Kinematics Analysis of Manipulator using Soft Computing Technique

Ashwani K., Vijay K., Darshan K.

***Abstract:** In this paper, the analysis and modeling of six joint axes of a robotic arm having three DOF spherical arm and three DOF spherical wrist have been done to solve the kinematics and inverse kinematics. Kinematics provides the rational explication of a robotic manipulator. For the analysis of industrial robotics manipulator a particular type of kinematics model is required. The Denavit Hartenberg criterion has been used to solve the kinematics equations. MATLAB, Firefly Algorithm (FFA) and Roboanalyzer have been used to get the home position and differences in error at different values of six-DOF manipulator. Error can be optimized to as low as $10^{-17}$with the firefly algorithm.*

***Keywords:** Robotic arm, MATLAB, Firefly, Roboanalyzer.*

I. INTRODUCTION

Robotic systems are not simply robots, but can also be used with other devices and systems to perform different tasks. Today's Robots are used in various applications where human work can be replaced and automated. There are various applications where a robotic arm is used in painting, carpentry and hardware verification [1]. Due to developments in manufacturing technologies, micro-assembly can be done easily and finds applications in semiconductor processing and assembly, agricultural, aeronautics, railways, energy industries, the aerospace industry and precision material processing [2], [3].

Today, Robots have been used not only in the manufacturing industries, but also finds applications in wiping, discerning and extricating operations [6]. Forward and inverse kinematics studies of industrial robots have been presented in a number of papers in the last five decades. Using Denavit-Hartenberg conventions and associated homogenous transformation matrix presents only in few papers shows that kinematics modeling can be directly implemented on real-time industrial robotic arms [5], [7], [25].

Industrial robots can be employed to carry out undiversified tasks that require more accuracy and swiftness which cannot be easily achieved by human beings. By using these robots the quality of products and efficiency has been improved in manufacturing [11].

In the robotic arm dexterity and flexibility can be managed with three DOF for positioning the gripper at a particular location and the remaining for orientation assimilation. The first three joints and the last three joints that find position and orientation are used to control and teaching the robotic arm [7], [8], [21].

It was very difficult to find the trajectory planning and analysis of the kinematics solution of higher DOF and multi-link manipulators as they show variations for very small change in parameters [10], [22].

Various swarm algorithms have been used to solve the kinematics of an industrial manipulator. Firefly algorithm [27]-[32] which is based on swarm intelligence is used to optimize the Euclidean distance and the absolute error.

II. KINEMATICS

Robots may be moved relative to different coordinate frames. Robot motions are accomplished in the following three coordinate frames [1].

1. World Reference Frame: Here Joints move simultaneously to give motions along the x, y, and z axes as shown in fig.1.

2. Joint Reference Frame: shows movements of individual joints as shown in fig.2.

3. Tool Reference Frame: shows movements of the hand relative to a frame of the robot's hand. Unlike the world frame, the local tool frame moves with the robot as shown in fig.3.

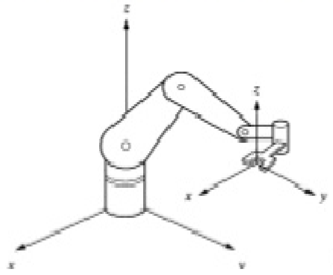

Fig.1: World Reference Frame [1]

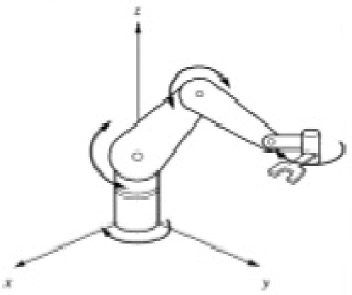

Fig.2: Joint Reference Frame [1]

Revised Manuscript Received on October 15, 2019
***** Correspondence Author

Ashwani K.*, Department of ECE, YCOE, Punjabi University Guru Kashi Campus, Talwandi Sabo, Bathinda, India. Email: singla_ash2001@yahoo.co.in

Vijay K., Department of ECE, Amritsar College of Engineering & Technology, Amritsar, India. Email: vijaykumar.banga@gmail.com

Darshan K., Depatment of ME, Beant College of Engineering & Technology, Gurdaspur, India. Email: darshanjind@gmail.com

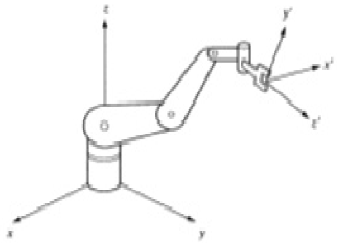

Fig.3: Tool Reference Frame [1]

The Homogeneous transformation that gives the reciprocity in the joint variables & the location, with the orientation of a robotic arm, can also be referred as Kinematic transformations [6], [12], [19, [20].

Kinematics is of two types:

a. Direct kinematics

b. Inverse kinematics

Denavit Hartenberg gave four independent parameters to derive a formula for the rotary and prismatic joints of a robotic arm. These parameters are called D-H parameters [12], [15], [24].

