Novel Adaptive Forward Neural MIMO NARX Model for the Identification of Industrial 3-DOF Robot Arm Kinematics

1. Introduction

The robot control problems can be divided into two main areas: kinematics control (the coordination of the links of a kinematics chain to produce the desired motion of the robot) and dynamic control (the driving of the actuator of a mechanism to follow the commanded positional velocities). In general, the control strategies used in robots involve position coordination in a Cartesian space by direct or indirect kinematics methods. Forward kinematics comprises the computation needed to find the joint angles for a given Cartesian position and the orientation of the end effectors. This computation is fundamental to the control of robot arms but it is very difficult to calculate a forward kinematics solution of the robot arm. For this solution, most industrial robot arms are designed using a nonlinear algebraic computation for finding the forward kinematics solution. It is well known in literature that there is no unique solution for forward kinematics. This is the reason why it is important to apply artificial neural network models to obtain a solution.
Novel approaches of artificial neural network (ANN) models have been proposed to control the motion of robot arms. In these works, two types of ANN models are used. The first kind of ANN model is the MLP (multi-layer perceptron), which is popular as a back propagation neural model. In this network, gradient descent types of learning rules are applied. The second kind of ANN model is the PPN (polynomial poly-processor neural network), where a polynomial equation was used. Here, work has been undertaken to find the best ANN configuration for the problem. It was found that between MLP and PPN, MLP gives better results as compared with PPN, considering the average percentage error as the performance index.

Alavandar and Nigam [1] developed a Neuro-Fuzzy-based approach for a forward kinematics solution for industrial robot arms. In their paper, it is shown that by using the ability of an Adaptive Neuro-Fuzzy Inference System (ANFIS) to learn from training data, it is possible to create an implementation of a representative fuzzy inference system using a BP neural network-like structure, with limited mathematical representation of the system. Computer simulations conducted on 2-DOF and 3-DOF robot arms showed the effectiveness of the approach.

Morris et al. [2] developed an artificial neural network for finding the kinematics of a robot arm using a look-up table. The neural networks utilized were multi-layered perceptrons with a back-propagation training algorithm. They used 5 hidden layer neurons with the rate of training \( \eta \) set at 2.018, and the momentum factor \( \alpha \) set at 0.54. The average percentage error - i.e., the percentage of the difference between the actual and the targeted/desired output - at the last iteration of the final session was 0.02% for the first joint and 0.3% for the second joint.

Guez and Ahmad [3] developed a solution to the forward kinematics problem in robots using a neural network. It was found that the neural network could be trained to generate a fairly accurate solution which would result in a minimal burden on the processing load of each control cycle and, thus, enable real-time robot control. The back propagation algorithm simulating a three-layer perceptron was employed to tackle this problem. Symmetric sigmoidal nonlinearity was used. The learning rate and the momentum term assumed the values of 0.1 and 0.4 respectively. In addition, the desired outputs were normalized between -0.9 and +0.9. The average error was less than 0.01 radians while the maximum error was 0.25 radians.

Karlik et al. [4] developed an improved approach to the solution of forward kinematics problems for robot arms. In their approach, a structured artificial neural network (ANN) was proposed to control the motion of a robot arm. Many neural network models use threshold units with sigmoid transfer functions and gradient descent-type learning rules. The learning equations used were those of the back propagation algorithm. In their work, the solution of the kinematics for a 6-DOF robot arm is implemented using an ANN. The training of the ANN required 6000 iterations.

Furthermore, various numerical and intelligent methods to solve the forward kinematics of industrial robot arms have been investigated in [5-10]. Recently, robust adaptive control approaches combining conventional methods with new learning techniques have been applied to the offline and online learning of robot arm dynamics [11-13]. Ahn and Anh in [14-15] have successfully optimized a NARX neural model of the forward kinematics of a PAM-based robot arm using a genetic algorithm. These authors - in [16] - have identified the forward kinematics of the industrial robot arm based on adaptive neural networks. The drawback of all these results is related to the forward kinematics of the industrial robot arm being considered as an independent decoupling system with the external force variation having negligible effect. Consequently, not all of the intrinsic cross-effect features of the forward kinematics of industrial robot arms have been represented in the resulting adaptive neural model.

