A Thesis report on

INVERSE KINEMATIC STUDY OF SPATIAL SERIAL MANIPULATORS USING ANFIS APPROACH

Submitted in partial fulfillment of the requirement for the award of degree of

MASTER OF ENGINEERING

IN

CAD/CAM ENGINEERING

Submitted by

Jyotindra Narayan

Roll No. 801584013

Under the guidance of

Dr. Ashish Singla
Assistant Professor
Department of Mechanical Engineering
Thapar University, Patiala

DEPARTMENT OF MECHANICAL ENGINEERING

Thapar University

Patiala-147004, India

July, 2017

DECLARATION

I hereby declare that work done in this thesis entitled, **"INVERSE KINEMATIC STUDY OF SPATIAL SERIAL MANIPULATORS USING ANFIS APPROACH"** submitted towards partial fulfilment of requirement for the award of degree of **Master of Engineering** in **CAD/CAM Engineering** at **Thapar University**, **Patiala**, is an authentic record of work carried out by me under the supervision of **Dr. Ashish Singla** (Assistant Professor, Mechanical Engineering Department, Thapar University, Patiala.)

Date: June 30, 2017

Jyotindra Narayan
Roll No. 801584013

This is to certify that above declaration made by the student is correct to the best of my knowledge and belief.

Dr. Ashish Singla
Assistant Professor
Mechanical Engineering Department
Thapar University, Patiala.

ACKNOWLEDGEMENT

Through this, I would like to acknowledge all those great personalities who have an impactful influence on my professional life. It all started, when I was studying in last year of Bachelors, I got the drive to proceed with higher studies in the field of robotics from **Late Dr. O P Kaushal**. I was greatly inspired by his level of knowledge and decided to do research in the domain of Robotics.

Fortunately, in July 2014, I got admission in the CAD/CAM Engineering in one of the most prestigious university (Thapar University, Patiala). After six months, I heard the name of a great personality, named **Dr. Ashish Singla**, having excellence in the research area of Robotics and got the golden opportunity to work under his supervision.

Now, I would like to express my profound exuberance and gratitude to my mentor **Dr. Ashish Singla** (Assistant Professor, Mechanical engineering Department, Thapar University, Patiala) for his amiable guidance, constructive propositions and overwhelming inspiration in the nurturing work. It has been a blessing for me to spend many opportune moments under his guidance. The present work is little tribute to his activity, inspiration and ardent personal interest, taken by him during the complete research work.

Now, it's my pleasure to acknowledge the friends who always support me throughout the research work, especially, **Mr. Amardeep Singh, Mr. Ishan Chawla** and **Mr. Sumit Mehta**. Also, I am deeply indebted to my family for their love and encouragement.

Jyotindra Narayan

Roll No. 801584013

ABSTRACT

Inverse Kinematics (IK) is defined as the problem of determining a set of appropriate joint configurations for which the end effectors move to desired positions as smoothly, rapidly, and as accurately as possible. During the last decades, several methods and techniques, sophisticated or heuristic, such as Algebraic Method, Geometrical Method, Numerical Approach, Redundancy Resolution based concepts have been presented to produce fast and realistic solutions to the IK problem. However, most of the currently available methods suffer from high computational cost and production of unrealistic poses. This report reviews the most popular IK methods regarding their application functionality. Along with that, it enlightens the ongoing recent approach Adaptive Neuro-Fuzzy Logic Intelligent System (ANFIS), which is a combined effort of Neural Networks and Fuzzy logic. This approach ensures minimum drift to the values of Joint variables.

In this report, a graphical comparison is drawn between the analytical solutions and ANFIS one. Also, desired path is planned by taking an example of SCARA manipulator having graphical comparison using MATLAB. It is noticed that finding joint variables through ANFIS is one of the most promising option, consequently for path planning too. Interestingly, it is found that without even finding the inverse kinematic equations for higher degree of freedom manipulators, the acceptable joint parameters can be obtained using ANFIS technique. Furthermore, the path planning for 5-DOFs Spatial Medical manipulator is presented for a certain path. Along with this, percentage error between desired value and ANFIS predicted value for *x*-, *y*- and *z*- coordinates is calculated. Thereafter, for a certain arm matrix corresponding to home position and orientation of spatial medical manipulator, a comparative analysis of inverse kinematic solution is explained. Finally, in the end of the thesis, obstacle avoidance using 3-DOFs manipulator with three obstacles is presented to show the effectiveness of proposed ANFIS approach.

CONTENTS

LIST OF ABBRIVATIONS

FK / F Kin	Forward Kinematics
IK / I Kin	Inverse Kinematics
DOF	Degree of Freedom
SVD	Singular Value Decomposition
FLOPS	Floating Point Operations Per Second
ANFIS	Adaptive Neuro-Fuzzy Logic Intelligent System
MF	Membership Function
3D	Three Dimensional
CAD	Computer Aided Design
DH	Denavit Hartenberg
MDH	Modified Denavit Hartenberg
R-R-P-R	Revolute Revolute Prismatic Revolute
EE	End Effecter
SCARA	Selective Compliance Assembly Robot Arm
MATLAB	Mathematical Laboratory

LIST OF FIGURES

LIST OF SYMBOLS

θ_i, θ'_i = joint angle parameter

x_e, y_e = end effector coordinates

ϕ_e = cumulative angle measured from base to end effector

x_w, y_w = wrist coordinates

l_i = link length

α, β, γ = included joint angle

C = cosine angle

S = sine angle

θ_{i-j-k} = difference of angles for index I j k

α_i = link twist parameter

a_i = link length parameter

d_i, b_i = joint offset parameter

w = tool configuration vector

w_i = tool configuration elements

R_{ij} = elements of orientation matrix

q = joint variables

n = number of joints

k = number of end effectors

R^3 = cartesian space

s_k = position vector with respect to origin

t = target vector

e_i = change in the position of *i-th* end effector

s_d = desired configuration vector

j = Jacobian Matrix

v_j = unit vector pointing along the current axis of rotation for the joint.

J^{-1} = inverse of a Jacobian

J^t = transpose of a Jacobian

J^+ = pseudo inverse

$U\ and\ V$ = orthogonal matrices

D = diagonal matrix

H = joint availability criterion

∇H = gradient of joint availability criterion

λ = Lagrange Multiplier

x_i = unit vector along x-axis

y_i = unit vector along y-axis

z_i = unit vector along z-axis (joint axis)

Oij = node function Oij (for $i\text{-}th$ position of $j\text{-}th$ layer)

r_{Ai}, r_{Bi}, r_{Ci} = membership function for particular input

a = curve width in membership function

c = distance from the origin in membership function

${}^{0}_{i}A$ = transformation matrix

$m, n, o,$ = first row elements of rotation matrix

p, q, r = second row elements of rotation matrix

s, t, u = third row elements of rotation matrix

m = link masses

i = number of links

$\dot{\theta}_i$ = initial joint angle velocity

O_i = obstacle

LIST OF TABLES

LIST OF PUBLICATIONS

1. Narayan, J., Ashish Singla, **"ANFIS based kinematic analysis of 4-DOFs SCARA robot"**, in International Conference on Signal Processing, Computing and Control, ISPCC, 2017. (Communicated on 30th May, 2017)

Chapter 1

Introduction

1.1 Kinematics of Robots

Forward Kinematics (FK) can be defined as the problem of determining the end effector's positions after applying known transformations to the chain, given the joint variables are known.

Inverse kinematics (IK) refers to the use of the kinematics equations of a robot to determine the joint parameters that provide a desired position of the end-effector. Specification of the movement of a robot so that its end-effector achieves a desired task is known as motion planning. Inverse kinematics transforms the motion plan into joint actuator trajectories for the robot.

The movement of a kinematic chain whether it is a robot or an animated character is modeled by the kinematics equations of the chain. These equations define the configuration of the chain in terms of its joint parameters. Forward kinematics uses the joint parameters to compute the configuration of the chain, and inverse kinematics reverses this calculation to determine the joint parameters that achieves a desired configuration.

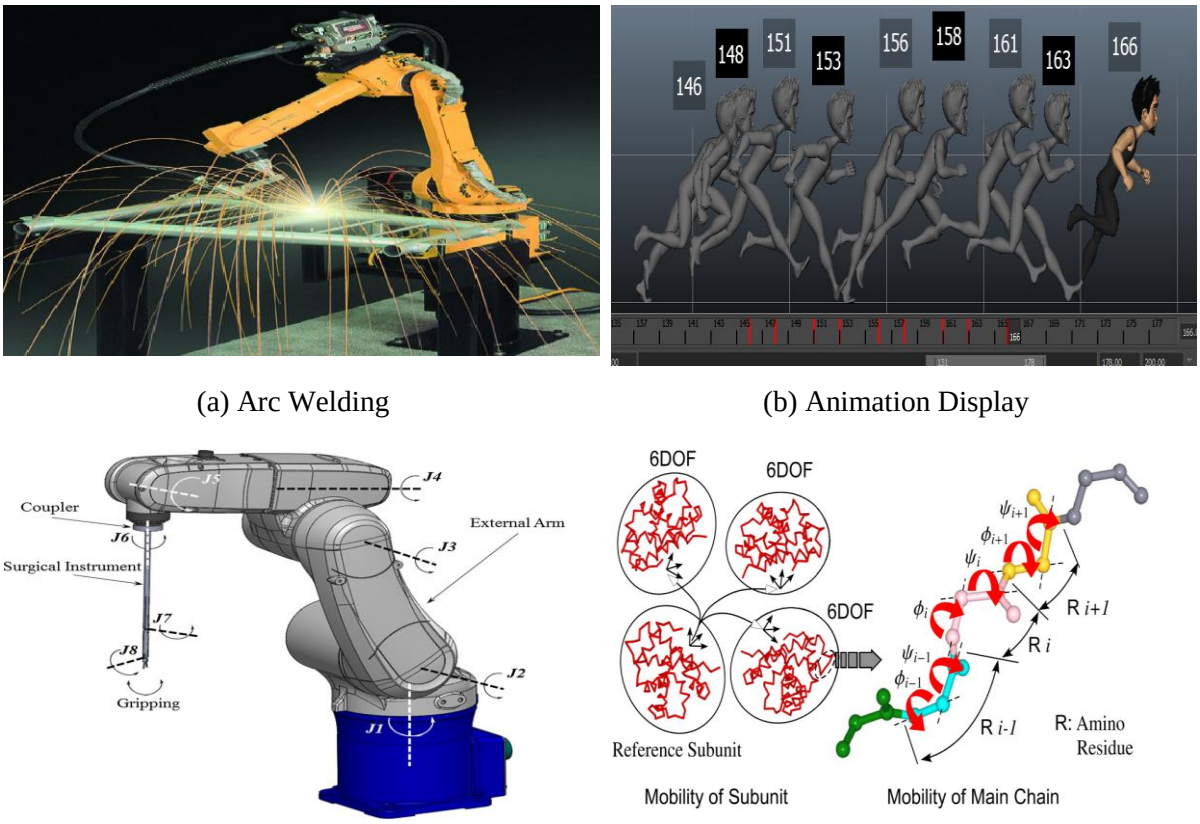
For example, inverse kinematics formulas allow calculation of the joint parameters that position a robot arm to pick up a part. Similar formulas determine the positions of the skeleton of an animated character that is to move in a certain way.

Kinematic analysis allows the designer to obtain information on the position of each component within the mechanical system. This information is necessary for subsequent dynamic analysis along with control paths.

Inverse kinematics is an example of the kinematic analysis of a constrained system of rigid bodies, or kinematic chain. The kinematic equations of a robot can be used to define the loop equations of a complex articulated system. These loop equations are non-linear constraints on the configuration parameters of the system. The independent parameters in these equations are known as the degrees of freedom of the system.

1.2 Need and Motivation

With contrast to the past, nowadays, robots perform a vital role in many industrial applications as pick and place operation, different assembly operations like welding, soldering, painting etc. [1]. In medical assisted surgery, importance of robot's end effector's position with respect to joint variables is appreciable for better accuracy and precision; serves as utmost part in development of medical improvements [2]. The need for accurate biomechanical modeling and body sizing based on anthropometric data make IK methods a popular approach for fast and reliable solution. IK has been used in rehabilitation medicine to observe asymmetries or abnormalities. As, Inverse Kinematics (IK) is a method for computing the posture via estimating each individual degree of freedom to satisfy a given task; it plays an important role in the computer animation and simulation of articulated figures or control different virtual creatures [3]. They are also very popular in the video games industry. Also, the protein position in the metabolism of human being can also be determined with same concepts of kinematics of robots considering structure of a protein is moreover similar to combination of different linkages as in a robot [4]. Moreover, the author's interest in serving the society by providing robotic surgery at effective cost cutting is one of the important point of motivation.



(a) Arc Welding (b) Animation Display

(c) Surgical Robot (d) Protein Structure

Fig. 1.1: Applications of Inverse Kinematics [W1][W2][W3][W4]

1.3 Scope of the Thesis

Kinematics of robots affirms crucial role for recent advancements within industrial sectors and numerous medical assessments. Finding forward kinematics, using DH convention is easy task. But obtaining inverse kinematic solutions is one of the tedious and complex task as degree of freedom increases consecutively. As degree of freedom increases, traditional techniques for inference of inverse kinematic solutions become insufficient because of unavailability of unique solution; leads to highly time consuming approach. All these issues can be resolved by applying the concept of fuzzy logic and neural networks as a conjugate, having qualitative feature of fast computation.

Fig. 1.2: Real-time demonstration of "da Vinci surgical system" [W5]

Figure 1.2, demonstrates the real time application of a robotic assisted surgical system which is "da Vinci surgical system". By using this system, most of the surgeries performed in conventional manner can be performed in minimal invasive manner. Path following or trajectory tracking of robots is advantageous within industrial sectors and several medical practices. For example, many pick and place operations based work by robots within industry follow a certain path and it's not easy to change the coordinate system every time whenever the desired path is changed. Same problem comes into picture in the case of robot assisted surgery. Thus, the best solution is to train the robot for all the possible workspace in which they are desired to move. After that, whenever the coordinate system will change, they automatically update their joint variables and consequently tracking desired trajectory is done. This training concept is only viable with the help of neural networks alone or the combinations of neural

networks with fuzzy logic. This led to the development of alternate technique using fuzzy inference system and neural network approach, named ANFIS.

In this thesis, path planning for different degree of freedom surgical manipulator has been presented with the help of ANFIS approach. First, inverse kinematic solutions for these surgical manipulators have been estimated using ANFIS approach.

1.4 Organization of the Thesis

The organization of the thesis is as given below:

In Chapter 2, literature on different methodologies, dealing with inverse kinematic study of different classes of robotic manipulators, has been reviewed. In this review, the general procedures of each method, describing the structures of manipulator, is presented. Also, an attempt has been made to present these methods with their descriptive formulations, in the sense of their development.

In Chapter 3, application of ANFIS technique on 4-DOFs SCARA manipulator has been presented for finding inverse kinematic solutions. Also, the desired path planning has been done by 4-DOFs SCARA manipulator using ANFIS approach. Furthermore, the comparative analysis has been done between the solutions obtained from ANFIS one and the analytical one.

In Chapter 4, path planning of 5-DOFs P-R-R-R-R spatial medical manipulator has been presented for desired path, using ANFIS approach, for a fixed orientation. For solving the purpose of same, formulation of forward kinematic equations has been used as input and output parameters for the training dataset in ANFIS.

In Chapter 5, obstacle avoidance by 3-DOFs manipulator has been presented using ANFIS approach in case of three obstacles, two active at the same time and all of three active at the same time, showing the application of redundant manipulators in narrow channels.

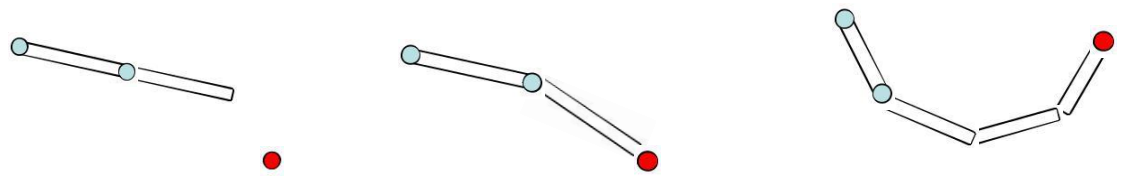
In Chapter 6, the major findings of this thesis are reported. It also highlights of future directions of present work.

Chapter 2

Literature Review

2.1 Introduction

During recent decades, a variety of methods have been anticipated for solving IK problems. However, it is quite significant for a comprehensive IK solver to manipulate the joint configurations, according to joint type. Moreover, several end effectors and multiple target models are to be dealt with sequential and non-sequential operations. Managing complicated figures by sequencing the work one by one is not the practical approach. Therefore, there is needful requirement of suitable resolution approach to control the multiple task criterion.