The four parameters for connecting reference frames to other frames of a robotic arm are as follows:

- d_i - distance from the previous frame to the perpendicular convergence of next frame x_i through z_{i-1} [7].
- a_i - Offset distance between the common perpendicular of axes $\mathbf{z}_{i-1}$ and $\mathbf{z}_i$ [17].
- θ_i - Joint angle
- α_i – offset angle

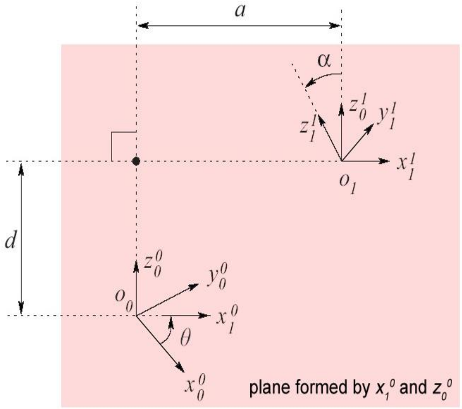

Fig.4: D-H Parameters for Frames [17]

For the multiplication of two matrices, there should be the same number of rows and column. The homogeneous transformation matrix is given by [1], [7]:

$$\boldsymbol{T} = \begin{bmatrix} nx & sx & ax & px \\ ny & sy & ay & py \\ nz & sz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

Here n, s, a, and p are the elements of rotation and position matrices for x, y, and z-axis. Three basic rotation matrices about x, y or z-axis for an angle theta is [6] as follows:

$$\mathrm{R}_{x,theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(theta) & -s(theta) \\ 0 & s(theta) & -c(theta) \end{bmatrix} \qquad (2)$$

$$\mathrm{R}_{y,theta} = \begin{bmatrix} c(theta) & 0 & s(theta) \\ 0 & 1 & 0 \\ -s(theta) & 0 & c(theta) \end{bmatrix} \qquad (3)$$

$$\mathbf{R}_{z,theta} = \begin{bmatrix} c(theta) & -s(theta) & 0 \\ s(theta) & c(theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (4)$$

Here c(theta) and s(theta) are the sine and cosine of the joint angle theta. Each homogeneous transformation $\mathbf{T_i}$ for an ith frame can be written as:

$$\boldsymbol{T}_i = \mathbf{R}_{\mathbf{z,thetai}} * \mathbf{Trans}_{\mathbf{z,di}} * \mathbf{D}_{\mathbf{x,a(i-1)}} * \mathbf{R}_{\mathbf{x,\alpha(i-1)}} \qquad (5)$$

The matrices for a given manipulator can be obtained using matrix T [1], [15] for any angle θ using (6):

$$T = \begin{bmatrix} c(\theta) & -s(\theta) & 0 & a_{i-1} \\ s(\theta)*c\alpha_{i-1} & c\theta*c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}*d_i \\ s\theta*s\alpha_{i-1} & c\theta*s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}*d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (6)$$

$$\boldsymbol{T}_{\boldsymbol{i}}^{\boldsymbol{i-1}} = \boldsymbol{T}_1^0 * \mathbf{T}_2^1 * \mathbf{T}_3^2 * \mathbf{T}_4^3 * \mathbf{T}_5^4 \ldots\ldots\ldots \mathbf{T}_n^{n-1}; \qquad (7)$$

2.1 Stanford Robotic Arm

Victor Scheinmann (1969), designed the first classic robot, called “Stanford Arm”, has been used for computer control and computations.

The first three joints, two revolute and one prismatic joint constitute the spherical configuration (RRP) and last three (RRR) joint motions roll (joint4), pitch (joint5) and roll (joint6) respectively constitute the spherical wrist, which orients the end-effector of the arm. The wrist configuration is known as “Euler Wrist”. Here P is used to represent prismatic joint and R for revolute joints.

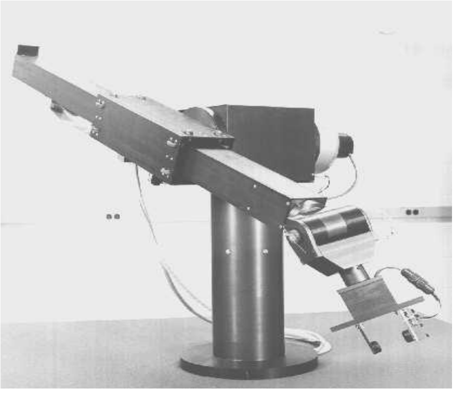

Fig.5: Stanford manipulator [16]

Consider a 6 DOF manipulator having five revolute joints and one prismatic joint. There are seven frames having different coordinates by using the right-hand coordinate rule as shown below in the fig.6.

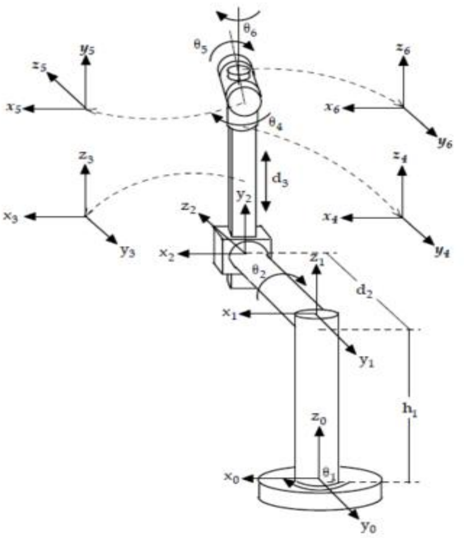

Fig.6: Coordinate frames

The zero position of the manipulator is shown in fig.7 for theta1 theta2 at 0^0.

