2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Identification of Dynamic Systems Using a Differential Evolution-Based Recurrent Fuzzy System

Cristian K. dos Santos, Rogério P. Espíndola, Vinícius F. Vieira and Alexandre G. Evsukoff

***Abstract*—This work presents the development of a simulation model based on a recurrent fuzzy system with structure and parameter identification by a differential evolution algorithm. The proposed model is formulated by state space equation, in which the state transition function is a recurrent fuzzy system with two feedback connections and adjustable delay operators and the output function is a linear function of the states. The identification process relies on two instances of the differential evolution algorithm in a hierarchical fashion. The outermost is considered for combinatorial structure optimization and the innermost for optimization of continuous parameters. The new model is evaluated in some benchmark problems and the results showed the model achieved good numerical performance. Moreover, the results demonstrated the ability of differential evolution algorithm to optimize both the parameters as well as the structure of the model.**

I. Introduction

Recurrent fuzzy systems (RFS) [1] have been attracting great interest of researchers due to promising results in processing nonlinear dynamical systems [2]-[18]. They are characterized by feedback connections in their structure. Owing to their internal dynamic behavior, they deal with the reduction of the input dimension, yielding more compact models.

Many recurrent fuzzy systems have been recently proposed. They differ mainly in the recurrent structure and in the parameter identification method employed. Regarding recurrent structure, some approaches consider feedback connections from the output [2], [6], [10] while others use feedback connections from internal state variables [1], [3]-[5], [7]-[9], [11]-[18]. Some approaches that account for feedback connections from internal variables also explore features of local recurrence by feeding the output of internal elements locally back to itself [3], [7], [13]-[16], [18].

Regarding the parameter identification task, the main methods are based on evolutionary or gradient-based optimization, the latter being the most common approach [1]-[3], [5], [7], [9]-[10], [15]. Although their wide use, the gradient computations for recurrent structures of the neural learning methods are complex in derivation, easily trapped at local minima and highly dependent on the model structure as well.

To avoid gradient issues in recurrent fuzzy systems development, some authors propose evolutionary and swarm algorithms since they are derivative-free, stochastic and population-based optimization methods. Some examples are genetic algorithms [4], [5], [8], [12]; particle swarm optimization [11]; ant colony optimization [17]; and differential evolution [13]. In the last years, differential evolution (DE) [19], an evolutionary algorithm for continuous search spaces, has been attracting increasing interest for its simplicity as well as its ability to find the global optimum in different types of complex optimization problems [20]-[21], including system identification [13], [22]-[24]. In this particular case, DE seems to be a good choice since it can be used for learning models with different types of structures.

This work presents a simulation model based on a recurrent fuzzy system with simple structure and good performance. The model structure has two feedback connections and adjustable delay operators, as shown in Fig. 1. To provide memory to the model, the state feedback connection feeds the model internal variables back to its input, while the output feedback connection feeds the estimated model output back to the model input. Additionally, the adjustable delay operators are used to improve the model ability to cope better with dynamic systems with time delay. Furthermore, it is also supplied a method to identify the best blend of the structure and the parameters of a model using two instances of a differential evolution algorithm in a hierarchical fashion.

The contributions of this work are threefold. First, the model is designed to cope with simulation problems as in [1], [9], [12] while most of RFS found in the literature can cope only with prediction problems. Second, the model proposed has a simple recurrent structure that is different from others RFS. The last contribution is to employ the DE algorithm to identify the structure of the model, besides the usual approach of optimizing its parameters [13], [22]-[24].

The paper is organized as follows. The next section describes the model proposed. Section III introduces the

C. K. dos Santos is with the Inertial Systems Laboratory, Brazilian Navy Research Institute (IPqM), Rio de Janeiro, RJ 21931-095 Brazil (phone: +55-21-2126-5770; e-mail: cristian@ipqm.mar.mil.br).

R. P. Espíndola is with the Department of Computing and Applied Mathematics, Westside State University Center (UEZO), Rio de Janeiro, RJ 23070-200 Brazil (e-mail: rpespindola@uezo.rj.gov.br).