(a) (b) (c)

Fig. 2.1: Feasible solutions of the IK problem: (a) The target is not approachable; the case where linked structure is not able to approach the target, (b) One solution; the certain cases with only one solution, (c) Many solution; most of the cases where multiple solution exists for IK problem.

Solving the FK problem for a unique solution completely depends on the fact that the joints can do the desired transformation or not. In contrast, solving the IK does not always guarantee the solution. There are several cases where the goal is not approachable or conflict of two and more tasks lead to the non-availability of IK solution. Unapproachable targets are those, whose reachability can be farther than the chain's end effector reach or can be at a point where no pivoting of links can bend the chain to reach (see Figure 2.1). These problems are termed as *over-constrained* problems. Also, so many cases exist in which multiple solution exist and solely depends on the IK solver to find the optimized solution. The ranking of IK solver is determined by estimation of most realistic solution and computational cost of estimating that solution. Various methods for finding the inverse kinematic solution are explained in next section.

2.2 Different methods of Inverse Kinematics:

2.2.1 Geometrical Method

Consider the three DOFs as shown below in Figure 2.2-(a). To solve its inverse kinematics, the kinematics diagram is drawn again in Figure 2.2-(b). The problem is to find three joint angles $\theta_1, \theta_2, \theta_3$ who lead the end effecter to a desired position and orientation, $x_e, y_e, \emptyset_e$.

Using two-step approach [5] firstly, the position of the wrist, point B, from $x_e, y_e, \emptyset_e$ is found. Then the θ_1, θ_2 from the wrist position is found. Angle θ_3 can be immediately calculated from the wrist position. Let x_w and y_w be the coordinates of the wrist and l_1, l_2, l_3 are the link lengths.

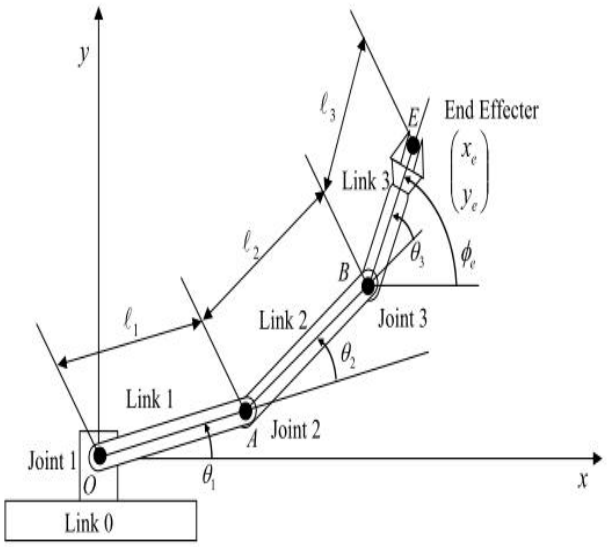


Fig. 2.2-(a): Links and Joints Arrangement of 3R Robot [5]

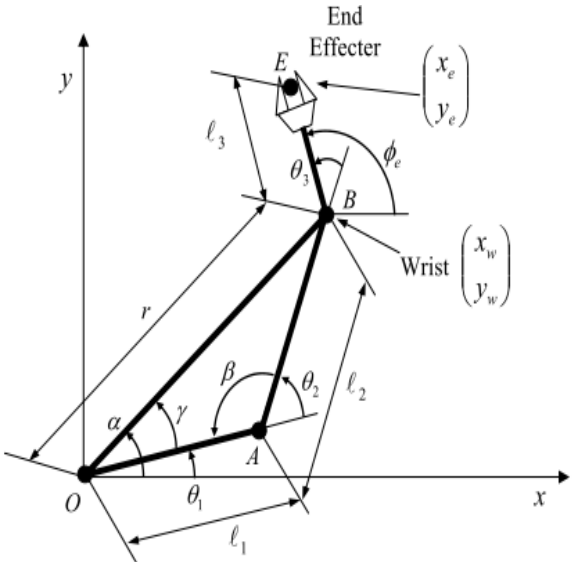


Fig. 2.2-(b): Line Diagram Arrangement of 3R Robot [5]

After solving for geometrical entities, the joint variables are:

$$\theta_1 = \alpha - \gamma = \tan^{-1}\frac{y_w}{x_w} - \cos^{-1}\frac{x_w^2 + y_w^2 + \ell_1^2 - \ell_2^2}{2\ell_1\sqrt{x_w^2 + y_w^2}} \tag{2.1}$$

$$\theta_2 = \pi - \beta = \pi - \cos^{-1}\frac{\ell_1^2 + \ell_2^2 - x_w^2 - y_w^2}{2\ell_1\ell_2} \tag{2.2}$$

and from the above θ_1, θ_2, the joint variable θ_3 can obtain

$$\theta_3 = \phi_e - \theta_1 - \theta_2 \tag{2.3}$$

Interestingly, there exists an alternative way of reaching the same end-effecter position and orientation, which is another solution to the IK problem. Figure 2.2-(c) displays two configurations of the arm that lead to the same end-effecter location: the elbow down & the elbow up configurations. First one depicts the solution obtained above. In respect of line OB, the second one with position of the elbow at point A', is symmetric to the first configuration, as depicted in the Figure 2.2-(c). Hence, the relation between both the solutions can be expressed as

$$\begin{aligned}\theta_1' &= \theta_1 + 2\gamma \\ \theta_2' &= -\theta_2 \\ \theta_3' &= \phi_e - \theta_1' - \theta_2' = \theta_3 + 2\theta_2 - 2\gamma\end{aligned} \tag{2.4}$$

Since, IK problems are non linear, they often have more than one solutions, as shown in above instance. The whole configuration of the system is not uniquely determined by the specified end-effecter position and orientation. Therefore, the collective position and orientation of the end-effecter (vector p), cannot be used as generalized coordinates.

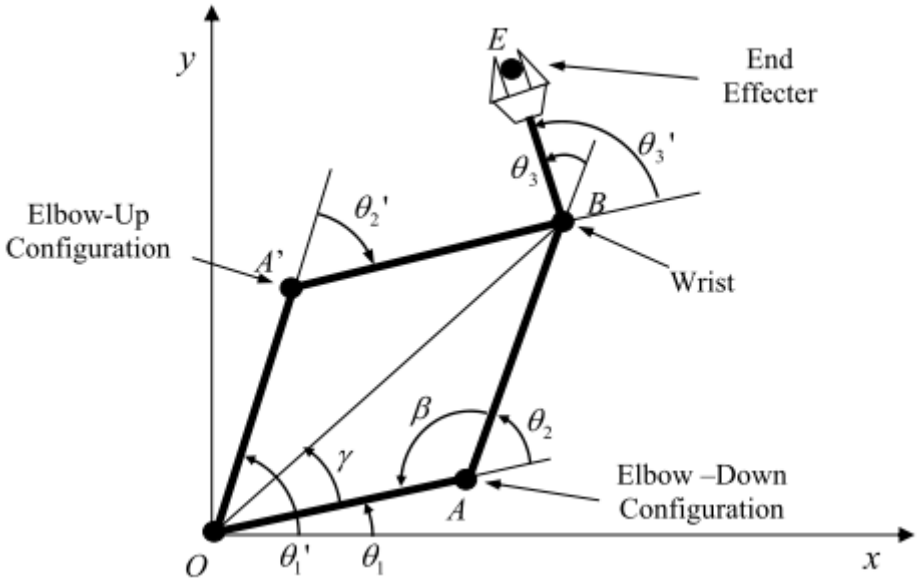


Fig. 2.2-(c): Multiple Solutions of 3R Robot [5]

Nevertheless, an extra degree of flexibility is provided to the robot by the existence of numerous solutions. In case of existence of more than one configuration for the same end-effecter location, a robot, working in a crowded environment, can choose a configuration

evading any interference with the environment. However, the solutions to the IK problem do not always give feasible configurations because of physical limitations. It must be checked that each solution satisfies the constraint of movable range (i.e. stroke limit of each joint) or not.

Constraints

This method is only applicable to a special set of kinematics. If the kinematics have higher DOF, then the solutions of first three joint variables should be known in prior for finding the rest of solutions.

2.2.2 Algebraic Method

In this, tool configuration vector and Arm matrix for the robot are used for finding the inverse kinematics. For example, the arm matrix and tool configuration vector for the SCARA robot [6] are given as:

Arm Matrix:

$$K_{base}^{tool} = \begin{bmatrix} C\theta_{1-2-4} & S\theta_{1-2-4} & 0 & a_1C\theta_1 + a_2C\theta_{1-2} \\ S\theta_{1-2-4} & -C\theta_{1-2-4} & 0 & a_1S\theta_1 + a_2S\theta_{1-2} \\ 0 & 0 & -1 & d_1 - d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

Tool Configuration vector:

$$\mathrm{w}\ (\theta) = \begin{bmatrix} a_1C\theta_1 + a_2C\theta_{1-2} \\ a_1S\theta_1 + a_2S\theta_{1-2} \\ d_1 - d_3 - d_4 \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ 0 \\ 0 \\ -\exp(\theta_4/\pi) \end{bmatrix} \tag{2.6}$$

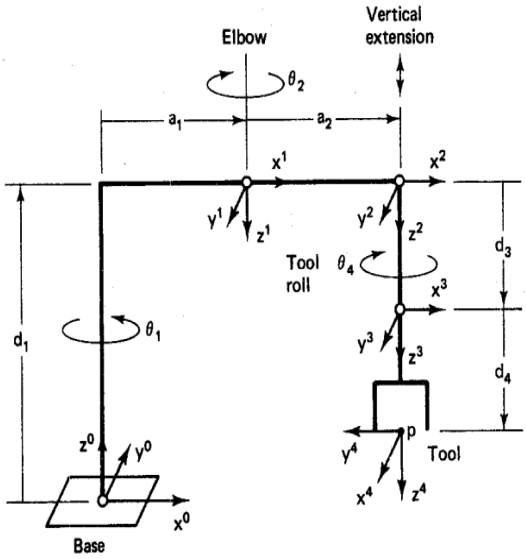


Fig. 2.3-(a): Line Diagram for SCARA Robot [6]

Fig. 2.3-(b): CAD Model of SCARA [W6]

Solving above two equations, the joint variables are:

Elbow Joint:

$$\theta_2 = \pm arccos\left(\frac{w_1^2+w_2^2-a_1^2-a_2^2}{2a_1a_2}\right) \tag{2.7}$$

Base Joint:

$$\theta_1 = arctan2[a_2S\theta_2w_1 + (a_1 + a_2C\theta_2)w_2, (a_1 + a_2C\theta_2)w_1 - a_2S\theta_2w_2] \tag{2.8}$$

Vertical extension Joint:

$$d_3 = d_1 - d_4 - w_6 \tag{2.9}$$

Tool Roll Joint:

$$\theta_4 = \pi ln|w_6| \tag{2.10.1}$$

An alternative way to find tool roll angle,

$$\theta_4 = \theta_1 - \theta_2 - \theta_{1-2-4} \text{ where } \theta_{1-2-4} = arctan2(R_{21}, R_{11}) \tag{2.10.2}$$

Constraints

According to this Method [6], there are 12 constraints implicit in the arm equation which represent only 6 independent constraints on the 'n' unknown components of the vector of joint variables 'q'. If there is a general solution to the inverse kinematics problem, one for which 'q' can be found which generates an arbitrary tool configuration, then the number of unknowns must at least match the number of independent constraints. That is:

General Manipulation $\rightarrow n \geq 6$

This lower bound on the number of axes n is a necessary but not sufficient condition for the existence of a solution to the IK problem when arbitrary tool configurations are specified. Clearly, the tool position must be within the work envelope of robot, and the tool orientation must be such that none of the limits on the joint variables are violated. Even when these additional constraints 'p' and 'R' are satisfied, there is no guarantee that a closed-form expression for a solution to the IK problem can be obtained.

2.2.3 Numerical Approach

Let the complete joint configuration of the multibody be specified by the scalars $\theta_1, ..., \theta_n$, if there are n joints and each θ_j value is called a joint angle (joint configuration may not always be an angle), where θ_j is the angle in the plane of rotation assuming we also have knowledge of the rotation axis. Certain points on the links are identified as end effectors. To solve the IK problem, the joint angles must be settled so that the resulting configuration of the multibody places each end effector at, or as close as possible to, its target position. If there are k end effectors, let their positions be denoted as $s_1, ..., s_k$ relative to a fixed origin. Each end effector position s_i is a function of the joint angles. The column vector $(s_1, s_2, ..., s_k)^T$ can be written as $\vec{s}$; this can be viewed as a column vector either with m = 3k scalar entries or with k entries from R^3. One way to control the multibody is to specify target positions, one for each end effector. The target positions are also defined by a vector $\vec{t} = (t_1, t_2, ..., t_k)^T$, where t_i is the target position for the i-th end effector. Let $e_i = t_i - s_i$, be the desired change in position of the i-th end effector (moving to the desired i-th target). This Equation [7] can be rewritten as $\vec{e} = \vec{t} - \vec{s}$.The joint angles are also written as a column vector $\theta = (\theta_1, ..., \theta_n)^T$. The end effector positions are functions of the joint angles; this fact can be expressed as

$$\vec{s} = f(\theta) \tag{2.11}$$

or, for i = 1, ..., k, $\vec{s}_i = f_i(\theta)$. This is called the Forward Kinematics (FK) solution.

The aim of Inverse Kinematics is finding a vector θ so as $\vec{s}$ is equal to a given desired configuration $\vec{s}_d$:

$$\theta = f^{-1}(\vec{s}_d) \tag{2.12}$$

where, 'f' being highly non-linear operator and difficult to invert.

Nevertheless, iterative methods may be used to deduce a good approximate solution. This involves the linear approximation of the functions 's_i' using the Jacobian matrix. The Jacobian matrix, J which is a function of the θ values, is expressed as

$$J(\theta)_{i,j} = \left(\frac{\partial s_i}{\partial \theta_j}\right)_{i,j} \tag{2.13}$$

A method to calculate the entries of the Jacobian matrix for different representations of joints and multi bodies was discussed by Orin and Schrader [8]. Entries of the Jacobian matrix for the j-th rotational joint can be deduced using following relation:

$$\left(\frac{\partial s_i}{\partial \theta_j}\right)_{i,j} = v_j \times (s_i - p_j) \tag{2.14}$$

where, p_j being position of the joint, and v_j being the unit vector pointing along the current axis of rotation for the joint. If the ith end effector is not affected by the j_{th} joint, then of course $\partial s_i / \partial \theta_j = 0$. If the j_{th} joint is translational, the entry in the Jacobian matrix is even easier to compute. If the i_{th} end effector is affected by the j_{th} joint,

$$\left(\frac{\partial s_i}{\partial \theta_j}\right)_{i,j} = v_j \tag{2.15}$$

It may be noted that J can be viewed either as a k × n matrix whose entries are vectors from R^3, or as m × n matrix with scalar entries (with m = 3k).

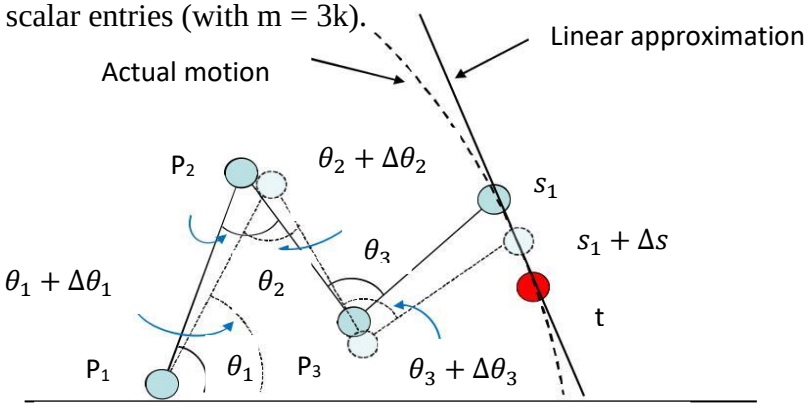


Fig. 2.4: Basis of Jacobian Solution [8]

So, Equation 2.11 for forward dynamics can also be expressed as follows:

$$\dot{\vec{s}} = J(\theta)\dot{\theta} \tag{2.16}$$

where, the dot notation depicts the first derivative w.r.t. time. The Jacobian J = J(θ) can be deduced using the current values θ, $\vec{s}$ and $\vec{t}$. We then seek an update value Δθ for incrementing the joint angles θ by Δθ:

$$\theta := \theta + \Delta\theta \tag{2.17}$$

The change in end effector positions caused by this change in joint angles can be estimated as

$$\Delta\,\vec{s} \approx J\Delta\theta \tag{2.18}$$

Such value of the Δθ is to be selected so that $\Delta\vec{s}$ becomes approximately equal to $\vec{e}$. Also, sometimes Δθ is chosen in such a way so as to match the approximate movement $\Delta\vec{s}$ in the end effectors (partially) with the target positions velocities.