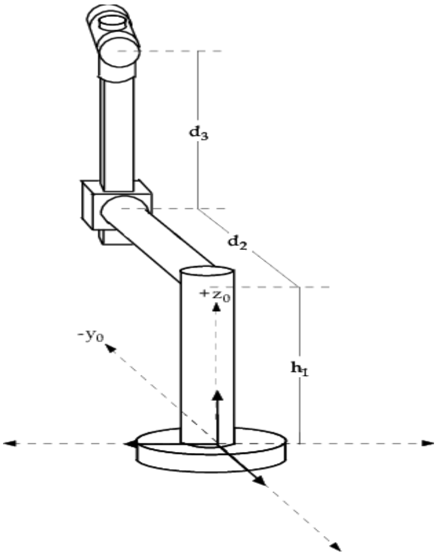

Fig.7: Joint offset (d2, d3) and height (h1) [17]

Table- I: DH Convention [6], [17]

Links	DH Parameters			
	theta$_i$	**a$_{i-1}$**	**α_{i-1} (degree)**	**d$_i$**
J1	Theta1	zero	zero	h1
J2	Theta2	zero	pi/2	d_2
J3	Zero	zero	- pi/2	d_3
J4	Theta4	zero	zero	zero
J5	Theta5	zero	pi/2	zero
J6	Theta6	zero	- pi/2	zero

Here the values are calculated from the coordinate frame using DH representation as shown in Table-1 [6], [17].

The DH matrices can be obtained using the (6) and the matrices are as follows:

$$T_1^0 = \begin{bmatrix} c(theta1) & -s(theta1) & 0 & 0 \\ s(theta1) & c(theta1) & 0 & 0 \\ 0 & 0 & 1 & h1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$T_2^1 = \begin{bmatrix} c(theta2) & -s(theta2) & 0 & 0 \\ 0 & 0 & -1 & -d2 \\ s(theta2) & c(theta2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$T_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$T_4^3 = \begin{bmatrix} c(theta4) & -s(theta4) & 0 & 0 \\ s(theta4) & c(theta4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$$T_5^4 = \begin{bmatrix} c(theta5) & -s(theta5) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s(theta5) & c(theta5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

$$T_6^5 = \begin{bmatrix} c(theta6) & -s(theta6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s(theta6) & -c(theta6) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

The Transformation matrix T_6^0 can be calculated by multiplying the matrices T_1^0 to T_6^5 as shown in (14)

$$\mathbf{T}_6^0 = \mathbf{T}_1^0 * \mathbf{T}_2^1 * \mathbf{T}_3^2 * \mathbf{T}_4^3 * \mathbf{T}_5^4 * \mathbf{T}_6^5 \quad (14)$$

Here c1,c2 and s1,s2 are the sine and cosines for angles 1 and 2 respectively. Similarly, other angles are

$$\cos(theta4) = c4, \sin(theta4) = s4; \quad (15)$$

$$\cos(theta5) = c5, \sin(theta5) = s5; \quad (16)$$

$$\cos(theta6) = c6, \sin(theta6) = s6; \quad (17)$$

The forward kinematics equations after solving the T_6^0 matrix are given by

$$px = d2s1 - d3c1s2; \quad (18)$$

$$py = -d2c1 - d3s1s2; \quad (19)$$

$$pz = h1 + d3c2; \quad (20)$$

$$nx = -c6(c5(s4s1 - c4c2c1) + s5s2c1 - s6s4c2c1 + s1c4; \quad (21)$$

$$ny = \big(c6(c5(c2c4s1 + c1s4) + s5s2s1 + s6 - s4c2s1 + c4c1; \quad (22)$$

$$nz = (c6(s5c2 + s2c4c5) - s2s4s6); \quad (23)$$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

$$\mathrm{sx} = \big(\mathrm{s6c5(s1s4 - c4c1c2)} + \mathrm{c1s2s5)-c6s4c2c1+s1c4;} \qquad (24)$$

$$\mathrm{sy} = \mathrm{(c6(c1c4 - c2s1s4)} - \mathrm{s6c5c1s4 +c2c5c4s1^{-}s5s1s2);} \qquad (25)$$

$$\mathrm{sz} = \mathrm{(-s6(c2s5 + c4c5s2) - c6s2s4);} \qquad (26)$$

$$\mathrm{ax} = \mathrm{s1s4s5 - c4s5c2c1 - c1c5s2;} \qquad (27)$$

$$\mathrm{ay} = \mathrm{(-s5(c2c4s1 + c1s4) - s1c5s2);} \qquad (28)$$

$$\mathrm{az} = \mathrm{(-c4s2s5 + c2c5);} \qquad (29)$$

The location of the end-effector can be evaluated from the above equations if the values of the joint variables have been defined properly.

