An Approach for Time Series Forecasting by simulating Stochastic Processes through time lagged feed-forward neural network

C. Rodríguez Rivero[1], J. Pucheta[1], J. Baumgartner[1], H.D. Patiño[2] and B. Kuchen[2]
[1]Departments of Electrical and Electronic Engineering, Laboratory of Research in Applied Mathematics Control (LIMAC), at Faculty of Exact, Physical and Natural Sciences - National University of Córdoba, Córdoba, Argentina.
[2]Institute of Automatics (INAUT), at Faculty of Engineering-National University of San Juan, San Juan, Argentina.

Abstract - *In this work an approach for time series forecasting by simulating stochastic processes through time lagged feed-forward neural network is presented. The learning rule used to adjust the neural network (NN) weights is based on the Levenberg-Marquardt method. In function of the long or short term stochastic dependence of the time series, an on-line heuristic law to set the training process and to modify the NN topology is employed. The NN output tends to approximate the current value available from the series by applying a time-delay operator. The approach is tested over samples of the Mackey-Glass delay differential equations (MG). Four sets of parameters for MG definition were used. The performance is shown by forecasting the 18 future values of four time series of 102 data length each. Each time series was simulated by a Monte Carlo of 50 trials at the final of each data series.*

Keywords: Neural networks, time series forecast, Hurst's parameter, Mackey-Glass.

1 Introduction

Natural phenomena prediction is a challenging topic, useful for control problems from agricultural activities. Before starting with the agriculture venture, the availability of estimated scenarios for water predictability would help the producer to decide. There are several approaches based on NN that face the rainfall forecast problem for energy demand purposes [5], for water availability and seedling growth [22] by taking an ensemble of measurement points [12], [14]. Here, the proposed approach is based on the classical NAR filter using time lagged feed-forward neural networks, where the data from the MG benchmark equation whose forecast is simulated by a Monte Carlo [4] approach. The number of filter parameters is put function of the roughness of the time series, in such a way that the error — between the smoothness of the time series data and the forecasted data, modifies the number of the filter parameters.

1.1 Overview of the MG equation

The MG equation serves to model natural phenomena and has been used by different authors to perform comparisons between different techniques for foretelling and regression models [7] [20]. Here we propose an algorithm to predict values of time series taken from the solution of the MG equation [9]. The MG equation is explained by the time delay differential equation defined as:

$$\dot{y}(t) = \frac{\alpha y(t-\tau)}{1+y^{c}(t-\tau)} - \beta y(t) \tag{1}$$

where α, β, and c are parameters and τ is the delay time. According as τ increases, the solution turns from periodic to chaotic. Equation (1) is solved by a standard fourth order Runge-Kutta integration step, and the series to forecast is formed by sampling values with a given time interval.
Thus, a time series with a random-like behavior is obtained, and the long-term behavior changes thoroughly by changing the initial conditions. Furthermore, by setting the parameter β ranging between 0.1 and 0.9 the stochastic dependence of the deterministic time series obtained varies according to its roughness.

1.2 Overview of the NN Approach

One of the motivations for this study follows the closed-loop control scheme [16] where the controller considers meteorological future conditions for designing the control law as shown in Fig. 1. In that scheme the controller considers the actual state of the crop by a state observer and the meteorological variables, referred by $x(k)$ and R_o, respectively. However, in this paper only the controller's portion concerning with the prediction system is presented by using a benchmark time series. The controller design is inspired on the one presented in [16].

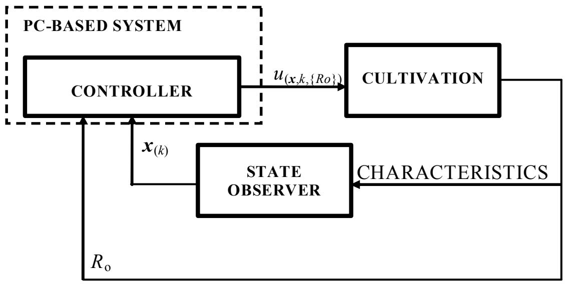

Fig. 1. Closed-loop PC-based control approach.

