Test error versus training error in artificial neural networks for systems affected by noise

Fernando Morgado Dias and Ana Antunes

***Abstract*—**This paper reports an empirical study of the behavior of the test and training errors in different systems. Frequently the test error of Artificial Neural Networks is presented with a monotonic decreasing behavior as a function of the iteration number, while the training error also continuously decreases. The present paper shows examples where such behavior does not hold, with data collected from systems where it is corrupted by either noise or actuation delay. This shows that selecting the best model is not a simple question and points to automatic procedures for the selection of models as the best solution to optimize their capacity, either with the Regularization or Early Stopping techniques.

***Keywords*—** Early Stopping, Feedforward Neural Networks, Regularization, Test Error, Training Error, Weight decay.

I. Introduction

THIS paper reports an empirical study of the behavior of the test and training errors in different systems.

It is very common in the literature to present the test error of an Artificial Neural Network (ANN) with a monotonic decreasing behavior as a function of the iteration number, while the training error also continuously decreases. This behavior is, most of the times, illustrated by drawings instead of simulations or data from a real system. Some examples of exceptions can be found in [1] and [2].

The present paper shows examples where such behavior does not hold, with data collected from systems where it is corrupted by either noise or actuation delay.

The behavior of the test error presented points to automatic procedures for the selection of models as the best solution to optimize their capacity, either with the Regularization or Early Stopping techniques.

The models presented here were trained using a non-variable pre-established initial set of weights to enable the comparison of the results without the random effect of a variable set of weights.

Manuscript received April 1, 2008 Revised version received June 30, 2008.

F. Morgado Dias is with the Department of Mathematics and Engineering, University of Madeira, 9000-390 Funchal, Portugal (+351-291-705307; fax: +351-291-705307; e-mail: morgado@ uma.pt).

A. Antunes, is with Escola Superior de Tecnologia de Setúbal of the Instituto Politécnico de Setúbal, Campus do IPS, Estefanilha, 2910-761 Setúbal, Portugal (e-mail: aantunes@est.ips.pt).

II. The Test Systems

The data presented here is collected from two different systems.

A. Cruise Control Distributed System

The first one is from a first order system corresponding to a cruise control system as shown in equation 1:

$$H(s) = \frac{1}{s+1} \tag{1}$$

The cruise control system is distributed through three processing nodes over a fieldbus according to the architecture presented in figure 1.

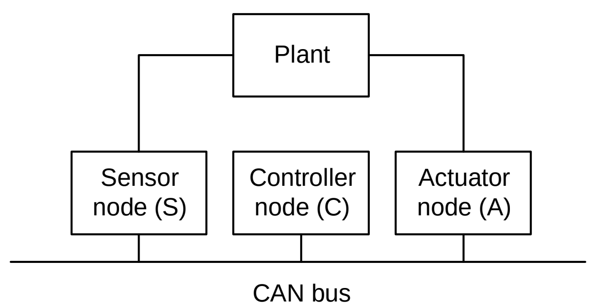

Fig. 1. Architecture of the distributed control system.

The sensor node samples the plant and sends the sampled value to the controller node. The controller receives the sampled value and computes the actuation value to send to the actuator node. The actuator node receives the actuation value and acts on the plant.

The distribution of the controller and the use of a fieldbus to connect the nodes of the control loop induce variable delay between the sampling instant and the actuation instant. The delays are introduced due to the medium access control (MAC) of the network, the processing time, the processor scheduling in the nodes and the scheduling mechanism used to schedule the bus time. These delays are usually variable from iteration to iteration of the control loop.

To simulate this architecture the delays were introduced according to the representation of figure 2.

