COMPRASION FOR VOLTERRA SYSTEM IDENTIFICATION

Hasan Zorlu[1] Şaban Özer[2] Şeref Sağıroğlu[3]
e-mail:hzorlu@erciyes.edu.tr *e-mail:sozer@erciyes.edu.tr* *e-mail:ss@erciyes.edu.tr*
[1,2]Department of Electronics Engineering, Erciyes University, 38039, Melikgazi, Kayseri, Tırkey.
[3]Department of Computer Engineering, Erciyes University, 38039, Melikgazi, Kayseri, Turkey.

Key words: Artificial neural network, Fuzzy inference system, Recursive least square method

ABSTRACT

This paper introduces a comprehensive comparison for Volterra system identification. Volterra recursive least square, neural models trained with eight different learning algorithms and four fuzzy inference systems were compared in terms of accuracy and convergence speed. Two Volterra systems were tested with noise and noiseless cases. The simulation results have shown that classical method is successful in noiseless case. When the noisy is considered neural models have found most successful identification tool.

I. INTRODUCTION

The Volterra model has a firm mathematical function [1-4] and has been successfully used to solve many problems in science and engineering, such as communication channel equalization, echo cancellation, characterization of semiconductor devices, distortion analysis of audio components, modelling nonlinear phenomenon in random seas, identification of inherently nonlinear system for control and many other areas [5-11].

The utilization of the Volterra model in nonlinear system identification and analysis has become widespread in recent years [1-2]. This is perhaps due to generality and mathematical tractability of the Volterra model. The Volterra model is general nonlinear model in the sense that many nonlinear systems of engineering interest can be appropriately approximated by a truncated Volterra series [1,2,12-14]. It is also mathematically tractable in the sense that the Volterra kernels (or transfer function) of a nonlinear system can be determined based on the higher order statistics of the input and output of the system [3]. One of the major tasks in Volterra modelling of nonlinear systems is to determine the Volterra kernels (in the time domain) or the Volterra transfer function (in the frequency domain). In literature, many methods are developed for determination of the Volterra transfer functions [15-18]. Early works on Volterra filters were based on the least mean square (LMS) algorithm [19-20]. Another alternative method is to use recursive least squares (RLS) algorithms. Recently, Lee and Mathews presented a fast transversal algorithm for RLS adaptive filtering [21].

The Volterra series approach has received the most attention in the literature [18-23].The Volterra series expansion offers a useful parameterization of arbitrary analytic nonlinear systems. Such systems with finite memory “N+1" can be approximated by a truncated Volterra series as follows:

$$x_n = h_0 + \sum_{i=0}^{N} h_i u_{n-i} + \sum_{i=0}^{N}\sum_{j=0}^{N} q_{i,j} u_{n-i} u_{n-j} + \sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{k=0}^{N} q_{i,j,k} u_{n-i} u_{n-j} u_{n-k} + \dots \quad (1)$$

where x_n is the output, u_n is the input, h is the linear parameter and q is the nonlinear parameter.

Learning and generalization ability, providing plausible solution to nonlinear problem, fast real time operation and ease of implementation features have made artificial neural networks popular in system identification as well [25,37]. Fuzzy inference systems (FIS) [34-36] are nonlinear systems capable of inferring complex nonlinear relationships between input and output variables. The system can learn the nonlinear mapping by being presented a sequence of input signal and desired response pairs, which are used in conjunction with an optimization algorithm to determine the values of the system parameters.

In this work, the neural and FIS models were used in identifying Volterra systems. The performances of these models were also compared with RLS method. Neural models trained with Broyden-Fletcher-Goldfarb-Shanno (BFGS), Powell-Beale conjugate gradient (PB), Fletcher-Powell conjugate gradient (FP), Polak-Ribiere conjugate gradient (PR), standard backpropagation (BP), Levenberg-Marquardt (LM), one step secant (OSS) and resilient propagation (RP) learning algorithms. FIS models are the first order Sugeno type fuzzy system with Gaussian curve (GSMF), generalized bell curve (GBMF), trapezoidal (TPMF) and triangular (TRMF) membership functions.