The main contribution of this work is in the learning process, which employs the Levenberg-Marquardt rule and considers the long or short term stochastic dependence of passed values of the time series to adjust at each time-stage the number of patterns, the number of iterations, and the length of the tapped-delay line, in function of the Hurst's value, *H* of the time series. According to the stochastic characteristics of each series, *H* can be greater or smaller than 0.5, which means that each series tends to present long or short term dependence, respectively. In order to adjust the design parameters and show the performance of the proposed prediction model, solutions of the MG equation are used. The NN-based nonlinear filter is applied to the time series obtained from MG to forecast the next 18 values out of a given historical data set of 102 values.

1.3 Overview on fractional Brownian motion

In this work the Hurst's parameter is used in the learning process to modify on-line the number of patterns, the number of iterations, and the number of filter's inputs. This *H* serves to have an idea of roughness of a signal, and to determine its stochastic dependence. The definition of the Hurst's parameter appears in the Brownian motion from generalizing the integral to a fractional one. The Fractional Brownian Motion (fBm) is defined in the pioneering work by Mandelbrot [13] through its stochastic representation:

$$B_H(t) = \frac{1}{\Gamma\left(H+\frac{1}{2}\right)}\left(\int_{-\infty}^{0}\left((t-s)^{H-\frac{1}{2}} - (-s)^{H-\frac{1}{2}}\right)dB(s) + \int_{0}^{t}(t-s)^{H-\frac{1}{2}}dB(s)\right) \quad (2)$$

where, $\Gamma(\cdot)$ represents the Gamma function

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}e^{-x}dx \quad (3)$$

and $0<H<1$ is called the Hurst parameter. The integrator *B* is a stochastic process, ordinary Brownian motion. Note, that *B* is recovered by taking $H=1/2$ in (2). Here, it is assumed that *B* is defined on some probability space (Ω, F, P), where Ω, *F* and *P* are the sample space, the sigma algebra (event space) and the probability measure respectively. Thus, an fBm is a time continuous Gaussian process depending on the so-called Hurst parameter $0<H<1$. The ordinary Brownian motion is generalized to $H=0.5$, and whose derivative is the white noise.

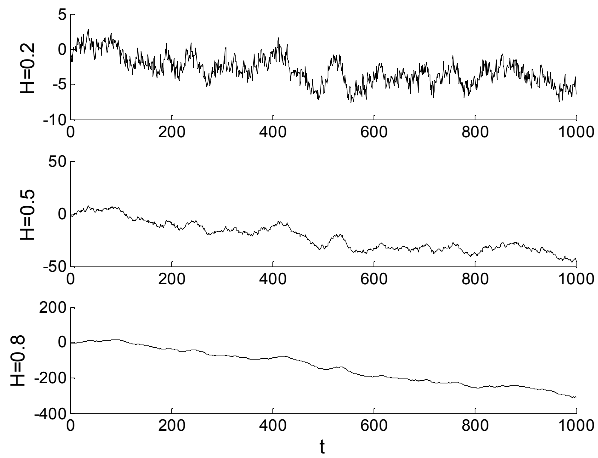

Fig. 2. Three sample path from fractional Brownian motion for three values of *H*.

The fBm is self-similar in distribution and the variance of the increments is defined by

$$Var\left(B_H(t) - B_H(s)\right) = v\left|t-s\right|^{2H} \quad (4)$$

where *v* is a positive constant.
This special form of the variance of the increments suggests various ways to estimate the parameter *H*. In fact, there are different methods for computing the parameter *H* associated to Brownian motion [2] [6] [11]. In this work, the algorithm uses a wavelet-based method for estimating *H* from a trace path of the fBm with parameter *H* [1] [6] [8]. Three trace paths from fBm with different values of *H* are shown in Fig. 2, where the difference in the velocity and the amount of its increments can be noted.