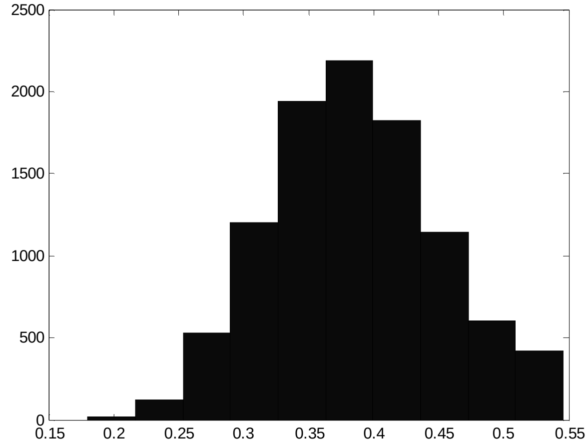

Fig. 2. Histogram of the number of samples versus the delay introduced.

B. *Reduced Scale Prototype Kiln*

The second system is a reduced scale prototype kiln affected by measurement noise. Additional details about this system can be found in [6] and [8].

The system is composed of a kiln, electronics for signal conditioning, power electronics module, cooling system and a Data Logger from Hewlett Packard HP34970A to interface with a Personal Computer (PC).

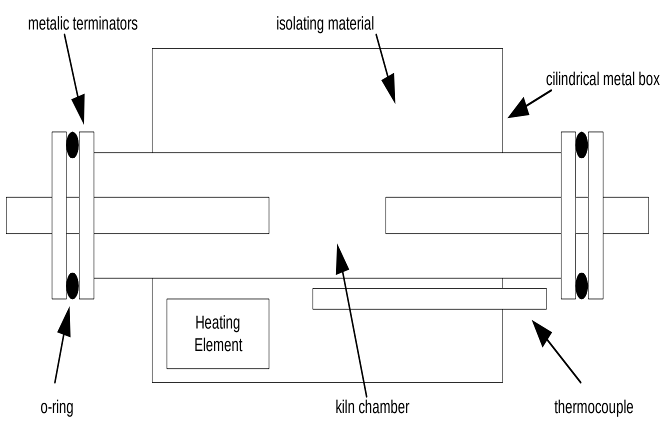

Fig. 3. Schematic view of the kiln.

Details about the kiln can be seen in figure 3 and the connections between the modules can be seen in figure 4.
The kiln is a cylindrical metal box of steal which is completely closed, filled with an isolating material up to the kiln chamber. The kiln chamber is limited by the metallic terminators and o-rings.
Inside the chamber there is an oxygen pump and an oxygen sensor that will be used for the second loop mentioned above.
The heating element is an electrical resistor that is driven by the power module.

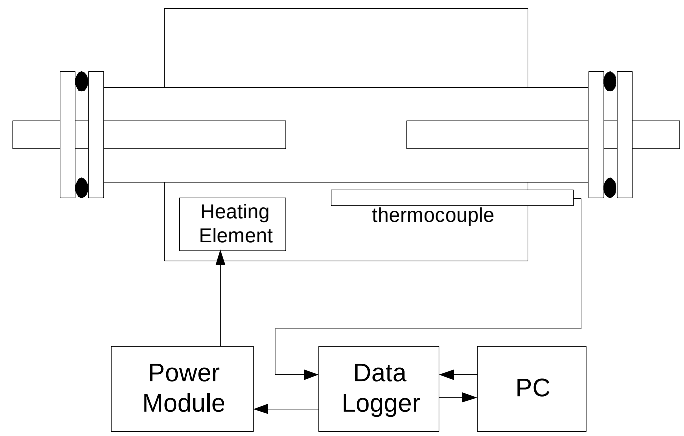

Fig. 4. Block diagram of the system.

The Data Logger acts as an interface to the PC where the controller is implemented using MATLAB. Through the Data Logger bi-directional information is passed: control signal in real-time supplied by the controller and temperature data for the controller. The temperature data is obtained using a thermocouple.

The power module receives a voltage signal from the controller implemented in the PC, which ranges from 0 to 4.095V and converts this signal in a power signal ranging from 0 to 220V.

