Solution Of A 7-DOF Serial Robot Manipulator. Sigma Journal of Engineering and Natural Sciences .Vol.36(1), pp.77–85.

[32] Consolini, L., et al.(2019), "Optimal Time-Complexity Speed Planning for Robot Manipulators," in IEEE Transactions on Robotics, vol. 35, no. 3, pp. 790-797, doi: 10.1109/TRO.2019.2899212.

[333] Ali, F. (2020). The Kinematic Analysis Of Four Degrees Of Freedom For A Medical Robot And Control It By Labview And Arduino. Journal of Advances in Computer Engineering and Technology (JACET) .Vol.6, Issue(1), pp.33–38.

[34] Afrisal, H., Setiyono, B., et al. (2020). Trajectory Planning with Obstacle Avoidance of 3 DoF Robotic Arm for Test Tube Handling System. Int. Conf. on Information Tech., Computer, and Electrical Engineering (ICITACEE),pp.41–46.

[35] Benotsmane, R., et al. (2020). Simulation And Trajectory Optimization Of Collaborating Robots By Application Of SOLIDWORKS And MATLAB Software In Industry. Academic Journal Of Manufacturing Engineering .Vol.18, Issue(4), pp.191–197.

[36] Kim, J., Jin, M., Park, et al. (2020). Task Space Trajectory Planning for Robot Manipulators to Follow 3-D Curved Contours. Electronics Journal, Vol.9,Issue 9,No. 9, pp.1-20.

[37] Cesar, J., Muñoz, C., Yahaira, H.et al. (2021). Design of Control Interface for a SCARA Manipulator with Subactuated Final Effector. Científica journal.Vol. 25,pp. 71-82.

[38] Zhang, X., Kou, et al. (2021). Trajectory Planning and Control of Handling Robot Manipulator Trajectory Planning and Control of Handling Robot Manipulator. Journal of Physics: Conference Series,pp.1-11.https://doi.org/10.1088/1742-6596/1802/2/022067.

[39] Salman, H. D., Hamzah, et al. (2021). Kinematic Analysis And Implementation Of Three Degrees Of Freedom Robotic Arm. The Iraqi Journal For Mechanical And Material Engineering. Vo.2l, No.2.pp.118-129.

[40] Ohri, J. (2020). Trajectory Tracking in a 3-DOF Robotic Manipulator using Sliding Mode Controller, IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA) ,pp.9–14.

[41] Cesar J. et al.(2021). Design of Control Interface for a SCARA Manipulator with Subactuated Final Effector. Científica. Vol.25, pp.71-82.

[42] Yu, X., Dong, M., & Yin, W. (2022). Time-optimal trajectory planning of manipulator with simultaneously searching the optimal path. Computer Communications, Vol.181, pp.446–453.

https://doi.org/10.1016/j.com..10.005.

[٤3] Siciliano B. et al. (2016), “Springer Handbook of Robotics”, Springer.

[٤4] Jazar R. (2010), “Theory of Applied Robotics: Kinematics, Dynamics, and Control”, second edition., Springer Science & Business Media.

[٤5] Craig J. (2005), “Introduction to Robotics: Mechanics and Control”, third edition, USA.

[46] Shabeeb A. and Mohammed L. , (2015). “Inverse Kinematics Analysis Using Close Form Solution Method for 5 DOF Robot Manipulate”, Eng.& Tech. J., Vol. 33, No. 9, pp. 2094–2106.

[47] Park J., Kim J. et al. , (2012) “An Iterative Algorithm for Inverse Kinematics of 5-DOF Manipulator with Offset Wrist”, World Acad. Sci. Eng. Technol., Vol.6, No.12, pp.150–155.

[48] Siciliano B., Sciavicco L. ,et al. (2012) “Modelling and Control of Robot Manipulators”, Springer Science & Business Media.

[49] Murray R. (2017) , “A mathematical introduction to robotic manipulation”, CRC press.

[50] Huei- Huang Lee,(2014) , Engineering Dynamics Labs with SolidWorks Motion, SDC Publications.

[51] Singh R., R. Sugandhi,et al. (2018) "Development of digital control system in LabVIEW for stepper motor drives," International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), Kottayam, India , pp. 1-6, doi: 10.1109/ICCSDET. 8821122.

[52] Fundamentals, Labview(2005). "National Instruments." Austin, Texas, USA.

[53] Green K. and Tashman L. (2009). Percentage Error: What Denominator?. Foresight: The International Journal of Applied Forecasting. Vol. 12. pp.36-40.