

Adaptive MIMO Neural Network Model Optimized by Differential Evolution Algorithm for Manipulator Kinematic System Identification

Nguyen Ngoc Son

Faculty of Electrical and Electronics Engineering
HoChiMinh City University of Technology
HoChiMinh City, Vietnam
son.nguyen.fet@gmail.com

Ho Pham Huy Anh

DCSELAB / FEEE
HoChiMinh City University of Technology
HoChiMinh City, Vietnam
hphanh@hcmut.edu.vn

Abstract—In this paper, an adaptive MIMO neural network model is used for simultaneously modeling and identifying the forward kinematics of a 3-DOF robot manipulator. The nonlinear features of the robot manipulator kinematics system are modeled by an adaptive MIMO neural network model based on differential evolution algorithm. A differential evolution algorithm is used to optimally generate the appropriate neural weights so as to perfectly characterize the nonlinear features of the forward kinematics of a 3-DOF robot manipulator. This paper supports the performance of the proposed differential evolution algorithm in comparison with the conventional back-propagation algorithm. The results show that the proposed adaptive MIMO neural network model trained by the differential evolution algorithm for identifying the forward kinematics of a 3-DOF robot manipulator is successfully modeled and performed well.

Keywords—Differential Evolution (DE); Back-Propagation Algorithm; Nonlinear System Identification; Robot Manipulator.

I. INTRODUCTION

The neural networks were considered as a promising approach for identifying nonlinear system. Studies in [1] and [2] indicated that neural networks can be used effectively in identifying and controlling nonlinear system. Their paper proposed static and dynamic back-propagation algorithm to optimally generate the weights of neural networks and to adjust of parameters. Anh in [3] proposed a neural MIMO NARX model used to identifying the industrial 3-DOF robot manipulator. In this paper, the back-propagation algorithm was used to optimally generate the weights of neural MIMO NARX model. The process of identification based on experimental input-output training data of the forward kinematics of a robot manipulator. Simulation results showed that the performance identification using neural MIMO NARX model trained by back-propagation algorithm performed well. However, the drawback of the back-propagation algorithm applied in the studies [4], [5], [6] and [7] was that the convergence speed became slow, a large computation for learning and the cost function might lead to local minima.

To overcome this drawback, the evolutionary algorithm (EA)-based training procedures are considered as promising alternatives. Differential evolution (DE) is considered as one of the most powerful stochastic real-parameter optimization algorithms in current use. The DE algorithm emerged as a

very competitive form of evolutionary computing with the first published article on DE appeared as a technical report of R. Storn and K. V. Price in 1995 [8]. The DE algorithm was capable of handling non-differentiable, nonlinear, and multimodal objective functions. DE method had been used to train neural model through optimizing real and constrained integer weights. Its simplicity and straightforwardness in implementation, excellent performance, fewer parameters involved, and low space complexity, had made DE as one of the most powerful tool in the field of optimization [8]. The paper [9]-[12] successfully developed a DE-based trained neural network for nonlinear system identification. Thus these papers demonstrated that DE algorithm can be effectively used for training neural network models applied in versatile applications.

In this paper, we introduce a novel adaptive MIMO (Multiple Input Multiple Output) neural network model based on differential evolution for modeling and identifying the forward kinematics of a 3-DOF robot manipulator. This paper also supports the performance of the proposed differential evolution algorithm in comparison with the conventional back-propagation algorithm. The results show that the proposed adaptive MIMO neural network model based on differential evolution algorithm for identifying the forward kinematics of a 3-DOF robot manipulator is successfully modeled and performed well.

II. KINEMATICS OF THE INDUSTRIAL 3-DOF ROBOT MANIPULATOR

In this section, the forward and inverse kinematics of a 3-DOF robot manipulator are investigated. The industrial 3-DOF robot manipulator structure is illustrated in Fig.1.

