Al-Khwarizmi Engineering Journal,Vol. 13, No. 1, P.P. 13- 25 (2017)

Al-Khwarizmi Engineering Journal

Solving the Inverse Kinematic Equations of Elastic Robot Arm Utilizing Neural Network

Hussein M. Al-Khafaji* **Muhsin J. Jweeg****

**Department of Mechanical Engineering / University of Technology*
***College of Engineering/ Al- Nahrain University*
*Email: hussalkhafaji@gamil.com
**Email: muhsinnah_eng@nahrainuniv.edu.iq

(Received 15 March 2016; accepted 14 November 2016)
https://doi.org/10.22153/kej.2017.11.002

Abstract

The inverse kinematic equation for a robot is very important to the control robot's motion and position. The solving of this equation is complex for the rigid robot due to the dependency of this equation on the joint configuration and structure of robot link. In light robot arms, where the flexibility exists, the solving of this problem is more complicated than the rigid link robot because the deformation variables (elongation and bending) are present in the forward kinematic equation. The finding of an inverse kinematic equation needs to obtain the relation between the joint angles and both of the end-effector position and deformations variables. In this work, a neural network has been proposed to solve the problem of inverse kinematic equation. To feed the neural network, experimental data were taken from an elastic robot arm for training the network, these data presented by joint angles, deformation variables and end-effector positions. The results of network training showed a good fit between the output results of the neural network and the targets data. In addition, this method for finding the inverse of kinematic equation proved its effectiveness and validation when applying the results of neural network practically in the robot's operating software for controlling the real light robot's position.

Keywords: *Elastic robot, forward and inverse kinematic equation of elastic robot. Neural networks.*

1. Introduction

The forward kinematic equation is a functional relationship between the generalized parameters (joint displacements) and the end – effector positions. By substituting a set of values of generalized parameters into the forward kinematic equation, the considering ends – effector can be obtained. The problem of finding the end-effector position for a given set of generalized parameters is called the direct kinematic problem [1]. In robot's end-effector positioning and control, it is required to find the generalized parameters that lead the end – effector to the specified position and orientation. This is done by finding the inverse of the forward kinematic equation (inverse kinematic problem). The complexity of the inverse kinematic problem depends heavily on the number of joints and structure of the robot [2]. Flexibility of robot arms will add additional terms in the forward kinematic equation, these terms presented by the deformation variables. The magnitude of these variables could be not estimated easily, because their magnitude depends upon external forces, inertia forces, positions and orientation of robot arms [3].

The importance of the inverse kinematic problems in robot control, and the complications of solving it, make many researches deal with this subject, and many methods investigated to solve the inverse kinematic equation. The common and widely applied solution is the closed form solution, literatures [3-8] presented different methods of closed form. A closed form solution is not always a possible way to find the solution, so various studies have been proposed using other techniques. Juan et. Al. [4] suggested the using of a Fuzzy reasoning system to position the arm of an articulated robot without explicitly solving the inverse kinematic equation. Yong et. al. [5] used Genetic algorithm to solve the inverse kinematic. Kuroe et. al. [11, 12] stated the use of neural networks as a new method of solving the inverse kinematics of a rigid robot manipulator. Previous research [13-15] examined the use of artificial neural networks to find the inverse kinematic equation for articulated robots. While, Shiv et. Al. [6] applied neuro-fuzzy to obtain the inverse kinematic equation for robot with three degrees of freedom.

Most of researches proposed their solutions for the mathematical model of a rigid manipulator. With the presence of deflection parameters, the problem will be more complicated. Therefore, in this work, the proposed neural network with multiple layers was suggested as a function approximation to find the inverse kinematic equation for an elastic robot arm (ERA). The multiple layer neural network is useful for function approximation because [15]:

- It is universal approximations.
- It is efficient approximations.
- It could be implemented as a learning machine.

2. Kinematic Equations of an Elastic Robot Arm (with Error)

The flexibility causes elastic deformations of the structural members of the manipulator, resulting in large end-effector errors, especially in long reach manipulator systems. Hence, as a result, the frames defined at the manipulator joints are displaced from their expected locations. So, the use of kinematic equation of the rigid robot to position the manipulator end–effector will place the manipulator in a different position than the desired one.

In Figure 1, the frame O_{i-1} is the base frame , O_i^I is the ideal location and O_i^a is the actual location due to the deflection.

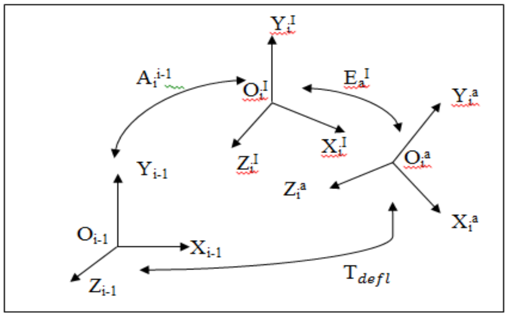

Fig. 1. Frame displacement due to errors

The transformation equation between the coordinates O_i^a and O_i^I consists of two parts, rotation and translation. According to the Euler angle principle, the rotation part is the result of three sets of rotations which are roll, pitch and yaw about the axes z, y, x, respectively. The sequence of rotation is:

$$E_{Ra}^{I} = \text{Rot}(z, d\phi)\,\text{Rot}(y, d\beta)\,\text{Rot}(x, d\psi) \quad \ldots(1)$$

That is a rotation of $d\psi$ about the x axis, followed by a rotation $d\beta$ about the y axis and finally a rotation of $d\phi$ about the z axis.

$$\text{Rot}(x, d\psi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos d\psi & -\sin d\psi & 0 \\ 0 & \sin d\psi & \cos d\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \ldots(2)$$

$$\text{Rot}(y, d\beta) = \begin{bmatrix} \cos d\beta & 0 & -\sin d\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin d\beta & 0 & \cos d\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ...(3)$$

$$\text{Rot}(z, d\phi) = \begin{bmatrix} \cos d\phi & -\sin d\phi & 0 & 0 \\ \sin d\phi & \cos d\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \ldots(4)$$

$$E_{Ra}^{I} = \begin{bmatrix} a_{11} & b_{12} & c_{13} & 0 \\ a_{21} & b_{22} & c_{23} & 0 \\ a_{31} & b_{32} & c_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \ldots(5)$$

Where:

$a_{11} = \cos d\phi \cos d\beta$

$a_{21} = \sin d\phi \cos d\beta$

$a_{31} = -\sin d\beta$

$b_{12} = \cos d\phi \sin d\beta \sin d\psi - \sin d\phi \cos d\psi$

$b_{22} = \sin d\phi \sin d\beta \sin d\psi + \cos d\phi \cos d\beta$

$b_{32} = \cos d\beta \sin d\psi$

$$c_{13} = \cos d\phi \sin d\beta \cos d\psi + \sin d\phi \sin d\psi$$
$$c_{23} = \sin d\phi \sin d\beta \cos d\psi - \cos d\phi \sin d\psi$$
$$c_{33} = \cos d\beta \cos d\psi$$

The equation …(5) represents the rotation part of the transformation matrix. The translation matrix is:

$$E_{Ta}^{I} = \begin{bmatrix} 1 & 0 & 0 & \delta x \\ 0 & 1 & 0 & \delta y \\ 0 & 0 & 1 & \delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ...(6)$$

The transformation between coordinates O_i^a and O_i^I becomes:

$$E_a^I = E_{Ta}^I \, E_{Ra}^I \quad …(7)$$

$$E_a^I = \begin{bmatrix} a_{11} & b_{12} & c_{13} & \delta x \\ a_{21} & b_{22} & c_{23} & \delta y \\ a_{31} & b_{32} & c_{33} & \delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad …(8)$$

Here, the δx, δy, δz, $d\psi$, $d\beta$ and $d\phi$ are called generalized error parameters. Usually, these generalized parameters are introduced on the basis of different approximations, such as assumed modes, finite elements, or Ritz-Kantorovich expansions, with different implications on the model complexity and accuracy [9].

