Effective System Identification using Fused Network and DE Based Training Scheme

Saikat Singha Roy, Joyshri Das, Susovan Mondal

Abstract*— Adaptive direct modeling or system identification finds extensive applications in telecommunication, control system, instrumentation, power system engineering and geophysics. If the plants or systems are nonlinear, dynamic, single-input single-output (SISO), the identification task becomes more difficult. The dynamic system identification task is basically a model estimation process of capturing the dynamics of the system using the measured data. The Functional Link Artificial Neural Network (FLANN) is a single neuron single layer network first proposed by Pao. The structure of the FLANN is simple as it represents a flat net with no hidden layers. Therefore the computation and learning algorithm used in the architecture is straight forward.

In the present investigation the identification problem is performed on three standard benchmark nonlinear dynamic series-parallel models using Differential Evolution (DE) for training the weights of FLANN structure. The performance of the proposed FLANN-DE identification model is compared with FLANN-Genetic Algorithm and FLANN-Back Propagation method.

Index Terms*— Differential Evolution, FLANN, Genetic Algorithm, System Identification.

I. INTRODUCTION

Identification of a nonlinear dynamic plant is a major area in engineering today. System identification is widely used in a numerous 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 Winner-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 (1990) [9] have employed the multilayer perceptron (MLP) networks for effective identification and control of dynamic systems like truck-backer-upper problem [10].

Manuscript Received July, 2013

Saikat Singha Roy, Electronics & Communication Engg.,CITM Boinchi, Hooghly, NIT Durgapore, India.

Joyshri Das, Radio Physics, University of Calcutta, India.

Susovan Mondal, Electronics & Communication Engg.,SIEM, Siliguri, India.

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 genetic algorithm (GA) is one of the most popular population based stochastic search algorithm, inspired by Darwin's theory of survival and is being used as a very useful optimization technique in many fields.[11-13]. In recent past a promising variant of GA namely, the differential evolution (DE) has been proposed by Storn and Price [14], which is also a population, based stochastic optimization [15]. The DE has proven to be superior to other optimization algorithms [16-19], in terms of convergence speed and robustness. It has also been used for many diverse applications such as digital filter design [20], electromagnetics [21], power system [22-23] and designing of array antenna [24].

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 paper. 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 paper has been organized into 6 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. Section 4 provides evolutionary algorithm like GA and DE. Section 5 provides GA and DE based nonlinear system identification. Final section provides simulation and results.

II. NONLINEAR DYNAMIC SYSTEM IDENTIFICATION

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.

Where x(k) are input to both the plant and the model and y_p 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.

For mathematical evaluation, three types of single-input single-output (SISO) plants [9] are described in difference equations (1) to (3) are considered.

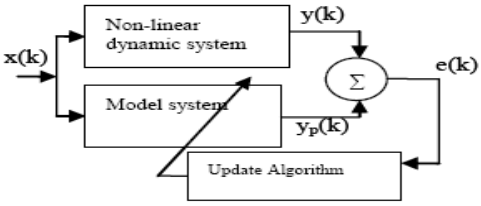


Fig. 1 Block diagram of nonlinear system identification

Type-I: (1)

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

Type-2:

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

Type-3 (3)

$$y(k) = f[y\ (k-1), y(k-2), \ldots y(k-n)] + \sum_{i=0}^{m-1} a_i x(k-i) + g[x(k), x(k-1), \ldots x(k-m+1)]$$

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.

III. DEVELOPMENT OF A NOVEL NONLINEAR IDENTIFICATION SCHEME

Figure 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. A structure of a FLANN is shown in Fig.2.

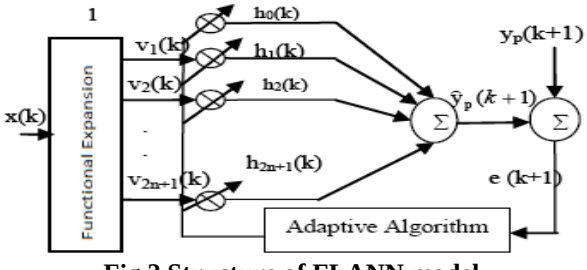


Fig.2 Structure of FLANN model

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:

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

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

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+1There are a total of *(2n + 1)* numbers of terms in the input vector. The weight vector related to the k^{th} input vector defined in (6) is given as:

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

Hence the estimated output of the identification model is computed as:

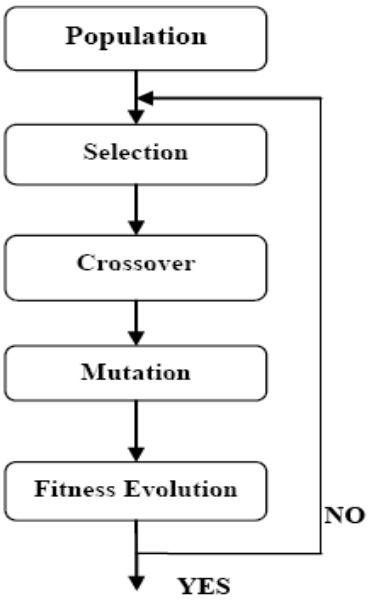
$$\hat{y}_p(k+1) = v(k).h(k) \quad (7)$$

