FORWARD & INVERSE KINEMATIC ANALYSIS OF A 6 DOF INDUSTRIAL ROBOT CONTROL WITH HUMAN ROBOT INTERACTION

By

Dantham Arunbabu [1], Nati Venkata Rama Krishna [2], Tirumalla Suresh [3], Yadla Prasanna Kumar [4], Ch. Pavan Kumar [5]

[1,2,3,4]B.Tech, [5]Assistant Professor, Aditya College of Engineering & Technology, Surampalem, A.P, India, 533437

ABSTRACT

Many institutions face challenges in training individuals to operate expensive equipment. A common issue involves limited accessibility to costly robotics equipment for students to gain practical experience. Hence, Robot Simulation Software (RSS) has become increasingly essential. Furthermore, hands-on experience with programmable robots provides students with profound understanding. This study presents the creation of a visual software package, focusing on the AL5B Robot arm as a case study. It employs virtual reality interface design methodology and utilizes MATLAB/Simulink and AutoCAD for testing the motion characteristics of the AL5B Robot arm. Additionally, the developed model is implemented and tested to analyze and refine algorithms for Kinematics, Inverse Kinematics, Velocity Kinematics (Jacobian), and Trajectory Planning. The software package lifecycle is documented, followed by a comparison between the simulated package and the physical arm in terms of motion, trajectories, and kinematics. The developed package serves as an educational tool to enhance applied and experimental research opportunities and improve robotics curricula at both the graduate and undergraduate levels.

Keywords: Virtual Reality, Modeling, Simulation, Interface, MATLAB/Simulink, AL5B Robot arm, Forward Kinematics, Inverse Kinematics, Trajectory Planning, Jacobian

1. INTRODUCTION

1.1 Motivation

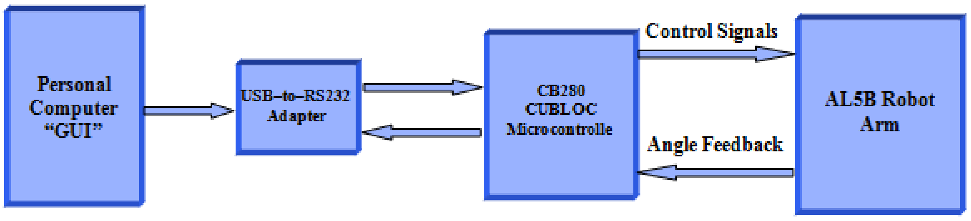
Mobile robotics and manipulator-based robotics have been the foundations of robotics teaching for the past 20 years. Robotics is still a developing field in the Gaza Strip, and obtaining commercial robots is virtually impossible. However, the availability of small, inexpensive mobile robots has encouraged their use in the classroom across a spectrum of educational levels around the world [KOL 01]. Researchers from all across the world have created educational models and given middle- to high-school students[WED 02] and kindergarten pupils [MIL 00] accessto hands-on learning using mobile robots. The primary focusof educators continues to be undergraduate and graduate robotics instruction [FER 00]. Due to the high initial beginning costs, manipulator-based robotics teaching has not received much attention. In addition to providing interactive learning opportunities and robot competitions, Murphy [MUR 00] encouraged the use of robotics to educate artificial intelligence. Sutherland [SUT 00] proposed a practical method for introducing undergraduate students to robotics while working with constrained resources. All students are welcome to participate in the yearly competition held by Palestine Technical College in Deir el Balah (PTCDB) [PTCDB]. The results of this study benefit these colleges by helping to create a software package that can be utilised as a teaching aid in robotics programmes by adding simulation and hands-on labs. The package that has emerged strengthens the mixed theoretical robotics presentations that have been introduced in these colleges.

1.2PROBLEM STATEMENT AND GOAL

The Islamic University, Al-Azhar University, and Palestine Technical College are just a few of the universities and colleges in the Gaza Strip that offer robotics courses. There are no actual labs to put the theoretical concepts learned in robotics courses at nearby universities, thus they are primarily theoretical. The objective of this research is to create a visual software package that mimics a five degrees of freedom (DOF) robot arm. This package will include the majority of the key concepts covered in the beginning robotics manipulators course. Given the restricted supply of educational resources for robotics courses and the high costof robot equipment and tools, this will increase education, training, and research in graduate and undergraduaterobotics studies. The AL5B robot arm [LYN 06] is chosen as a case study in this research because it offers a straightforward, affordable solution and serves as an excellent illustration of robotic manipulators. The motional characteristics of the arm will be tested using MATLAB/Simulink and AutoCAD. The forward kinematics, inverse kinematics, velocity kinematics (Jacobian), and trajectory planning problems are described, implemented, and tested using athorough research and mathematical analysis.

1.2 SystemOverview

The full system block diagram depicted in Figure (1.1) is made up of numerous components, including an AL5B robot arm, a CUBLOC microcontroller [COM 05], and a personal computer with a serial connection adapter. The forward, inverse kinematic, path and trajectory planning, Jacobian, and controller components make up the Graphical User Interface (GUI) created by the MATLAB software. Finding the end-location effector's in space while being aware of its joint movements is the goal of forward kinematics. The process of inverse kinematics involves identifying the joint variables that correlate to a specific endeffector position and orientation. Trajectory is concerned with the timing of each component of the path, whereas path is defined as asuccession of robot configurations in a specific order without consideration of timing. It is necessary to determine each joint velocity at the designated joint positions;Jacobian is used to do this. The last component isa straightforward controller block that a GUI software uses to control the robot arm



Figure(1.1):SystemBlock Diagram

The simplest form of communication between two devices is serial communication. The SERIAL function of MATLAB can be used to build a serial port object, which is how a serial interface is established. The CUBLOCmicrocontroller's primary job is to create an interface between the PC and AL5B robot arm by taking data from the serial port and transferring it to the arm servo motors. following that by transmitting the data from the servomotor encoders to the PC via the serial port

1.3 Literature Review

To make the accompanying kinematics calculations easier, many industrial robot arms are constructed with straight forward geometries like intersecting or parallel joint axes [MAN 96]. However, they are expensive for researchers and students. Due of its low cost, flexibility, and resemblance to industrial robot arms, AL5B is a good substitute for such robot manipulators. A computer simulation is produced using the forward and inverse kinematics from papers that developed software for modelling 2D and 3D robot arms, such as [MAN 96, KOY 07, and GUR 97]. Programming languages are then used to prepare the simulation and test the properties of this robot arm. To create a user- friendly GUI for users as an educational aid, 2D and 3D visualisation are employed. The lack of research onpath and trajectory planning renders the software unfinished. For virtual reality development and evaluating the practicality of designs prior to the implementation phase forthe industrial SCARA robot, located in the Control Robot Lab of the University of Oradea, [PAS 07] used V- Realm Builder 2.0 and Simulink. They also demonstrated how to manipulate items in a virtual world using the3D Joystick. Using VRML and MATLAB Simulink, Martin and Arya created Robot Simulation Software for forward and inverse kinematics in [ROH 00, WIR 04]. The system's output was graphically capable and flexible in terms of3D representation. The system, however, lacked user friendliness and was unable to function as a standalone application. 3 A Mitsubishi RV-2AJ robot was used as a case study in [JAM 08], which described the creation of the Robot Simulation Software (RSS). The project used MATLAB/Simulink and V-Realm Builder as tools and applied thevirtual reality interface design methodology. A RSS software life cycle was constructed, and a robot created. A Visual C++ and OpenGL programme for 3D simulation of the serial industrial robots was shown in [MAR 06]. It began by taking into account the robot's forward kinematics. Each robot joint's position and orientationmay be determined using the functions written into the source code, including the gripper's position and orientation. The application was able to simulate and render the 3D kinematics of the robot with the aid of OpenGL features. Using the most potent non-real-time modelling and control design tools, a method for creating real-time simulators of complicated electromechanical systems was proposed in [FER 08]. This method utilised open source software and commercially available tools, and only a small number of interface blocks had to be createdin order to be added to the Simulink and Dymola models, respectively. The real-time simulation model may be fully integrated with the modelling and validation work done on a joint prototype during the early stages of the arm development process, producing quite precise and dependable results nearly without effort. It is very simpleto verify incrementally the Simulink arm controller description. A lot of work was invested into developing a sophisticated open source package that would serve as the foundation for a human machine interface that could accept input for force disturbances, motion commands, and 3D arm motion visualisation. The kinematics analysis of industrial robots is discussed in a lot of literature [CRA 05]. The majority of them avoid talking about the inexpensive educational robot arms. After reviewing the most recent batch of articles, we can see that none of them provide college-level students with a comprehensive educational tool for controlling the AL5B robot arm. As a result, the AL5B instructional robot's mathematical model and kinematicanalysis will be examined in this study. To display the robot arm motion in relation to its mathematical analysisand interaction with a physical robot, a Visual Software Program (VSP) will also be created.

1.4 Objectives

In order to achieve the main goal objectives of this study, the work is going to bedividedinto two phases.

