Efficient System Identification using a Low Complexity Nonlinear Network with Differential Evolution and its variant based Training Schemes

H. Pal Thethi
School of Electronics
KIIT University,
Bhubaneswar, India

Babita Majhi
Dept. of Information Technology
ITER,Siksha 'O' Anusandhan University, Bhubaneswar, India

G. Panda
School of Electrical Sciences
Indian Institute of Technology
Bhubaneswar, India

ABSTRACT

Direct modeling plays a very important role in many engineering applications including telecommunication, power system, image processing, VLSI design, biological processes, control engineering and geophysics applications. In case of control and telecommunication applications, direct modeling is used for channel estimation, parameter estimation and forecasting. There are standard algorithms and models which can be conveniently used for effectively identifying the parameters of simple direct and inverse systems. However, in practice we encounter with various complex systems, whose direct models needs to be created for various applications. As an illustration, the system can be non linear, dynamic or both of it. In such situations, creation of direct models is a difficult task. It is evident from the literature survey that, many sincere attempts have been made to create direct model of such complex systems. However, their performance has been observed to be unsatisfactory. Therefore in the present work, a sincere attempt has been made to address all these issues and provide possible satisfactory solutions by using low complexity nonlinear network and population based differential evolution(DE) based learning algorithm.

General Terms

Direct modeling, dynamic systems, low complexity nonlinear network

Keywords

System identification for dynamic systems, FLANN, DE

1. INTRODUCTION

Identification of a nonlinear dynamic plant is a major area in engineering today. System identification is widely used in a number of applications like biological processes [1], control system [2], signal processing [3] and communication engineering [4]. Many practical systems used in process control, robotics and autonomous system are nonlinear and dynamic in nature. To find a perfect model of these type of plants is a challenging task. There are certain classical parameterized models such as Wiener-Hamarstein [5], Voltera Series [6] and Polynomial identification model [7-8] which offer a reasonable precision, but the problem with these methods is that they involve lot of computational complexity. Subsequently, many neural network based models using multi-layer perception (MLP), radial basis function (RBF) and recurrent neural network have been proposed for nonlinear system identification problem. For basic neural network generally back propagation (BP) is used as an adaptive algorithm, to provide better accuracy. Earlier, Nerandra and Parthasarathy [9] have employed the multilayer perceptron (MLP) networks for effective identification and control of dynamic systems like truck-backer-upper problem [10].However, the major disadvantage of earlier methods is that, they employ derivative based learning algorithm such as back propagation algorithm (BP), to train the system parameters which at times lead to local minima thereby leading to incorrect estimation. On the other hand the functional link artificial neural network (FLANN) is basically a single layer structure in which nonlinear mapping of the input is achieved by expanding them with nonlinear functions. The literature survey reveals that the identification models need further improvement in terms of achieving performance accuracy and architectural simplicity. These two issues have been addressed in this work. Firstly, a single layer nonlinear architecture incorporating nonlinear mapping of the inputs have been introduced as the back bone of the model. Secondly, the feed forward as well as feedback parameters are proposed to be updated more accurately with DE based learning rule. The work has been organized into five sections. In Section 2 a brief introduction of nonlinear identification scheme is presented. A low complexity nonlinear architecture which serves as the backbone of the model is dealt in Section 3. It also outlines the fundamental of DE algorithm and its variants which are used for training the weights of the model. The DE based training scheme used in the proposed identification model is developed in same section. The simulation study of some benchmark problem and the results obtained are outlined in Section 4. Finally in Section 5 the conclusion of the investigation is dealt.

2. IDENTIFICATION OF NONLINEAR DYNAMIC SYSTEMS

System Identification is defined as the problem of determining a mathematical model satisfying a set of input-output data. Once a system has been identified, its output can be predicted for a given input. Fig 1 shows an identification model of nonlinear dynamic plant.

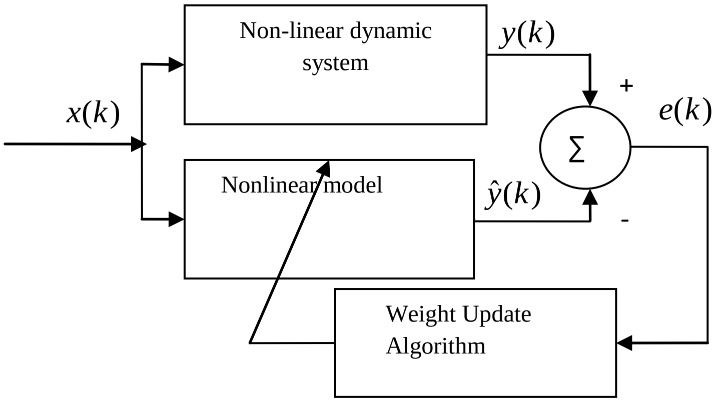


Fig 1 A nonlinear system identification scheme

, where $x(k)$ is the input to both the plant and the model and $y(k)$ and $\hat{y}(k)$ are the desired and estimated outputs at k^{th} instant respectively. The objective of the identification task is to minimize the error $e(k)$ recursively, such that $\hat{y}(k)$ approaches the desired plant output when same input $x(k)$ is applied to both the plant and the model. Three types of single-input single-output (SISO) plants [9] used in the study are described in form of difference equations (1) to (3).