V. F. Vieira is with the Computer Science Department, Federal University of São João Del-Rei (UFSJ), São João Del Rei, MG 36301-360 Brazil (e-mail: vinicius@ufsj.edu.br).

A. G. Evsukoff, was with Federal University of Rio de Janeiro, Rio de Janeiro, RJ 21941-972 Brazil. He is now with the School of Applied Mathematics, Getulio Vargas Foundation (FGV), Rio de Janeiro, RJ 22250-900 Brazil (e-mail: alexandre.evsukoff@fgv.br).

This work was supported in part by the Brazilian Research Agencies, FAPERJ, FINEP, CNPq and CAPES.

differential evolution algorithm. Section IV presents the model's identification approach. The experiments and results are covered in Section V. Ultimately, the conclusions are discussed in Section VI.

II. Recurrent Fuzzy Systems

Recurrent Fuzzy Systems (RFS) are extensions of traditional fuzzy systems with some kind of recurrence in their structures, allowing the approximation of unknown order dynamic processes [1]. In this work, the structure of the model proposed has two feedback (state and output values) connections and adjustable delay operators, as shown in Fig. 1. Discrete in time non-linear dynamic systems can be represented by state space equations as:

$$\begin{aligned} \boldsymbol{x}_1(t+1) &= f\left(\boldsymbol{x}(t), \boldsymbol{u}(t-\delta_u), \hat{y}(t-\delta_{\hat{y}})\right) \\ \hat{y}(t) &= g(\boldsymbol{x}(t)) \end{aligned} \tag{1}$$

in which $\boldsymbol{x}_1(t+1)$ and $\boldsymbol{u}(t)$ are the state and input vectors, respectively; $f(\cdot)$ and $g(\cdot)$ are the state transition and output functions; $\hat{y}(t)$ is the output value of the system; and δ_u and $\delta_{\hat{y}}$ are the adjustable delay operators of the input and feedback signals. In special,

$$\boldsymbol{x}(\mathrm{t}) = [\boldsymbol{x}_1(t), \ldots \boldsymbol{x}_1(t-n-1)] \tag{2}$$

is the state vector made up by delayed copies of the first state variable, and the output function

$$g(\boldsymbol{x}(t)) = \lambda_0 + \sum_{i=1}^{n} \lambda_i \cdot \boldsymbol{x}_i(t) \tag{3}$$

represents a polynomial function in which λ_i is the coefficient of the state variable $\boldsymbol{x}_i(t)$, and λ_0 is the independent term. In Fig. 1, respectively, $q^{-\delta_u}$, q^{-1} and $q^{-\delta_{\hat{y}}}$ are the delay operators of the input, state and output signals.

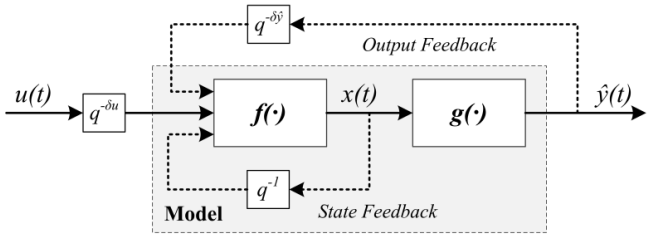

Figure 1. The structure of the proposed recurrent fuzzy system.

The differential evolution-based recurrent fuzzy system (DE-RFS) proposed in this paper is a zero-order TSK fuzzy system using membership functions [12] with regular and strong partitions of the variables spaces, such that the fuzzy partition is parameterized only by the location of triangles centers. The recurrent fuzzy rules are written as:

$$\begin{aligned} &\text{IF } \boldsymbol{x}(t) \text{ is } X_i \text{ AND } \boldsymbol{u}(t-\delta_u) \text{ is } U_j \text{ AND} \\ &\hat{y}(t-\delta_{\hat{y}}) \text{ is } Q_k \\ &\text{THEN } \boldsymbol{x}(t+1) = \theta_n \end{aligned} \tag{4}$$

in which X_i, U_j and Q_k are the fuzzy sets of the state, input and output variables, respectively; and θ_n is the parameter which defines the conclusion of the *n*-th rule.