1.4 Drawing a 3D Model for the Robot Arm using MATLAB/Simulink andAutoCAD.

1.4.1 Developing a visual software package, for testing motional characteristics of theAL5B Robot arm

1.4.2 Designing a Graphical User Interface "GUI" using MATLAB.

1.4.3 Derivation of a complete kinematic model for the robot.

1.4.4 Studying the theory of kinematics in order to analyze of the 5 DOFAL5B Robot Arm.

1.4.5 Applying the Denavit-Hartenberg (D-H) model to the physical arm linksand joints to derive the forward kinematic equations.

1.4.6 Finding the Inverse kinematics solutions for this educational manipulatorand suggesting a method for decreasing multiple solutions in IK.

1.4.7 Derivation of the Velocity Kinematics (Jacobian) of the Manipulator consideringsingularity.

1.4.8 Applying a Path and Trajectory planning algorithm

Phase2:

Development of an electronic interfacing circuit between the AL5B robot armand the developed GUI program

2. THEORETICAL BACKGROUND

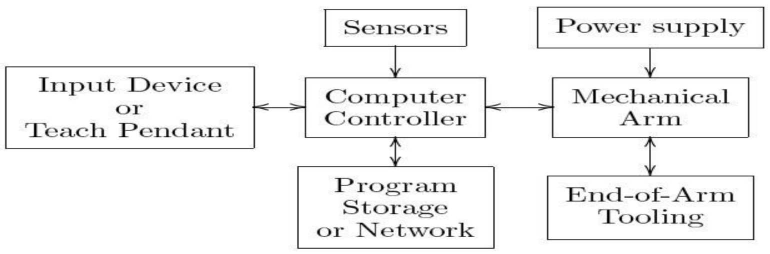
2.1 CommonKinematicArrangementsofManipulators

Modern technology's relatively new discipline of robotics transcends conventional engineering limitations. Electrical engineering, mechanical engineering, systems and industrial engineering, computer science, economics, and mathematics are all necessary to comprehend the complexity of robots and the applications theyare used for. To address the complexity of the robotics and factory automation fields, new engineering disciplines such as manufacturing engineering, applications engineering, and knowledge engineering have emerged. The foundational concepts of robotics—kinematics, motion planning, velocity kinematics, computer interface, and control—are the focus of this thesis. The most crucial ideas in these fields are introduced in this chapter as they relate to industrial robot manipulators. The bulk of robot applications deal with industrial robot arms functioning in structured manufacturing environments so that a first introduction to the subject of robotics mustinvolve a careful discussion of the themes in this thesis. A Czech playwright coined the term "robot" in 1920 to refer to labour. A robot is essentially any computer- controlled autonomous equipment, including teleoperators, submersible vehicles, autonomous land rovers, etc..

Figure(2.1):AL5BRoboticArm

A conventional robot, which is essentially a mechanical arm controlled by a computer, is depicted in Figure (2.1). Even though these gadgets are far different from the robots of science fiction, they are nevertheless incredibly complicated electro-mechanical systems that demand sophisticated analytical techniques, creating avariety of difficult and intriguing research topics. It is recognised that a robot manipulator is more than just a collection of mechanical links. Figure (2.2), which shows an arm, external power source, end-of-arm tooling, external and internal sensors, computer interface, and control computer, is simply one component of a comprehensive systems.

Figure (2.2)ComponentsofRoboticSystem.



Although there are numerous techniques to build kinematic chains using prismatic and revolute joints, only afew of these are frequently employed in actuality. Here, we give a quick description of a few of the most common layouts.

2.1.1.ArticulatedManipulator(RRR)

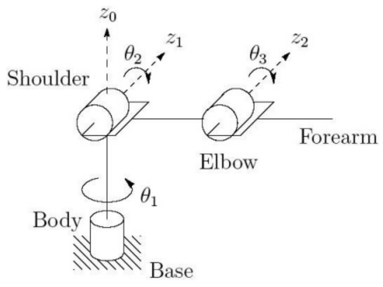
Figure (2.3) shows the (ABB IRB1400) articulated manipulator which called a revolute manipulator [SPO 05]. The (RRR) means the type of joint is (Revolute – Revolute – Revolute) and (P) means the type of joint is (Prismatic)

Figure(2.3):TheABBIRB1400Robot.

The parallelogram linkage, like the Motorman SK16 in Figure, is a typical revolute jointdesign (2.4) In all of these configurations, the joint axis z2 is perpendicular to z0 and parallel toz1, respectively. An elbow manipulator is the name for this kind of device. Figure (2.5) depictsthe elbow manipulator's structure, nomenclature, and workspace. Figure (2.6) depicts the elbow manipulator's workspace (2.6)

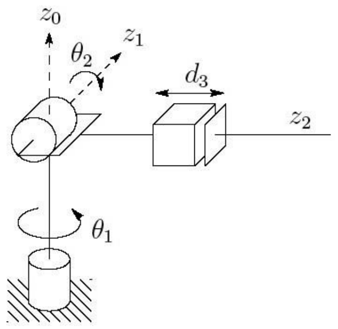
The elbow manipulator has a number of benefits that make it an appealing and well-liked design. Therevolute manipulator offers relatively large freedom of movement in a small space. The actuator for joint 3 is found on link 1 of the parallelogram linkage manipulator. Links 2 and 3 can be made lighterand the motors themselves can be less powerful because link 1 carries the weight of the motor.

Figure(2.4):The Motor man SK16Manipulator



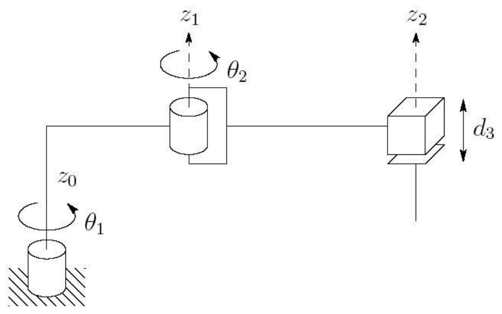
Figure(2.5):Structure of The Elbow Manipulator

2.1.2 Spherical Manipulator (RRP) By substituting a prismatic joint for the third or elbow joint in the revolute manipulator, as shownin Figure, the spherical manipulator can be created (2.7). The term "spherical manipulator" comesfrom the fact that the first three joint variables are the same asthe spherical coordinates describingthe position of the end-effector with respect to a frame whose origin sits at the intersection of thethree z-axes. The Stanford arm is shown in Figure (2.8). [SPO 05], one of the most well-known spherical robots. The workspace of a spherical manipulator is shown in Figure (2.9).



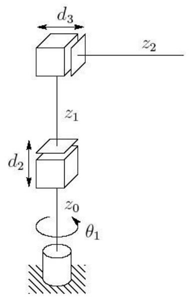
Figure(2.6):The Spherical Manipulator.

2.1.3 SCARA Manipulator (RRP) Popular manipulators include the SCARA arm (short for Selective Compliant Articulated Robot for Assembly), which is depicted in Figure (2.10) [SPO 05]. The SCARA has an RRP structure and differs significantly from the spherical manipulator in terms of both look and application scope. Z0, Z1, and Z2 are mutually parallel in the SCARA. The Epson E2L653S manipulator [SPO 05] is seen in Figure (2.11). Figure displays the SCARA manipulator workspace (2.12)



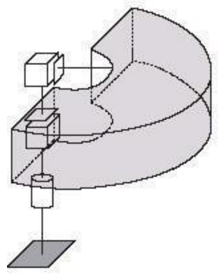
Figure(2.7):TheSCARA(SelectiveCompliantArticulatedRobotforAssembly).

Figure(2.8):TheEpsonE2L653SSCARARobot.



Figure(2.9):TheCylindricalManipulator.

Figure(2.10):TheSeikoRT3300Robot.



Figure(2.11):WorkspaceoftheCylindricalManipulator.

2.1.4 Cartesian Manipulator(PPP)

A Cartesian manipulator is one with prismatic first three joints, as depicted in Figure (2.16). The Cartesian coordinates of the end-effector in relation to the base are the joint variables for the Cartesian manipulator. Figureillustrates a Cartesian robot from Epson-Seiko, model number [SPO 05]. (2.17). Figure depicts a Cartesian manipulator's workspace (2.18)

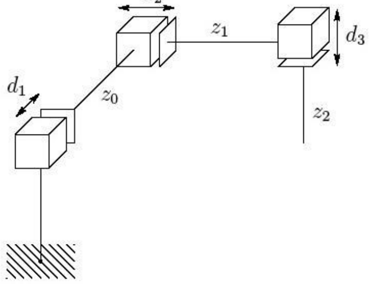


Figure (2.12):TheCartesianManipulator.

Figure (2.13):The Epson Cartesian Robot.

