Neural Process Lett (2008) 27:285–296
http://dx.doi.org/10.1007/s11063-008-9077-x

archived in dspace@nitr

Neural Process Lett (2008) 27:285–296
DOI 10.1007/s11063-008-9077-x

Differential Evolution and Levenberg Marquardt Trained Neural Network Scheme for Nonlinear System Identification

Bidyadhar Subudhi · **Debashisha Jena**

Abstract This paper proposes a new nonlinear system identification scheme using differential evolution (DE), neural network and Levenberg Marquardt algorithm (LM). Here, DE and LM in a combined framework are used to train a neural network for achieving better convergence of neural network weight optimization. A number of examples including a practical case-study have been considered for implementation of different system identification methods namely, only NN, DE + NN and DE + LM + NN. After, a series of simulation studies of these methods on the different nonlinear systems it has been confirmed that the proposed DE and LM trained NN approach to nonlinear system identification has yielded better identification results in terms of time of convergence and less identification error.

Keywords Differential evolution · Evolutionary computation · Nonlinear system identification · Levenberg Marquardt

1 Introduction

There is a strong research in nonlinear system identification from the stand point of industry and real world applications. In the past, a number of nonlinear system identification techniques have been proposed such as Voltera series, Winner-Hammerstein model and polynomial identification [1] methods which involve computational complexities. Apart from these methods there is lot of research directed towards applying neural networks [2,3] for nonlinear system identification due to its function approximation capabilities. A number of theoretical and practical system identification problems have been solved using neural network approach with multi-layered perceptron (MLP) with back-propagation training [3]. From the previous neural network system identification approaches, it is observed that even neural network has been proved to be a successful technique for nonlinear system identification but there is a little concern about its convergence and problem of being trapped at local minima. The Wavelet

B. Subudhi (✉) · D. Jena
Department of Electrical Engineering, National Institute of Technology, Rourkela 769008, India
e-mail: bidyadhar@nitrkl.ac.in

networks techniques [2] were also applied to system identification of nonlinear systems in which adaptive techniques such as back propagation algorithm found to provide better accuracy compared to non-adaptive ones such as Voltera series, Winner-Hammerstein modeling and polynomial methods.

In the last three decades Evolutionary algorithms (EAs), such as genetic algorithm (GA) and evolutionary strategies (ES) have become very popular as function optimizers, because these are easy to implement, and exhibit fair performance for a wide range of functions. However, continued development in the research community has established that pure GAs are often not good enough for fine tuning in complex search spaces.

A variant of evolutionary computing technique namely the Differential evolution (DE) [4, 5] is a population based stochastic optimization method similar to GA that finds an increasing interest in the recent year as an optimization technique due to its achievement of a global minimum. Differential evolution is an effective, efficient and robust optimization method capable of handling nonlinear and multimodal objective functions. The beauty of DE is its simple and compact structure which uses a stochastic direct search approach and utilizes common concepts of EAs. Furthermore, DE uses few easily chosen parameters and provides excellent results for a wide set of benchmark and real-world problems. Experimental results have shown that DE has good convergence properties and outperforms other well known EAs [5]. Therefore, there is scope of using DE approach to neural network weight optimization.

It may be noted that by using the DE approach alone may yield slow convergence speed although it provides the global minimum as compared to classical optimization methods such as the gradient descent and LM method. Therefore, the authors propose a DE + LM + NN approach in view of achieving global minimum with good convergence speed. In this work, a differential evolution method combined with LM has been applied as a global optimization method for training a feed-forward neural network. In the proposed scheme, the DE is used to train the neural network that is chosen as a suitable candidate for nonlinear system identification. After observing the trends of training towards minimum through DE, the network is then trained by LM. The role of the DE here is to approach towards global minimum point and then LM is used to move forward achieving fast convergence. The nonlinear systems as considered in [6,7] have been chosen in this work for demonstrating the efficacy of the proposed system identification approach.