The fuzzy sets X_i and U_j are multidimensional fuzzy sets and are defined as the Cartesian product of the fuzzy sets of their vectors' components. Therefore, their membership functions are the tensor product of their components membership vectors [12]. By using the product t-norm, the activation value of the *n*-th rule is calculated as:

$$w(\mathrm{t}) = \boldsymbol{\mu}_X(\boldsymbol{x}(t)) \otimes \boldsymbol{\mu}_U(u(t-\delta_u)) \otimes \boldsymbol{\mu}_Q(\hat{y}(t-\delta_{\hat{y}})) \tag{5}$$

in which $\boldsymbol{\mu}_X(\boldsymbol{x}(t))$, $\boldsymbol{\mu}_U(u(t-\delta_u))$ and $\boldsymbol{\mu}_Q(\hat{y}(t-\delta_{\hat{y}}))$ are the vectors of the membership functions of the fuzzy sets of the state, input and output variables, respectively and $\otimes$ is the Kronecker tensor product. So, the output of the SFR proposed is:

$$x(\mathrm{t}+1) = \boldsymbol{w}(t)\boldsymbol{\theta} \tag{6}$$

The size of the rule base is determined by the number of fuzzy sets used by the fuzzy partitions of the variables in the antecedent. As all state variables represent the same information, they can use the same partitioning scheme. Therefore, only the input variables are partitioned in a specific way and the size *M* of the rule base is calculated as:

$$M = p_x^{n_x} \times p_{\hat{y}} \times \prod_{i=1..n_u} p_{u_i} \tag{7}$$

in which n_x and n_u are the amounts of states and input variables; p_x is the number of fuzzy sets defined for the state variables domain; $p_{\hat{y}}$ is the number of fuzzy sets for the feedback variable domain; and p_{u_i} is the number of fuzzy sets of the *i*-th input variable.

III. Differential Evolution

The differential evolution optimization is a simple, efficient and robust technique able to deal with non-differentiable, non-convex, nonlinear and multimodal objective functions. Its main feature is the use of difference vectors to create new candidate solutions in the search for the best solution. There are several differential evolution variants and this paper adopts the canonical strategy DE/rand/1/bin, the most often one [13], [20]-[21], [23]-[24].

The population of the DE $\boldsymbol{V} \in R^{NP \times D}$ is comprised by NP real encoded vectors $v_i \in R^D$ randomly initialized in a continuous space, such that

$$v_{i,j}^{\min} \le v_{i,j} \le v_{i,j}^{\max}, \qquad i = 1,\ldots,NP; j = 1,\ldots,D \tag{8}$$

in which $v_{i,j}^{\min}$ and $v_{i,j}^{\max}$ are respectively the lower and upper limits of each variable.

Three evolutionary operators – mutation, crossover and selection – are used to evolve the initial population towards the best solution, in which the mutation and crossover operators are used to generate new experimental vectors. The mutation operator is defined as:

$$\boldsymbol{z}_i^c = \boldsymbol{v}_{r1}^c + F \times \left(\boldsymbol{v}_{r2}^c - \boldsymbol{v}_{r3}^c\right) \tag{9}$$

in which c is the generation; $\boldsymbol{z}_i^c$ is the new vector; $\boldsymbol{v}_{r1}^c$ is the base vector; F is the scale factor of the difference between the differential vectors $\boldsymbol{v}_{r2}^c$ and $\boldsymbol{v}_{r3}^c$; and $r_1 \neq r_2 \neq r_3 \in [1, ND]$ are all indexes randomly selected.

In turn, the binomial crossover operator is defined as:

$$\boldsymbol{s}_{i,j}^c = \begin{cases} \boldsymbol{z}_{i,j}^c & if\ \left(\mathrm{rnd}_j \le CR\right) \text{or } j = j_{rndn} \\ \boldsymbol{v}_{i,j}^c & otherwise \end{cases} \tag{10}$$

in which the indexes c, i e j are related respectively to the generation, the vectors and their components; $\boldsymbol{s}_{i,j}^c$, $\boldsymbol{z}_{i,j}^c$ and $\boldsymbol{v}_{i,j}^c$ are respectively the trial vector component, the vector component after mutation and the base vector component; CR is the crossover constant; $\mathrm{rnd}_j \in [0,1]$ and $\mathrm{j}_{\mathrm{rndn}} \in [1, ND]$.

