ABCM Symposium Series in Mechatronics - Vol. 3 - pp.746-753

IDENTIFICATION OF AN EXPERIMENTAL YO-YO MOTION CONTROL SYSTEM BY EVOLUTIONARY B-SPLINE NEURAL NETWORK

Leandro dos Santos Coelho
Industrial and Systems Engineering Graduate Program, PPGEPS
Pontifical Catholic University of Paraná
Rua Imaculada Conceição, 1155
80215-901, Curitiba, Paraná, Brazil
leandro.coelho@pucpr.br

Abstract*. In an attempt to accurately model nonlinear systems, a wide variety of techniques have been developed, such as the Volterra series, Wiener models, Hammerstein models, and others. Such approaches have had limited success in industry, due primarily to their complexity. Recently, artificial neural networks have generated considerable interest as alternative nonlinear modeling tool. B-spline neural network (BSNN), a type of basis function neural network, is trained by gradient-based methods, which may fall into local minimum during the learning procedure. To overcome the problems encountered by the conventional learning methods, differential evolution (DE) — an evolutionary computation methodology — can provide a stochastic search to adjust the control points of a BSNN is proposed. The potentialities of DE are its simple structure, easy use, convergence property, quality of solution and robustness. In this paper, we propose a DE approach to train a BSNN. The numerical results presented here indicate that the DE is effective in building a good BSNN model for nonlinear identification of an experimental nonlinear yo-yo motion control system.*

Keywords*: B-spline neural network, nonlinear identification, differential evolution, yo-yo motion control system.*

1. INTRODUCTION

Artificial neural networks (or neural networks) are originally inspired by biologic neural networks' functionality that may learn complex functional relations through a limited number of training data. Artificial neural networks may serve as black-box models of nonlinear multivariable dynamic systems and may be trained using process measured data (Mcloone *et al*., 1998; Narendra and Parthasarathy, 1990). The usual neural network consists of multiple simple processing elements, called neurons, interconnections among them and the weights attributed to the interconnections. The relevant information of such methodology is stored in the weights.

In recent years, researchers (Liu *et al*., 1998; Sarimveis *et al*., 2003; Kukojl and Levi, 2004) have proposed a varied spectrum of methodologies for identification and nonlinear forecasting based upon neural networks to deal with nonlinear dynamic systems.

A relevant approach is to find the best approximation with respect to certain class of basis functions for neural networks representation. In this case, there are many possible choices of basis functions, such as radial basis function, associate memory networks, wavelets, and B-spline. The main advantage of the B-spline functions over other radial functions e.g., the Bezier curve, is the local control of the curve shape, as the curve only changes in the vicinity of a few control points that have been changed (Newmann and Sproull, 1979). A B-spline neural network (BSNN) consists of the piecewise polynomials with a set of local basis functions to model an unknown function for which a finite set of input-output samples are available. The performance of the identification depends on an optimization algorithm for the training procedure of the BSNN in order to avoid any possible local minima and also of the BSNN configuration.

In this context, the development of training methods and improvements for BSNN is an emergent research area. Several metaheuristics have been developed in recent years to improve the performance and set up the parameters of the BSSN design and also fuzzy systems approaches (Chu and Tomizuka, 1995; Logghe and Wang, 1997; Starrenburg *et al*., 1996; Saranli and Baykal, 1998; Ruano and Azevedo, 1999; Zhang and Knoll, 1999; Yiu *et al*., 2001; Reay, 2003).

Recently, as an alternative to the conventional mathematical approaches based on gradient information (Chan *et al*., 1998), modern bio-inspired optimization techniques such as evolutionary algorithms (Wang *et al*., 2002; Coelho and Krohling, 2006) paradigms have received much attention by many researchers due to their ability to find an almost global optimal solution.