So, the FK problem can be expressed as $\vec{e}$ = JΔθ & the IK problem can be rewritten as $\Delta\theta = J^{-1}\vec{e}$. However, the IK equation cannot be solved uniquely, most of the times. Although, it is true that the Jacobian J may not be square or invertible, even then J might work poorly because of its near singularity despite being invertible. Multiple methods have been put forward to counter such problems. These approaches have been mentioned and discussed in the following pages.

2.2.3 (a) The Jacobian Transpose Method

As far as basic idea behind Jacobian transpose method [9, 10] is concerned, it is simply using transpose of J in place of the inverse of J. Means, Δ θ is equated as

$$\Delta\,\theta = \alpha J^{T}\vec{e}, \tag{2.19}$$

for some appropriate scalar α. Indeed, the transpose and the inverse of the Jacobian is not the same; even then, the use of the transpose can be justified in terms of virtual forces so as the change in end effector position be exactly as $\alpha JJ^{T}\vec{e}$, & here, α is chosen in such a way such that this value can be made as close as feasible to that of $\vec{e}$.

2.2.3 (b) The Pseudo-Inverse method with SVD

This method [11] sets the value of Δ θ as

$$\Delta\,\theta = J^{\dagger}\vec{e}, \tag{2.20}$$

where the n × m matrix $J^{\dagger}$ is the pseudo-inverse of J, also known as Moore-Penrose inverse of J. It is defined for all matrices J, irrespective of the fact that they are square or not or of full row rank or not. The pseudo inverse provides the most excellent feasible solution to the equation $J\Delta\,\theta = \vec{e}$ in the sense of least squares. Few finest properties of pseudo inverse are described below.

Let $\Delta\,\theta$ be defined by equation. Firstly, let us take that $\vec{e}$ is in the range (i.e., the column span) of J. For this scenario, $J\Delta\,\theta = \vec{e}$; wherein, $\Delta\,\theta$ is the unique vector of smallest magnitude satisfying $J\Delta\,\theta = \vec{e}$. Secondly, let us take that $\vec{e}$ is not in the range of J. For this scenario, $J\Delta\,\theta = \vec{e}$ becomes impossible. Nevertheless, as an inherent property, $\Delta\,\theta$ minimizes the magnitude of the difference $J\Delta\,\theta - \vec{e}$. Also, $\Delta\,\theta$ is the unique vector of smallest magnitude which minimizes $\| J\Delta\,\theta - \vec{e}\|$, or equivalently, which minimizes $\| J\Delta\,\theta - \vec{e}\|^2$.

The pseudo-inverse generally shows some stability issues in the neighborhoods of singularities. Because there is an unachievable direction of movement of the end effectors, the Jacobian matrix doesn't have full row rank at a singularity. The pseudo inverse will be well-behaved and will not show an impossible direction movement, if the configuration is at a singularity. In case of mere being in proximity of the singularity, the pseudo inverse method will present very large deflections in joint angles, even in case of small movements in the target position. Practically speaking, roundoff errors denotes that true singularities are rare to achieve and as an alternative, values must be checked for being near-zero to detect the singularity.

Another property of pseudo inverse is that the matrix $(I - J^{\dagger}J)$ performs a projection onto the null space [12] of J. Hence, for all vectors, $J(I - J^{\dagger}J)\,\varphi = 0$. So, $\Delta\,\theta$ can be set up using

$$\Delta\,\theta = J^{\dagger}\vec{e} + (I - J^{\dagger}J)\,\varphi \tag{2.21}$$

for any vector φ and even then, anyone can obtain a value for $\Delta\,\theta$ which tend to minimize the value of $J\Delta\,\theta - \vec{e}$.

In case of J having full row rank, JJ^{T} will be definitely invertible. For this scenario, the minimum magnitude solution $\Delta\,\theta$ will be in following form

$$\Delta\,\theta = J^{T}(J\,J^{T})^{-1}.\,\vec{e} \tag{2.22}$$

To use this equation, it is a must that J has a full row rank.

In spite of being ample coverage in literatures, pseudo inverse method generally provides poor results mainly due to instability in the neighborhoods of singularities. The (selectively) damped least squares methods provide much more accurate result.

To get improved analysis and visualization of the Pseudo-Inverse method, the *Singular Value Decomposition* of J can be used in which J can be expressed as [7, 13]

$$J = UDV^T \tag{2.23}$$

where U and V are orthogonal matrices and D is diagonal. If J is m × n, then U is m × m, D is m × n, and V is n × n. The only non-zero entries in the matrix D are the values $\sigma_i = d_{i,i}$ along the diagonal. We henceforth assume m ≤ n. Without loss of generality, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$.

Note that the values σi may be zero. In fact, the rank of J is equal to the largest value r such that $\sigma_r \neq 0$.

The pseudoinverse of J is equal to

$$J^{\dagger} = V D^{\dagger} U^T . \tag{2.24}$$

Thus,

$$J^{\dagger} = \sum_{i=1}^{r} \sigma_i^{-1} v_i u_i^T \tag{2.25}$$

The vectors $v_{r+1}, \ldots, v_n$ are an orthonormal basis for the null space of J .

2.2.4 Redundancy Resolution

The basis for many methods to resolve the redundancy is the Jacobian matrix which relates differential joint motion Δ θ to the resulting end-effector motion Δx. Suppose, there are 'n' joints and 'm' end effector degrees of freedom. If the system is non-redundant, then J is square with order n. Unless the system is at a kinematic singularity, where one or more end effector degrees of freedom cannot be achieved, the Jacobian will be full rank and Gaussian elimination can be used to solve for Δ θ. Define the degree of redundancy as r = n − m: For a redundant case, n > m and there are an infinite number of solutions. Whitney [11] used the pseudoinverse to find the minimum norm joint change as

$$\Delta \theta = J^{\dagger} \Delta x \tag{2.26}$$

However, in most applications, simple pseudo-inverse control is not satisfactory. Pseudo-inverse control is a local criterion, and if the manipulator end effector traces a closed path, the joint state will not generally return to the same state [14]**.** While in some cases, this joint drift will reach a limit, the poor repeatability is a serious problem for obstacle avoidance by inboard joints.

Many objectives to be achieved with a redundant manipulator can be expressed in terms of minimizing a criterion H. In this format, redundancy resolution is accomplished by moving the joints such that the end effector is moved in the desired way and the criterion H is always kept at a minimum. Thus, there is no repeatability problem. One such formulation was done by Liegeois who added a secondary term to pseudo-inverse control to project the negative gradient of H into the null space of J. Here, H is the Joint Availability Criterion.

$$\Delta\,\theta = \mathrm{J}^{\dagger}\Delta x - \alpha\,(\mathrm{I} - \mathrm{J}^{\dagger}\mathrm{J})\,\nabla H \tag{2.27}$$

This method has two weaknesses [12]. One is that the choice of the gain α is sensitive and configuration dependent; if α is too small, the system is sluggish and if α is too large, the system can go unstable. Even when the control is stable, the resulting lag implies that the joint is not a function of the end-effector position and therefore, the system is not strictly repeatable.

2.2.4 (a) Extended Jacobian Method

Baillieul [15] introduced a formulation that directly updates the joint position to cause the desired end-effector motion and forces ∇H to stay in the null spaceof J. Where η_i are a linearly independent basis of the null space such that

$$\begin{bmatrix} J \\ \dots \\ N\nabla(\nabla H) - W \end{bmatrix} \Delta\theta = \begin{bmatrix} \Delta x \\ \dots \\ \theta \end{bmatrix} \tag{2.28}$$

where, W is an r × n matrix whose element w_{ij} is given by

$$w_{ij} = (\nabla^T H) J^T (J\,J^T)^{-1} \frac{\partial J}{\partial \theta_j} \eta_i$$

2.2.4 (b) Lagrange Multiplier Method

Chang [16] has presented an alternative formulation to resolve redundancy by keeping H at a minimum. The first key point in his work is that the Lagrange multiplier technique can be used. If ∇H is not in the null space of J, then it must be in the row space of J, or

$$\nabla H = J^T \lambda \tag{2.29}$$

where, λ is the Lagrange multiplier.

Meeting with the end-effector motion requirements forms a system that Chang recommends solving numerically.

In comparing the number of FLOPS, the most compact version of the Lagrange multiplier method [17] comes out slightly ahead of the extended Jacobian method. However, the most compact version of the Lagrange multiplier method has additional singularities which makes its use a difficulty. One could use the faster version and switch to the slower, non-singular version only when necessary. However, for real-time control there is a strong advantage in using a method that has a predictable computation time independent of configuration.

2.2.5 Commercial Software (RoboAnalyzer)

A commercial software named RoboAnalyzer is a visualization based robotics software utilizing the Object Oriented Programming in Visual C# programming language, since 2009. OpenGL, an open-source library, has been implemented to visualize 3D models of robots through Tao Framework. It has been developed in such a way, that if there will be a change within the existing modules, rest will not be affected. Serial robots with revolute joint was reported for analyzing the Forward Kinematics (FKin) [18], which follows skeleton models for better description of robotic structures.

Animation of the robot motion were used for the analysis of results. The results of the analysis were also plotted as graphs using ZedGraph, an open-source plotting library in C#. Addition of the prismatic joints, Inverse Dynamics (IDyn) and Forward Dynamics (FDyn) analyses were reported in [19]. Modules on "Visualization of DH Parameters and Transformations", "3D CAD Model Importer" and "Inverse Kinematics" (IKin) were reported in [20]. The interactions of these modules are shown in Figure 2.6. The description of DH Parameters are shown in Figure 2.5, as given below:

Joint Offset (b_i) = Distance between X_i and X_{i+1} along Z_i

Joint Angle (θ_i) = Angle between X_i and X_{i+1} about Z_i

Link Length (a_i) = Distance between Z_i and Z_{i+1} along X_{i+1}

Twist Angle (α_i) = Angle between Z_i and Z_{i+1} about X_{i+1}

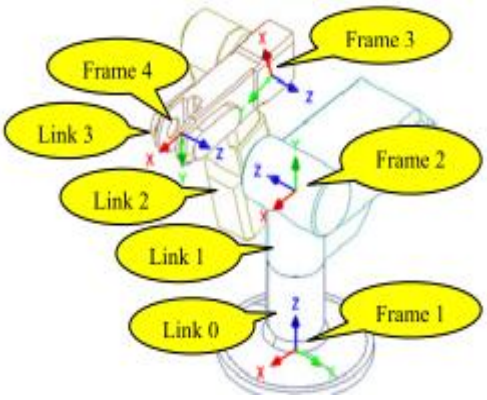


Fig. 2.5-(a): Visualization of DH Parameters [20]

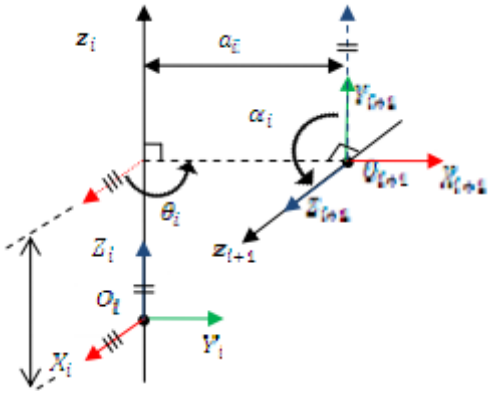


Fig. 2.5-(b): DH Representation [20]

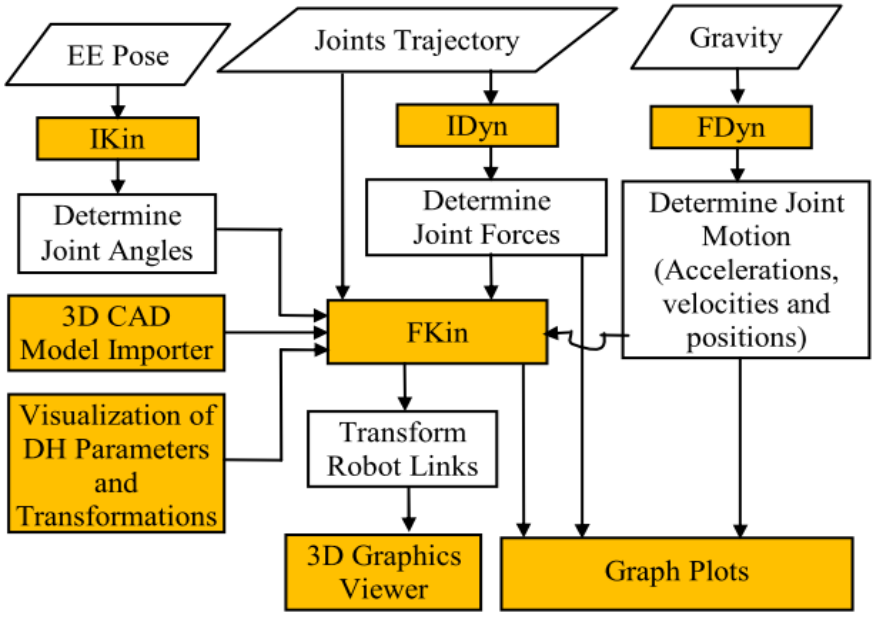


Fig. 2.6: Flowchart for the Modules of RoboAnalyzer [20]

2.2.6 ANFIS Technique

ANFIS technique [21] is a hybrid neuro-fuzzy approach which combines the concept of artificial neural networks and fuzzy logics; utilizing the discrete primacy. It is practically identical to Sugeno fuzzy inference system. It forms a bridge between fuzzy inference system and learning potential of artificial network; where membership functions for inferences are adjusted by back propagation method or conjunction of least square method.

Fuzzy logic is the most significant approach in soft computing field as it is fully accomplished to comprehend human expertise in terms of precursor IF and successor THEN formats. Nature

of the system in this dominion is described by using linguistic expressions. The classic architecture of these systems is initiated by Wang [22], Tagaki and Sugeno [23] and Jang [24].

In [22], such type of inference system is formed which constitutes Gaussian membership function, product inference concept and weighted average defuzzifier.

In [23], Defuzzification part is established in terms of dynamic systems. The qualitative feature of this concept is that under certain restraints, stability of systems can be interpreted.

Artificial Neural Network is a knowledge proceeding paradigm that is motivated with the fact how biological nervous system, such as the brain, process and proceeds information. It comprises of highly interconnected elements (neurons) combining in unison to resolve the problems. Main advantages of this system are: adaptive learning, self-organization, real time operation and fault tolerance via redundant information coding. Neural networks based solution for inverse kinematic problem of different robotic manipulators have intensively explained in [25, 26, 27].

Applications of Neural networks and fuzzy logic for obtaining the inverse kinematics solutions thoroughly described in [28, 29, 30]. There are multiple options after combining neural networks and fuzzy logic in such a way that they can subjugate their disadvantages as well as grasp the advantages from each other's classified qualities.

The literature shows so many conjunctions of Adaptive Neural Networks and Fuzzy Inference Systems as a hybrid: Adaptive Neuro-Fuzzy Inference Systems (ANFIS). It has been proposed for 2 DOF, 3 DOF and 5 DOF [31]. A comparative analysis of inverse kinematics solution for 3 DOF [32] and 5 DOF with wrist movement [33] industrial robotic manipulator has been done with experimental validation.

Within 4 DOF robotic class, kinematic modelling and simulation of SCARA has been proposed, analytically [34]. A neural network based approach is also elaborated in [35], for finding inverse kinematic solutions. But they did not emphasize on the idea of trajectory tracking through ANFIS technique. Though, in [36], tracking control of 3 DOF SCARA is explained with ANFIS controller and PID controller, but they concentrated mainly on different rise functions.

ANFIS Architecture

ANFIS method works in five layers (as shown in Figure 2.7). The role of each layer is described as follows:

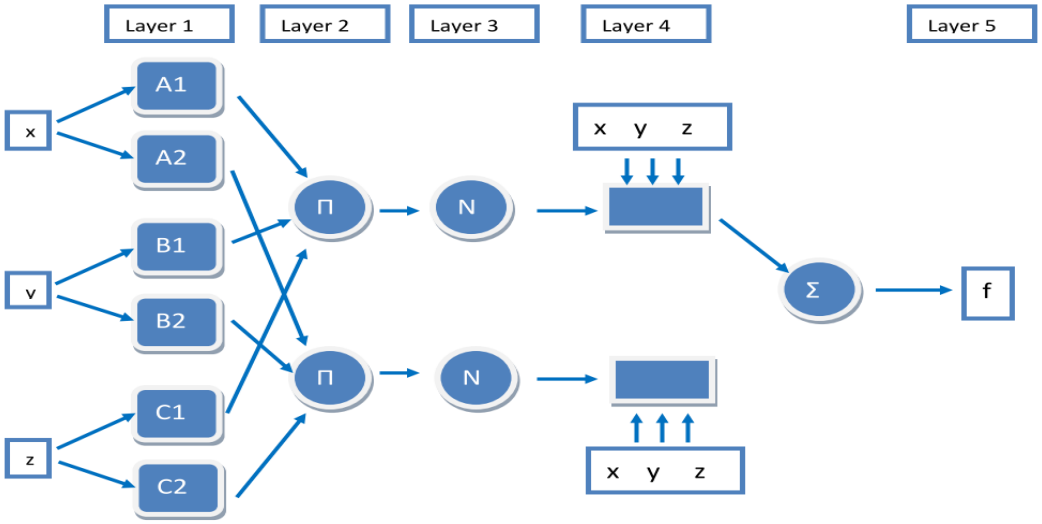


Fig. 2.7: ANFIS Architecture

Layer 1: The Node Layer

This layer provides the input parameters to the succeeding layer. In this, every node itself forms a complete fuzzy set and output in corresponding fuzzy set works as the input parameters membership class. Here, the node function O_i^j (for i^{th} position of j^{th} layer) is evaluated as:

$$O_i^1 = r_{Ai}(x);\ O_i^1 = r_{Bi}(y);\ O_i^1 = r_{Ci}(z) \tag{2.30}$$

Where, (r_{Ai}, r_{Bi}, r_{Ci}) is membership function for that particular input and (x, y, z) is the input vector.