The contribution of the paper is as follows. We proposed here an improved neural network based nonlinear system identification scheme where the training of the neural network (NN) employed for the identification has been made faster and accurate. This improved training algorithm was achieved by virtue of two important benefits of hybrid use of two different optimization schemes namely a stochastic evolutionary algorithm i.e. Differential Evolution algorithm which provides a global search whilst the convergence of the proposed hybrid (DE + LM + NN) training has been accelerated by the gradient based optimization technique i.e. Levenberg Maquardt (LM) algorithm. We clearly demonstrated in the paper that our proposed approach has been found to be successful in providing the best identification performance amongst the other varieties of system identification schemes such as NN trained by LM, and NN trained by DE.

2 Differential Evolution

In a population of potential solutions to an optimization problem within an n-dimensional search space, a fixed number of vectors are randomly initialized, then evolved over time to explore the search space and to locate the minima of the objective function.

In DE, individuals are represented as real-valued vectors. For each generation of the evolution process, each individual (target individual) of the population competes against a new individual (trial individual) for survival to the next generation. Only the fitter of the two survives. The trial individual is created by recombining the target individual with another individual created by mutation (mutant individual). Mutation is performed on the best individual found so far in the evolution process. For each target vector $x_{i,G}$ a mutant vector is produced using the following formula

$$v_{i,G+1} = x_{r1,G} + F(x_{r2,G} - x_{r3,G}) \tag{1}$$

where $i, r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$ are randomly chosen and must be different from each other. In Eq. 1, F is the mutation factor. Recombination creates an offspring (trial individual) by selecting parameters from either the target individual or the mutant individual. There are two methods of recombination in DE, namely, binomial recombination and exponential recombination. In binomial recombination, a series of binomial experiments are conducted to determine which parent contributes which parameter to the offspring. Each experiment is mediated by a crossover constant, *CR*, $(0 \leq CR \geq 1)$. Starting at a randomly selected parameter, the source of each parameter is determined by comparing *CR* to a uniformly distributed random number from the interval [0, 1). If the random number is greater than *CR*, the offspring gets its parameter from the target individual; otherwise, the parameter comes from the mutant individual. In exponential recombination, a single contiguous block of parameters of random size and location is copied from the mutant individual to a copy of the target individual to produce an offspring. A vector of solutions are selected randomly from the mutant individuals when $rand_j$ ($rand_j \in [0, 1]$, is a random number) is less than *CR*.

In crossover, the parent vector is mixed with mutated vector to produce a trial vector $t_{ji,t+1}$

$$t_{ji,t+1} = \begin{cases} v_{ji,t+1} & \text{if } (rand_j \leq CR) \\ w_{ji,t+1} & \text{if } (rand_j > CR) \end{cases} \tag{2}$$

$j = 1, 2, \ldots, D$, where D is the number of parameters to be optimized.

At each generation, new vectors are generated by the combination of vectors randomly chosen from the current population (mutation). The upcoming vectors are then mixed with a predetermined target vector. This operation is called recombination and produces the trial vector. Finally, the trial vector is accepted for the next generation iff it yields a reduction in the value of the objective function. This last operator is referred to as a selection.

Figure 1 shows a two dimensional objective function that illustrates the different vectors, x_i which are important in differential evolution. It shows the process of generating trial vector for the scheme explained in Eq. 2.

3 Proposed DE–LM–NN System Identification Scheme Training Algorithm

In the sequel, we describe how a DE is applied for training neural network in the frame work of system identification (see Algorithm-1). DE can be applied to global searches within the weight space of a typical feed-forward neural network. Output of a feed-forward neural network is a function of synaptic weights **w** and input values **x**, i.e. $\mathbf{y} = f(\mathbf{x}, \mathbf{w})$. The role of LM in the proposed algorithm has been described in Sect. 1. In the training processes, both the input vector **x** and the output vector **y** are known and the synaptic weights in **w** are adapted to obtain appropriate functional mappings from the input **x** to the output **y**. Generally, the adaptation can be carried out by minimizing the network error function **E**