Type-I

$$y(k) = \sum_{j=0}^{n-1} ajy(k-j) + g[x(k), x(k-1),..x(k-m-1)] \quad (1)$$

$$\text{y (k)} = \text{f[y(k-1),y(k}-2),....\text{y(k}-\text{n}+1)] + \sum_{\text{i}=0}^{\text{m}-1} b_{\text{i}}\text{x(k)}-\text{i)} \quad (2)$$

$$\text{y(k)} = \text{f[y (k-1),y(k}-2),....\text{y(k}-\text{n)]} + \sum_{\text{i}=0}^{\text{m}-1} a_{\text{i}}\text{x(k}-\text{i)} + \text{g[x(k),x(k}-1),....\text{x(k}-\text{m}+1) \quad (3)$$

where, $x(k)$ and $y(k)$ are the input and output of the SISO plant respectively at the k^{th} time instant under the condition that $m \le n$. Here $a_i(i \le n-1)$ and $a_i(i \le m-1)$ are the parameters of the feed forward and feedback paths of the plant. In addition $f(.)$ and $g(.)$ represent the nonlinear function associated with the output. The error signal obtained by the difference between plant and model outputs as well as the output information are used by a suitable learning algorithms to train the weights of the model so that the squared error value progressively decreases to a minimum value as iteration proceeds. When the squared error attains a lowest value, training is stopped and the adaptive structure corresponding to the last weight vector represents the desired identification model.

3. Development of a novel nonlinear Identification scheme

Fig 2 shows a single layered nonlinear structure proposed by Pao, which is capable of forming complex decision regions by generating nonlinear decision boundaries [25]. In this structure, the nonlinear adaptive architecture input dimension is increased by nonlinearly mapping the input patterns by using trigonometric functions. For nonlinear dynamic system identification, a similar structure has been proposed in [26] in which the weights of the model are updated using a steepest decent algorithm. In order to identify dynamic plants a series-parallel scheme is employed during training phase [9] where the feedback is taken from the plant output instead of the model.

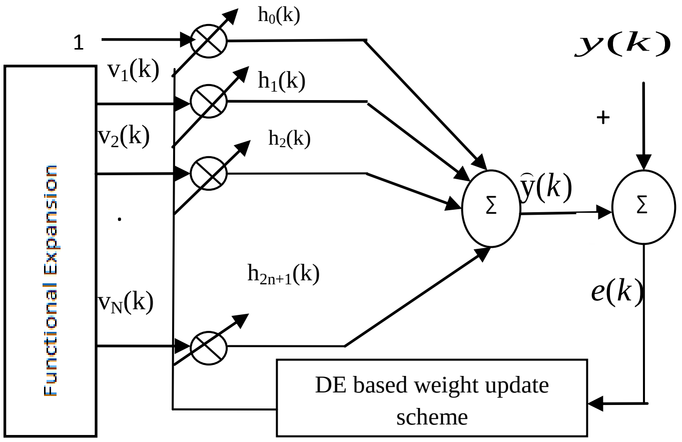


Fig 2 Structure of the nonlinear identification model

Each input $x(k)$ undergoes an nonlinear expansion and then applied to an adaptive linear combiner whose weights are updated by using adaptive algorithm. In [26] trigonometric expansion has been proposed, because it has yielded better performance for most of the applications. Accordingly in the proposed model sine and cosine expansions have been adopted. The expanded vector $v(k)$ of $x(k)$ is written as follows:

$$V(k) = [1, \sin\{\Pi x(k)\}, \cos\{\Pi x(k)\}...\sin\{n\Pi x(k)\}, \cos\{n\Pi x(k)\}] \quad (4)$$

$$= [v_0(k), v_1(k)...v_{2n+1}(k)]^T \quad (5)$$

If ***n*** numbers of sine and cosine expansions of input samples are made and the first term is an unity input then after expansion the total number of terms become $N = 2n+1$. Then the corresponding weight vector related to the k^{th} input vector is mathematically defined by the following equation:

$$H(k) = [h_0(k), h_1(k), h_2(k)...h_{2n+1}(k)]^T \quad (6)$$

Then the estimated output of the model is given by:

$$\hat{\boldsymbol{Y}}(\boldsymbol{k}) = \boldsymbol{V}^T(\boldsymbol{k}).\boldsymbol{H}(\boldsymbol{k}) \quad (7)$$

If the plant output at k^{th} instant is denoted as $y(k)$ then the corresponding error generated from the model is given by:

$$e(k) = y(k) - \hat{y}(k) \qquad (8)$$

This nonlinear structure employs single but efficient sine/cosine expansions and does not require more layers to offer the desired nonlinearity. Because of this the heavy computational requirement as in case of multilayer artificial neural network (ANN) is avoided and hence it is named as a low complexity adaptive structure. For designing the proposed model its weight are adjusted using the input and error information of the model and DE based training scheme. Before this scheme is detailed a brief overview of DE is provided in this section. Further in this section other standard forms of DE which are employed for performance comparison are outlined.

3.1 Basic Principle of Differential Evolution and its variants

Differential Evolution (DE) is a global optimization algorithm and adapts an encoding scheme with real valued number as used in binary GA. In DE, initially some vectors which are possible solution within a D-dimensional search space are randomly created, and then evolved over a time, to explore the entire search area, so that location corresponding to a minimum of objective function is achieved. The initial population is denoted by NP which is represented as $X_{i,j}(g),(i=0,1..(NP-1))$ and $(j=1,2,...D)$, where i is the population j is the number of parameters and g is the generation to which the population belongs.

Stepwise algorithm:

Initialization: The upper and lower bounds of each parameter are specified before population initialization. A random number is assigned to each parameter of every vector a value from within the prescribed range. As an example ($g=0$) the value of a j^{th} parameter of an i^{th} vector is given by

$$\mathrm{x}_{\mathrm{j,i}}(0) = rand_j(0,1).(p_{j,U} - p_{j,L}) + b_{j,L} \qquad (9)$$

Where p_U and p_L are upper and lower bound, of that variable.

