International Journal of Hybrid Information Technology
Vol.8, No.8 (2015), pp.393-404
http://dx.doi.org/10.14257/ijhit.2015.8.8.39

A Nonlinear System Identification based on Additive Expression Tree Model with Cuckoo Search

Bin Yang

*School of Information Science and Engineering, Zaozhuang University,
Zaozhuang, P.R. China 277160
batsi@126.com*

Abstract

In this paper, an efficient approach of combining additive expression tree model (AET) with hybrid evolutionary method is proposed to identify nonlinear systems. As linear variant of additive tree model, additive expression tree model is proposed to encode the mathematical formulations. For finding the optimal structure and parameters of systems, a hybrid evolutionary method integrating a new structure based evolutionary algorithm and cuckoo search is employed. We illustrate some experimental comparisons with neural network, neural network integrating fuzzy system and symbolic regression methods. Experimental results reveal that our model and optimization method perform better.

Keywords: *Nonlinear system identification, additive expression tree, cuckoo search, hybrid evolutionary*

1. Introduction

It is clear that most systems in the real life are nonlinear in nature, such s signal processing, biological process, communication, control, etc [1–4]. In order to understand and model the mechanism of these real systems, we usually face the problem of nonlinear system identification. System identification is modeling procedure where the mathematical representation of the output, past output or past input or both, can be obtained with the observed data. n this process, two main tasks must be addressed: mathematical models and model parameters [5].

For nonlinear system identification, the most commonly used model is artificial neural networks (ANN). ANN with intrinsic robustness and nonlinear characteristics, could approximate any continuous function, and is suitable or modeling complex system dynamics in a systematic approach [6,7]. The researches contain two parts: the one is to select the appropriate neural network model and the other is to select appropriate evolutionary algorithm which s used to optimize the parameters of neural network model. Wavelet neural network (WNN), recurrent neural network (RNN) and radial basis function RBF) network are examples of ANNs commonly used for nonlinear system identification [8–10]. In order to improve the performance of a neural network, some neural fuzzy networks have proposed. Juang proposed a TSK-type recurrent fuzzy network (TRFN), TRFN with supervised learning (TRFN-S) and TRFN with genetic learning (TRFN-G) in which free parameters were designed by genetic algorithm [11]. On the basis of TSK fuzzy system, Cheng used wavelet neural network (WNN) to replace the constant or a linear function of inputs in conclusion part of traditional TSK fuzzy model (FWNN) [12]. A hybrid method based improved particle swarm optimization and gradient descent algorithm was used to find the optimal parameters. Leandro also proposed FWNN for nonlinear system identification, and maximum correntropy criterion was used as the cost function of error backpropagation algorithm n order to reduce the negative effects of the unknown noise [13]. Bidyahar proposed a new algorithm named opposition based differential evolution, to optimize the parameters of neural network [14].

ISSN: 1738-9968 IJHIT

Neural network is very powerful, but it is a black box building mapping from inputs to outputs. It is difficult to analyze the internal mechanism of system. Recently some researcher have obtained mathematical formulations directly based on the observed data, whose theoretical and numerical analysis could provide insight and guidance for the real system. Fallah-Mehdipour used genetic programming (GP) to develop a reservoir operation policy simultaneously with inflow prediction [15]. Gandom presented a new multi-stage GP strategy (MSGP) for modeling nonlinear systems, and proved that MSGP as more accurate than GP and ANN models [16]. Zhang used a nonlinear ordinary differential equation (ODE) model to model the complex regulatory relationships among genes. According to analysis of ODE, gene regulatory network was inferred [17].

All complicated nonlinear maps are additive models of a number of linear and nonlinear terms. In this paper, as linear variant of additive tree model, additive expression tree (AET) model is first proposed to encode the linear and nonlinear systems. We propose a hybrid evolutionary method, in which a new structure-based evolutionary algorithm is used to optimize the architecture of systems and selection of input variables and corresponding parameters are evolved by cuckoo search.

