Solving Forward Kinematics Problem of Stewart Robot Using Soft Computing

Yaser Salehinia, Farid Najafi, S. Hossein Sadati, Mohsen Shiee
Dept. Mechanical Engineering
K. N. Toosi University of Technology
Tehran, Iran

Saeed Salehinia
Dept. Mechanical Engineering
Shahid Rajaee Teacher Training University
Tehran, Iran

Abstract—in this paper, we consider the problem of efficient computation of the forward kinematics of a 6 DOF robot manipulator built to use in rehabilitation purpose. Forward kinematics problem (FKP) of parallel robots is very difficult to solve in comparison to the serial manipulators. This problem is almost impossible to solve analytically. Numerical methods are one of the common solutions for this problem. But, accuracy, speed and convergence of these methods are fully dependent on the initial guess vector that is fed to the numerical algorithm. In this paper, soft computing approach like Artificial Neural Networks, fuzzy-neural network and nonlinear Auto Regressive eXogenous (NARX) identification method is used to solve the FKP of the Stewart robot. This problem is solved in the typical workspace of this robot. The results show the advantages of Nonlinear ARX identification method in providing very small modeling errors and provide excellent position and orientation angle estimation.

Keywords— Stewart robot, Artificial Neural Networks, ANFIS, Nonlinear ARX, Identification.

I. INTRODUCTION

In the last 30 years, parallel manipulators have been among the most considerable research topics in Robotics. These robots are now applied in real-life applications such as force sensing robots, fine positioning devices, and medical applications [1, 2].

The parallel manipulator (PM) was first proposed by Stewart in 1965 as an aircraft simulator platform. Few others have ventured into the study of PM after Stewart. The kinematics attracted the interests of Lee and Shah [3], Behi [4], Zang [5], Shahinpoor [6] and Fangella [7]. Stewart platforms, have received increasing attention because of their high stiffness, high speed, high accuracy and high carrying capability [1]. Although mechanical simplicity of Stewart platforms provides potential for many engineering applications, their forward (direct) kinematics is very complex which limit their real-time applications. This is due to the requirements of solving a set of highly nonlinear equations or high degree polynomials for a general Stewart platform. Broadly speaking, there are two approaches to the solution of the forward kinematics of the platform, namely analytical and numerical. In the first approach attempts are made to develop closed form solutions in special cases where some of the connection points at the platform or at the base are coalesced in groups of two or three. In this way the general 6-6 configuration, i.e. 6 separate joints at base and platform is reduced to less than six at the base and/or the platform [8, 9].

Another approach in the analytical category is to place restriction on the geometry of the platform or the base, or assume a certain relationship (e.g. linear) between the base and platform coordinates at attachment points [10, 11]. In the numerical approaches, Newton-Raphson method or a variation of it is used to solve iteratively the set of nonlinear forward kinematics equations. This approach finds only one solution assuming a good starting point, but no feasible solutions is guaranteed [11, 12].

Artificial Neural Networks (ANNs) are computational models comprising numerous nonlinear processing elements networks. These computational models have now become exciting alternatives to conventional approaches in solving a variety of engineering and scientific problems.

Some researchers have tried using neural networks for solving the FKP of parallel robots [13-14].

For nonlinear system identification, researches have been mainly focused on NARX models, notably since the emergence of artificial neural networks applied to nonlinear system identification. NARX models are frequently used in black-box nonlinear system identification. NARX models can be used for various purposes (control, simulation, behavior analysis, monitoring, etc.).

In this paper, neural network approach, fuzzy-neural network and NARX identification is used to solve forward kinematic problem of Stewart robot.

II. THE ROBOT MANIPULATOR

The Stewart robot consists of six pneumatic actuators whose length can be varied. The actuators are fixed at bottom and are connected to a platform at the other. By changing length of the different actuators one can access six degrees of freedom. A universal joint and a spherical joint are used to assemble the actuators onto the fixed base and mobile platform, respectively. Each actuator is equipped with a linear potentiometer and two pressure transducers.

There are two principal boxes in the system, called electric box and pneumatic box. Air is compressed by the air compressor and delivered to pneumatic box. To make the
quality of air acceptable, air service equipment is utilized in the pneumatic box. After air service, the flow enters two manifolds onto which 12 valves are mounted. Finally, the flow cross 12 flow control valves, on the top of pneumatic box to reach actuator chambers. Valves exciting signals and sensor outputs are all connected to electric box. The electric box consists of power supplies, data conversation cards terminals and valves driving circuits.