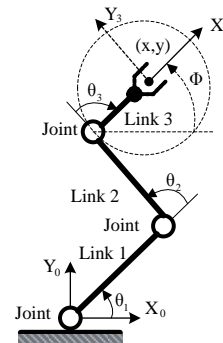


Figure 1. The industrial 3-DOF robot manipulator structure

Based on the vector algebra solution to analyze the graph, the coordinates of the robot end-effector can be solved as follows

$$\begin{aligned} x &= l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) + l_3 \cos(q_1 + q_2 + q_3) \\ y &= l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) + l_3 \sin(q_1 + q_2 + q_3) \end{aligned} \quad (1)$$

Where, θ_1, θ_2 and θ_3 represent for joint angle and x and y represent for the position of the end-effector of a 3-DOF robot manipulator system. Call $f = q_1 + q_2 + q_3$. By eliminating θ_1, θ_2 and θ_3 from “(1)”, we obtain

$$\begin{aligned} q_1 &= \arctan(y - l_3 \sin f, x - l_3 \cos f) - \arctan(k_2, k_1) \\ q_2 &= \arctan(\sin q_2, \cos q_2) \\ q_3 &= \arctan(\sin f, \cos f) - (q_1 + q_2) \end{aligned} \quad (2)$$

Where,

$$\begin{aligned} k_1 &= l_1 + l_2 \cos q_2 \quad \text{and} \\ k_2 &= l_2 \sin q_2 \\ \cos q_2 &= \frac{(x - l_3 \cos f)^2 + (y - l_3 \sin f)^2 - l_1^2 - l_2^2}{2l_1 l_2} \end{aligned}$$

Based on analysis above, the kinematic parameters include length and angle of each robot link. In some cases, the parameters of each robot link can be obtained from the CAD models of robot manipulator or can be measured from the individual part of the robot. A simple kinematics of 3-DOF robot manipulator can be made based on “(1)” and “(2)”. In other cases, these parameters are unknown. The kinematics of 3-DOF robot manipulator can be modeled and identified by a proposed adaptive MIMO neural network model optimized by differential evolution algorithm.

III. ADAPTIVE MIMO NEURAL NETWORK MODEL BASED ON DE ALGORITHM

In this section, a novel adaptive MIMO neural network model based on differential evolution algorithm (DE-AMNN) is now investigated for modeling and identifying the forward kinematics of a 3-DOF robot manipulator. The AMNN model is combined between the Multilayer Perceptron Neural Network (MLPNN) structure and the Auto-Regressive with exogenous input (ARX) model. Due to this combination, the AMNN model possesses both of powerful universal approximating feature from MLPNN structure and strong predictive feature from ARX model. The forward kinematics of a 3-DOF robot manipulator is applied by embedding a 3-layer MLPNN in a 1st order ARX model. The block diagram of DE-AMNN is illustrated in Fig.2. Where, $u(t) = (\theta_1, \theta_2, \theta_3)$ or (q_1, q_2, q_3) represents for joint angle and $y(t) = (x, y)$ or (p_x, p_y) represents for the position of the end-effector of a 3-DOF robot manipulator.

Based on the differential evolution algorithm, we do training AMNN model for manipulator kinematics system

identification. DE can be applied to global searches within the weight space of a typical feed-forward neural network. Output of a neural network is a function $\hat{y}(t|q)$ of synaptic weights θ and input values $u(t)$. In the training process, both the input vector $u(t)$ and the output vector $y(t)$ are known and the synaptic weights in θ are adapted to obtain appropriate functional mappings from the input $u(t)$ to the output $y(t)$. Generally, the adaptation process can be carried out by minimizing the network error function E_N which is based on the introduction of a measure of closeness in terms of a mean sum of square error (MSSE) criterion:

$$E_N(q, Z^N) = \frac{1}{2N} \sum_{t=1}^N (\hat{y}(t) - y(t|q))^2 \quad (3)$$

Where, the training data Z^N is specified by $Z^N = \{\hat{y}(t), y(t) | t = 1, \dots, N\}$. The optimization goal is to minimize the objective function E_N by optimizing the values of the network weights $q = (w_1, w_2, \dots, w_D)$, where D is the number of weights of the AMNN model. Now, we explain the working steps involved in employing DE identification algorithm as follows:

Step 1: Parameter setup. Choose the parameters of population size NP , the boundary constraints of optimization variables, the mutation factor (F), the crossover rate (C), and the stopping criterion of the maximum number of generations (G_{max}).