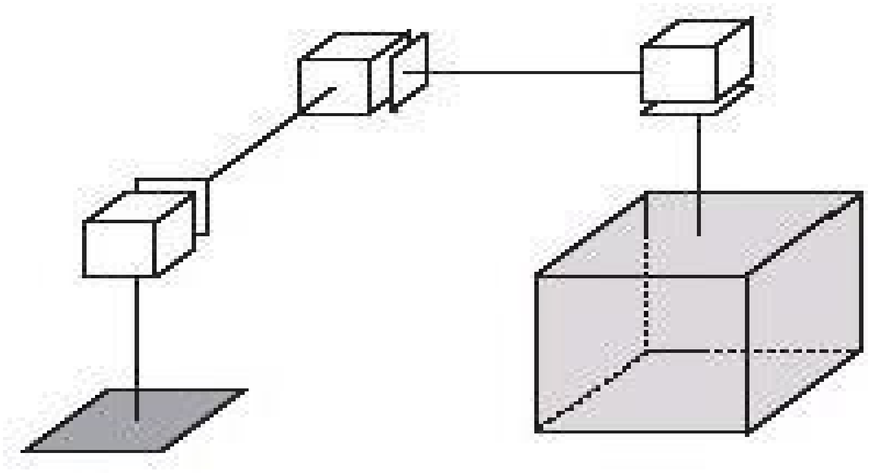


Figure (2.14):Work space of the Cartesian manipulator.

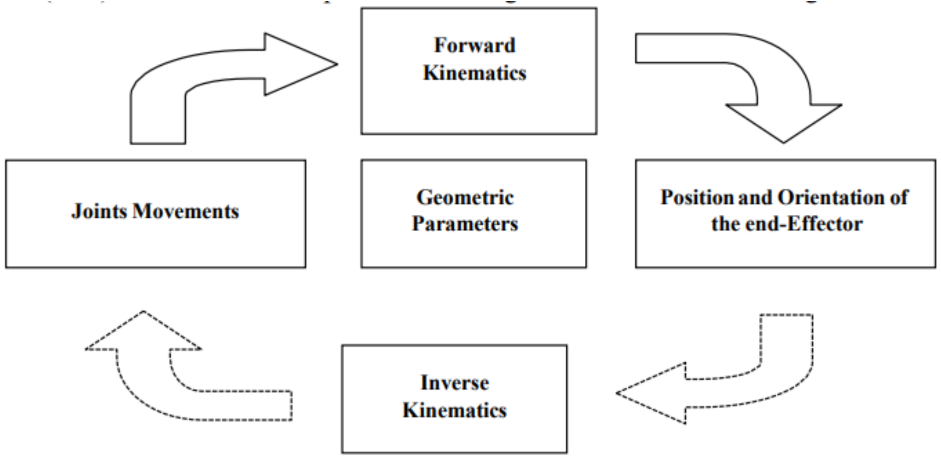
2.2 Kinematic Modeling

System analysis, such as the kinematics analysis, is necessary for simulating robots because it allows researchers to examine how each component of the robot mechanism moves and how those motions relate to one another. Inverse and forward analyses of kinematics are both included. Finding the end-location effector's inspace while being aware of how its joints move constitutes the forward kinematics.

$F(1,2,n)$ $[x,y,z,R]$, and the inverse kinematics consists of the determination of

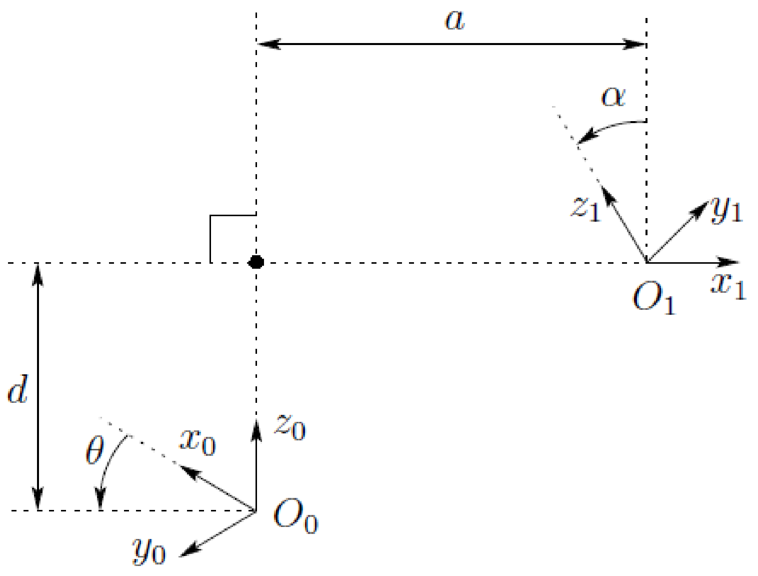
the jointvariables corresponding to a given end-effector position and orientation as F (x, y, z, n .θ,…2 ,θ_1,θ_2 and so on)

Figure (2.20) below shows a simplified block diagram of kinematic modeling.



Figure(2.15):Kinematics Block Diagram

A commonly usedconventionfor selectingframes of referencein roboticapplicationsistheDenavit-HartenbergorD-HconventionasshowninFigure(2.21).In thisconventioneachhomogenoustransformationT_i isrepresentedasapro "four"basictransformationsT_i□$Rot(z$,□$_i)Trans(z,d_i)Trans(x,a_i)Rot(x,$□$_i)$



Figure(2.16):DHFrameAssignment

Where the notation *Rot*(x, θ_i)stands for rotation about x_i axis by θ_i, *Trans*(x,a_i) is translation along x_i axis by a distance a_i, *Rot*(z, θ_i) stands for rotation about z_i axis by θ_i, and *Trans*(z,d_i) is the translation along z_i axis by a distance d_i.

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

$$= \begin{bmatrix} c_{\theta i} & -s_{\theta i}c_{\alpha i} & s_{\theta i}s_{\alpha i} & a_i c_{\theta i} \\ s_{\theta i} & c_{\theta i}c_{\alpha i} & -c_{\theta i}s_{\alpha i} & a_i s_{\alpha i} \\ 0 & s_{\alpha i} & c_{\alpha i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2.2)$$

Where the four quantities θ_i, a_i,d_i Are the parameters of link I and joint i.

The Figure below illustrates the link frames attached so that frame {i} attached rigidly to linki. The various parameters in previous equation are given the following names: ai (Length) is the distance from zi to zi 1 , measured along zi ; i (Twist), is the angle between zi and zi 1 , measured about xi ; di (Offset), is the distance from xi to xi 1 measured along zi ; and (Angle) , is the angle between measured about ziιθ ;
In the usual case of a revolute joint, is called the joint variable, the other three quantities are thefixed link parameters. Another expression can be used where a homogeneous transformation matrix H represents a rotation by angle α about the current x-axis followed by a translation of a units along the current x-axis, followed by a translation of d units along the current z- axis, followed by arotation by angle θ about the current z-axis, is given by H where His given by:

$$H = Rot_{x,\alpha} Trans_{x,a} Trans_{z,d} Rot_{z,\theta} \tag{2.3}$$

$$= \begin{bmatrix} c_\theta & -s_\theta & 0 & a \\ c_\alpha s_\theta & c_\theta c_\alpha & -s_\alpha & -ds_\alpha \\ s_\alpha s_\theta & s_\alpha c_\theta & c_\alpha & dc_\alpha \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

Aspecialcaseofhomogeneouscoordinates,whicharewidelyemployedinthefieldofcomputer graphics, isthehomogeneous representation mentioned in the above equation. There, in addition to translation and rotation, oneis also interested in scaling and/or perspective transformations. The form of the most prevalent homogeneoustransformationis

$$H = \begin{bmatrix} R_{3x3} & d_{3x1} \\ f_{1x3} & s_{1x1} \end{bmatrix} = \begin{bmatrix} Rotation & Translation \\ Perspective & Scale\ factor \end{bmatrix}$$

$$= \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

$$T_i^j = T_{i+1}^j .. T_i^{i-1} \tag{2.6}$$

3.KINEMATICS

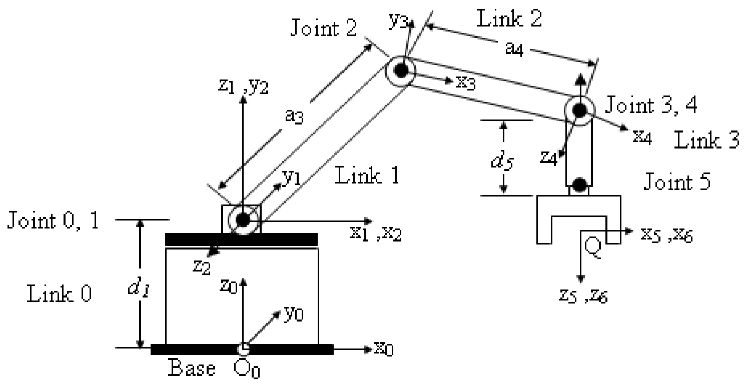
Introduction

The kinematics solutions of any robot manipulator are divided into two solutions, the first one is the solution of Forward kinematics, and the second one is the inverse kinematics solution. Forward kinematics will determine where the robot's manipulator hand will be if all joints are known. Where the inverse kinematics will calculate what each joint variable must be if the desired position and orientation of end-effector is determined. Hence, Forward kinematics is defined as transformation from joint space to Cartesian space where as Inverse kinematics is defined as transformation from Cartesian space to joint space.