At the home position (theta1 = theta2 = theta4 = theta5 = theta6 = 0^0 and d3 = L3), assuming L3 has a minimum size of prismatic link, the location of the robotic arm can be computed as:

$$\mathrm{T(home)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -d2 \\ 0 & 0 & 1 & h1+L3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (30)$$

Using Roboanalyzer, 3-D view of a Stanford manipulator at home position has been shown in Fig.8.

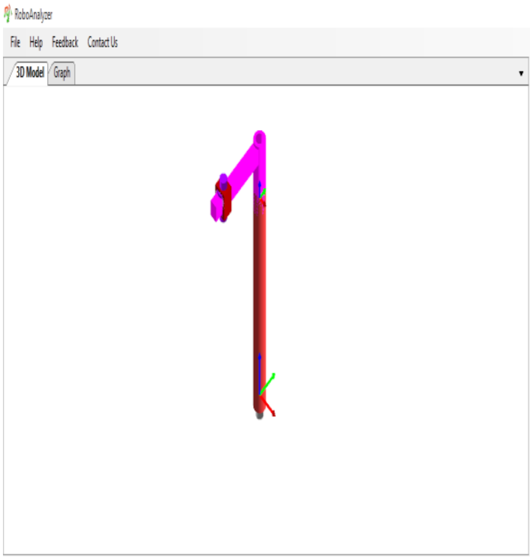

Fig.8: 3-D view of Stanford Manipulator

In inverse kinematics, tool configurations are required, which is described by the orientation matrix (first 3*3 elements) and positions (px, py, and pz) of the manipulator. Here, as we are dealing with six-axis manipulator to control its position & orientation, its inverse kinematics equations are required. Inverse kinematics results in multiple solutions [18].

No direct solution can be found for any of the joint variables from the above equations. To solve for a set of angles inverse matrix approach has been used where we can pre-multiply or post-multiply both the sides with the inverse of matrix one by one.

For any matrix inversion method

$$\left[T_1^0\right]^{-1} * T_1^0 = I \qquad (31)$$

Here I is the Identity matrix

$$\mathrm{T1} = (\mathrm{T}_1^0 * \mathrm{T}_2^1)^{-1}; \qquad (32)$$

$$\mathrm{T2} = (\mathrm{T}_1^0 * \mathrm{T}_2^1) * \mathrm{T1}; \qquad (33)$$

To get the values of angle theta post-multiply T1 with T from the (1)

$$\mathrm{T1} * \mathrm{T} = \mathrm{T}_6^2; \qquad (34)$$

Terms on both sides are equated and unknown joint angles and d3 can be calculated.

The equation of joint angle becomes

$$\mathrm{theta1} = \mathrm{atan2(px, -py)} \pm \mathrm{atan2(k1, k2);} \qquad (35)$$

Here k1 and k2 are

$$\mathrm{k1} = \sqrt{(\mathrm{px})^2 + (-\mathrm{py})^2 - \mathrm{d2}^2}; \qquad (36)$$

$$\mathrm{k2} = \mathrm{d2}; \qquad (37)$$

atan2 function has been used to uniquely specified the joint angle in the range of 2π.

$$\mathrm{theta2} = \mathrm{(c1px + s1py, h1 - pz)} \qquad (38)$$

$$\mathrm{d3} = \mathrm{pzc2 - h1c2 - pxc1s2 - pys1s2} \qquad (39)$$

$$\mathrm{theta4} = \mathrm{atan2((axs1 - ayc1),} \big(\mathrm{-(azs2 + axc1c2+ayc2s1;} \qquad (40)$$

$$\mathrm{theta5} = \mathrm{atan2(s5, c5)} \qquad (41)$$

$$\mathrm{theta6} = \mathrm{atan2(s6, c6)} \qquad (42)$$

By using inverse kinematics joint angles can be obtained that have multiple solutions. With the increase in the DOFs the complexity of the robotic arm structure increases and it becomes very difficult to find the solution using classical methods. Simulations have been performed using MATLAB software with Firefly algorithm (FFA) to reduce the error between the predicted and deduced values.

III. SIMULATION

The firefly is a swarm technique where fireflies get attracted by the light intensity of other fireflies [5], [27]-[30].

The absorption of intensity is as follows:

$$\mathrm{I} = \mathrm{I}_0 * e^{-\gamma r^2} \qquad (43)$$

The attractiveness can be written as:

$$\beta = \beta_0 * \mathrm{e}^{-\gamma r^2} \qquad (44)$$

I_0 is the actual intensity and β_0 is attractiveness at distance r=0. The values of Firefly parameters have been specified after performing iterations with random values of combinations in their particular limit. The values of parameters can be changed depending on their maximum and minimum value.

Retrieval Number: A9454109119/2019©BEIESP
DOI: 10.35940/ijeat.A9454.109119

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table- II: Firefly Parameters [32]

S.No.	Parameters	
	Variables	***Range***
1.	No. of Iterations(It)	100-2000
2.	Swarm Size	20-50
3.	γ	0 to 1
4.	β_0	1 to 2
5.	α	0.02 to 0.2
6.	α_damp	0.98
7.	δ	0.05*(VarMax-VarMin)

In the Table-II the firefly parameters γ, α, β_0, and δ are light absorption coefficient, mutation coefficient, attraction coefficient base value, and uniform mutation range respectively. α_damp is the mutation coefficient damping ratio [31], [32].

