International Journal of Robotics and Automation (IJRA)
Vol. 6, No. 1, March 2017, pp. 1~14
ISSN: 2089-4856, DOI: 10.11591/ijra.v6i1.pp1-14

Robotic Arm Movement Optimization Using Soft Computing

Surender Kumar[1], Kavita Rani[2], V. K. Banga[3]
[1,3] Department of Electronics Engineering, SECG, Mohali, Punjab, India
[2] Department of Computer Science, GNKGC, Goraya Punjab, India

Article Info

Article history:

Received Sep 20, 2016
Revised Dec 22, 2016
Accepted Jan 6, 2017

Keyword:

Artificial intelligence
Fuzzy logic
Genetic algorithm
Kinematics
Neural networks
Robotics arm

ABSTRACT

Robots are commonly used in industries due to their versatility and efficiency. Most of them operating in that stage of the manufacturing process where the maximum of robot arm movement is utilized. Therefore, the robots arm movement optimization by using several techniques is a main focus for many researchers as well as manufacturer. The robot arm optimization is This paper proposes an approach to optimal control for movement and trajectory planning of a various degree of freedom in robot using soft computing techniques. Also evaluated and show comparative analysis of various degree of freedom in robotic arm to compensate the uncertainties like movement, friction and settling time in robotic arm movement. Before optimization, requires to understand the robot's arm movement i.e. its kinematics behavior. With the help of genetic algorithms and the model joints, the robotic arm movement is optimized. The results of robotic arm movement is optimal at all possible input values, reaches the target position within the simulation time.

Corresponding Author:

V. K. Banga,
Departement of Electronics Engineering,
ACET, Amritsar,
Punjab Technical University, Punjab, India.
Email: vijaykumar.banga@gmail.com, principal@acetedu.in

1. INTRODUCTION

In the early 1960s, the industrial revolution put industrial robots in the factory to release the human operator from risky and harmful tasks. The appearance and capabilities of robots vary vastly, all robots share the features of a mechanical, movable structure under some form of autonomous control [8], [15]. The kinematic chain which is formed of links (its bones), actuators (its muscles) and joints which can allow one or more degrees of freedom. Most contemporary robots use open serial chains in which each link connects the one before to the one after it. These robots are called serial robots and often resemble the human arm. Some robots, such as the Stewart platform, use closed parallel kinematic chains. The mechanical structure of a robot must be controlled to perform tasks. The control of a robot involves three distinct phases - perception, processing and action. Sensors give information about the robot itself (the position of its joints). Using strategies from the field of control theory, this information is processed to calculate the appropriate signals to the actuators (motors) which move the mechanical structure.

The control of a robot involves:
Outer Space: Manipulative arms that are controlled by a human are used to unload the docking bay of space shuttles to launch satellites or to construct a space station
The Intelligent Home: Automated systems like home security, environmental conditions and energy usage are pre programmed to activate. This assists occupants irrespective of their state of mobility.
Exploration: Robots can monitor volcano, oceans and planetary exploration.

Military Robots: Airborne robot as drones are used for surveillance in today's modern army. In the future automated aircraft and vehicles could be used to carry fuel and ammunition or clear minefields.
Farms: Automated harvesters can cut and gather crops. Robotic dairies are available allowing operators to feed and milk their cows remotely.
The Car Industry: Robot are used in the car manufacturing welding, cutting, lifting, sorting and bending.
Hospitals: Under development is a robotic suit that will enable nurses to lift patients without damaging their backs.

2. ROBOTICS ARM

A robotic arm is a robotic manipulator having similar functions to a human arm. The links of such a manipulator are connected by joints allowing either rotational motion (such as in an articulated robot) or translational (linear) displacement. The links of the manipulator can be considered to form a kinematic chain [1], [9]. The end of the kinematic chain of the manipulator is called the end effector and it is analogous to the human hand. The end effector can be designed to perform any desired task such as welding, gripping, spinning etc., depending on the application such as welding and parts rotation and placement during assembly. Robitic arm with seven degrees of freedom as shown in Figure 1.

Figure 1. Robitic arm with seven degrees of freedom [1]

Robot arms are categorized by the number of controlled degree of freedom (DOF) they can execute. This number is equal to the sum of the DOF of each of a robot arm's individual joints. Generally these will be either Hinge joints or Pivot joints, both of which are only capable of rotation about a sing The number of DOF that a manipulator possesses is the number of independent position variables that would have to be specified in order to locate all parts of the mechanism. In other words, it refers to the number of different ways in which a robot arm can move [8]. A normal human arm is redundant in that it has seven DOF. The shoulder gives pitch, yaw and roll. The elbow allows for pitch.

The wrist allows for pitch and yaw. And the elbow and wrist together allow for Roll. In a Two Dimensional (2-D) space (like a table-top or the floor) there are three Degrees of Freedom. These include displacement along the X and Y axes, plus rotation. In a Three Dimensional (3-D) space there are six degrees of freedom. These consist of displacement along three perpendicular axes (X, Y, and Z), and rotation about those same axes.
Heave: Moving up and down
Surge: Moving forward and backward
Sway: Moving left and right Rotations
Yaw: Turning left and right flight
Roll: Tilting side to side
Pitch: Tilting forward and backward

Robotic Arm structure
Cartesian Robot/Gantry Robot: Used for pick and place application of sealant, assembly operations, handling machine tools and arc welding. It's a robot whose arm has three prismatic joints, whose axes are coincident with a Cartesian coordinator.
Cylindrical Robot: Used for assembly operations, handling at machine tools, spot welding, and handling at die-casting machines. It's a robot whose axes form a cylindrical coordinate system.
Spherical Robot/Polar Robot (such as the Unmated): Used for handling at machine tools, spot welding, die-casting, fettling machines, gas welding and arc welding. It's a robot whose axes form a polar coordinate system

SCARA Robot: Used for pick and place application of sealant, assembly operations and handling machine tools. It's a robot which has two parallel rotary joints to provide compliance in a plane.