The references [17-19] have supposed that the generalized deflection parameters are small, so a first order approximation can be applied to their trigonometric functions and product, and higher order equals zero.

$\sin d\psi = d\psi$, $\sin d\beta = d\beta$
$\sin d\phi = d\phi$
$\cos d\psi = \cos d\beta = \cos d\phi = 1$

Based on these assumptions, the matrix E_a^I in …(8) will be equal to:

$$E_{defl} = \begin{bmatrix} 1 & -d\phi & d\beta & \delta x \\ d\phi & 1 & -d\psi & \delta y \\ -d\beta & d\psi & 1 & \delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad … (9)$$

In this work, the values of deflection's parameters are taken in consideration.

Figure 1 shows the coordinate for serial of flexible links, the kinematic equation covering the positions of this links with presence of flexibility is:

$$T_{defl} = A_1^o(q_1) E_{a\,1}^{I\,1} \; A_2^1(q_2) E_{a\,2}^{I2} \cdots A_n^{n-1}(q_n) E_{a\,n}^{I\,n} \quad …(10)$$

A_n^{n-1} is the homogenous transformation matrix in Denavit- Hartenber (DH) equation (11):

$$A_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad … (11)$$

Where:

θ_i , α_i , a_i and d_i represent the joint parameter defined as in Figure 2.

By using equation (10), the transformation relationship between the two frames O_i^a and O_{i-1} can be found.

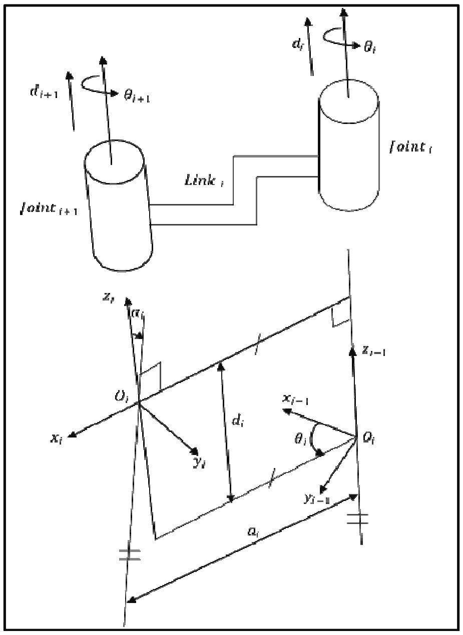

Fig. 2. The Denavit-Hartenberg notatio.

T_{defl} may be expressed in a simple compact form as a nonlinear function of q and ϵ vectors:

$$T_{defl} = f(q, \epsilon) \quad …(12)$$

Where:

T_{defl} is the set of Cartesian coordinates describing the position and orientation of the manipulator's end- effector with respect to the inertial frame.

$$q = [q_1 \quad q_2 \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad q_n]^T \quad ...(13)$$

$$\epsilon = [\epsilon_1 \quad \epsilon_2 \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \epsilon_g]^T \quad …(14)$$

$$\epsilon = [\delta x_m \quad \delta y_m \quad \delta z_m \quad d\psi_m \quad d\beta_m \quad d\phi_m]^T \qquad \dots(15)$$

Where:

q is the vector in joint space.

ϵ is a dimensional link deflection vector space, spanned to be the link deflections of the manipulator arm.

n is the number of joints.

g generally equals to 6 x m.

m is the number of flexible links.

The elements of the vector ϵ cannot easily specify their values, since these elastic coordinates are dependent on the rigid coordinate q and on the applied forces F.

For more generally:

$$\epsilon = E(\mathrm{F}, \mathrm{q}) \qquad \dots(16)$$

By substituting equation …(16) into equation (12), T_{defl} will be:

$$\mathrm{T}_{defl} = f(\mathrm{q}, \mathrm{E}(\mathrm{F}, \mathrm{q})) \qquad \dots(17)$$

As mentioned in the previous section, the two degrees of freedom of the arm robot manipulator used in this study are confined to move within the vertical plane, and so is the deformation of each link. Hence, there is no rotation about the axes x and y, also there is no transformation in the z direction (the parameters **δ**z, dψ and dβ are equal zero). Figure 3 shows a single flexible link coordinate system, the O_{i-1} represents the base local coordinate system, and the frame $\mathrm{O}_i^{\mathrm{I}}$ is the local coordinate system assigned to link i in its undeformed position, while the $\mathrm{O}_i^{\mathrm{a}}$ is the actual local coordinate system of link i , when it is under deformation.

The homogenous transformation matrix for this single link was derived as follow:

- According to the motion of the links within the vertical plane, the homogenous transformation matrix $\mathrm{E}_{\mathrm{ia}}^{\mathrm{I}}$ between $\mathrm{O}_i^{\mathrm{a}}$ and $\mathrm{O}_i^{\mathrm{I}}$ becomes:

$$\mathrm{E}_{ia}^{i\mathrm{I}} = \begin{bmatrix} \cos d\phi & -\sin d\phi & 0 & \delta x \\ \sin d\phi & \cos d\phi & 0 & \delta y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots(18)$$

- The homogenous transformation matrix between the $\mathrm{O}_{\mathrm{i}}^{\mathrm{I}}$ and $\mathrm{O}_{\mathrm{i-1}}$ is presented by the matrix $\mathrm{A}_{\mathrm{i}}^{\mathrm{i-1}}$ in equation (9).

- The homogeneous transformation matrix for this a single link is found from the relations (12) and (18).

$$\mathrm{T}_{ia}^{i-1\mathrm{a}} = \mathrm{A}_i^{i-1\mathrm{a}}\, \mathrm{E}_{\mathrm{ia}}^{i\mathrm{I}} \qquad \dots(19)$$

$$\mathrm{T}_{ia}^{i-1} = \begin{bmatrix} \mathrm{T}_{11} & \mathrm{T}_{12} & \mathrm{T}_{13} & \mathrm{T}_{14} \\ \mathrm{T}_{21} & \mathrm{T}_{22} & \mathrm{T}_{23} & \mathrm{T}_{24} \\ \mathrm{T}_{31} & \mathrm{T}_{32} & \mathrm{T}_{33} & \mathrm{T}_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \dots(20)$$

Where:

$c_i = cos\,\theta_i$

$s_i = \sin\theta_i$

$c\alpha_i = \cos\alpha_i$

$s\alpha_i = \sin\alpha_i$

$cd_i = \cos d\phi_i$

$sd_i = \sin d\phi_i$

$\mathrm{T}_{11} = c_i cd_i - c\alpha_i s_i sd_i$

$\mathrm{T}_{21} = \mathrm{s_i cd_i} + \mathrm{sd_i c\alpha_i c_i}$

$\mathrm{T}_{31} = s\alpha_i sd_i$

$\mathrm{T}_{12} = -c_i sd_i - c\alpha_i s_i cd_i$

$\mathrm{T}_{22} = -s_i sd_i + c_i cd_i c\alpha_i$

$\mathrm{T}_{32} = s\alpha_i cd_i$

$\mathrm{T}_{13} = s\alpha_i s_i$

$\mathrm{T}_{23} = -s\alpha_i c_i$

$\mathrm{T}_{33} = c\alpha_i$

$\mathrm{T}_{14} = c_i[\delta x_i + a_i] - c\alpha_i s_i \delta y_i$

$\mathrm{T}_{24} = s_i[\delta x_i + a_i] + \delta y_i c\alpha_i c_i$

$\mathrm{T}_{34} = s\alpha_i \delta y_i + \mathrm{d}_i$

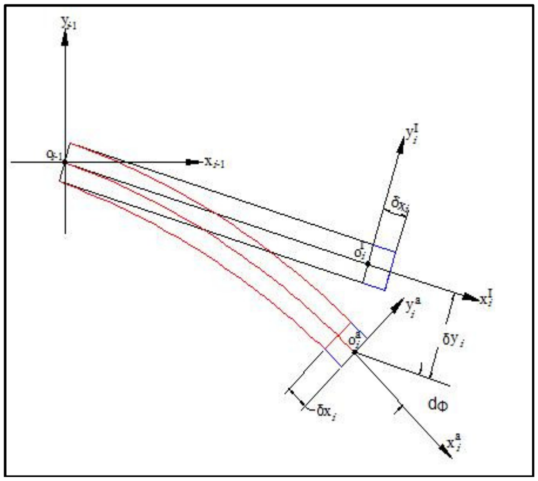

Fig. 3. The coordinates system of a single flexible link under deformation.

As shown in Figure 4, the total kinematic equation of flexible robot of two degrees of freedom is:

$$\mathrm{T}_{defl} = \mathrm{T}_{1a}^{0a}\, \mathrm{T}_{2a}^{1a} \qquad \dots(21)$$

The transformation matrix of robot in general form is:

$$T_{defl} = \begin{bmatrix} n_x & o_x & t_x & p_x \\ n_y & o_y & t_y & p_y \\ n_z & o_z & t_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \ldots(22)$$

Where:

$$\begin{aligned} n_x = {} & c_1c_2[cd_1cd_2 - sd_1sd_2c\alpha_2] \\ & - c_1s_2[cd_1c\alpha_2sd_2 + sd_1cd_2] \\ & + s_1s_2[c\alpha_1c\alpha_2sd_1sd_2 \\ & - c\alpha_1cd_1cd_2] \\ & - c_2s_1[cd_2sd_1c\alpha_1 \\ & + c\alpha_1cd_1c\alpha_2sd_2] \\ & + s_1s\alpha_1s\alpha_2sd_2 \end{aligned} \qquad \ldots(23)$$

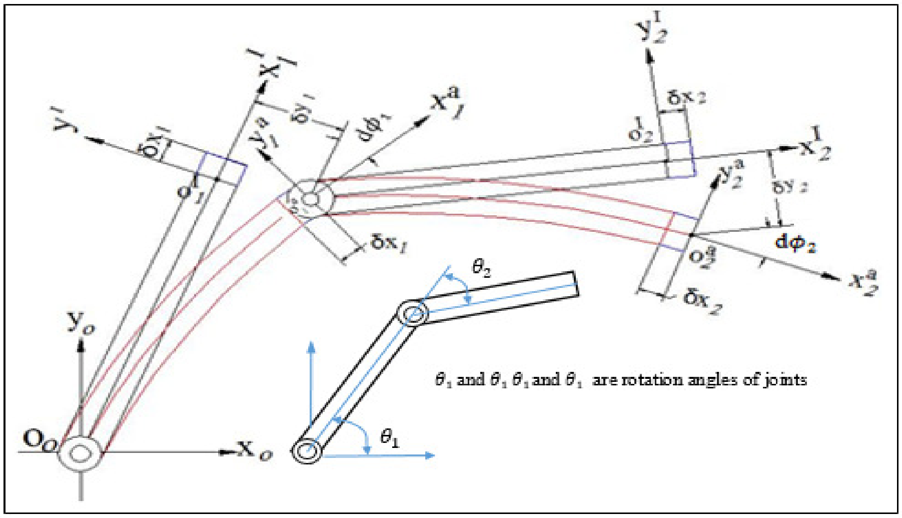

Fig. 4. the coordinates systems of two elastic links and the angles of joints.

$$\begin{aligned} n_y = {} & c_1c_2[cd_2sd_1c\alpha_1 + c\alpha_1cd_1c\alpha_2sd_2] \\ & - s_1s_2[sd_1cd_2 + cd_1c\alpha_2sd_2] \\ & + c_2s_1[cd_1cd_2 - sd_1sd_2c\alpha_2] \\ & + c_1s_2[cd_2cd_1c\alpha_1 \\ & - sd_1c\alpha_1sd_2c\alpha_2] \\ & - c_1[s\alpha_1sd_2s\alpha_2] \end{aligned} \qquad \ldots(24)$$

$$\begin{aligned} n_z = {} & c_2[s\alpha_1sd_1cd_2 + s\alpha_1cd_1c\alpha_2sd_2] \\ & + s_2[s\alpha_1cd_1cd_2 \\ & - s\alpha_1sd_1sd_2c\alpha_2] + c\alpha_1sd_2s\alpha_2 \end{aligned} \qquad \ldots(25)$$

$$\begin{aligned} o_x = {} & s_1s_2[c\alpha_1sd_1c\alpha_2cd_2 + c\alpha_1cd_1sd_2] \\ & - c_1c_2[cd_1sd_2 + sd_1cd_2c\alpha_2] \\ & + c_1s_2[sd_1sd_2 - cd_1c\alpha_2cd_2] \\ & + s_1c_2[c\alpha_1cd_1cd_2 \\ & - c\alpha_1sd_1sd_2c\alpha_2] \\ & + s_1s\alpha_1s\alpha_2cd_2 \end{aligned} \qquad \ldots(26)$$

$$\begin{aligned} o_y = {} & c_1c_2[cd_1c\alpha_1cd_2c\alpha_2 - sd_1c\alpha_1sd_2] \\ & - s_1c_2[sd_1cd_2c\alpha_2 + cd_1sd_2] \\ & - c_1s_2[cd_1c\alpha_1sd_2 \\ & + sd_1c\alpha_1c\alpha_2cd_2] \\ & + s_1s_2[sd_1sd_2 - cd_1c\alpha_2cd_2] \\ & - c_1[s\alpha_1s\alpha_2cd_2] \end{aligned} \qquad \ldots(27)$$

$$\begin{aligned} o_z = {} & c_2[s\alpha_1cd_1cd_2c\alpha_2 - s\alpha_1sd_1sd_2] \\ & + s_2[s\alpha_1cd_1sd_2 \\ & - s\alpha_1sd_1c\alpha_2cd_2] + c\alpha_1s\alpha_2cd_2 \end{aligned} \qquad \ldots(28)$$

$$\begin{aligned} t_x \text{(25)} = {} & c_1s_2cd_1s\alpha_2 - s_1s_2c\alpha_1sd_1s\alpha_2 \\ & + c_1c_2sd_1s\alpha_2 + s_1c_2c\alpha_1cd_1s\alpha_2 \\ & + s_1s\alpha_1c\alpha_2 \end{aligned} \qquad \ldots(29)$$