2 Problem statement

The classical prediction problem may be formulated as follow. Given past values of a process that are uniformly spaced in time, as shown by x(n-T), x(n-2T), . . . , x(n-mT), where T is the sampling period and m is the prediction order, it is desired to predict the present value x(n) of such process. Therefore, to obtain the best prediction of the present values from a random (or pseudo-random) time series is desired. The predictor system may be implemented using an autoregressive model-based nonlinear adaptive filter. The NNs are used as a nonlinear model building, in the sense that smaller the prediction error is (in a statistical sense), the better the NN serves as model of the underlying physical process responsible for generating the data. In this work, time lagged feed-forward neural networks are used. The present value of the time series is used as the desired response for the adaptive filter and the past values of the signal serve as input of the adaptive filter. Then, the adaptive filter output will be the one-step prediction signal. In Fig. 3 the block diagram of the nonlinear prediction scheme based on a NN filter is shown. Here, a prediction device is designed such that starting from a given sequence $\{x_n\}$ at time n corresponding to a time series it can be obtained the best prediction $\{x_e\}$ for the following sequence of 18 values. Hence, it is proposed a predictor filter with an input vector l_x, which is obtained by applying the delay operator, Z^{1}, to the sequence $\{x_n\}$. Then, the filter output will generate x_e as the next value, that will be equal to the present value x_n. So, the prediction error at time k can be evaluated as:

$$e(k) = x_n(k) - x_e(k) \quad (5)$$

which is used for the learning rule to adjust the NN weights.

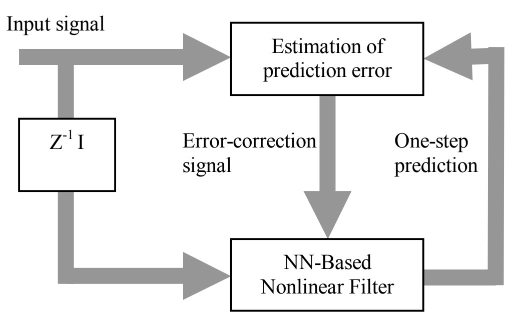

Fig. 3. Block diagram of the nonlinear prediction.

The coefficients of the nonlinear filter are adjusted on-line in the learning process, by considering a criterion that modifies at each pass of the time series the number of patterns, the number of iterations and the length of the tapped-delay line, in function of the Hurst's value H calculated from the time series. According to the stochastic behavior of the series, H can be greater or smaller than 0.5, which means that the series tends to present long or short term dependence, respectively [17].

3 Proposed approach

3.1 NN-Based NAR Model

Some results had been obtained from a linear autoregressive approach, which are detailed on [18]. These results were promising and deserve to be improved by more sophisticated filters. Here, a NN-based NAR filter model [10] [15] [21] is proposed. The NN used is a time lagged feed-forward networks type. The NN topology consists of l_x inputs, one hidden layer of H_o neurons, and one output neuron as shown Fig. 4. The learning rule used in the learning process is based on the Levenberg-Marquardt method [3]. However, if the time series is smooth or rough then the tuning algorithm may change in order to fit the time series. So, the learning rule modifies the number of patterns and the number of iterations at each time-stage according to the Hurst's parameter H, which gives short or long term dependence of the sequence $\{x_n\}$. From a practical standpoint, it gives the roughness of the time series.

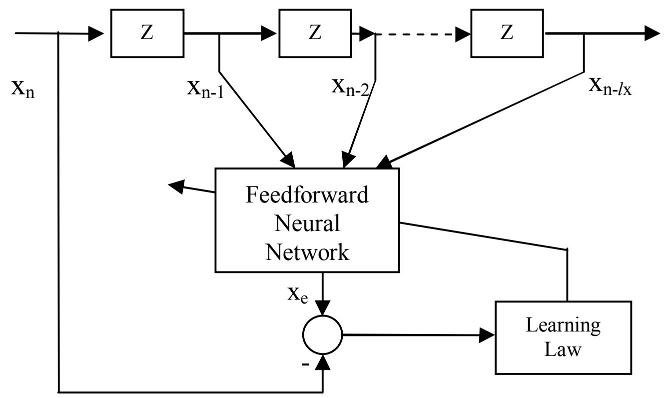

Fig. 4. Neural Network-based nonlinear predictor filter.

In order to predict the sequence $\{x_e\}$ one-step ahead, the first delay is taken from the tapped-line x_n and used as input. Therefore, the output prediction can be denoted by:

$$x_e(n+1) = \mathrm{F}_p\left(Z^{-1}I(\{x_n\})\right) \quad (6)$$

where F_p is the nonlinear predictor filter operator, and $x_e(n+1)$ is the output prediction at $n+1$.