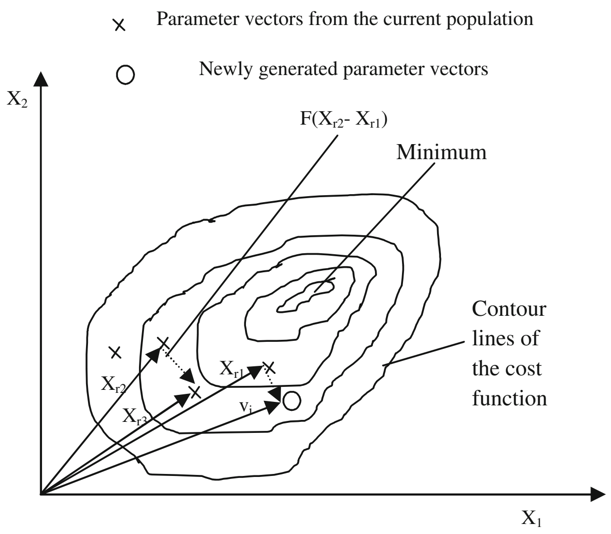
× Parameter vectors from the current population
○ Newly generated parameter vectors
X2
F(Xr2- Xr1)
Minimum
Contour lines of the cost function
Xr2
Xr3
Xr1
vi
X1

Fig. 1 Two dimensional objective function

which is of the form $\mathbf{E}(\mathbf{y}, f(\mathbf{x}, \mathbf{w}))$. In this work we have taken $\mathbf{E}$ as mean squared error i.e. $\mathbf{E} = \frac{1}{N}\sum_{k=1}^{N}\left[\mathbf{y} - f(\mathbf{x}, \mathbf{w})\right]^2$, where N is the number of data considered.

The optimization goal is to minimize the objective function $\mathbf{E}$ by optimizing the values of the network weights $\mathbf{w}$, where $\mathbf{w} = (w_1, \ldots, w_d)$.

Algorithm-1 *(DE+LM+NN Identification Algorithm)*

Step 1 Initialize population ***pop***: Create a population from randomly chosen object vectors with dimension *NP*.

$$\mathbf{P}_G = (\mathbf{w}_{1,G}, \ldots, \mathbf{w}_{NP,G})^T, \quad G = 1, \ldots, G_{\max}$$

$$\mathbf{w}_{i,G} = (w_{1,i,G}, \ldots, w_{D,i,G}), \quad i = 1, \ldots, NP$$

where D is the number of weights in the weight vector and in $\mathbf{w}_{i,G}$, i is index to the population and G is the generation to which the population belongs.

Step 2 Evaluate all the candidate solution inside pop for a specified number of iterations.

Step 3 For each ith candidate in pop select the random variables $r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$

Step 4 Apply mutation operator to each candidate in population to yield a mutant vector i.e.

$$v_{j,i,G+1} = w_{j,r1,G} + F(w_{j,r2,G} - w_{j,r3,G}), \quad \text{for } j = 1, \ldots, D$$
$$(i \neq r_1 \neq r_2 \neq r_3) \in \{1, \ldots, NP\} \qquad \text{and } F \in (0, 1+]$$

Step 5 Apply crossover i.e. each vector in the current population is recombined with a mutant vector to produce trial vector.

$$t_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \text{if } rand_j[0,1) \leq CR \\ w_{j,i,j} & \text{otherwise} \end{cases}$$

where $CR \in [0, 1]$.

Step 6 Apply selection i.e. between the trial vector and target vector. If the trial vector has an equal or lower objective function value than that of its target vector, it replaces target vector in the next generation; otherwise, the target retains its place in the population for at least one more generation

$$\mathbf{w}_{i,G+1} = \begin{cases} t_{i,G+1} & \text{if } \mathbf{E}(\mathbf{y}, f(\mathbf{x}, \mathbf{w}_{i,G+1})) \le \mathbf{E}(\mathbf{y}, f(\mathbf{x}, \mathbf{w}_{i,G})) \\ \mathbf{w}_{i,G} & \text{otherwise} \end{cases}$$

Step 7 Initialize the weight matrix of Levenberg–Marquardt algorithm taking the values of weights obtained after the fixed number of iterations. Find out the value of **E**.