$$\begin{aligned} t_y = {} & s_1s_2cd_1s\alpha_2 + c_1s_2sd_1c\alpha_1s\alpha_2 \\ & + s_1c_2sd_1s\alpha_2 - c_2cd_1c\alpha_1s\alpha_2 \\ & - c_1s\alpha_1c\alpha_2 \end{aligned} \qquad \ldots(30)$$

$$t_z = s_2s\alpha_1sd_1s\alpha_2 - c_2s\alpha_1cd_1s\alpha_2 + c\alpha_1c\alpha_2 \qquad \ldots(31)$$

$$\begin{aligned} p_x = c_1c_2[cd_1(\delta x_2 + a_2) - sd_1\delta y_2 c\alpha_2] \\ - c_1s_2[cd_1c\alpha_2\delta y_2 \\ + sd_1(\delta x_2 + a_2)] \\ - s_1c_2[c\alpha_1 sd_1(\delta x_2 + a_2) \\ + c\alpha_1 cd_1\delta y_2 c\alpha_2] \\ + s_1s_2[c\alpha_1 sd_1 c\alpha_2\delta y_2 \\ - c\alpha_1 cd_1(\delta x_2 + a_2)] \\ + s\alpha_1 s_1 s\alpha_2\delta y_2 + s\alpha_1 s_1 d_2 \\ + c_1(\delta x_1 + a_1) - c\alpha_1 s_1\delta y_1 \end{aligned} \quad \dots (32)$$

$$\begin{aligned} p_y = s_1c_2[cd_1(\delta x_2 + a_2) - sd_1\delta y_2 c\alpha_2] \\ - s_1s_2[sd_1(\delta x_2 + a_2) \\ + cd_1c\alpha_2\delta y_2] \\ + c_1c_2[sd_1c\alpha_1(\delta x_2 + a_2) \\ + cd_1c\alpha_1\delta y_2 c\alpha_2] \\ + c_1s_2[cd_1c\alpha_1(\delta x_2 + a_2) \\ - sd_1c\alpha_1c\alpha_2\delta y_2] + c_1[c\alpha_1\delta y_1 \\ - s\alpha_1 s\alpha_2\delta y_2 - s\alpha_1 d_2] \\ + s_1[\delta x_1 + a_1] \end{aligned} \quad \dots (33)$$

$$\begin{aligned} p_z = c_2[s\alpha_1 sd_1(\delta x_2 + a_2) + s\alpha_1 cd_1\delta y_2 c\alpha_2] \\ + s_2[s\alpha_1 cd_1(\delta x_2 + a_2) \\ - s\alpha_1 sd_1 c\alpha_2\delta y_2] + c\alpha_1 s\alpha_2\delta y_2 \\ + c\alpha_1 d_2 + s\alpha_1\delta y_1 + d_1 \end{aligned} \quad \dots (34)$$

$c_i = \cos\theta_i$ $cd_i = \cos d\phi_i$ $c\alpha_i = \cos\alpha_i$
$s_i = \sin\theta_i$ $sd_i = \sin d\phi_i$ $s\alpha_i = \sin\alpha_i$

The degree of freedom of the robot arm manipulator which is used in this work, is two degrees of freedom, and confined to move within the vertical plane with the kinematic parameters listed in the Table (1).

Table 1,
The parameters of two links.

Joints	θ_i	d_i	α_i	a_i
1	θ_1	0	0	a_1
2	θ_2	0	0	a_2

By substituting the values of parameters of the manipulator from the Table (1) in to the equations (23) to (34) and with triangular operations, these equations will be as follow:

$$n_x = [c_{12}][cd_{12}] - [s_{12}][sd_{12}] \quad \dots(35)$$

$$\begin{aligned} n_y = c_1c_2[sd_{12}] + s_1s_2[sd_{1-2}] + c_2s_1[cd_{1-2}] \\ + c_1s_2[cd_{12}] \end{aligned} \quad \dots (36)$$

$$n_z = 0 \quad \dots(37)$$

$$\begin{aligned} o_x = s_1s_2[sd_{1-2}] - c_1c_2[sd_{12}] - c_1s_2[cd_{1-2}] - \\ s_1c_2[cd_{12}] \end{aligned} \quad \dots(38)$$

$$o_y = c_{12}[cd_{12}] - s_{12}[sd_{1-2}]] \quad \dots(39)$$

$$o_z = 0 \quad \dots(40)$$

$$t_x = 0 \quad \dots(41)$$

$$t_y = 0 \quad \dots(42)$$

$$t_z = 1 \quad \dots(43)$$

$$\begin{aligned} p_x = c_1c_2[cd_1(\delta x_2 + a_2) - sd_1\delta y_2] \\ - c_1s_2[cd_1\delta y_2 \\ + sd_1(\delta x_2 + a_2)] \\ - s_1c_2[sd_1(\delta x_2 + a_2) \\ + cd_1\delta y_2] + s_1s_2[sd_1\delta y_2 \\ - cd_1(\delta x_2 + a_2)] + c_1(\delta x_1 \\ + a_1) - s_1\delta y_1 \end{aligned} \quad \dots (44)$$

$$\begin{aligned} p_y = s_1c_2[cd_1(\delta x_2 + a_2) - sd_1\delta y_2] \\ - s_1s_2[sd_1(\delta x_2 + a_2) \\ + cd_1\delta y_2] + c_1c_2[sd_1(\delta x_2 \\ + a_2) + cd_1\delta y_2] \\ + c_1s_2[cd_1(\delta x_2 + a_2) \\ - sd_1\delta y_2] + c_1\delta y_1 + s_1[\delta x_1 \\ + a_1] \end{aligned} \quad \dots (45)$$

$$p_z = 0 \quad \dots(46)$$

Where:
$c_{12} = \cos(\theta_1 + \theta_2)$
$s_{12} = \sin(\theta_1 + \theta_2)$
$cd_{12} = \cos(d\phi_1 + d\phi_2)$
$sd_{12} = \sin(d\phi_1 + d\phi_2)$
$cd_{1-2} = \cos(d\phi_1 - d\phi_2)$
$sd_{1-2} = \sin(d\phi_1 - d\phi_2)$

Equations (44) to (46) represent the end-effector position with presence of flexibility of robot, which will be called, the flexible kinematic equation. The inverse of these equations will be found by using the neural network.

3. Experimental Work

a. The rig of Experiment

In this work, the test rig was represented by two degrees' manipulator (two elastic links) confined to move in a vertical plane. The two links were designed to have properties gathering the lightness approximate to the flexible beam and stiffness approximate to the rigid beam to reduce the vibration effects; the two links are hallow rectangular beams made from the aluminum type (6061-T6) with length 57 cm and 62 cm for first and second arm, respectively. In addition, the manipulator has a measurement system (bonded strain gauges on each arm) and data acquisition system to measure the deflection variable, To find the rotation of the tip of the beam $d\phi$ experimentally, the measuring of the deflections at two points along the beam was done at the points where the strain gauge No.3 and at the tip, and by applying these values of the deflections into the following equation [20] :

$$\tan d\phi = \frac{\Delta\delta y}{\Delta x} \qquad \ldots (47)$$

Where:

δy represent the deflection.

Δδy represents the difference between the two deflections.

Δx represents the distance between the two points where the deflections measured.