Fitness function to optimize the error is given by:

$$\text{Fitness} = w_1 * \text{Fit} + w_2 * \text{Error}; \qquad (45)$$

The fitness function shown in (45) has been defined based on the position of X, Y and Z. The fitness Fit is the distance between the target and calculated value and Error is the absolute sum of errors for the random set of joint angles. Here w_1 and w_2 are the weights having sum to be 1.

The value of w_1 depends to the extent those results based on the initial position to target position having X, Y and Z coordinates only, but w_2 gives values of the absolute error between the predicted and deduced values. The priority of position or error decides the value of w_1 and w_2.

IV. RESULTS

In this paper to obtain the target position of (px py pz) at (5, -12, 15) the values considered are h1=15cm, d2=12cm, d3=5cm.

$$\text{VarMin} = \begin{bmatrix} -180 & -90 & -180 & -25 & -180 \end{bmatrix}; \qquad (46)$$

$$\text{VarMax} = \begin{bmatrix} 180 & 90 & 180 & 25 & 180 \end{bmatrix}; \qquad (47)$$

The results for the fitness function as shown in (45) and calculated values of f_1, f_2, X, Y, Z are shown in Table-III at different iterations (It).

Table- III: Iteration and Fitness

S.No.	Fitness		
	It	***Fitness***	***Calculated X,Y,Z***
1	100	0.0451	5.0655, -11.9723, 15.0654
2	200	0.0040	4.9914, -12.0036, 15.0027
3	300	0.0006	5.0002, -11.9999, 14.9993
4	400	6.54E^{-05}	5.0001, -12.0000, 15.0000
5	500	1.36E^{-05}	5.0000, -12.0000, 15.0000

The joint angles theta1, theta2, theta4, theta5 and theta6 (T1, T2, T4, T5 and T6) has been calculated for the Fitness function of (45) at different iterations (It) are shown in Table-IV.

Table- V: Iteration and Joint angles

S. No.	Fitness					
	It	***T1***	***T2***	***T4***	***T5***	***T6***
1	100	-62.83	-7.84	77.28	5.11	-7.30
2	200	-11.78	7.85	7.39	17.28	-20.51
3	300	-80.89	32.98	31.18	-14.14	33.02
4	400	25.92	-17.28	63.00	-14.14	-57.54
5	500	-55.76	58.12	-3.67	-7.85	9.87

The Graph obtained at the end of the 500th iteration for the Fitness function of (45) is shown in fig.9

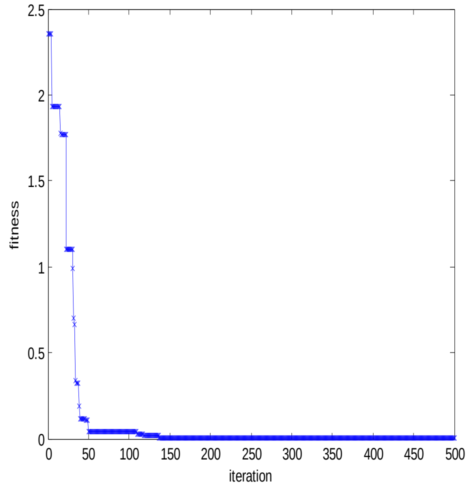

Fig.9: Fitness vs. No. of iteration

When the distance will be intimidating and w_2 is zero, then fitness function will be considered as Fit (distance) as shown in (45). The Fitness with Iteration and the calculated values of X, Y and Z are shown in Table-V and joint angles in Table-VI.

Table- V: Iteration and Fitness

S.No.	Fit (Distance)		
	It	***Fitness***	***Calculated X,Y,Z***
1	100	0.01665	5.0186, -11.9922, 14.9686
2	200	0.00477	5.0059, -11.9976, 14.9939
3	300	0.00054	5.0003, -11.9999, 14.9994
4	400	0.00078	5.0001, -12.0000, 14.9999
5	500	8.06E^{-06}	5.0000, -12.0000, 15.0000

Table- VI: Iteration and Joint angles

S. No.	Fitness					
	It	***T1***	***T2***	***T4***	***T5***	***T6***
1	100	0.002	-58.13	103.35	-20.03	-21.44
2	200	-55.76	58.12	-1.59	-7.85	8.07
3	300	-43.98	-7.85	108.14	-7.46	-48.58
4	400	113.89	-23.56	-116.79	-20.42	47.94
5	500	-100.53	-39.27	-4.08	17.67	129.2

Retrieval Number: A9454109119/2019©BEIESP
DOI: 10.35940/ijeat.A9454.109119

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

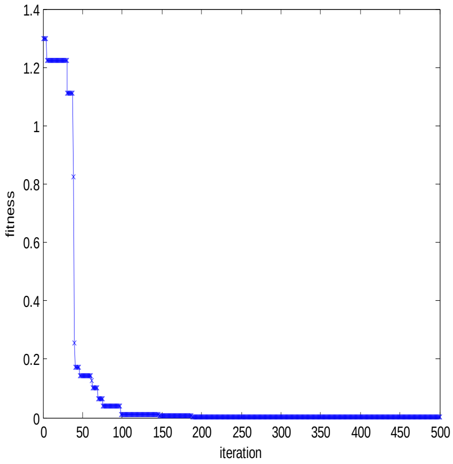

Fig.10: Fit (distance) vs. Iteration

Graph has been plotted for the 500th iteration as shown in fig.10. The values of joint angles found for the Fit (distance) have been shown in Table-VI.