III. THE INVERSE KINEMATICS

There are two frames describing the motion of the moving plate: frames $A(x, y, z)$ and $B(u, v, w)$ as shown in Fig. 2, located at the center of the base plate and at the center of the moving plate, respectively. The location of the moving platform can be described by a position vector, $p$, and a rotation matrix, $R_{AB}$. Let the rotation matrix be defined by the roll, pitch, and yaw angles, namely, a rotation of $\phi_x$ about the fixed x-axis, followed by a rotation of $\phi_y$ about the fixed y-axis, and a rotation of $\phi_z$ about the fixed z-axis. Thus the rotation can be written as [15]:

$$
^{A}R_{B} = \begin{bmatrix}
\cos \phi_x \cos \phi_y & -\sin \phi_x \cos \phi_z & \sin \phi_y \\
\sin \phi_x \cos \phi_y & \cos \phi_x \cos \phi_z & -\sin \phi_y \\
-\sin \phi_y & \cos \phi_y & \cos \phi_x \\
\end{bmatrix}
$$

(1)

As shown in Fig. 2, let $a_i = [a_{x_i}, a_{y_i}, a_{z_i}]^T$ and $b_i = [b_{x_i}, b_{y_i}, b_{z_i}]^T$ be the position vectors of points $A_i$ and $B_i$ in the coordinate frames $A$ and $B$, respectively. We can write a vector-loop equation for the $i$th limb of the manipulator as follows:

$$
\overline{AB}_i = p + ^{A}R_{B}^i b_i - a_i
$$

(2)

The length of the $i$th limb is obtained by taking the dot product of the vector $\overline{AB}_i$ with itself:

$$
d_i = \left[p + ^{A}R_{B}^i b_i - a_i\right]^T \left[p + ^{A}R_{B}^i b_i - a_i\right] \quad \text{for } i = 1, 2, ..., 6
$$

(3)

where $d_i$ denote the length of the $i$th limb. Expanding Eq. 3 yields:

$$
d_i^2 = p^T p + ^{A}R_{B}^i b_i^T b_i + a_i^T a_i + 2 p^T ^{A}R_{B}^i b_i
$$

$$
-2p^T a_i - 2^{A}R_{B}^i b_i^T a_i
$$

(4)

for the IKP, the position vector $p$ and the rotation matrix $^{A}R_{B}$ of frame $B$ with respect to $A$ are given and the limb lengths, $d_i$, $i = 1, 2, ..., 6$, are to be found. The solution is very straightforward. Taking the square root of Eq. 4 we obtain

$$
d_i = \left(p^T p + ^{A}R_{B}^i b_i^T b_i + a_i^T a_i + 2 p^T ^{A}R_{B}^i b_i
$$

$$
-2p^T a_i - 2^{A}R_{B}^i b_i^T a_i\right)^{1/2}
$$

(5)

for $i = 1, 2, ..., 6$. Hence, corresponding to each given location of the moving platform, there are generally two possible solutions for each limb. However, the negative limb length is physically not feasible [16].

I. SOFT COMPUTING SOLUTION FOR FKP

A. Workspace analysis

It is well known that parallel manipulators have a rather limited and complex workspace. Six parameters consisting of three coordinates of position of center of mass of mobile platform in the base frame ($X, Y, Z$) and three RPY orientation angles of mobile platform with respect to the base frame (three
angles of mobile platform orientation in space consist of $\phi_x$, $\phi_y$, and $\phi_z$ angles, see Fig. 2 vary in the Stewart workspace. Complete analysis of Stewart workspace is presented in [17] by A. Bonev. The geometric parameters of the robot were given in Table 1.

**TABLE 1. Geometric Parameter of Stewart Robot**

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>p</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>220mm</td>
<td>160mm</td>
<td>650mm</td>
<td>550mm</td>
</tr>
</tbody>
</table>

**B. Neural network solution for FKP**

In order to model the FKP with ANNs, first, typical workspace of the robot is determined. Then, IKP is solved in some poses of the workspace and finally ANNs is trained with the data of IKP solution in typical robot workspace. The data consist of nonlinear moves in unequal time for all position and orientation angle.

Now MLP network will be trained with the data generated by solution of IKP. There is no basis at all to determine which the most efficient configuration ANN should have for any particular problem. Therefore the ANN has to be tested for different configurations and the performances compared for selecting the best choice. The simplest way to gauge the performance of an ANN is to look at the error at the output. A simple error function is calculated during the training cycles to monitor the performance.

Note that the activation functions used in the hidden layers and output layer are tan-sigmoid and pure linear, respectively. The number of patterns used for training and test are %70 and %30 of the data, respectively. After getting both 6-member vectors L and D, all components of the two vectors are normalized to values in the range of [0, 1]. This normalization step is introduced to cut down overflow problems in the calculations.