The relations between the deflections and the voltages of strain gauges were estimated experimentally by curve fitting between deflection and voltage as shown in Figure 5.

For Link-1:

The relation between the tip deflection and the voltage of the strain gauge No.1 is:

$$\delta y_{tip} = 0.0026\,V + 8 \times 10^{-5} \qquad \ldots(48)$$

The relation between the tip deflection and the voltage of the strain gauge No.2 is:

$$\delta y_{tip} = 0.001\,V + 3 \times 10^{-5} \qquad \ldots(49)$$

The relation between the deflection at the point 31cm from the center of the shaft of the first joint and the voltage of the strain gauge No.3 is:

$$\delta y_{at\ 31\ cm} = 0.0008\,V - 2 \times 10^{-5} \qquad \ldots(50)$$

The Link-2:

The relation between the tip deflection and the voltage of the strain gauge No.1 is:

$$\delta y_{tip} = 0.0016\,V + 6 \times 10^{-5} \qquad \ldots(51)$$

The relation between the tip deflection and the voltage of the strain gauge No.2 is:

$$\delta y_{tip} = 0.0022\,V + 3 \times 10^{-5} \qquad \ldots(52)$$

The relation between the deflection at the point 36 cm from the center of the shaft of the second joint and the voltage of the strain gauge No.3 is:

$$\delta y_{at\ 36\ cm} = 0.0022\,V - 3 \times 10^{-5} \qquad \ldots(53)$$

δx was experimentally found by extracting the strain from two sensors at the upper and lower surface at the same position of the strain gauge No.1 and No. 2. The difference value represents the extension of the beam in the longitudinal direction, as represented by the relation [20]:

$$\delta x = \epsilon . L \qquad \ldots(54)$$

Figure 6 shows the two degrees robot arm manipulator and the strain gages.

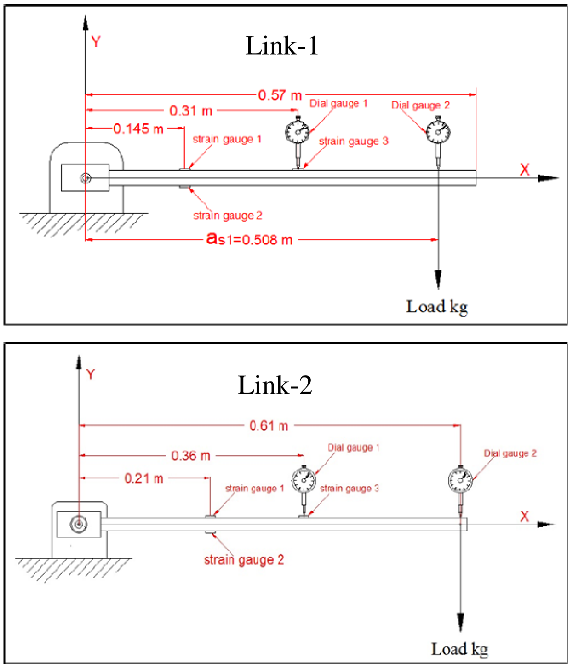

Fig. 5. the distribution of strain gauges and applied load for the two links.

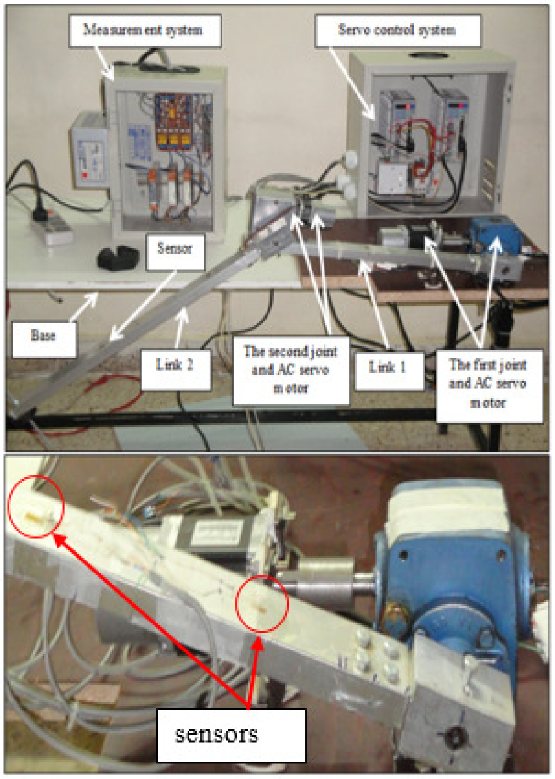
Measurem ent system
Servo control system
Sensor
Base
Link 2
The second joint and AC servo motor
Link 1
The first joint and AC servo motor
sensors

Fig. 6. the two degrees robot arm manipulator and the strain gages.

b. Collection of Data

The experimental data for feeding the neural networks for training include joint's angle, deformation variables and corresponding end-effector positions. To collect data, a control mode was designed to operate the robot in specific sequence of motions, which include moving the first links in a range of angles (0 to 30^o step 5^o) and the second link (-35^o to 60^o step 5^o). When the control mode is activated, the first link is still at angle 0^o, and the second link is still a while at the start angle, then moves 5^o to the next step. After completion the range of angles, the second arm returns back to the start angle, and the first joint moves 5^o to next step and so on. At each step (position), the deformation variable and end-effector position were measured and saved automatically. This procedure was repeated for four values of load (0,1,2, and 3) kg hanging at the end of the link two. The number of readings for each parameter was 560 reading, which will be used for training neural network. Figure 7 shows the samples of positions during this test.

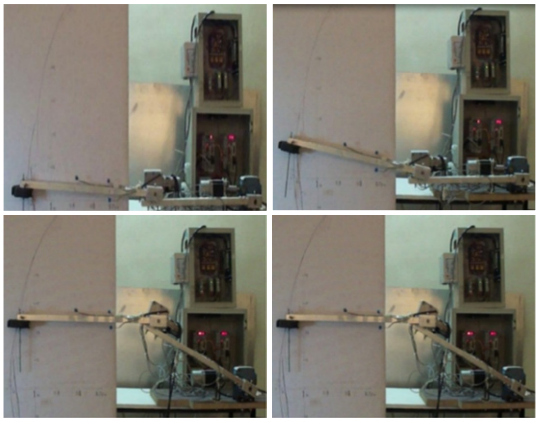

Fig. 7. samples of positions during experimental work.

Some samples of the results of the deformation parameters, which were read in above test, are shown in the figures (figure 8 to figure 13), these figures represent the change of values for δ**y**, δ**x** and $d\boldsymbol{\phi}$ at the tip with angles of the two links. These results were used as inputs to equations (44) and (45) to calculate the end effector in x and y coordinate that would be with the measured generalized parameters as input values to the neural network.

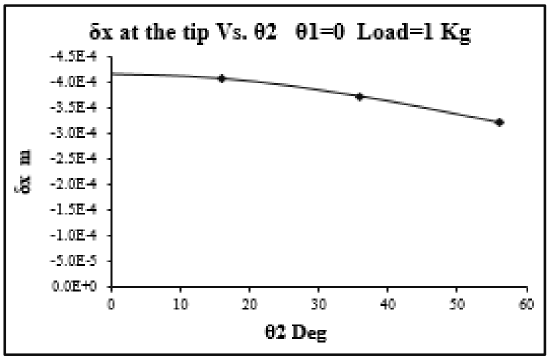

Fig.8 δx Vs. θ2 at θ1=4 Deg. link-1.