Layer 2: The Membership Layer

Here, the output of every node which indicates the firing strength is calculated as product of all membership function; giving the specification of degree with which inputting to the next layer is done.

$$O_i^2 = R_i = r_{Ai}(x) r_{Bi}(y)\ r_{Ci}(z), i = 1,2,3 \tag{2.31}$$

Layer 3: The Rule Layer

In this, computation of activation level of every rule is done by comparing each of the firing strengths with summation of all firing strengths. It comprises of fixed nodes which computes the ratio of firing strength of the rules.

$$O_i^3 = \left(\frac{R_i}{R_1+R_2+R_3}\right) \tag{2.32}$$

Layer 4: The Defuzzzification Layer

In this layer, each IF-THEN rule as resulting output is inferred by precursor parameter or linear combination of input fuzzy individuals of ANFIS with constant expressions.

$$O_i^4 = R_i K_i = (u_i x + v_i y + w_i z + c_i) \tag{2.33}$$

where, (u_i, v_i, w_i, c_i) is consequent design parameters set.

Here, fuzzy IF-THEN rules are defined as:

If *x* is A and *y* is B and *z* is C, then $K = ux + vy + wz + c$

where, A, B and C are fuzzy set in precursor, and *u*, *v*, *w*, *c* are design parameters that are found in training procedure.

Layer 5: The Output Layer

In this layer, summation of all inputs coming from layer 4; consequently, whole ANFIS setup is automatically tuned up by the application of least square approximation and back propagation structure for the required membership function.

$$O_i^5 = \sum_i O_i^4 = \sum R_i K_i \tag{2.34}$$

The literature survey suggests that based on the selection criteria of being flat and non-zero at all points, gaussian (gaussmf and gauss2mf), sigmoidal (dsigmf and psigmf), generalized bell curve (gbellmf) and spline based curve (pimf) could be applied for the decision of membership function. From these six, only Bell and Gaussian MF is procuring admissible results. The detailed description of Bell and Gaussian MF is as follows:

Case a: The generalized bell MF's with product inference rule have been used in fuzzification level while defuzzification has been performed using weighted average method. A bell MF is given by Equation (2.35), where parameter 'c' gives distance from origin, parameter 'a' shows curve width and parameter 'b' is normally positive. Its representation is shown in Figure 2.8(a)

$$\mu\,(x;\, a,\, b,\, c) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}} \tag{2.35}$$

Case b: For fuzzification process, Gaussian MF's with product inference rule have been used; on the other hand, for defuzzification process, weighted average method has been applied. A symmetric Gaussian MF is defined by the Equation (2.36), where parameter 'σ' denotes width of the curve and 'a' shows the distance from the origin. A small 'σ' generates the thin MF while a big 'σ' leads to the flat MF. Representation of Gaussian MF is shown in the Figure 2.8-(b).

$$\mu\,(x;\, a,\, \sigma) \;=\; e^{\frac{-(x-a)^2}{2\sigma^2}} \tag{2.36}$$

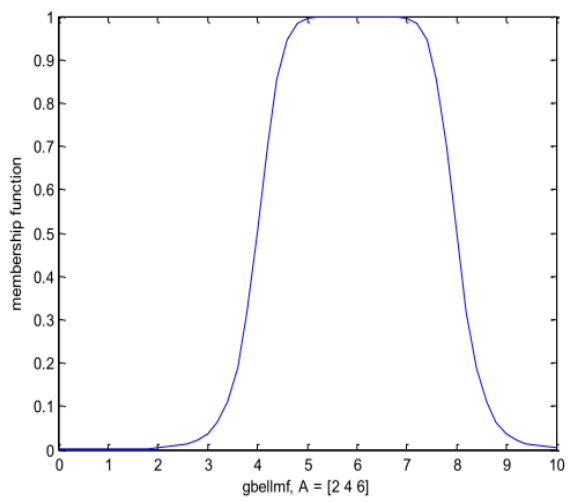


Fig. 2.8-(a): Representation of Bell MF [31]

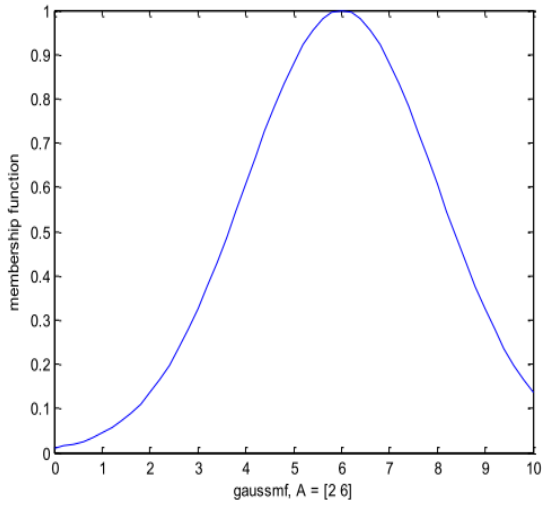


Fig. 2.8-(b): Representation of Gaussian MF [31]

2.3 Summary

In Geometrical method, for higher DOFs, solutions to first three joints should be prior known for finding other joint solutions. From the literature survey, it is found that there is no guarantee of closed-form solution for more than six DOFs in Algebraic method. In Jacobian transpose method, transpose used instead of inverse, for some appropriate scalar. If Jacobian matrix is of rectangular type, then Pseudo inverse method is used to calculate the inverse. However, EE position with respect to joint movements shows poor repeatability in case of Pseudo inverse method. In Extended Jacobian method, grad (H) stay in null space of J and in Lagrange Multiplier method, grad (H) stay in row space of J. However, it has some additional singularities. Extended Jacobian and Lagrange Multiplier methods are conservative but computationally expensive. In RoboAnalyzer, there is no provision of MDH Parameters or conversion of MDH Parameters into DH Parameters. Also, IKS are available for 2, 3 and 5

DOFs only in RoboAnalyzer in the present state. Nowadays, ANFIS approach is one of the most promising method to find IK solutions of higher DOFs manipulator, as an intelligent technique. In ANFIS approach, manipulator with wrist in motion is easily solved. With Bell and Gauss membership function, ANFIS approach generally produce better results. Trajectory tracking can easily be done with the ANFIS method, even in the case of obstacle avoidance.

Chapter 3

Application of ANFIS on SCARA Manipulator

3.1 Introduction

After conducting literature survey, it is known that ANFIS approach can produce satisfactory results for finding inverse kinematic solutions in case of unavailability of closed form solutions. Moreover, this includes less calculations in comparison with traditional methods of finding inverse kinematic solutions. In this chapter, a 4-DOFs spatial SCARA manipulator of industry and medical assisted use has been taken into consideration for finding the inverse kinematic solutions and desired path planning, consequently. The robotic manipulator constitutes of 3-DOFs at joints and 1-DOF at the end effector forming an R-R-P-R configuration. The Prismatic link is set to be move up and down which makes the robotic manipulator spatial. The 4-DOFs robotic manipulator presented in this work is shown in Figure 3.1. The robotic manipulator has been designed in the SOLIDWORKS software. The detailed description of the movements of joints and end-effector is tabulated in Table 3.1. The analytical method for solving the inverse kinematic problem for 4-DOFs SCARA manipulator has been described using algebraic approach in Chapter 2.

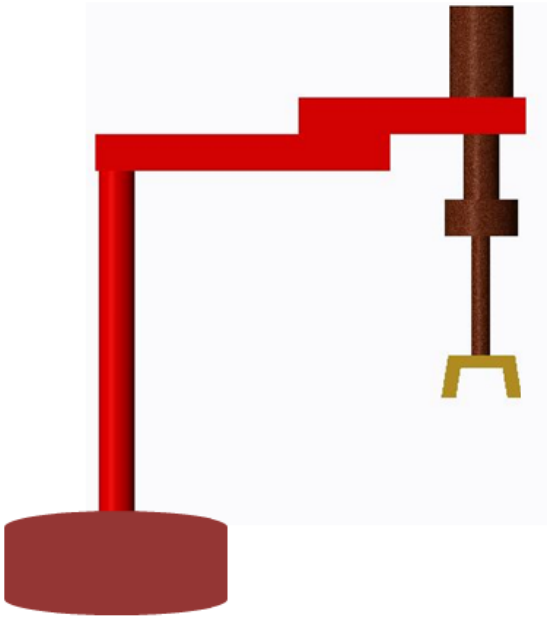


Fig. 3.1: CAD model of the 4-DOF SCARA

Table 3.1: Description of workspace for movement of SCARA manipulator

S. No.	Type	Part of Robot	Movement	Workspace
1.	Link 1	Waist	Left/Right	0 - π/4 (rad)
2.	Link 2	Shoulder	Left/Right	0 - π/4 (rad)
3.	Link 3	Elbow	Up/Down	100-130(mm)
4.	Wrist	Wrist Roll	Clockwise/ Anticlockwise	0 - π/4 (rad)

3.2 Formulation of Forward Kinematics

The kinematic analysis for any multi-DOFs robotic manipulator can be classified as: forward kinematics and inverse kinematics. With *n* degree of freedom, joint variables can be denoted by $\theta_i = \theta(t), \forall\, i = 1, 2, 3, \ \dots\ , \ n$ and position variables by $x_j = x(t), \forall\, j = 1, 2, 3, \ \dots\ , \ m$. The expression between end effector x_t and joint angle $\theta(t)$ can be formulated by forward kinematic equation as

$$\boldsymbol{x}(t) = \boldsymbol{f}\big(\boldsymbol{\theta}(t)\big) \tag{3.1}$$

where, $\boldsymbol{f}$ is a nonlinear, continuous and differentiable function. Due to availability of DH convention, finding forward kinematics is an easy task. Systematic assignment of different frames using D-H conventions is shown in Figure 3.2.

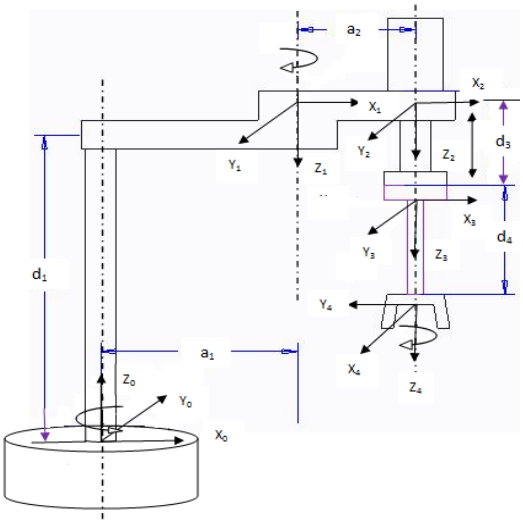


Fig. 3.2: DH frame representation of the SCARA robot

The DH convention has been used to formulate the forward kinematic equations, as tabulated in Table 3.2.

Table 3.2: DH parameters of the SCARA robot

S. No.	Joint Angle $\boldsymbol{\theta_i}$ (rad)	Joint offset $\boldsymbol{d_i}$ (mm)	Link Length $\boldsymbol{a_i}$(mm)	Twist Angle $\boldsymbol{\alpha_i}$ (rad)
1.	θ_1	$d_1 = 400$	$a_1 = 250$	0
2.	θ_2	0	$a_2 = 150$	π
3.	0	d_3	0	0
4.	θ_4	$d_4 = 150$	0	0

The general transformation matrix K_i^{i-1} amid two adjacent frames F_{i-1} and F_i is as follows in (3.2),

$$K_i^{i-1} = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

After, incorporating the DH parameters from Table 3.2 in (3.2), the forward kinematics equations can be formulated for 4-DOFs SCARA manipulator by multiplying K_1, K_2, K_3 and K_4 are as

$$\begin{aligned} K_4^0 &= \begin{bmatrix} R_{3\times3} & \vdots & w_{1\times3} \\ \cdots & \cdots & \cdots \\ 0 \quad 0 \quad 0 & \vdots & 1 \end{bmatrix} \\ &= \begin{bmatrix} C\theta_{1-2-4} & S\theta_{1-2-4} & 0 & a_1 C\theta_1 + a_2 C\theta_{1-2} \\ S\theta_{1-2-4} & -C\theta_{1-2-4} & 0 & a_1 S\theta_1 + a_2 S\theta_{1-2} \\ 0 & 0 & -1 & d_1 - d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{3.3}$$

where, $C\theta_i = cos(\theta_i)$, $S\theta_i = \sin(\theta_i)$, $C\alpha_i = cos(\alpha_i)$ and $S\alpha_i = sin(\alpha_i)$, $C\theta_{i-j} = cos(\theta_i - \theta_j)$, $S\theta_{i-j} = sin(\theta_i - \theta_j)$, $C\theta_{i-j-k} = cos(\theta_i - \theta_j - \theta_k)$, $S\theta_{i-j-k} = sin(\theta_i - \theta_j - \theta_k)$,

Using Equation (3.3),

$$w_1 = a_1 C\theta_1 + a_2 C\theta_{1-2} \quad (3.4)$$

$$w_2 = a_1 S\theta_1 + a_2 S\theta_{1-2} \quad (3.5)$$

$$w_3 = d_1 - d_3 - d_4 \quad (3.6)$$

where, w_1 , w_2, w_3 denotes translation about x-, y-, z- axis respectively.

3.3 ANFIS Procedure

ANFIS method performs in two segments -training and testing segment. In trained ANFIS data, the x, y and z coordinates of tip of the SCARA manipulator behave as the inputs and joint angles; θ_1, θ_2, d_3 and θ_4 as the outputs. In fourth data set, effect of wrist roll is considered. Here, four training data sets consisting of coordinates and joint angles has been regarded as (x, y, z, θ_1), (x, y, z, θ_2), (x, y, z, d_3) and (x, y, z, $S\theta_{1-2-4}$, $C\theta_{1-2-4}$, θ_4), respectively.

The corresponding MF's and number of rules have been designated for each training data set. First three datasets comprise of three MFs for each tip coordinate, leads to 27 rules. Last data set contains three MFs for each tip coordinate and wrist roll angle, leads to a total of 243 rules. The epochs used for the training data set are 10.

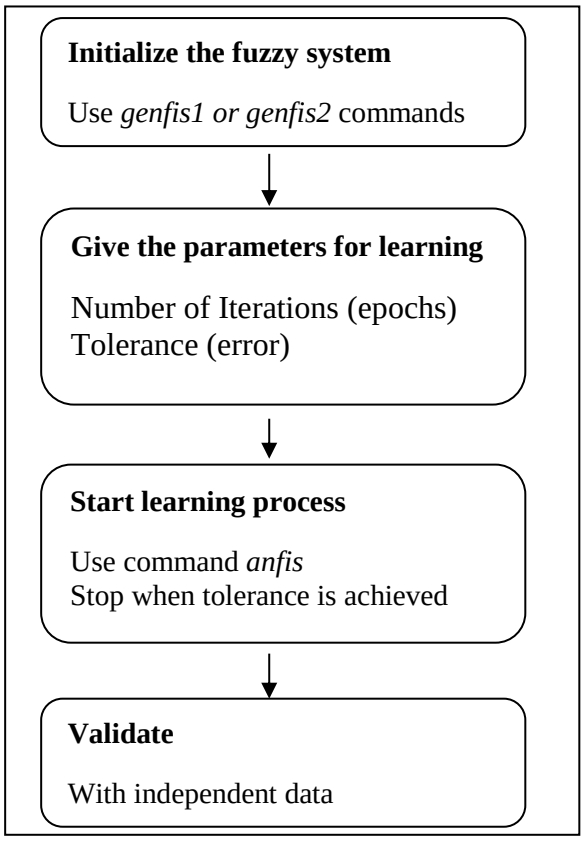


Fig. 3.3: Flowchart of ANFIS procedure

The basic procedure of ANFIS approach is defined in four steps; as shown in Figure 3.3. In the testing segment, the premise and consequent limits in ANFIS approach are modifiable as they are adaptive in character. This adaptive procedure of ANFIS approach is subdivided into two steps. In the first step, training of consequent parameter is done with the help of least square method because linear combination of same parameters gives ANFIS output.