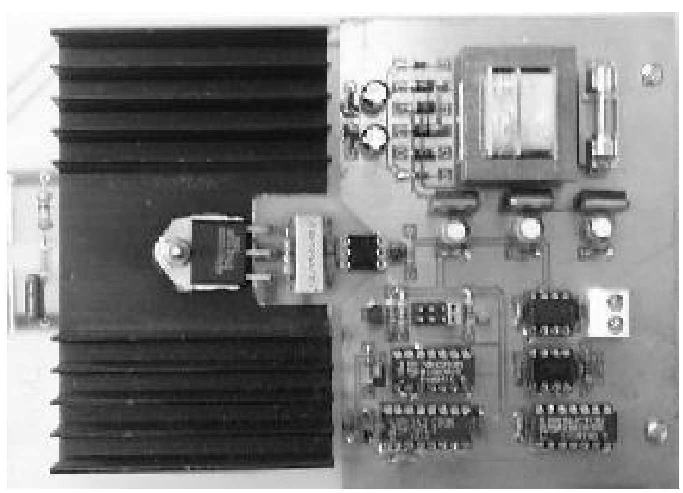

Fig. 5. Picture of the power module.

The signal conversion is implemented using a sawtooth wave generated by a set of three modules: zero-crossing detector, binary 8 bit counter and D/A converter. The sawtooth signal is then compared with the input signal generating a PWM type signal.

The PWM signal is applied to a power amplifier stage that produces the output signal. The signal used to heat the kiln produced this way is not continuous, but since the kiln has integrator behavior this does not affect the functioning.

The actual implementation of this module can be seen in figure 5 and a block diagram of the power module processing can be seen in figure 6.

Operating range of the kiln under normal conditions is between 750ºC and 1000ºC. A picture of the kiln and electronics can be seen in figure 7.

Both systems were chosen because the effect of the perturbations provides a behavior which is different from a simulated system.

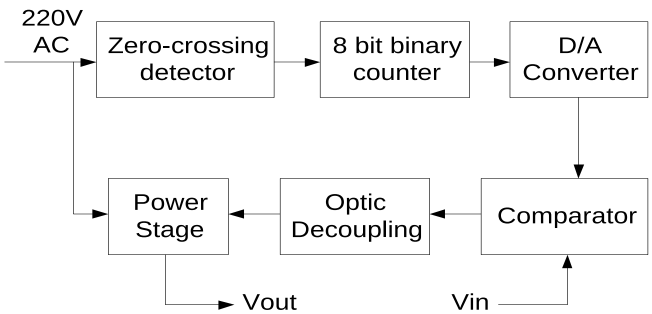

Fig. 6. Block diagram of the power module.

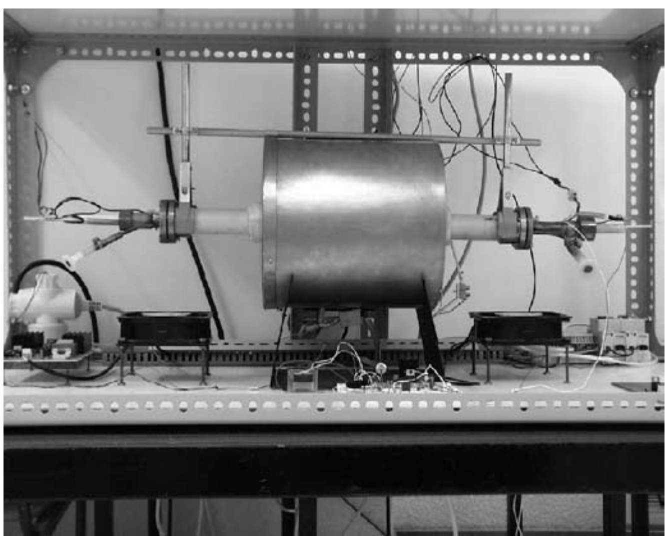

Fig. 7. Picture of the kiln and electronics.

III. Training versus Test Error

For both systems training was performed for 10000 iterations using the Levenberg-Marquardt algorithm [3] [4]. After each training epoch, the test sequence was evaluated to allow permanent monitoring of the training and test error.