2. Materials and Methods

2.1. Additive Expression Tree Model

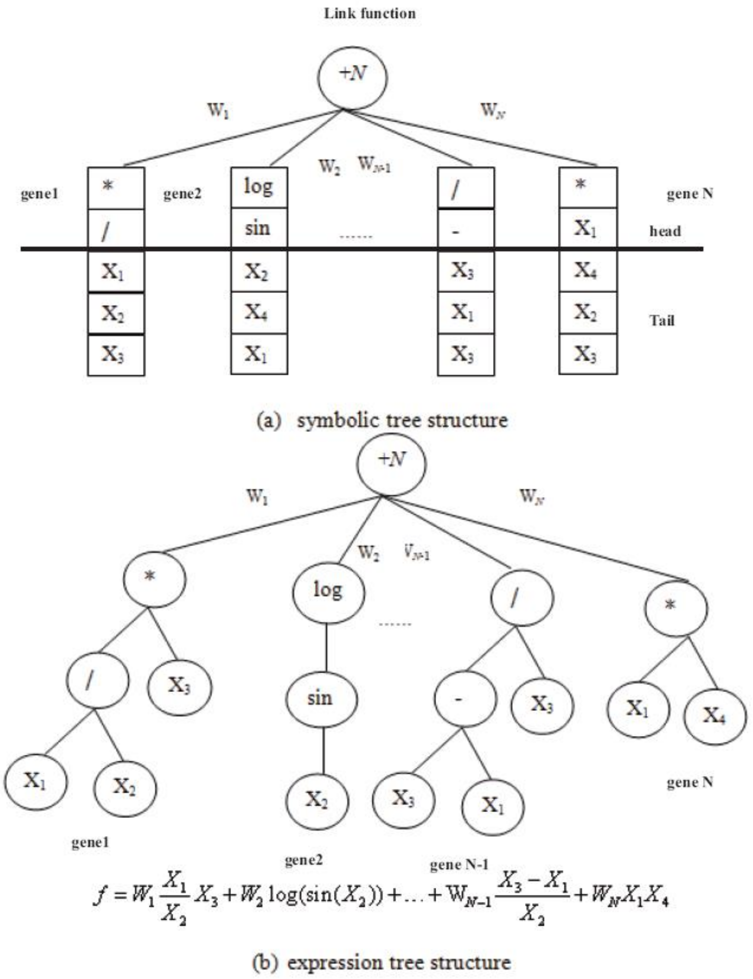
We use a structure-based evolutionary algorithm to evolve the architecture of the additive expression models for the linear/nonlinear system. For this purpose, we encode the linear/nonlinear expression into an additive expression tree model as illustrated in Figure 1. Figure 1(a) is the linear/nonlinear expression of symbolic tree structure, which need be created randomly, and Figure 1(b) is the corresponding expression tree structure.

N is an integer number and represents the maximum number of linear/nonlinear terms. Each term is encoded as gene form of gene expression programming (GEP). A GEP gene is a string of function and terminal symbols, which is composed of a head and a tail [19]. The head part contains both function and terminal symbols, whereas the tail part contains terminal symbols only. The head could be created through selecting symbols randomly from the set I_1. The symbols of tail are selected from function set F only. For each problem, user must determine the head length (h). The tail length (t) is computed as:

$$t=(n-1)\times h+1 \tag{1}$$

Where n is the maximum number of arguments of functions. As illustrated in Fig.1(a), h is 2, n is set as 2, h is 3, and the length of gene is 5.

The node (link function) could be selected from instruction set $I_0=\{+2,+3,\ldots,+N\}$, which returns the weighted sum of a number of linear/nonlinear terms according to the GEP gene expressions. A GEP string could be created randomly from operator set $I_1=F\bigcup T=\{+,-,*,/,sin,cos,tan,tanh,\mathrm{e}^x,rlog,x,R\}$. According to the need of problems solved, we could add some complex and classic nonlinear functions into operator set I_1, such as Gaussian function, Sigma function, Hill function, etc. If the nonlinear system is a polynomial, then the operator set I_1 can be defined as $I_1=\{*2,*3,\ldots,*n,x,R\}$ (*n represents that n values multiply).