The premise parameters are static during the first step. In the next step, back propagation of approximation error is done by apprising premise parameters over each layer. This fraction of learning process is in accordance with gradient descent principle which is correspondent to training of back propagation algorithm in neural networks.

This hybrid learning procedure is computationally inexpensive as the search domain of initial back propagation neural network is far decreased and ANFIS converges with less number of iterations.

3.4 Validation with Analytical Solutions

After implementation of ANFIS, simulation results for joints variables of 4-DOFs SCARA spatial manipulator are compared with analytical inverse kinematic solutions. Here, the workspace for which 4-DOFs SCARA manipulator is trained, tabulated in Table 3.1. The desired path is defined as

$$
\begin{aligned}
x_0 &= 0; \\
y_0 &= 0; \\
r &= 400; \\
x &= x_0 + r\cos\theta; \\
y &= y_0 + r\sin\theta; \\
z &= 135 \pm 15;
\end{aligned}
$$

where, $0 \leq \theta \leq 45$ (degrees)

where, x_0 and y_0 are the manipulators base or foot positions. The radius of the path is denoted by 'r'. x, y and z are the position coordinates along the respective directions. SCARA as a spatial robot, *z* coordinates vary simultaneously with *x* and *y* coordinates throughout the path generation process.

3.4.1 Results and Observations

Using MATLAB, the responses for joint variables (θ_1, θ_2, d_3, θ_4) for the desired path are shown in Figure 3.4. These results estimate the similarity between the analytical method and ANFIS approach for finding the inverse kinematic solutions. The '*x*-axis' show the joint variables (θ_1, θ_2, d_3, θ_4) and on the '*y*-axis', number of samples are shown. The blue circles in the responses infers ANFIS solutions whereas red cross symbols depict the analytical solutions. The deviation between these two arrays of colors is termed as error.

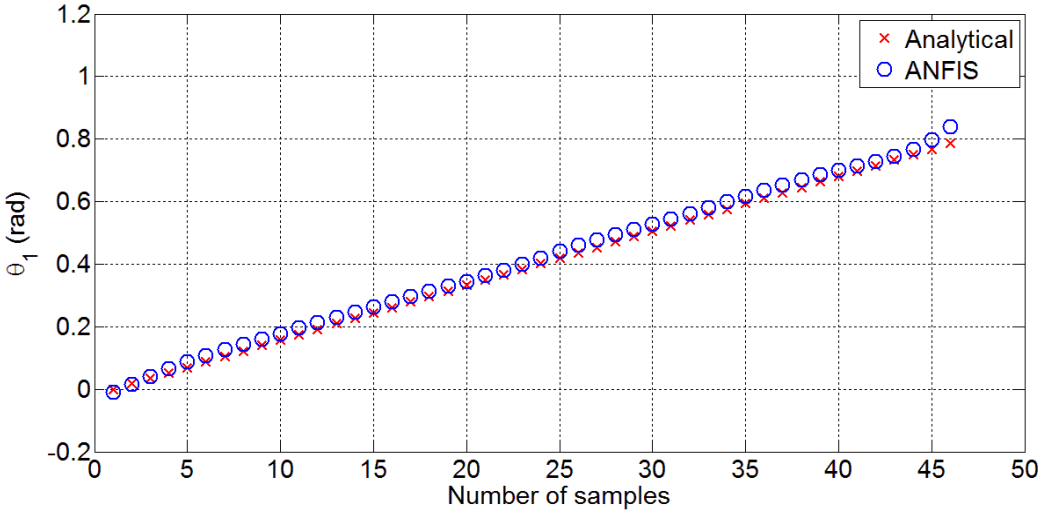


Fig. 3.4-(a): Simulation response of 'θ_1' with 'Number of samples'

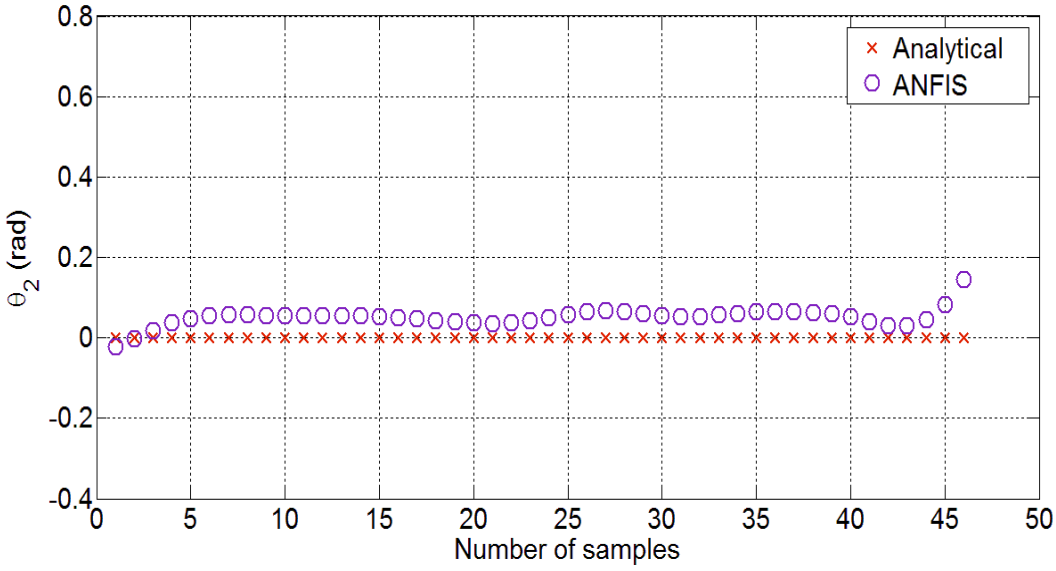


Fig. 3.4-(b): Simulation response of 'θ_2' with 'Number of samples'

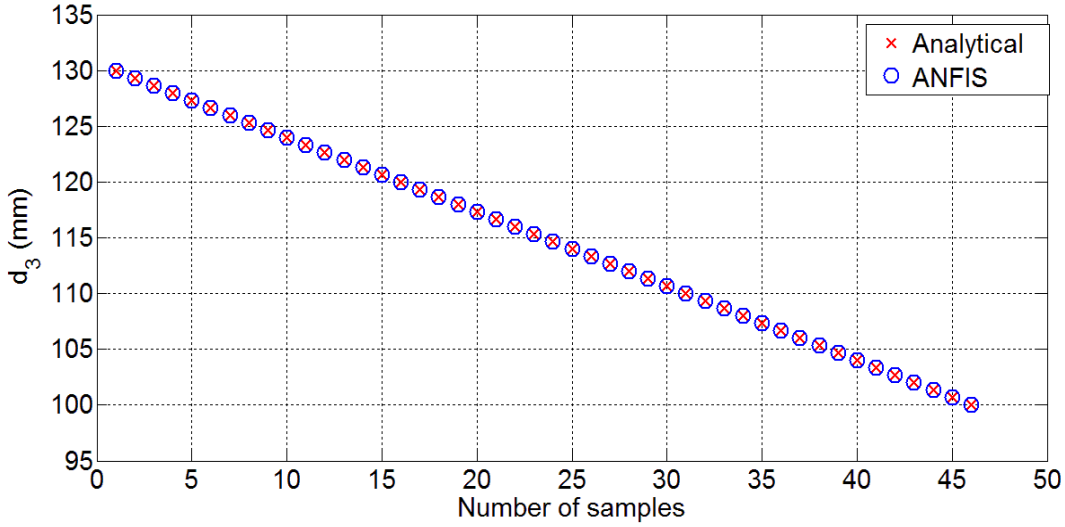


Fig. 3.4-(c): Simulation response of 'd_3' with 'Number of samples'

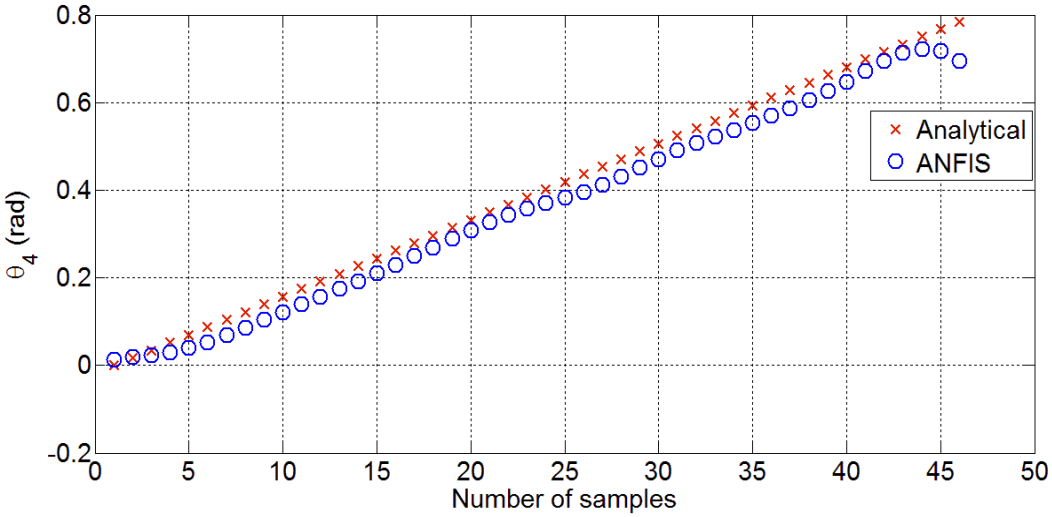


Fig. 3.4-(d): Simulation response of 'θ_4' with 'Number of samples'

From Figure 3.4, it is observed that maximum error in the inverse kinematic solutions after comparing ANFIS technique with the analytical method is 0.0173 radians, 0.1647 radians, 0.0032 mm and 0.1284 radians for *'θ_1', 'θ_2', 'd_3'* and *'θ_4'*, respectively.

3.5 Path Planning by 4-DOFs SCARA

Path planning is one of the important aspect associated with inverse kinematic solutions. After obtaining the joint variables, desired path is back propagated using ANFIS approach and compared with analytical solutions.

3.5.1 Results and Observations

A combined comparative analysis for analytical and ANFIS solutions with respect to desired path is shown in Figure 3.5-(a) and 3.5-(b) within two different views. Furthermore, in Figure 3.5-(c), the simulation runs for desired solutions is shown along with 4-DOFs SCARA links, where tip of robot is coordinated with analytical and ANFIS solutions, simultaneously. The *x*-axis, *y*-axis and *z*-axis represents the coordinate system in respective directions for end effector's tip movement.

The light grey dot represents the training workspace and red coloured star shows the desired path which need to be planned by analytical and ANFIS solutions. Analytical solutions are depicted by symbol of green circles whereas ANFIS planned path is represented by blue diamond symbols. The deviation between the respective *x, y* and *z* coordinates for the different methods with respect to desired coordinates is considered as error.

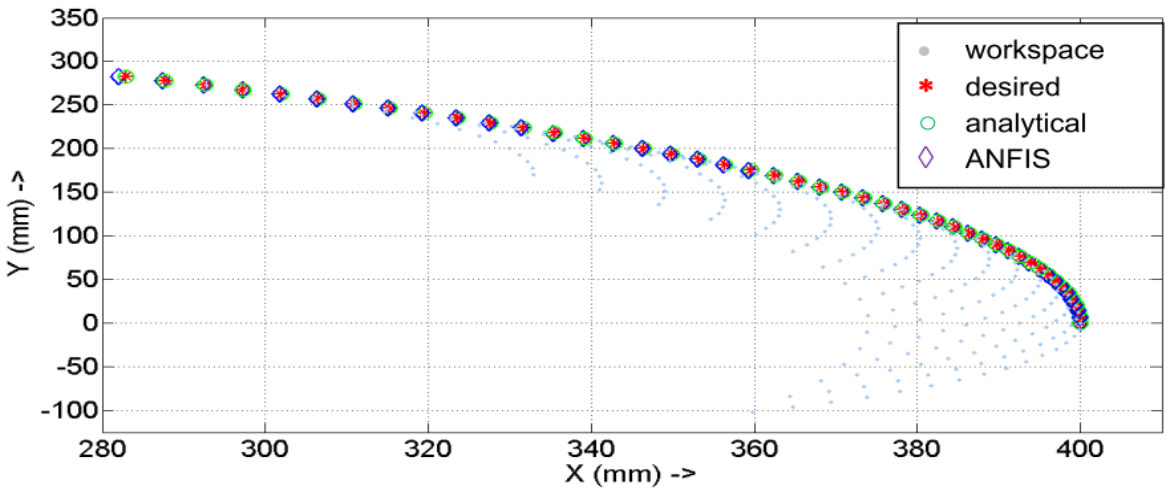


Fig. 3.5-(a): Analytical and ANFIS solutions for desired path in XY plane

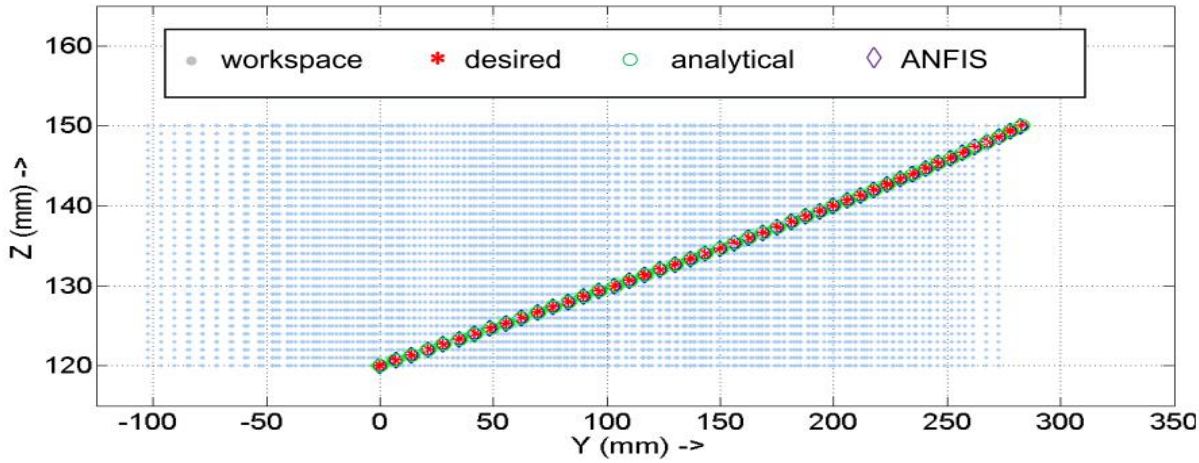


Fig. 3.5-(b): Analytical and ANFIS solutions for desired path in YZ plane

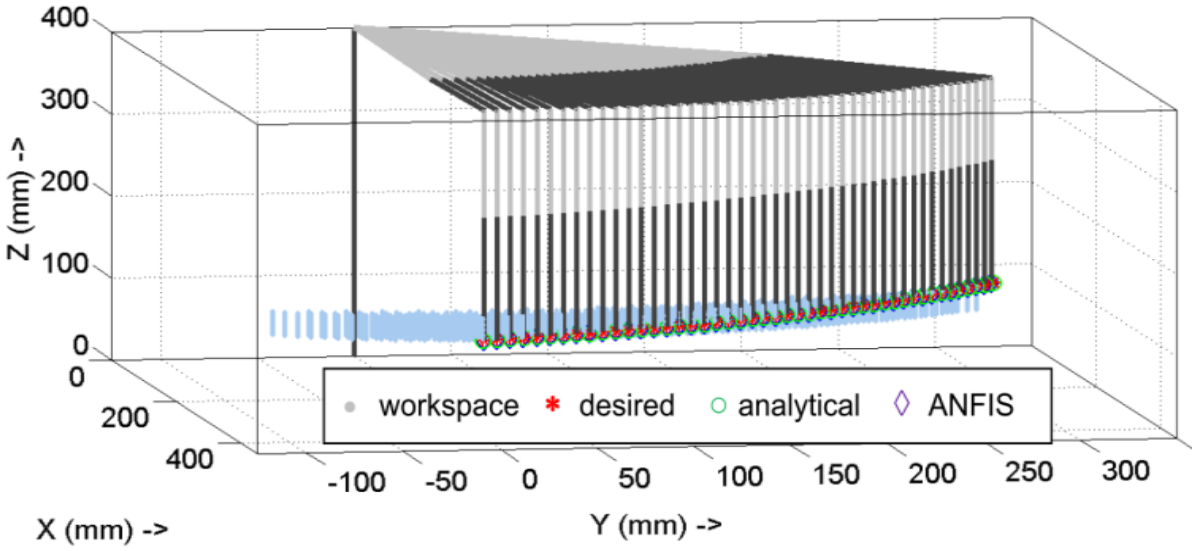


Fig. 3.5-(c): Desired path generation for analytical and ANFIS approach by manipulator links

From Figure 3.5, it is observed that with ANFIS approach, maximum deviation of end effector's position of robotic manipulator from the desired path is 0.3774 mm in *x* direction, 0.4135 mm in *y* direction and 0.0027 mm in *z* direction. Moreover, it is noticed that ANFIS approach is quite precise in comparison with analytical approach as deviation is 0.0003 mm, 0.0007 mm and 0.0001 mm in *x*, *y* and *z* directions, respectively.