Usually a train and a test sequence are used. The first one is used to update the weights based in the error obtained at the output and the second is used to test if the ANN is learning the general behavior of the system to be modeled, instead of learning the training sequence.

Some authors consider using a third sequence to ensure that the ANN is able to generalize the behavior that is desired in different situations or to compare the quality of different networks [5].

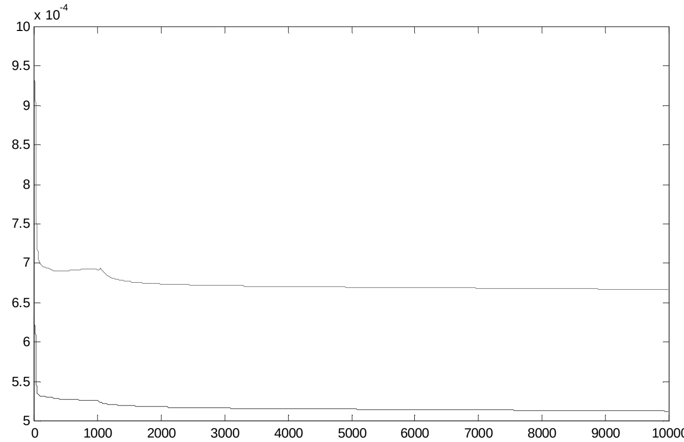

Fig. 8- The test and training error as a function of the iteration for the direct model of the first system with 4 neurons.

Figures 8 to 15, for the first system, and 16 to 27, for the second, represent the plots of both train and test errors for the training of both direct and inverse models of the systems.

In all the figures the training error is always the line with the lower values.

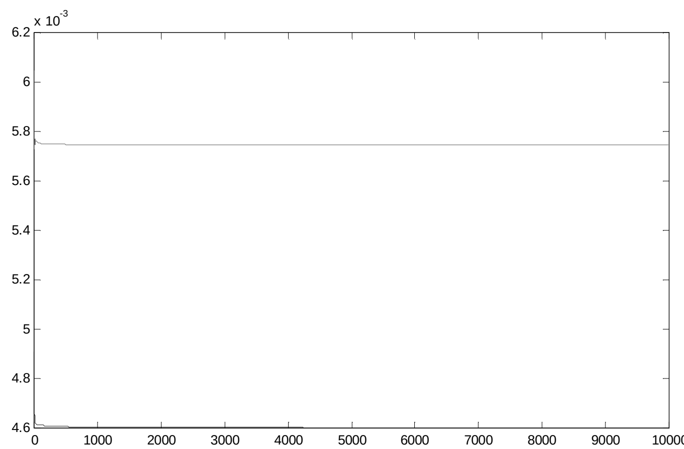

Fig. 9- The test and training error as a function of the iteration for the inverse model of the first system with 4 neurons.

IV. Discussion

As stated in the introduction, the literature presents frequently both the training and test errors with a monotonic behavior. It can be seen from the two examples shown here that such behavior is not always found in real systems.

The examples show, in many situations, that both for direct and inverse models, while the training error is always monotonic, the test error finds frequently hills and vales.

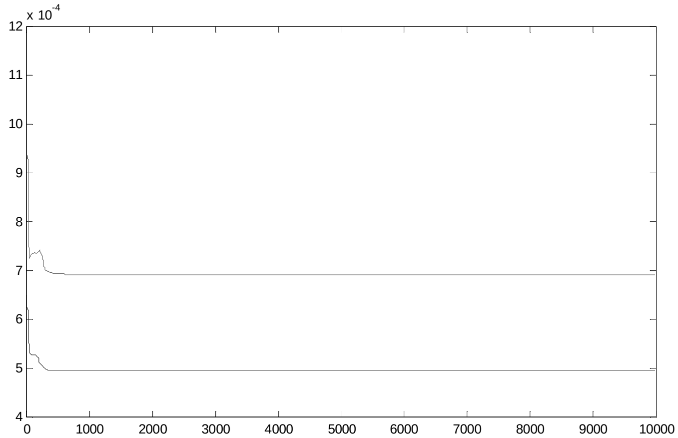

Fig. 10- The test and training error as a function of the iteration for the direct model of the first system with 6 neurons.