3.1 DIRECT & FORWARD KINAMATICS

The forward kinematics problem can be stated as follows: Given the joint variables of the robot, determine the position and orientation of the end-effector. Since each joint has a single degree of n the angle ofθ2,… θ1, θfreedom, the action ofeach joint can be described by a single number, i.e.

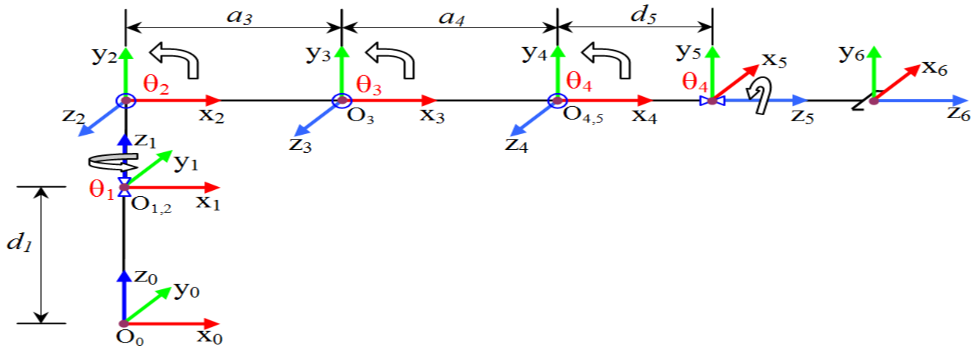
rotation in the case of a revolute joint. The objective of forward kinematic analysis is to determine the cumulative effect of the joint variables. Suppose a robot has n+l links numbered from zero to n starting from the base of the robot, which is taken as link 0. The joints are numbered from one to n, and zi is a unit vector along the axis in space about which the links i-1 and i are connected. The i- th joint variable is denoted by q, In the case of a revolute joint, q, is the angle of rotation, while in the case of a prismatic joint q, is the joint translation. Next, a coordinate frameis attached rigidly to each link. To be specific, we choose frames 1 throughn such that the frame i is rigidly attached to link i. Figure (3.1) illustrates the idea of attaching frames rigidly to links in the case of an AL5B robot.



Figure(3.1):AL5BRobotArmFrameAssignment

3.2Assigning the Coordinate Frames

AL5B has five rotational joints and a moving grip as shown in Figure (3.1). Joint 1 represents the shoulder and its axis of motion is z1. This joint provides a rotational 1 angular motion around z1 axis in x1y1 plane. Joint 2 is identified as the Upper Arm and its axis is perpendicular to Joint 1 axis. It provides a rotational angular motion around z2 axis in x2y2 plane.



Figure(3.2):CoordinateFrames ofAL5BRoboticArm

3.3 AL5B DH Parameters

As explained in chapter 2 many methods can be used in the direct kinematics calculation. The Denavit-Hartenberg analysis is one of the most used, in this method the direct kinematics is determined from some parameters that have to be defined, depending on each mechanism. However, it was chosen to use the homogeneous transformation matrix. In this, analysis, once it is easily defined one coordinate transformation between two frames, where the position and orientation are fixed one with respect to the other it is possible to work with elementary homogeneous transformation operations. D-H parameters for AL5B defined for the assigned frames in Table 3.

Table (3.1): DH Parameter for AL5B Robot Arm

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	d_1	θ_1^*
2	90	0	0	θ_2^*
3	0	a_3	0	θ_3^*
4	0	a_4	0	$(\theta_4\text{-}90)^*$
5	-90	0	d_5	θ_5^*
6	0	0	0	Gripper

By substituting the parameters from Table (3.1) into equation (2.4), the transformation matrices T1 to T6 can be obtained as shown below. For example, T1 shows the transformation between frames 0 and 1 (designating Ci as cos θi and i S as sinθi etc).

$$T_1^{0} = \begin{bmatrix} c_{\theta_1} & -s_{\theta_1} & 0 & 0 \\ s_{\theta_1} & c_{\theta_1} & 0 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.8)$$

$$T_2^{1} = \begin{bmatrix} c_{\theta_2} & -s_{\theta_2} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_{\theta_2} & c_{\theta_2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.9)$$

$$T_3^{2} = \begin{bmatrix} c_{\theta_3} & -s_{\theta_3} & 0 & a_3 \\ s_{\theta_3} & c_{\theta_3} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.10)$$

$$T_4^{3} = \begin{bmatrix} c_{\theta_4} & -s_{\theta_4} & 0 & a_4 \\ s_{\theta_4} & c_{\theta_4} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.11)$$

$$T_5^{4} = \begin{bmatrix} c_{\theta_5} & -s_{\theta_5} & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ s_{\theta_5} & c_{\theta_5} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.12)$$

$$T_{Gripper} = \begin{bmatrix} c_{\theta_5} & -s_{\theta_5} & 0 & 0 \\ s_{\theta_5} & c_{\theta_5} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.13)$$

Using the above values of the transformation matrices; the link transformations can be concatenated (multiplied together) to find the single transformation that relates frame (5) to frame (0):

$$T_5^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.14)$$

The transformation given by equation (3.14) is a function of all 5 joint variables. From the robot's joint position, the Cartesian position and orientation of the last link may be computed using above equation (3.14). The first three columns in the matrices represent the orientation of the end effectors, whereas the last column represents the position of the end effectors [MOH 09].
Simulation and Interfacing of 5 DOF Educational Robot Arm 24 orientation and position of the end effectors can be calculated in terms of joint angles using:

$$\begin{aligned} n_x &= ((c_1c_2c_3 - c_1s_2s_3)c_4 + (-c_1c_2s_3 - c_1s_2c_3)s_4)c_5 + s_1s_5 \\ n_y &= ((s_1c_2c_3 - s_1s_2s_3)c_4 + (-s_1c_2s_3 - s_1s_2c_3)s_4)c_5 - c_1s_5 \qquad (3.15) \\ n_z &= ((s_2c_3 + c_2s_3)c_4 + (-s_2s_3 + c_2c_3)s_4)c_5 \end{aligned}$$

$$\begin{aligned} o_x &= -((c_1c_2c_3 - c_1s_2s_3)c_4 + (-c_1c_2s_3 - c_1s_2c_3)s_4)s_5 + s_1c_5 \\ o_y &= -((s_1c_2c_3 - s_1s_2s_3)c_4 + (-s_1c_2s_3 - s_1s_2c_3)s_4)s_5 - c_1c_5 \qquad (3.16) \\ o_z &= (c_2c_3 - s_2s_3)s_4)s_5 - ((s_2c_3 + c_2s_3)c_4 \end{aligned}$$

$$\begin{aligned} a_x &= -(c_1c_2c_3 - c_1s_2s_3)s_4 + (-c_1c_2s_3 - c_1s_2c_3)c_4 \\ a_y &= -(s_1c_2c_3 - s_1s_2s_3)s_4 + (-s_1c_2s_3 - s_1s_2c_3)c_4 \qquad (3.17) \\ a_z &= (c_2c_3 - s_2s_3)c_4 - (s_2c_3 + c_2s_3)s_4 \end{aligned}$$

$$\begin{aligned} d_x &= (-(c_1c_2c_3 - c_1s_2s_3)s_4 + (-c_1c_2s_3 - c_1s_2c_3)c_4)d_5 + (c_1c_2c_3 - c_1s_2s_3)a_4 + c_1c_2a_3 \\ d_y &= (-(s_1c_2c_3 - s_1s_2s_3)s_4 + (-s_1c_2s_3 - s_1s_2c_3)c_4)d_5 + (s_1c_2c_3 - s_1s_2s_3)a_4 + s_1c_2a_3 \qquad (3.18) \\ d_z &= (-(s_2c_3 + c_2s_3)s_4 + (-s_2s_3 + c_2c_3)c_4)d_5 + (s_2c_3 + c_2s_3)a_4 + s_2a_3 + d_1 \end{aligned}$$

3.4 Inverse kinematics

Inverse Kinematics (IK) analysis determines the joint angles for desired position and orientation in Cartesian space. Total transformation matrix in equation. (3.14) will be used to calculate inverse kinematics equations. IK is more difficult problem than forward kinematics. The solution of inverse kinematic is more complex than direct kinematics and there is not any global analytical solution method. Each manipulator needs a particular method considering the system structure and restrictions. There are two solutions approaches namely, geometric and algebraic used for deriving the inverse kinematics solution. Let's start with geometric approach.