II. CLASSICAL METHODS

Many methods have been developed for Volterra system identification [15-23]. Adaptive Volterra RLS filters can be developed using linear techniques by embedding the nonlinear problem into a linear multivariate problem More details can be found in [23]. Computational complexity of the algorithm can be especially prohibitive for large observation window sizes or high filter orders. One way of reducing the complexity of the algorithm is to introduce a partial decoupling of the sets of filter weights. This leads to both a reduction in computational complexity and more rapid convergence. This algorithm requires fewer computations per iteration to be implemented than the standard LMS algorithm. It is the advantage of the partially decoupled RLS algorithm. Because of this, RLS algorithm is used in this work.

III. ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANNs) are applied in many areas because of their learning ability, ease of implementation, parallel processing and real-time realization. There are many available ANN structures in the literature [25]. Multilayered perceptrons (MLPs) are the simplest and therefore most commonly used neural network architectures. In this work, they have been adapted for modeling the Volterra systems. MLPs can be trained using many different learning algorithms [25-33]. Backpropagation (BP) is a gradient descent method and the most commonly adopted MLP training algorithm [26]. It has a local minima problem. Resilient propagation (RP) algorithm [27] generally provides faster convergence than most other algorithms and the role of the RP is to avoid the bad influence of the size of the partial derivative on the weight update. Conjugate gradients (CGS) are alternative approach as to speed up the training process as of MLPs. These algorithms are second order methods which restrict each step direction to be conjugate to all previous step directions. This restriction simplifies the computation greatly because it is no longer necessary to store or calculate the Hessian or its inverse. There are a number of versions of CGs. The Polak-Ribiere (PR) version of CG is said to be faster and more accurate than Fletcher-Reeves (FR) because PR makes more complex assumptions. Powell-Beale (PB) version is the recent one [28-29]. Broydon-Fletcher-Goldfarb-Shanno (BFGS) method uses an update formula derived from the quasi-Newton update of Hessian [30]. The inverse Hessian is taken as the identity matrix at each step so that the matrix is never stored explicitly. Levenberg-Marquardt (LM) is a least-squares estimation method based on the maximum neighbourhood idea [31-32]. The LM combines the best features of the Gauss-Newton technique and the steepest-descent method, but avoids many of their limitations. In particular, it generally does not suffer from the problem of slow convergence. One step secant (OSS) method is between CGS and BFGS [33]. The Hessian matrix in the previous iteration is assumed to be a unit matrix. With the help of this, the inverse matrix is not calculated.

IV. FUZZY INFERENCE SYSTEM

Fuzzy inference system (FIS) is a popular computing framework based on the concepts of fuzzy set theory, fuzzy if-then rules and fuzzy reasoning [35]. Basically, a fuzzy inference system is composed of fuzzification, fuzzy rule base, fuzzy inference system and defuzzification blocks. A number of learning algorithms used in FIS are available in the literature [34-36]. These learning algorithms can be used to construct FISs with different properties and characteristics. Some of these algorithms are data intensive, recursive (thus giving the FIS an adaptive nature), offline, and application specific. Some of they have computational complexity. In FIS design, it is important to determine the number of membership functions (MFs) or rules necessary to adequately represent a given system. Given an initial set of membership functions, one has to select the best possible subset of membership functions for an effective representation.

V. SIMULATIONS and RESULTS

Two examples of Volterra systems are tested for comparison. First example is a second-order Volterra system [2] and is given as

System#1

$$y(n)= -0.64u(n) + u(n-2) +0.9u^2(n) + u^2(n-1) \qquad (2)$$

Second example is a fourth-order Volterra system and is given as

System#2

$$y(n) = 2u(n-6) + 5u^2(n-6) + 3.4u(n-6)u(n-7) - 4u^3(n-6) - 3u^4(n-6) \qquad (3)$$

These systems are tested for noisy and noiseless observations. In the simulations, for the both test systems, only one input signal used was a zero-mean with Gaussian with unit-variance. These both systems were also tested for the noiseless and the signal to noise ratio (SNR) of 25 dB cases. 750 training data sets and 250 test data sets were used for neural and FIS models. The results obtained from noisy and noiseless simulations were shown in Tables 1, 2, and 3 for the system#1 and system#2. In these tables, the computation time and root mean squared (RMS) errors obtained from the methods were also illustrated for accurate comparison.