The selection mechanism is responsible for choosing the best vector between the base vector and its trial alternative for the next generation. It is made by comparing their fitness values using the objective function $J(\cdot)$ in such a way that:

$$\boldsymbol{v}_i^{c+1} = \begin{cases} \boldsymbol{s}_i^c & if\ \left(J(\boldsymbol{s}_i^c) < J(\boldsymbol{v}_i^c)\right) \\ \boldsymbol{v}_i^c & otherwise \end{cases} \tag{11}$$

The objective function $J(\cdot)$ used in this paper is the root mean square error (RMSE) between the real output and its estimation by the model output, computed as:

$$J = \sqrt{\frac{1}{N} \times \sum_{t=1}^{N} \left(y(t) - \hat{y}(t)\right)^2} \tag{12}$$

These three operators are iteratively applied to the vectors of the population until some stop criterion is met. Generally, a maximum number of generations/evaluations is used.

IV. The Identification Approach

The proposed approach simultaneously identifies the structure and the parameters of the model using two instances of the differential evolution algorithm in a hierarchical fashion. The outermost algorithm (DE1) deals with the structure optimization while the innermost one (DE2) copes with the parameters optimization. For each structure evaluated, a new parameter set is optimized in order to find the best blend structure-parameters. The proposed approach is depicted using flowcharts (Fig. 2, Fig. 3 and Fig. 4). The Fig. 2 and Fig. 3 represent DE1 while Fig. 4 represents DE2. The Fig. 3 portrays the objective function of DE1.

The structure identification process defines the model order, the number of fuzzy sets in the domain of each variable considered in the antecedent of the rules and the values of the delay operators. In such a case, the following design vector is defined:

$$\bar{\boldsymbol{v}} = \left[n_x, p_x, p_u, p_{\hat{y}}, \delta_u, \delta_{\hat{y}}\right] \tag{13}$$

in which each component $\bar{v}_i$ is an integer number which may take only a small finite number of discrete values defined a priori. Therefore, the structure identification is addressed as a combinatorial optimization problem.

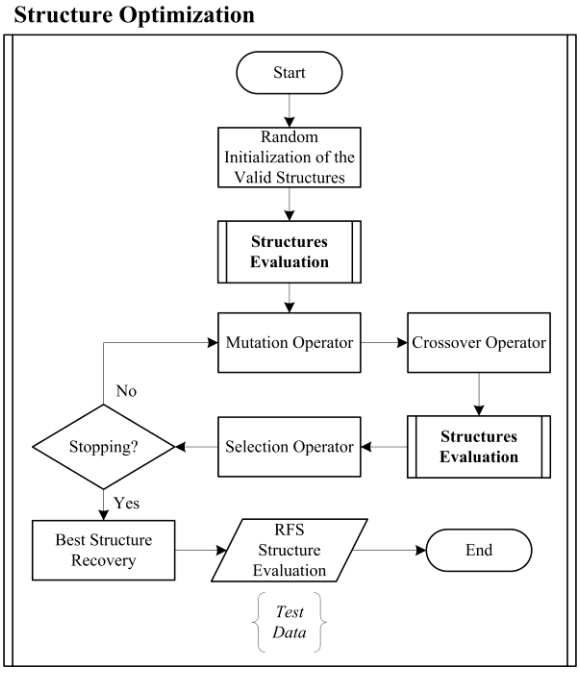

Figure 2. Simultaneous structure and parameters identification approach.

Since differential evolution encodes all its vectors internally as floating-point numbers regardless the type of the design variables, the continuous values of the vectors are rounded towards the nearest allowed integers only in the objective function in such a way that no change is made in the differential evolution algorithm. This strategy is a very simple one and presented good results.