Articulated Robot: Used for assembly operations, die-casting, fettling machines, gas welding, arc welding and spray painting. It's a robot whose arm has at least three rotary joints. It is most widely used in the industry.

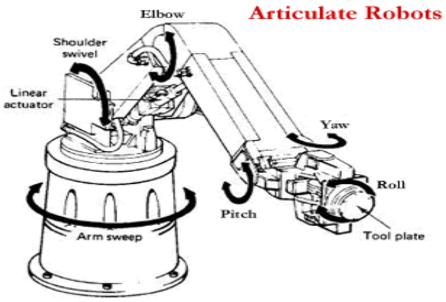

Figure 2. Articulate robot [15]

A rotary joint is a connection between two objects, the ability to rotate or have movement up to 360 degrees. Most of the time these two objects that are connected together are cylindrical. The connection gives both objects increased capabilities to perform work functions. Articulated robots as shown in Figure 2 usually have several of these connections which gives them a great deal of flexibility in performing work duties. Each joint that a robotic has represents an increase in freedom to perform tasks. There is no limit to the number of rotary joints that articulated robots can have and a robotic may have other types of joints to increase its capability even more [15], [20].

Parallel Robot: Used for mobile platform handling cockpit flight simulators. It's a robot whose arms have concurrent prismatic or rotary joints.

Anthropomorphic Robot: Shaped in a way that resembles a human hand, i.e. with independent fingers and thumbs.

3. OPTIMISATION

Optimization is of great importance for the engineers, scientists and managers and it is an important part of the design process for all disciplines. The optimal design of a machine, the minimum path for a mobile robot and the optimal placement of a foundation are all optimization problems. A constrained optimization problem has three main elements; design variables, constraints and objective function/functions. Design variables are independent variables of the Serial and Parallel Robot Manipulators–Kinematics, Dynamics, Control and Optimization objective function and can take continuous or discrete values. The ranges of these variables are given for the problems. Constraints are the functions of design variables and limit the search space. The objective function is the main function dependent on the design variables. If there is more than one objective function, the problem is called multi-objective optimization problem.

4. KINEMATICS

It (from Greek κινεῖν, kinein, to move) is the branch of classical mechanics that describes the motion of bodies (objects) and systems (groups of objects) without consideration of the forces that cause the motion. Kinematics is not to be confused with another branch of classical mechanics: analytical dynamics (the study of the relationship between the motion of objects and its causes), sometimes subdivided into kinetics (the study of the relation between external forces and motion) and statics (the study of the relations in a system at equilibrium). Kinematics also differs from dynamics as used in modern-day physics to describe time-evolution of a system [1]. Kinematics is the process of calculating the position in space of the end of a linked structure, given the angles of all the joints. It is easy, and there is only one solution.

Inverse Kinematics does the reverse. Given the end point of the structure, what angles do the joints need to be in the achieve that end point. It can be difficult, and there are usually many or infinitely many solutions. This process can be extremely useful in robotics. You may have a robotic arm which needs to grab an object as shown in Figure 3. If the software knows where the object is in relation to the shoulder, it simply needs to calculate the angles of the joints to reach it. The simplest application of kinematics is for particle motion, translational or rotational [7]. The next level of complexity comes from the introduction of rigid bodies, which are collections of particles having time invariant distances between themselves. Rigid bodies might undergo translation and rotation or a combination of both. A more complicated case is the kinematics of a system of rigid bodies, which may be linked together by mechanical joints.

Forward Kinematics: The forward kinematic is that the positions of particular parts of the model at a specified time are calculated from the position and orientation of the object, together with any information on the joints of an articulated model. If the object to be animated is an arm with the shoulder remaining at a fixed location, the location of the tip of the thumb would be calculated from the angles of the shoulder, elbow, wrist, thumb and knuckle joints. Three of these joints (the shoulder, wrist and the base of the thumb) have more than one degree of freedom, all of which must be taken into account. If the model were an entire human figure, then the location of the shoulder would also have to be calculated from other properties of the model [3], [5].

Inverse Kinematics: It will enable us to calculate what each joint variable must be if we desire that the hand be located at particular point and have a particular position. The position and orientation of the end effector relative to the base frame compute all possible sets of joint angles and link geometries which could be used to attain the given position and orientation of the end effector [1], [4].