IV. EVOLUTIONARY ALGORITHMS

A. Genetic Algorithm (GA)

Genetic algorithm is a part of evolutionary computing. GA was first introduced by John Holland and was later developed by Goldberg and De Jong [27]. The Algorithm begins with an initial set of solutions (represented by chromosomes) which are called as population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the older one. Solutions which are then selected to form new solutions (offspring) are selected according to their fitness - the more suitable they are the more chances that they will reproduce. This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied. Ideally, there are five phases in a GA program: initialization, selection, crossover, mutation and elimination.

Flow graph for Simple Genetic Algorithm



GA Operators:

Depending upon the application, many individual solutions are created randomly to form an initial population, covering the entire range of possible solutions (this is called as search space). Each point in the search space represents one possible solution marked by its value (fitness). The crossover and mutation are the most important parts of the genetic algorithm which helps the algorithm to be trapped in local minima. The performance is influenced mainly by these two operators.

Crossover:

In crossover recombination between two parent chromosomes is done to produce offspring that contain some parts of both parents "genetic material". A probability term, P_c, is set to determine the operation rate. Generally the probability of crossover is high. This is a determining factor that distinguishes the GA from all other algorithms.

Mutation:

Mutation operation introduces variations into the chromosomes. This is a process of making a good chromosome bad and vice versa in terms of its fitness. This variation can be global or local. The operation occurs occasionally (usually with small probability P_m) but it randomly alters the value of a particular string position. Each bit of a bit string is replaced by a randomly generated bit if a probability test is passed. Within a specific probability, certain digits are altered from either 0 to 1 or 1 to 0 in binary encoding.

Selection:

Finally the fitness (cost function) of original parents, offspring (children after crossover of selected parents) and mutated offspring are calculated. Again the best chromosomes are selected from the entire pools which are then treated as the parents of the next generation. The entire process continues till the global optimum is reached.

B. Differential Evolution:

Differential Evolution (DE) is a global optimization algorithm. It is used frequently used in the field of numerical optimization problem because it adapts an encoding scheme with real valued number, instead of using binary encoding as was the case in GA. In DE, initially some vectors which are possible solution within a D-dimensional search space are randomly created much like GA, then evolved over a time, to explore the entire search area, thereby locating the minimum of objective function. The initial population is denoted by NP which is represented as $\mathbf{X_{i,j}(G)}$, (i=0, 1, 2… (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.

STEPS IN DE:

Initialization:

The upper and lower bound of each parameter is to be specified before population initialization. Once initialization bounds have been specified, a random number generator assigns each parameter of every vector a value from within the prescribed range. For an example (G=0) the value of a j^{th} parameter of an i^{th} vector is

$$x_{j,i}(0) = rand_j(0,1).(p_{j,U} - p_{j,L}) + b_{j,L} \quad (8)$$

Where p_U and p_L are upper and lower bound, for range notification.

Mutation:

After initialization DE mutates unlike GA where first crossover is done after initialization In mutation the difference of two randomly selected vectors are multiplied by a constant factor and then added with a randomly selected vector from the population i.e. in DE a differential mutant operator is used. Equation (9) shows how to combine three different, randomly chosen vectors to create a mutant vector; $V_{i,j}(G+1)$:

$$v_{i,j}(G+1) = x_{r1,j}(G) + F.(x_{r2,j}(G) - x_{r3,j}(G)) \quad (9)$$

This type of mutation is called as de/rand/1. There are certain other formats of creating the mutant vector. DE mutation strategies like de/rand/2, de/best/1, de/best/2 and de/current to best/2 which are as described follows:

$$V_{i,j}(G+1) = X_{r5,j}(G) + F.(X_{r1,j}(G) + X_{r2,j}(G) - X_{r3,j}(G) - X_{r4,j}(G)) \quad (10)$$

$$V_{i,j}(G+1) = X_{best,j}(G) + F.(X_{r2,j}(G) - X_{r3,j}(G)) \quad (11)$$

$$V_{i,j}(G+1) = X_{best,j}(G) + F.(X_{r1,j}(G) + X_{r2,j}(G) - X_{r3,j}(G) - X_{r4,j}(G)) \quad (12)$$

$$V_{i,j}(G+1) = X_{i,j}(G) + \lambda.(X_{best,j}(G) - X_{i,j}(G)) + F.(X_{r1,j}(G) - X_{r2,j}(G)) \quad (13)$$

The scale factor, *F* is a positive real number which controls the rate at which the population evolves. While there is no upper limit on *F*, effective values are seldom greater than 1. In our simulation we used de/best/1 mutation strategy.