Mutation: In mutation operation the difference of two randomly selected vectors are multiplied by a constant factor and then added with a randomly selected vector from the population .Equation (10) shows how a mutant vector; $v_{i,j}(g+1)$ is created.

$$\mathrm{v}_{\mathrm{i,j}}(g+1) = x_{r1,j}(g) + F.(x_{r2,j}(g) - x_{r3,j}(g)) \qquad (10)$$

This type of mutation is called as de/rand/1. There are other forms of creating the mutant vector like de/rand/2, de/best/1, de/best/2 and de/current to best/2 which are obtained as given in equations (11) to (14) :

$$\mathrm{V}_{\mathrm{i,j}}(\mathrm{g+1}) = \mathrm{X}_{\mathrm{r5,j}}(\mathrm{g}) + \mathrm{F.}(\mathrm{X}_{\mathrm{r1,j}}(\mathrm{g}) + \mathrm{X}_{\mathrm{r2,j}}(\mathrm{g}) - \mathrm{X}_{\mathrm{r3,j}}(\mathrm{g}) - \mathrm{X}_{\mathrm{r4,j}}(\mathrm{g})) \qquad (11)$$

$$\mathrm{V}_{\mathrm{i,j}}(\mathrm{g+1}) = \mathrm{X}_{\mathrm{best,j}}(\mathrm{g}) + \mathrm{F.}(\mathrm{X}_{\mathrm{r2,j}}(\mathrm{g}) - \mathrm{X}_{\mathrm{r3,j}}(\mathrm{g})) \qquad (12)$$

$$\mathrm{V}_{\mathrm{i,j}}(\mathrm{g+1}) = \mathrm{X}_{\mathrm{best,j}}(\mathrm{g}) + \mathrm{F.}(\mathrm{X}_{\mathrm{r1,j}}(\mathrm{g}) + \mathrm{X}_{\mathrm{r2,j}}(\mathrm{g}) - \mathrm{X}_{\mathrm{r3,j}}(\mathrm{g}) - \mathrm{X}_{\mathrm{r4,j}}(\mathrm{g})) \qquad (13)$$

$$\mathrm{V}_{\mathrm{i,j}}(\mathrm{g+1}) = \mathrm{X}_{\mathrm{i,j}}(\mathrm{g}) + \lambda.(\mathrm{X}_{\mathrm{best,j}}(\mathrm{g}) - \mathrm{X}_{\mathrm{i,j}}(\mathrm{g})) + \mathrm{F.}(\mathrm{X}_{\mathrm{r1,j}}(\mathrm{g}) - \mathrm{X}_{\mathrm{r2,j}}(\mathrm{g})) \qquad (14)$$

The scale factor, *F* is a positive real number mostly lying between 0 to 2 and controls the rate at which the population evolves.

Crossover: Subsequently a trail vector $u_{i,j}(g+1)$ is created using crossover operator:

$$u_{i,j}(g+1) = \begin{cases} v_{i,j}(g+1) \text{ if rand } (0,1) \le \text{ CR or j} = \mathrm{j}_{\mathrm{rand}} \\ x_{i,j}(g) \text{ if rand } (0,1) > \text{ CR or j} \ne \mathrm{j}_{\mathrm{rand}} \end{cases} \qquad (15)$$

The crossover ratio within the range 0 to 1, controls the fraction of parameter values that are to be copied from the mutant vector. If the value of a first random number is less than the chosen CR then the corresponding element of mutant vector is inherited to the target vector otherwise it is copied from the trial vector. This process is repeated for all elements and for the entire population.

Selection: If the objective function value of the trial vector, $U_{i,j}$ has an equal or lower than that of its target vector, $X_{i,j}$, it replaces the target vector in the next generation; otherwise, the target retains its place in the population for at least one generation.

In other words

$$x_{i,j}(g+1) = \begin{cases} u_{i,j}(g) \text{ if } \mathrm{f(u_{i,j})} \le \mathrm{f(x_{i,j})} \\ x_{i,j}(g) \text{ otherwise} \end{cases} \qquad (16)$$

3.2 DE based training scheme of the model parameters

The identification problem can be viewed as an optimization problem. Referring to Figs 1 and 2 it may be observed that the squared error value needs to be minimized so that the plant and the model outputs would match. This can be achieved either through an one time or iterative solution. In the proposed approach the DE optimization technique is suitably employed to iteratively minimize the squared error cost function. To achieve this objective the following DE based identification algorithm is proposed. In this approach instead of minimizing the instantaneous error, the mean squared error computed from a number of input samples is minimized in an iterative manner using DE tool. In this section the algorithm pertaining to identification of nonlinear plant using DE based training of weights of adaptive structure is outlined in steps. For this purpose the identification schemes depicted in Figs 1 and 2 are used to develop the algorithms. In this study a set of weight vector (trail vector) each consisting of N weights is considered as the population of the DE(which needs optimization). The mean square error (E) of the identification scheme is considered as the cost or objective function which is iteratively minimized by suitably varying the trail vectors. In this case the cost function due to p^{th} trail vector is computed from (17) by successively feeding m input samples and generating m errors $e(m, p)$ from the model.

$$E(p) = \frac{1}{M}\sum_{m=1}^{M} e^{2}(m, p) \qquad (17)$$

The various steps involved in the algorithm are

Step 1: Initialize the DE parameter:

P = Number of members in the population each corresponding to the weight vector of the model
F = Scale factor which controls the rate at which the population evolves.
C_R = Crossover ratio which is a constant value, which selects whether the new population is copied from the trial vector or from the target vector.
N = number of weights of the model present in the trail vector
g = generation number
M = number of input samples.

Step 2: Referring to Figs 1 and 2, M uniformly distributed random signals generated between [-1, 1] are again generated and are used as output. Similarly P sets of D-dimensional random vectors between [-1, 1] are generated to represent the trail vectors which constitutes the set of weight vectors H of the model.

Step 3: The plant output acts as the training signal and the estimated output is computed from adaptive model using (7).