During the past decade, a novel evolutionary algorithm, Differential Evolution (DE), has been proposed and attracted increasing attention (Storn and Price, 1995, 1997). Unlike the simple genetic algorithms that use binary coding for representation, candidate solutions in DE are represented as individuals based on floating-point numbers. DE is firstly initialized with a population of random solutions. At each generation, the target population is perturbed with a mutant factor, and the crossover operator is then introduced to combine the mutated population with the target population so as to generate a trial population. Then the selection operator is applied to compare the objective values of both competing populations, namely, target and trial populations. The better individuals of the two populations become members of the population for the next generation. This process is repeated until a predefined stopping condition occurs. Due to its simple concept, easy implementation and quick convergence, DE has been successfully applied to a variety of unconstrained continuous optimization problems (Huang *et al*., 2007).

In this paper, a DE approach to train a BSNN is proposed. Numerical results for identification of the nonlinear dynamics of an experimental yo-yo motion system show the feasibility and effectiveness of the proposed approach.

The remainder of this paper is organized as follows. In section 2, the fundamentals of BSNN are presented, while section 3 explains the concepts of DE optimization method. Section 4 presents the simulation results for identification of experimental nonlinear yo-yo motion control system. Lastly, section 5 outlines our conclusion and future research.

2. B-SPLINE NEURAL NETWORK

BSNN is introduced as a class of one-hidden-layer feedforward neural networks composed of B-spline functions. Each basis function is composed of q polynomial segments. There exists a simple and stable recursive relationship for evaluating the membership of a B-spline basis function of order k,

$$N_q^j(x) = \left(\frac{x-\lambda_{j-q}}{\lambda_{j-1}-\lambda_{j-q}}\right) N_{q-1}^{j-1}(x) + \left(\frac{\lambda_j - x}{\lambda_j - \lambda_{j-q+1}}\right) N_{q-1}^{j}(x) \tag{1}$$

$$N_1^j(x) = \begin{cases} 1 & if\,(x \in I_j) \\ 0 & otherwise \end{cases} \tag{2}$$

where $N_q^j(\cdot)$ is defined as the j-th univariate basis function of order q and λ_j the j-th knot and I_j is the j-th interval. The output of neural network is given by

$$\hat{o}_k = f(x_k) = \sum_{j=1}^{p} w_j N_q^j(x_k) \tag{3}$$

where x_k and $\hat{o}_k$ are the inputs and output of network, respectively, w_j is the weight attached to the j-th basis function and $N_q^j(\cdot)$ is given by the recursive form (2). The index j is associative with the region of local support, $\lambda_{(j-q)} \le x \le \lambda_{(j)}$, where the index q indicates the order of the basis functions.

The quality of approximation depends on the placement of knots of B-spline functions. The objective of optimization of BSNNs by DE is determination of the knots of each B-spline basis functions. In particular, the number of basis functions was determined by trials in this work.

3. DIFFERENTIAL EVOLUTION APPROACH FOR BSNN TRAINING

Unlike other evolutionary algorithms, differential evolution (DE) does not make use of some probability distribution function in order to introduce variations into the population. Instead, DE uses the differences between randomly selected vectors (individuals) as the source of random variations for a third vector (individual), referred to as the target vector. Trial solutions are generated by adding weighted difference vectors to the target vector. This process is referred to as the mutation operator where the target vector is mutated. A recombination (or crossover) operator is then applied to produce an offspring which is only accepted if it improves on the fitness of the parent individual (Salman *et al*., 2006).

The variant implemented here of DE was the DE/*rand*/1/*bin*. The pseudo code of DE used in the present study is given in Fig. 1 (Angira and Babu, 2006).

Let P a population of size NP, x^j the j-th individual of dimension D in population P

Input of parameters setup

Dimension, D;

Bounds of optimization variables (potential solutions), x: $lower\ (x_i)$ and $upper\ (x_i)$, $i = 1,\ldots, D$

Population size, $NP \geq 4$;

Mutation factor, f_m;

Crossover rate, CR;

Stopping criterion of maximum number of generations, t_{max};

Generation counter $t = 1$;

Step 1: Initialize the population $P = \{ x^1, x^2, x^3, \ldots, x^{NP}\}$ using uniformly distributed random numbers $rand_i$ in range [0,1] as

For each individual $j \in P$

$x_i^j(t) = lower\ (x_i) + rand_i \cdot \{ upper\ (x_i) - lower\ (x_i) \}$, $i = 1,\ldots, D$

End For each

Step 2: Evaluate the fitness value f_j, $j = 1,\ldots, NP$, of each individual in population P.

