# Neurocomputation of the Hurst Parameter

Danielo G. Gomes*, Nelson L. S. da Fonseca &, Nazim Agoulmine*, and José N. de Souza#[1]

* Laboratory of Complex Systems – LSC, University of Evry, France
& Institute of Computing, State University of Campinas, Brazil
# Department of Electrical Engineering, University of Ceará, Brazil

***Abstract*- Network traffic presents long range dependences and is usually modeled either by fractal or by multifractal processes. The Hurst parameter is a measure of the self-similarity of a process. In this paper, a neuro estimator of the Hurst parameter is proposed and compared to statistical estimators**

## I. INTRODUCTION

Several studies have claimed that different types of network can be accurately modeled using a self-similar process. A self-similar process is able to capture the long-range dependence (LRD) phenomenon exhibited by such traffic. Moreover, studies have demonstrated that the long range dependencies may have a pervasive effect on queuing performance. In fact, there is clear evidence that it can potentially cause massive cell losses. Furthermore, such a queuing system suffers from the buffer inefficacy phenomenon. Increasing the buffer size is not effective for decreasing the buffer overflow probability significantly [1].

A self-similar process presents bursts in different time-scales, *i.e.*, when observing a self-similar process in different time-scales it is verified a similar pattern of the process samples [2]. The Hurst parameter or parameter $H$ characterizes the degree of self-similarity of a process degree. This parameter assumes values between 0 and 0.1. $H > 0.5$ indicates positive correlations, whereas $H < 0.5$ indicates a negative correlation. The closer to 1 the parameter value is, the stronger is the impact on queuing. Moreover, small variations of the value of $H$ may imply in a significant change on network resource utilization.

There are several statistical estimators for the Hurst parameter [3]. Some of them (such as the R/S statistic) are based on visual interpretation [3]. Others provide an estimation of $H$ with confidence intervals [4]. However, all these estimators need a large number of samples for an accurate analysis. On the other hand, in some cases, such as on-line video transmission, it is not possible to previously estimate the value of $H$. In this way, it is necessary to use estimators that can calculate in real-time so that any change in the Hurst parameter value can be immediately reported to the network controller.

A neural network is a system composed by a high number of simple processors (neurons or nodes), highly interconnected and based on a simplified model of a neuron. Neural networks can adapt the computation taking into account previous knowledge of solutions for the problem under investigation (training) [5] [6].

In this paper, a feedforward neural network with back propagation training is introduced for the estimation of the Hurst parameter of a traffic stream. The proposed approach is validated by comparing estimations given by it against estimations given by three statistical estimators: the R/S statistical, the Higuchi estimator [7] and the Abry-Veitch estimator [8] [9]. Results indicate that the neurocomputation approach provides reasonably accurate results and is proper for real time implementation.

The remainder of this paper is organized as follows. Section II provides some notions of the Hurst parameter. The statistical estimators are presented in Section III and the neuro estimator in Section IV. Numerical results are shown in Section V and the Section VI concludes the paper.

## II. THE HURST PARAMETER

It has been shown that different type of traffic present long-range dependencies, such as video, LAN and Internet traffic. Video and LAN traffic can be modeled as self-similar

[1] Danielo G. Gomes and Nazim Agoulmine are with the Laboratory of Complex Systems – LSC, University of Evry, 40, Rue du Pelvoux – CE 1455 Courcouronnes 91020 Evry Cedex, France, e-mail: {dgomes,nazim}@iup.univ-evry.fr, Nelson L. S. da Fonseca is with the Institute of Computing, State University of Campinas (UNICAMP), P.O. Box 6176 13084-971 Campinas, SP, Brazil, e-mail: nfonseca@ic.unicamp.br José N. de Souza is with the Department of electrical Engineering, University of Ceará – UFC, Campus do Pici – Bloco 705, 60455-760, Fortaleza, CE, Brazi, e-mail: neuman@ufc.br

processes, whereas Internet traffic can be modeled by multifractal processes. Multifractal processes can be represented by a series of "local" self-similar processes.

Let $x(t)$, with $t = 0, 1, 2, ...$, a stationary stochastic process [2]. For each $m = 1,2,...$, let $x^{(m)}(k)$, $k = 1,2,3,\ldots$, denote a new series obtained by averaging the original series $x(t)$ over non-overlapping blocks of size $m$.