Step 8 Compute the Jacobian matrix $\mathbf{J}(w)$.

Step 9 Find Δw using the following equation

$$\Delta w = \left[\mathbf{J}^T(w)J(w) + \mu I\right]^{-1} \mathbf{J}^T(w)\mathbf{E}$$

Step 10 Recompute **E** using $(w + \Delta w)$, if this new **E** is smaller than that computed in Step 7 then reduce μ and go to Step 1, where μ is the damping factor.

Step 11 The algorithm is assumed to have converged when the norm of the gradient i.e. $\|\nabla \mathrm{E}\| = \|\mathbf{J}^T(w)\mathbf{y} - f(\mathbf{x}, w)\|$ is less than some predetermined value, or when the sum of squares of errors has been reduced to some error goal.

4 Results and Discussion

We present here the performance achieved through using the proposed DE+LM+NN scheme to a number of bench mark problems as follows

4.1 Example-1

The nonlinear system [6] to be identified is expressed by

$$y_p(k+1) = \frac{y_p(k)[y_p(k-1)+2][y_p(k)+2.5]}{8.5 + [y_p(k)]^2 + [y_p(k-1)]^2} + u(k) \tag{3}$$

where $y_p(k)$ is the output of the system at the kth time step and $u(k)$ is the plant input which is uniformly bounded function of time. The plant is stable at $u(k) \in [-2\ 2]$. The identification model be in the form of

$$y_{pi}(k+1) = N(y_p(k), y_p(k-1)) + u(k) \tag{4}$$

where $N(y_p(k), y_p(k-1))$ is the nonlinear function of $y_p(k)$ and $y_p(k-1)$. The inputs to the neural network are $y_p(k)$ and $y_p(k-1)$. The output from neural network is $y_{pi}(k+1)$. The goal is to train the networks such that when an input $u(k)$ is presented to the network and to the nonlinear system, the network outputs $y_{pi}(k)$ and the actual nonlinear system output $y_p(k)$ will match as close as possible.

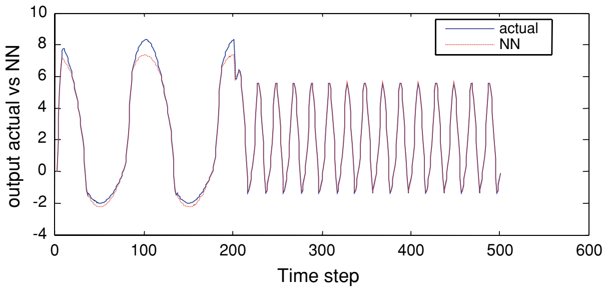

Fig. 2 Identified and actual models (NN identification)

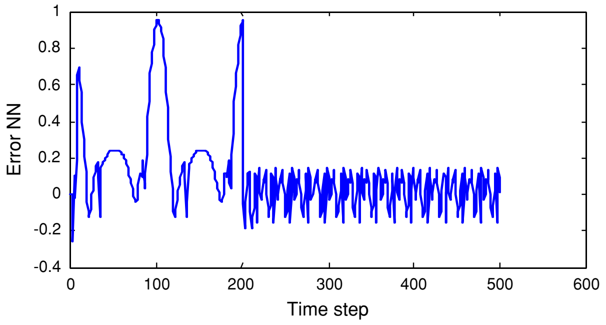

Fig. 3 Error in modeling (NN identification)

4.1.1 NN Identification (Figures 2, 3)

The neural network identifier structure consisted of eleven numbers of neurons in the hidden layer. After 1,000 epochs the training of the neural identifier has been stopped. After the training is over, its prediction capability has been tested for input.

$$u(k) = 2\cos(2\pi k/100) \quad k \leq 200 \text{ and}$$
$$u(k) = 1.2\sin(2\pi k/20) \quad 200 < k \leq 500$$

Figure 2 shows the system identification results obtained with using NN. The error is more at time steps 100 and 200. Figure 3 shows the identification error.