3.2 The Proposed Learning Process

The NN's weights are tuned by means of the Levenberg-Marquardt rule, which considers the long or short term stochastic dependence of the time series measured by the Hurst's parameter H. The proposed learning approach consists of changing the number of patterns, the filter's length and the number of iterations in function of the parameter H for each corresponding time series. The learning process is performed using a batch model. In this case the weight updating is performed after the presentation of all training examples, which forms an epoch. The pairs of the used input-output patterns are

$$(x_i, y_i) \quad \mathrm{i} = 1,2,....,N_p \quad (7)$$

where, x_i and y_i are the corresponding input and output pattern respectively, and N_p is the number of input-output patterns presented at each epoch. Here, the input vector is defined as:

$$X_i = \mathrm{Z}^{-1}\mathrm{I}(\{x_i\}) \quad (8)$$

and its corresponding output vector as:

$$Y_i = x_i. \quad (9)$$

Furthermore, the index i is within the range of N_p given by

$$l_x \le N_p \le 2 \cdot l_x \tag{10}$$

where l_x is the dimension of the input vector.

In addition, the number of iterations performed by each epoch it is given by

$$l_x \le i_t \le 2 \cdot l_x. \tag{11}$$

The proposed criterion to modify the pair (i_t,N_p) is given by the statistical dependence of the time series $\{x_n\}$, supposing that it is an fBm. The dependence is evaluated by the Hurst's parameter H, which is computed by a wavelet-based method [1] [8]. Then, a heuristic adjustment for the pair (i_t,N_p) in function of H according to the membership functions shown in Fig. 5 is proposed.

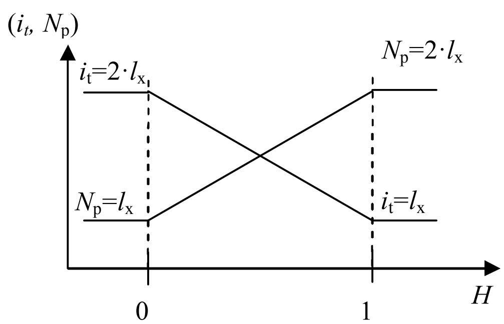

Fig. 5. Heuristic adjustment of (i_t,N_p) in terms of H after each epoch.

Finally, after each pass the number of inputs of the nonlinear filter is tuned —that is the length of tapped-delay line, according to the following heuristic criterion. After the training process is completed, both sequences —$\{x_n\}$ and $\{\{x_n\},\{x_e\}\}$— should have the same H parameter. If the error between $H(\{x_n\})$ and $H(\{\{x_n\},\{x_e\}\})$ is grater than a threshold parameter θ, the value of l_x is increased (or decreased) according to $l_x \pm 1$. Explicitly,

$$l_x = l_x + sign(\theta). \tag{12}$$

Here, the threshold θ was set about 1%.

4 Main results

4.1 Generations of data series from MG equations

Data time series are obtained from the MG equations (1) with the parameters shown in Table 1, with τ=100 and α=20. This collection of coefficients was chosen to generate time series whose H parameters vary between 0 and 1.

Series No.	β	H
1	0.5	0.62
2	0.75	0.72
3	0.8	0.47
4	0.85	0.26

Table 1. Parameters to generate the times series.

4.2 Set-up of Model and Learning Process

The initial conditions for the filter and learning algorithm are shown in Table 2. Note that the first number of hidden neurons and iteration are set in function of the input number. These initiatory conditions of the learning algorithm were used to forecast the time series, whose length is 102 values.

Variable	Initial Condition
l_x	*15*
H_o	$l_x/2$
i_t	l_x
H	*0.5*

Table 2. Initial conditions of the parameters.

4.3 Performance measure for forecasting

In order to test the proposed design procedure of the NN -based nonlinear predictor, an experiment with time series obtained from the MG solution was performed. The performance of the filter is evaluated using the Symmetric Mean Absolute Percent Error (SMAPE) proposed in most of metric evaluations, defined by

$$SMAPE_S = \frac{1}{n}\sum_{t=1}^{n}\frac{|X_t - F_t|}{(X_t + F_t)/2}\cdot 100 \tag{13}$$

where t is the observation time, n is the size of the test set, s is each time series, X_t and F_t are the actual and the forecasted time series values at time t respectively. The SMAPE of each series s calculates the symmetric absolute error in percent between the actual X_t and its corresponding forecast value F_t, across all observations t of the test set of size n for each time series s.