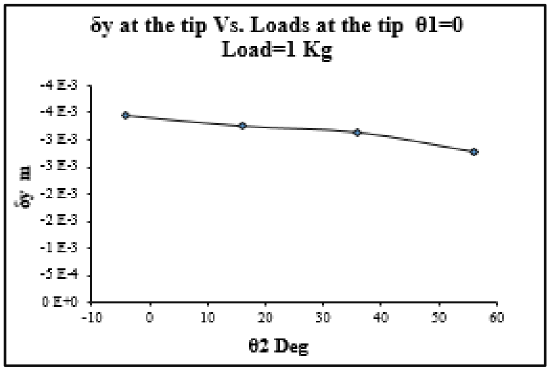

Fig. 9. δy Vs. θ2 at θ1=0 Deg. link-1.

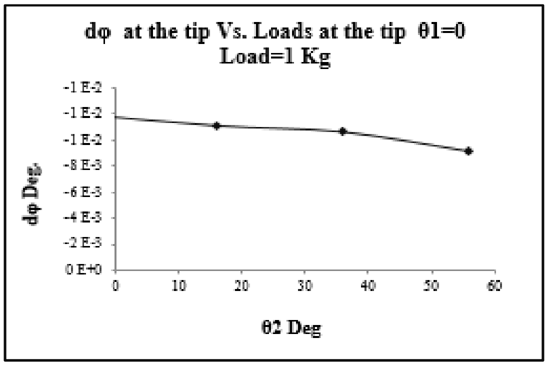

Fig. 10. dϕ Vs. θ2 at θ1=0 Deg. link-1.

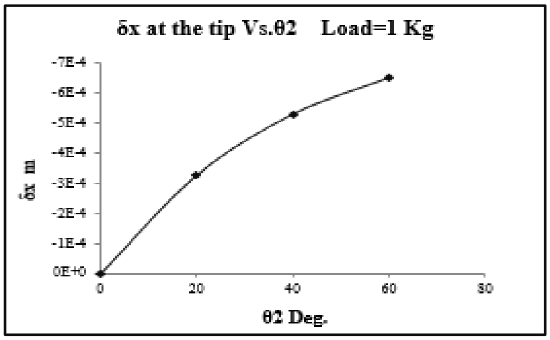

Fig. 11. δx Vs. θ2 Deg. Link-2.

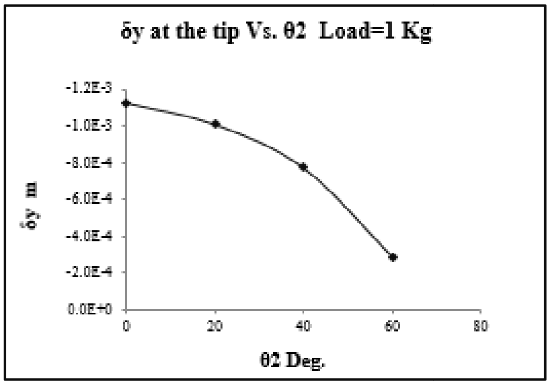

Fig. 12. δy Vs. θ2 Deg. Link-2.

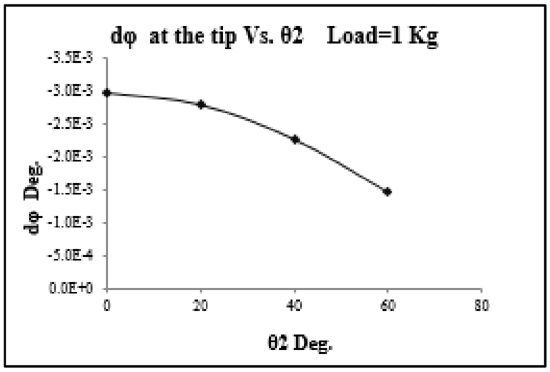

Fig. 13 dϕ Vs. θ2 Deg. Link-2.

4. Neural Network for Inverse Kinematic Equation of Elastic Robot

As mentioned in previous section, there is a difficulty to solve the problem for finding the inverse of the kinematic equation with presence of the deformation parameters. So, to overcome this difficulty, the neural network was used to find the inverse of the kinematic equations (44) to (46). These equations relate the vector of operating (task) space coordinate to the joints and deformations vector, these equations could be presented in compact form:

$$P = f(q, \epsilon) \qquad \ldots(55)$$

Where:

$$P = [p_x, p_y] \;, \qquad p_z = 0 \qquad \ldots(56)$$

$$q = [\theta_1, \theta_2] \qquad \ldots(57)$$

$$\epsilon = [\delta x_1, \delta y_1, d\phi_1, \delta x_2, \delta y_2, d\phi_2] \qquad \ldots(58)$$

The inverse kinematic problem of rigid robot is defined as giving end effector positions (P) to get the joint variable vector (q), in the elastic robot. beside the vector (P), there will be the deformations vector ϵ that should be given to get (q). So, the inverse equation will be:

$$q = f^{-1}(P, \epsilon) \qquad \ldots(59)$$

The proposed neural network was used to model the equation (59), which is called the kinematic inverse neural network. In this work, many neural networks were trained. The structure of each network was with (one input layer and two hidden layers-output layer). Each network had a different number of neurons in the hidden layers. The numbers was chosen randomly. The selection of neurons number was specified by trial and error. The input layer received the eight variables (δy1, δx1, dϕ1, δy2, δx2, dϕ2, px, and py). The output layer gave the angles (θ1, θ2) as a network result, as shown in Figure 14.

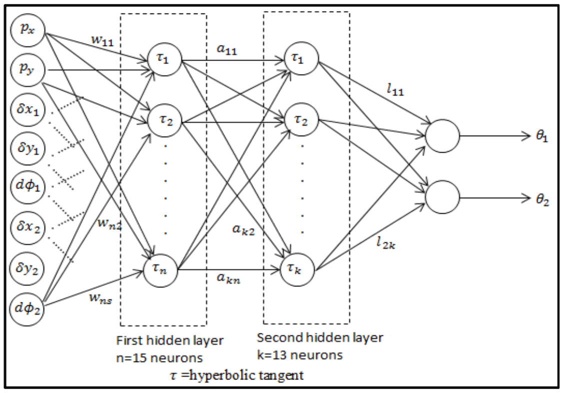

Fig. 14. the structure of kinematic inverse neural network.

The transfer function (τ) used in the hidden layers for each network was the hyperbolic tangent function (due to the bipolar value for the output of the function) as shown in equation [21]:

$$\tau(\mathrm{x}) = \tanh(x) = \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}} \qquad \ldots(60)$$

Where β is the slope parameter.

For the output layer was pure line. The back propagation training algorithm with one variable learning rate was used to train these neural networks.

The input variables (δy1, δx1, dϕ1, δy2, δx2, dϕ2) were imported from the results of the test in the part of experimental work and by using these variables with corresponding angles (θ1 and θ2) in equations (44) and (46) to calculate the last two variables (px and py). Hence, these variables were used to train the networks.

The networks that used were simulated by using the scientific and engineering package Matlab.

To find the optimum neural network, the tested networks output was compared with respect to the angles that were used in equations (44) and (45) to get px and py. The best fit obtained with the network has the following structure, as shown in Figure 14:

1- One input layer (receives the eight variables).
2- Two hidden layers:
   a- n=15 neurons in the first hidden layer.
   b- k=13 neurons in second hidden layer.
3- One output layer (gives two outputs).