Step 3: While the stopping criterion t_{max} is not satisfied do

For All $j \leq NP$

Randomly select $r_1, r_2, r_3 \in (1,\ldots, NP)$, $j \neq r_1 \neq r_2 \neq r_3$

Randomly select $i_{rand} \in (1,\ldots, D)$

For All $i \leq D$

$x_i^{'}(t) = x_i^{r3}(t) + f_m \cdot \{ x_i^{r1}(t) - x_i^{r2}(t) \}$ if ($random$[0,1) < CR or $i = i_{rand}$

$x_i^{'}(t) = x_i^j(t)$ otherwise

End For All i

If $f(x_i^{'}(t)) \leq f(x_i^j(t))$ Then $x_i^j(t) = x_i^{'}(t)$; $f(x_i^j(t)) = f(x_i^{'}(t))$;

End For All j

Update the generation counter, $t = t + 1$;

End While

Figure 1. Pseudo code of DE.

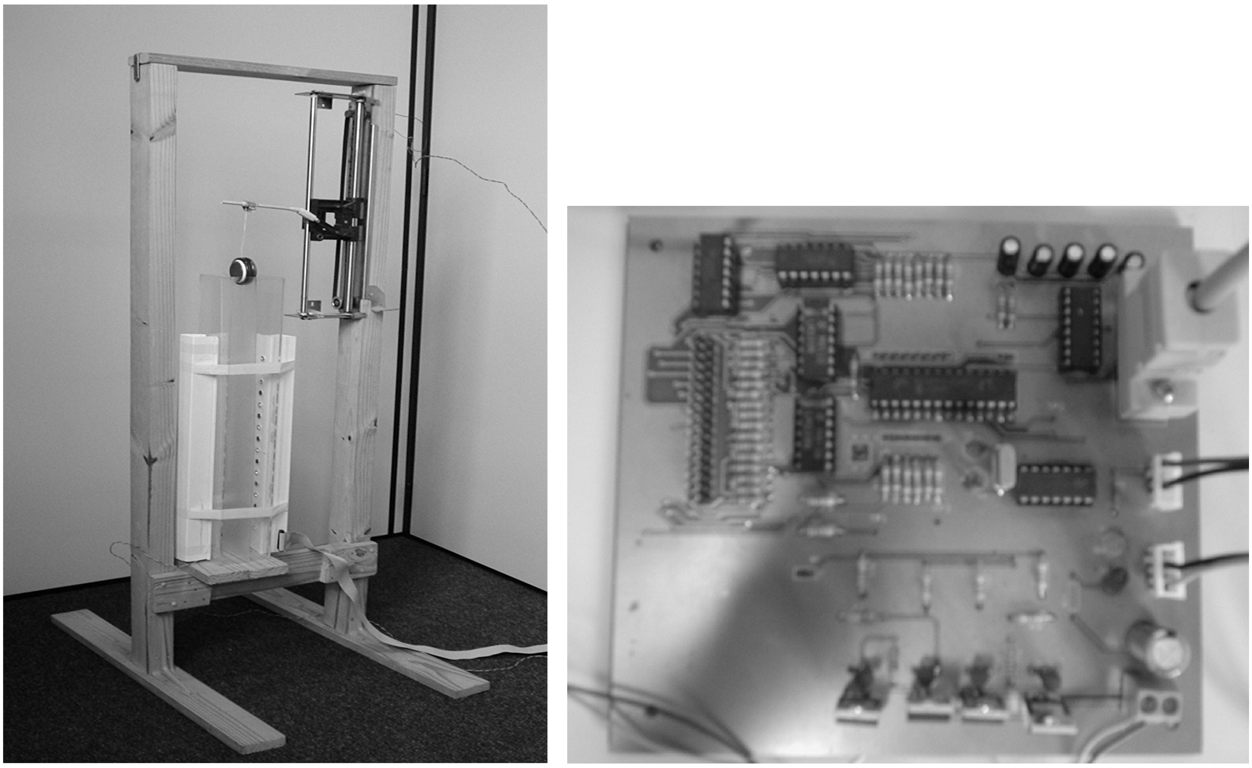
4. DESCRIPTION OF THE EXPERIMENTAL APPARATUS AND IDENTIFICATION RESULTS

4.1. Description of the Yo-Yo Motion Control System

Yo-yo playing is considered a representative example of open-loop unstable control problems that involve intermittent dynamic environments. Stable control of yo-yo playing relies on a proper phase relationship between the controller's action and the motion of the yo-yo (Jin and Zacksenhouse, 2003).