3.4.1Geometric Approach

Using IK-Cartesian mode, the user specifies the desired target position of the gripper in Cartesian space as

(x, y, z) where z is the height, and the angle of the gripper relative to ground, ψ (see Figure 3.4), is held constant. This constant ψ allows users to move objects without changing the object's orientation (the holding a cup of liquid scenario). In addition, by either keeping ψ fixed in position mode or keeping the wrist fixed relative to the rest of the arm, the inverse kinematic equations can be solved in closed form as we now show for the case of a fixed ψ [MOH 09].

$$\begin{aligned} d &= \sqrt{x_d^2 + y_d^2} \\ x_d &= d\cos(\theta_1) \\ y_d &= d\sin(\theta_1) \end{aligned} \qquad (3.19)$$

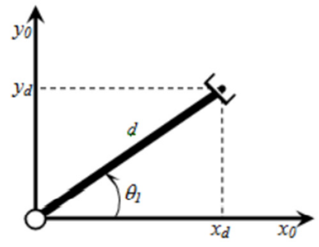


Figure (3.3): Top View of Robot.

Moving now to the planar view in Figure (3.4), we find a relationship between joint angles θ₂, θ₃ and θ₄ and ψ as follows:

$$\psi = \theta_2 + \theta_3 + \theta_4 \qquad (3.20)$$

Since ψ is given, we can calculate the radial distance and height of the wrist joint:

$$\begin{aligned} r_4 &= r_d - a_5\cos(\psi) \\ z_4 &= z_d - a_5\sin(\psi) \\ &or \\ r_4 &= a_3\cos(\theta_2) + a_4\cos(\theta_2+\theta_3) \\ z_4 &= a_3\sin(\theta_2) + a_4\sin(\theta_2+\theta_3) + d_1 \end{aligned} \qquad (3.21)$$

Now we want to determine θ₂ and θ₃. We first solve for α, β and s (from Figure 3.4) uses the law of cosines as:

$$\begin{aligned} \beta &= a\tan 2(s^2 + a_3^2 - a_4^2, 2a_3 s) \\ \alpha &= a\tan 2(z_4 - d_1, r_4) \\ s &= \sqrt{(z_4-d_1)^2 + r_4^2} \end{aligned} \qquad (3.22)$$

With these intermediate values, we can now find the remaining angle values as:

$$\begin{aligned} \theta_2 &= \alpha \pm \beta \\ \theta_3 &= a\tan 2(s^2 - a_3^2 - a_4^2, 2a_3 a_4) \\ \theta_4 &= \psi - \theta_2 - \theta_3 \end{aligned} \qquad (3.23)$$

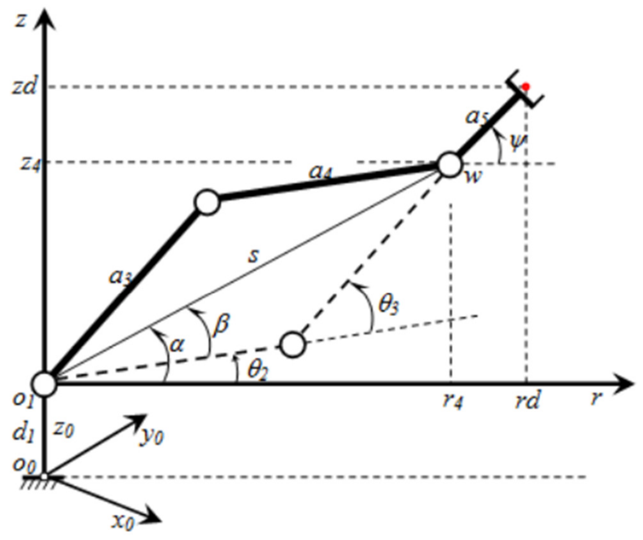


Figure (3.4): Planar View of AL5B Robot Arm.

3.4.2 Analytical (algebraic) Approach

Using the X, Y and Z resultants gotten in the direct kinematics:

$$x = c_1[a_3c_2 + a_4c_{23} + d_5c_{234}] \tag{3.24}$$

$$y = s_1[a_3c_2 + a_4c_{23} + d_5c_{234}] \tag{3.25}$$

$$z = [d_5c_{234} + a_4s_{23} + a_3s_2] + d_1 \tag{3.26}$$

The simplified equation is gotten:

$$(3.24)^2 + (3.25)^2 \Rightarrow a_3c_2 + a_4c_{23} = \pm\sqrt{x^2 + y^2} - d_5c_{234} \tag{3.27}$$

The first joint movement, defined by θ$_1$, can be calculated using geometric parameters only:

$$\therefore \theta_1 = a\tan 2(y, x)$$

Now we can calculate θ$_3$, by using equation (3.26):

$$(3.26) \Rightarrow a_3s_2 + a_4s_{23} = z - d_5s_{234} - d_1 \tag{3.28}$$

$$(3.27)^2 + (3.28)^2 \Rightarrow c3 = \frac{(z - d_5s_{234} - d_1)^2 + (\pm\sqrt{x^2 + y^2} - d_5c_{234})^2 - a_3^2 - a_4^2}{2a_3a_4} \tag{3.29}$$

$$c_\psi = c_{234}, s_\psi = s_{234}$$

where

$$c_{234} = \cos(\theta_2 + \theta_3 + \theta_4), s_{234} = \sin(\theta_2 + \theta_3 + \theta_4)$$

$$\therefore c_3 = \frac{(z - d_5s_\psi - d_1)^2 + (\pm\sqrt{x^2 + y^2} - d_5c_\psi)^2 - a_3^2 - a_4^2}{2a_3a_4} \tag{3.30}$$

$$s_3 = \pm\sqrt{1 - c_3^2}$$

Which results,

$$\therefore \theta_4 = a\tan 2(s_4, c_4) \tag{3.48}$$

4.DIFFERENTIAL KINEMATICS AND STATICS

4.1 Velocity Kinematics/Arm Jacobian

The Jacobian is one of the most important quantities in the analysis and control of robot motion. It is used for smooth trajectory planning and execution in the derivation of the dynamic equation. To investigate target with specified velocity, each joint velocity at the specified joint positions needs to be found. This is accomplished using Jacobian, which is used to relate joint velocities to the linear and angular velocities of the endeffector [SPO 05]. The relationships between joint velocities θ& , and the linear and angular velocities, p& and ω respectively, of the end effector:

$$\dot{p} = J_P(\theta)\dot{\theta} \qquad (4.1)$$

$$w = J_w(\theta)\dot{\theta} \qquad (4.2)$$

$$\text{where} \quad \theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \qquad (4.3)$$

The above equations are combined to form J, which relates both linear, and angular velocity:

$$v = J(\theta)\dot{\theta} \qquad (4.4)$$

Where J(θ) is in the form

$$J = \begin{bmatrix} J_{p1} & & J_{pn} \\ & 6\times n & \\ J_{w1} & & J_{wn} \end{bmatrix} \qquad (4.5)$$

The number of columns of the Jacobian represents the number of degrees of freedom or links of the manipulator. There are always three rows for linear velocity in the x, y and z directions, and three for angular velocity. Hence, for a six degree of freedom manipulator, the Jacobian is a 6 by 6 square matrix. The Jacobian can be calculated from the following equation.

$$J_{vi} = \begin{cases} z_{i-1}\times(o_n - o_{i-1}) & \textit{for revolute } \text{joint } i \\ z_{i-1} & \textit{for prismatic } \text{joint } i \end{cases}$$
$$J_{wi} = \begin{cases} z_{i-1} & \textit{for revolute } \text{joint } i \\ 0 & \textit{for prismatic } \text{joint } i \end{cases} \qquad (4.6)$$

Where: vi J angular velocity and wi J linear velocity. For the AL5B robot arm the Jacobian matrix is equal 6x5
[MOH 09].

$$J(q) = \begin{bmatrix} z_0\times(o_5-o_0) & z_1\times(o_5-o_1) & z_2\times(o_5-o_2) & z_3\times(o_5-o_3) & z_4\times(o_5-o_4) \\ z_0 & z_1 & z_2 & z_3 & z_4 \end{bmatrix} \qquad (4.7)$$

From the forward kinematic we can find

$$o_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\ o_1 = \begin{bmatrix} 0 \\ 0 \\ d1 \end{bmatrix},\ o_2 = \begin{bmatrix} 0 \\ 0 \\ d1 \end{bmatrix},\ o_3 = \begin{bmatrix} c_1c_2a_3 \\ c_1s_1a_3 \\ s_2a_3 + d_1 \end{bmatrix}$$

$$o_4 = \begin{bmatrix} c_1c_{23}a_4 + c_1c_2a_3 \\ s_1c_{23}a_4 + s_1c_2a_3 \\ s_{23}a_4 + s_2a_3 + d_1 \end{bmatrix},\ o_5 = \begin{bmatrix} c_1c_{234}d_5 + c_1c_{23}a_4 + c_1c_2a_3 \\ s_1c_{234}d_5 + s_1c_{23}a_4 + s_1c_2a_3 \\ s_{234}d_5 + s_{23}a_4 + s_2a_3 + d1 \end{bmatrix} \qquad (4.8)$$

We Can Find

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

$$z_3 = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix},\ z_4 = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix},\ z_5 = \begin{bmatrix} c_1c_{234} \\ s_1c_{234} \\ -s_{234} \end{bmatrix} \qquad (4.9)$$