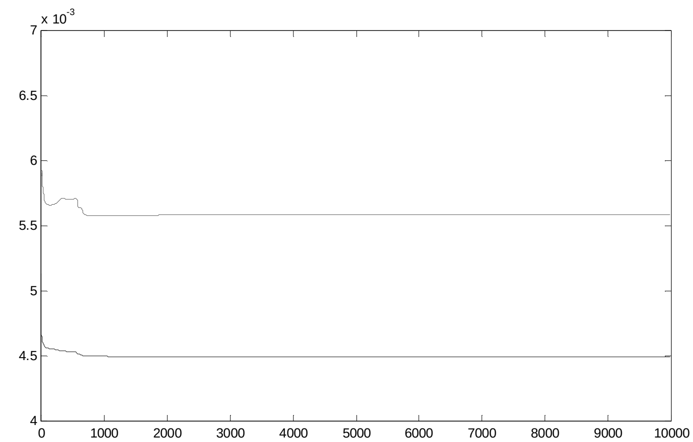

Fig. 11- The test and training error as a function of the iteration for the inverse model of the first system with 6 neurons.

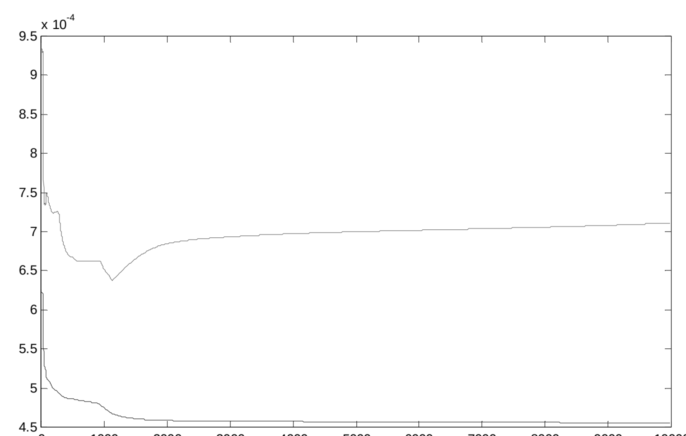

Fig. 12- The test and training error as a function of the iteration for the direct model of the first system with 8 neurons.

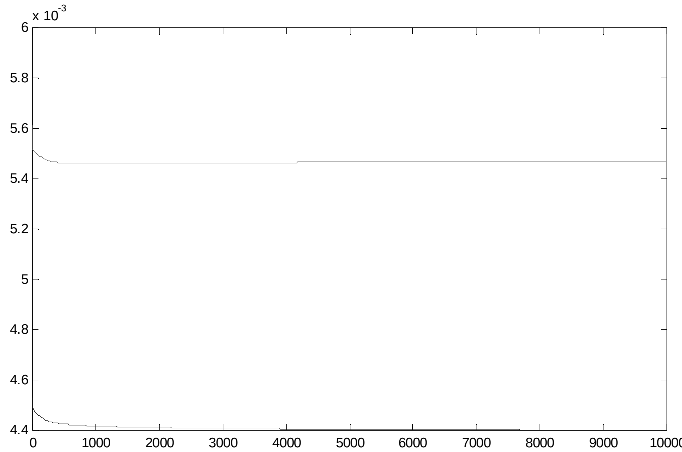

Fig. 13- The test and training error as a function of the iteration for the inverse model of the first system with 8 neurons.