(a) symbolic tree structure

(b) expression tree structure

Figure 1. Example of a System in the Form of Additive Expression Tree Model

2.2. Structure Optimization Methods

To search an optimal or near-optimal additive expression tree model is formulated as an evolutionary finding process. We used the structure operators as following:

(1) Mutation. We use two mutation operators to generate offsprings from the parents, which are described as following:

(a) One-point mutation. Select one point in the tree randomly, and replace it with another symbol, which selects from set I_1. Notice that in the head any symbol could be changed, but in the tail the terminal symbols are allowed to be changed only.

(b) One-gene mutation. Randomly select one GEP gene in the tree, and replace it with anther newly generated gene.

(2) Recombination. First two parents are selected according to the predefined crossover probability P_c. Select one pint in one GEP gene randomly. The symbol string after the point is exchanged between parents, creating two new offprings.

(3) Selection. The roulette-wheel method is used to select offsprings from parent population according to the fitness. The fittest individuals have the higher probability of being selected. Individuals with worse fitness may not be chosen at all.

2.3. Parameter Optimization of Models using Cuckoo Search

To find the optimal coefficients of an additive expression tree model, the cuckoo search (CS) method is used below. CS is one of the latest nature-inspired metaheuristic algorithms, proposed in 2009 by Xin-She Yang [20]. This algorithm is based on the brood parasitism of many cuckoo species and enhanced by the Lévy flights [20]. By analysis of some evolutionary algorithms, CS is potentially far more efficient than particle swarm optimization, genetic algorithms and artificial bee colony [23].

CS follows three idealized rules: (1) Each cuckoo only lays one egg at a time, which is dumped in randomly selected nest; (2) The nests with high quality of eggs could be selected to the next generation; (3) The number of available nests is fixed, and the egg laid by other cuckoo is discovered by the host cuckoo with a probability $p_a \in [0,1]$.

According to the above three idealized rules, the process of CS is described as followed.

(1) Create the initial population randomly $X_i (i = 1, 2, \ldots n)$, which represent host nests.

(2) Compute the fitness values of all population. If the maximum number of generations is reached or a satisfactory solution is found, then stop.

(3) Lévy flight is performed to generate new solutions.

$$X_i^{t+1} = X_i^t + \alpha \oplus L\acute{e}vy(\lambda) \tag{2}$$

Where X_i^t is the i-th solution at t-th generation, α is step size which could control the scale of random search. In general, $\alpha = 1$. $\oplus$ means entrywise multiplications, $L\acute{e}vy(\lambda)$ abides by Lévy probability distribution:

$$L\acute{e}vy(\lambda)\, u = t^{-1-\lambda},\ 0 < \lambda \le 2 \tag{3}$$

$L\acute{e}vy(\lambda)$ could be computed using the following equation [23]:

$$L\acute{e}vy(\lambda)\, \frac{\phi \times \mu}{|v|^{\frac{1}{\lambda}}}$$

$$\phi = \left(\frac{\Gamma(1+\lambda) \times \sin(\Pi \times \frac{\lambda}{2})}{\Gamma((\frac{1+\lambda}{2}) \times \lambda \times 2^{\frac{\lambda-1}{2}})}\right)^{\frac{1}{\lambda}} \tag{4}$$

Where μ and v follow Gaussian distributions.

(4) According to probability p_a, discard the worse solutions. Create the same number of new solutions using preference random walk.

$$X_i^{t+1} = X_i^t + r(X_m^t - X_n^t) \tag{5}$$

Where *r* is scaling factor, which is created randomly from [0, 1]. X_m^t and X_n^t are random solutions at *t*-th generation. Go to step (2).

2.4. Fitness Function Definition

Root mean square error (RMSE) is used as fitness function to evaluate the performance of candidate model.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathrm{x}_1^i - \mathrm{x}_2^i)^2} \tag{6}$$

Where *N* is the number of samples, x_1^i and x_2^i are the actual and model output of *i*-th sample, respectively.

3. Experimental Results and Illustrative Examples

In this section, our proposed model and learning algorithm are applied to two kinds of nonlinear system identifications. The parameters used in the additive expression tree model and hybrid learning are chosen experimentally and listed in Table 1.