Step 2: Initialization of the population. Create a population from randomly chosen object vectors with dimension NP

$$P_G = (q_{1,G}, q_{2,G}, \dots, q_{NP,G})^T, \quad G = 1, \dots, G_{max} \quad (4)$$

$$q_{i,G} = (w_{1,i,G}, w_{2,i,G}, \dots, w_{D,i,G}), \quad i = 1, \dots, NP \quad (5)$$

Where D is the number of weights in the AMNN model; i is index to the population and G is the generation to which the population belongs.

Step 3: Evaluate all the candidate solution inside population for a specified number of iterations.

Step 4: For each i^{th} candidate in population select the random variables

$$r_1, r_2, r_3 \in [1, 2, \dots, NP], \quad \text{except } r_1 = r_2 = r_3 = i \quad (6)$$

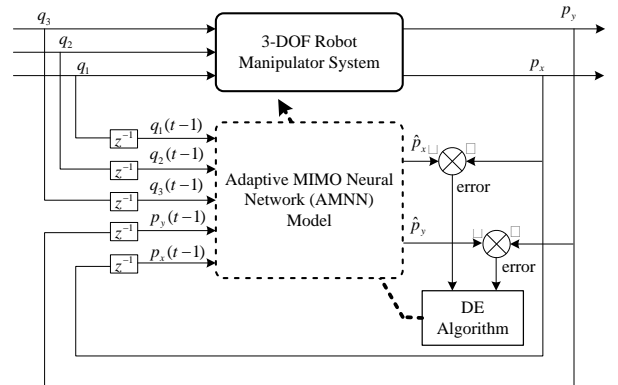


Figure 2. The forward kinematics system identification using DE-AMNN

Step 5: Apply mutation operator to each candidate in population to yield a mutant vector

$$mv_{j,i,G+1} = w_{j,r_1,G} + F(w_{j,r_2,G} - w_{j,r_3,G}), \text{ for } j = 1, \dots, D \quad (7)$$

Where F is the mutation factor, $F \hat{\in} [0,1]$.

Step 6: Apply crossover each vector in the current population is recombined with a mutant vector to produce trial vector

$$tv_{j,i,G+1} = \begin{cases} mv_{j,i,G+1} & \text{if } rand_j [0,1] \leq C \\ w_{j,i,G} & \text{otherwise} \end{cases} \text{ where } C \hat{\in} [0,1] \quad (8)$$

Step 7: Apply selection between the trial vector and target vector

$$q_{i,G+1} = \begin{cases} tv_{i,G+1} & \text{if } E_N(y, \hat{y}(u, q_{i,G+1})) \leq E_N(y, y(u, q_{i,G})) \\ q_{i,G} & \text{otherwise} \end{cases} \quad (9)$$

Step 8: Repeat step 4 to 7 until stopping criteria is reached

IV. MODELING AND IDENTIFICATION RESULTS

In general, the procedure which must be executed when attempting to identify the forward kinematics of a 3-DOF robot manipulator consists of four basic steps as follows:

A. Getting training data

By using the forward kinematics of industrial 3-DOF robot manipulator to generate a collection of experimental data relating the joint angles to the position of the end-effector. The input signals $u(t) = (q_1, q_2, q_3)$ represent for joints angle applied to the 3-DOF robot manipulator in order to obtain a curve trajectory from the output signals $y(t) = (p_x, p_y)$ represent for the position of the end-effector. Fig.3 shows a collected input-output data composed of the three input signals $q_1(t)$, $q_2(t)$, and $q_3(t)$ and the two output signals $p_x(t)$ and $p_y(t)$. The data set composed of input-output signal estimation is used for training, while the data set composed of input-output signal validation is used for validation purpose. Where, the data set composed of input-output signal estimation and the data set composed of input-output signal validation are differently.

B. Select model structure

Assuming that a data set has been acquired, the next step is to select a model structure. The idea is to select the regressors based on inspiration from linear system identification and then determine the best possible network architecture with the given regressors as inputs. This paper investigates the AMNN model structure as follows:

Regression vector

$$\varphi(t) = [Py(t-1) Px(t-1) q_1(t-1) q_2(t-1) q_3(t-1)]^T \quad (10)$$

And predictor

$$\hat{y}(t|\theta) = y(t|t-1, \theta) = g(\varphi(t), \theta) \quad (11)$$

Where $\varphi(t)$ is a vector containing the regressors, θ is a vector contain the weights and g is the function realized by the neural network. The structure of AMNN model that includes a fully connected 3-layer feed-forward MLPNN with 5 inputs, 5 hidden neurons and 2 outputs units, is illustrated in Fig.4.