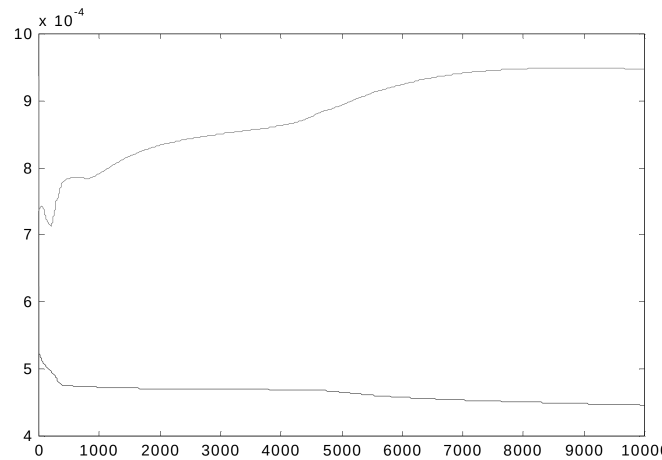

Fig. 14- The test and training error as a function of the iteration for the direct model of the first system with 10 neurons.

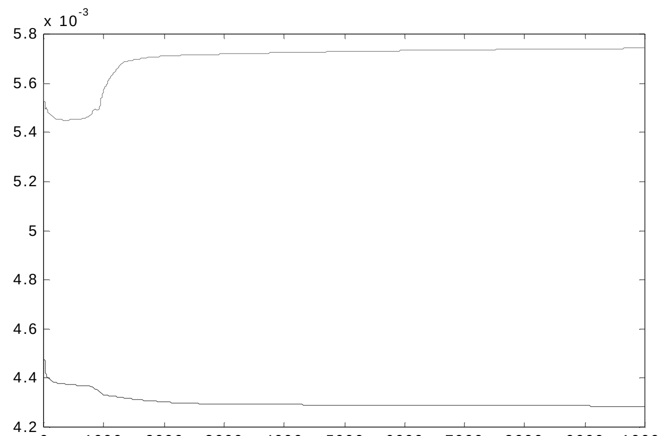

Fig. 15- The test and training error as a function of the iteration for the inverse model of the first system with 10 neurons.

Figures 8 to 11 of the first model and all the figures of the direct model of the second system show an initial phase of learning, followed by a very flat zone where learning is very slow for the test and training error.

In the rest of cases it is possible to find a very different behavior for the train and test error. While the training error continues to decrease with iterations, the test error shows frequently hills and vales.

The different tests shown in the figures correspond to the two systems referred above, modeled with different number of neurons. While usually a larger set of weights allows obtaining a lower training error, the existence of more degrees of freedom enable the test error to be composed of much more ups and downs then it is the case with less parameters.

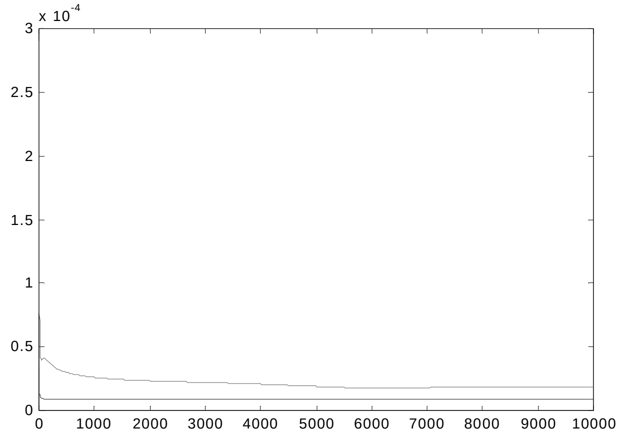

Fig. 16- The test and training error as a function of the iteration for the direct model of the second system with 4 neurons.