Table 1. Parameters used in Two Experiments

Parameters	Initial value
Population size, *PS*	30
Mutation rate, *mr*	0.4
Recombination rate, p_c	0.7
Maximum search step	200
Initial coefficients	Rand [-3.0, 3.0]
CS **Error! Not a valid link.**	30
CS p_a	0.25
CS maximum search step	100

3.1. Experiment 1

The nonlinear system to be identified is described by

$$y(k) = 0.72y(k-1) + 0.025y(k-2)u(k-2) + 0.01u^2(k-3) + 0.2u(k-4) \tag{7}$$

Where $y(k)$ is the output of the system at the *k*-th time point, $u(k)$ is the plant input. In order to make the comparison fairly, the input signals used for training additive expression tree model are same as in literatures [11,12,21], which is an iid uniform sequence over [-2,2] for about half of the 900 time points and the remaining data is given by $1.05\sin(\pi k/45)$. To test the performance of identification model, the following input is used for test.

$$u(k)=\begin{cases} sin(\pi k/25), & k<250 \\ 1.0, & 250\le k<500 \\ -1.0, & 500\le k<750 \\ 0.3sin(\pi k/25)+0.1sin(\pi k/32) & \\ +0.6sin(\pi k/10) & 750\le k<1000 \end{cases} \tag{8}$$

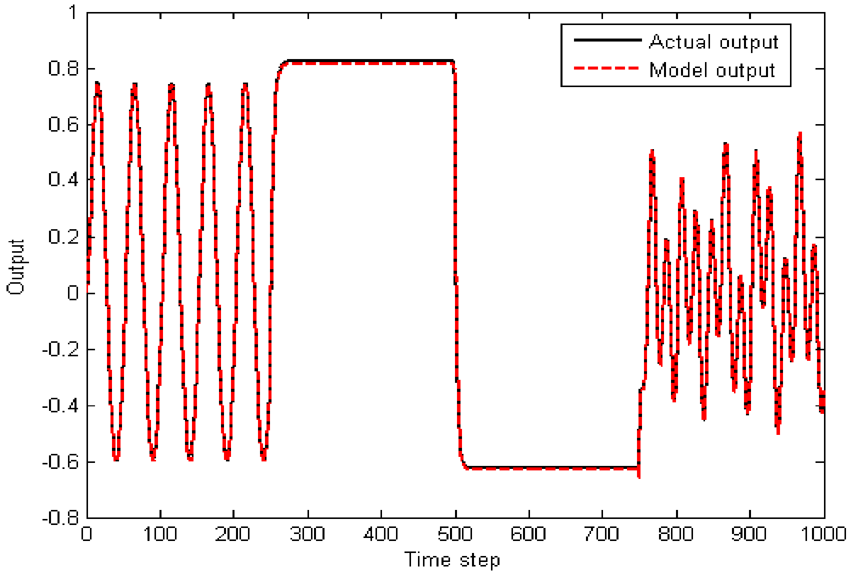
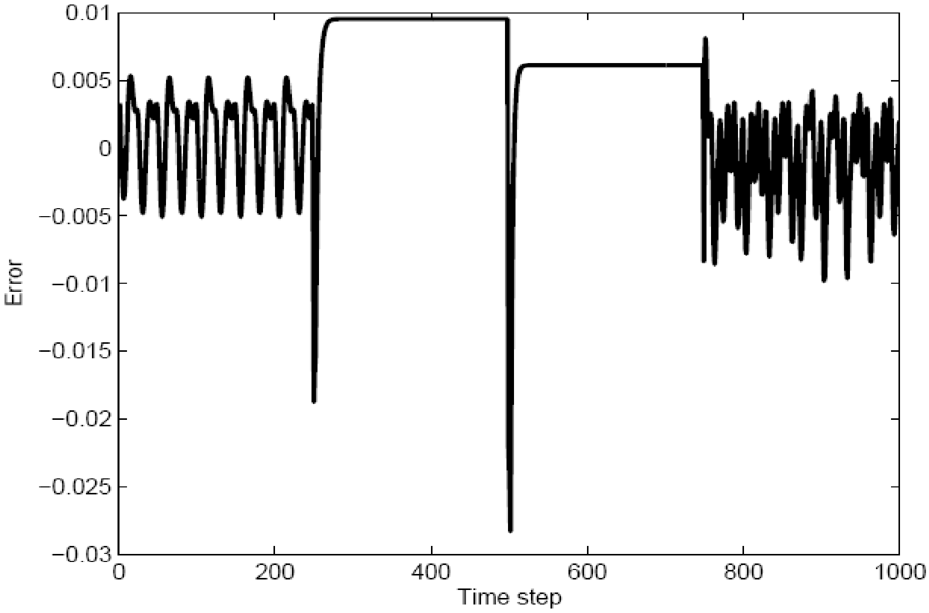


Figure 2. Our Method Identification Performance for Example 1



Error! Not a valid link.