Random initialization was used for the weights. Different configurations of the MLP network were tested by varying the number of neurons in the hidden layer between 8 and 38 and configurations of the MLP network were tested by varying the number of neurons in the hidden layer from 8 to 38.

**TABLE II. Multi feed forward one hidden layer.**

<table>
<thead>
<tr>
<th>Number of Hidden Layer Neurons</th>
<th>Training Time(s)</th>
<th>MSE</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>3421</td>
<td>2.52e-5</td>
<td>0.0143</td>
</tr>
<tr>
<td>32</td>
<td>3880</td>
<td>1.03e-6</td>
<td>0.00135</td>
</tr>
<tr>
<td>35</td>
<td>4220</td>
<td>5.23e-6</td>
<td>0.00394</td>
</tr>
</tbody>
</table>

About 30 multilayer feed forward networks with two hidden layers were trained by varying the number of neurons from 12 to 30 in the first hidden layer and from 8 to 25 in the second hidden layer. All these networks were trained over 1000 training epochs.

**TABLE III. Multi-layer feed forward with 2 hidden layers.**

<table>
<thead>
<tr>
<th>Number of Hidden Layer</th>
<th>Training Time(s)</th>
<th>MSE</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>13-13</td>
<td>3749</td>
<td>8.13e-6</td>
<td>0.0396</td>
</tr>
<tr>
<td>18-13</td>
<td>4972</td>
<td>3.30e-6</td>
<td>0.010</td>
</tr>
<tr>
<td>30-10</td>
<td>6808</td>
<td>2.39e-5</td>
<td>0.0188</td>
</tr>
</tbody>
</table>

Using the whole stated criteria, three networks with best performance were selected from each configuration. Table II and table III summarize the performance of these networks. It should be also noted that the mean square of error is approximately equal to the square of the maximum error, so a mean square error of 1e-5 will correspond to about 0.1 degree of accuracy for the forward kinematics solution.

**C. Fuzzy-Neural Network**

The Adaptive Network-based Fuzzy Inference System (ANFIS) that was proposed by Roger Jang [18] is one of the most commonly used fuzzy inference systems and it is widely employed for modeling and forecasting complicated and nonlinear systems. ANFIS is a multilayer feed forward network with a supervised learning scheme which is equal to Takagi-Sugeno-Kang (TSK) fuzzy inference system (Takagi and Sugeno, 1985; Sugeno and Kang, 1988).A fuzzy rule in a Sugeno fuzzy inference system is presented as:

$$If \ x \ is \ A \ and \ y \ is \ B \ then \ z = f(x, y)$$  \hspace{1cm} (6)

where A and B are the membership functions for inputs x and y; f(x, y) is a crisp function in output space. The output function (f) can be derived in different ways to make relations between the input and output spaces and usually is a linear function of inputs. Suppose that the rule base contains the following two Sugeno-type fuzzy if-then rules:

Rule 1:\ if \ x \ is \ A_i \ and \ y \ is \ B_i \ then \ f_i = p_i x + q_i y + r_i

Rule 2:\ if \ x \ is \ A_j \ and \ y \ is \ B_j \ then \ f_j = p_j x + q_j y + r_j$$  \hspace{1cm} (7)

where /Pi, qi, ri/ are the consequent parameters that are determined during the training process. The main objective of ANFIS is to optimize the parameters of a given fuzzy inference system by a learning process.

As in Fig. 3, the ANFIS architecture consists of the following five layers: fuzzification, inference, normalization, consequent and output. Let $o_j$ denotes the output of the jth node in jth layer. We use 6 ANFIS models for 6 outputs that we have in workspace. The best result is for triangle membership function (MF) between 8 type of MFs, and with 2 number of MFs for all inputs, we use product for AND method in the fuzzy rule base. Six separate ANFIS’s with similar structures are utilized to approximate the forward kinematics of the stewart robot.
TABLE IV. ANFIS performance.