This paper proposes the novel use of an adaptive neural NARX model to generate the Forward Neural MIMO NARX (FNMN) model for the nonlinear forward kinematics of a 3-DOF industrial robot arm system. A Back Propagation (BP) learning algorithm is used to process the experimental input-output data that is measured from the forward kinematics of the industrial robot arm system to optimize all of the nonlinear and dynamic features of this system. Thus, the BP algorithm optimally generates the appropriate neural weightings so as to perfectly characterize the dynamic features of the forward kinematics of the industrial robot arm. These good results are due to the proposed FNMN model, which combines the extraordinary approximating capability of the neural system with the powerful predictive and adaptive potentiality of the nonlinear ARX structure that is implied in the proposed FNMN model. Consequently, the proposed method for the industrial robot arm Forward Neural NARX model identification approach successfully modelled the nonlinear forward kinematics of the industrial robot arm system and performed well.

The rest of the paper is organized as follows. Section II introduces the calculation of the 3-DOF industrial robot
arm forward kinematics. Section III presents the novel neural NARX model using the forward kinematics model identification of the industrial 3-DOF robot arm. The results from the proposed forward kinematics identification are presented in Section IV. Finally, Section V makes some concluding remarks.

2. Forward Kinematics of the Industrial 3-Dof Robot Arm System

The Forward kinematics analysis for a planar 3-R robot arm appears to be complicated, but we can derive analytical solutions. Recall that the direct kinematics equations (1) are:

\[ x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \]  
\[ y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \]  

Call \( \Phi = \theta_1 + \theta_2 + \theta_3 \) and from the given coordinates, \( x, y \) and \( \Phi \), we want to find analytical expressions for the joint angles \( \theta_1, \theta_2 \) and \( \theta_3 \) in terms of the Cartesian coordinates.

Substituting \( \Phi \) into (1a) and (1b), we can eliminate \( \theta_3 \) so that we have two equations in \( \theta_1 \) and \( \theta_2 \):

\[ x - l_3 \cos \Phi = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \]  
\[ y - l_3 \sin \Phi = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \]  

Rename the left hand sides \( x' = (x - l_3 \cos \Phi) \) and \( y' = (y - l_3 \sin \Phi) \) for convenience. We can regroup the terms in (1c) and (1d), square both sides in each equation, and then we have:

\[ (-2l_1 x') \cos \theta_1 + (-2l_1 y') \sin \theta_1 + (x'^2 + y'^2 + l_3^2 - l_1^2) = 0 \]  

After rearranging the terms we get a single nonlinear equation in \( \theta_1 \). Hence, there are two solutions for \( \theta_1 \), given by:

\[ \theta_1 = \gamma + \sigma \cos \left[ \frac{-\left( x'^2 + y'^2 + l_3^2 - l_1^2 \right)}{2l_1 \sqrt{x'^2 + y'^2}} \right] \]  

with:

\[ \gamma = a \tan \left( \frac{-y' \pm \sqrt{y'^2 + l_3^2 - l_1^2}}{x'} \right) \]

and \( \sigma = \pm 1 \).

Substituting any one of these solutions back into equations (1c) and (1d) gives us:

\[ \cos(\theta_1 + \theta_2) = \frac{x' - l_1 \cos \theta_1}{l_2} \]
\[ \sin(\theta_1 + \theta_2) = \frac{y' - l_1 \sin \theta_1}{l_2} \]

Hence:

\[ \theta_2 = \tan\left( \frac{y' - l_1 \sin \theta_1}{x' - l_1 \cos \theta_1} \right) - \theta_1 \]  

and:

\[ \theta_3 = \Phi - \theta_1 - \theta_2 \]  

Equations (1f), (1g) and (1h) are the forward kinematics solution for the 3-R manipulator. For a given end effector’s position and orientation, there are two different ways of reaching it, each corresponding to a different value of \( \sigma \). These different configurations are shown in Figure 2:

In this paper, the testing data was taken from training data. In general, because the forward kinematics of a 6-DOF robot arm is a nonlinear mapping from a \( (X, Y, Z, \phi_x, \phi_y, \phi_z) \) space to a \( (\theta_1, \theta_2, \theta_3, \phi_1, \phi_2, \phi_3) \) space, it can be regarded as an input-output process with an unknown transfer function. The approximation of such a relationship is an example of the general approximation problem and as such it is well-suited to learning by an adaptive MIMO NARX neural model. Two problems
encountered when teaching a NN robot forward kinematics are the low accuracy of approximation and the need to create a teaching data set with a relatively large number of data points. In the case, where the kinematic parameters of each robot link are known, a simple kinematic model of the 3-DOF robot can be made based on equations 1(f) to 1(h) and a required number of data points can be readily generated. In the case where the joint parameters are unknown, the required data has to be obtained by measuring the position and orientation of the robot gripper although with a number of arm configurations. Such measurement is very difficult in practice.

3. Proposed Adaptive Forward Neural Mimo Narx (Fnmn) Model of the Nonlinear Forward Kinematics of the Industrial 3-Dof Robot Arm System

The adaptive Forward Neural MIMO NARX (FNMN) model used in this paper is a combination of the Multi-Layer Perceptron Neural Networks (MLPNNs) structure and the Auto-Regressive with exogenous input (ARX) model. Due to this combination, the proposed Forward Neural MIMO NARX (FNMN) model possesses a powerful universal approximating feature from the MLPNN structure and a strong predictive feature from the nonlinear ARX model.

A fully connected 3-layer feed-forward MLP-network with \( n \) inputs, \( q \) hidden units (also called “nodes” or “neurons”) and \( m \) output units is shown in Fig. 3.

In Figure 1, \( w_{10},...,w_{q0} \) and \( W_{10},...,W_{m0} \) are the weighting values of the bias neurons of the input layer and the hidden layer respectively.

We consider an Auto-Regressive with exogenous input (ARX) model with noisy input, which can be described as:

\[
A(q^{-1})y(t) = B(q^{-1})u(t-T) + C(q^{-1})e(t) \tag{2}
\]

with
\[
A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2}
\]
\[
B(q^{-1}) = b_1 + b_2 q^{-1}
\]
\[
C(q^{-1}) = c_1 + c_2 q^{-1} + c_3 q^{-2}
\]

where \( e(t) \) is the white noise sequence with a zero mean and unit variance, \( u(t) \) and \( y(t) \) are the input and output of the system respectively, \( q \) is the shift operator and \( T \) is the time delay.

From equation (2), and not considering the noise component \( e(t) \), we have the general form of the discrete ARX model in a z-domain (with the time delay \( T=n=1 \)):

\[
y(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + ... + b_n z^{-n}\]
\[
u(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n} \tag{3}
\]

in which \( n_u \) and \( n_m \) are the order of output \( y(z^{-1}) \) and input \( u(z^{-1}) \) respectively.

This paper investigates the potentiality of various simple adaptive Neural MIMO NARX models in order to exploit them in modelling, identification and control. The forward kinematic model of the industrial 3-DOF robot arm is investigated. Thus, by embedding a 3-layer MLPNN (with the number of neurons of the hidden layer equal 5) in a 1st-order ARX model, its characteristic equation induced from (3) is as follows:

\[
yhat(k) = b_{11} y(k-1) + b_{12} y(k-2) + b_{13} y(k-3) \]
\[
- a_{11} x(k-1) - a_{12} x(k-2) \tag{4a}
\]

\[
xhat(k) = b_{21} y(k-1) + b_{22} y(k-2) + b_{23} y(k-3) \]
\[
- a_{21} x(k-1) - a_{22} x(k-2) \tag{4b}
\]

![Figure 3. Structure of the feed-forward MLPNN.](image-url)
We will design the proposed Forward Neural MIMO NARX11 (FNMN11) model (with \( n_x = 1, n_r = 1, n_y = 1 \)) for the industrial 3-DOF robot arm with 5 inputs (including \( q_1(k), q_2(k), q_3(k) \)), recurrent delayed values \( y(k-1), x(k-1) \), and 2 output values \( y_{\text{out}}(k), x_{\text{out}}(k) \). We recall that three input values \( q_1(k), q_2(k) \) and \( q_3(k) \) represent the three joint angles [rad] of the 3-DOF robot arm and the two output values \( y_{\text{out}}(k) \) and \( x_{\text{out}}(k) \) represent the Cartesian coordinates \((x,y)\) of the 3-DOF robot arm end-effector respectively. Its structure is shown in Fig. 4a.