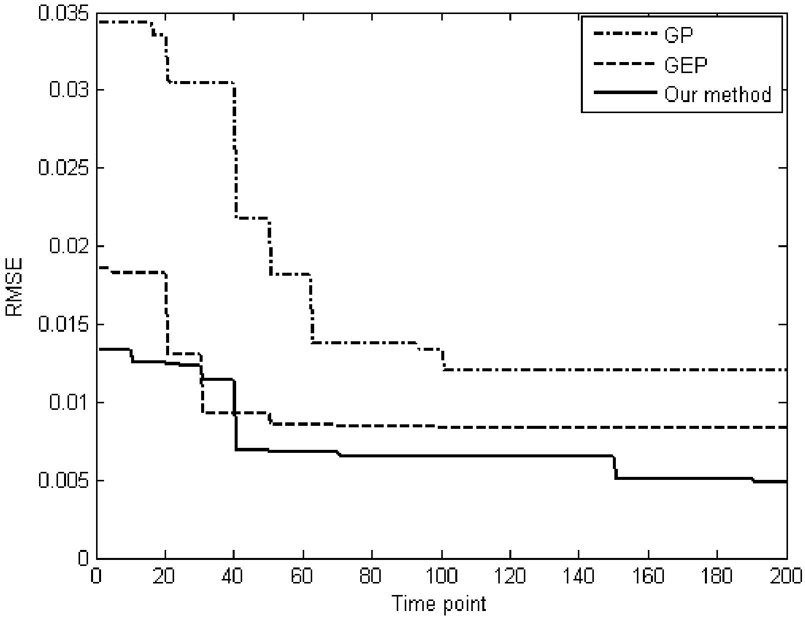


Figure 4. RMSE Values Obtained using Three Methods for Example 1

The used instruction set $I_0 = \{+2,+3,+4,+5,+6,+7,+8\}$ and $I_1 = \{*,+,-,/,\sin,\cos,\mathrm{elog}, y(k-1), \mathrm{y(k-2)}, u(k-2), u(k-3), u(k-4), R\}$. Figure 2 gives the identification performance between actual and identification model using our method. Figure 3 gives the identification error. From the Figures 2 and Figure 3, it can be see that identification system is nearly as same as actual one. Figure 4 illustrates the RMSE reduction curve using GP, gene expression programming (GEP) and additive expression tree during training. As shown in Figure 4, the convergence of our method is faster than GP and GEP, which are commonly used to identify the mathematical formulations. The RMSE values of identification system using some different methods for training and testing data are listed in Table 2. From Table 2, it is clear that our method performs better than neural network, neural network combining fuzzy rules and symbolic regression algorithms.

Table 2. Comparison of Different Models for Example 1

Model	RMSE of training	RMSE of testing
ERNN[12]	0.036	0.078
RSONFIN[22]	0.03	0.06
TRFN-S[11]	0.0067	0.0313
FWNN[21]	0.018713	0.0201069
FWNN-inline-PSO[12]	0.0067	0.0163
GP	0.011	0.018
GEP	0.008337	0.009002
Our method	**0.0048**	**0.006342**