A process $X$ is called *exactly second-order self-similar* with parameter $H = 1 - \beta/2$, $0 < \beta < 1$, if its autocorrelation function is [2]:

$$r^{(m)}(k) = \frac{1}{2}\left[(k+1)^{2-b} - 2k^{2-b} + (k-1)^{2-b}\right] \equiv g(k),\ 0 < b < 1,\ k = 1,2,... \quad (1)$$

and $X$ is called *asymptotically second-order self-similar* with parameter $H = 1 - \beta/2$, $0 < \beta < 1$, if for all $k = 1,2,\ldots$,

$$\lim_{m\to\infty} r^{(m)}(k) = \frac{1}{2}\left[(k+1)^{2-b} - 2k^{2-b} + (k-1)^{2-b}\right] \equiv y(k) \quad (2)$$

In self-similar processes, the autocorrelations decay hyperbolically implying in a non-summable autocorrelation function $\Sigma_k r(k) = \infty$ (*long-range dependences*),

The Hurst parameter ($H$) gives the degree of self-similarity of a process, and, consequently, expresses the pattern of dependencies of a process. If $0.5 < H < 1$, the process is a long-range Dependent (LRD) process. If $0 < H < 0.5$ it is an anti-persistence process, and if $H = 0.5$ it is a short-range dependent (SRD) process

Figure 1 illustrates the auto-correlation decay for different values of the Hurst parameter.

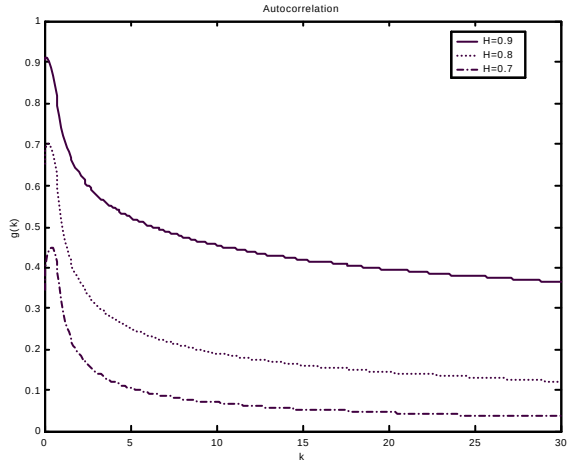


Fig. 1. Autocorrelation function (1) of an exact second-order self-similar with parameter $H = 1 - \beta/2$..

## III. STATISTICAL ESTIMATORS

Three statistical estimators were used to validate results given by the neuro estimator: the R/S statistical, the Higuchi method and the Abry-Veitch estimator.

### A. The R/S Statistical Estimator

The R/S estimator, defined by Hurst (1951), is one of the most well-known and simplest methods for estimation of dependence level of a series of samples [3].

For a stochastic process x(t) defined at discrete-time intervals $\{x_t, t = 0, 1, 2, ...\}$, the rescaled range of x(t) over a time interval $N$ is defined as the ratio R/S:

$$\frac{R}{S} = \frac{\max\limits_{1\le j\le N}\left[\sum_{k=1}^{j}(X_k - jM(N))\right] - \min\limits_{1\le j\le N}\left[\sum_{k=1}^{j}(X_k - jM(N))\right]}{\sqrt{\frac{1}{N}\sum_{j=1}^{N}(X_j - M(N))^2}} \quad (3)$$

with $M(N)$ being the sample mean over the time period $N$:

$$M(N) = \frac{1}{N}\sum_{j=1}^{N} X_j \quad (4)$$

*Let N* be the sample size divided in $K$ subsets where $K$ is an input parameter. If we plot log(R/S) versus $N$ on a log-log scale, then, the value of parameter $H$ can be estimated by linear regression over the $K$ points obtained from the data sets.