The structures evaluation process employs two verification phases. The first one ensures that only valid structures will have their parameters optimized. In this case, invalid structures

are penalized with a very high fitness values. The other verification ensures that only new structures are evaluated into every generation to avoid unnecessary reevaluation of structures that had already been discovered.

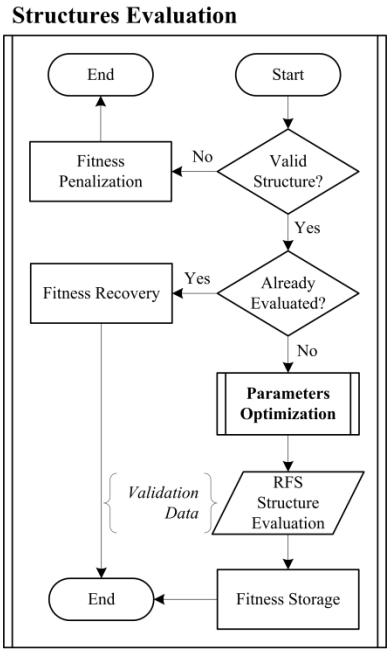

Figure 3. Structures Evaluation Process.

The parameter identification is simpler than the structure identification and must adjust the free parameters associated with the rule conclusion. In this case, another design vector is defined:

$$\bar{\bar{v}} = [\theta] \tag{14}$$

Therefore, the parameter identification issue is addressed as a continuous optimization problem.

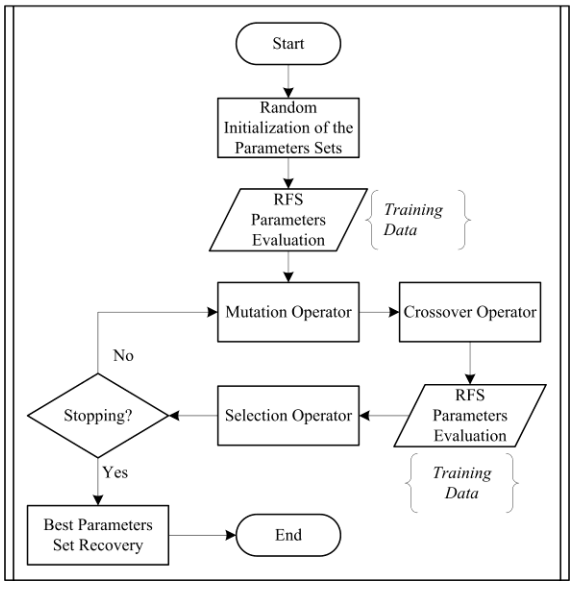

Figure 4. Parameters Optimization Process.

V. Experiments and Results

This section illustrates the performance of the proposed model in three benchmark examples. The first two examples are nonlinear dynamic systems and the last one is a chaotic time series.

A. Simulation Examples

1) Example #1

The system to be identified in this example is described by the following equation:

$$y(t+1) = \frac{y(t)y(t-1)y(t-2)u(t-1)[y(t-2)-1]+u(t)}{1+y^2(t-1)+y^2(t-2)} \tag{15}$$

The training dataset was computed from an input generated as follows: 400 samples of an independent and identically distributed uniform sequence over the interval $[-1,1]$ and 400 samples of a sinusoidal signal $u(t) = 1.05\sin(\pi t/45)$. The validation dataset was generated considering 1000 samples of an APRBS sequence on the interval $[-1,1]$. Ultimately, the testing dataset was generated considering 1000 samples of the following signal:

$$u(t) = \begin{cases} \sin(\pi t/25) & t < 250 \\ 1 & 250 \le t < 500 \\ -1 & 500 \le t < 750 \\ 0.3\cdot\sin(\pi t/25) & \\ +0.1\cdot\sin(\pi t/32) & \\ +0.6\cdot\sin(\pi t/10) & 750 \le t \le 1000 \end{cases} \tag{16}$$

2) Example #2

In this example, the system to be identified is portrayed by the following equation, which has longer input delays than Example #1:

$$\begin{aligned} y(t+1) = 0.72y(t) + 0.025y(t-1)u(t-1) + \\ +0.01u^2(t-2) + 0.2u(t-3) \end{aligned} \tag{17}$$

The training, validation and testing datasets for this example were generated using the same control inputs described in the previous example.