3.2. Experiment 2

In this section, the plant to be identified is described by the following equation:

$$y(k+1) = f(y(k), y(k-1), y(k-2), u(k), u(k-1)), \tag{9}$$

where

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^2 + x_2^2} \tag{10}$$

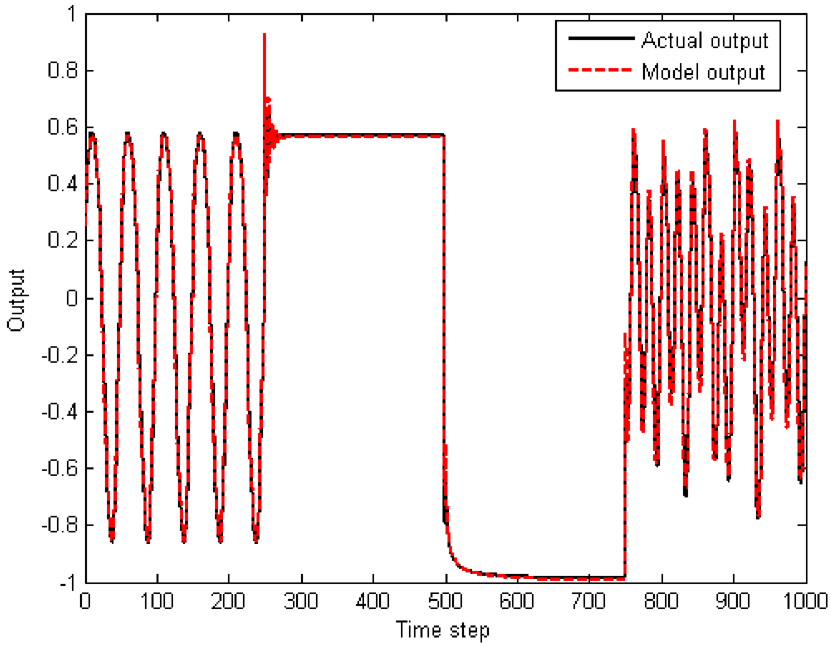


Figure 5. Our Method Identification Performance for Example 2

The identification system has been used in [11, 12, 21]. In order to make the comparison fairly, the training and testing signals are same as [21]. The used instruction set $I_0 = \{+4, +5, +6, +7 + 8\}$ and

$$I_1 = \{+, -, *, /, \frac{1}{1-e^{-x}}, \sin, \tan, \cos, \mathrm{e}^{-\frac{(\mathrm{x}-\mathrm{a})^2}{b^2}}, \frac{2|a|}{1+e^{-2|a|x}}, y(k), y(k-1), y(k-2), u(k), u(k-1)\}$$

Figure 5 gives the identification performance between actual and identification model using our method. Fig.6 gives the identification error. From the two Figures, it can be see that identification system is very close to actual one. Figure 7 illustrates the RMSE reduction curve using GP, GEP and our method during training, which shows that the convergence of our method is faster than GP and GEP. The RMSE values of identification system using some different methods for training and testing data are listed in Table 3. From Table 3, it is clear that our method performs better than the other models in terms of RMSE of testing data.