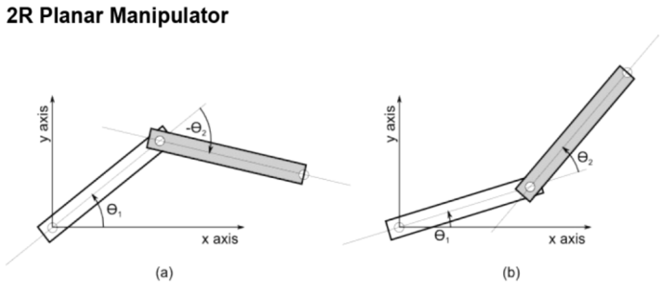

Figure 3. Two link robotic arm [4]

Forward kinematics (1-2)

$$x=l_1\cos\theta_1+l_2\cos(\theta_1+\theta_2) \quad (1)$$

$$y=l_1\sin\theta_1+l_2\sin(\theta_1+\theta_2) \quad (2)$$

Inverse kinematics (3-4)

$$x^2+y^2=l_1^2\cos^2\theta_1+l_2^2\cos^2(\theta_1+\theta_2)+2l_1 l_2\cos\theta_1\cos^2(\theta_1+\theta_2)+l_1^2\sin^2\theta_1+l_2^2\sin^2(\theta_1+\theta_2)$$
$$+2l_1 l_2\sin\theta_1\sin^2(\theta_1+\theta_2) \quad (3)$$

$$=l_1^2+l_2^2+2l_1 l_2\cos\theta_1\cos(\theta_1+\theta_2)+\sin\theta_1\sin^2(\theta_1+\theta_2) \quad (4)$$

Next we use the following equalities (5-6)

$$\sin(x\pm y)=\sin x\cos y\pm\cos x\sin y \quad (5)$$

$$\cos(x\pm y)=\cos x\cos y\pm\sin x\sin y \quad (6)$$

Therefore (7-8)

$$x^2+y^2=l_1^2+l_2^2+2l_1 l_2[\cos\theta_1\cos\theta_2-\sin\theta_1\sin\theta_2+\sin\theta_1(\cos\theta_2\sin\theta_1+\cos\theta_1\sin\theta_2)] \quad (7)$$

$$=l_1^2+l_2^2+2l_1 l_2[\cos^2\theta_1\cos\theta_2+\sin^2\theta_2\cos\theta_2]$$

$$=l_1^2+l_2^2+2\ l_1\ l_2 \cos\theta_2 \qquad (8)$$

And (9)

$$\cos\theta_2=\frac{x^2\ +y^2-{l_1}^2-{l_2}^2}{2l_1l_2} \qquad (9)$$

Here, the angle directly using the arc cos function but this function is very inaccurate for small angle s, .the typical way to avoid this accuracy is to convert further until we can use the a tan 2 function:

$$\cos^2\theta_2+\sin^2\theta_2=1 \text{ and } \sin\theta_2=\pm\sqrt{1-cos^2\theta_2}$$

the two solutions corresponding to the ‘elbow up’ and ‘elbow down ’configuration as shown in above figure and finally (10-11)

$$\theta_2=a\ \tan\ 2(\sin\theta_2\ \cos\ \theta_2) \qquad (10)$$

$$=a\ \tan\ 2\ (\pm\sqrt{1-cos^2\theta_2}.\ \cos\ \theta_2$$

$$=a\ \tan\ 2(\pm\sqrt{1-(\frac{x^2\ +y^2-{l_1}^2-{l_2}^2}{2l_1l_2}}),\frac{x^2\ +y^2-{l_1}^2-{l_2}^2}{2l_1l_2} \qquad (11)$$

for solving θ_1 we rewrite the original nonlinear equations using a change of variables as follow Figure 4 and the equation as shown in (12-15)

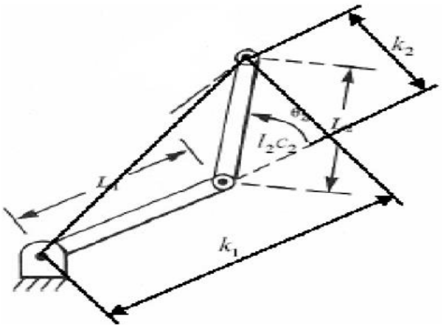

Figure 4. Mathematical expression for two link [4]

$$x=l_1 \cos\theta_1+l_2 \cos\ (\theta_1+\theta_2) \qquad (12)$$

$$y=l_1 \sin\theta_1+l_2 \sin\ (\theta_1+\theta_2) \qquad (13)$$

$$x=k_1\ \cos\theta_1+k_2\ \sin\theta_1 \qquad (14)$$

$$y=k_1\sin\theta_1+k_2\ \cos\theta_1 \qquad (15)$$

where (15-18)

$$k_1=l_1+l_2\ \cos\theta_2 \qquad (16)$$

$$k_2=l_2\ \sin\theta_2 \qquad (17)$$

the constant k_1 k_2 as shown in Figure 5

$$r=\sqrt{{k_1}^2\ +\ {k_2}^2}$$

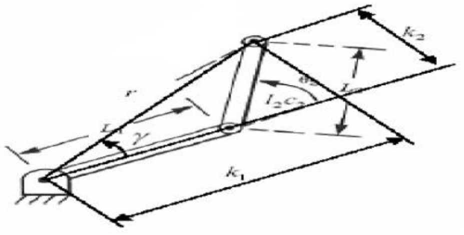

Figure 5. Mathematical expression [4]

$$'\Upsilon\text{=a tan }2(k_2, k_1) \quad (18)$$

$$K_1\text{=r cos}'\Upsilon$$

$$K_2\text{=r sin}'\Upsilon$$

Inserting into previous transformations of x and y yields (19-20).