<table>
<thead>
<tr>
<th>Variable</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>1.0525e-5</td>
</tr>
<tr>
<td>y</td>
<td>1.1812e-5</td>
</tr>
<tr>
<td>z</td>
<td>8.924e-6</td>
</tr>
<tr>
<td>φ</td>
<td>1.1192e-6</td>
</tr>
<tr>
<td></td>
<td>2.026e-6</td>
</tr>
<tr>
<td></td>
<td>3.1266e-6</td>
</tr>
</tbody>
</table>

D. Nonlinear ARX Model

Typically, the NARX models will be used as black-box structures. NARX models have structure as Eq. 8:

\[
y(t) = f \left( y(t-1), y(t-n_y), u(t-n_u), u(t-n_u-1) \right)
\]  

(8)

Where the function \( f \) depends on a finite number of previous inputs \( u \) and outputs \( y \). \( n_y \) is the number of past output terms and \( n_u \) is the number of past input terms used to predict the current output. \( n_k \) is the delay from the input to the output, specified as the number of samples. The nonlinear function of the NARX model is a flexible nonlinearity estimator with parameters that need not have physical significance. The NARX model uses a parallel combination of nonlinear and linear blocks [19]. Fig. 4 shows the NARX model structure.

![Fig. 3. Nonlinear ARX Model Structure [20]](image)

The NARX model uses regresses as variables for nonlinear and linear functions. Regresses are functions of measured input-output data [20]. The predicted output \( \hat{y}(t) \) of a nonlinear model at time \( t \) is given by the general Eq. 9:

\[
\hat{y}(t) = F(x(t))
\]  

(9)

Where \( x(t) \) represents the regresses, \( F \) is a nonlinear regresses command, which is estimate \( d \) by nonlinearity estimators/classes [19]. As shown in Fig. 4, the command \( F \) can include both linear and nonlinear functions of \( x(t) \). Eq. 10 gives the description of \( F \).

\[
F(x) = \sum_{i=1}^{d} \alpha_i \kappa(\beta_i(x - \gamma_i))
\]  

(10)

Where \( \kappa \) is the unit nonlinear command, \( d \) is the number of nonlinearity units, and \( \alpha_i, \beta_i, \) and \( \gamma_i \) are the parameters of the nonlinearity estimators/classes [20].

The NARX model computes the output in two stages [20]:

1. Computes regresses from the current and past input values and past output data. (In this work, all regresses are inputs to both the linear and the nonlinear function blocks of the nonlinearity estimator.)
2. The nonlinearity estimator block maps the regresses to the model output using a combination of nonlinear and linear functions.

The nonlinearity estimators used in this paper are Wavelet network and sigmoid network. Using %70 of the data for identification, several configuration is considerate for the best order and number of unit. Table V shows the output of the obtained models based on best fit:

TABLE V. CONFIGURATION OF THE ESTIMATOR

<table>
<thead>
<tr>
<th>Output</th>
<th>Class of estimator</th>
<th>Order of estimator</th>
<th>Number of unit</th>
<th>Best Fit (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Wavelet network</td>
<td>( n_x = 2, n_y = 2, n_k = 1 )</td>
<td>3</td>
<td>99.85</td>
</tr>
<tr>
<td></td>
<td>Sigmoid network</td>
<td>( n_x = 0, n_y = 2, n_k = 3 )</td>
<td>3</td>
<td>99.74</td>
</tr>
<tr>
<td>y</td>
<td>Wavelet network</td>
<td>( n_x = 2, n_y = 2, n_k = 1 )</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Sigmoid network</td>
<td>( n_x = 0, n_y = 2, n_k = 3 )</td>
<td>3</td>
<td>99.97</td>
</tr>
<tr>
<td>z</td>
<td>Wavelet network</td>
<td>( n_x = 2, n_y = 2, n_k = 1 )</td>
<td>3</td>
<td>97.47</td>
</tr>
<tr>
<td></td>
<td>Sigmoid network</td>
<td>( n_x = 0, n_y = 2, n_k = 3 )</td>
<td>3</td>
<td>99.71</td>
</tr>
<tr>
<td>φ</td>
<td>Wavelet network</td>
<td>( n_x = 1, n_y = 2, n_k = 3 )</td>
<td>3</td>
<td>99.39</td>
</tr>
<tr>
<td></td>
<td>Sigmoid network</td>
<td>( n_x = 0, n_y = 2, n_k = 3 )</td>
<td>3</td>
<td>95.09</td>
</tr>
</tbody>
</table>

After considering all models, the best result achieved by using wavelet network and sigmoid network for position and orientation angle respectively. As it can be seen in the above table, the output of the NARX models has high best fit. This value shows the accordance between the predicted output of the model with the real output of the data.
IV. RESULT AND DISCUSSION

In this section the performance of the best models from different approaches will be compared together. All models will be evaluated using other %30 of the data for validation. From each approach the model that has the best result is selected (the best model highlighted with bold and italic).

The best result is obtained by using NARX. NARX models which uses combination of nonlinear and linear function with regresses terms as a black-box model, clearly outperforms the above ANN and ANFIS. Its accuracy is in the range which is suitable for robots. An interesting observation is that the best models from ANN or ANFIS produces output that has a much higher error than average model from NARX. Table VI shows the comparison between all models errors.