Step 4: Each of the training samples, $y(k)$ is compared with the corresponding estimated output $\hat{y}(k)$ and thus M errors are produced using (8).

Step 5: The MSE (E) defined in (17) is calculated for each p^{th} trail vector.

Step 6: Corresponding to each trail weight vector three vectors are randomly selected from the initial population and a corresponding mutant weight vector is obtained by using (10) by suitably choosing a scale factor F . This process continues for all the trail vectors and the desired mutant vectors are generated.

Step 7: After mutations a random number between 0 to 1 is generated and compared with CR. Then using (17) a population of target vector is created.

Step 8: Based on the survival of fittest principle one out of the trail and corresponding target vectors which yields lower mean square error, E is chosen. In this way N sets of weight vectors are obtained after the first generation. The average of the MSE contributed by finally chosen all the weight vector is computed.

Step 9: Steps 2 to 9 are repeated so long the average MSE in each generation continues to decrease and remains almost constant at the end.

Step 10: According to equation (15) population for next generation are selected from the trail vector or from the target vector and new population is created for next generation.

4. SIMULATION STUDY

In this section simulation study of three types of standard dynamic plants explained in Section 2 are used for identification using the proposed method. Three types of plants chosen for identification are:

Type-1: Pole-zero plants with nonlinearity on the input side.

Type-2: Pole-zero plants with nonlinearity on the output side.

Type-3: Pole-zero plants with nonlinearity introduced on both sides.

For training purpose zero-mean random input uniformly distributed between -0.5 to 0.5 is used to feed both the plants and model. But for validating the performance of the proposed scheme sinusoidal input [9] defined in (18) is employed in all cases. The testing is analyzed by parallel scheme. The input to the identified model is given as

$$x(k) = \begin{cases} \sin(2\pi k/250) \\ 0.8\sin(2\pi k/250) + 0.2\sin(2\pi k/25) \end{cases} \qquad (18)$$

For comparing the performance and assessing the potentiality the proposed method of identification (Scheme-3) gradient descent (Scheme-1)[26] and GA(Scheme-2) based training schemes are also simulated. For assessing and comparing the performance of different models normalized MSE defined in (19) as performance index I

$$I = \frac{1}{\sigma^{2}}\sum_{s=1}^{S}[y(s) - \hat{y}(s)]^{2} \qquad (19)$$

4.1 Identification of SISO Dynamic Systems Type-I Plants

In this example, the plant is represented by the difference equation [9] given in equation (20)

$$\mathrm{y}(k) = 0.3y(k-1) + 0.6y(k-2) + g[x(k)] \quad (20)$$

where g represents a nonlinear function. In the present simulation three nonlinearities defined in (21), (22) and (23) are used.

$$\mathrm{g}_1(\mathrm{x}) = 0.6\sin(\pi \mathrm{x}) + 0.3\sin(3\pi \mathrm{x}) + 0.1\sin(5\pi \mathrm{x}) \qquad (21)$$

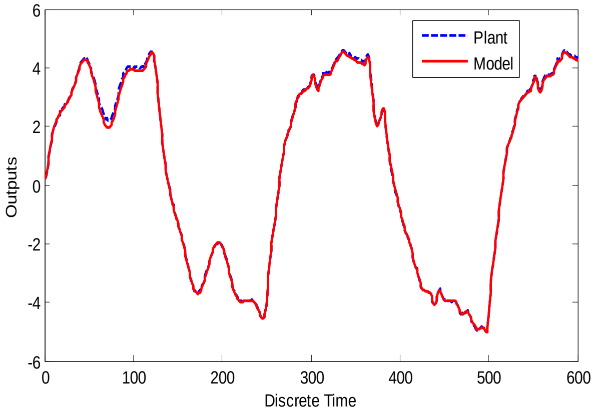
$$\mathrm{g}_2(\mathrm{x}) = \frac{4.0x^{3} - 1.2x^{2} - 3.0x + 1.2}{0.4x^{5} + 0.8x^{4} - 1.2x^{3} + 0.2x^{2} - 3.0} \qquad (22)$$

$$g_3(x) = 0.5\sin^3(\pi x) - \frac{2.0}{x^3+2.0} - 0.1\cos(4\pi x) + 1.125 \qquad (23)$$

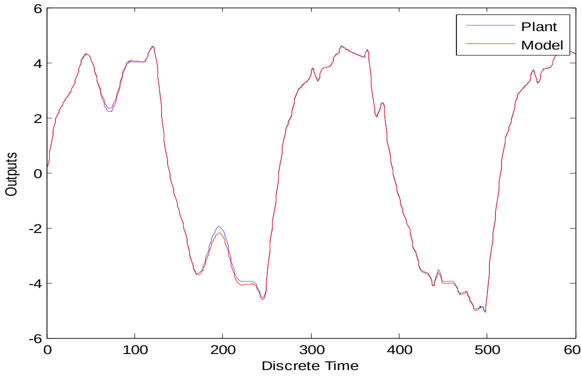
The corresponding model used for identification is given by (24)

$$\hat{y}_p(k) = 0.3y(k-1) + 0.6y(k-2) + G[x(k)] \qquad (24)$$

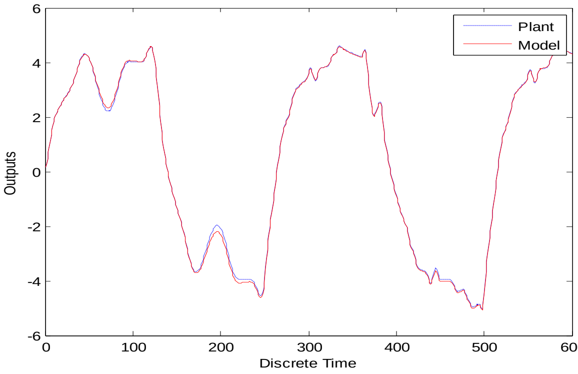
where $G[x(k)]$ represents the nonlinear schemes 1 to 3 used in the simulation study. The input is expanded to ten terms for (21) and eleven terms for (22) and (23) respectively. In scheme 3 the training is carried out for 500 iterations. The mutation probability and selection rate are chosen to be 0.05 and 0.5 respectively. In scheme 1 the data used are : CR =0.5, F =0.5, number of generations = 500 and DE/best/1 mutation process. Both the convergence parameter μ and the momentum factor η are chosen to be 0.1 for scheme-2.The response of the models of three schemes are shown in Figs 3,4 and 5 respectively.