3) Example #3

This example deals with the identification of the Henon chaotic discrete-time series, described by the following delay-difference equation:

$$y(t+1) = -1.4y^2(t) + 0.3y(t-1) + 1 \tag{18}$$

Three thousand samples are generated by the initial state $[y(1), y(0)] = [0.4, 0.4]$ in which the first 1000 samples are used for training, the following 1000 samples for testing and the remaining 1000 samples are used for validation.

B. Experimental Setup

After some preliminary experiments, the control parameters of both differential evolution algorithms DE1 and DE2 were defined as:

- **DE1**: NP=12; CR=0.3; F=random(0.3, 0.9) ; MAX_GEN=50
- **DE2**: NP=20; CR=0.8; F=random(0.3, 0.9) ; MAX_GEN=1500

The identification approach was run 50 times for each problem with different random number generation seeds due to the stochastic nature of the differential evolution.

The design vector $\bar{v}$ used by DE1 is defined in such a way that $\bar{v}_1 = [1,3]$ accounts for the number of states variables; $\bar{v}_2, \bar{v}_3, \bar{v}_4 = [2,9]$ account for the number of fuzzy sets in the domain of state, input and feedback output variables, respectively; $\bar{v}_5 = [0,5]$ accounts for the lag of input variable; and $\bar{v}_6 = [1,5]$ accounts for the lag of feedback output variable. The design vector $\bar{\bar{v}}$ used by DE1 is defined in such a way that its components $\bar{\bar{v}}_i \in [0,1]$.

The identification process starts with the standardization of the variables used in the interval $[-1,1]$. The minimum and maximum outputs of the plant are then stored to convert the model output into the actual system scale. The results and discussions of the experiments are described in the next subsection.

C. Results and Discussion

The plot of the DE-RFS output against the actual systems output, as described in the section V-A, are shown respectively in Fig. 5, Fig. 6 and Fig. 7. It can be observed that the DE-RFS can track adequately the actual systems.

Table 1 compares the results obtained by the proposed method (DE-RFS) to those ones recently reported in the literature. The performances of the presented model are described in terms of RMSE averages and standard deviations after 50 runs with different random number generation seeds. The proposed method is better than those ones presented in the Table 1, at the same time, only in the Example #2. Regarding Examples #1 and #3, the proposed method is comparable to the others, except the methods presented in [13] and [7], respectively, which achieved better performance than this proposed method.

As well as the RFS proposed in this paper, the RFS presented in [9] and [12] are both based on state-space equations and their structures do not rely on the actual system output. On the other hand, all others RFS in the Table 1 consider the actual system output in their structures. Likewise, only [12] and [13] use an approach based on evolutionary optimization algorithms while all others are based on neural learning. In [12] the authors use a simple genetic algorithm to optimize the parameters of rule conclusions while in [13] the authors use a differential evolution algorithm to adjust fuzzy weights and biases of a recurrent fuzzy neural network, expressed as triangle fuzzy numbers.

TABLE I. COMPARATIVE ANALYSIS ON TEST DATA

Model	Example #1		Example #2		Example #3	
	Params	RMSE	Params	RMSE	Params	RMSE
DE-RFS (std.dev.)	92	0.0261 (0.0021)	26	**0.0080** (0.0009)	128	0.0121 (0.0013)
RSEFNN–LF [15]	34	0.0383	30	0.0279	94	0.0031
LRFNN–SVR [14]	29	0.0296	29	0.0306	31	0.0155
TRFN–S [5]	33	0.0346	33	0.0313	36	0.0206
IRSFNN [18]	42	0.0310	26	0.0220	40	0.0140
RFNN [3]	112	0.0114	-	-	60	0.0469
RIFNN [16]	-	-	36	0.0288	54	0.0510
RCNFS [7]	-	-	27	0.0221	-	**0.0035**
RFNN–DEO [13]	96	**0.0064**	-	-	-	-
CReNN [9]	51	0.2247	-	-	-	-
RFS–TSK [12]	47	0.0600	-	-	-	-

However, unlike this proposed approach, which uses the differential evolution as a tool to identify both the structure and the parameters at the same time, the methods presented in [12] and [13] employ evolutionary optimization algorithms as a tool to identify only the parameters. Moreover, all methods in Table 1 but [12] use Gaussian membership functions, whose parameters are also optimized, yielding fuzzy sets without linguistic meaning.