4.1.2 DE + NN Identification (Figures 4,5)

Figure 4 shows the identification performance of the system using NN and differential evolution. Here the network is trained by using differential evolution instead of classical ones such as gradient descent and Levenberg Marquardt algorithm. The results obtained with NN–DE indicate no significant improvement over the previously discussed existing ones. Here also eleven number of hidden layer neurons and thousands number of epochs were taken Fig. 5 shows the identification error in the case of NN + DE approach.

4.1.3 DE + NN + LM Identification (Figures 6,7)

Figure 6 gives the result of proposed DE + LM + NN scheme. In this case the network is trained by both DE and Levenberg Marquardt algorithm. Here eleven number of hidden

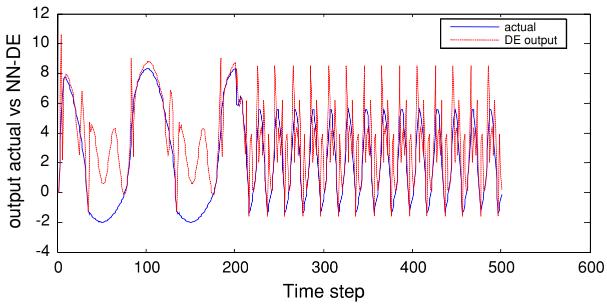

Fig. 4 Identified and actual models (NN–DE identification)

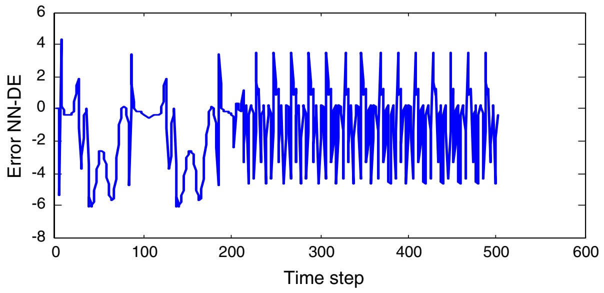

Fig. 5 Error in modeling (NN-DE identification)

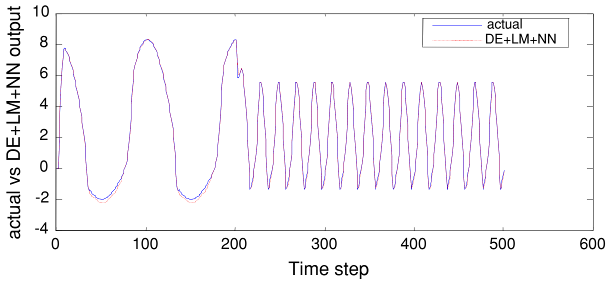

Fig. 6 Identified and actual models (DE + LM + NN identification)

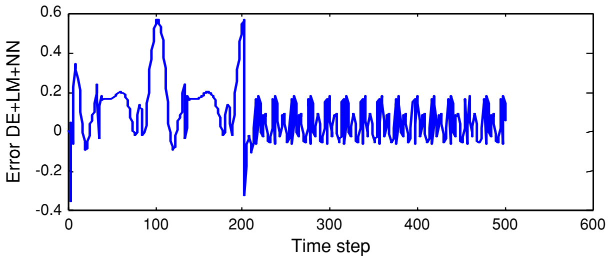

Fig. 7 Error in modeling (DE + LM + NN identification)

layer neurons are considered. This result (Fig. 7) clearly indicates accurate identification of nonlinear system is achieved i.e. the superior identification capability of the proposed scheme over the other methods NN and NN + DE.

Figure 7 shows its identification error curve for DE + LM + NN system identification. From Fig. 7 it is clear that identification error is smaller compared to error in NN and NN + DE.