The results for Error prevailing distance having Fitness (Error) function as in (45) have been shown in Table-VII and Table-VIII and Graph between the Fitness and Iteration is shown in fig.11.

Table- VII: Iteration and Fitness

S.No.	Error		
	It	*Fitness*	*Calculated X,Y,Z*
1	100	0.04986	5.0716, -11.9699, 15.0331
2	200	0.00386	4.9979, -12.0009, 15.0090
3	300	0.00040	5.0008, -11.9997, 15.0000
4	400	5.49E^{-05}	5.0000, -12.0000, 15.0000
5	500	1.12E^{-05}	5.0000, -12.0000, 15.0000

Table- VIII: Iteration and Joint angles

S. No.	Error					
	It	*T1*	*T2*	*T4*	*T5*	*T6*
1	100	0.006	-45.55	-2.89	-7.47	61.87
2	200	-0.00	-89.53	-38.4	17.67	113.95
3	300	7.073	64.40	-52.0	-7.85	-11.09
4	400	0.79	26.70	34.5	4.71	-64.90
5	500	-50.27	-70.69	103.1	-1.18	21.40

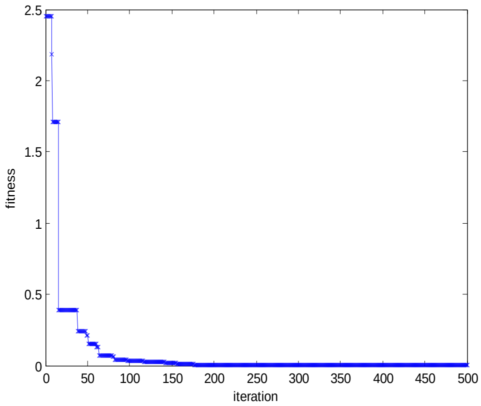

Fig.11: Fitness (Error) vs. Iteration

The comparison graph between the Fitness function (with fit and Error), Fit (distance), and Fitness (Error) has been plotted in fig.12. Top and Bottom plotted line (Red and green respectively) shows results for Fitness (Error) & Fitness (distance) and the centre line (Blue) give results for the Fitness function which is a combination of distance and error. The Fitness plotted line (Blue line) indicates that results can be improved by combining the distance and error function as shown in fig.12.

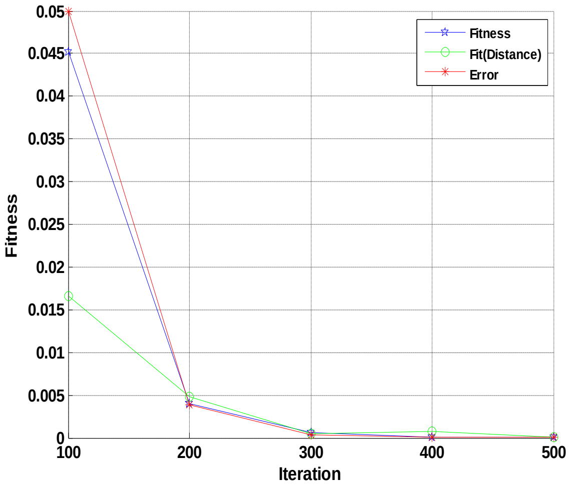

Fig.12: Graph Fitness vs. Iteration

Fitness of 1.36E^{-05} has been calculated with 500th iteration of (45). In the test at 1000th Iteration Fitness of 3.55E^{-10} with joint angles theta1, theta2, theta4, theta5 and theta6 (-74.61, 14.14, 35.035, -1.57, 27.94) and with 2000th Fitness of 1.3E^{-14} with joint angles (32.21, 14.14, -72.43, 10.99, 14.81) has been obtained as shown in fig.13 and fig.14.

Transformation Matrix at 1000th Iteration

$$T = \begin{bmatrix} -0.4547 & -0.829 & 0.326 & 5 \\ -0.878 & 0.354 & -0.324 & -12 \\ 0.153 & -0.433 & -0.888 & 15 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (48)$$

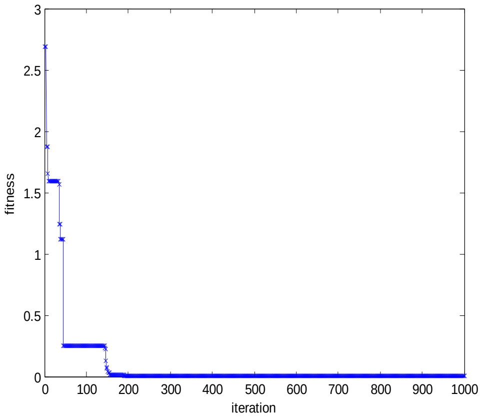

Fig.13: Fitness vs. Iteration (at 1000th)