Crossover:

Now trail vector $U_{i,j}(G+1)$ is created using crossover operator:

$$u_{i,j}(G+1) = \begin{cases} v_{i,j}(G+1) \text{ if rand}(0,1) \le \text{CR or } j = j_{rand} \\ x_{i,j}(G) \text{ if rand}(0,1) > \text{CR or } j \ne j_{rand} \end{cases} \quad (14)$$

The crossover probability, C_r [0,1], controls the fraction of parameter values that are copied from the mutant. In crossover if the value of the randomly generated number is less than C_r then the trial parameter is inherited from the mutant, $V_{i,j}$, otherwise it is copied from $x_{i,j}$.

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.

$$x_{i,j}(G+1) = \begin{cases} u_{i,j}(G) \text{ if } f(u_{i,j}) \le f(x_{i,j}) \\ x_{i,j}(G) \text{ otherwise} \end{cases} \quad (15)$$

V.GA AND DE BASED NONLINEAR SYSTEM IDENTIFICATION

A. GA based Nonlinear System Identification:

System Identification on a nonlinear dynamic system using FLANN as structure and GA as update algorithm is carried out according to these following stages:

Step 1: Initialization of Parameter.

N= Initial population.

$\mathbf{P_m}$= Mutation probability.

S= Selection rate.

Step 2: ***n*** uniformly distributed random signals over the interval [-1, 1] are generated and applied to the actual nonlinear dynamic system and to the adaptive model simultaneously. This serves as the input to both the system and the adaptive model. In the present investigation a series parallel identification model is used.
Step 3: The plant's output serves as the desired output. The estimated output is obtained from adaptive model by using equation (7).
Step 4: Each of the desired output is compared with the corresponding estimated output and thus ***n*** numbers of errors produced.
Step 5: For each i^{th} weight vector the mean square error (MSE) is calculated by using the following fitness function:

$$MSE(i) = \sum_{i=1}^{n} e^2(n,i)/n \qquad (16)$$

Step 6: Random population of ***n*** chromosomes (suitable solutions for the problem) is then generated.
Step 7: The fitness of each chromosome in the population is evaluated.
Step 8: New population is then created by repeating following steps until the new population is complete.
Step 9: Two parent chromosomes are selected from the initial population according to their fitness (the better fitness, the bigger chance to be selected).
Step 10: With a crossover probability, the parents are crossed over to form new offspring (children). If no crossover is performed, offspring is the exact replica of the parents.
Step 11: With a mutation probability new offspring at each locus (position in chromosome) are mutated. New offspring are then placed in the new population.
Step 12: After each iteration minimum of MSE (MMSE) is calculated which shows the learning behavior of the adaptive model.
Step 13: Use newly generated population for a further run of the algorithm. If the end condition is satisfied then stop the process and return the best solution in current population.

B. DE based Nonlinear System Identification:

Step 1: Initialize the parameter used in DE:

Np = Select the total number of population

F = Select the scale factor which controls the rate at which the population evolves.

CR = Initialize it with some constant value, which selects whether the new population is copied from the trial vector or from the target vector.

D = number of parameters of the FLANN model is to be optimized.

G = number of generations.

N = number of input samples.

Step 2: ***n*** uniformly distributed random signals over the interval [-1, 1] are again generated and are applied to both the actual nonlinear dynamic system and to the adaptive model simultaneously as we did in GA also. Here again the same series parallel identification scheme is used.
Step 3: The plant's output serves as the desired output and estimated output is obtained from adaptive model using equation (7).
Step 4: Each of the desired output is compared with the corresponding estimated output and n numbers of errors are produced.
Step 5: The mean square error (MSE) is calculated for each i^{th} weight vector by using fitness function described in equation (16).
Step 6: Three random vectors are selected from the initial population and a mutant vector is obtained by using equation (11).
Step 7: Then it is checked whether the vector elements are within the search range or not. If it is not within the search range then bring it into specified search range.
Step 8: Generate a random number and compare this with Crossover Ratio (CR), initially selected. Then using equation (14) obtains the trail vector.
Step 9: The estimated output is obtained from the adaptive model by using this newly generated trail vectors. ***n*** numbers of errors are obtained by comparing the estimated output with the desired output. The MSE is then calculated from equation (16).
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.
Step 11: After each iteration minimum of MSE (MMSE) is calculated which shows the learning behavior of the adaptive model.

Step 12: Using the new generation the steps from 6 to 11 are repeated. If the end condition is satisfied then stop the process and return the best solution in the current population.