4.2 Example 2

The plant [6] to be identified is governed by the difference equation

$$y_p(k+1) = 0.3y_p(k) + 0.6y_p(k-1) + f\,[u(k)] \tag{5}$$

where the unknown function has the form:

$$f(u) = 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u) \tag{6}$$

In order to identify the plant a series parallel model governed by the difference equation

$$\hat{y}_p(k+1) = 0.3y_p(k) + 0.6y_p(k-1) + N\,[u(k)] \tag{7}$$

was used.

4.2.1 NN Identification (Figures 8, 9)

The neural network identifier structure consisted of 11 numbers of neurons in the hidden layer. After 1,500 epochs, the training of the neural identifier has been stopped. After the training is over, its prediction capability has been tested for input given as

$$u(k) = \sin(2\pi k/250)$$

As shown in the Fig. 8, the learned network is predicting the nonlinear system outputs. Figure 9 shows the identification error. The figures clearly indicate the poor identification performance of the neural identifier.

4.2.2 NN-DE Identification (Figure 10)

Figure 10 shows the identification performance of the system using differential evolution. Here the network is trained by using differential evolution instead of classical ones such as

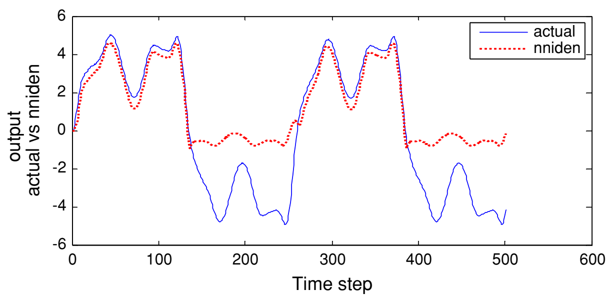

Fig. 8 Identified and actual models (NN identification)

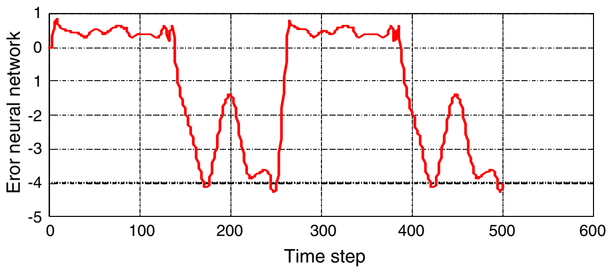

Fig. 9 Error in modeling (NN identification)

gradient descent and LM algorithm. The results obtained with only DE indicate no significant improvement over the previously discussed existing ones.

4.2.3 DE + LM + NN Identification (Figures 11, 12)

Figure 11 gives the result of proposed DE + LM + NN scheme. In this case the network is trained both by DE and LM algorithm.

This result clearly indicates the superior identification capability of the proposed scheme over the other two methods discussed i.e. NN and NN–DE approaches. Figure 12 shows its identification error curve for DE + LM + NN system identification.

4.3 Example 3

Box and Jenkins' gas furnace data are frequently used in performance evaluation of system identification methods [7]. The data can be obtained from the site. The example consists of 296 input–output samples recorded with a sampling period of 9 s. The gas combustion

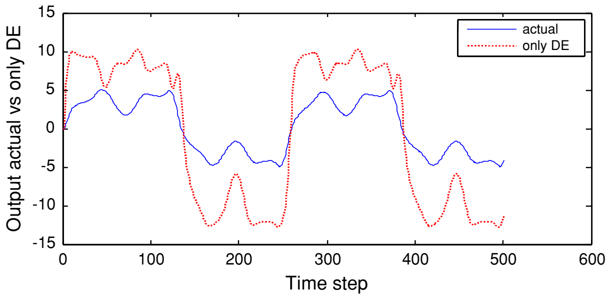

Fig. 10 Identified and actual models (DE identification)