$$X\text{=r cos}'\Upsilon \cos\theta_1\text{+r }\sin\theta_1 \sin'\Upsilon \quad (19)$$

$$y\text{= r cos}'\Upsilon \cos\theta_1\text{+r }\sin\theta_1 \sin'\Upsilon \quad (20)$$

$$\text{or}\frac{y}{r}=\sin(\theta_1+'\Upsilon)$$

$$\frac{x}{r}=\cos(\theta+'\Upsilon)$$

Apply the a tan 2 function (21-22)

$$'\Upsilon+\theta_1 \text{ a tan } 2(\frac{y}{r},\frac{y}{r})\text{=a tan }2(y,x) \quad (21)$$

$$\theta_1\text{=a tan}(y,x)\text{-a tan }2(k_2, k_1) \quad (22)$$

Mathematical model of three degree-of-freedom (3DOF) robotic system

To calculate movements in dynamic systems made up of several parts, the main approach is to calculate possible movements with the aid of mathematical models. At the same time it is necessary to understand both the mechanics and the physical aspects. A vertical articulated robotic arm with 3 links as shown in Figure 6 having length l_1, l_2, and l_3 respectively, is considered which has a three degree-of- freedom [2], [3]. In three degree-of-freedom robotic arm the inverse kinematics equations are as below (23-24):

$$x=l_1 \cos\theta_1+l_2 \cos(\theta_1+\theta_2)+l_3 \cos (\theta_1+\theta_2+\theta_3 \quad (23)$$

$$y=l_1 \sin\theta_1+l_2 \sin(\theta_1+\theta_2)+l_3 \sin (\theta_1+\theta_2+\theta_3) \quad (24)$$

$$\theta=\theta_1+\theta_2+\theta_3$$

Knowing the arm link lengths l_1, l_2, and l_3 for position (x, y) we had calculated the values of joint angles θ_1, θ_2, θ_3 [20].

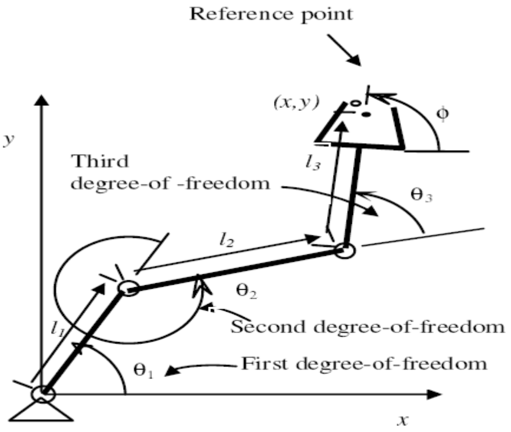

Figure 6. Three link robotic arm [20]

4.1. Modeling of Robotics Arm

The inverse kinematics problem is much more interesting and its solution is more useful. “Given the desired position of the robot's hand, what must be the angles at all of the robots joints?” Humans solve this problem all the time without even thinking about it. How most robots have to solve the problem.

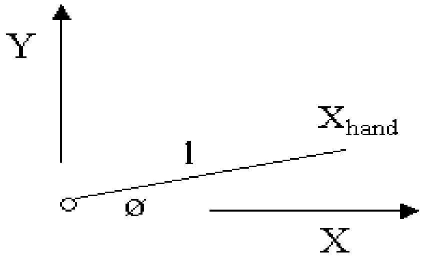

Figure 7. Single link manipulator [20]

The Figure 7 above is a schematic of a simple robot lying in the X-Y plane. The robot has one link of length l and one joint with angle Ø. The position of the robot's hand is Xhand. The inverse kinematics problem (at the position level) for this robots as follows: Given Xhand what is the joint angle Ø? We'll start the solution to this problem by writing down the forward position equation, and then solve for Ø.

Xhand=lcosØ (forward position solution)

CosØ=Xhand / l

Ø=cos-1(Xhand/l)

To finish the solution let's say that this robot's link has a length of 1 foot and we want the robot's hand to be at X= .7071 feet. That gives:

Ø=cos-1(.7071)=+/- 45 degrees

There are two solutions to the inverse kinematics problem: one at plus 45 degrees and one at minus 45 degrees! The existence of multiple solutions adds to the challenge of the inverse kinematics problem. Typically we will need to know which of the solutions is correct. All programming languages that I know of supply a trigonometric function called A Tan 2 that will find the proper quadrant when given both the X and Y arguments: Ø=A Tan 2(Y/X). There is one more interesting inverse kinematics problem. Two link manipulator as shown in Figure 8.

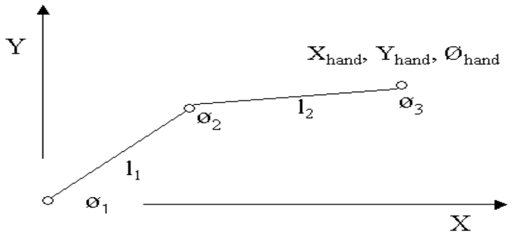

Figure 8. Two link manipulator [20]

You may have to use your imagination a bit, but the schematic above is the planar part of the SCARA robot we discuss in the industrial robots section. Here's the statement of the inverse kinematics problem at the position level for this robot:

Given: Xhand, Yhand, Øhand

Find:Ø1, Ø2, and Ø3To aid in solving this problem, I am going to define an imaginary straight line that extends from the robot's first joint to its last joint as follows:

B: length of imaginary line

q1: angle between X-axis and imaginary line

q2: interior angle between imaginary line and link l1

Then we have (25-27):

B2 =Xhand2+Yhand2 (25)

q1=A Tan 2 (Yhand/Xhand) (26)

q2=a cos [(l12-l22+B2)/2l1B] (27)

Ø1=q1+q2

4.2. CO-Ordinate Frame of 5 DOF Robotic Arm

Coordinate frames for the AL5B robotic arm are assigned as shown in Figure 9.They are established using the principles of the Denavit-Hartenbarg (D-H) convention. For the kinematic model of 5 dof robotic arm first we have to assign frame to each link starting from base (frame 0) to end-effector (frame 5).