Retrieval Number: A9454109119/2019©BEIESP
DOI: 10.35940/ijeat.A9454.109119

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Transformation Matrix at 2000th Iteration

$$T = \begin{bmatrix} 0.1128 & -0.9858 & -0.124 & 5 \\ -0.984 & -0.1282 & 0.1234 & -12 \\ -0.1376 & 0.1085 & -0.9845 & 15 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (49)$$

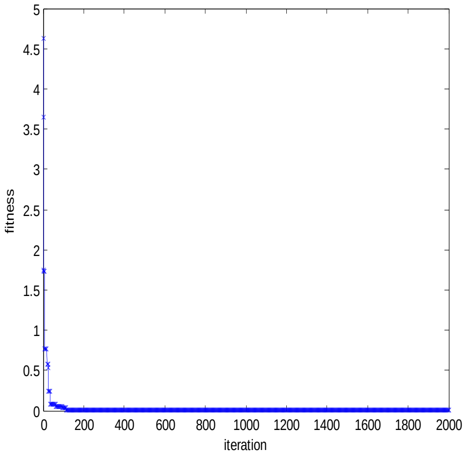

Fig.14: Fitness vs. Iteration (at 2000th)

Homogeneous matrix for the 1000th and 2000th iteration for joint angles has been written in (48) and (49). By combining the fit (distance) and Error accurate target position has been achieved.

V. CONCLUSION

This paper presents the kinematic modeling of 6-DOF robotic arm using matrix inversion methods to get the joint angles. The motion of the robot through various angles and coordinates can be controlled in various directions along with various joint angle combinations.

The direct and inverse kinematics concepts were also used to find the position of end-effectors for various joint angles, and joint angles for the end-effectors. Roboanalyzer has been used to show the 3D model of the Stanford manipulator.

Different simulations have been performed to fix the values of parameters of the firefly algorithm.

The results were verified for the forward kinematics and inverse kinematics using MATLAB software with Firefly algorithm. The simulation results for the different joint angles and offset with error difference has been calculated. The best solution has been obtained using different values of firefly parameters.

REFERENCES

1. S.B. Niku. Introduction to Robotics: Analysis, System and Applications, Pearson Education Inc., 2005.
2. R. Koker,"A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization", *Information Sciences* 222 (2013), pp.528–543. http://dx.doi.org/10.1016/j.ins.2012.07.051
3. A.N. Barakatl, K.A. Gouda, and K.A. Bozed,"Kinematics Analysis and Simulation of a Robotic Arm using MATLAB", *Proc. 2016 4th International Conference on Control Engineering & Information Technology (CEIT-2016), 16-18 December, 2016, Hammamet, Tunisia*, pp.642-646. DOI:10.1109/CEIT.2016.7929032
4. X.S. Yang. Nature-Inspired Metaheuristic Algorithms, 2nd edition, Luniver Press, 2010.
5. A.F. Nicolescu, F.M. Ilie, and T.G. Alexandru,"Forward and Inverse Kinematics Study of Industrial Robots Taking into Account Constructive And Functional Parameter's Modeling", *Proc. 2015 Manufacturing Systems*, Vol.10, No.4, pp.157−164, 2015.
6. B. Siciliano, and O. Khatib. Handbook of Robotics, Springer-Verlag Berlin Heidelberg, 2008.
7. K.S. Fu, R.C. Gonzalez, and C.S.G. Lee. Robotics : Control, Sensing, Vision and Intelligence, McGraw Hill International Editions, 1987.
8. A. Khatamian,"Solving Kinematics Problems of a 6-DOF Robot Manipulator", *Proc. 2015 Int'l Conf. Scientific Computing CSC'15*, pp.228-233.
9. S. Alavandar, and M.J. Nigam,"Neuro-Fuzzy based Approach for Inverse Kinematics Solution of Industrial Robot Manipulators", *Int. J. of Computers, Communications & Control*, Vol. III (2008), No. 3, pp. 224-234.
10. M. Dahari, and J.D. Tan,"Forward and Inverse Kinematics Model for Robotic Welding Process Using KR-16KS KUKA Robot", *Proc. 2011 Fourth International Conference on Modeling, Simulation and Applied Optimization*, Kuala Lumpur, Malaysia, 19-21 April, 2011. DOI: 10.1109/ICMSAO.2011.5775598
11. C. Urrea, J. Cortés, and J. Pascal,"Design, Construction and Control of a SCARA manipulator with 6 Degrees of Freedom", *Journal of Applied Research and Technology* 14 (2016) 396–404. https://doi.org/10.1016/j.jart.2016. 09.005
12. V.K. Banga,"Movement optimization of robotic arm using soft computing techniques", *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, Vol:10, No:9, 2016, pp.1624-1628.
13. zapdoc.tips/introduction-to-robotics-fag-348042107c224aa f039aae24813cfa310b9334349.html
14. J.J. Craig. Introduction to Robotics : Mechanics and Control, Pearson Education Inc., 2003.
15. A. Ghosal. Robotics: Fundamental Concepts and Analysis, ISBN-139780195673913, Oxford University Press, 2006.
16. R.M. Murray, Z. Li, and S.S. Sastry. A Mathematical Introduction to Robotic Manipulation, CRC Press, 1994.
17. S. Kucuk, and Z. Bingul (2006). Robot Kinematics: Forward and Inverse Kinematics, Industrial Robotics: Theory, Modeling and Control, Sam Cubero (Ed.), ISBN: 3-86611-285-8, InTech. DOI: 10.5772/5015
18. R.J. Schilling. Fundamental of Robotics: Analysis and Control, Prentice Hall, India Pvt. Ltd., 2002.
19. M.P. Groover, M. Weiss, R.N. Nagel, and N.G. Odrey. Industrial Robotics: Technology, Programming and applications,Tata McGraw-Hill Edition 2008.
20. S.R. Deb. Robotics Technology and Flexible Automation, Tata McGraw Hill, 2002.
21. M. Sailaja, S.P. kumar, and M.R. Roy,"Modeling, Simulation and Kinematics Calculations of Stanford Manipulator", *International Journal of Engineering and Management*, Volume-5, Issue-2, pp.136-141, April-2015.
22. M. More, R. Abande, A. Dadas, and S. Joshi,"Matlab Simulator of a 6 DOF Stanford Manipulator and its Validation Using Analytical Method and Roboanalyzer", *International Journal for Research in Applied Science & Engineering Technology*, Volume 5, No.VII, pp.25-31, July 2017.
23. P. Thavamani, K. Ramesh, and B. Sundari,"Simulation and Modeling of 6-DOF Robot Manipulator Using Matlab Software," *International Research Journal of Innovations in Engineering and Technology*, Volume 2, Issue 4, pp 6-10, June-2018.
24. J. Denavit, and R.S. Hartenberg,"A Kinematic Notation for Lower-Pair Mechanisms based on Matrices", *Journal of Applied Mechanics*, Vol.22, pp.215-221, June 1955.
25. R.U. Islam, J. Iqbal, S. Manzoor, A. Khalid, and S. Khan, "An Autonomous Image-guided Robotic System Simulating Industrial Applications", *Proc. 2012 7th International Conference on System of Systems Engineering*, Genoa, Italy, 16-19, July 2012, pp.314-319.
26. www.roboanalyzers.com
27. B. Bhushan, and S.S. Pillai,"Particle Swarm Optimization and Firefly Algorithm: Performance Analysis", *Proc. 2013 3rd IEEE International Advance Computing Conference*, Ghaziabad, India, pp.746-751. DOI:10.1109/IAdCC.2013.6514320
28. N. Ali, M.A. Othman, M.N. Husain, and M.H. Misran,"A Review of Firefly Algorithm", *ARPN Journal of Engineering and Applied Sciences*, Vol.9, No.10, October 2014.