Now we can write the Jacobian matrix as shown in equation (4.10):

$$J(q) = [J_1 \quad J_2 \quad J_3 \quad J_4 \quad J_5] \qquad (4.10)$$

$$J_1 = z_0 \times (o_5 - o_0) = \begin{bmatrix} -s_1c_{234}d_5 - s_1c_{23}a_4 - s_1c_2a_3 \\ c_1c_{234}d_5 + c_1c_{23}a_4 + c_1c_2a_3 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad (4.11)$$

$$J_2 = z_1 \times (o_5 - o_1) = \begin{bmatrix} -s_1c_{234}d_5 - s_1c_{23}a_4 - s_1c_2a_3 \\ c_1c_{234}d_5 + c_1c_{23}a_4 + c_1c_2a_3 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad (4.12)$$

$$J_3 = z_2 \times (o_5 - o_2) = \begin{bmatrix} -c_1(s_{234}d_5 + s_{23}a_4 + s_2a_3) \\ -s_1(s_{234}d_5 + s_{23}a_4 + s_2a_3) \\ s_1(s_1c_{234}d_5 + s_1c_{23}a_4 + s_1c_2a_3) + c_1(c_1c_{234}d_5 + c_1c_{23}a_4 + c_1c_2a_3) \\ s_1 \\ -c_1 \\ 0 \end{bmatrix} \qquad (4.13)$$

$$J_4 = z_3 \times (o_5 - o_3) = \begin{bmatrix} -c_1(s_{234}d_5 + s_{23}a_4) \\ -s_1(s_{234}d_5 + s_{23}a_4) \\ s_1(s_1c_{234}d_5 + s_1c_{23}a_4) + c_1(c_1c_{234}d_5 + c_1c_{23}a_4) \\ s_1 \\ -c_1 \\ 0 \end{bmatrix} \qquad (4.14)$$

$$J_5 = z_4 \times (o_5 - o_4) = \begin{bmatrix} -c_1 s_{234} d_5 \\ -s_1 s_{234} d_5 \\ s_1^2 c_{234} d_5 + c_1^2 c_{234} d_5 \\ s_1 \\ -c_1 \\ 0 \end{bmatrix} \qquad (4.15)$$

4.2 Kinematic Singularities

The Jacobian can also be used to indicate possible configurations at which singularities are present. Singularities are manipulator configurations in which one or more degrees of freedom are made redundant. This reduces the ability of the robot to move in 3D space close to a singularity, even though the area could be well within its workspace. In order to calculate the joint velocities necessary to produce a given Cartesian velocity the Jacobian matrix is inversed. If the inverse Jacobian is applied close to a singularity, the joint velocities approach infinity. For this reason, it is essential that the robot be designed so that it operates away from singularities, both boundary and internal. This is particularly important if the inverse Jacobian is to be calculated in a real time system. The rank of a matrix is not necessarily constant. Indeed, the rank of the manipulator Jacobian matrix will depend on the configuration q. Configurations for which the rank J(q) is less than its maximum value are called singularities or singular configurations. Identifying manipulator singularities is important for several reasons:

Boundary singularities

The boundary singularities occur when the end effector is on the boundary of the workspace, that is, the manipulator is either fully stretched out or fully retracted. For example, consider the case of two links, 2-DOF planar arm fully stretched out, as shown in Figure (4.1). In this configuration, two links are in a straight line and the end effector can be moved only in a direction perpendicular to the two links because it cannot move out of the workspace. Thus, the manipulator loses one degree of freedom. A similar situation will occur with θ2 equal 180°. Boundary singularities can be avoided by ensuring that the manipulator is not driven to boundaries of the reachable workspaces during its work cycle [SPO 05].

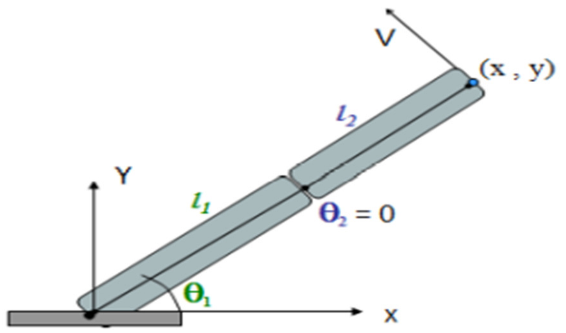


Figure (4.1): 2-DOF planar manipulator fully stretched out

(ii) Internal singularities

Internal singularities as shown in Figure (4.2) occur when the end effector is located inside the reachable workspace of the manipulator. These are caused when two or more joint axes become collinear or at specific end effector configurations [SPO 05].

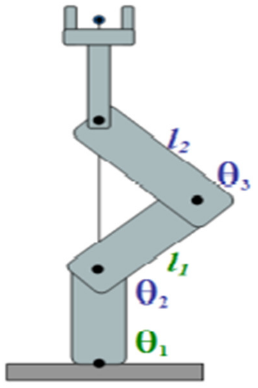


Figure (4.2): Internal Singularities Type

4.2.1. Computation of Singularities

The computation of internal singularities can be carried out by analyzing the rank of the Jacobian matrix. The Jacobian matrix loses its rank becomes ill conditioned at values of joint variables q at which its determinant vanishes, that means if the Jacobian is a 6 × n matrix and a configuration q is singular if and only if:

$$\det J(q) = 0 \tag{4.16}$$

If we now partition the Jacobian J into 3x3 blocks as

$$J = [Jp \mid Jo] = \left[\frac{J_{11}}{J_{21}} \mid \frac{J_{12}}{J_{22}}\right] \tag{4.17}$$

As the singularities are typical of configuration and are not dependent on frames chosen for kinematic analysis, the origin of the end effector frame can be chosen at the end of arm point this will make $J_{12} = 0$. In such a situation computation of determinant is greatly simplified, as

$$|J| = |J_{11}||J_{22}|$$

Hence, for a manipulator with a spherical wrist, the arm singularities are found from $|J_{11}| = 0$, and wrist singularities are found from $|J_{22}| = 0$. However, in our case the Jacobian matrix is non-square then we can find the det J (q) by using the pseudo inverse [SPO 05]. Let A be an m × n matrix, and let A^+ be the pseudo inverse of A. If A is of full rank, then A^+ can be computed as:

$$A^+ = \begin{cases} A^T[AA^T]^{-1} & m \le n \\ A^{-1} & m = n \\ [A^T A]^{-1} A^T & m \ge n \end{cases}$$

Then by applying the MATLAB function (pinv), the non square matrix can be solved.

4.3 Inverse Velocity and Acceleration

The Jacobian relationship

$$\zeta = J_Q \qquad (4.18)$$

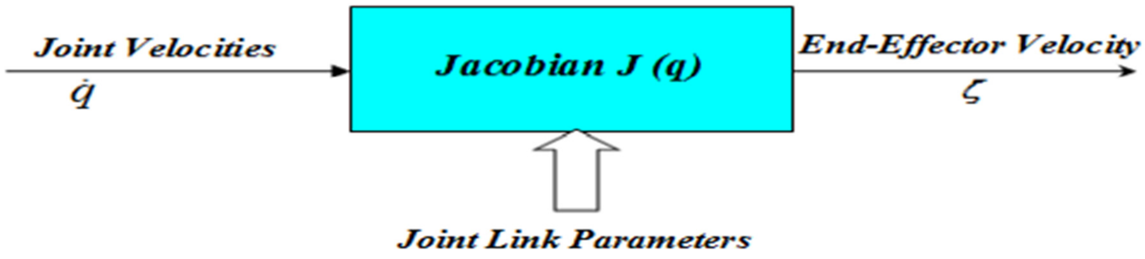


Figure (4.3): The Forward Differential Motion Model

The inverse velocity problem is the problem of finding the joint velocities q& that produce the desired end-effector velocity. It is perhaps a bit surprising that the inverse velocity relationship is conceptually simpler than inverse position. When the Jacobian is

square (i.e., $J \in R^{n \times n}$) and nonsingular, this problem can be solved by simply inverting the Jacobian matrix to give

$$\dot{q} = J^{-1}\zeta \qquad (4.19)$$

For manipulators that do not have exactly six links, the Jacobian cannot be inverted. In this case, there will be a solution to equation (4.18) if and only if ζ lies in the range space of the Jacobian. This can be determined by the following simple rank test. A vector ζ belongs to the range of J if and only if

$$rank\ J(q) = rank\ [J(q)|\zeta] \qquad (4.20)$$

In other words, equation (4.18) may be solved for $\dot{q} \in R^n$ provided that the rank of the augmented matrix $[J(q)|\zeta]$ is the same as the rank of the Jacobian J (q). This is a standard result from linear algebra, and several algorithms exist, such as Gaussian elimination, for solving such systems of linear equations. For the case when n > 6 we can solve for $\dot{q}$ using the right pseudo inverse of J.