The development of automatic control systems that efficiently control a yo-yo represents a significant challenge for the development of electromechanical designs (Hashimoto and Toshiro, 1996; Zlajpah and Nemec, 2003). One of the main difficulties is the lack of sensors to obtain the motion measure of the toys. Another difficulty is the lack of mathematical models of this measurement device type, which justifies the use of the BSNNs to identify the dynamic behavior of a yo-yo motion in a real system.

The control system prototype employed in this work uses a yo-yo, and a Direct Current (DC) motor for its motion presents nonlinearity and complex behavior. A block diagram of the described system and a photograph of the system are presented in Fig. 2 (Herrera *et al*., 2006).



(a) yo-yo motion system (b) sensor module

Fig. 2. Photograph of yo-yo motion system and its devices.

The components of this prototype are divided into software and hardware modules, where (Coelho and Herrera, 2007):

① *Control module (software)*: consists of the implementation of control techniques, such as PID (proportional-integral-derivative) control, fuzzy logic control, and PI (Proportional-Integral) adaptive controllers integrated into a computer with communication with the yo-yo system using an I/O interface;

② *Sensor module (hardware / firmware)*: the sensors employed include the digital electronic circuits (power amplification), A/D and D/A converters, and the micro controller running firmware;

③ *Actuator module (hardware / firmware)*: consists of DC motors integrated to the *Sensor module*, electronic circuits and micro controller running firmware;

④ *Sensor sub-module*: made up of 16 infrared LEDs (Light-Emitting Diodes) able to inform the position of the yo-yo.

The prototype modules are composed of hardware and firmware and are connected to the same printed circuit board, called the control board. The control board contains two hardware modules and communicates with a personal computer through the RS-232 I/O interface. All the components used for the yo-yo system are off the shelf items to keep the cost minimal.

4.2. Simulation Results of Nonlinear Identification using BSNN and DE optimization

The mathematical model employed in this work to represent the yo-yo motion system is a NARX (Nonlinear AutoRegressive with eXogenous inputs). In this case, the NARX model with series-parallel conception is used for one-step-ahead prediction of the BSNN system.

A computer with a data acquisition board for generating the control signal (identification in closed-loop using a proportional controller design) and position value of the yo-yo was used to obtain system measurements. In the identification procedure based on the BSNN model, 290 samples of input (tension applied to the DC motor) and output (position of yo-yo) were collected with a time sampling of 40ms (see Fig. 3). The tension value corresponds to the maximum value configuration of the driver in PWM (Pulse Width Modulation) control of a DC motor.