VI. SIMULATION AND RESULTS

For nonlinear dynamic plants described by difference equation in (1), (2) & (3), extensive simulation studies have been carried out. In this investigation a series-parallel model is used along with DE for training the weights of FLANN structure. The performance of the proposed FLANN-DE identification model is then compared with FLANN-GA and FLANN-BP methods. In FLANN-BP model an uniformly distributed random signal over the interval [-1, 1] is used as input, and **50,000** iterations are carried out for training.
The testing is analyzed by parallel scheme. The input to the identified model is given as

$$x(k) = \begin{cases} \sin(2\pi k/250) & \text{For } k \le 250 \\ 0.8\sin(2\pi k/250) + 0.2\sin(2\pi k/25) & \text{For } k > 250 \end{cases} \qquad (17)$$

Normalized mean square error (NMSE) [28] is used, for comparison of two models.

$$\text{NMSE} = \frac{1}{\sigma^2 s}\sum_{k=1}^{s}[y_p(k) - \hat{y}_p(k)]^2 \qquad (18)$$

Where σ^2 is the variance of desired signal and **s** is the testing sample.

A. Identification of SISO dynamic systems

Example 1:
In this example, the plant to be identified is of model-1 type and is represented by the difference equation [9] as follows:

$$y_p(k+1) = 0.3y(k) + 0.6y(k-1) + g[x(k)] \qquad (19)$$

Where unknown nonlinear function g (.) are given by

NL-1:

$$g_1(x) = 0.6\sin(\pi x) + 0.3\sin(3\pi x) + 0.1\sin(5\pi x) \quad (20)$$

NL-2:

$$g_2(x) = \frac{4.0x^3 - 1.2x^2 - 3.0x + 1.2}{0.4x^5 + 0.8x^4 - 1.2x^3 + 0.2x^2 - 3.0} \quad (21)$$

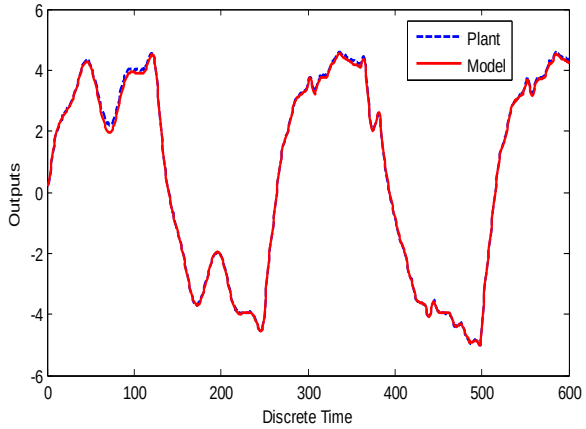
NL-3:

$$g_3(x) = 0.5\sin^3(\pi x) - \frac{2.0}{x^3 + 2.0} - 0.1\cos(4\pi x) + 1.125 \quad (22)$$

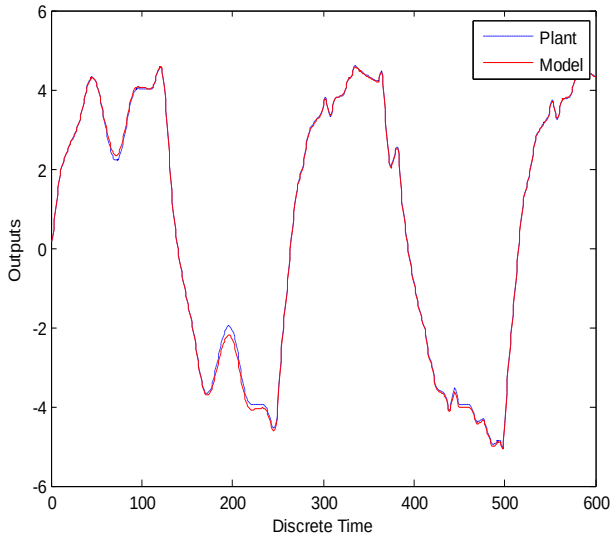
T**o** identify the plant a series-parallel model is used which is described by following difference equation

$$\hat{y}_p(k+1) = 0.3y(k) + 0.6y(k-1) + N[x(k)] \quad (18)$$

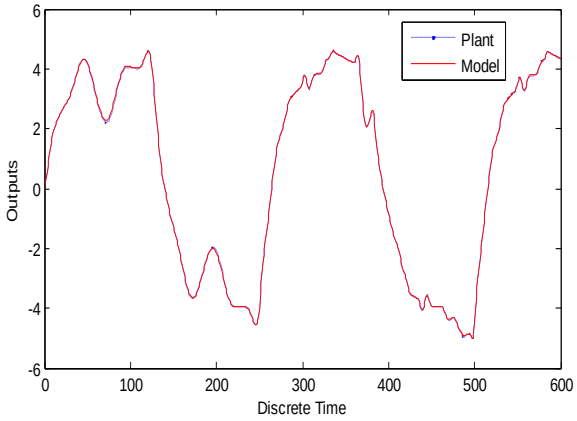
N[x(k)] is FLANN-BP, FLANN-GA or FLANN-DE model. The FLANN input is expanded to ten terms for NL-1 (20) and eleven terms for NL-2 (21) and NL-3 (22) respectively. In FLANN-GA model training is done for 500 iterations. The mutation probability (P_m) and selection rate (S) are chosen to be 0.05 and 0.5 respectively. In DE, CR=0.5, F=0.5, number of generations = 500 and DE/best/1 scheme is used for mutation. Both the convergence parameter μ and the momentum factor η are chosen to be 0.1 for FLANN-BP. The results of identification of (19) with nonlinear function (20), (21) and (22) are shown in Fig.3, Fig.4 and Fig.5 respectively.