4.4 Prediction Results for the MG Time Series

Each time series is composed by sampling the MG solutions. However, there are two classes of data sets: one is used for the algorithm in order to give the forecast, which comprises 102 values. The other one is used to compare if the forecast is acceptable or not where the 18 last values can be used to validate the performance of the prediction system. Thus, 102 values form the data set, 120 values form the Forecasted and the Real ones. The Monte Carlo method was used to forecast the next 18 values. Here it was performed an ensemble of 50 trials with a Gaussian noise sequence of zero mean and variance of 0.08. Such outcomes are shown through Fig. 6 to Fig. 10 for each case respectively. In each figure, the legend “Forecasted” represents the obtained values by Eq. (6), the legend “Data” denotes the available data set, and the legend “Real” denotes the actual values -not available in practice- used here for verification purposes only. From time k equal 103 to 120, the inputs of Eq. (6) include the outputs delayed one time interval. The obtained time series has a mean value, which is indicated at the foot of the figure by “Forecasted Mean”. The “Real Mean” it is not available at time 102. This procedure is repeated 50 times for each time series shown in Fig. 6.

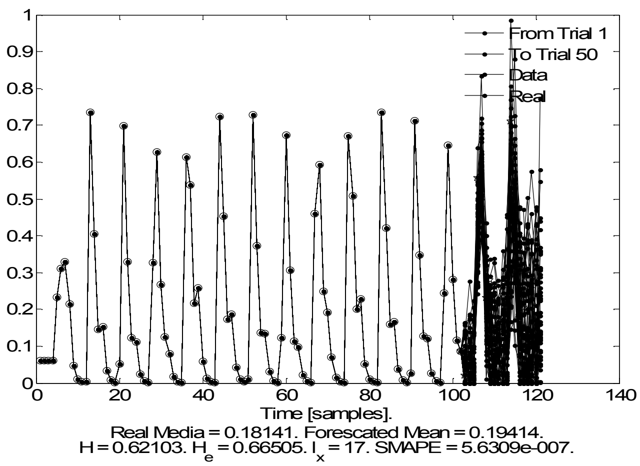

Fig. 6. Forecast for MG time series composed of 50 Trials.

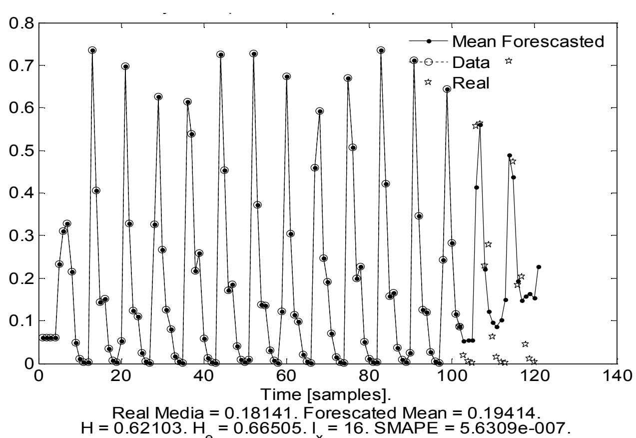

Fig.7. Forecast for MG time series where α=20, β=0.5, c=10 and τ=100.

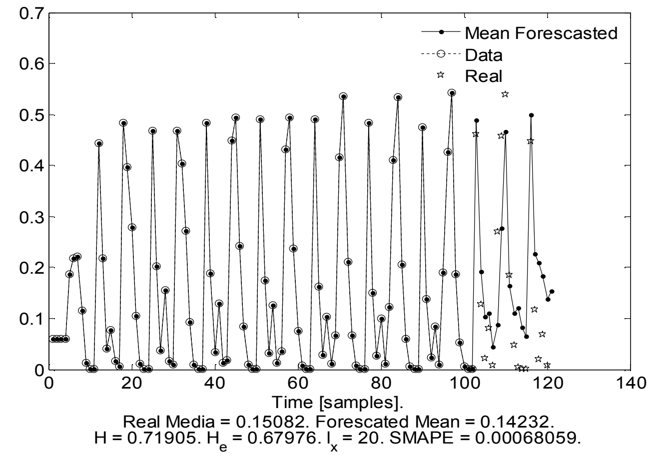

Fig. 8. Forecast for MG time series where α=20, β=0.75, c=10 and τ=100.