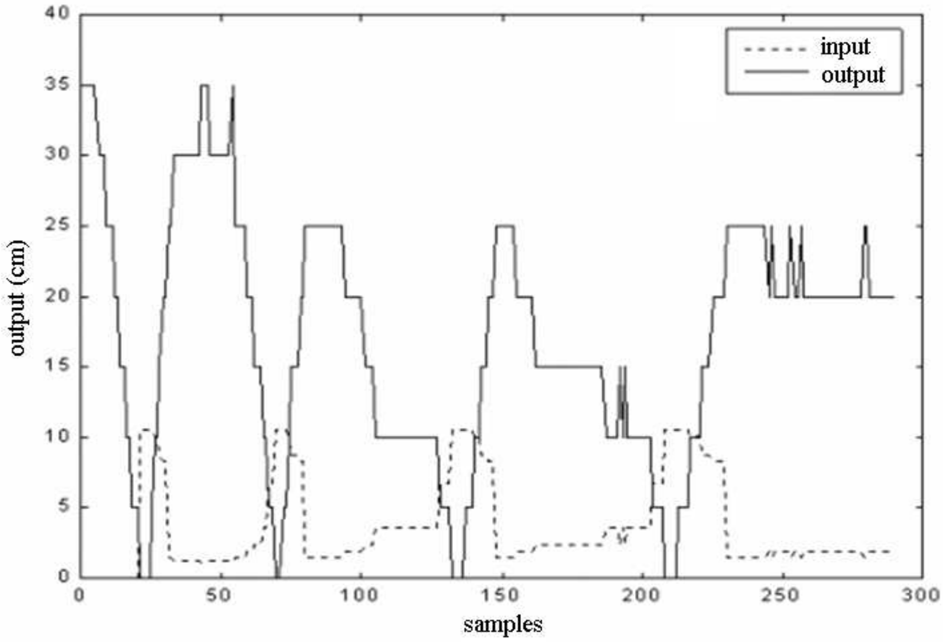


Fig. 3. Yo-yo motion system input and output data.

Experiments for the estimation phase (training phase) of the mathematical model of the yo-yo motion system are carried out using samples 1 to 150. For the validation phase, the BSNN model uses the input and output signals of samples 151 to 290. The system identification by BSNN model based on DE optimization is appropriate if a performance index is in permissible values for the user's needs. In this case, the fitness function for maximization proposes using DE and is given by the harmonic mean of multiple correlation indices of estimation (training) and validation phases. The fitness function (to be maximized) is calculated using the expression of R_{est}^2 given by:

$$R_{est}^2 = 1 - \frac{\sum_{t=1}^{150} [y(t) - \hat{y}(t)]^2}{\sum_{t=1}^{150} [y(t) - \bar{y}]^2} \tag{4}$$

where R_{est}^2 is the multiple correlation index of the estimation phase, $y(t)$ is the output of the real system, $\hat{y}(t)$ is the output estimated by the BSNN, and $\bar{y}$ is the mean value of the system's output. For the validation phase (verification of generalization capability) of optimized BSSN is employed the R_{val}^2 index give by

$$R_{val}^2 = 1 - \frac{\sum_{t=151}^{290} [y(t) - \hat{y}(t)]^2}{\sum_{t=151}^{290} [y(t) - \bar{y}]^2} \tag{5}$$

where R_{val}^2 is the multiple correlation index of the validation phase. When the value R^2 =1.0 (estimation or validation phases), it indicates an overfitting phenomenon, i.e. a model error exists. A R^2 value between 0.9 and 1.0 is considered sufficient for applications in designs of identification and model-based controller.

All the computational programs were run on a 3.2 GHz Pentium IV processor with 3 MB of RAM. In each case study, 50 independent runs were made for each of the optimization methods involving 50 different initial trial solutions for each optimization method. The setup of classical DE approaches used was the following:

- DE(1): classical DE using a constant mutation factor given by f_m = 0.4 and a crossover rate of CR = 0.8;

- DE(2): DE using a linear reduction of f_m with initial and final values of 0.8 and 0.3, respectively;
- DE(3): DE using a linear increasing of f_m with initial and final values of 0.3 and 0.8, respectively.
- DE(4): DE using f_m generated randomly with uniform distribution in range [0.4; 1.0];