The Network training parameters that used in this network are listed as follow:

Training parameters	**Value**
Epochs	5000
Goal	1×10-8
Minimum gradient	1×10-10
Learning rate	1×10-5
Learning increment	0.9

5. Results and Discussion

Figures 15 and 16 show the comparison between the optimum neural network (ONN) outputs and the target data for (θ1 and θ2), respectively. These figures reveal a good fitting between the network results and the target angles. The roots square errors of result network were (0.999742) and (0.999804) for the first and the second angles, respectively.

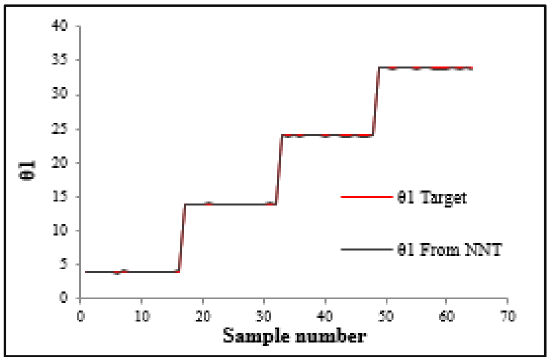

Fig. 15. The comparison between θ1 target and θ1 ONN.

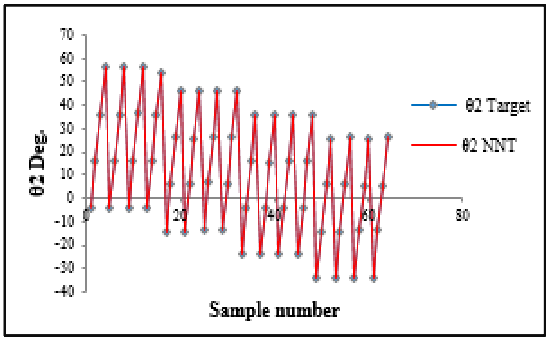

Fig. 16 the comparison between θ2 target and θ2 ONN.

Figures 17 and 18 depict the comparison between the px and py, that resulted from the equations (44) and (45) with the neural network. Each figure has two curves; the first one is from applying the target angles, and the second one is resulted from applying the ONN.

The Figures from 15 to 18 show a good fitting between the neural network results and the target angles, so that the results of this neural network were used in the control system of the elastic robot to control the motion and position.

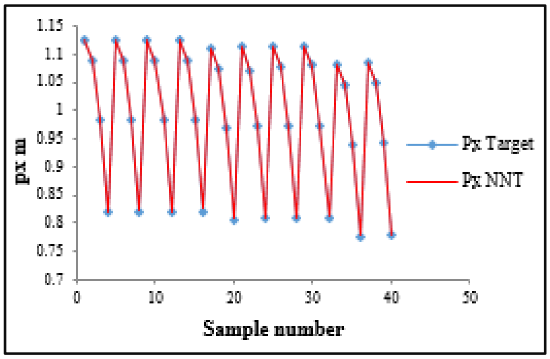

Fig. 17 the comparison between px target and px ONN.

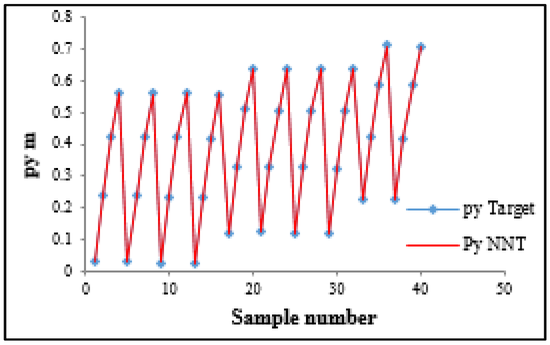

Fig. 18. the comparison between py target and py ONN.

As mentioned above, there are many networks with different structures that were experimented to find the best one. At the beginning, the networks were with one hidden layer trained, but the results were far from the target in spite of increasing the number of neurons. Therefore, by using networks with two hidden layers, the result began to approach from the target. Figures 19 to 22 illustrate the results for some of the networks that were experimented. The way that these networks were named depended upon how many neurons were used in each layer, so the name (network 12-40) means the first layer has 12 neurons, and the second layer has 40 neurons.

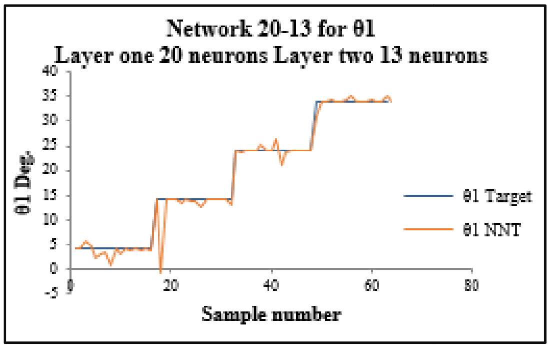

Fig.19. θ1 Target Vs.θ1 neural network two layers.

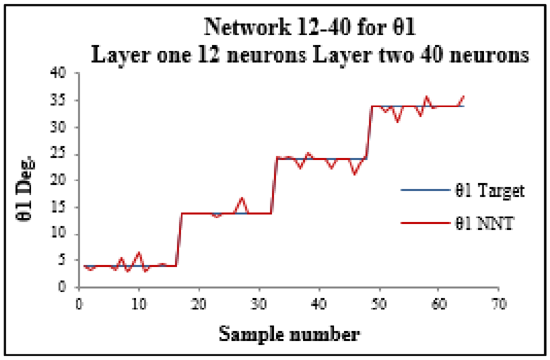

Fig. 20. θ1 target Vs. θ1 neural network two layers.

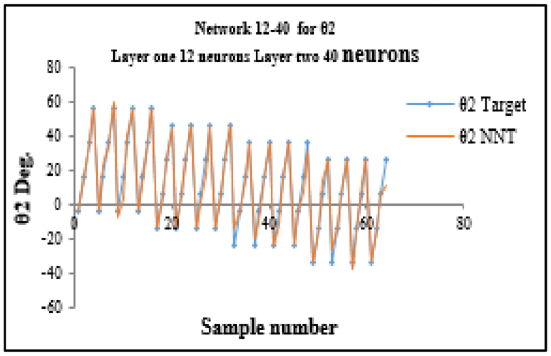

Fig. 21. θ2 target Vs. θ2 neural network two layers.

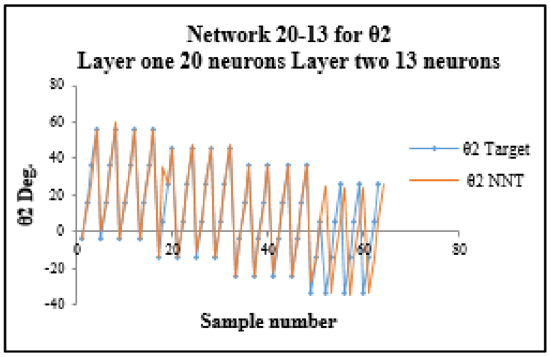

Fig. 22. θ2 target Vs. θ2 neural network two layers.