4.4 Force/Torque Relationship

Interaction of the manipulator with the environment will produce forces and moments at the end-effector or tool. Let F = (Fx, Fy, Fz, nx, ny, nz) T represents the vector of forces and torques at the end-effector, expressed in the tool frame. Thus Fx, Fy, Fz are the components of the force at the end-effector, and nx, ny, nz are the components of the torque at the end-effector [SPO 05]. Let τ denote the vector of joint torques, and let δX represents a virtual endeffector displacement caused by the force F. Finally, let δq represents the corresponding virtual joint displacement. These virtual displacements are related through the manipulator Jacobian J (q) according to

$$\delta x = J(q)\delta q. \tag{4.21}$$

The virtual work δw of the system is

$$\delta w = F^T \delta X - \tau^T \delta q. \tag{4.22}$$

Substituting (4.21) into (4.22) yields

$$\delta w = (F^T J - \tau^T)\delta q \tag{4.23}$$

This is equal to zero if the manipulator is in equilibrium. Since the generalized coordinate q is independent, we have the equality

$$\tau = J(q)^T F. \tag{4.24}$$

In other words, the end-effector forces are related to the joint torques by the transpose of the manipulator Jacobian according to (4.24). Figure (4.4) shows the AL5B manipulator with a force.

$$F = \left(\mathrm{F_X},\ \mathrm{F_Y},\ \mathrm{F_z}, \mathrm{n_x},\ \mathrm{n_y},\ \mathrm{n_z}\right)^{\mathrm{T}} \tag{4.25}$$

applied at the end effector. The Jacobian of this manipulator is given by equation (4.10). The resulting joint torques $\tau = [\tau_1\ \tau_2\ \tau_3\ \tau_4\ \tau_5]^T$ is then given by equation (4.24). The transpose of J is:

$$J(q)^T = \begin{bmatrix} J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \end{bmatrix} \tag{4.26}$$

Substituting Equations (4.25) and (4.26) in Equation (4.24) gives

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ \tau_5 \end{bmatrix}_{(5x1)} = \begin{bmatrix} J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \end{bmatrix}_{(5x6)} * \begin{bmatrix} F_X \\ F_Y \\ F_z \\ n_x \\ n_y \\ n_z \end{bmatrix}_{(6x1)} \tag{4.27}$$

Using MATLAB, we can find the torque for each joint of the robot arm.

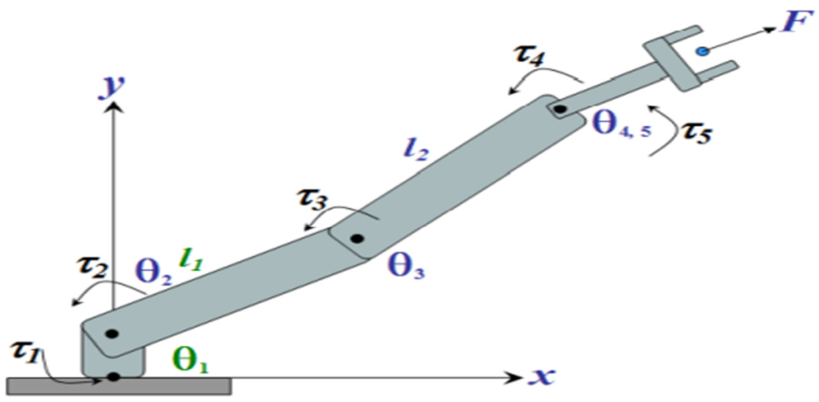


Figure (4.4): AL5B Robot Arm Torque Label

In conclusion this chapter presented the importance of the Jacobian Matrix in the robotics manipulators, this matrix is s defined to represent the mapping of velocities from joint space to Cartesian space. The Jacobian of a manipulator is also used for mapping of forces and torques. When the force and moment at the end-effector are given and the set of joint torques is required. In addition, the Jacobian is very helpful in understanding the singular configuration of the manipulator.

5.TRAJECTORY PLANNING

The main problem of this chapter is to find a trajectory that connects an initial to a final configuration while satisfying other specified constraints at the endpoints (e.g., velocity and/or acceleration constraints). Without loss of generality, we will consider planning the trajectory for a single joint, since the trajectories for the remaining joints will be created independently and in exactly the same way. Thus, we will concern ourselves with the problem of determining q (t), where q (t) is a scalar joint variable [SPO 05]. We suppose that at time t0 the joint variable satisfies

We suppose that at time t0 the joint variable satisfies

$$q(t_0) = q_0 \qquad (5.1)$$

$$\dot{q}(t_0) = v_0 \qquad (5.2)$$

and we wish to attain the values at t_f

$$q(t_f) = q_f \qquad (5.3)$$

$$\dot{q}(t_f) = v_f \qquad (5.4)$$

Figure (5.1) shows a suitable trajectory for this motion. In addition, we may wish to specify the constraints on initial and final accelerations. In this case, we have two additional equations

$$\ddot{q}(t_0) = \alpha_0 \qquad (5.5)$$

$$\ddot{q}(t_f) = \alpha_f \qquad (5.6)$$

The desired path is approximated by a class of polynomial functions. It generates a sequence of time-based “control set points” for the control of manipulator from the initial configuration to its destination. Figure (5.2) shows the trajectory planning block diagram.

5.1 Cubic Polynomial Trajectories

Suppose that we wish to generate a trajectory between two configurations, and that we wish to specify the start and end velocities for the trajectory. One way to generate a smooth curve such as that shown in Figure (5.1) is by a polynomial function of t. If we have four constraints to satisfy, such as (5.1)-(5.3), we require a polynomial with four independent coefficients that can be chosen to satisfy these constraints. Thus, we consider a cubic trajectory of the form

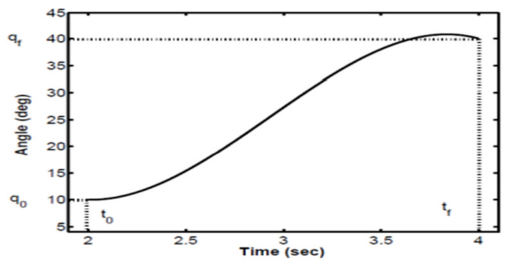


Figure (5.1): Typical Joint Space Trajectory

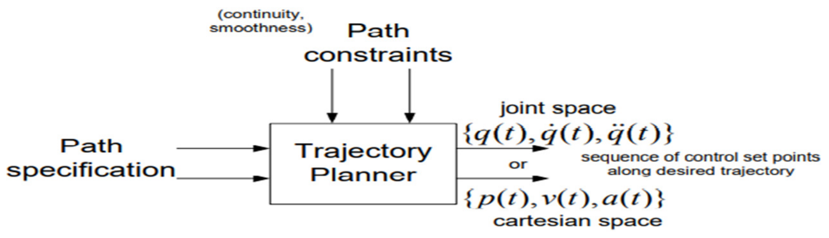


Figure (5.2): Trajectory Planning Block Diagram

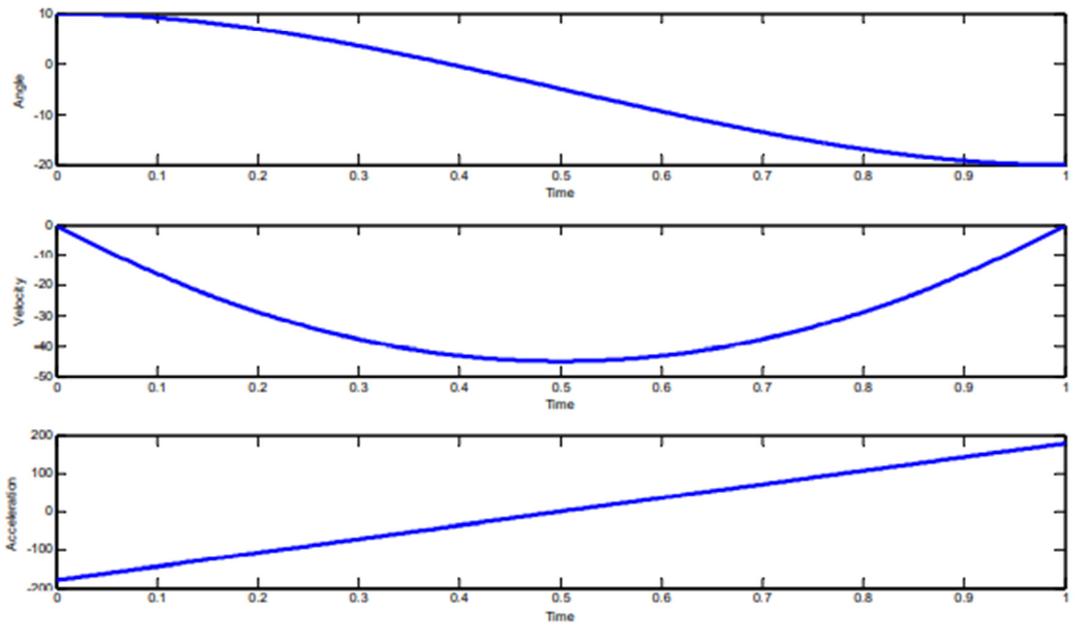


Figure (5.3): Cubic polynomial trajectory

5.2 Quantic Polynomial Trajectories

As can be seen in Figure (5.3), a cubic trajectory gives continuous positions and velocities at the start and finish points times but discontinuities in the acceleration. The derivative of acceleration is called the jerk. A discontinuity in acceleration leads to an impulsive jerk, which may excite vibration modes in the manipulator and reduce tracking accuracy.