3.6 Summary

It is quite evident from this chapter that ANFIS methodology for finding inverse kinematic solutions works satisfactorily, even without the need of formulating inverse kinematic equations. Moreover, path planning has been done with the ANFIS approach in this chapter. Comparative analysis with analytical method proves that combination of neural networks and fuzzy logic can be applied for path planning in case of higher degree of freedom manipulator. In the next chapter, path planning for 5-DOFs Spatial Medical manipulator is presented with the help of ANFIS approach.

Chapter 4

Path Planning by 5-DOFs Spatial Medical Manipulator Using ANFIS

4.1 Introduction

After applying ANFIS concept on the SCARA robot in chapter 3, the same is being applied on 5-DOFs Spatial Medical Manipulator for desired path planning to show its effectiveness. Path planning is one of the important phenomena associated with concept of inverse kinematic solutions. Through ANFIS, the joint parameters for certain path is back propagated for finding its coordinate system. In this chapter, path planning results for 5-DOFs manipulator having fixed orientation is presented along with their deviation from the original path.

4.2 Formulation of Forward Kinematics

In this section, a case study of surgical manipulator is taken, which is a 5-DOFs spatial manipulator. Proximal variant of DH parameter method is used for the kinematic study of this manipulator, whose prototype was developed at CSIR-CSIO, Chandigarh. It is a *patient-side manipulator*, which is used to track the movements of a *surgeon-side manipulator* during a robot-assisted surgery. The physical prototype of the 5-DOFs spatial manipulator is shown in the Figure 4.1 and its corresponding line diagram with frame assignment is shown in Figure 4.2. The DH parameters of this manipulator are calculated and tabulated in Table 4.1.

Table 4.1: D-H parameters obtained using proximal variant

S. No.	Link Length (a_{i-1}) (mm)	Joint Distance (d_i) (mm)	Link Twist (α_{i-1}) (deg.)	Joint Angle (θ_i) (deg.)	Home Position
1	$l_0 = 0$	d_1	$\alpha_0 = 0$	$\theta_1 = 0$	$1720\ mm$
2	$l_1 = 175$	$d_2 = 45$	$\alpha_1 = 0$	θ_2	$0°$
3	$l_2 = 324$	$-d_3 = 5$	$\alpha_2 = 0$	θ_3	$90°$
4	$l_3 = 0$	$d_4 = 558$	$\alpha_3 = 138$	θ_4	$90°$
5	$l_4 = 20$	$d_5 = 0$	$\alpha_4 = 90$	θ_5	$62°$
6	$l_5 = 290$	$d_6 = 0$	$\alpha_5 = 0$	$\theta_6 = 0$	-

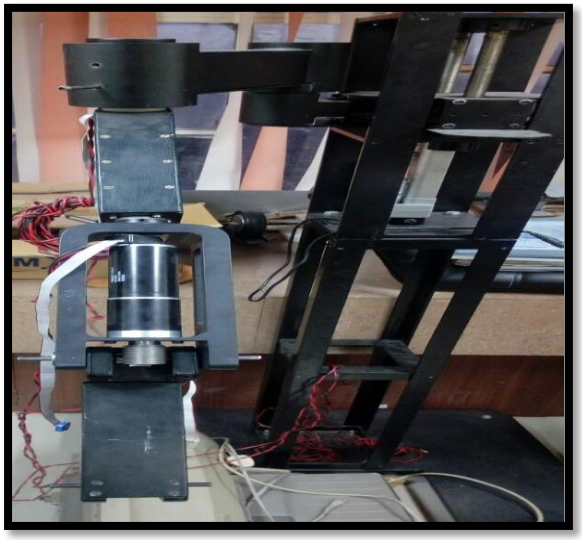


Fig. 4.1: Physical prototype of the 5-link spatial manipulator developed at CSIR-CSIO Chandigarh [37].

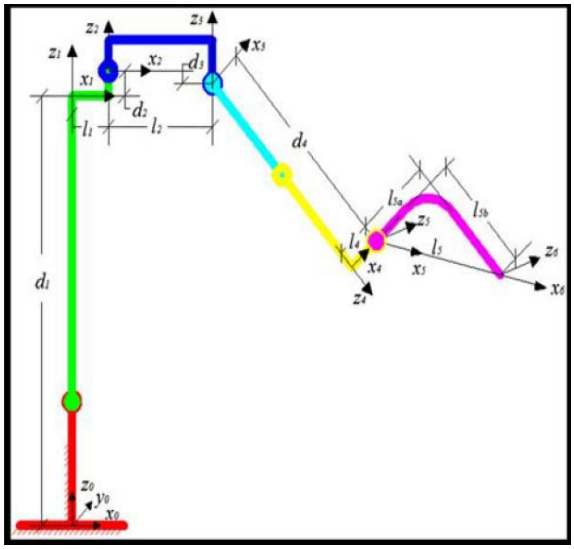


Fig. 4.2: Line diagram of the 5-link spatial manipulator [37].

Using the DH parameters given in Table 4.1, the tool-tip to base transformation matrix is given as [37],

$$ {}^{0}_{6}\boldsymbol{A} \qquad {}^{0}_{1}\boldsymbol{A}\,{}^{1}_{2}\boldsymbol{A}\,{}^{2}_{3}\boldsymbol{A}\,{}^{3}_{4}\boldsymbol{A}\,{}^{4}_{5}\boldsymbol{A}\,{}^{5}_{6}\boldsymbol{A} \tag{4.1} $$

$$ = \left[\begin{array}{ccc|c} 0.94 & -0.34 & 0 & 1159.58 \\ 0 & 0 & 1 & 0 \\ -0.34 & -0.94 & 0 & 1259.07 \\ 0 & 0 & 0 & 1 \end{array}\right] $$

4.3 Implementation of ANFIS Method

In trained ANFIS data, the x, y and z coordinates of tip of the 5-DOFs Spatial Medical manipulator and m, n, o, p, q, r, s, t, u, from orientation matrix behave as the inputs and joint angles d, θ_2, θ_3, θ_4, and θ_5 as the outputs. In this ANFIS simulation run. five data sets being considered for the training are (x, y, z, m, n, o, p, q, r, s, t, u, d), (x, y, z, m, n, o, p, q, r, s, t, u, θ_2), (x, y, z, m, n, o, p, q, r, s, t, u, θ_3), (x, y, z, m, n, o, p, q, r, s, t, u, θ_4), (x, y, z, m, n, o, p, q, r, s, t, u, θ_5) respectively. Each data set consists of three MFs for each tip coordinate leading to a total of 5,31,441 rules. The number of epochs used is 10.

Due to such large number of rules and having more number of inputs in training dataset, the generation of fuzzy inference is done by Subtractive Clustering method instead of Grid

Partitioning method. Subtractive Clustering is a fast, one-pass algorithm for estimating the number of clusters and the cluster centers in a set of data. The cluster estimates obtained from the *subclust* function can be used to initialize iterative optimization-based clustering methods (`fcm`) and model identification methods (like `anfis`). The *subclust* function finds the clusters by using the subtractive clustering method.

The *genfis1* function is used for activating the Grid Partitioning fuzzy inference system. The *genfis2* function builds upon the subclust function to provide a fast, one-pass method to take input-output training data and generate a Sugeno-type fuzzy inference system that models the data behaviour.

4.3.1 Results and Observations

The use of above manipulator is considered to track the desired path, covering the position coordinates of arm matrix having fixed orientation, which is as given below:

$$
\begin{aligned}
x_0 &= 0; \\
y_0 &= 0; \\
r &= 1159.58; \\
x &= x_0 + r\cos\theta; \\
y &= y_0 + r\sin\theta; \\
z &= 1270 \pm 50; \\
&\text{where, } -15 \leq \theta \leq 15 \text{ (degrees)}
\end{aligned}
$$

where, x_0 and y_0 are the manipulators base or foot positions. The radius of the path is denoted by *'r'*. x, y and z are the position coordinates along the respective directions.
A comparative analysis for ANFIS solutions with respect to desired path is shown in Figure 4.3-(a) and 4.3-(b) within two different views. Furthermore, in Figure 4.3-(c), the simulation runs for desired solutions is shown along with 5-DOFs Spatial Medical manipulator links, where tip of robot is coordinated with ANFIS solutions, simultaneously. The *x*-axis, *y*-axis and *z*-axis represents the coordinate system in respective directions for end effector's tip movement.

The light grey dot represents the training workspace and red colored star shows the desired path which need to be planned by analytical and ANFIS solutions. ANFIS planned path is represented by blue diamond symbols. The deviation between the respective *x*, *y* and *z*

coordinates for the ANFIS method with respect to desired coordinates is considered as error. The percentage error in the desired and ANFIS obtained *x*-, *y*- and *z*- coordinates are shown in Figure 4.4-(a), -(b) and -(c).

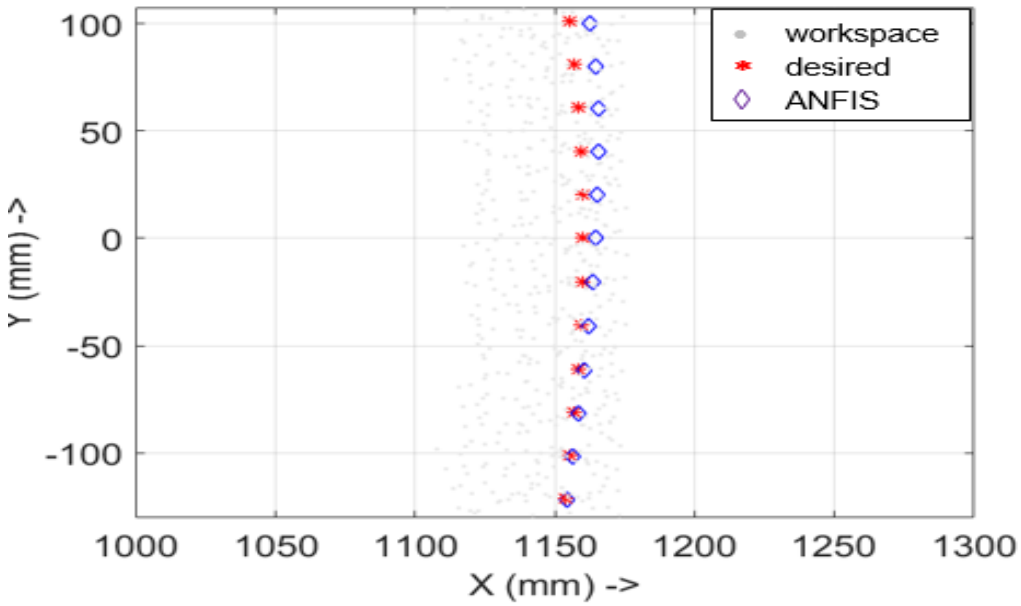


Fig. 4.3-(a): ANFIS solutions for desired path in XY plane

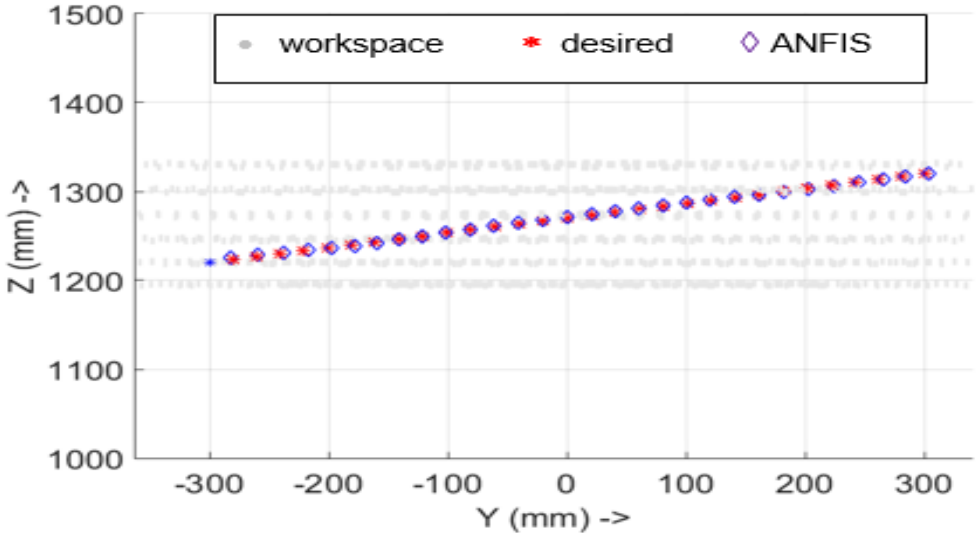


Fig. 4.3-(b): ANFIS solutions for desired path in YZ plane

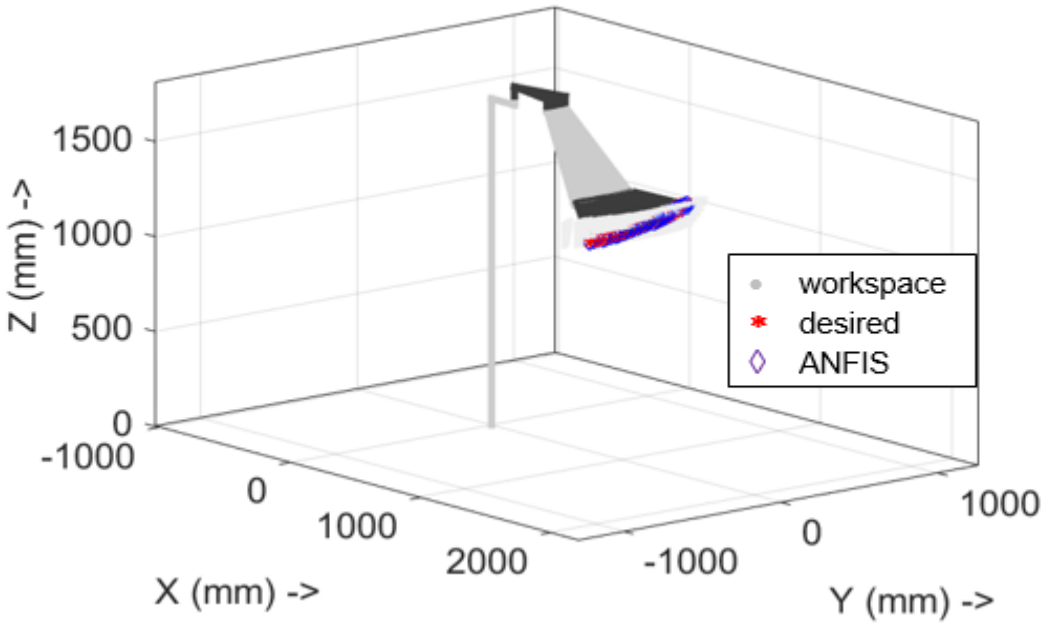


Fig. 4.3-(c): Desired path generation for ANFIS approach by manipulator links

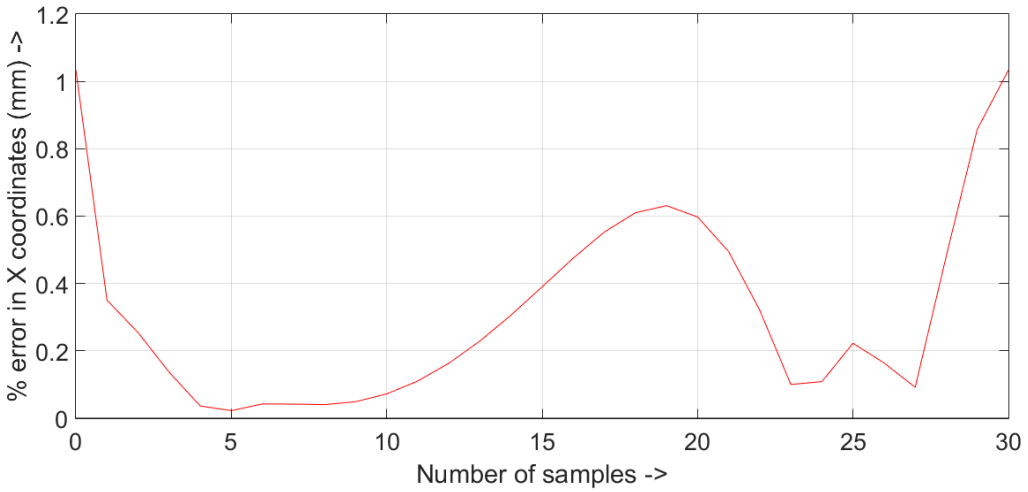


Fig. 4.4-(a): Plot of 'Error in X- coordinates' with 'Number of samples'

It is concluded from the Figure 4.4-(a), the maximum absolute deviations between the desired and ANFIS obtained X-coordinate are at initial and final points, with the value of 1.03 % and 1.07 %. From the Figure 4.4-(b), the maximum absolute deviations between the desired and ANFIS obtained Y-coordinate are present near the initial point and final point, with the value of 1.32 % and 1.35 %. In the Figure 4.4-(c), the maximum absolute deviations between the desired and ANFIS obtained Z-coordinate is at sample number two, with the value of 0.19 %.