For system#1, FIS model is a first order Sugeno type fuzzy system with 4-inputs and 1-output. Each input has 2 membership functions and the output has a linear membership function. Neural model has 4 neurons in input layer, 8 neurons in each hidden layer and 1 neuron in output layer. The input and the output neurons have linear transfer functions. Each hidden layer has a hyperbolic tangent or sigmoid transfer function.

For system#2, FIS model is a first order Sugeno type fuzzy system with 5-inputs and 1-output. Each input has 2 membership functions and the output has a linear membership function. Neural model has 5 neurons in input layer, 10 neurons in each hidden layer and 1 neuron in output layer. Transfer functions are the same order as System#1.

VI. CONCLUSIONS

This study presents a comprehensive comparison of Volterra system identification using classic and different intelligent techniques. The performances of the methods were evaluated for noisy and noiseless cases on the base of computer simulations. For noiseless case, RLS algorithm was found the best method for the both systems. For noisy observation, ANN trained with LM was found the best method for among all. When computation time is considered in general, the fastest method is RLS. ANN models followed this. FISs requested largest computation time. As can be seen from tables intelligent techniques provide more accuracy when the noisy Volterra systems are identified. It should be emphasized that ANN models and FIS models require large time in training but the response of this models are in millisecond levels.

Table 1: RLS method identification results

Method	System#1				System#2			
	Noiseless case		Noisy case		Noiseless case		Noisy case	
	RMS error	Time(s)	RMS error	Time(s)	RMS error	Time(s)	RMS error	Time(s)
RLS	$5.3x10^{-13}$	6.590	0.2860	6.930	$6.3x10^{-12}$	8.250	0.4832	8.870

Table 2: Neural model identification results

Systems	ANN Algorithm	Noiseless case			Noisy case			Epoch
		RMS error		Training Time (s)	RMS error		Training Time (s)	
		Train	Test		Train	Test		
System#1	BFGS	$1.2x10^{-5}$	$4.0x10^{-5}$	227.95	$1.2x10^{-4}$	0.0453	245.49	600
	FR	0.0014	0.0056	87.160	0.0049	0.0501	92.270	300
	PB	0.0024	0.0039	60.190	0.0027	0.0472	63.000	300
	PR	0.0025	0.0069	72.770	0.0054	0.0486	79.980	300
	BP	0.0349	0.0464	676.18	0.0382	0.0591	690.58	6000
	LM	$4.3x10^{-6}$	$6.3x10^{-5}$	313.68	$4.8x10^{-8}$	0.0422	364.07	400
	OSS	$5.5x10^{-4}$	0.0019	437.31	$7.6x10^{-4}$	0.0445	452.75	2000
	RP	$4.9x10^{-4}$	0.0014	585.72	0.0013	0.0473	594.56	5000
System#2	BFGS	$4.8x10^{-5}$	$1.4x10^{-4}$	483.180	0.0294	0.0303	495.170	600
	FR	0.0067	0.0020	155.550	0.0318	0.0311	180.300	300
	PB	$8.2x10^{-4}$	0.0062	93.3800	0.0316	0.0330	111.060	300
	PR	$5.7x10^{-4}$	0.0028	107.050	0.0369	0.0354	122.650	300
	BP	0.0291	0.0114	1005.47	0.0413	0.0310	1020.14	6000
	LM	$8.4x10^{-6}$	$2.0x10^{-6}$	998.600	0.0298	0.0282	1035.87	400
	OSS	0.0028	0.0021	719.580	0.0310	0.0310	750.920	2000
	RP	$7.6x10^{-4}$	0.0010	1068.19	0.0518	0.0302	1083.81	5000