### B. The Higuchi Estimator

This method [7] considers the fractal dimension $D$ of a time series such that $H = 2+D$. The method takes the partial sums $Y(n) = \sum_{i=1}^{n} X_i$ of a random sample series $\{X_i\}$, i = 1,…,$N$ and it calculates the normalized size of the curve defined by the data as:

$$L(m) = \frac{N-1}{m^3}\cdot\sum_{i=1}^{m}\left\lfloor\frac{N-i}{m}\right\rfloor^{-1}\cdot\sum_{k=1}^{\lfloor(N-i)/m\rfloor}\left|Y(i+km) - Y(i+(k-1)m)\right|, \quad (5)$$

### C. The Abry-Veitch Estimator

The Abry-Veitch estimator is based on wavelets theory. It decomposes a sequence of samples in approach coefficients (low-pass filter) and detail coefficients (high-pass filter). These coefficients are obtained by projected digital filters. From the original sample sequence, successive approaches and detail sequences are calculated which are obtained by recursive digital filtering. In other words, the output of a digital filter is applied again (feedback) to the same digital filter. A detailed explanation of this method can be found in [8, 9]. The Abry-Veitch estimator is the most accurate H estimator known up to date and presents low computational complexity.

## IV. A NEURAL NETWORK ESTIMATOR

A neural network target to the solution of a specific problem requires training during which the network learns by adjusting the weights of the network connections. In fact, the

weights represent the knowledge of the neural network at the end of the training process, i.e., learning is a process in which the synaptic connections of the neural network are adapted by a continuous stimulus process from the environment where the network is inserted.

A feedforward network architecture with backpropagation momentum training algorithm was used. The backpropagation algorithm was considered since it is the most successful algorithm for the design of multilayer feedforward networks. The number of neurons in the input-output layers was defined according to the structure of the problem. The output variable is the parameter *H*, i.e., the neural network presents one neuron on the output layer.

Fractal Brownian Motion samples [9] [10] were generated to train the neural network and to generate the statistical estimator inputs. The three statistical estimators described before were used for the verification of the precision of the neural network. Results were derived using the Stuttgart Neural Network Simulator was used in the experiment [11].

Sequences of 10 traffic samples to compose the input variables. The reasons for this choice are:

(i). patterns appear at different levels of aggregation (self-similarity). Sequences of 10 samples have characteristics similar to sequences of 100, 1000 or 10000 samples;

(ii). training time is short with little neurons.

There is no established procedure of choice for the optimum number of neuron. Then, experiments with 2,5,10 15 and 20 neurons were tried and the better results were the ones with 15 hidden neurons (Table 2).

The patterns were chosen in the following way:

- 200 traces sequences with $H$=0.5;
- 100 traces sequences with $H$=0.6;
- 100 traces sequences with $H$=0.7;
- 100 traces sequences with $H$=0.8;
- 100 traces sequences with $H$=0.9;

Each sequence has 10 bursts of samples. For the trace with $H$=0.5, it was necessary to have twice the number of samples due to short range dependences.

After the learning phase the performance of the neural network was checked. These patterns should not have been presented to the network before.

For the definition of the number of neurons and of the activation function, the first training used a stopping criterion of 1000 epochs. Each epochs contains one complete cycle (sweeping) through all training pattern set This training allows to selection of the neural network topology which is more adapted to the problem. The second training considered another stopping criterion: a small error.

After the training phase, 600 sequences of the test set were applied to NN. These sequences are not known by the network since it is necessary to verify the generalization capacity of the neural network.

## V. RESULTS AND DISCUSSION

Estimations of *H* and respective errors of the three statistical methods are shown in Table 1.

TABLE I
STATISTICAL ESTIMATORS ERRORS

| *H* value | Higuchi method | | R/S Statistical | | Abry-Veitch | |
|---|---|---|---|---|---|---|
| | Estimation | Error | Estimation | Error | Estimatior | Error |
| 0.5 | 0.4859 | 0.0141 | 0.5437 | -0.0437 | 0.5006 | -0.0006 |
| 0.6 | 0.5652 | 0.0347 | 0.5780 | 0.0219 | 0.5916 | 0.0083 |
| 0.7 | 0.637 | 0.063 | 0.6749 | 0.0250 | 0.6911 | 0.0088 |
| 0.8 | 0.7056 | 0.0943 | 0.7672 | 0.0327 | 0.7987 | 0.0012 |
| 0.9 | 0.7827 | 0.1172 | 0.8542 | 0.0457 | 0.9181 | -0.0181 |

Table 2 presents different scenarios used in the experiments. The difference of magnitude of the mean-square error given by the hyperbolic tangent and the logistics activation function is evident.