(a) FLANN-BP (expansion 10)

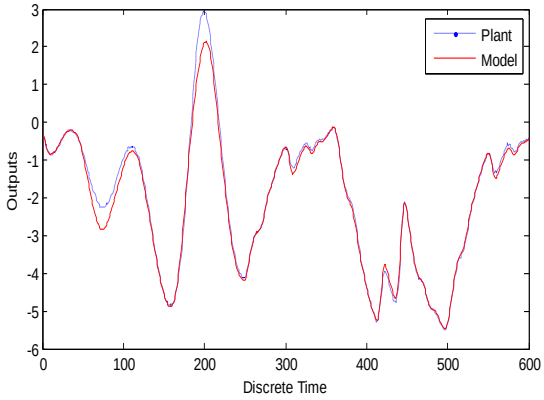


(a) FLANN-GA (expansion 10)

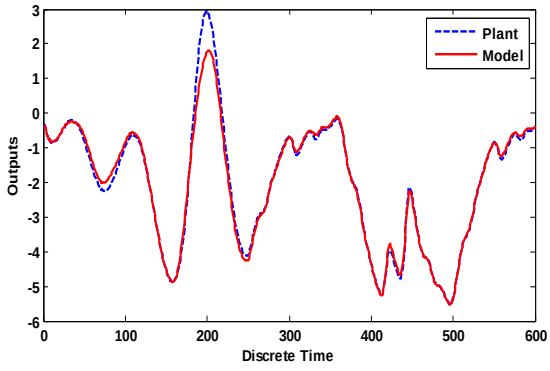


(a) FLANN-DE (expansion 10)

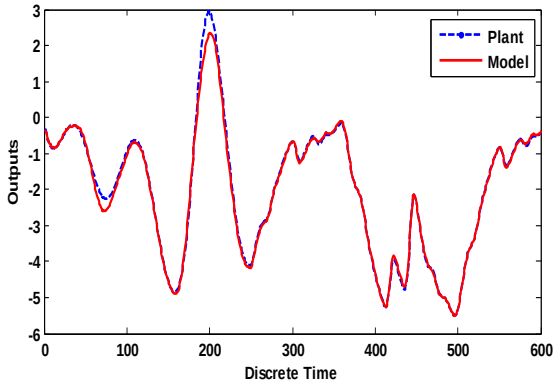
Fig.3 Comparison of output response of example-1 using nonlinearity defined in (20)



(a) FLANN-BP (expansion 11)

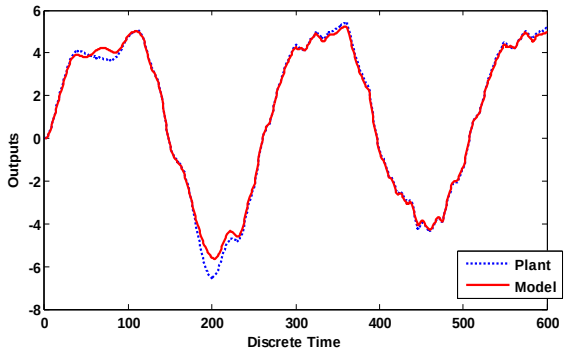


(b) FLANN-GA (expansion 11)

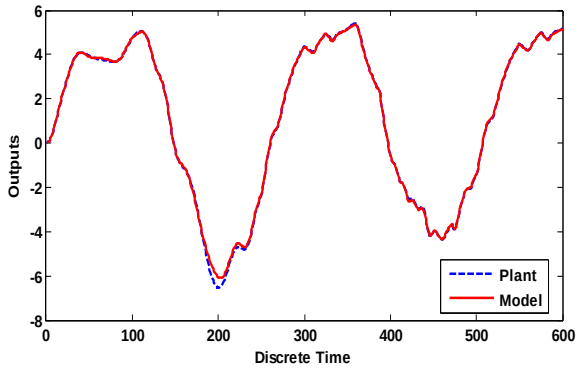


(c) FLANN-DE (expansion 11)

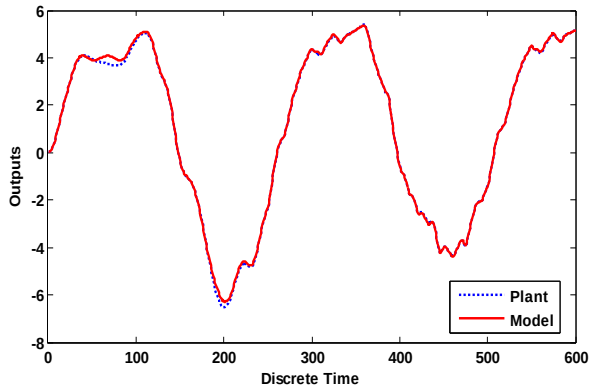
Fig.4 Comparison of output response of example-1 using nonlinearity defined in (21)