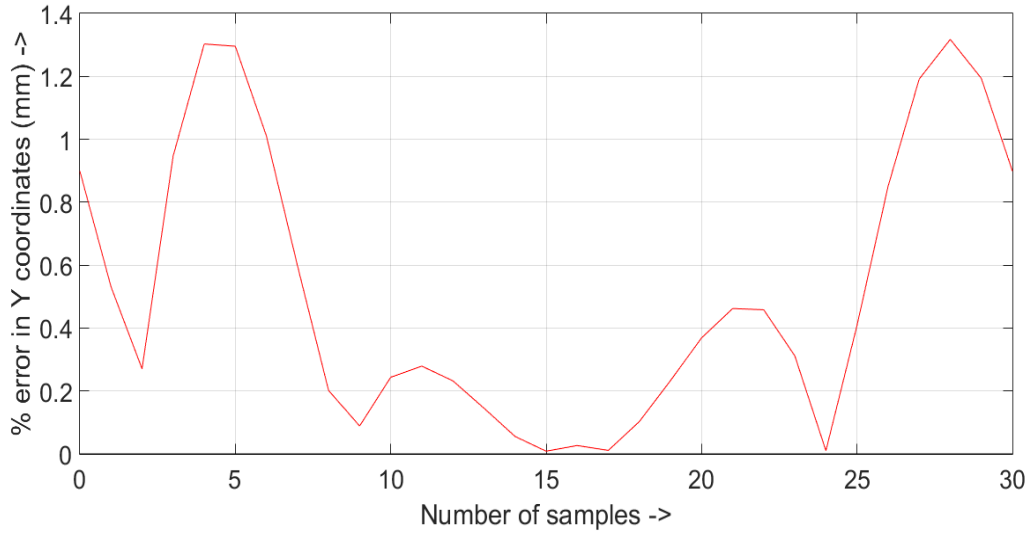


Fig. 4.4-(b): Plot of 'Error in Y- coordinates' with 'Number of samples'

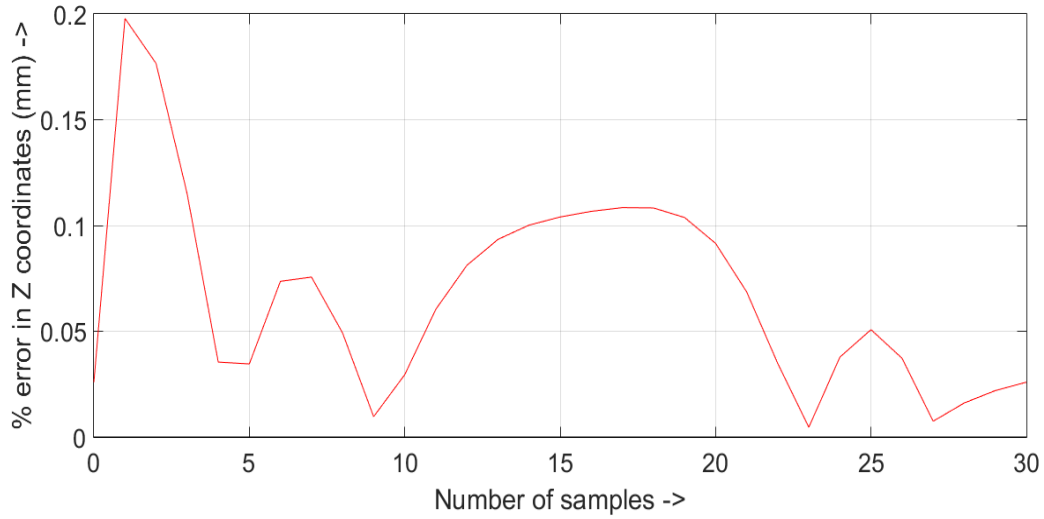


Fig. 4.4-(c): Plot of 'Error in Z- coordinates' with 'Number of samples'

Furthermore, comparison between actual values and ANFIS based obtained values of joint parameters, termed as predicted values, for the arm matrix calculated in Equation 4.1, is done. The maximum error is found for the second joint parameter (θ_2) with the modulus value of 4.22 degrees whereas the minimum error is 0.41 degrees in case of third joint parameter (θ_3). The sensitiveness of the manipulator links is inherently dependent on the workspace of training data set. Therefore, the error could be reduced by more refinement of training data set domain, input and output parameters and number of epochs. The result for this comparative analysis is shown in Table 4.2.

Table 4.2. Error between actual value and predicted value for joint parameters

Joint Parameters	Actual Value	Predicted Value	|Error|
D	1720 mm	1717.5 mm	2.5 mm
θ_2	0^0	-4.22^0	4.22^0
θ_3	90^0	89.59^0	0.41^0
θ_4	90^0	88.78^0	1.22^0
θ_5	62^0	58.99^0	3.01^0

4.4 Summary

In this chapter, a 5-DOFs spatial medical manipulator has been considered for desired path planning with the ANFIS approach. With the help of forward kinematics, as input and output parameters for training datasets in ANFIS, the desired path planning has been done. Thereafter, the percentage error between the actual and ANFIS predicted values for the *x*-, *y*- and z-coordinates has been calculated. Moreover, to show the inverse kinematic analysis, the joint parameters have been predicted by the ANFIS for arm matrix corresponding to home position of 5-DOFs spatial medical manipulator. In the next chapter, the application of redundant manipulators in narrow channels is described with the help of 3-DOFs manipulator path planning having three obstacles, to show the efficacy of ANFIS approach.

Chapter 5

Obstacle Avoidance using ANFIS Technique

5.1 Introduction

The obstacle avoidance problem is well-known in robotics. The first solutions were based on an initial trajectory planning of a collision-free path for the robot. However, such solutions are expensive and bound the interactivity of the robot with its environment. Hence, new approaches have been developed to deal with dynamically varying obstacles in real-time. In this paper, an attempt has been made, to investigate the potential of ANFIS for obstacle avoidance by 3-link manipulator, with two and three obstacles. Obstacle avoidance is very impactful in different application of robotics like underwater welding in narrow tanks, checking the blockage of sewerage pipes etc.

5.2 Case study: A 3-DOFs Manipulator with Three Obstacles

In this case-study, a 3-DOFs manipulator is considered with all revolute joints. The physical dimensions of the links of the manipulator and the link parameters are given in Table 5.1.

Table 5.1: Physical dimensions and parameters of 3-DOFs manipulator

No. of Links *(i)*	Link Masses *(m) (kg)*	Link Parameters				
		Initial Joint Angle *(θ_i) (deg)*	Link Length ($l_i = a_i$) *(m)*	Joint Offset Distance (d_i) *(m)*	Twist Angle (α_i)*(deg)*	Initial Joint Angle Velocity *($\dot{\theta}_i$) (deg/min)*
1	20	50	1.3	0	0	1
2	20	20	0.9	0	0	1
3	8	-110	0.7	0	0	1

The considered 3-DOFs manipulator moves in a straight line trajectory from point *(*1.69, 1.4*)* at time *t = 0 min.* to point *(1.69, 0)* at time *t = 1 min.* In the simulation runs, the desired trajectory is shown in pink color. The base of the manipulator is fixed at origin. The position of the i^{th} obstacle in the Cartesian space can be calculated from its centre of gravity (x_i, y_i) and

the size can be found from its length (l_i) and breadth (b_i) dimensions. The complete geometrical information of all the three obstacles is tabulated in Table. 5.2. In Figure 5.1, the simulation is shown for the case of no obstacles in the task space.

Table 5.2: Geometrical data of three obstacles

Obstacle	Centre of Gravity		Dimensions *(m)*	
O_i	x_i	y_i	l_i	b_i
O_1	0.3	0.7	0.2	0.2
O_2	1.0	0.4	0.4	0.8
O_3	1.5	0.5	0.2	0.4

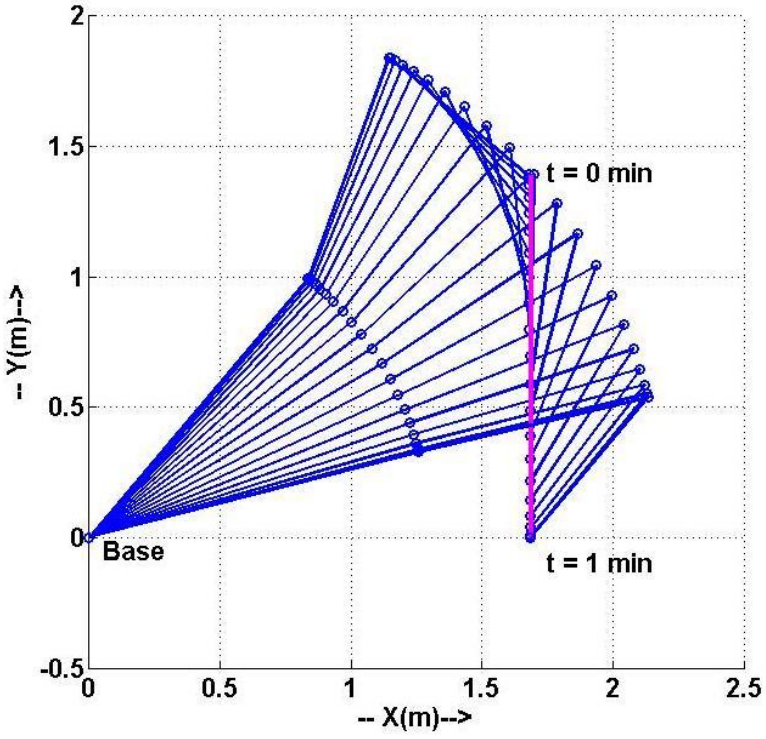


Fig. 5.1: 3-DOFs manipulator following straight line trajectory with no obstacle in its path.

Further, the 3-DOFs manipulator is considered with three obstacles (O_1, O_2, *and* O_3) in its workspace and it is shown in Figure 5.2, that how the manipulator changes its joint

configuration trajectory while considering these obstacles in the task space. ANFIS approach is implemented to show its effectiveness.

ANFIS method works in two phases - training phase and testing phase. In trained ANFIS data, the x, y and z coordinates of tip of the manipulator act as the inputs and joint angles; θ_1, θ_2, and θ_3 as the outputs. Here, four training data sets comprising of coordinates and joint angles has been considered as (x, y, z, θ_1), (x, y, z, θ_2), (x, y, z, θ_3), respectively. The respective MF's and number of rules have been assigned for each training data set. Three data set consists of three MFs for each tip coordinate leading to a total of 27 rules. The number of epochs used is 10. In testing phase, validation of data sets in terms of mean square error, is done by comparing calculated (x, y, z) coordinates using forward kinematics and (x, y, z) using ANFIS approach as individual data set.

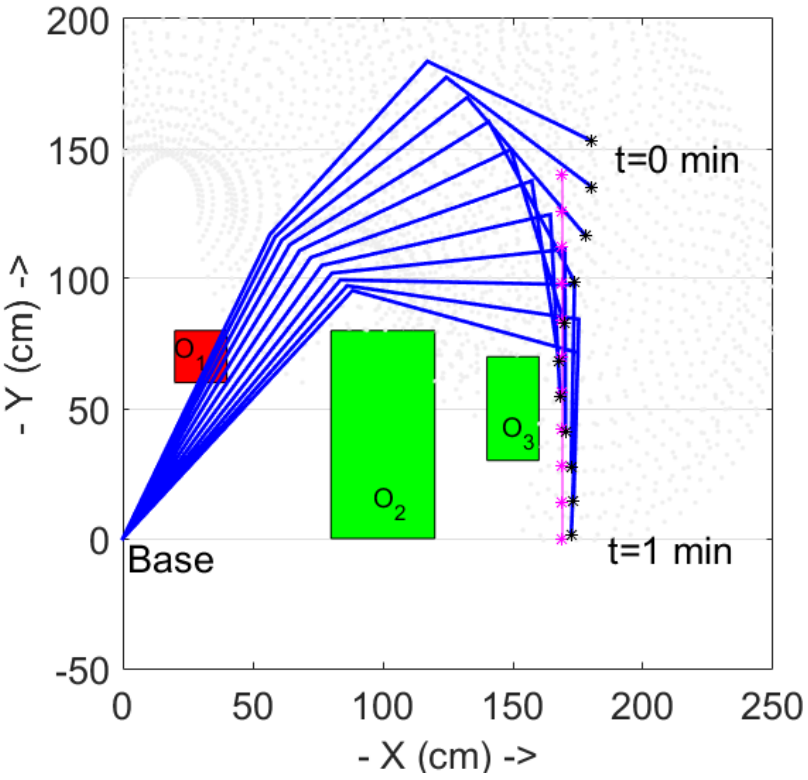


Fig. 5.2: 3-DOFs manipulator following trajectory avoiding two obstacles using ANFIS

5.3 Results and Observations

The implementation of the ANFIS approach is shown in Figure 5.2 and Figure 5.3, where the light-grey data points are showing the trained workspace. The desired path is followed by black colored star '*' symbol. If the given trajectory or any obstacle or both falls outside the trained

workspace, the manipulator behaves erratically and might result in unexpected movements. As seen in Figure 5.2, the first obstacle in red color is not successfully avoided by the manipulator link because the respective coordinate system of the same obstacle has not been included within training dataset of ANFIS algorithm. However, if the trajectory and obstacles are within the trained workspace, as shown in Figure 5.3, the ANFIS approach shows good results for avoidance of multiple obstacles by maneuvering their links with any joint configuration within the joint limit to trace the straight line trajectory within the trained workspace.

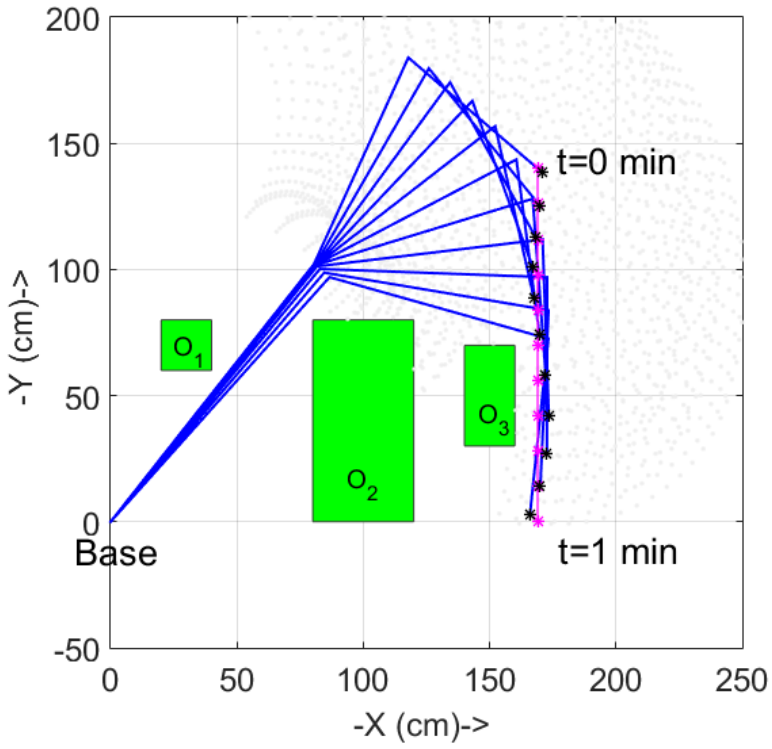


Fig. 5.3: 3-DOFs manipulator following trajectory avoiding three obstacles using ANFIS

5.4 Summary

In this chapter, a case study of 3-DOFs manipulator has been presented for avoidance of three active obstacles at the same time, using ANFIS approach. The produced results are quite convincing for the tracking of desired path within the trained workspace of the manipulator. These results prove that in case of higher degree of freedom redundant manipulators for their application in narrow channels, the concept of ANFIS approach will act positively. However, the precision and accuracy of the results are inherently dependent on the training data sets of the concerned manipulator.

Chapter 6

Conclusions and Future Directions

6.1 Conclusions

From the current work, the following conclusions can be drawn:

- From the thorough literature review, it can be concluded that the most widely used methods for finding inverse kinematic solution are the, algebraic and geometric methods. However, they are applicable only when degree of freedom of any manipulator is less than six and closed form solutions are possible.