TABLE 2
MEAN-SQUARE ERROR OF THE INITIAL TRAINING

| Hidden neurons number | Hidden layer activation function | |
|---|---|---|
| | Logistic | Hyperbolic tangent |
| | Mean-Square Error (1000 epochs) | Mean-Square Error (1000 epochs) |
| 2 | 0.3371 | 0.2191 |
| 5 | 0.3219 | 0.1225 |
| 10 | 0.3190 | 0.0197 |
| 15 | 0.1355 | 0.0060 |
| 20 | 0.2008 | 0.0131 |

For both in logistic and hyperbolic tangent functions, the worst and the best cases happened with 2 and 15 neurons, respectively. Table 2 reveals that the neural network with a hidden layer of 15 neurons and with hyperbolic tangent activation function is the best option.

After choosing the neural network topology chosen (10 input neurons, 15 hidden neurons and 1 output neuron), the real training was persued. The stopping criterion used for the learning process was an error less than 0.0010 (0.1%). The NN was trained with 2616 epochs (Fig. 2).

Six hundred 600 sequences of the test set were applied to neural network in the execution phase. These sequences had never been known by the network.

The *neural estimator* error for different values of the parameter *H* can be seen in the Fig. 3. Notice that the error has an initial increase, from $H$=0.5 to $H$=0.6, and then decreases. The three statistical estimators have used the complete traces (10000 samples) to make their calculations.

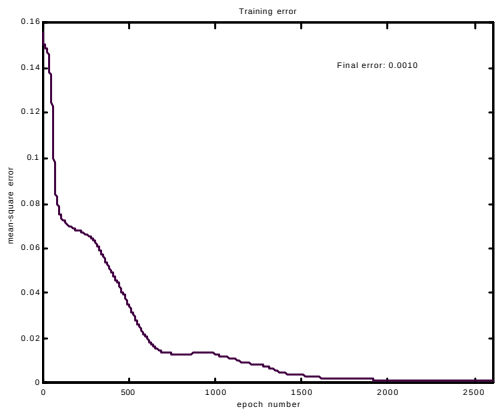


Fig. 2. Training error with the selected topology.

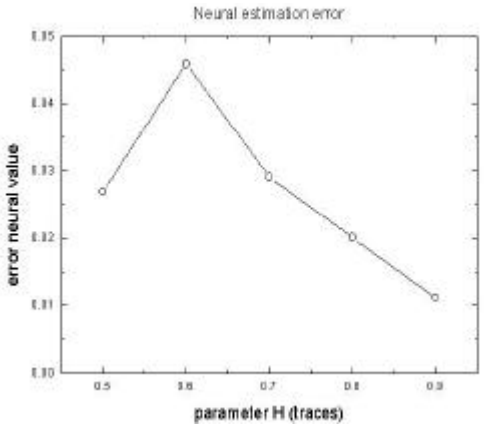


Fig. 3. *Neural estimator* evolution error.

The neural network needed only one tenth of the total trace, 1000 samples, for each LRD trace (*H* = 0.6, *H* = 0.7, *H* = 0.8 and *H* = 0.9). When the input samples were doubled duplicated for the SRD trace (*H* = 0.5), i.e., 2000 samples, there was considerable improvement in the error (Fig. 3). In other words, the neural networks enhanced the precision of the estimations given that sample size increase needed by the statistical estimators.

It can be seen in Figure 4 that for values of the Hurst parameter close to 0.5 (SRD), the neural network estimator produces the least accurate results. However, for the range of interest for network traffic (H > 0.7), the neural network estimator gives more precise results than the R/S and the Higuchi estimators.

For the range of interest of the Hurst parameter, the neural network produces values of *H* which differ at most 0.02 from the results produced by the Abry-Veitch estimator which is the most precise one. The difference between the results by these two estimators decreases as *H* increases.

Moreover, the neural networks demanded half of the number of the samples required by the statistical estimators, i.e., the neural network estimator can generate quite accurate results much faster than the statistical estimators which is specially advantageous to quickly detect variations of the Hurst parameter in real time. The drawback of the neural networks is the delay in learning which impact, however, decreases for large traces.