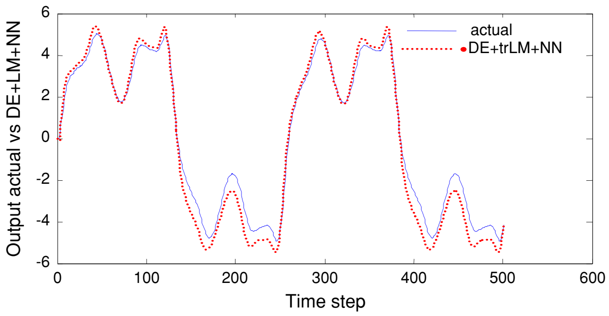

Fig. 11 Identified and actual models (DE + LM + NN identification)

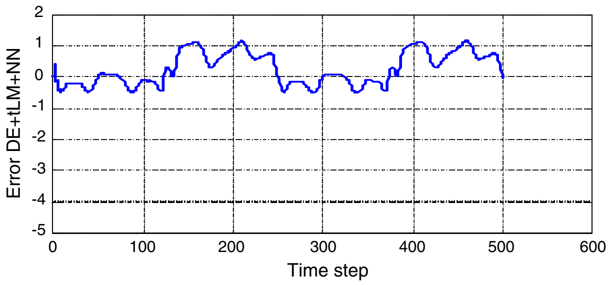

Fig. 12 Error in modeling (DE+LM+NN identification)

process has one variable, gas flow $u(k)$, and one output variable, the concentration of carbon dioxide (CO_2), $y(k)$. The instantaneous values of output $y(k)$ have been regarded as being influenced by ten variables $y(k-1)$, $y(k-2)$, $y(k-3)$, $y(k-4)$, $y(k-5)u(k-1)$, $u(k-2)$, $u(k-3)$, $u(k-4)$, $u(k-5)$. In the literature, the number of variables influencing the output varies from 2 to 10. In the proposed method, ten variables were chosen. Results shown gives a comparison of the identification methods such as neural networks trained with conventional methods and neural networks trained with DE and hybrid differential evolution methods. For all the methods eleven number of hidden layer neurons were taken and the results obtained after 1,000 epochs. The number of training data was taken as 100 for all the cases and rest 196 data were the test data.

4.3.1 NN Identification (Figures 13)

Figure 13 shows the graphs of the identified obtained with NN and the actual system. Here, the NN fails to identify the system dynamics at time step of 260 thus leads to a big identification error.

4.3.2 NN-DE Identification (Figure 14)

The NN–DE identified system dynamics and the actual system dynamics were plotted in Fig. 14, where from it is observed that no improvement in identification is observed with respect to the previous one i.e. the NN identifier.

4.3.3 DE+LM+NN Identification (Figures 15, 16)

Figure 15 shows the identification performance of the proposed new DE+LM+NN approach.

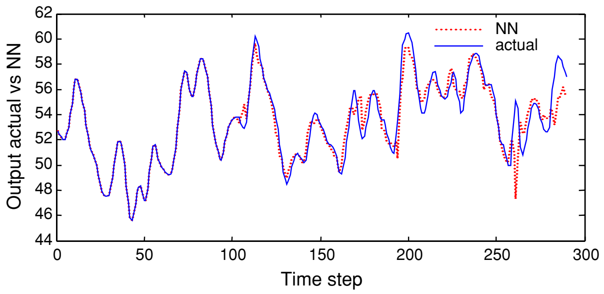

Fig. 13 Identified and actual models (NN identification)

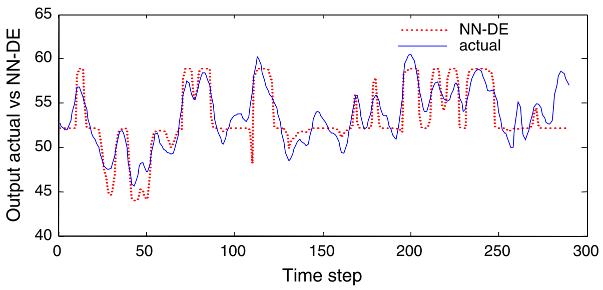

Fig. 14 Identified and actual models (NN–DE identification)