29. Dereli S., Köker R., Öylek İ. ve Ay M.,“A Comprehensive Research on the use of Swarm Algorithms in the Inverse Kinematics Solution”, *Journal of Polytechnic*, 2019; 22(1): 75-79. DOI:10.2339/politeknik. 374830
30. J.C. Bansal et al. (eds.), *Soft Computing for Problem Solving*, Advances in Intelligent Systems and Computing 817, 2019, pp.521-530. https://doi.org/10.1007/978-981-13-1595-4_42
31. Xin-She Yang and Xingshi He, (2013). ‘Firefly Algorithm: Recent Advances and Applications’, *Int. J. Swarm Intelligence*, Vol. 1, No. 1, pp. 36–50. DOI:10.1504/IJSI. 2013.055801
32. Rokbani N., Casals A. and Alimi A.M.,“IK-FA, a New Heuristic Inverse Kinematics Solver Using Firefly Algorithm”, *Computational Intelligence Applications in Modeling and Control*, 575: 369-395, (2015). DOI:10.10 07/978-3-319-11017-2_15

AUTHORS PROFILE

Ashwani Kumar is research scholar and is pursuing his Ph.D degree from IKG Punjab Technical University, Kapurthala, Punjab, India. He is working as an Assistant Professor in Electronics and Communication Engineering Section at Yadavindra College of Engineering, Punjabi University Guru Kashi Campus, Talwandi Sabo, Punjab, India. His areas of Interest are Robotics, Artificial intelligence and Soft Computing. He is a life member of Indian Society of Technical Education (ISTE) and International Association of Engineers (IAENG).

Dr. Vijay Kumar has received his Ph.D. in Elecronics (Artificial Intelligence) from Thapar University, Patiala, Punjab, India in 2011. He is working as a Professor in Electronics and Communication Engineering and Principal at Amritsar College of Engineering & Technology, Amritsar, Punjab, India. He has more than 200 research papers to his credit in International Journal and conferences. He is a 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, IEI etc. His areas of interest and research include Robotics, Artificial Intelligence, Automated control systems and Genetic algorithms.

Dr. Darshan Kumar has received his Ph.D. in Mechanical Engineering from Punjab Technical University, Jalandhar in 2011. He is working as Associate Professor in Mechanical Engineering at Beant College of Engineering & Technology, Gurdaspur, Punjab, India. His areas of Interest and Research include Fuzzy Logic, Supply chain design for industries, Wire EDM. He is an Associate member of Institute of Engineers (AMIE), India, Member of Indian Society of Technical Education (ISTE) and Society of Automotive Engineers (SAE).