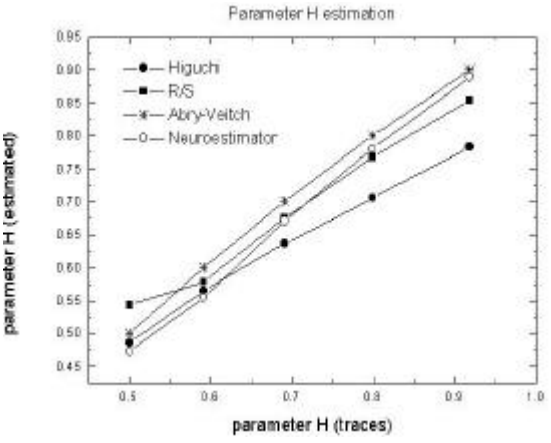


Fig. 4 .Comparison between statisticals and neuro estimators.

TABLE 3
COMPARISON OF ESTIMATORS ERRORS

| *H value* (traces) | Higuchi | R/S | Abry-Veitch | Neural estimator |
|---|---|---|---|---|
| | Error | Error | Error | Error |
| 0.5 | 0.0141 | -0.04378 | -0.000662 | 0.0269 |
| 0.6 | 0.03479 | 0.02194 | 0.008396 | 0.0458 |
| 0.7 | 0.063 | 0.02503 | 0.008893 | 0.029 |
| 0.8 | 0.09435 | 0.03277 | 0.001278 | 0.02 |
| 0.9 | 0.11722 | 0.04571 | -0.018102 | 0.0111 |

## VI. CONCLUSION

Long-range dependences have a significant impact on both network dimensioning and traffic management. The Hurst parameter is a measure of self-similarity of a process, and, thus, the intensity of the long-range dependences of a process. Small variations in the Hurst parameter value may lead to considerable changes on traffic control. Therefore, an accurate and quick evaluation of the Hurst parameter is of paramount importance. The statistical estimators requires large sample size. Consequently, it presents limitations for real time evaluation of *H*.

The present work investigated the effectiveness of a neural network estimator for the Hurst parameter. Neural networks, even demanding a significant time for training, represent an accurate and fast estimation of the parameter *H*.

To our best knowledge, this is the first time that a Hurst parameter estimation is pursued using neural networks.

## REFERENCES


[1] K. Park and W. Willinger, Self Similar Network Traffic and Performance Evaluation, Willey, 2000.

[2] W. Leland, M. Taqqu, W. Willinger and D. Wilson, On the Self-Similar Nature of Ethernet Traffic (Extended Version), IEEE/ACM Transaction on Networking,vol 2, no 1, pp. 1-15, February 1994.

[3] M.Taqqu, V. Teverovsky, and W. Willinger, *Estimators for Long-Range Dependence: an Empirical Study*, Fractals, vol 3, No 4, pp. 785-788, 1995.

[4] M. Taqqu, and V. Teverovsky, *Robustness of Whittle-type Estimators for Time Series with Long-Range Dependence*. Stochastic Models 13 (1997) pp 723-757. http://math.bu.edu/people/murad/articles.html

[5] R. Hect-Nielsen, neurocomputing, Addison-Wesley Publishing Company, 1990.

[6] L. Fausset, Fundamentals of neural networks, Prentice-Hall International, new jersey, 1994

[7] T. Higuchi, *Approach to an irregular time series on the basis of the fractal theory*. Physica D, 31:277-283, 1988.

[8] P. Abry, and D. Veitch, *Wavelet Analysis of Long-Range Dependence Traffic*. IEEE Transactions on Informations Theory, vol. 44, No. 1, pp. 2-15, 1998.

[9] D.Veitch, and P. Abry, *A Wavelet-Based Joint Estimator of the Parameters of Long-Range Dependence*. IEEE Transactions on Informations, vol. 45, No. 3, pp. 878-897, 1998.

[9] M. Chi, E. Neal, and G. Young, *Practical Applications of Fractional Brownian Motion and Noise to Synthetic Hydrology*. Water Resources Research, 9:1523-1533, December, 1973.

[10] B. Mandelbrot, and J.W. Ness, *Fractional brownian motions, fractional noises and applications*. SIAM Review, 10:422-437, October 1968.

[11] U. of Stuttgart, *SNNS - Stuttgart Neural Network Simulator* - User Manual, Version 4.1, 1995.( http://www-ra.informatik.uni-tuebingen.de/SNNS/ .).