![Figure 4a. Structure of the proposed Forward Neural MIMO NARX11 (FNMN11) model.](image)

In this way, the parameters \( a_{i1}, a_{i2}, b_{i1}, b_{i2}, b_{i3} \) and \( a_{n1}, a_{n2}, b_{n1}, b_{n2}, b_{n3} \) of the ARX structure of the output variables \( y_{\text{out}}(t) \) and \( x_{\text{out}}(t) \) respectively, become adaptively nonlinear and will be determined from the weighting values \( W_{o} \) and \( W_{x} \) of the proposed Neural MIMO NARX11 model. This feature makes the adaptive Neural MIMO NARX model very powerful for modelling, identification and for model-based advanced control as well.

In the same way, we will design the proposed Forward Neural MIMO NARX22 (FNMN22) model (with \( n_x = 2, n_r = 2, n_y = 1 \)) for the industrial 3-DOF robot arm by embedding a 3-layer MLPNN (with the number of neurons of the hidden layer equal 5) in a 2\(^{rd}\)-order ARX model with its characteristic equation induced from (3) as follows:

\[
\begin{align*}
\dot{y}_{\text{out}}(k) = & b_{o1}(k) + b_{o2}(k) + b_{o3}(k-1) + b_{o4}(k-2) + \\
& + b_{o5}(k-1) - a_{o1}y(k-1) - a_{o2}x(k-1) - a_{o3}y(k-2) - a_{o4}x(k-2) \\
\dot{x}_{\text{out}}(k) = & b_{x1}(k) + b_{x2}(k) + b_{x3}(k-1) + b_{x4}(k-2) + \\
& + b_{x5}(k-1) - a_{x1}y(k-1) - a_{x2}x(k-1) - a_{x3}y(k-2) - a_{x4}x(k-2) \\
\end{align*}
\]  

(4b)

In this way, the proposed Forward Neural MIMO NARX22 (FNMN22) model (\( n_x = 2, n_r = 2, n_y = 1 \)) for the industrial 3-DOF robot arm illustrated in Fig. 4b is composed of 16 inputs (including \( q_1(k), q_2(k), q_3(k) \), \( q_1(k-1), q_2(k-1) \)), \( q_3(k-1), q_3(k-2) \), \( x(k-1), x(k-2) \), \( y(k-1), y(k-2) \), \( y_{\text{out}}(k), x_{\text{out}}(k) \). We should once more recall that the three input values \( q_1(k), q_2(k) \) and \( q_3(k) \) represent the three joint angles [rad] of the 3-DOF robot arm and the two output values \( y_{\text{out}}(k) \) and \( x_{\text{out}}(k) \) represent the Cartesian coordinates \((x,y)\) of the industrial 3-DOF arm end-effector, respectively. Its structure is shown in Fig. 4b.

![Figure 4b. Structure of the proposed Forward Neural MIMO NARX22 (FNMN22) model.](image)

Furthermore, the class of MLPNN-networks considered in this paper is confined to those having only one hidden layer and using sigmoid activation functions. From Figure 3, the predictive output value \( \hat{y}(t) \) (or \( y_{\text{out}}(t) \)) is calculated as follows:

\[
\begin{align*}
\hat{y}_{\text{out}}(w,W) &= \sum_{j=1}^{q} W_{ij} f_{j}\left( \sum_{i=1}^{n} w_{ij} z_{i} + w_{i0} \right) + W_{i0} \\
\end{align*}
\]  

(5)

The weights (specified by the matrices \( w \) and \( W \)) are the adjustable parameters of the network and they are determined from a set of examples through a process called training. The exemplars - or the training data, as they are usually called - are a set of inputs, \( u(t) \) and corresponding desired outputs, \( y(t) \).

We specify the training set by:

\[
Z^N = \left\{ (u(t), y(t)) \right\}_{t=1}^{N} \]

(6)

The objective of training is then to determine a mapping from the set of training data to the set of possible weights, \( Z^N \rightarrow \hat{\theta} \), so that the network will produce predictions \( \hat{y}(t) \), which in some sense are "closest" to the true joint angle outputs \( y(t) \) of the PAM robot arm.