Table 3: FIS model identification results

Systems	FIS Membership Function	Noiseless case			Noisy case			Epoch
		RMS error		Training Time (s)	RMS error		Training Time (s)	
		Train	Test		Train	Test		
System#1	GSMF	$1.7x10^{-8}$	$1.9x10^{-8}$	661.69	$1.6x10^{-8}$	0.0435	670.97	250
	GBMF	$7.1x10^{-7}$	$7.6x10^{-7}$	599.24	$1.6x10^{-8}$	0.0470	592.26	250
	TPMF	$8.8x10^{-6}$	0.0066	563.59	$1.4x10^{-6}$	0.0716	580.84	250
	TRMF	$2.9x10^{-6}$	0.0049	587.60	$6.2x10^{-7}$	0.0446	570.79	250
System#2	GSMF	$8.9x10^{-6}$	$2.6x10^{-5}$	4653.18	0.0292	0.0444	4714.190	250
	GBMF	$9.6x10^{-6}$	$1.4x10^{-4}$	4359.660	0.0291	0.0413	4670.97	250
	TPMF	$4.9x10^{-5}$	$1.7x10^{-4}$	4412.05	0.0290	0.0396	4581.83	250
	TRMF	$3.9x10^{-6}$	$6.7x10^{-5}$	4608.03	0.0297	0.0336	4466.70	250