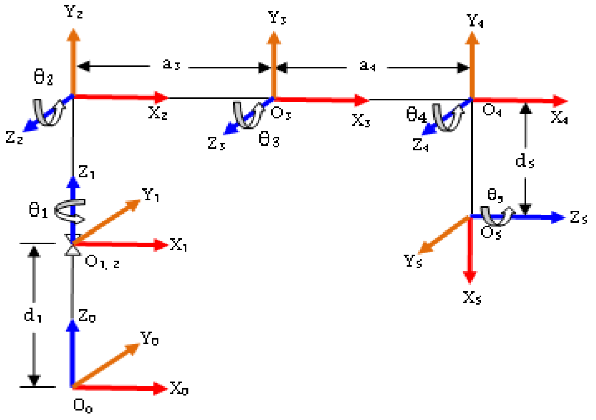

Figure 9. Coordinate frame assignment [19]

4.3. Mathematical and Kinematic Modeling of 5 DOF Robotic Arm

Robot manipulator is named according to number of DOF, which refers to the number of joints. The robot manipulator arm has 5 joints, which mean the robot has 5 DOF. The kinematics robot manipulator is derived by using Denavit-Harterberg (DH) representation. In this convention, each homogeneous transformation A_i is represented as a product of four basic transformations as shown in (28-29).

$$T_{e} = R_Z(\Theta_i)\, D_Z(d_i)\, D_x(a_i)\, R_x(\alpha_i) \tag{28}$$

$$\begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & \alpha_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & \alpha_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{29}$$

where R_X and R_Z present rotation, D_X and D_Z denote translation, and $C\theta_i$ and $S\theta_i$ are the short hands of $\cos\theta_i$ and $\sin\theta_i$, respectively. The forward kinematics of the end-effectors with respect to the base frame is determined by multiplying all of the T_5 matrices. The four quantities θ_i, a_i, d_i, α_i are parameters associated with link i and joint i. The four parameters a_i, α_i, d_i, and θ_i are generally given the names link length, link twist, link offset, and joint angle respectively.

An alternative representation of T_e can be written as (30):

$$T_e = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{30}$$

where r_{kj}'s represent the rotational elements of transformation matrix (k and j=1, 2 and 3). p_x, p_y, and p_z denote the elements of the position vector. For a five robotic arm, the position and orientation of the end-effector with respect to the base is given by (31-32)

$${}^{0}T_{5} = {}^{0}T_{1}\,{}^{1}T_{2}\,{}^{2}T_{3}\,{}^{3}T_{4}\,{}^{4}T_{5} \tag{31}$$

$${}^{0}T_{1} = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{1}T_{2} = \begin{bmatrix} C_2 & 0 & S_2 & 0 \\ S_2 & 0 & -C_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{2}T_{3} = \begin{bmatrix} C_3 & -S_3 & 0 & a_3 * C_3 \\ S_3 & C_3 & 0 & a_3 * C_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{3}T_{4} = \begin{bmatrix} C_4 & -S_4 & 0 & a_4 * C_4 \\ S_4 & C_4 & 0 & a_4 * C_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{4}T_{5} = \begin{bmatrix} C_5 & 0 & 0 & 0 \\ S_5 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{0}T_{5} = {}^{0}T_{1}\,{}^{1}T_{2}\,{}^{2}T_{3}\,{}^{3}T_{4}\,{}^{4}T_{5}$$

$$ {}^{0}T_5 = \begin{bmatrix} C_{12}C_{345} & S_{12} & C_{12}S_{345} & S_{12}d_5 + C_{12}a_4C_{34} + C_{12}a_3C_3 \\ S_{12}C_{345} & -C_{12} & S_{12}S_{345} & -C_{12}d_5 + S_{12}a_4C_{34} + S_{12}a_3C_3 \\ S_{345} & 0 & -C_{345} & a_4S_{34} + a_5S_5 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ T_e = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (31) $$

$^{o}T_5 = T_e$

Where,

$r_{11}=C_{12}C_{345}$

$r_{12}=S_{12}$

$r_{13}=C_{12}S_{345}$

$r_{21}=S_{12}C_{345}$

$r_{22}=-C_{12}$

$r_{23}=S_{12}S_{345}$

$r_{31}=S_{345}$

$r_{32}=0$

$r_{33}=-C_{345}$

$p_x=S_{12}d_5+C_{12}a_4C_{34}+C_{12}a_3C_3$

$p_y=-C_{12}d_5+S_{12}a_4C_{34}+S_{12}a_3C_3$

$p_z=a_4S_{34}+a_5S_5+d_1$

INVERSE KINEMATICS ANALYSIS

The following equations will be used to obtain the solution for the inverse Kinematics problem.

${}^{0}T_5={}^{0}T_1{}^{1}T_2{}^{2}T_3{}^{3}T_4{}^{4}T_5=T_e$

Inverse kinematics solution for the first joint as a function of the known elements of T_e, the link transformation inverses are premultiplied as follows:

$X_1=[({}^{0}T_1)^{-1}]*{}^{0}T_5={}^{1}T_2*{}^{2}T_3*{}^{3}T_4*{}^{4}T_5={}^{1}T$

Similarly, to find the other variables, the following equations are obtained in similar manner.