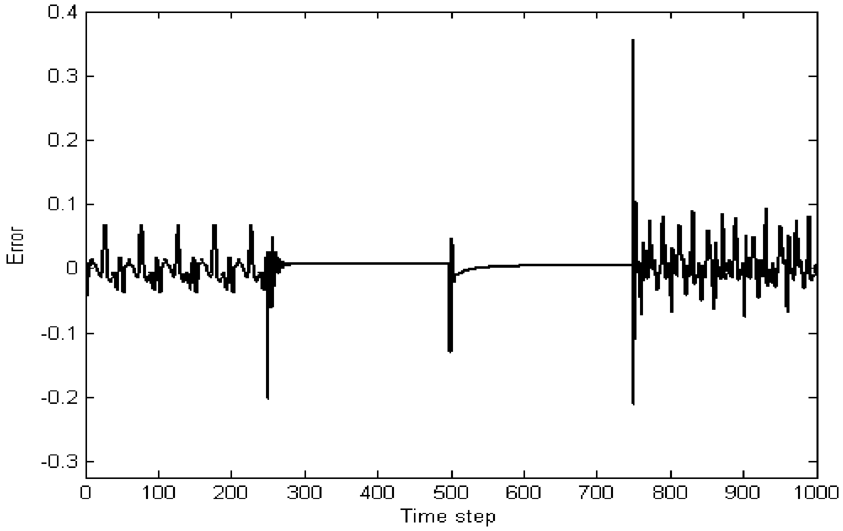


Figure 6. Our Method Identification Error for Example 2

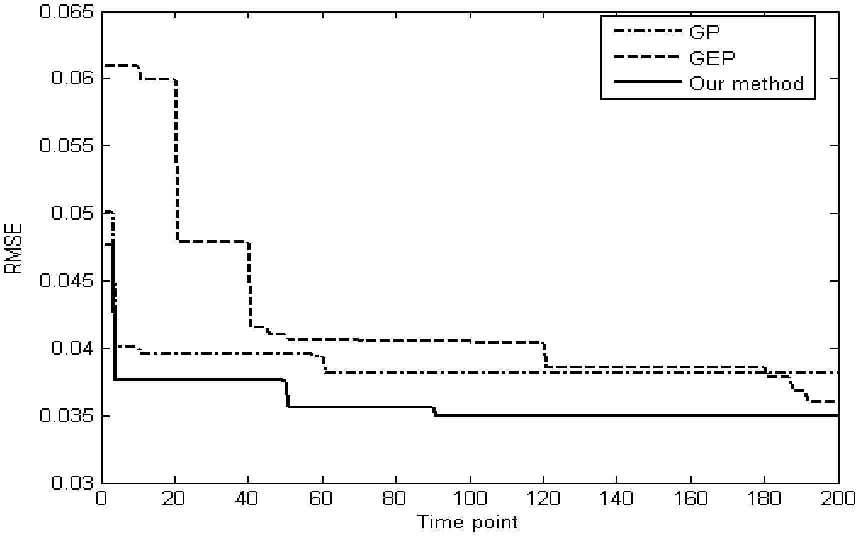


Figure 7. RMSE Value Obtained using three Methods for Example 2

Table 3 Comparison of different models for Example 2

Model	RMSE of training	RMSE of testing
RFNN[12]	0.0114	0.0575
RSONFIN[22]	0.0248	0.0780
TRFN-S[11]	**0.0084**	0.0346
FWNN[21]	0.028232	0.030125
FWNN-inline-PSO[12]	0.0202	0.0274
GP	0.037909	0.03112
GEP	0.036788	0.027364
Our method	0.034956	**0.02683**

4. Conclusion

In this paper, an additive expression tree model and novel hybrid evolutionary method are proposed for identification of nonlinear system. Additive expression tree model is used to encoded mathematical formulations, and a novel hybrid algorithm based on a new structure-based evolutionary algorithm and cuckoo search is employed to find the optimal structure and parameters of systems. Additive expression tree model has the following advantages. (1) The model contains linear entities. Thus user may reduce the difficulties of program implementation. Due to not handle the nonlinear tree structure, runtime reduces sharply. (2) The form of additive expression tree model is very near to the representation of the system which we need identify. (3) The additive root node could make the linear/nonlinear system divided into several blocks, which reduces the problem complexity. Therefore additive expression tree model performs better than neural network, neural network with fuzzy rules, GP and GEP in both two simulation examples of system identification.

Acknowledgements

This work was supported by the PhD research startup foundation of Zaozhuang University (No.1020702), and the Natural Science Foundation of Shandong Province, China (No.ZR2015PF007).

References

[1] H. Hujiberts, H. Nijmeijer and R. Willems, “System identification in communication with chaotic systems”, IEEE Trans. Circuits Syst. I47, vol. 6, **(2000)**, pp. 800-808.

[2] Y. Xie, B. Guo, L. Xu, J. Li and P. Stoica, “Multistatic adaptive microwave imaging for early breast cancer detection”, IEEE Trans. Biomed. Eng., vol. 53, no. 8, **(2006)**, pp. 1647-1657.

[3] Z. Tian and M. J. Zuo, “Health condition prediction of gears using a recurrent neural network approach”, IEEE Trans. Rel., vol. 59, **(2010)**, pp. 700-705.

[4] Q. Hua, W. T. M. Wentao and Z. J. hong, “Prediction method for network traffic based on maximum correntropy criterion”, Communications, China, vol. 10, **(2013)**, pp. 134-145.

[5] Z. Sigrist, P. Legrand, E. Grivel and B. Alcoverro, “Considering the measurement noise for a nonlinear system identification with evolutionary algorithms”, 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, **(2012), pp.** 2114-2118.