REFERENCES

1. M. Schetzen, The Volterra and Wiener Theories of Nonlinear System, New York: Wiley, 1980.
2. C. H. Tseng, A Mixed-Domain Method for Identification of Quadratically Nonlinear Systems, IEEE Trans. Signal Processing, Vol. 45, pp. 1013-1024, 1997.
3. C. L. Nikias, M. R. Raghuver, Bispectrum estimation: A Digital Signal Processing Framework, Proc. IEEE, Vol. 75, pp.869-891, 1987.
4. I. W. Sandberg, The Methematical foundations of Associated Expansions for Mildly Nonlinear Systems, IEEE Trans. Circuit Sys., Vol. CAS-30, pp. 441-455, 1983.
5. S. Benedetto, E. Biglieri, Nonlinear Equalization of Digital Satellite Channels, IEEE J. Select. Areas Commun., Vol. 40, pp.1438-1442, 1992.
6. O. Agazzi, D. G. Messerschmitt, Nonlinear Echo Cancellation of Data Signals, IEEE Trans. Commun., Vol. COMM-30, pp. 2421-2433, 1982.
7. W. Reiss, Nonlinear Distortion Analysis of p-i-n Diode Attenuators Using Volterra Series Representation, IEEE Trans. Circuit Sys., Vol. 31, pp. 535-542, 1984.
8. Y.S Cho, S. B. Kim, E. J. Powers, A Digital Technique to Estimate Second-Order Distortion Using Higher Order Coherence Spectra, IEEE Trans. Signal Processing, Vol. 40, pp. 1029-1040, 1992.
9. D. W. Choi, R. W. Miksad, E. J. Powers, F. J. Fisher, Application of Digital Cross-Bispectral Analysis Techniques to Model the Nonlinear Response of Moored Vessel System in Random Seas, J. Sound Vibr., Vol. 99, no. 3, pp. 309-326, 1985.
10. M. Kortmann, K. Janiszowski, H. Unbehauen, Application and Comprasion of Different Identification Schemes Under Industrial Conditions, Int. J. Control, Vol. 48, no. 6, pp.2275-2296, 1988
11. S. Ozgunel, A. H. Kayran, E. Panayirci, Nonlinear Channel Equalization and Identification, Proc. Of Int. Conf. on Digital Signal Proc., Florence, Italy, pp. 260-265, 1991
12. N. Kalouptsidis, Signal Processing Systems: Theory and Design, New York:Wiley, 1997.
13. I. Pitas and A. N. Venetsanopoulos, Nonlinear Digital Filters-Principles and Applications, Boston, MA:Kluwer, 1990.
14. G-O. A. Glentis, P. Koukoulas, N. Kalouptsidis, Efficient Algorithms for Volterra System Identification, IEEE Trans. Signal Processing, vol. 47, pp. 3042-3057, 1999.
15. R. Harris, D. Charbies, and F. Bishop, A Variable Step (VS) Adaptive Filter Algorithm, IEEE Trans. on Acoustics, Speech, Signal Processing, vol.ASSP-34, no.2, pp.309-316, 1986.
16. S. Hurram, Nonlinear Adaptive Filters for System Modelling, Ph.D. Thesis, Boston U., 1994.
17. E. Mumolo, A. Carini, On The Stability Of Discrete Time Volterra Filters, IEEE Signal Processing Letters, vol.6, no.9, pp.230-232, 1999.
18. X. Li, W.K. Jenkins, C.W. Therrien, A Computationally Efficient Algorithm For Adaptive Quadratic Volterra Filters, IEEE Int. Sympo. on Circuits and Systems, Hong Kong, pp.2184-2187, 1997.
19. T. Koh, E.J. Powers, Second-Order Volterra Filtering and Its Application to Nonlinear System Identification, IEEE Trans. on Acoustics, Speech, Signal Processing, vol.ASSP-33, pp.1445-1455, 1985.
20. D.W. Griffith, G. R. Arce, Partially Decoupled Volterra Filters: Formulation and LMS Adaptation, IEEE Trans. On Signal Proc., Vol. 45, No. 6, pp. 1485-1494, 1997
21. J. Lee, V. J. Mathews, A Fast Recursive Least Squares Adaptive Second-Order Volterra Filter and Its Performance Analysis, IEEE Trans. Signal Proc., Vol. 41, pp. 1087-1101, 1993.
22. M. T. Ozden, A. H. Kayran, E. Panayirci, Adaptive Volterra Filtering with Complete Lattice Orthogonalization, IEEE Trans. On Signal Proc., Vol. 44, No. 8, pp. 2092-2098, 1996
23. D.W. Griffith, G. R. Arce, A Partially Decoupled RLS Algorithm for Volterra Filters, IEEE Trans. On Signal Proc., Vol. 47, pp. 579-582, 1999
24. F.Rauf, H.M.Ahmed, New Nonlinear Adaptive Filters With Applications To Chaos, Int. Jour, Of Bifur. and Chaos, vol.7, no.8, pp.1791-1809, 1997.
25. S. Haykin, Neural Networks: A Comprehensive Foundation, New York: Macmillan College Publishing Company, 1994.
26. D. E. Rumelhart, J. L. McClelland, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol.1, Cambridge, MA:MIT Pres., 1986.
27. M. Riedmiller, H. Braun, A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm, Proceedings of the IEEE Int. Conf. On Neural Networks, San Francisco, CA, pp.586-591, 1993.
28. M. J. D. Powell, Restart Procedures for the Conjugate Gradient Method, Mathematical Programming, Vol.12, pp.241-254, 1977.
29. L. E. Scales, Introduction to Non-Linear Optimization, New York: Springer-Verlag, 1985.
30. P.E. Gill, W. Murray, M. H. Wright, Practical Optimization, New York: Academic Pres., 1981.
31. K. Levenberg, A Method For the Solution of Certain Nonlinear Problems in Least Squares, Quart. Appl. Math., Vol.2, pp.164-168, 1944.
32. D. W. Marquardt, An Algorithm For Least-Squares Estimation of Nonlinear Parameters, J. Soc. Ind. Appl. Math., Vol.11, pp.431-441, 1963.
33. R. Battiti, First and Second Order Methods for Learning: Between Steepest Descent and Newton's

Method, Neural Computation, Vol.4, No.2, pp.141-166, 1962.
34. L. X. Wang, Adaptive Fuzzy Systems and Control, Englewood-Cliffs, NJ: Prentice-Hall., 1994.
35. R.J. Jang and C. Sun, Neuro-Fuzzy Modelling and Control, Proc.IEEE, Vol. 83, pp. 378-406, 1995.
36. J.M. Mendel, G.C. Mouzouris, Designing Fuzzy Logic Systems, IEEE Trans. On Circ. and Syst.-II: Analog and Digital Signal Processing, Vol. 44, pp. 885-895, 1997.
37. K. S. Narendra, K. Parthasarathy, Identification and Control of Dynamical Systems Using Neural Networks IEEE Trans. on Neural Networks, vol 1 pp 4-27, 1990