$X_2=[({}^{0}T_1)({}^{1}T_2)]^{-1}*{}^{0}T_5={}^{2}T_3*{}^{3}T_4*{}^{4}T_5={}^{2}T_5$

$X_3=[({}^{0}T_1)({}^{1}T_2)({}^{2}T_3)]^{-1}*{}^{0}T_5={}^{3}T_4*{}^{4}T_5={}^{3}T_5$

$X_4=[({}^{0}T_1)({}^{1}T_2)({}^{2}T_3)({}^{3}T_4)]^{-1}*{}^{0}T_5={}^{4}T_5={}^{4}T_5$

By solving these equations, we can calculate the values of Θ_1, Θ_2, Θ_3, Θ_4, and Θ_5.
So, the values will be (32-36)

$$\Theta_1=\Theta_{12}-\Theta_2 \quad (32)$$

$$\Theta_2=\tan^{-1}(p_y/p_x) \quad (33)$$

$$\Theta_3=\tan^{-1}(S_3/C_3) \quad (34)$$

$$\Theta_4=\tan^{-1}(S_4/C_4) \pm\sqrt{1+C_4^2} \quad (35)$$

$$\Theta_5=\Theta_{345}-\Theta_{34} \quad (36)$$

Using these value the values of all the joint angles i.e. Θ_1, Θ_2, Θ_3, Θ_4, Θ_5 of 5 DOF robotic arm will be determined. This is solution of denavit-hartenberg representation.

5. SOFT COMPUTING

Soft computing (SC) is an evolving collection of methodologies to exploit tolerance for imprecision, uncertainty and partial truth to achieve robustness, tractability and low cost. SC provides an attractive opportunity to represent the ambiguity inhuman thinking with real life uncertainty. Fuzzy logic (FL), neural networks (NN), and evolutionary computation (EC) are the core methodologies of soft computing. While an analytical technique is difficult, moving an arm in the presence of an obstacle can be instinctively performed b a child. Neuro-fuzzy systems excel in using sample data to determine an input-output relationship. The field of neuro-fuzzy technology has gone in many directions. The neuro-fuzzy technique replaces the traditional fuzzy logic system with a multilayer back propagation neural network.

This type of system is beneficial for several reasons. While it is true that a child is able to move an arm around an obstacle to reach a desired goal, that ability is intuitive. Putting the instructions for performing such a task into a neat, fuzzy logic, IF/THEN rule base is not easy [13]. Thus, there is a necessity for the neural network to learn the rules. The fuzzifiers and defuzzifiers necessary for any fuzzy system provide an interface between an expert's control of a simulated arm and the neural network. GAs are tools on probabilistic and casualty, not necessarily they will have the same type of evolution when applied to the same problem. GAs are slower because they are tools of evolution and not for specific optimizations [12]. They are simpler, easy programming, and demand less mathematics complexity to describe the process to be optimized. ANN and fuzzy logic techniques required more information regarding system and more mathematics as compare to GA. The great advantage observed in the GAs are tool of easy application and in robotics they could be thoroughly used to do several tasks, needing for that only small description of the problem.

6. GENETICS ALGORITHM

Characteristics of present computer methods inspired by biological evolution are classified as evolutionary computation. Evolutionary computation is the name given to a collection of algorithms based on the evolution of a population toward a solution for a specific problem. These algorithms can be used successfully in many different applications that require the optimization of a certain multi-dimensional function. The population of possible solutions evolves from one generation to the next, ultimately arriving at a satisfactory solution to the problem. These algorithms differ in the way anew population is generated from the present one, and in the way the members are represented within the algorithm.

The three main elements of evolutionary computation are: Evolution Algorithms (EA); Genetic Programming (GP); Genetic Algorithms (GA). Each of these three techniques imitates the processes observed in natural evolution, and provides efficient search results [7]. A Fuzzy Logic Controller (FLC) is viewed as an individual. A population includes a group of FLCs. The running of the robot with the FLCs is the evaluation process. As the antecedents of an FLC are pre-defined, only the FLC consequences are encoded as chromosomes. There are M rules in one FLC in one FLC. Therefore, one chromosome has M genes. The first gene corresponds to the first rule’s consequence. Each gene could be one of fuzzy singletons ck and illustrated in Figure 10.

C^k_1	C^k_2	…………	C^k_M

Figure 10. Chromosomes

The operations used in the GA include:
***Initialization*:** The first generation is initialized randomly. Each gene in each chromosome is chosen from the *K* fuzzy singletons evenly.
***Reproduction*:** The best individual in current generation is automatically copied into next generation.
Selection: Individuals are copied into next generation as their offspring according to their fitness values. The individuals with higher fitness values have more offspring than those with lower fitness values.
***Crossover*:** The crossover will happen for two individuals in offspring with the crossover probability *pc*. One point crossover is used to exchange the genes.
***Mutation*:** The mutation is taken for one gene of an offspring with the mutation probability *pm*. The operator randomly chooses one fuzzy singleton from the allowed set to replace the current gene.
The results showed that it is feasible to use GA learning because the learning task can be decomposed into the learning of individual behaviors

7. FUZZY AND EXPERT SYSTEM

Fuzzy Logic FL and Expert System ES are well established as useful technologies that complement each other in powerful hybrid system. Hybrid intelligent systems are now part of the repertoire of computer systems developers and important research mechanisms in the study of Artificial Intelligent. The integration of ES and FL has proven to be a way to develop useful real-world applications, and hybrid systems involving robust adaptations. In order to reach a goal, learning vehicles rely on the interaction with their environment to extract information. ES and FL have been recently recognized to improve the learning and adaptation where information is inaccurate, uncertain and imprecise. Particularly, the use of this integration (FL and ES) is necessary to bring Intelligent Autonomous Vehicle (IAV) behavior near the human one in recognition, learning, decision-making, and action. Thus, several integrations of FL and ES based navigation approaches have been developed. The interest in FL and ES aims to understand principles of the human thinking and to build machines that are able to perform complex tasks requiring massively parallel computation. Essentially, this approach deals with cognitive tasks such as learning, adaptation, generalization and optimization.