C. Estimate model

Based on DE training algorithm, we have results of weighting θ . The AMNN model is estimated or determined the structure of the regression vector, the additional argument NN has to be passed.

D. Validate model

This step is to test the network using input data sets not used in the training process. The error is again examined as above. If it is of an acceptable value, then the network has successfully generalized and can be used with confidence as a model of the real plant. The AMNN model is said to possess the ability of generalization when the system input-output relationship computed by the network is approximately correct for input-output patterns never used in the training of the network.

Finally, we present the performance of identifying the forward kinematics of a 3-DOF robot manipulator of the proposed AMNN model based on differential evolution and compare with the conventional back propagation algorithm. Table 1 gives some parameters used in identification. Fig.5 shows the comparison of training MSSE for BP and DE approaches. Fig.6 shows the identification performance of the forward kinematics of a 3-DOF robot manipulator using DE algorithm and BP algorithm.

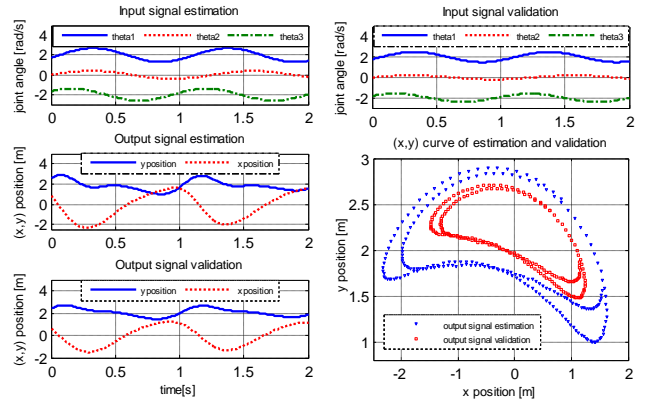


Figure 3. Collected data composed for identifying the forward kinematic of a 3-DOF robot manipulator

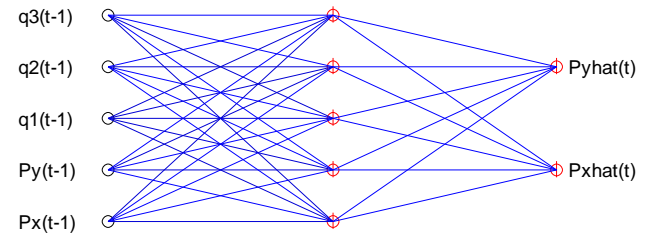


Figure 4. The AMNN model with 5 hidden neurons

TABLE I. PARAMETERS USED IN IDENTIFICATION

Method	Parameter	Value
General	Total sampling number, T	200
	Number of hidden layer neurons	5
	Number of iterations (generations)	3000
	Upper and lower bound on weighs	[-1 1]
BP-AMNN	BP learning parameter, η	0.0001
DE-AMNN	Population size, NP	10
	Mutation factor, F	0.4
	Crossover constant, C	0.37

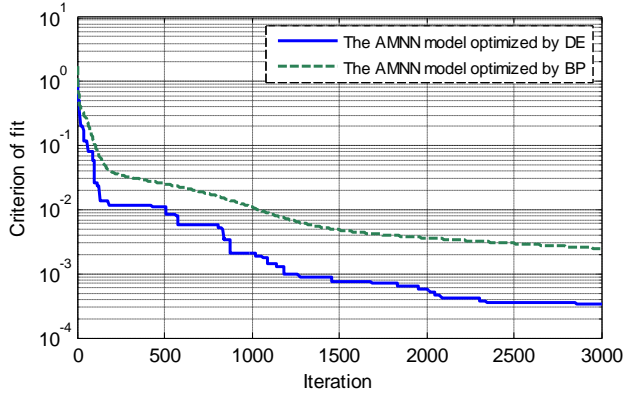


Figure 5. Comparison of training MSSE for BP and DE approaches

Based on results above, we see that the forward kinematics of a 3-DOF robot manipulator can be simultaneously modeled and identified by the AMNN model optimized by the differential evolution algorithm is possessing faster convergence and better identification performance than the back propagation algorithm.