6. Conclusion

The first advantage of using neural network is to overcome the difficulties of kinematic inverse equation of the flexible robot, and the second advantage is that it takes the values of training from the experimental work by operating the robot itself, and measures the variables with presence of more than one effect during the operation (like joint's stiffens, joint's gear box backlash, noise, etc..). These effects will be taken in consideration to establish the network automatically (during learning to find the function approximation), Therefore, the result of network when used in the control system of robot will overcome the problem of these effects, but may be when using the other methods, especially the algebraic method, the inverse kinematic equation will be solved mathematically and will not take in consideration these effects. The proposed neural network gave a good fit with the target. The roots square errors of results network were (0.999742) and (0.999804) for the first and the second angles, respectively. And it is found from the training and testing of many networks that, not conditionally, the increase of number of neurons in the hidden layers will give the best result. Also, this neural network was used in the operating software of ERA robot and showed a good performance to position the end effector of the elastic robot.

7. References

[1] Slotine, H. Asada and J. J. E., Robot Analysis and Control, New York: John Wiley and Sons, 1985.

[2] Manseur, Rachid, Robot Modeling and KIneamtics, India : Firwall Media, 2007.

[3] Bingul, Serdar Kucuk and Zafer, "The inverse kinematics solutions of industrial

robot manipulators," in Mechatronics, 2004. ICM '04. Proceedings of the IEEE International Conference on, 3-5 June 2004.

[4] Albert Castellet and Federico Thomas, "Towards an Efficient Interval Method for Solving Inverse Kinematic Problems," in pp. 3615-3620, IEEE International Conference on Robotics and Automation, New Mexico, April, 1997.

[5] Michael Wenz and Heinz Wörn, "Solving the inverse Kinematics Problem Symbolically by Means of Knowledge-Based and Linear Algebra-Based Methods," in IEEE Conference on Emerging Technologies and Factory Automation, pp. 1346-1353, 2007.

[6] Jian Xie, Wenyi Qiang, Bin Liang and Cheng Li, "Inverse Kinematics Problem for 6-DOF Space Manipulator Based on the Theory of Screws," in International Conference on Robotics and Biomimetics,IEEE, Sanya, China, December 15-18 2007.

[7] E. Sariyildiz and H. Temeltas, "Solution of Inverse Kinematic Problem for Serial Robot Using Dual Quaterninons and Plücker Coordinates," in Advanced Intelligent Mechatronics, IEEE/ ASME International Conference on , Singapor, 2009 .

[8] Emre Sariyidiz, Eray Cakiray and Hakan Temeltas, "A Comparative Study of Three Invese Kinematic Mehtods of Serial Industrial Robot Manipulators in the Screw Theory Framework," International Journal of Advanced Robotic Systems, 2011.

[9] Juan Martinez, John Bowles and Patrick Mills, "A Fuzzy Logic Positioning System for an Articulated Robot Arm," in IEEE, 1996.

[10] Yong-Gui Zhang, Li-Ming Xie, Hao Shen and Lan Jin, "Simulation on Whole Inverse Kinematics of a 5R Robot Based on Hybrid Genetic Algorithm," in The Eighth International Conference on Machine Learning and Cybernetics, 12-15 July, 2009.

[11] Yasuaki Kuroe, Yasuhiro Nakai and Takehiro Mori, "A New Neural Network Approach to the Inverse Kinematics Problem in Robotic," in Conference Motion and Control Proceedings, Asia- Pasific work shop, IEEE, 1993.

[12] Yasuaki Kuroe, Yasuhiro Nakai and Takehiro Mori, "A Neural Network Learning of Nonlinger Mappings with Considering Their Smoothness and Its Application to Inverse Kinematics," in Industrial Electronic, Control and Instrumentation, 20th International Conference on (Volume 2). IEEE, 1994.

[13] Jolly Shah, S. S. Rattan, B.C. Nakra , "Kinematic Analysis of a Planer Robot Using Artificial Neural Network," International Journal of Robotics and Automation (IJRA), vol. 1, no. 3, pp. 145-151, September 2012.

[14] Panchanand Jha, BB Biswal, "A Neural Network Approach for Inverse Kinematic of a SCARA Manipulator," International Journal of Robotics and Automation (IJRA) , vol. 3, no. 1, pp. 52-61, March 2014.

[15] Adrian- Vasile Duka , "Neural Network Based Inverse Kinematic Solution for Trajectory Tracking of a Robotic arm," Procedia Technology , Science Direct, Elsevier, 2014.

[16] Shiv Manjaree , Vijyant Agarwal and B.C. Nakra, "Kinematic Analysis Using Neuro-Fuzzy Intelligent Technique for Robotic Manipulator," International Journal of Engineering Research and Technology , vol. 6 , no. 4, pp. 557-562, 2013.

[17] U. Y. X. a. T. Y. Hiroshi, "Modeling and control strategy of A 3-D Flexible Space Robot," in Workshop on Intelligent Robots and Systems IROS, 1991.

[18] C. S. D. P. D. J. H. a. J. F. Mavroidis, "A systematic Error Analysis of Robotic Manipulators: Application to High Performance Medical Robot," in International Conference in Robotic and Automation, Albuquerque, 1997.

[19] J. Zhao-Hui, "Compensability of End-Effector Motion Errors and Evaluation of Manipulability for Flexible Space Robot Arms," IEEE, pp. 486-491, 1992.

[20] Stephen P. Timoshenko and James M. Gere, "Mechanics of Materials", Book, London-Britain: Van Nostrand Reinhold Company Ltd., 1978.

[21] Simon Haykin, "Neural Networks and Learnings Machines", Book, New Jersey-United States of America: Pearson Education, NC., 2009.

حل معكوس المعادلة الكينماتيكية لرجل آلي ذي أذرع مرنة باستخدام الشبكات العصبية

حسين محمد حسين الخفاجي*   محسن جبر جويج**

**قسم الهندسة الميكانيكية / الجامعة التكنولوجية*
***كلية الهندسة / جامعة النهرين*
*البريد الالكتروني: hussalkhafaji@gamil.com
**البريد الالكتروني:muhsinnah_eng@nahrainuniv.edu.iq

الخلاصة

ان معكوس المعادلة الكينماتيكية للروبوت يكون مهما في السيطرة على موقع الروبوت وحركته وان حل هذه المعادلة يكون معقداً بالنسبة للروبوت الصلب بسبب اعتماد هذه المعادلة على تركيب المفاصل وهيكل الروبوت. في الروبوتات الخفيفة الاذرع حيث تكون المرونة موجودة، فأن الحل سيكون اعقد لوجود متغيرات التشوه في المعادلة المباشرة (الامامية). ان إيجاد معكوس المعادلة الكينماتيكية يحتاج الى إيجاد العلاقة ما بين زوايا المفاصل وكل من موقع الطرف النهائي للروبوت ومتغيرات التشوه. في هذا العمل، تم اقترح استخدام الشبكة العصبية لحل مشكلة معكوس المعادلة الكينماتيكية ولتغذية الشبكة العصبية وامدادها بالبيانات الخاصة بتدريبها، تم استعمال بيانات عملية اخذت من روبوت مرن، تلك البيانات تمثلت بزوايا المفاصل، متغيرات التشوه ومواقع الطرف النهائي. ان نتائج تدريب الشبكة العصبية بينت تطابقا جيدا ما بين مخرجات الشبكة والقيم المطلوبة. فضلا عن ذلك ان هذه الطريقة في إيجاد معكوس المعادلة الكينماتيكية اثبتت فاعليتها عند تطبيق الشبكة العصبية في منظومة السيطرة الخاصة بتحديد موقع الطرف النهائي للروبوت الخفيف المرن.