(a) Scheme-1 model of Type-1 plant

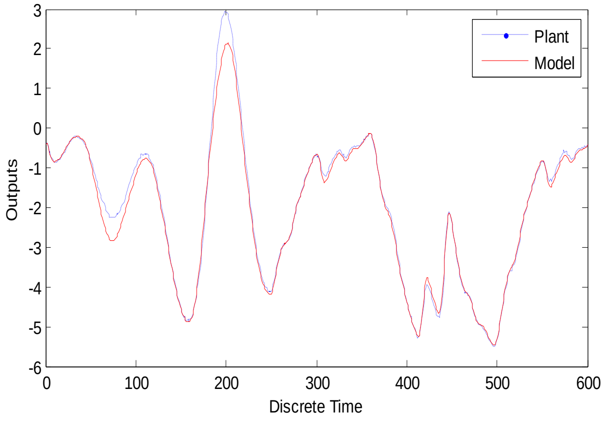


(b) Scheme-2 model of Type-1 plant

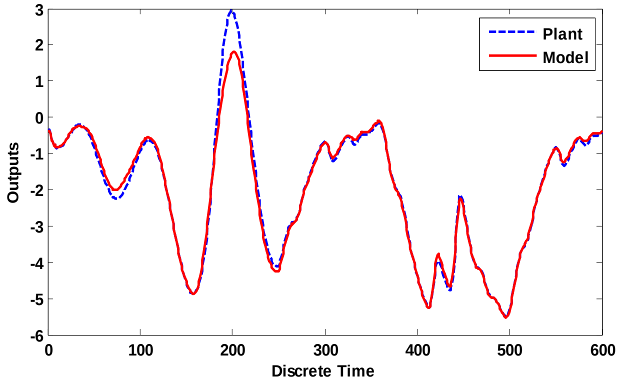


(c) Scheme-3 model of Type-1 plant

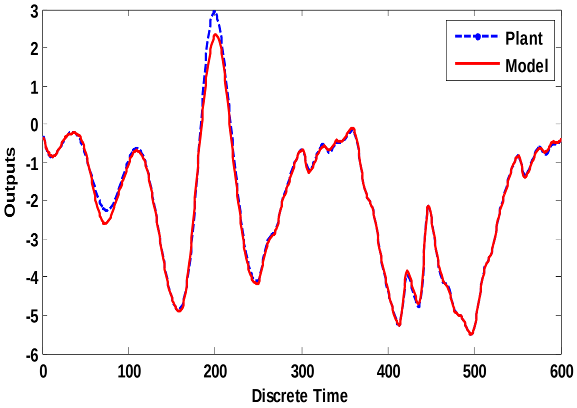
Fig 3 Comparison of responses of Type-1 plant with nonlinearity (21) between three different models



(a) Scheme-1 model of Type-1 plant

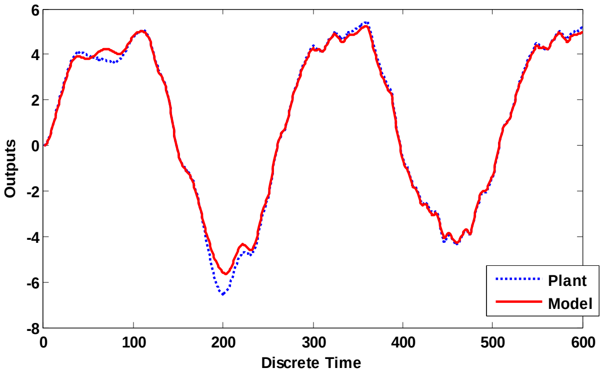


(b) Scheme-2 model of Type-1 plant

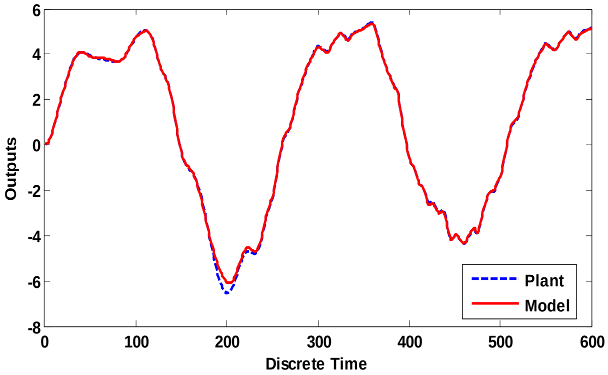


(c) Scheme-3 model of Type-1 plant

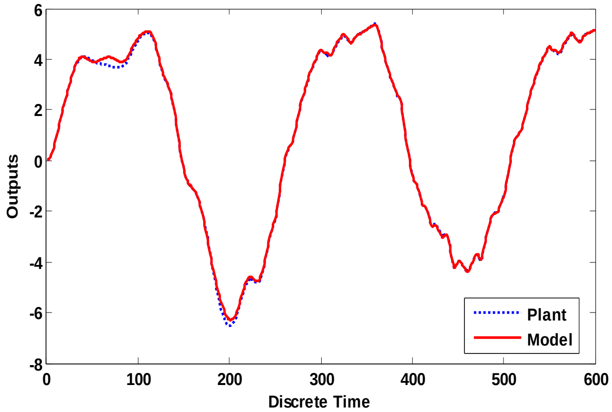
Fig 4 Comparison responses of Type-1 plant with nonlinearity (3.22) and three different models