V. CONCLUSION

This paper introduces a new approach study of a novel adaptive MIMO neural network model based on differential evolution for simultaneously the modeling and identifying the forward kinematics of a 3-DOF manipulator. The results show that the robot manipulator kinematic system is successfully modeled and performed well. Moreover, the proposed differential evolution algorithm applied to an adaptive MIMO neural network model performed better

results in term of faster convergence and lower MSSE error than conventional back propagation algorithm. Hence, this new method is promising for efficiently identifying and controlling not only the nonlinear 3-DOF robot manipulator system but also other highly nonlinear dynamic systems.

ACKNOWLEDGEMENTS

This research is supported by DCESELAB and funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number B2011-20b-02TĐ.

REFERENCES

- [1] Narendra, K.S., and Parthaasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," IEEE Trans Neural Networks, 1990, vol.1, pp 4-27.
- [2] Kuschewski, John G. Hui S. and Zak S.H. "Application of feedforward neural networks to dynamical system identification and control", IEEE Transactions on Control Systems Technology, 1993, vol. 1, pp 37-49.
- [3] Anh H.P.H. and Nam.N.T. "Novel Adaptive Forward Neural MIMO NARX Model for the Identification of Industrial 3-DOF Robot Arm Kinematics", International Journal of Advanced Robotic Systems, 2012, vol.9, pp.1-12.
- [4] Patra JC, Kot AC, "Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks", IEEE Transaction on Systems, Man and Cybernetics, 2002, Part B, vol.22, pp.505-511.
- [5] X.G. Wang, Z. Tang, H. Tamura, M. Ishii, W.D. Sun, "An improved backpropagation algorithm to avoid the local minima problem", Neurocomputing 56, 2004, pp.455 – 460.
- [6] M. Gori, A. Tesi, "On the problem of local minima in back-propagation", IEEE Trans. Pattern Anal. Mach. Intell. 14, 1992, pp.76–86.
- [7] Qun Dai, Ningzhong Liu, "Alleviating the problem of local minima in Backpropagation through competitive learning", Neurocomputing 94, 2012, pp.152–158.
- [8] Storn R. and Price K.V. "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces", International Conference on Swarm Intelligence, USA, Technical Report. TR-95-012, 1995.
- [9] Saikat Singha Roy, Joyshri Das, Susovan Mondal, "Effective System Identification Using Fused Network and DE Based Training Scheme", International Journal of Soft Computing and Engineering (IJSCE), Volume-3, Issue-3, July 2013, pp. 105-112.

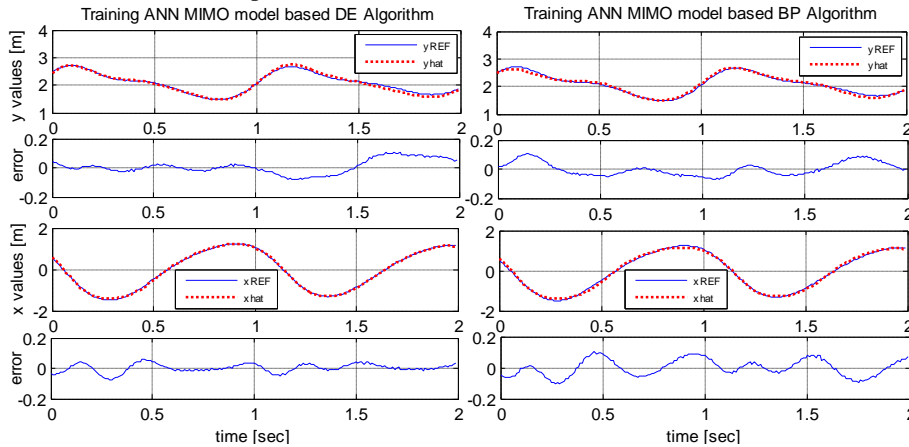


Figure 6. Identification performance of the 3-DOF robot kinematic system DE and BP algorithm