Fuzzy Logic*:* fuzzy model is that it is relatively simple to construct and is in itself a simple structure. It does not require the modeler to have a deep mathematical insight, but relies more on experience of the process. Its greatest value must be, therefore, in those areas where such qualitative process knowledge is predominant and essential for understanding. The theory has shown that the fuzzy models can be successfully constructed; the overall concept needs a considerably more detailed investigation before its true worth can be evaluated. Many of the successful application of fuzzy logic have shown the importance of rule bas fuzzy control. Control is meant in the most general sense: it includes actual closed loop control, expert systems, and all kinds of man machine systems where the decision of the human 'component' is supported or modified by the conclusion obtained by the application of approximate reasoning on the set of available rules. [14], [16], [18].

Expert System: An ES is a computer program that functions, is in a narrow domain, dealing with specialized knowledge, generally possessed by human experts. ES is able to draw conclusions without seeing all possible information and capable of directing the acquisition of new information in an efficient manner.

Neuro-fuzzy Techniques: From a historic perspective, neuro-fuzzy systems became the first representative of hybridization in soft computing. Neuro-fuzzy systems incorporate the knowledge representation of fuzzy logic with the learning capabilities of artificial neural networks. Both methodologies are concerned with the design of intelligent systems albeit from different directions. The power of neural networks stems from the distributed processing capability of a large number of computationally simple elements. In contrast fuzzy logic is closer related to reasoning on a higher level. Pure fuzzy systems do not possess the capabilities of learning, adaptation or distributed computing that characterize neural networks. On the other hand, neural networks lack the ability to represent knowledge in a manner comprehensible to humans, a key feature of fuzzy rule based systems.

Neuro-fuzzy systems bridge the gap between both methodologies ,as they synthesize the adaptation mechanisms of neural networks with the symbolic components of fuzzy inference systems, namely membership functions, fuzzy connectives, fuzzy rules and aggregation operators. Ahrns et al apply neuro-fuzzy control to learn a collision avoidance behavior. Their approach relies on reinforcement learning for behavior adaptation. The learner incrementally adds new fuzzy rules as learning progresses and simultaneously tunes the membership functions of the fuzzy RBF-network. Godjavec et al present a neuro-fuzzy approach to learn an obstacle avoidance and wall-following behavior on a small size robot. Their scheme allows it tossed an initial behavior with expert rules, which are refined throughout the learning process. During training the robot is controlled either by a human or a previously designed controller.

The recorded state-action pairs serve as training examples during supervised learning of neuro-fuzzy control rules. The robot successfully imitates the demonstrated behavior after 1500 iterations Ye et al propose a neuro-fuzzy system for supervised and reinforcement based learning of an obstacle avoidance behavior. The scheme follows a two-stage tuning approach; in a first phase supervised learning determines the coarse structure of input-output membership functions. The second reinforcement learning stage fine-tunes the output membership functions [6], [18], [16].

Swarm Intelligence Technique in Robotics: Intelligent robot whose behavior is neither random nor predictable. Intelligent swarm is a group of non-intelligent robots forming a group, an intelligent robot. In other words, a group of “machines” capable of forming “ordered” material patterns “unpredictably” [8]. SI systems are typically made up of a population of simple agents interacting with one another and with their environment. The group of individuals acting in such a manner is referred to as a swarm. Problem-solving behavior that emerges from such interactions is called swarm intelligence. The two best known SI algorithms are: Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO).

Particle Swarm Optimization (PSO): It was originally inspired by the crowd behavior of birds. In terms of this bird flocking analogy, a particle swarm optimizer consists of a number of particles, or birds, that fly around and search sky, for the best location. The individuals communicate either directly or indirectly with one another search directions.

The Ant Colony Optimization (ACO): It represents the model of the collective act of searching food behavior of ants. In the Binary Bridge experiment, two ants are taking the paths of different length from the home to a food source. The ant that will return first to the source is the one taking the shorter path. As more and more ants start to follow the trail of higher pheromone concentration, a positive feedback loop is created until all the ants follow the shortest path [9].

8. CONCLUSION

The presence of several optimization attributes for a physical system of higher order manipulator, soft computing techniques are alternatives to find the solutions of kinematics problem. Soft computing approaches are more preferable over conventional methods of problem solving, for problems that are difficult to describe by analytical or mathematical models. Autonomous robotics is such a domain in which knowledge about the environment is inherently imprecise, unpredictable and incomplete. Therefore, the features of fuzzy control, neural networks and evolutionary algorithms and swam intelligence are of particular benefit to the type of problems emerging in behavior based robotics and multi-agent robotics. Soft computing techniques contribute to one of the long term goal in robotics, to solve the problems that are unpredictable and imprecise namely in unstructured real-world environments.