For this reason, one may wish to specify constraints on the acceleration as well as on the position and velocity. In this case, we have six constraints (one each for initial and final configurations, initial and final velocities, and initial and final accelerations). Therefore, we require a fifth order polynomial

$$\begin{aligned} q(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 && \textit{For Distance} \\ \dot{q}(t) &= a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 && \textit{For Velocity} \\ \ddot{q}(t) &= 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 && \textit{For Acceleration} \end{aligned} \tag{5.15}$$

Using equations (5.1) - (5.6) and taking the appropriate number of derivatives, we obtain the following equations,

$$q_0 = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + a_4 t_0^4 + a_5 t_0^5 \tag{5.16}$$
$$v_0 = a_1 + 2a_2 t_0 + 3a_3 t_0^2 + 4a_4 t_0^3 + 5a_5 t_0^4 \tag{5.17}$$
$$\alpha_0 = 2a_2 + 6a_3 t_0 + 12a_4 t_0^2 + 20a_5 t_0^3 \tag{5.18}$$
$$q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \tag{5.19}$$
$$v_f = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \tag{5.20}$$
$$\alpha_f = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 \tag{5.21}$$

This we can written as

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^3 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 0 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} q_0 \\ v_0 \\ \alpha_0 \\ q_f \\ v_f \\ \alpha_f \end{bmatrix} \tag{5.22}$$

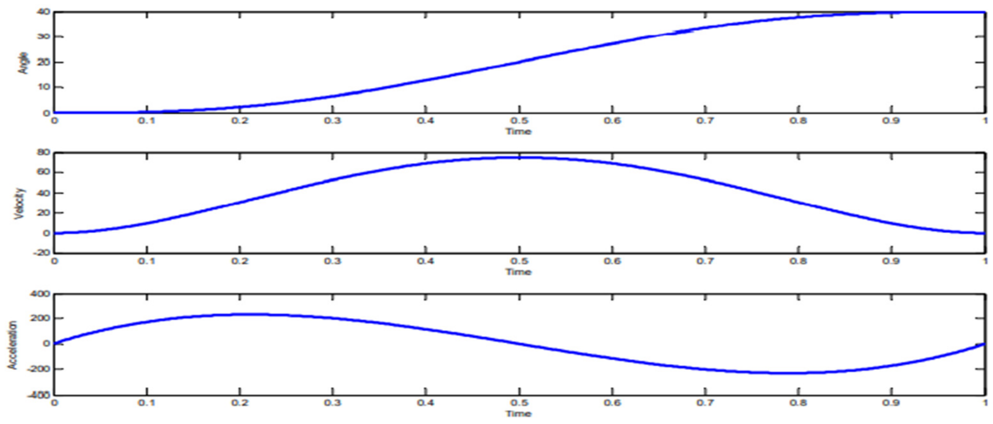


Figure (5.4): Quintic Polynomial Trajectory

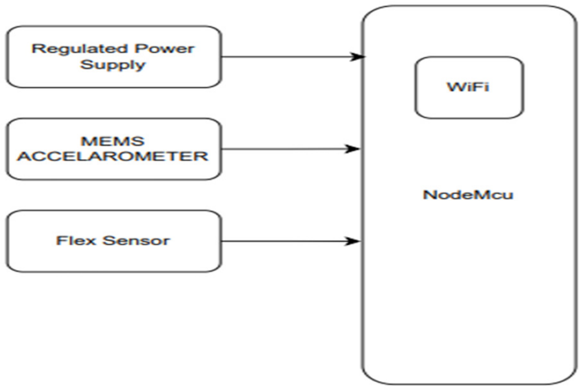
Figure (5.4) shows a quintic polynomial trajectory with q(0) = 0, q(2) = 40 with zero initial and final velocities and accelerations

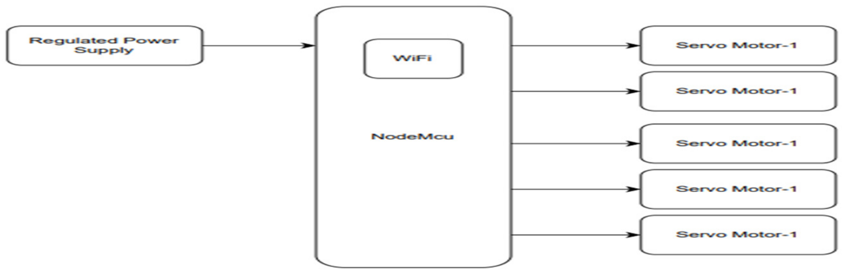
.

6. ROBOT HARDWARE AND SOFTWARE

To achieve a control of a robot arm by using personal computer, we must make the connection between the robot and PC. This connection is called interface connection and it is done by using a microcontroller. Microcontrollers are inexpensive devices commonly used in embedded computing applications to impart computing and smart decision-making capabilities to machines, products, and processes. Microcontrollers are designed to interface to and interact with electrical/electronic devices, sensors and actuators, and high-tech gadgets to automate systems. Microcontrollers are generally embedded directly into the product or process for automated decision-making. They are not meant to interface with human beings; however, microcontrollers do not have graphical user interface (GUI) capabilities that are common in many personal computer (PC) applications. The complete control process can be divided into two categories, hardware and software. Now we will discuss individually each one of these categories.

6.1 Hardware Environment





7.CONCLUSIONS

This report presented the development of educational software package using MATLAB/Simulink and 3D model. AL5B robot arm was modeled in this research. The complete software life cycle was implemented and validated. A complete mathematical model of AL5B robot is developed including complete Kinematics analyses of the AL5B robot arm. Forward and inverse kinematics equations were derived using Denavit-Hartenberg notation. Velocity Kinematic "Jacobian", and Trajectory Planning solutions were generated and implemented by the developed software. Simulation studies were performed by using MATLAB software's. By using 3D graphics program, structure for the AL5Brobot was built which enable the researchers to investigate robot parameters using both forward and inverse kinematics and in turn, this was facilitated the process of designing, constructing and inspecting on the robots in the real world.

8. REFERENCES

1. Prabhu, R. Raja, and R. Sreevidya. "Design of Robotic Arm based on Hand Gesture Control System using Wireless Sensor Networks." International Research Journal of Engineering and Technology 4.3 (2017).
2. Sun, Dong, et al. "A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations." IEEE Transactions on Robotics 25.5 (2009): 1074-1086.
3. Hughes, John M. Arduino: a technical reference: a handbook for technicians, engineers, and makers. " O'Reilly Media, Inc.", 2016.
4. Vujović, Vladimir, and Mirjana Maksimović. "Raspberry Pi as a Wireless Sensor node: Performances and constraints." 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, 2014.
5. Chen, Haisheng, et al. "Progress in electrical energy storage system: A critical review." Progress in natural science 19.3 (2009): 291-312.
6. SAHNI, GURUDUTT. "Micro electro mechanical systems (MEMS)." (2000).
7. Balavalad, Kirankumar B., and B. G. Sheeparamatti. "A critical review of MEMS capacitive pressure sensors." Sensors & Transducers 187.4 (2015): 120.
8. Kraft, Michael. "Micromachined inertial sensors: The state-of-the-art and a look into the future." Measurement and Control 33.6 (2000): 164- 168.
9. Sniegowski, Jeffry J. Multi-level polysilicon surface-micromachining technology: applications and issues. No. SAND-96-1986C; CONF-961105-8. Sandia National Labs., Albuquerque, NM (United States), 1996.
10. Sugana, Ms C. Arul, Mrs S. Arokia Magdaline, and Mrs Dr G. Gandhimathi. "Virtual Reality Implementation in Real XD Based Environment." (2018).
11. Lee, Boon Giin, and Su Min Lee. "Smart wearable hand device for sign language interpretation system with sensors fusion." IEEE Sensors Journal 18.3 (2017): 1224-1232.
12. Data-sheet, J. N. D. S. "JN5148-001." IEEE 802.15.
13. Rafael Rey, Jose Antonio Cobano et al. "A novel robot co-worker system for paint factories without the need of existing robotic infrastructure." Robotics and Computer-Integrated Manufacturing 70 (2021): 102122.
14. Abhinav Malviya, Rahul Kala. "Social robot motion planning using contextual distances observed from 3D human motion tracking." Applications 184 (2021): 115515.
15. Saeid Alirezazadeh et al. "Optimal algorithm allocation for robotic network cloud systems." Robotics and Autonomous Systems, May 2022, Robotic Networks.
16. GongfaLi FanXiao. An inverse kinematics method for robots after geometric
17. parameters compensation. Mechanism and machine theory volume 174,AUGUST 2022.
18. Fan Xiao, Gongfa Li et al. "An effective and unified method to derive the inverse kinematics formulas of general six-DOF manipulator with simple geometry." Mechanism and Machine Theory, May 2021, 104265.