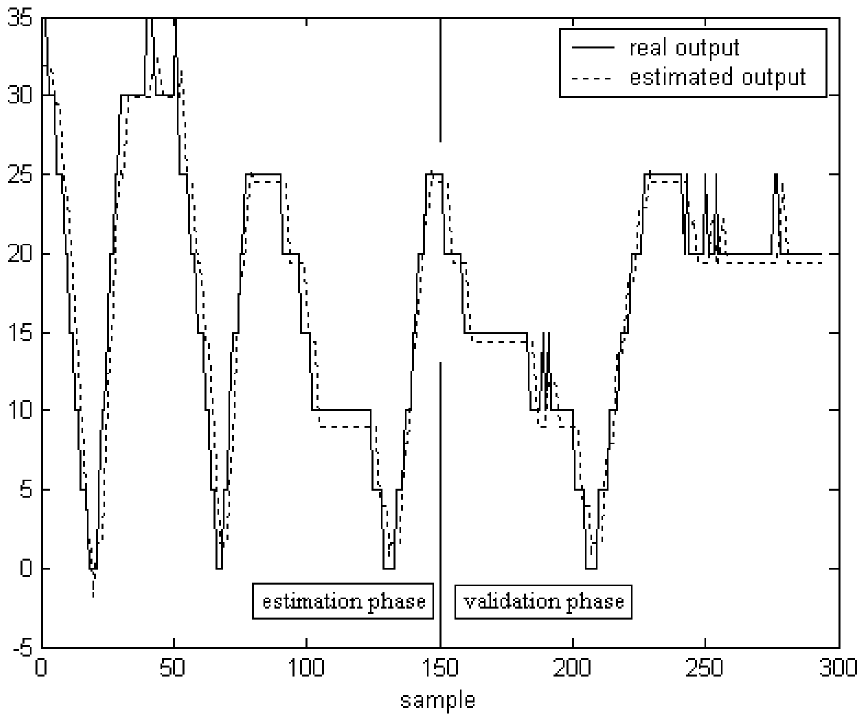
In these case studies, the population size *NP* was 20 and the stopping criterion t_{max} was 50 generations for the DE approaches. The three chosen vectors of BSNN's input were [$u(t-1)$; $y(t-2)$; $y(t-1)$]. The space searches for knots of each B-spline basis functions are [-1.0; 1.0]. Simulation tests using 5 knots in each input of BSNN were realized.

Table 1 presents the simulation results (best of 50 experiments with 50 generations for each run) for DE in optimization of BSNN using 5 knots. As indicated in Table 1, the results of the optimized BSNN present precision and provide an appropriate experimental mathematical model for the yo-yo motion system.

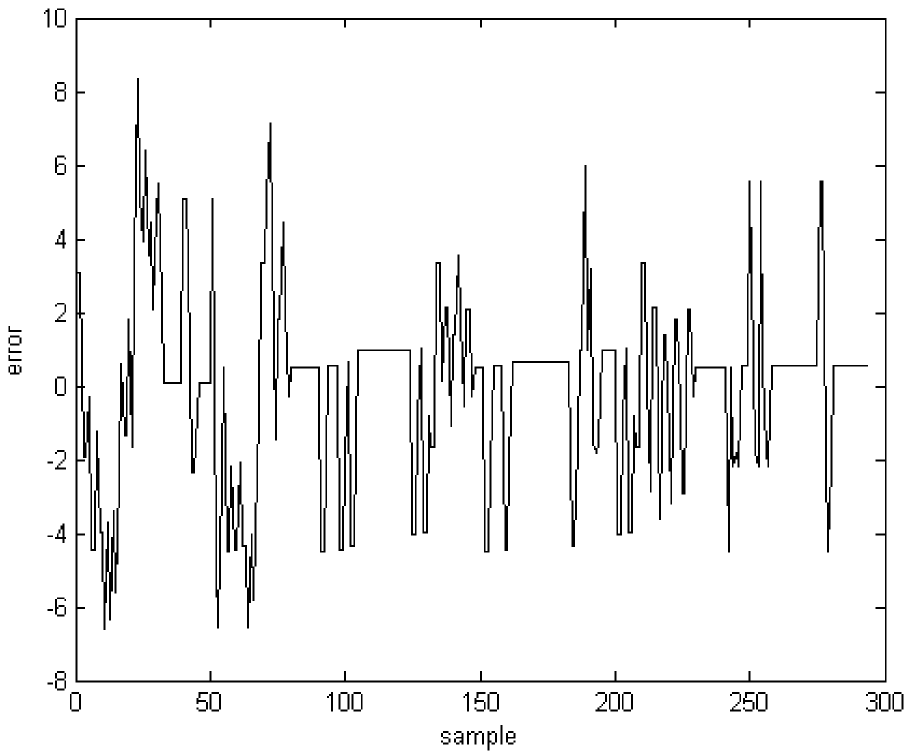
Table 1. Results obtained in estimation and validation phases by the maximization of R^2_{est} using DE approaches (it is adopted the solution vector with best R^2_{est} after the accomplishment of 50 runs)