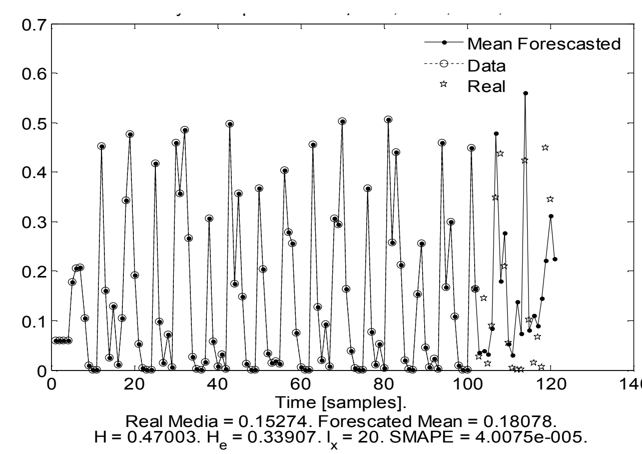

Fig. 9. Forecast for the first time where α=20, β=0.8, c=10 and τ=100.

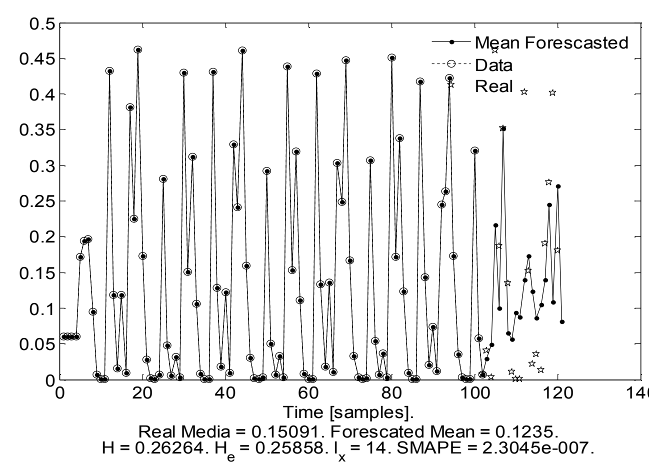

Fig. 10. Forecast for MG time series where α=20, β=0.85, c=10 and τ=100.

4.5 Comparative Results

The performance of the stochastic NN-based predictor filter is evaluated through the SMAPE index —Eq. (13), shown in Table 3 along the time series from MG solutions.

Series No.	H	H_e	Real mean	Mean Forecasted	SMAPE
1	0.621	0.665	0.181	0.194	5.63 10-7
2	0.719	0.679	0.150	0.142	6.8 10-4
3	0.47	0.339	0.152	0.18	4 10-5
4	0.262	0.258	0.15	0.123	2.34 10-7

Table 3. Figures obtained by the proposed approach.

The comparison between the deterministic approach [19] and the present forecasted time series is shown in Fig. 11. The SMAPE index for each time series is obtained by a deterministic NN-based filter, which uses the Levenberg–Marquardt algorithm with fixed parameters (i_t, N_p). In addition, the results of the SMAPE obtained by the stochastic NN-based filter proposed here are also shown in Fig. 11. Thus, the legend "Deterministic" refers to the first filter and the legend "Stochastic" refers to the second one. In Fig. 11 the values of SMAPE are indicated for both filters. The improvement can be noted since the SMAPE index diminishes from 0.21409 to 0.0019156, which means an improvement of 111 times averaging over the four time series. The performance of the first case -Fig. 6- was very close for both filters, so it was omitted in this comparative. This behavior is due to the fact that H>1, therefore the proposed algorithm here does not vary the pairs (i_t, N_p).