REFERENCES

[1] K. S. Fu, et al., Robotics: Control, Sensing, Vision and Intelligence, *Tata McGraw-Hill Education,* 1987.

[2] P. Lozano, et al., “A Simple Motion-Planning Algorithm for General Robot Manipulators”, *IEEE Journal of Robotics and Automation,* 3(3), pp. 224–238, 1987.

[3] S. Cameron, “Obstacle Avoidance and Path Planning”, *International Journal of Industrial Robot*, 21(5), pp. 9-14, 1994.

[4] Saotti Alessandro, “Fuzzy Logic in Autonomous Robot Navigation a Case Study”, Nov 1995.

[5] J. M. Ahuactzin, *et al.*, “The kinematic Roadmap: a Motion Planning Based Global Approach for Inverse Kinematics of Redundant Robots” *IEEE Transactions on Robotics and Automation*, 15(4), pp. 653-669, 1999.

[6] P. Bonissone Piero, *et al.*, “Hybrid Soft Computing Systems: Industrial and Commercial Applications”, *Proc. IEEE,* vol. 87(9), Sept. 1999.

[7] M. Galicki, “Time-optimal Controls of Kinematic ally Redundant Manipulators with Geometric Constraints”, *IEEE Transactions on Robotics and Automation,* 16(1), pp. 89-93, 2000.

[8] Deb S. R. Deb, “Robotics Technology and Flexible Automation”, *Tata Mc-Graw Hill,* 2002.

[9] R. J. Schilling, “Fundamental of Robotics: Analysis and Control”, *Prentice Hall,* India Pvt. Ltd., 2002.

[10] Frank Hoffmann, “An Overview on Soft Computing in Behavior Based Robotics” 2002.

[11] Gu Dongbing, *et al.*, “GA-based Learning in Behavior Based Robotics”, *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, Kobe*, Japan, 16-20 July 2003.

[12] M. Gemeinder, *et al.*, “GA-Based Path Planning for Mobile Robot Systems Employing an Active Search Algorithm”, *Elsevier Applied Soft Computing,* 3(2), 149-158, 2003.

[13] Y. Tang, *et al.,* “Robust Fuzzy Control of Mechanical Systems”, *IEEE Transactions on Fuzzy System,* 11(3), pp. 411-418, 2003.
[14] DuskoKatic, *et al.,* “Genetic Algorithms in Robotics”, *International Series on Microprocessor-Based and Intelligent System Engineering, v*ol. 25, 2003.
[15] J. J. Craig, “Introduction to Robotics: Mechanics and Control”, *Pearson Education,* Asia, 2004.
[16] Beni Gerardo, “From Swarm Intelligence to Swarm Robotics”, *Swarm Robotics Springer,* Verlag Berlin Heidelberg, 2005.
[17] C. Aleksandar, *et al.*, “Swarm Intelligence and Its Applications in Swarm Robotics”, *6th WSEAS Int. Conference on Computational Intelligence, Man-Machine Systems and Cybernetics*, Tenerife, Spain, Dec. 14-16, 2007.
[18] O. Hachour, “The proposed Fuzzy Logic Navigation Approach of Autonomous Mobile Robots in Unknown Environments”, *International Journal of Mathematical Models and Methods in Applied Sciences*, 2009.
[19] V. K. Banga, et al., “Fuzzy Genetic Optimal Control for Robotics System”, *International Journal of Physical Sciences,* 6(2), pp.204-212, 2011.
[20] V. K. Banga, et al., “Modeling and Simulation of Robotics Arm Movement using Soft Computing”, W*orld Academy of Science*, 75, pp. 614-620, 2011.

BIOGRAPHIES OF AUTHORS

Surender Kumar was worked as a Assistant Lecturer, Lovely Professional University, Phagwara Punjab, India. He obtained his B. Tech (Control and Instrumentation and Engineering) from National Institute of Tecnology (NIT) under Punjab Technical University, Jalandhar, Punjab, India. M.Tech (Control and Instrumentation and Engineering) from Indian Institute of Technology (IIT) Roorkee, India.

Kavita Rani was worked as a Assistant Lecturer and Head of the Department of Computer Science, Guru Nanak Khalsa Girls College, Baba Sang-Dhesian Goraya, Punjab, India under Guru Nanak Dev University, Amritsar Punjab India. She obtained his B. Tech (Computer Science and Engineering) from Beant College of Engineering and Technology under Punjab Technical University, Jalandhar, Punjab, India. M.Tech (Computer Science and Engineering) from Lovely Professional University, Phagwara Punjab, India.

Vijay Kumar Banga is working as a Professor and Principal of the Amritsar College of Engineering and Technology, Amritsar, Punjab, India. He obtained his B. E (Electronics and Instrumentation Engineering) from Punjabi University, Patiala, Punjab, India, M. Tech (Electronics and Instrumentation) from Panjab University, Chandigarh, India and Ph. D. in Electronics (Artificial Intelligence) from Thapar University, Patiala., India. Presently, he has 16 years of research and UG & PG teaching experience. He has 200 research papers to his credit in various international journals and conferences. He is member of Board of Studies in Electronics and Communication Engineering of IKG Punjab Technical University, Kapurthala, Punjab, India. He is member of ISTE, IEEE, IACSIT, WASET, WSEAS etc. His areas of research and interest include Robotics, Artificial Intelligence, automated control systems and Genetic algorithms.