(a) Scheme-1 model of Type-1 plant



(b) Scheme-2 model of Type-1 plant



(c) Scheme-3 model of Type-1 plant

Fig 5 Comparison of responses of type-1 plant with nonlinearity (23) and three different models

4.2 Type-2 Plants

In plants simulated under this category is represented by the difference equation [3.9]:

$$y(k) = f[y(k-1), y_n(k-2)] + x(k) \qquad (25)$$

the nonlinear function associated with g(.) is given by

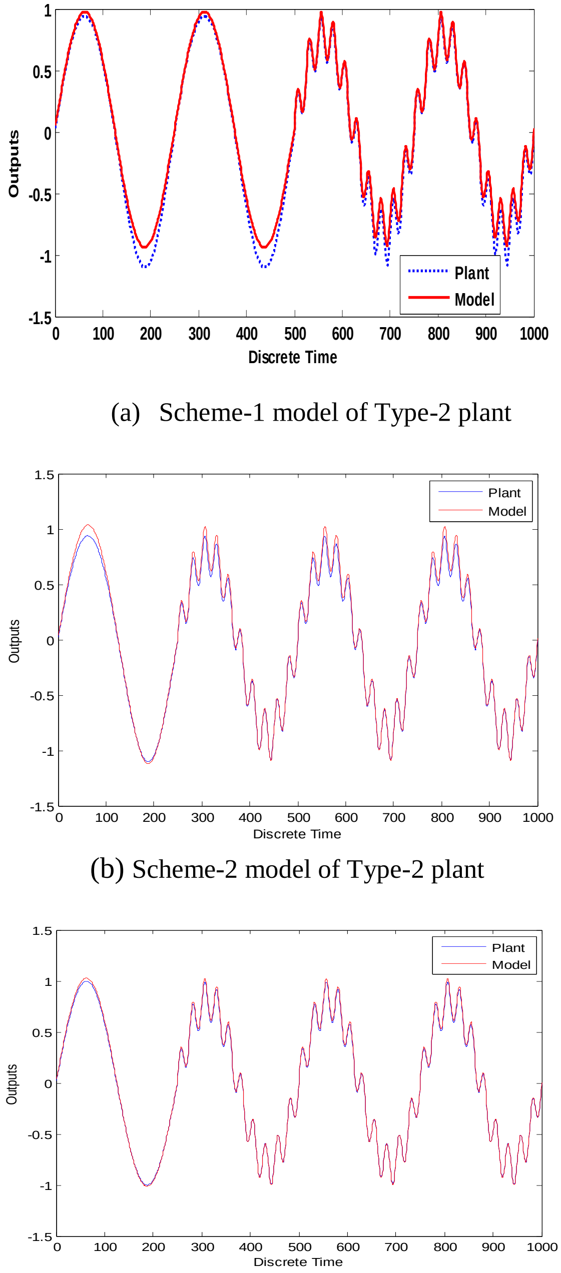
$$g(y(x),y(k\text{-}1)) = \frac{y(k)y(k\text{-}1)(y(k)\text{-}2.5)(y(k)\text{-}1.0)}{1.0 + y^2(k) + y^2(k-1)} \qquad (26)$$

To identify this plant the model used is of the form

$$\hat{y}(k+1) = G(y\ (k)y\ (k-1) + x(k)) \qquad (27)$$

Where G (.) represents the nonlinear schemes 1 to 3 used with simulation study.

The convergence and momentum parameters for Scheme-2 model are 0.05 and 0.1 respectively. All other parameters in other two schemes are same as used in Type-I case. The responses obtained from the plant and the models are shown in Fig.6. From the plots it is the proposed model of scheme provides accurate identification compared to that of other two schemes.



(a) Scheme-1 model of Type-2 plant

(b) Scheme-2 model of Type-2 plant

(c) Scheme-3 model of Type-2 plant

Fig 6 Comparison of responses of type-2 plant with nonlinearity (25) and three different models.

4.3 Type-3 Plants:

In this case the plant to be identified is of type-3 and is given by:

$$y_p(k+1) = f[y_p(k)] + g[x(k)] \tag{28}$$

Where f(.) and g(.) is nonlinearity associated as

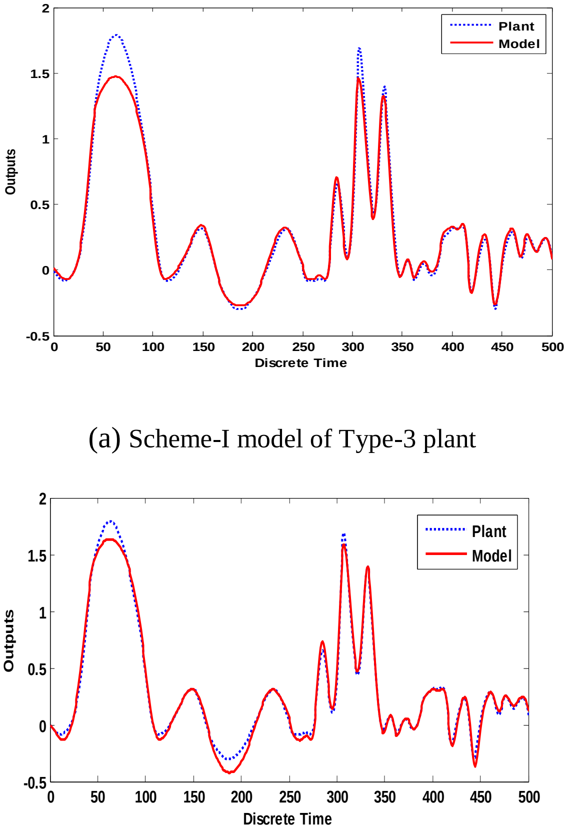
$$f(y) = \frac{y(y+3)}{1+y^2} \tag{29}$$

$$g(x) = x(x-0.8)(x-0.5) \tag{30}$$

The identification model for the plant (27) is given by:

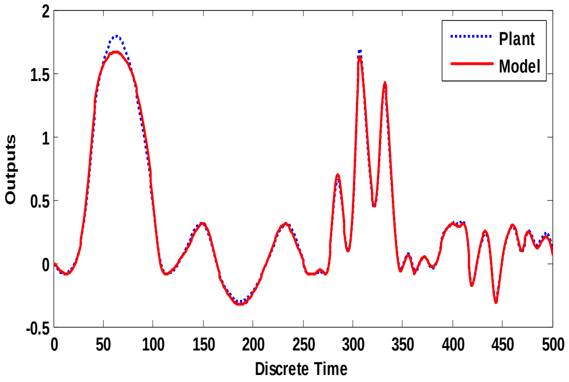
$$y_p(k+1) = G_1[y_p(k)] + G_2[x(k)] \tag{31}$$

where $G_1(.)$ and $G_2(.)$ are the nonlinear schemes 1 to 3 proposed in the paper. Various parameters used in this case are same as in case of type-I simulation. The results of identification are shown in Fig.7 where it is observed that the proposed scheme-3 shows better performance compared to other two.



(a) Scheme-I model of Type-3 plant

(b) Scheme-2 model of Type-3 plant



(c) Scheme-3 model of Type-3 plant

Fig 7 Comparison of responses of type-3plants with nonlinearities (28) and (29) and three different models.

Comparison of identification for each scheme and results are listed in Table-.1

Plants with different non linearities	No. of expansion used	NMSE in dB		
		Scheme-1	*Scheme-2*	*Scheme-3*
Type-1with (21)	10	-33.5807	-34.3218	-40.8738
Type-1 with(22)	11	-18.7387	-17.9180	-22.1631
Type-1 with(23)	11	-23.9670	-31.5231	-32.5503
Type-2 with(25)	9	-19.3256	-24.8696	-31.6770
Type-3 with (28) and (29)	14	-16.4079	-20.1859	-24.6417

Table-1 Comparison of NMSE obtained with different schemes

The results of Table-1 clearly demonstrate enhanced performance of the proposed DE based model compared to that of existing two schemes. For training the parameters of the proposed model all five variants of De have been employed. However the simulation study reveals that for identifying plants of Type 1 and 2 DE/best/1 and that for type-3 DE/current to best/2 mutation show best performance respectively. Therefore the results presented in this section pertain to that provided by such choice. The convergence characteristics of obtained during training of various DEs are shown in Figs 8,9 and 10.

Further, the convergence patterns of various variants of DEs are obtained during training and are displayed in Figs. 8 to 10 for Types 1-3 plants respectively. From these plots it is observed that for Type 1 and 2 the DE/best/1 performs best, but for Type-3 the DE/current to best/2 shows better results.

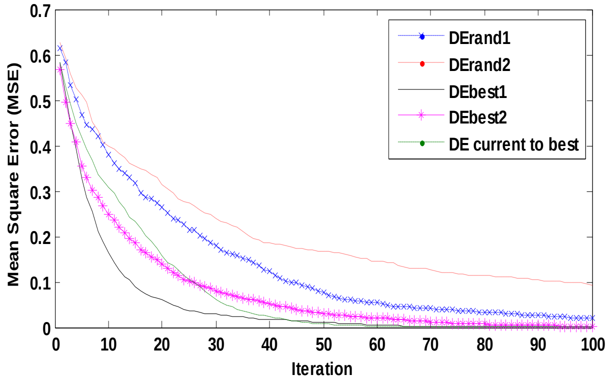


Fig 8 Comparison of convergence performance different DEs for identification of Type-1 Plant

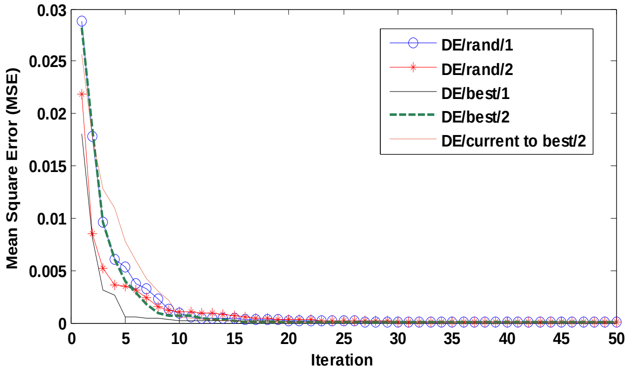


Fig 9 Comparison of convergence performance of different DEs for identification of Type-2 Plant

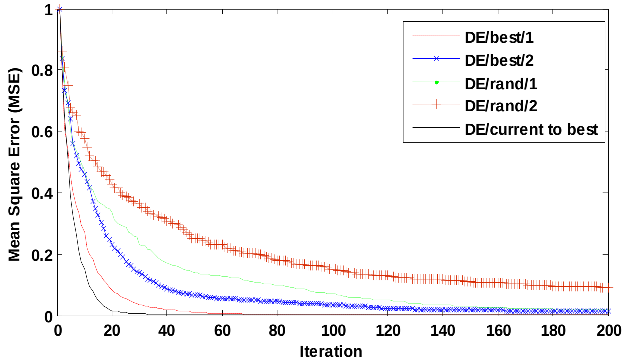


Fig 10 Comparison of convergence performance of different DEs for identification of Type-3 Plant

5. CONCLUSION

This work proposes a new adaptive identification scheme using a low complexity nonlinear structure with its connecting weights trained by a DE-based optimization strategy. For achieving improved performance different variants of DE has been employed and the appropriate ones have been chosen. Three different complex dynamic nonlinear plants are selected for identification. A DE based identification algorithm is proposed to develop the models. Two different existing identification models have also been simulated for comparing the results with those of the proposed model. Various simulation results reveal enhanced identification performance of the proposed method compared to other competing models. The proposed method is shown to be superior both in terms of less computation and simple learning strategy. On comparing the performance of different DEs, it is observed that DE/best/1 outperforms better than others for Types 1 and 2 plants whereas it is DE/current to best/2 for Type 3 plants.