Optimization Method	R^2_{est}			
DE type	maximum	mean	minimum	standard deviation
DE(1)	0.9503	**0.9442**	0.8237	0.0213
DE(2)	0.9524	0.9412	0.8237	0.0237
DE(3)	**0.9525**	0.9429	0.8237	0.0214
DE(4)	0.9515	0.9413	0.8237	0.0220

For the case study of the BSNN optimization, there is a consistent performance pattern across tested approaches with similar results in terms of R^2_{est}. The DE(1) presents the best mean of fitness function as shown in Table 1. However, the DE(3) approach presents better results in relation to the maximum fitness than the classical DE(1), DE(2) and DE(4). The best results shown in Figure 3 represent the BSNN (using DE(3)) with 5 knots for each network input.



(a) real (process) output and estimated output



(b) error signal

Fig. 4. Best result of R^2_{est} using BSNN with 5 knots for each input and optimization based on DE(3) (see Table 1).

5. CONCLUSION AND FUTURE RESEARCH

Many researches combining neural networks have been developed to improve the efficiency of nonlinear system identification. Traditionally, neural networks are trained by using gradient-based methods, which may fall into a local minimum during the learning process. Unfortunately, such techniques also suffer from difficulties, such as, the choice of starting guess and convergence.

DE is an evolutionary algorithm that uses a rather greedy and less stochastic approach to problem solving than do the EAs. DE combines simple arithmetical operators with the classical operators of recombination, mutation and selection to evolve from a randomly generated starting population to a final solution. In this work, DE optimization method to adjust the control points of a BSNN was proposed.

The preliminary experimental results indicate that the DE approaches presented very nice performances in identification of an experimental nonlinear yo-yo motion control system.

The future work of this study includes a comparative study between the DEC approaches and conventional training methods like Levenberg-Marquardt, gradient descent, Kalman filter, and quasi-Newton in BSNN training.

6. ACKNOWLEDGEMENTS

This work was supported by the National Council of Scientific and Technologic Development of Brazil — CNPq — under Grant 309646/2006-5/PQ. The author would also like to thank the reviewers for their insightful comments and suggestions.

7. REFERENCES

Angira, R. and Babu, B.V., 2006, “Optimization of Process Synthesis and Design Problems: A Modified Differential Evolution Approach”, Chemical Engineering Science, Vol. 61, pp. 4707-4721.

Chan, C.W., Jin, H., Cheung, K.C. and Zhang, H.Y., 1998, “An Adaptive Learning Rate for the Training of B-spline Networks”, Proceedings of UKACC International Conference on Control, Swansea, UK, pp. 342-347.

Chu, V.K. and Tomizuka, M., 1995, “Rule Generation for Fuzzy Systems Based on B-splines”, Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, pp. 6098-611.

Coelho, L.S. and Herrera, B.M., 2007, "Fuzzy Identification Based on a Chaotic Particle Swarm Optimization Approach Applied to a Nonlinear Yo-yo Motion System", IEEE Transactions on Industrial Electronics (accepted to future publication).

Coelho, L.S. and Krohling, R.A., 2006, "Nonlinear System Identification Based on B-spline Neural Network and Modified Particle Swarm Optimization", Proceedings of IEEE World Conference on Computational Intelligence (WCCI), International Joint Conference on Neural Networks, Vancouver, BC, Canada, pp. 7139-7144.

Hashimoto, K. and Toshiro, N., 1996, "Modeling and Control of Robotic Yoyo with Visual Feedback", IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, USA, pp. 2650-2655.

Herrera, B.M., Ribas, L. and Coelho, L. S., 2006, "Nonlinear Identification of a Yo-Yo System Using Fuzzy Model and Fast Particle Swarm Optimization", Applied Soft Computing Technologies: The Challenge of Complexity, Abraham, A., De Baets, B., Köppen, M. and Nickolay, B. (Org.). London, UK, Springer, pp. 302-316.

Huang, F.-Z., Wang, L. and He, Qie, 2007, "An Effective Co-Evolutionary Differential Evolution for Constrained Optimization", Applied Mathematics and Computation, Vol. 186, pp. 340-356.

Jin, H.-L. and Zacksenhouse, M., 2003, "Oscillatory Neural Networks for Robotic Yo-Yo Control", IEEE Transactions on Neural Networks, Vol. 14, No. 2, pp. 317-325.