The prediction error approach, which is the strategy applied here, is based on the introduction of a measure of closeness in terms of a mean sum of a square error (MSSE) criterion:

\[
E_{N}(\theta,Z^N) = \frac{1}{2N} \sum_{t=1}^{N} [y(t) - \hat{y}(\theta)]^2 \]  

(7)
Based on the conventional error Back-Propagation (BP) training algorithms, the weighting value is calculated as follows:

\[ W(k+1) = W(k) - \lambda \frac{\partial E(W(k))}{\partial W(k)} \]  

(8)

where \( k \) is the \( k \)th iterative step of the calculation and \( \lambda \) is the learning rate, which is often chosen as a small constant value.

More concretely, the weights \( W_i \) and \( w_i \) of the neural NAXR are then updated as:

\[ W_i(k+1) = W_i(k) + \Delta W_i(k+1) \]
\[ \Delta W_i(k+1) = \lambda \hat{\delta}_i \cdot O_j \]
\[ \hat{\delta}_i = \hat{y}_i(1 - \hat{y}_i)(y_i - \hat{y}_i) \]  

(9)

where \( \hat{\delta}_i \) is the search direction value of the \( i \)th neuron of the output layer (\( i=1 \rightarrow m \)), \( O_i \) is the output value of the \( j \)th neuron of the hidden layer (\( j=1 \rightarrow q \)), \( y \), and \( \hat{y}_i \) are the real and predicted outputs, respectively, of the \( i \)th neuron of output layer (\( i=1 \rightarrow m \)), and:

\[ w_j(k+1) = w_j(k) + \Delta w_j(k+1) \]
\[ \Delta w_j(k+1) = \lambda \delta_j \cdot u_i \]
\[ \delta_j = O_j(1 - O_j) \sum_{i=1}^{m} \delta_i w_{ij} \]  

(10)

in which \( \delta_j \) is the search direction value of the \( j \)th neuron of the hidden layer (\( j=1 \rightarrow q \)), \( O_i \) is the output value of the \( j \)th neuron of the hidden layer (\( j=1 \rightarrow q \)), and \( w \) is the input of the \( j \)th neuron of the input layer (\( j=1 \rightarrow n \)).

These results by equations (9) and (10) are demonstrated as follows in the case of the sigmoid being the activation function of the hidden and output layers. Consider the case of the output layer and the error to be minimized:

\[ E = \frac{1}{2} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 \]  

(11)

Using the chain rule method, we have:

\[ \frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial S_j} \frac{\partial S_j}{\partial W_{ij}} \]  

(12)

From equation (11), the following equation is derived:

\[ \frac{\partial E}{\partial \hat{y}_i} = \left( \hat{y}_i - y_i \right) \]  

(13)

with \( S_i = \sum_{j=1}^{q} W_{ij} O_j + \text{bias}_i \) as a sum calculation at the \( i \)th node of the output layer and \( \hat{y}_i = \frac{1}{1 + e^{-S_i}} \). This gives:

\[ \frac{\partial \hat{y}_i}{\partial S_i} = e^{-S_i} \frac{1 - 1}{(1 + e^{-S_i})^2} = \frac{1}{1 + e^{-S_i}} \]  

(14)

\[ \frac{\partial S_i}{\partial W_{ij}} = O_j \]  

(15)

Replace (13), (14), (15) with (12) and then substitute all of them into (8), then the following equation is derived:

\[ W_i(k+1) = W_i(k) + \Delta W_i(k+1) \]
\[ \Delta W_i(k+1) = \lambda \delta_j \cdot O_j \]
\[ \delta_j = O_j(1 - O_j)(y_j - \hat{y}_j) \]  

(16)

Equation (9) is, therefore, demonstrated.