6. REFERENCES

[1] Y. Xie, B. Guo, L. Xu, J. Li, P. Stoica, "*Multistatic adaptive microwave imaging for early breast cancer detection*", IEEE Trans. Biomed. Eng. 53 (8) (2006) 1647–1657.

[2] S. Chen, S.A. Billings, "*Representation of non-linear systems: the NARMAX model"*, Int. J. Control 49 (1989) 1013–1032

[3] .M. Adjrad, A. Belouchrani, "*Estimation of multi component polynomial phase signals impinging on a multisensor array using state-space modeling"*, IEEE Trans. Signal Process. 55 (1) (2007) 32–45.

[4] H. Hujiberts, H. Nijmeijer, R. Willems, "*System identification in communication with chaotic systems"*, IEEE Trans. Circuits Syst. I 47 (6) (2000) 800–808.

[5] F. Dinga, T. Chenb, "*Identification of Hammerstein nonlinear ARMAX systems"*, Automatica 41 (2005) 1479–1489.

[6] M. Schetzmen, "***The Voltera and Winner Theories on Nonlinear Systems****"*, Wiley, New York, 1980.

[7] E. Hernandez, Y. Arkun, "*Control of nonlinear systems using polynomial ARMA models"*, AICHE J. 39 (3) (1993) 446–460.

[8] T.T. Lee, J.T. Jeng, "*The Chebyshev polynomial-based unified model neural networks for functional approximation"*, IEEE Trans. Syst. Man Cybern. B 28 (1998) 925–935.

[9] Narendra, K. S., & Parthasarathy, K. "*Identification and control of dynamical systems using neural networks",* IEEE Transactions on Neural Networks, vol. 1 no. 1, pp. 4–27, Mar. 1990.

[10] Nguyen, D. H., & Widrow, B. "*Neural networks for self-learning control system.*" International Journal of Control, vol. 54, no. 6, pp. 1439–1451, 1991.

[11] M. Srinivas, L. M. Patnaik, "*Genetic Algorithm: A Survey*'', IEEE computer, vol. 27, no. 6, pp. 17-26, 1994.

[12] D. B. Fogel, "*An Introduction to Simulated Evolutionary Optimization*", IEEE Trans. On SMC., vol. 24, no. 1, pp.

[13] W. Atmar, "*Notes on the Simulation of Evolution*", IEEE Trans. On SMC., vol. 24, no. 1, pp. 130-147, 1994.

[14] R. Storn and K. Price, "*Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces*", Journal of Global Optimization, vol.11, pp.341-359, 1997.

[15] Vavak, F., Fogarty, T. and Jukes, K "*A genetic algorithm with variable range of local search for tracking changing environments*". In Proceedings of the 4th Conference on Parallel Problem Solving from Nature, 1996.

[16] K. Price, R. M. Storn, and J. A. Lampinen, "*Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)"*, 1st ed. New York: Springer, 2005, ISBN: 3540209506.

[17] J. Vesterstroem and R. Thomsen, "*A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems*," Proc. Congr. Evol. Comput., vol. 2, pp. 1980–1987, 2004.

[18] J. Andre, P. Siarry, and T. Dognon, "*An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization*," Advance in Engineering Software 32, pp. 49–60, 2001.

[19] O. Hrstka and A. Kuˇcerová, "*Improvement of real coded genetic algorithm based on differential operators preventing premature convergence*," Advance in Engineering Software 35, pp. 237–246, 2004.

[20] R.Storn, "*Differential evolution design for an IIR-filter with requirements for magnitude and group delay*". Technical Report TR-95-026, International Computer Science Institute, Berkeley, CA 1995.

[21] Qing, A.: "*Electromagnetic inverse scattering of multiple perfectly conducting cylinders by differential evolution strategy with individuals in groups (GDES)"*. IEEE Trans. Antennas Propagate. 52(5), 1223–1229 (2004).

[22] Lakshminarasimman, L., Subramanian, S.: "*Hydrothermal optimal power flow using modified hybrid differential evolution"*. Caledonian J. Engg. 3(1), 8–14 (2007a)

[23] Lakshminarasimman, L., Subramanian, S.: "*Short-term scheduling of hydrothermal power system with cascaded reservoirs by using modified differential evolution"*. IEE Proc. Gener. Transm. Distrib. 153(6), 693–700 (2006)

[24] Yang, S., Gan, Y.B., Qing, A, " *Sideband suppression in time-modulated linear arrays by the differential evolution algorithm*," IEEE Antennas Wireless Propagate. Lett. 1, pp. 173–175, 2002.

[25] Y.H. Pao, S. M. Phillips and D. J. Sobajic, "*Neural-net computing and intelligent control systems.*" *Int. J. Conr.* , vol. 56, no.2, pp.263-289, 1992.

[26] Patra, J. C., Pal, R. N., Chatterji, B. N., & Panda, G. "*Identification of nonlinear dynamic systems using functional link artificial neural networks*", IEEE Transactions in Systems Man and Cybernetics-Part B: Cybernetics, vol.29 no. 2,pp. 254–262, 1999.