- It is also concluded from literature review, numerical methods like Jacobian transpose and Pseudo inverse produce better results in case of non-availability of closed form solutions. However, Jacobian transpose fails, if type of Jacobian matrix is rectangular and Pseudo inverse show poor repeatability near singularities.

- It is also inferred from the literature review, the redundancy resolution methods like Extended Jacobian and Lagrange Multiplier show better repeatability unlike Pseudo inverse method. However, Lagrange Multiplier has some additional singularities. Moreover, there are ample of ambiguities in it as reported by different researchers.

- With the help of literature survey, it is also revealed that RoboAnalyzer software, which is commercially available, provide better platform to visualize the DH parameters and finding the forward and inverse kinematics, consequently. However, there is no provision of MDH parameters in the RoboAnalyzer software.

- From the above conclusions and this thesis work, it is finally decided that, nowadays, artificial intelligent techniques as a combination of neural networks and fuzzy logic named ANFIS supports better conformity in finding the inverse kinematic solution of any degree of freedom manipulator. A comparative analysis for finding the inverse kinematic solution of 4-DOFs SCARA shows the efficacy of the proposed ANFIS concept.

- Further, in favor of ANFIS approach, path planning by 5-DOFs spatial medical manipulator has been done to show the effectiveness of ANFIS approach which is

computationally inexpensive.

- Finally, it is concluded that obstacle avoidance, an important application for redundant manipulators in narrow channels, by 3-DOFs manipulator with three active obstacles has been presented using ANFIS approach.

6.2 Future Directions

The work completed in the present thesis, can be extended in some directions as

- The work present in thesis is limited to the inverse kinematic study only, which can be extended for the dynamic study of spatial medical manipulators.
- The work is also limited to path planning of 4-DOFs SCARA and 5-DOFs P-R-R-R-R spatial medical manipulator, which can be extended to higher degrees of freedom manipulator.

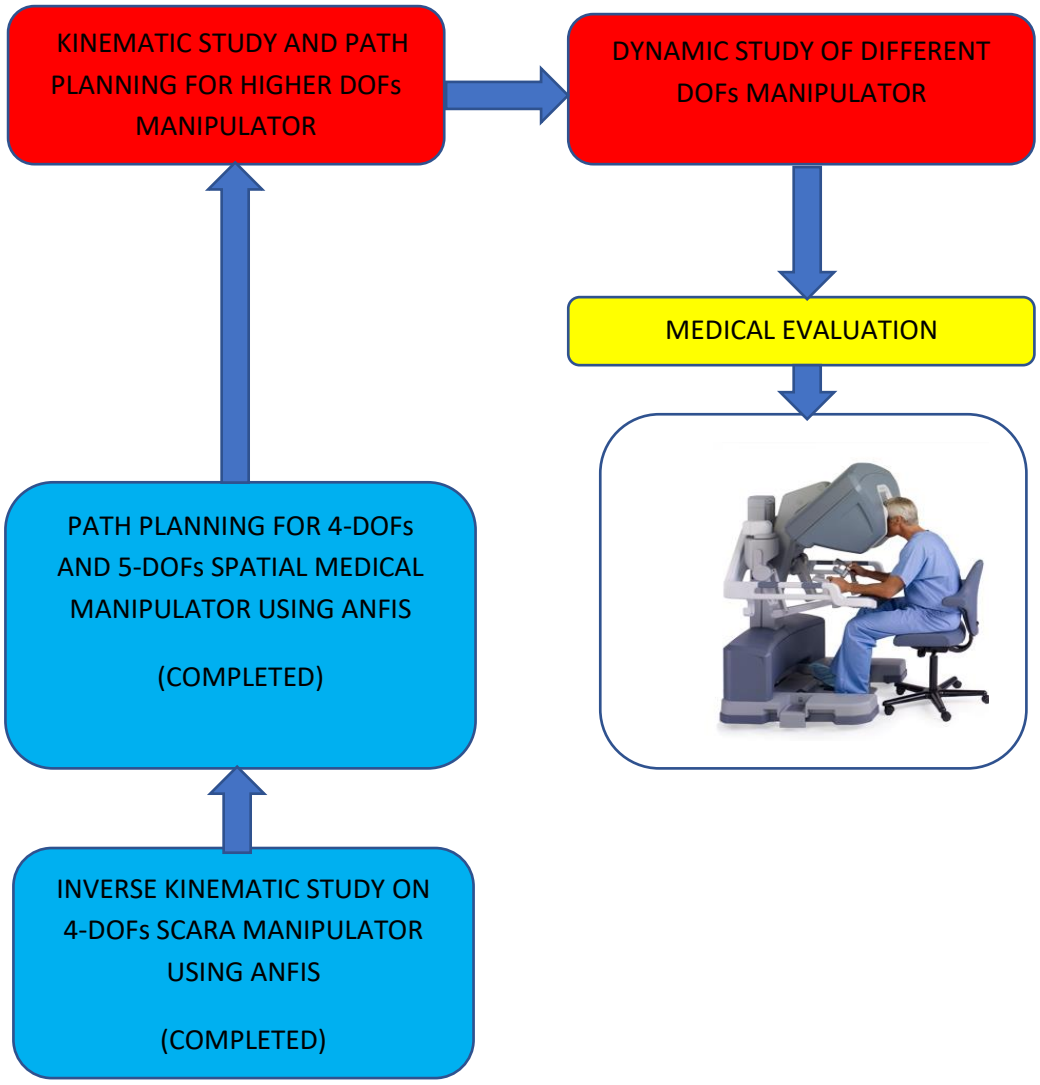


Fig. 6.1: The proposed general Layout of Surgery-Assisted Robotic Systems

- The reported work deals with 5-DOFs spatial medical manipulator, which is a surgeon side manipulator. The work can be extended for a set of coordinated robots, where both the surgeon- and patient-side manipulators works together.
- In case of obstacle avoidance, the work in this thesis is limited to 3-DOFs manipulator with three obstacles, which can be extended for higher degree of freedom manipulator with more number of obstacles.

REFERENCES

[1] Kim, Won, Frank Tendick, and Lawrence Stark. “Visual enhancements in pick-and-place tasks: Human operators controlling a simulated cylindrical manipulator.” *IEEE Journal on Robotics and Automation* 3, no. 5 (1987): 418-425.

[2] Singh, Amanpreet, Ashish Singla, and Sanjeev Soni. "Extension of DH parameter method to hybrid manipulators used in robot-assisted surgery." *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* (2015): 0954411915602289

[3] Huang, Jing, and Catherine Pelachaud. "An efficient energy transfer inverse kinematics solution." *International Conference on Motion in Games*, pp. 278-289. Springer Berlin Heidelberg, 2012.

[4] Kolodny, Rachel, Leonidas Guibas, Michael Levitt, and Patrice Koehl. "Inverse kinematics in biology: the protein loop closure problem." *The International Journal of Robotics Research* 24, no. 2-3 (2005): 151-163.

[5] Asada, Haruhiko, and J-JE Slotine. “*Robot analysis and control”*. John Wiley & Sons, 1986.

[6] Schilling, Robert J. “*Fundamentals of Robotics: Analysis and Control”*. Simon & Schuster Trade, 1996.

[7] Buss, Samuel R., and Jin-Su Kim. "Selectively damped least squares for inverse kinematics." *Journal of Graphics, GPU, and Game Tools* 10, no. 3 (2005): 37-49

[8] Orin, David E., and William W. Schrader. "Efficient computation of the Jacobian for robot manipulators." *The International Journal of Robotics Research* 3, no. 4 (1984): 66-75.

[9] Balestrino, A., De MARIA G, and L. Sciavicco. "Robust control of robotic manipulators." In *9th World Congress of the IFAC*. 1984

[10] Wolovich, William A., and H. Elliott. "A computational technique for inverse kinematics." In *Decision and Control, 1984. The 23rd IEEE Conference on*, pp. 1359-1363. IEEE, 1984.

[11] Whitney, Daniel E. "Resolved motion rate control of manipulators and human prostheses." *IEEE Transactions on man-machine systems* (1969).

[12] Liegeois, Alain. "Automatic supervisory control of the configuration and behavior of multibody mechanisms." *IEEE transactions on systems, man, and cybernetics* 7, no. 12 (1977): 868-871.

[13] Maciejewski, Anthony A., and Charles A. Klein. "The singular value decomposition: Computation and applications to robotics." *The International Journal of Robotics Research* 8, no. 6 (1989): 63-79.

[14] Klein, Charles A., and Ching-Hsiang Huang. "Review of pseudoinverse control for use with kinematically redundant manipulators." *IEEE Transactions on Systems, Man, and Cybernetics* 2 (1983): 245-250.

[15] Baillieul, John. "Kinematic programming alternatives for redundant manipulators." In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, pp. 722-728. IEEE, 1985.

[16] Chang, Pyung. "A closed-form solution for the control of manipulators with kinematic redundancy." In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3, pp. 9-14. IEEE, 1986.

[17] Klein, Charles A., and Li-Chung Chu. "Comparison of extended Jacobian and Lagrange multiplier based methods for resolving kinematic redundancy." *Journal of Intelligent and Robotic Systems* 19, no. 1 (1997): 39-54.

[18] Rajeevlochana, C. G., and S. K. Saha. "RoboAnalyzer: 3D model based robotic learning software." In *International Conference on Multi Body Dynamics*, pp. 3-13. 2011.

[19] Rajeevlochana, C. G., A. Jain, S. V. Shah, and S. K. Saha. "Recursive robot dynamics in RoboAnalyzer." In *Machines and Mechanisms (Proc. of the 15th Nat. Conf. on Machines and Mechanisms), edited by: Bandopadhyay, S., Gurunathan, SK, and Ramu, P., Narosa Publishing House, New Delhi*, pp. 482-490. 2011.

[20] Sadanand, R., R. G. Chittawadigi, and S. K. Saha. "Virtual robot simulation in RoboAnalyzer." In *1st International and 16th National Conference on Machines and Mechanisms*. 2013.

[21] Jang, J-SR. "ANFIS: adaptive-network-based fuzzy inference system." *IEEE transactions on systems, man, and cybernetics* 23, no. 3 (1993): 665-685.

[22] Wang, Li-Xin. *Adaptive fuzzy systems and control: design and stability analysis*. Prentice-Hall, Inc., 1994.

[23] Takagi, Tomohiro, and Michio Sugeno. "Fuzzy identification of systems and its applications to modeling and control." *IEEE transactions on Systems, Man, and*

Cybernetics 1 (1985): 116-132.

[24] Jang, Jyh-Shing Roger, Chuen-Tsai Sun, and Eiji Mizutani. "Neuro-fuzzy and soft computing; a computational approach to learning and machine intelligence." (1997).

[25] Wei, Li-Xin, Hong-Rui Wang, and Ying Li. "A new solution for inverse kinematics of manipulator based on neural network." In *Machine Learning and Cybernetics, 2003 International Conference on*, vol. 2, pp. 1201-1203. IEEE, 2003.

[26] Köker, Raşit, Cemil Öz, Tarık Çakar, and Hüseyin Ekiz. "A study of neural network based inverse kinematics solution for a three-joint robot." *Robotics and Autonomous Systems* 49, no. 3 (2004): 227-234.

[27] Köker, Raşit, Tarık Çakar, and Yavuz Sari. "A neural-network committee machine approach to the inverse kinematics problem solution of robotic manipulators." *Engineering with Computers* 30, no. 4 (2014): 641-649.

[28] Tejomurtula, Sreenivas, and Subhash Kak. "Inverse kinematics in robotics using neural networks." *Information Sciences* 116, no. 2 (1999): 147-164.

[29] Alavandar, Srinivasan, and M. J. Nigam. "Inverse kinematics solution of 3DOF planar robot using ANFIS." *International Journal of Computers, Communications and Control, Supplementary Issue: Proceedings of ICCCC*3 (2008): 150-155

[30] Alavandar, Srinivasan, and M. J. Nigam. "Neuro-fuzzy based approach for inverse kinematics solution of industrial robot manipulators." *International Journal of Computers Communications & Control* 3, no. 3 (2008): 224-234.

[31] Manjaree, Shiv, Vijyant Agarwal, and B. C. Nakra. "Kinematic analysis using neuro-fuzzy intelligent technique for robotic manipulator." *International Journal of Engineering Research and Technology* 6, no. 4 (2013): 557-562

[32] Manjaree, Shiv. "Inverse kinematic analysis of 3-degree-of-freedom robotic manipulator using three different methods." *International Journal of Advances in Science and Technology* 6, no. 3 (2013): 71-80.

[33] Manjaree, Shiv, Bahadur Chand Nakra, and Vijyant Agarwal. "Comparative analysis for kinematics of 5-DOF industrial robotic manipulator." *acta mechanica et automatica* 9, no. 4 (2015): 229-240.

[34] Jha, Panchanand, and B. B. Biswal. "A neural network approach for inverse kinematic of a SCARA manipulator." *IAES International Journal of Robotics and Automation* 3, no. 1 (2014): 52.

[35] Jasim, Wesam Mohammed. "Solution of inverse kinematics for SCARA manipulator using adaptive neuro-fuzzy network." *International Journal on Soft Computing (IJSC)* 2, no. 4 (2011): 59-66.

[36] Tinkir, Mustafa, and Arif Ankarali. "Neuro-Fuzzy trajectory control of a SCARA robot." In *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, vol. 2, pp. 298-302. IEEE, 2010.

[37] Singh, S., Singla, A., Singh, A., Soni, S. and Verma, S., 2016. Kinematic modelling of a five-DOFs spatial manipulator used in robot-assisted surgery. *Perspectives in Science*, *8*, pp.550-553.

ONLINE REFERENCES

[W1] https://static-content.springer.com/image/art%3A10.1186%2F1475-925X-13-130/MediaObjects/12938_2014_Article_868_Fig3_HTML.jpg (accessed on 22nd June 2017)

[W2] http://exploringanimation.blogspot.in/2015/03/week-2-uv-mapping-texturing-shaders.html?view=snapshot (accessed on 22nd June 2017)

[W3] https://commons.wikimedia.org/wiki/File:Arc-welding.jpg (accessed on 22nd June 2017)

[W4] http://mechanismsrobotics.asmedigitalcollection.asme.org/data/Journals/JMROA6/934846/jmr_008_02_021028_f002.png (accessed on 22nd June 2017)

[W5] https://www.intuitivesurgical.com/company/media/images/systems-s/da_Vinci_S_HD_System.jpg (accessed on 28nd June 2017)

[W6] http://www.assemblymag.com/articles/93338-whats-new-with-scara-robots?v=preview (accessed on 22nd June 2017)

nnn

ORIGINALITY REPORT

%16	%13	%6	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	en.wikipedia.org Internet Source	%5
2	www.andreasaristidou.com Internet Source	%2
3	ir.knust.edu.gh Internet Source	%2
4	andreasaristidou.com Internet Source	%1
5	Roberts, G.N.. "Intelligent ship autopilots--A historical perspective", Mechatronics, 200312 Publication	<%1
6	theaccents.org Internet Source	<%1
7	Lecture Notes in Computer Science, 2015. Publication	<%1
8	ethesis.nitrkl.ac.in Internet Source	<%1
9	Lecture Notes in Electrical Engineering, 2014. Publication	<%1

10	sofweb.phys.ocha.ac.jp Internet Source	<%1
11	"Méthodes et outils pour la biomécanique / Methods and Coah for Biomechanics", Archives of Physiology and Biochemistry, 2002 Publication	<%1
12	depositonce.tu-berlin.de Internet Source	<%1
13	Tiago Nascimento. "Coordinated Multi-Robot Formation Control", Repositório Aberto da Universidade do Porto, 2013. Publication	<%1
14	arizona.openrepository.com Internet Source	<%1
15	mydocs.epri.com Internet Source	<%1
16	doaj.org Internet Source	<%1
17	research.ijcaonline.org Internet Source	<%1
18	espace.curtin.edu.au Internet Source	<%1
19	Sohma, H.. "Quantitative reduction of type I adenylyl cyclase in human alcoholics", BBA - Molecular Basis of Disease, 19990531 Publication	<%1

20	macr.cis.ksu.edu Internet Source	<%1
21	nacomm2013.org Internet Source	<%1
22	Jha, Panchanand, Bibhuti Bhusan Biswal, and Om Prakash Sahu. "Intelligent Computation of Inverse Kinematics of a 5-dof Manipulator Using MLPNN", Lecture Notes in Computer Science, 2014. Publication	<%1
23	clp.dimi.uniud.it Internet Source	<%1
24	Chao, P.C.P.. "Intelligent actuation strategy for a three-DOF four-wire type optical pickup", Sensors & Actuators: A. Physical, 20050103 Publication	<%1
25	graphics.ucsd.edu Internet Source	<%1
26	www.i-scholar.in Internet Source	<%1
27	Yide Ma. "3D Kinematic Simulation for PA10-7C Robot Arm Based on VRML", 2007 IEEE International Conference on Automation and Logistics, 08/2007 Publication	<%1