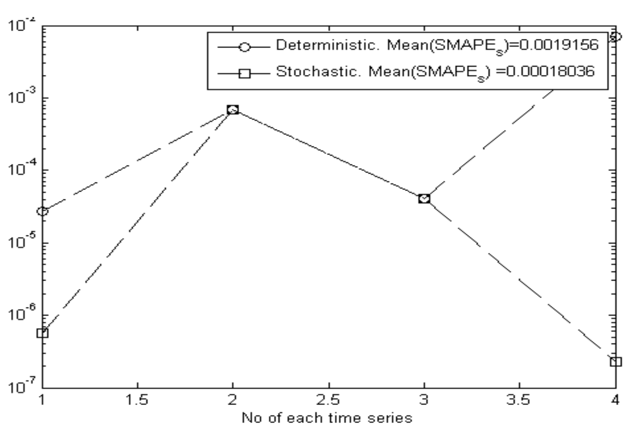

Fig. 11. The SMAPE index applied over the 4 time series by each algorithm.

5 Discussion

The assessments of the obtained results by comparing the performance of the proposed filter with the classic filter, both are based on NN. Although the difference between both filters resides only in the adjustment algorithm, the coefficients that each filter has, each ones performs different behaviors. In the four analyzed cases, the generation of 18 future values from 102 present values was made by each algorithm. The same initial parameters were used for each algorithm, but these parameters and the filter's structure are changed by the proposed algorithm that is not modified by the classic algorithm. The adjustment of the proposed filter, the coefficients and the structure of the filter are tuned by considering their stochastic dependency. It can be noted that in each of the figures —Fig. 6 to Fig. 10— the computed value of the Hurst's parameter is denoted either by H_e or H, both taken from the Forecasted time series or from the Data time series, respectively, since the Real time series (future time series) are unknown. Index SMAPE is computed between the complete Real time series (it includes the series Data) and the Forecasted one, as indicates the Eq. (13) for each filter. Note that there is no improvement of the forecast for any given time series, which results from the use of a stochastic characteristic to generate a deterministic result, such as a prediction.

6 Conclusions

In this work an approach for time series forecasting by simulating stochastic processes through time lagged feed-forward networks-based nonlinear autoregressive (NAR) model is presented. The learning rule proposed to adjust the NN's weights is based on the Levenberg-Marquardt method. Furthermore, in function of the long or short term stochastic dependence of the time series evaluated by the Hurst parameter H, an on-line heuristic adaptive law was proposed to update the NN topology at each time-stage. The major result shows that the predictor system supplied to time series has an optimal performance from several MG equations, owing to similar roughness between the original and the forecasted time series, which is evaluated by H and H_e, respectively. This fact encourages us to apply the proposed approach to meteorological time series when the observations are taken from a single point.

6.1 Acknowledgments

This work was supported by National University of Córdoba (Secyt UNC 69/08), National University of San Juan (UNSJ), National Agency for Scientific and Technological Promotion (ANPCyT) under grant PAV 076, PICT/04 No. 21592 and PICT-2007-00526.

7 References

[1] Abry, P.; P. Flandrin, M.S. Taqqu, D. Veitch., Self-similarity and long-range dependence through the wavelet lens. Theory and applications of long-range dependence, Birkhäuser, pp. 527-556. 2003.

[2] Bardet, J.-M.; G. Lang, G. Oppenheim, A. Philippe, S. Stoev, M.S. Taqqu. Semi-parametric estimation of the long-range dependence parameter: a survey. Theory and applications of long-range dependence, Birkhäuser, pp. 557-577. 2003.

[3] Bishop, C. Neural Networks for Pattern Recognition. Pp. 290 292. University Press. Oxford, 1995.

[4] Bishop, C. Pattern Recognition and Machine Learning. Springer. Boston, 2006.

[5] Chow, T.W.S.; Leung, C.T. Neural network based short-term load forecasting using weather compensation. Power Systems, IEEE Transactions on, Vol.11, Iss.4, Nov 1996, Pp. 1736-1742.

[6] Dieker, T. Simulation of fractional Brownian motion. MSc theses, University of Twente, Amsterdam, The Netherlands. 2004.

[7] Espinoza Contreras, Adriana Eliza. El Caos y la caracterización de series de tiempo a través de técnicas de la dinámica no lineal. Universidad Autónoma de Mexico. Campus Aragón. 2004.