<table>
<thead>
<tr>
<th>SOLUTION METHOD</th>
<th>PARAMETER</th>
<th>SSE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 LAYER FEED FORWARD NEURAL NETWORK WITH 32 NEURONS</td>
<td>x</td>
<td>0.0263</td>
<td>1.3184E-5</td>
</tr>
<tr>
<td></td>
<td>y</td>
<td>0.0354</td>
<td>1.77E-5</td>
</tr>
<tr>
<td></td>
<td>z</td>
<td>0.0274</td>
<td>1.37E-5</td>
</tr>
<tr>
<td></td>
<td>φ_x</td>
<td>0.0826</td>
<td>4.13E-5</td>
</tr>
<tr>
<td></td>
<td>φ_y</td>
<td>0.0281</td>
<td>1.40E-5</td>
</tr>
<tr>
<td></td>
<td>φ_z</td>
<td>0.0211</td>
<td>1.055E-5</td>
</tr>
<tr>
<td>4 LAYER FEED FORWARD NEURAL NETWORK WITH 18-13 NEURONS</td>
<td>x</td>
<td>0.0343</td>
<td>1.71E-5</td>
</tr>
<tr>
<td></td>
<td>y</td>
<td>0.0311</td>
<td>1.55E-5</td>
</tr>
<tr>
<td></td>
<td>z</td>
<td>0.0469</td>
<td>2.34E-5</td>
</tr>
<tr>
<td></td>
<td>φ_x</td>
<td>0.0914</td>
<td>4.57E-5</td>
</tr>
<tr>
<td></td>
<td>φ_y</td>
<td>0.0443</td>
<td>2.21E-5</td>
</tr>
<tr>
<td></td>
<td>φ_z</td>
<td>0.0373</td>
<td>1.865E-5</td>
</tr>
<tr>
<td>ANFIS - 2 MEMBERSHIP FUNCTION FOR EACH INPUT WITH 5 EPOCH</td>
<td>x</td>
<td>0.0181</td>
<td>9.05E-6</td>
</tr>
<tr>
<td></td>
<td>y</td>
<td>0.0129</td>
<td>6.45E-6</td>
</tr>
<tr>
<td></td>
<td>z</td>
<td>0.0261</td>
<td>1.3e-5</td>
</tr>
<tr>
<td></td>
<td>φ_x</td>
<td>0.0873</td>
<td>4.35E-5</td>
</tr>
<tr>
<td></td>
<td>φ_y</td>
<td>0.0209</td>
<td>1.04E-5</td>
</tr>
<tr>
<td></td>
<td>φ_z</td>
<td>0.0172</td>
<td>8.6E-6</td>
</tr>
<tr>
<td>NARX – WAVELET FOR POSITION AND SIGMOID FOR ORIENTATION ANGLE</td>
<td>x</td>
<td>1.4328E-07</td>
<td>1.1472E-10</td>
</tr>
<tr>
<td></td>
<td>y</td>
<td>1.2911E-08</td>
<td>1.0337E-11</td>
</tr>
<tr>
<td></td>
<td>z</td>
<td>5.2814E-04</td>
<td>6.3326E-07</td>
</tr>
<tr>
<td></td>
<td>φ_x</td>
<td>3.0808E-04</td>
<td>4.6327E-07</td>
</tr>
<tr>
<td></td>
<td>φ_y</td>
<td>2.5179E-04</td>
<td>3.7806E-07</td>
</tr>
<tr>
<td></td>
<td>φ_z</td>
<td>0.0030</td>
<td>4.4356E-06</td>
</tr>
</tbody>
</table>

The NARX models show very good results that outperform other method’s models. Fig. 5 shows the output of the NARX models for position and orientation angle.

The comparison shows that the model’s output follows that pattern of the real data to a satisfactory extent and indeed it is similar to the real output. It can be seen that the trajectory produced for position and orientation angle by NARX models is exactly a like the trajectory from the real data.
V. CONCLUSION

In this paper we applied the feed forward network structure, ANFIS and NARX to the forward kinematics problem for a Stewart Platform. The research results in this paper are interesting because they solve a problem for which, there is no known closed form solution, and also because there are many answer sets, to a given vector of leg lengths. The comparison between ANN, Fuzzy-Neural Network and NARX shows that the NARX has better performance. The trajectory produced by NARX models for position and orientation angle has high accordance with real trajectory from the data. This similarity is resulted from very low error of modeling the robot kinematic; as a result the position and orientation angle is very precise.

REFERENCES