Using the chain rule method, the same approach is used for updating the weights of the hidden layer, namely:

\[ \frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial O_j} \frac{\partial O_j}{\partial S_j} \frac{\partial S_j}{\partial W_{ji}} \]  

(17)

Then:

\[ \frac{\partial E}{\partial O_j} = \sum_{i=1}^{m} \frac{\partial E}{\partial S_j} \frac{\partial S_j}{\partial O_j} = \sum_{i=1}^{m} \left[ \frac{\partial E}{\partial S_j} \frac{\partial S_j}{\partial O_j} \left( \sum_{j=1}^{q} W_{ij} O_j + \text{bias}_i \right) \right] = \sum_{i=1}^{m} \left[ \frac{\partial E}{\partial S_j} W_{ij} \right] = \sum_{i=1}^{m} -\delta_i w_{ij} \]  

(18)

with \( S_j = \sum_{i=1}^{m} W_{ij} u_i + \text{bias}_j \) as the sum calculation at the \( j \)th node of the hidden layer and \( O_j = \frac{1}{1 + e^{-S_j}} \). We obtain:

\[ \frac{\partial O_j}{\partial S_j} = e^{-S_j} \frac{1 - 1}{(1 + e^{-S_j})^2} = \frac{1}{1 + e^{-S_j}} \left( 1 + \frac{1}{1 + e^{-S_j}} \right) = O_j(1 - O_j) \]  

(19)

\[ \frac{\partial S_j}{\partial W_{ji}} = u_i \]  

(20)
Replace (18), (19), (20) with (17) and then substitute all of them into (8). The following equation is derived:

\[
w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}(k+1) \\
\Delta w_{ij}(k+1) = \lambda \delta_j u_i \\
\delta_j = O_j (1 - O_j) \sum_{j=1}^{m} \delta_i W_{ij}
\]

Equation (10) is thus demonstrated.

4. Modelling and Identification Results

In general, the procedure which must be executed when attempting to identify a dynamical system consists of four basic steps:

- **STEP 1 (Getting the Training Data)**
- **STEP 2 (Selecting the Model Structure)**
- **STEP 3 (Estimating the Model)**
- **STEP 4 (Validating the Model)**

In Step 1, the identification procedure is based on experimental input-output data values measured from the forward kinematics of the industrial 3-DOF robot arm system. The input signals \(\theta_1, \theta_2, \theta_3\) (or \(q_1(k), q_2(k)\) and \(q_3(k)\) in the z-domain) represent the joint-angles applied to the three joints of the industrial robot arm in order to obtain a circular trajectory from the output signals \(x, y\) represented by the Cartesian coordinates of the end-effector of the industrial 3-DOF robot arm. Thus, the resulting output Cartesian coordinates \(x, y\) are chosen to formulate a circular trajectory of the robot arm’s end-effector. Figure 5a presents the collected input-output data composed of the three \(\theta_1, \theta_2, \theta_3\) angle inputs applied to the three joints of the industrial 3-DOF robot arm system and the responding \(x(t)\) and \(y(t)\) outputs of the industrial robot arm’s end-effector Cartesian coordinates.

This collected input-output data is used for training and validating the proposed Forward Neural MIMO NARX (FNMN) model. The Cartesian coordinates \((x, y)\) considered as reference outputs and the forward kinematics of the industrial 3-DOF robot arm angles \((\theta_1, \theta_2, \theta_3)\) are considered as reference inputs. Figure 5a shows that the input signals \(\theta_1, \theta_2, \theta_3\) or \(q_1(k), q_2(k)\) and \(q_3(k)\) in the z-domain represent joint-angles applied to the three joints of the industrial robot arm and the output signals \((x, y)\) represent the Cartesian coordinates of the end-effector of the industrial 3-DOF robot arm. From Figure 5a, we have the \((\theta_1, \theta_2, \theta_3)\) reference inputs (corresponding to \(q_1(k), q_2(k)\) and \(q_3(k)\) in the z-domain) and \((x, y)\) considered as reference outputs from \((0–1)\) [s] are used for training, while the \((x, y)\) reference outputs and the 3-DOF robot arm joint-angle inputs \((\theta_1, \theta_2, \theta_3)\) from \((1–2)\) [s] are used for validation purpose.

Back Propagation (BP) learning algorithm based on the error between the reference coordinates \((x, y)\) collected from the industrial 3-DOF robot arm system and the responding Cartesian coordinates \((x_0, y_0)\) collected from the designed FNMN model to update the weights of the proposed adaptive forward neural MIMO NARX model. Figure 5b and Figure 5c illustrate the identification scheme of the industrial 3-DOF robot arm using the proposed Forward Neural MIMO NARX11 (FNMN11) and Forward Neural MIMO NARX22 (FNMN22) models respectively.