[8] Flandrin, P. Wavelet analysis and synthesis of fractional Brownian motion" IEEE Trans. on Information Theory, 38, pp. 910-917. 1992.

[9] Glass L. and M. C. Mackey. From Clocks to Chaos, The Rhythms of Life. Princeton University Press, Princeton, NJ, 1988.

[10] Haykin, S. Neural Networks: A comprehensive Foudation. 2nd Edition, Prentice Hall. 1999.

[11] Istas, J.; G. Lang. Quadratic variations and estimation of the local Hölder index of a Gaussian process. Ann. Inst. Poincaré, 33, pp. 407-436. 1994.

[12] Liu, J.N.K.; Lee, R.S.T. Rainfall forecasting from multiple point sources using neural networks. In proc. of the International Conference on Systems, Man, and Cybernetics, Pages:429-434 Vol.3.1999.

[13] Mandelbrot, B. B. The Fractal Geometry of Nature, Freeman, San Francisco, CA. 1983.

[14] Masulli, F., Baratta, D., Cicione, G., Studer, L. Daily Rainfall Forecasting using an Ensemble Technique based on Singular Spectrum Analysis. In Proceedings of the International Joint Conference on Neural Networks IJCNN 01, pp. 263-268, vol. 1, IEEE, Piscataway, NJ, USA, 2001.

[15] Mozer, M. C. Neural Net Architectures for Temporal Sequence Processing. A. S. Weigend and N. A. Gershenfeld, eds. Time Series Predictions: Forecasting the Future and Understanding the Past. pp. 243-264. Reading, M.A.: Addison-Wesley. 1994.

[16] Pucheta, J., Patiño, H., Schugurensky, C., Fullana, R., Kuchen, B. Optimal Control Based-Neurocontroller to Guide the Crop Growth under Perturbations. Dynamics Of Continuous, Discrete And Impulsive Systems Special Volume Advances in Neural Networks-Theory and Applications. DCDIS A Supplement, Advances in Neural Networks, Watam Press, Vol. 14(S1), pp. 618—623. 2007.

[17] Pucheta, J., Patiño, H.D. and B. Kuchen, Neural Networks-Based Time Series Prediction Using Long and Short Term Dependence in the Learning Process. In proc. of the 2007 International Symposium on Forecasting, 24th to 27th of June 2007 Marriott Marquis Times Square, New York. 2007.

[18] Pucheta, J., Patino, D. and Kuchen, B. "A Statistically Dependent Approach For The Monthly Rainfall Forecast from One Point Observations". In IFIP International Federation for Information Processing Volume 294, Computer and Computing Technologies in Agriculture II, Volume 2, eds. D. Li, Z. Chunjiang, (Boston: Springer), pp. 787–798. (2009).

[19] Pucheta, J., Herrera, M., Salas C., Patiño, H.D., and B. Kuchen. "A Neural Network-Based Approach for Forecasting Time Series from Mackey-Glass Equations". In proc. Of the XIII Reunión de Trabajo en Procesamiento de la Información y Control ISBN 950-665-340-2. XII RPIC, organizado por el Laboratorio de Sistemas Dinámicos y Procesamiento de la Información, 16 al 18 de Setiembre de 2009 Rosario, Argentina. (2009).

[20] Velásquez Henao, Juan David, Dyna, Red. Pronóstico de la serie de Mackey glass usando modelos de regresión no-lineal. Universidad Autónoma de Mexico. Campus Aragón. 2004.

[21] Zhang, G.; B.E. Patuwo, and M. Y. Hu. Forecasting with artificial neural networks: The state of art. J. Int. Forecasting, vol. 14, pp. 35-62. 1998.

[22] Patino, H.D.; Pucheta, J.A.; Schugurensky, C.; Fullana, R.; Kuchen, B., "Approximate Optimal Control-Based Neurocontroller with a State Observation System for Seedlings Growth in Greenhouse," Approximate Dynamic Programming and Reinforcement Learning, 2007. ADPRL 2007. IEEE International Symposium on , vol., no., pp.318-323, ISBN: 1-4244-0706-0. 1-5 April 2007.