(a) FLANN-BP (expansion 11)



(b) FLANN-GA (expansion 11)



(c) FLANN-DE (expansion 11)

Fig.5 Comparison of output response of example-1 using nonlinearity defined in (22)

Example 2:

In this example the plant to be identified is of model-2 type [9] and is represented by the difference equation:

$$y_p(k+1) = f[y_p(k), y_p(k-1)] + x(k) \tag{23}$$

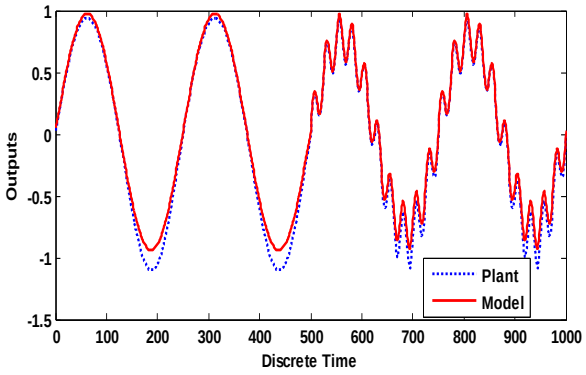
The unknown nonlinear function f(.) is given by

$$f(y(x), y(k-1)) = \frac{y(k)y(k-1)(y(k)-2.5)(y(k)-1.0)}{1.0 + y^2(k) + y^2(k-1)} \tag{24}$$

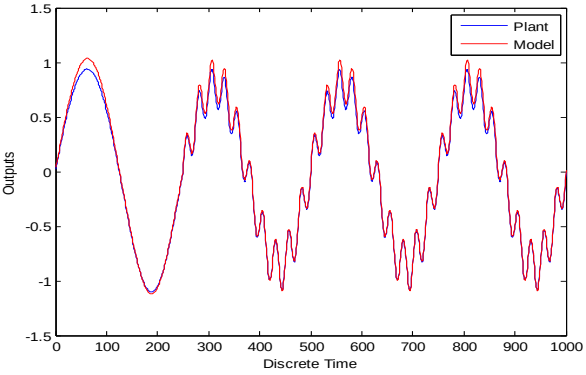
To identify the model a series-parallel model is used

$$\hat{y}_p(k+1) = N(y_p(k)y_p(k-1) + x(k)) \tag{25}$$

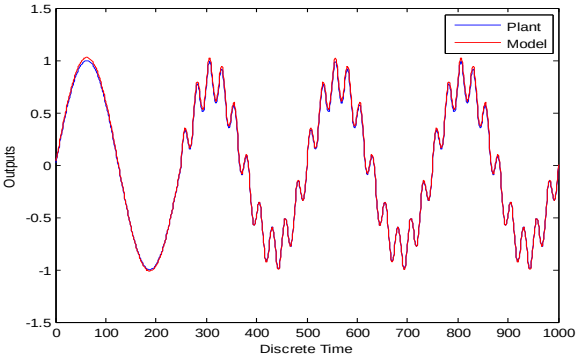
Here value of μ and η are set as 0.05 and 0.1 respectively for FLANN-BP. All the parameters in FLANN-GA and FLANN-DE models are same as used in example-1. The FLANN inputs are expanded to 9 terms using trigonometric expansion. The response of the example 2 is shown in Fig.6. From these results it is evident that FLANN-DE method gives accurate identification of the present system. From these result it is observed that FLANN-DE identification model gives better result than FLANN-GA and FLANN-BP model. From table-I it is also evident that FLANN-DE approach gives lower NMSE than the other identification model.



(a) FLANN-BP (expansion 9)



(b) FLANN-GA (expansion 9)



(c) FLANN-DE (expansion 9)

Fig.6 Comparison of output response of example-2

Example 3:

In this case the plant is of model-3 type [9] and represent by the difference equation:

$$y_p(k+1) = f[y_p(k)] + g[x(k)] \tag{26}$$

Where f( ) and g() is nonlinear function and is given by

$$f(y) = \frac{y(y+3)}{1+y^2} \quad \text{and} \tag{27}$$

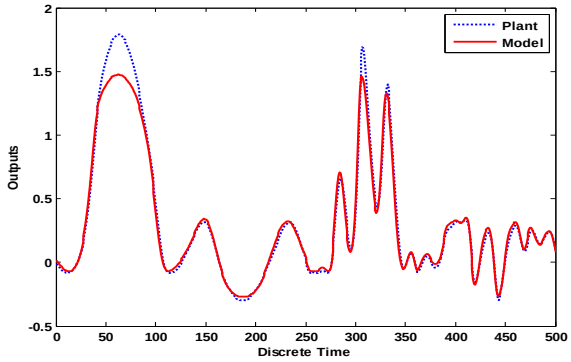
$$g(x) = x(x-0.8)(x-0.5) \tag{28}$$

The identification model of equation (22) is as follows:

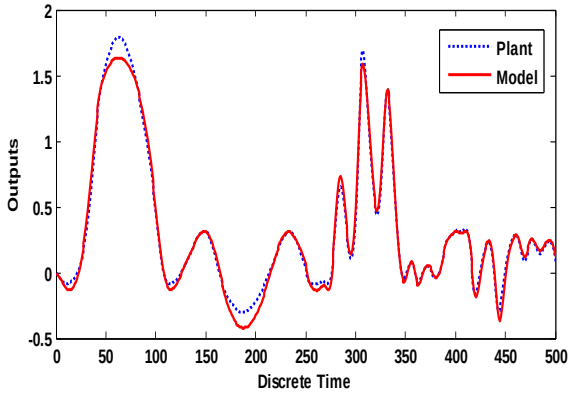
$$y_p(k+1) = N_1[y_p(k)] + N_2[x(k)] \tag{29}$$

Where N_1 and N_2 is FLANN-BP, FLANN-GA or FLANN-DE model.

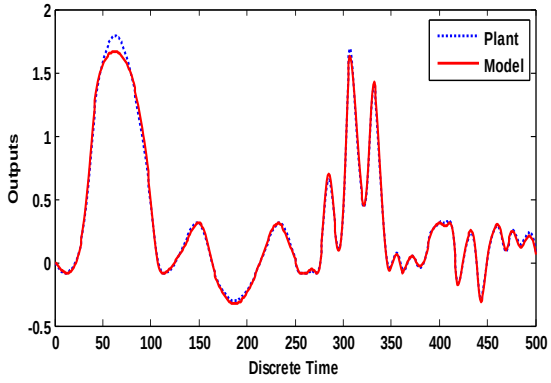
N_1 and N_2 contain 7 and 5 expansion respectively for FLANN-BP, FLANN-GA and FLANN-DE models. The parameters for GA and DE are retained same as was in example-1. For BP, convergence and momentum parameter is chosen as 0.1. The result of identification is shown in Fig.7 where it is observed that FLANN-DE model shows better result than other two identification models.



(a) FLANN-BP (expansion 14)



(b) FLANN-GA (expansion 14)



(c) FLANN-DE (expansion 14)

Fig.7 Comparison of output response of example-3

This is the comparison table of three identification model for three nonlinear dynamic systems with respect no NMSE defined in (18).

Table-I: Comparison of NMSE of different examples

Example	Expa-nsions	NMSE in dB		
		FLANN-BP	*FLANN-GA*	*FLANN-DE*
Ex-1with (20)	10	-31.8807	-35.8218	-41.9738
Ex-1 with (21)	11	-19.5387	-18.913	-23.2431
Ex-1 with (22)	11	-24.967	-31.6241	-33.5503
Ex-2	9	-19.1246	-23.8696	-31.567
Ex-3	14	-17.4129	-21.1247	-24.6214

Further, we have also investigated on the convergence pattern of various variants of DEs. The various variants of Des are described in equation (10), (11), (12) and (13). For example-1 the convergence of various variants of DEs is plotted in figure-8. Similarly, in figure-9 and figure-10 the convergences of various DEs are plotted for example-2 and example-3 respectively.

From the convergence graphs (Fig.8, Fig.9 and Fig.10) we conclude that for example-1 and example-2 DE/best/1 performs best, but for example-3 DE/current to best/2 shows better results.