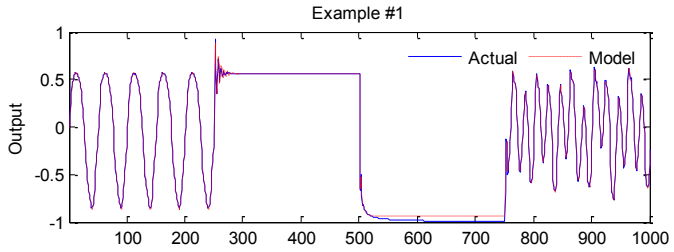

Figure 5. Model average performance on Example #1.

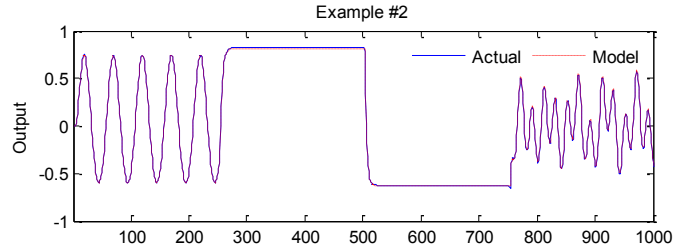

Figure 6. Model average performance on Example #2.

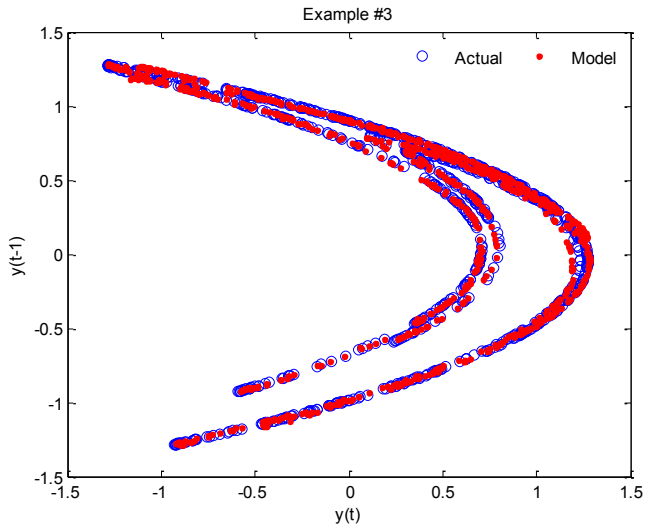

Figure 7. Model average performance on Example #3.

VI. CONCLUSION

This paper has presented an approach for the identification and simulation of dynamic systems using a differential evolution-based recurrent fuzzy system named DE-RFS. The structure of the DE-RFS embodies two feedback connections and adjustable delay operators, providing it the ability to process dynamic systems. The model identification is accomplished in a hierarchical fashion where two different instances of the differential evolution algorithm are considered. One of them is used to identify their structure and the other is used to identify their parameters.

The results obtained in the experiments showed the potential of the DE-RFS to cope with simulation of dynamic systems. Furthermore, it was demonstrated the ability of the differential evolution algorithm to deal not only with the continuous optimization of the parameters but also with the combinatorial optimization of the structure.