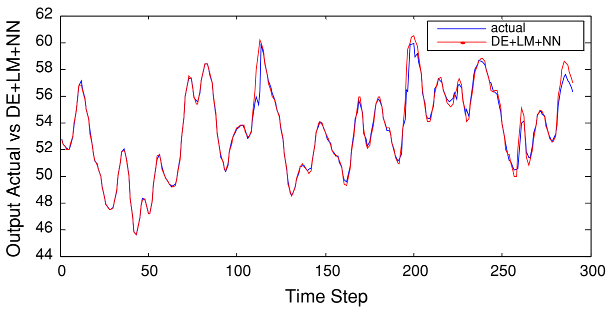

Fig. 15 Identified and actual models (DE + LM + NN identification)

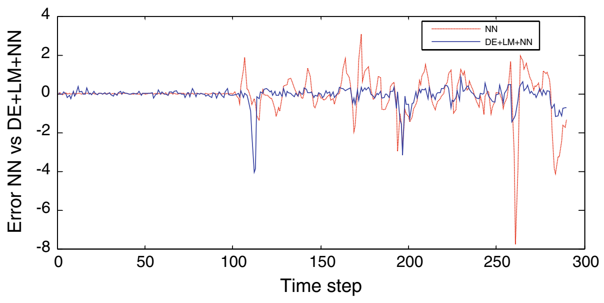

Fig. 16 Comparison of error in modeling [NN versus (DE + LM + NN identification)]

Table 1 Comparison of performances of three methods

Method	Time of convergence (in seconds)	MSE	
NN + LM	17.490	0.0047	Example 1
NN + DE	50.112	6.8830	
NN + LM + DE	24.899	0.0030	
NN + LM	18.991	0.3115	Example 2
NN + DE	121.148	3.5470	
NN + LM + DE	30.985	0.0059	
NN + LM	20.929	0.0038	Example 3
NN + DE	102.347	2.6680	
NN + LM + DE	54.895	0.0001	

Figure 16 gives the comparison of proposed DE + LM + NN scheme with the NN scheme. In the proposed DE + LM + NN scheme, the network is trained both by DE and Levenberg Marquardt algorithm. From this figure we marked that the proposed DE + LM + NN scheme has exhibited the expected identification performance i.e. the error between the true system and the identified one is minimum. Table 1 summaries the performance of the proposed method of system identification (NN + LM + DE) over the existing ones (NN, NN + DE) for different example and case studies.

5 Conclusions

The paper has described the scope of improving system identification of nonlinear systems by using proposed DE + LM + NN approach. In the proposed identification frame work, differential evolution is used only to find approximate values in the vicinity of the global minimum. These approximate weight values are then used as starting values for a faster convergence algorithm i.e. Levenberg Marquardt algorithm. From the results presented in Sect. 4, it is clear that there is certainly an improvement in identification performance for nonlinear systems over the existing approaches. In comparison to use of only differential evolution approach proposed DE + LM + NN approach provides better system identification performance in terms of speed of convergence and identification capability.

References

1. Chen S, Billings SA, Luo W (1989) Orthogonal least squares methods and their application to non-linear system identification. Int J Control 50:1873–1896
2. Billings SA, Wei HL (2005) A new class of wavelet networks for nonlinear system identification. IEEE Trans Neural Networks 16:862–874
3. Narendra KS, Parthaasarathy K (2005) Identification and control of dynamical systems using neural networks. IEEE Trans Neural Networks 16:8624–8874
4. Storn R (1999) System design by constraint adaptation and differential evolution. IEEE Trans Evol Comput 3:22–34
5. Ilonen J, Kamarainen JK, Lampinen J (2003) Differential evolution training algorithm for feed forward neural networks. Neural Proc Lett 17:93–105
6. Lin C-T, George Lee CS (1996) Neural Fuzzy Systems: a Neuro-fuzzy synergism to intelligent systems. PHI, Inc., New Jersey
7. Box GEP, Jenkins GM (1970) Time series analysis, forecasting and control. Holden Day, San Francisco