![Figure 5a](image-url)  
**Figure 5a.** Training data composed of \((\theta_1, \theta_2, \theta_3)\) inputs and \((x, y)\) outputs for the identification process.
The second step relates to the selection of the model structure. The proposed Forward Neural MIMO NARX11 (FNMN11) and Forward Neural MIMO NARX22 (FNMN22) model structures are attempted. The block diagrams in Figure 5b and Figure 5c illustrate the identification scheme of these two proposed FNMN models. Figure 4a and Figure 4b illustrate the final Forward Neural MIMO NARX11 and Forward Neural MIMO NARX22 model structures of these two proposed FNMN models.

The third step estimates the values for the trained Forward Neural NARX model. The optimal fitness value for the BP-based optimization and identification process is calculated. The estimation result is presented in Figure 6. This figure represents the fitness convergence values of the proposed forward kinematics of the industrial 3-DOF robot arm system which correspond to two different FNMN models (Forward Neural NARX11 and Forward Neural NARX22 models) and all two were identified and optimized with the Back
Propagation (BP) learning algorithm. A good convergence of the minimized fitness value after 500 training iterations is shown in Figure 6 with a perfect Mean Sum of Scaled Error (MSSE) = 0.000016 and MSSE = 0.0000012 with the proposed FNMN11 and FNMN22 model respectively. The training result shows that proposed Forward Neural MIMO NARX (FNMF) model, with an MSSE value which is perfectly minimized, ensures to learn well dynamic and nonlinear features of the industrial 3-DOF robot arm.

These good results are attributed to how the proposed FNMN model combines the extraordinary approximating capability of the neural system with the powerful predictive and adaptive potentiality of the nonlinear NARX structure that is implied in the FNMF model. Consequently, the BP-based Forward kinematics of the industrial robot arm FNMF model addresses all of the nonlinear features of the forward kinematics of the industrial robot arm system that are implied in the input angle signals ($\theta_1$, $\theta_2$ and $\theta_3$) and the responding output Cartesian co-ordinates ($x,y$).

Figure 6. Fitness convergence of the forward kinematics of the industrial robot arm:
- a) Forward Neural MIMO NARX11 model;
- b) Forward Neural MIMO NARX22 model.

Figure 7. Validation of the proposed forward kinematics of the industrial 3-DOF robot arm:
- a) Forward Neural MIMO NARX11 model;
- b) Forward Neural MIMO NARX22 model.
The last step relates to the validating of the resulting nonlinear FNMN models. Applying the same training diagram in Figure 5b and Figure 5c, a good validating result demonstrates the performance of the resulting Forward Neural MIMO NARX model, as presented in Figure 7 and Figure 8. The error optimized nearly zero between the real industrial 3-DOF robot arm system responding (x,y) and the Forward Neural MIMO NARX model responding (xh, yh) asserts the very good performance of proposed FNMN model. The error shown in Figure 7 and Figure 8 emphasises the fact that the quality of the Forward Neural MIMO NARX22 model is similar to the Forward Neural MIMO NARX11 model. Finally, Figure 9 and Figure 10 illustrate the auto-tuning variation of the adaptive ARX parameters of the proposed Forward Neural MIMO NARX11 and NARX22 models of the industrial 3-DOF robot arm. In essence, the ten parameters a11, a12, a13, b11, b12, b13, and a21, a22, b21, b22, b23 of the two 1st-order ARX structure integrated in the proposed FNMN11 model were adaptively auto-tuned, as illustrated in Figure 9. Similarly, the twenty parameters a11, a12, a13, a14, b11, b12, b13, b14, b15 and a21, a22, a23, a24, b21, b22, b23, b24, b25, b26 of the two 2nd-order ARX structures integrated in the proposed FNMN22 model were adaptively and flexibly auto-tuned, as illustrated in Figure 10. These results show that the parameters of the
ARX structure integrated in the proposed FNMMN models now become adaptively nonlinear and will be adaptively determined from the optimized weighting values $W_b$ and $W_f$ of the Neural MIMO NARX model. This feature once more proves that the proposed adaptive Forward Neural MIMO NARX (FNMMN) model is very powerful and adaptive for identification and model-based advanced control.