Kukolj, D. and Levi, E. Levi, 2004, "Identification of Complex Systems Based on Neural and Takagi-Sugeno Fuzzy Model", IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, Vol. 34, No. 1, pp. 272-82.

Liu, G.P., Kadirkamanathan, V. and Billings, S.A., 1998, "On-line Identification of Nonlinear Systems Using Volterra Polynomial Basis Function Neural Networks", Neural Networks, Vol. 11, pp. 1645-1657.

Logghe, D. and Wang, H., 1997, "Modelling a Non-Linear pH Process via the Use of B-splines Neural Network", Proceedings of the IEEE International Conference on Control Applications, Hartford, CA, USA, pp. 401-407.

Mcloone, S., Brown, M.D., Irwin, G. and Lightbody, A., 1998, "A Hybrid Linear/Nonlinear Training Algorithm for Feedforward Neural Networks", IEEE Transactions on Neural Networks, Vol. 9, No. 4, pp. 669-684.

Narendra, K.S. and Parthasarathy, K., 1990, "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Transactions on Neural Networks, Vol. 1, No. 1, pp. 4-27.

Newmann, W.M. and Sproull, R.F., 1979, "Principles of Interactive Computer Graphics", Mc-Graw-Hill, Inc, USA.

Reay, D.S., 2003, "CMAC and B-spline Neural Networks Applied to Switched Reluctance Motor Torque Estimation and Control", Proceedings of 29th Annual Conference of the IEEE Industrial Electronics Society, Roanoke, VI, USA, pp. 2453-2458.

Ruano, A.E. and Azevedo, A. B., 1999, "B-Splines Neural Networks Assisted PID Autoning", International Journal of Adaptive Control & Signal Processing, Vol. 13, No. 4, pp. 291-307.

Salman, A., Engelbrecht, A.P. and Omran, M.G.H., 2006, "Empirical Analysis of Self-Adaptive Differential Evolution", European Journal of Operational Research (accepted to future publication).

Saranli, A. and Baykal, B., 1998, "Complexity Reduction in Radial Basis Function (RBF) Networks by Using Radial B-spline Functions", Neurocomputing, Vol. 18, pp. 183-194.

Sarimveis, H., Alexandridis, A. and Bafas, G., 2003, "A Fast Training Algorithm for RBF Networks Based on Subtractive Clustering", Neurocomputing, Vol. 51, pp. 501-505.

Shimojima, K., Fukuda, T. and Arai, F., 1994, "Self-Tuning Fuzzy Inference Based on Spline Function", Proceedings of IEEE Int. Conference on Fuzzy Systems, Orlando, FL, USA, pp. 690-695.

Starrenburg, G., Van Luenen, W.T.C., Oelen, W. and Van Amerongen, J., 1996, "Learning Feedforward Controller for a Mobile Robot Vehicle", Control Engineering Practice, Vol. 4, No. 9, pp. 1221-1230.

Storn, R. and Price, K., 1995, "Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces", Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA, USA.

Storn, R., and Price, K., 1997, "Differential Evolution — A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization, Vol. 11, No. 4, pp. 341-359.

Wang,W.-Y., Lee, T.-T., Hsu, C.-C. And Li, Y.-H., 2002, "GA-Based Learning of BMF Fuzzy-Neural Network, Proceedings of International Conference on Fuzzy Systems, Honolulu, HI, USA, pp. 1234-1238.

Yiu, K.F.C., Wang, S., Teo, K.L. and Tsoi, A.C., 2001, "Nonlinear System Modeling via Knot-Optimizing B-spline Networks", IEEE Transactions on Neural Networks, Vol. 12, No. 4, pp. 1013-1022.

Zhang, J. and Knoll, A., 1999, "Designing Fuzzy Controllers by Rapid Learning", Fuzzy Sets and Systems, Vol. 101, pp. 287-301.

Zlajpah, L. and Nemec, B., 2003, "Control Strategy for Robotic Yo-Yo", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, Nevada, USA, pp. 767-772.

8. RESPONSIBILITY NOTICE

The author is the only responsible for the printed material included in this paper.