[6] P. J. Angeline, G. M. Saunders and J. B. Pollack, “An evolutionary algorithm that constructs recurrent neural networks”, IEEE Trans.Neural Networks, vol. 5, **(1994)**, pp. 54-65.

[7] X. Yao, “Evolving artificial neural networks”, Proc. IEEE, vol. 87, **(1999)**, pp. 1423-1447.

[8] C. J. Lin, “Nonlinear systems control using self-constructing wavelet networks”, Appl Soft Comput., vol. 9, no. 1, **(2009)**, pp. 71-9.

[9] J. L. X. Xue and W. Xiang, “Nonlinear system identification with modified differential evolution and rbf networks”, Proc. IEEE Fifth Int. Conf. Advanced Comput. Intell., Nanjing, Jiangsu, China, **(2012),** pp. 332-335.

[10] W. Yu, “Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms”, Information Sciences, vol. 158, **(2004)**, pp. 131-147.

[11] C. F. Juang, “A TSK-Type Recurrent Fuzzy Network for Dynamic Systems Processing by Neural Network and Genetic Algorithms”, IEEE Transactions on Fuzzy Systems, vol. 10, no. 2, **(2002)**, pp. 155-170.

[12] R. Cheng and Y. P. Bai, “A novel approach to fuzzy wavelet neural network modeling and optimization”, International Journal of Electrical Power & Energy Systems, vol. 64, **(2015)**, pp. 671-678.

[13] L. Linhares, A. Fontes, A. Martins, F. Araújo and L. Silveira, “Fuzzy Wavelet Neural Network Using a Correntropy Criterion for Nonlinear System Identification”, Mathematical problems in engineering (in press).

[14] B. Subudhi and D. Jena, “A differential evolution based neural network approach to nonlinear system identification”, Applied Soft Computing, vol. 11, **(2011)**, pp. 861-871.

[15] E. Fallah-Mehdipour, O. Bozorg Haddad and M. A. Mariño, “Real-Time Operation of Reservoir System by Genetic Programming”, Water Resources Management, vol. 26, no. 14, **(2012)**, pp. 4091-4103.

[16] A. H. Gandomi and A. H. Alavi, “Multi-stage genetic programming: A new strategy to nonlinear system modeling”, Information Sciences 181(23) **(2011)** 5227–5239.

[17] Y. Q. Zhang, Y. F. Pu, H. S. Zhang, Y. B. Su, L. F. Zhang and J. L. Zhou, “Using gene expression programming to infer gene regulatory networks from time-series data”, Computational Biology and Chemistry, vol. 47, **(2013)**, pp. 198-206.

[18] Y. H. Chen, J. Yang, Y. Zhang and J. W. Dong, “Evolving Additive tree models for System Identification”, International Journal of Computational Cognition, vol. 3, no. 2, **(2005)**, pp. 19-26.

[19] C. Ferreira, “Gene Expression Programming in Problem Solving”, Soft Computing and Industry, **(2002)**, pp. 635-653.

[20] X. S. Yang and S. Deb, “Cuckoo Search: Recent Advances and Applications”, Neural Computing and Applications, vol. 24, no. 1, **(2014)**, pp. 169-174.

[21] R. H. Abiyev and O. Kaynak, “Fuzzy wavelet neural networks for identification and control of dynamic plants a novel structure and a comparative study”, IEEE Trans Ind Electron, vol. 55, no. 8, **(2008)**, pp. 3133-40.

[22] C. F. Juang and C. T. Lin, “A recurrent self-organizing neural fuzzy inference network”, IEEE Trans Neural Networks, vol. 10, no. 4, **(1999)**, pp. 828-45.

[23] P. Civicioglu and E. Besdok, “A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms”, Artificial Intelligence Review, vol. 39, no. 4, **(2013)**, pp. 315-346.

Author

Bin Yang, is the teacher of Zaozhuang University. He has pursued his Ph.D. in School of Information science and Engineering from Shandong University, Jinan, P. R. China. He received his B.Sc. and Master degree in School of Information Science and Engineering from University of Jinan. His research interests include hybrid computational intelligence and its applications in time-series prediction, system identification and gene regulatory network.