In summary, Table 1 and Table 2 tabulate the optimized weighting values of the proposed Forward Neural MIMO NARX11 and Forward Neural MIMO NARX11 models. The final structures of the Forward Neural MIMO NARX11 and Forward Neural MIMO NARX22 models which were identified and optimized by the BP learning algorithm are shown in Figure 4a and Figure 4b respectively.

![Figure 10.](image)

**Figure 10.** Adaptive Parameters of the ARX Structure embedded in the proposed FNMMN22 model.

<table>
<thead>
<tr>
<th>$k$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>0</th>
<th>$k-1$</th>
<th>$k-2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.172</td>
<td>0.1081</td>
<td>0.1609</td>
<td>0.7753</td>
<td>-0.107</td>
<td>-0.699</td>
<td>-0.489</td>
<td>-0.489</td>
<td>0.016</td>
<td>0.582</td>
<td>2.5299</td>
</tr>
<tr>
<td>2</td>
<td>0.001</td>
<td>0.0004</td>
<td>0.0075</td>
<td>0.0066</td>
<td>0.0027</td>
<td>0.0018</td>
<td>0.0018</td>
<td>0.0018</td>
<td>0.003</td>
<td>51.618</td>
<td>13.859</td>
</tr>
<tr>
<td>3</td>
<td>0.0155</td>
<td>0.0011</td>
<td>0.0251</td>
<td>0.0004</td>
<td>0.0205</td>
<td>0.0128</td>
<td>0.0128</td>
<td>0.0128</td>
<td>0.004</td>
<td>3.7376</td>
<td>41.918</td>
</tr>
<tr>
<td>4</td>
<td>-0.05</td>
<td>-0.0299</td>
<td>0.0003</td>
<td>0.0043</td>
<td>0.0019</td>
<td>0.0016</td>
<td>0.0016</td>
<td>0.0016</td>
<td>0.001</td>
<td>31.64</td>
<td>7.8259</td>
</tr>
<tr>
<td>5</td>
<td>-0.0832</td>
<td>0.0236</td>
<td>-0.025</td>
<td>0.03</td>
<td>-0.0118</td>
<td>-0.0117</td>
<td>-0.0117</td>
<td>-0.0117</td>
<td>0.008</td>
<td>28.81</td>
<td>45.182</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.1159</td>
<td>-0.1296</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1.** Optimized weights of the Forward NEURAL MIMO-NARX11 – Total Number of weighting values = 57.

<table>
<thead>
<tr>
<th>$k$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>0</th>
<th>$k-1$</th>
<th>$k-2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0041</td>
<td>0.0072</td>
<td>0.0042</td>
<td>0.0012</td>
<td>0.0001</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
</tr>
<tr>
<td>2</td>
<td>0.0041</td>
<td>0.0072</td>
<td>0.0042</td>
<td>0.0012</td>
<td>0.0001</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
</tr>
<tr>
<td>3</td>
<td>0.0041</td>
<td>0.0072</td>
<td>0.0042</td>
<td>0.0012</td>
<td>0.0001</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
</tr>
<tr>
<td>4</td>
<td>0.0041</td>
<td>0.0072</td>
<td>0.0042</td>
<td>0.0012</td>
<td>0.0001</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
</tr>
<tr>
<td>5</td>
<td>0.0041</td>
<td>0.0072</td>
<td>0.0042</td>
<td>0.0012</td>
<td>0.0001</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 2.** Optimized weights of the Forward NEURAL MIMO-NARX22 – Total Number of weighting values = 97.
5. Conclusions

In this paper, a new approach in the form of a Forward Neural MIMO NARX (FNMN) model is introduced for the modelling and identification of the forward kinematics of the industrial robot arm-based intelligent actuator. The training and testing results show that the newly proposed forward dynamic neural MIMO NARX model presented in this paper can be used in online control with better dynamic properties and with significant robustness. This resulting proposed intelligent model is quite suitable for application in the modelling, identification and control of various complex plants, including linear and nonlinear processes, and without concern for any large change in the external environment.

6. Acknowledgments

This research was completely supported by the NAFOSTED (Vietnam).

7. References