REFERENCES

[1] Gorrini, V., Bersini, H.: Recurrent fuzzy systems. 3rd IEEE International Conference on Fuzzy Systems, pp. 193-198 vol.191 (1994)

[2] Zhang, J., Morris, A.J.: Recurrent neuro-fuzzy networks for nonlinear process modeling. IEEE T. Neural Networ. 10, 313-326 (1999)

[3] Lee, C.-H., Teng, C.-C.: Identification and control of dynamic systems using recurrent fuzzy neural networks. IEEE T. Fuzzy Syst. 8, 349-366 (2000)

[4] Surmann, H., Maniadakis, M.: Learning feed-forward and recurrent fuzzy systems: A genetic approach. J. Syst. Architect. 47, 649-662 (2001)

[5] Juang, C.-F.: A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms. IEEE T. Fuzzy Syst. 10, 155-170 (2002)

[6] Wang, Y.-C., Chien, C.-J., Teng, C.-C.: Direct adaptive iterative learning control of nonlinear systems using an output-recurrent fuzzy neural network. IEEE T. Syst. Man. Cy. B 34, 1348-1359 (2004)

[7] Lin, C.-J., Chen, C.-H.: Identification and prediction using recurrent compensatory neuro-fuzzy systems. Fuzzy Set. Syst. 150, 307-330 (2005)

[8] Juang, C.-F., Genetic recurrent fuzzy system by coevolutionary computation with divide-and-conquer technique. IEEE T. Syst. Man. Cy. C 35, 249-254 (2005)

[9] Gonzalez-Olvera, M.A., Tang, Y.: A new recurrent neurofuzzy network for identification of dynamic systems. Fuzzy Set. Syst. 158, 1023-1035 (2007)

[10] Savran, A.: An adaptive recurrent fuzzy system for nonlinear identification. Appl. Soft. Comput. 7, 593-600 (2007)

[11] Juang, C.-F., Chung, I.F., Hsu, C.-H.: Automatic construction of feedforward/recurrent fuzzy systems by clustering-aided simplex particle swarm optimization. Fuzzy Set. Syst. 158, 1979-1996 (2007)

[12] Gama, C.A., Evsukoff, A.G., Weber, P., Ebecken, N.F.F.: Parameter identification of recurrent fuzzy systems with fuzzy finite-state automata representation. IEEE T. Fuzzy Syst. 16, 213-224 (2008)

[13] Aliev, R.A., Guirimov, B.G., Fazlollahi, B., Aliev, R.R.: Evolutionary algorithm-based learning of fuzzy neural networks. Part 2: Recurrent fuzzy neural networks. Fuzzy Set. Syst. 160, 2553-2566 (2009)

[14] Juang, C.-F., Hsieh, C.-D.: A Locally Recurrent Fuzzy Neural Network With Support Vector Regression for Dynamic-System Modeling. IEEE T. Fuzzy Syst. 18, 261-273 (2010)

[15] Juang, C.-F., Lin, Y.-Y., Tu, C.-C.: A recurrent self-evolving fuzzy neural network with local feedbacks and its application to dynamic system processing. Fuzzy Set. Syst. 161, 2552-2568 (2010)

[16] Juang, C.-F., Lin, Y.-Y., Huang, R.-B.: Dynamic system modeling using a recurrent interval-valued fuzzy neural network and its hardware implementation. Fuzzy Set. Syst. 179, 83-99 (2011)

[17] Juang, C.-F., Chang, P.-H.: Recurrent fuzzy system design using elite-guided continuous ant colony optimization. Appl. Soft. Comput. 11, 2687-2697 (2011)

[18] Lin, Y.-Y., Chang, J.-Y., Lin, C.-T.: Identification and Prediction of Dynamic Systems Using an Interactively Recurrent Self-Evolving Fuzzy Neural Network. IEEE T. Neural Networ. 24, 310-321 (2013)

[19] Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. J. Global. Optim. 11, 341-359 (1997)

[20] Neri, F., Tirronen, V.: Recent advances in differential evolution: a survey and experimental analysis. Artif. Intell. Rev. 33, 61-106 (2010)

[21] Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-Art. IEEE T. Evolut. Comput. 15, 4-31 (2011)

[22] Tang, H., Xue, S., Fan, C.: Differential evolution strategy for structural system identification. Comput. Struct. 86, 2004-2012 (2008)

[23] Subudhi, B., Jena, D.: A differential evolution based neural network approach to nonlinear system identification. Appl. Soft. Comput. 11, 861-871 (2011)

[24] Subudhi, B., Jena, D.: Nonlinear system identification using memetic differential evolution trained neural networks. Neurocomputing 74, 1696-1709 (2011)