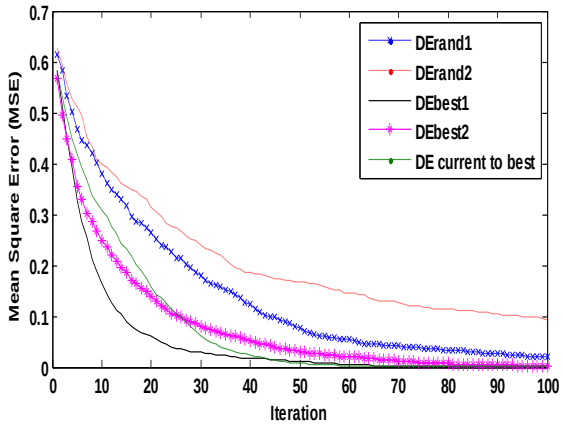


Fig. 8: Comparison of different DE strategies for example-1

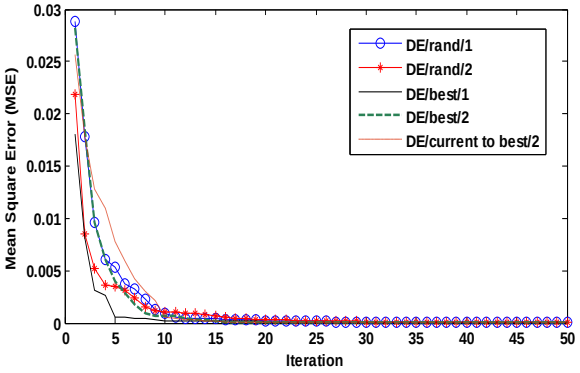


Fig. 9: Comparison of different DE strategies for example-2

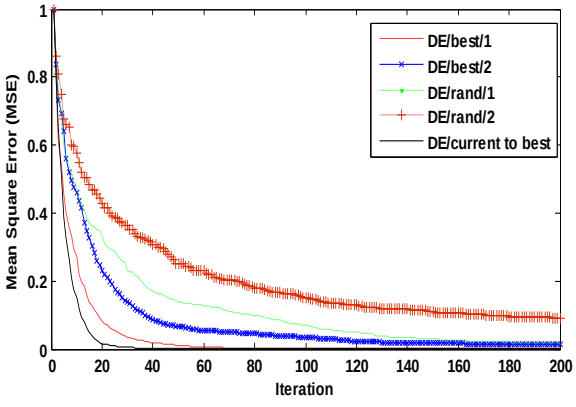


Fig. 10: Comparison of different DE strategies for example-3

VII. CONCLUSION

In the present investigation, the identification problem is performed on three standard benchmark nonlinear dynamic series-parallel models. From the simulation study it is evident that FLANN-DE provides accurate identification for nonlinear dynamic models. When compared to FLANN-GA and FLANN-BP the FLANN-DE identification model gives better results. From table-I it is also evident that FLANN-DE approach gives lower NMSE than the other two identification models.

From the convergence graphs (Fig.8, Fig.9 and Fig.10) we can conclude that for example-1 and example-2 DE/best/1 performs well but for example-3 DE/current to best/2 shows better result.

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.
5. M. Schetzmen, The Voltera and Winner Theories on Nonlinear Systems, Wiley, New York, 1980.
6. E. Hernandez, Y. Arkun, Control of nonlinear systems using polynomial ARMA models, AICHE J. 39 (3) (1993) 446–460.
7. 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.
8. 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.
9. Nguyen, D. H., & Widrow, B. "Neural networks for self-learning control system." International Journal of Control, vol. 54, no. 6, pp. 1439–1451, 1991.
10. M. Srinivas, L. M. Patnaik, "Genetic Algorithm: A Survey'',' IEEE computer, vol. 27, no. 6, pp. 17-26, 1994.
11. D. B. Fogel, "An Introduction to Simulated Evolutionary Optimization", IEEE Trans. On SMC., vol. 24, no. 1, pp.
12. W. Atmar, "Notes on the Simulation of Evolution", IEEE Trans. On SMC., vol. 24, no. 1, pp. 130-147, 1994.
13. 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.
14. 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.
15. 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.
16. 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.
17. 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.
18. O. Hrstka and A. Ku˘cerová, "Improvement of real coded genetic algorithm based on differential operators preventing premature convergence," Advance in Engineering Software 35, pp. 237–246, 2004.
19. 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.
20. 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).
21. Lakshminarasimman, L., Subramanian, S.: Hydrothermal optimal power flow using modified hybrid differential evolution. Caledonian J. Engg. 3(1), 8–14 (2007a)
22. 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)
23. 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.
24. 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.
25. 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.
26. D.E Goldbareg, Genetic algorithms in search, optimization, and Machine learning. Reading, M A: Addison-Wesley, 1989.
27. Gershenfeld, N. A., & Weigend, A. S., "The future of time series: Learning and understanding. Time series prediction: Forecasting the future and past." Reading, MA: Addison-Wesley, 1993.

AUTHORS PROFILE

SAIKAT SINGHA ROY did his Bachelors in Physics in 2006. Subsequently, he did his M.Sc in Electronic Science in 2008 , from Vidyasagar University and M.Tech in Communication Engg.in 2011, from KIIT University. He is currently the Head of the Department of Electronics & Communication Engg., CITM Boinchi, Hooghly, India. His research interests include Soft Computing, Evolutionary Computation and Pattern Recognition.

JOYSHRI DAS did her Bachelors in Electronics & Communication Engg. in 2007. Subsequently, she did her M.Tech in Mechatronics in 2010, from BESU, Sibpur. She is currently a faculty of Electronics & Communication Engg., CITM Boinchi, Hooghly, India. Her research interests include Soft Computing, Digital Signal Processing and Pattern Recognition.

SUSOVAN MONDAL did his Bachelors in Electronics in 2006. Subsequently, he did his M.Sc in Electronic Science in 2008, from Vidyasagar University and M.Tech in Communication Engg.in 2011, from KIIT University. He is currently a faculty of Electronics & Communication Engg., of SIEM, Siliguri, India.