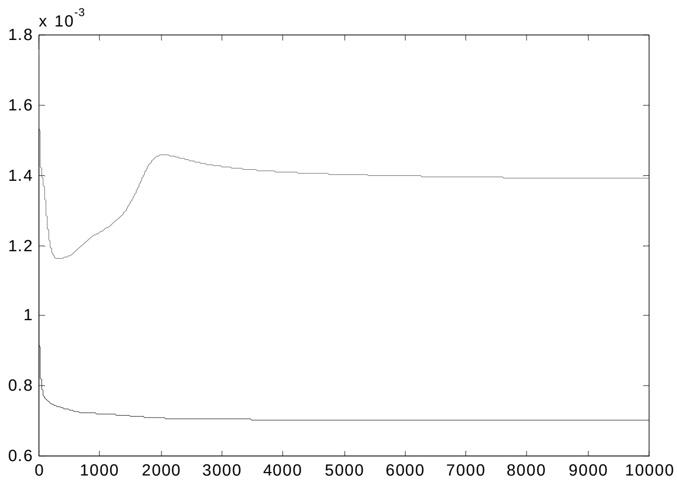

Fig. 17- The test and training error as a function of the iteration for the inverse model of the second system with 4 neurons.

For the second system more figures are shown because, with a larger number of parameters it presents curves which are less common.

V. Regularization and Early Stopping

The oscillations found in the test error of the systems chosen as an example lead to the necessity of choosing carefully the length of the training stage or to apply another solution to obtain the best quality for the models.

Two possible solutions that can be used to cope with these problems are Early Stopping and Regularization techniques.

A. Regularization

For the training algorithms that are based on derivatives the first parameters to be updated are the ones with larger influence in the criteria to be minimized, while in a second phase other less important parameters are updated.

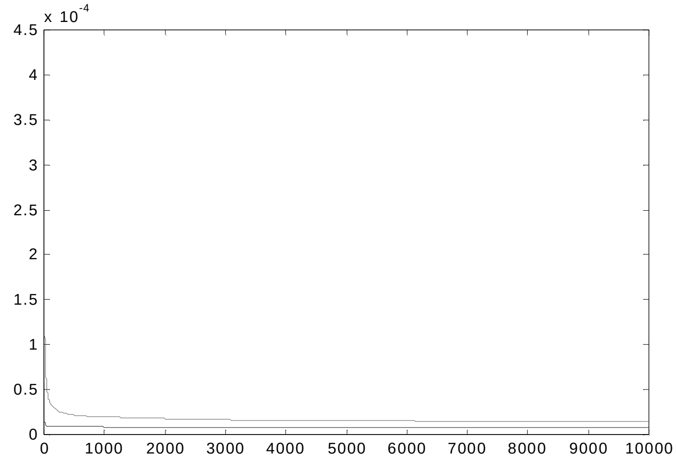

Fig. 18- The test and training error as a function of the iteration for the direct model of the second system with 6 neurons.

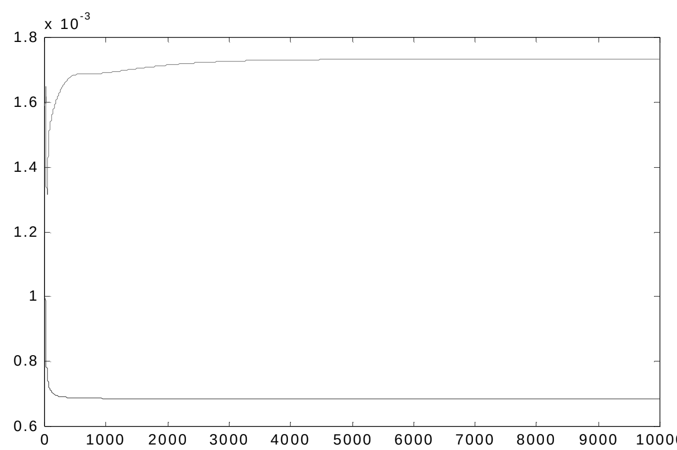

Fig. 19- The test and training error as a function of the iteration for the inverse model of the second system with 6 neurons.

These last parameters to be updated are the ones responsible for the overtraining problem by learning characteristics of the training signal and the noise. The overtraining, i.e. excessive training, situation results in a network that over fits the training sequence but is not capable of the same performance with a test sequence, because the ANN has learned details of the training sequence instead of the general behavior.

One way to avoid this second phase in training is called regularization and it consists of changing the criteria to be minimized according to:

$$W(\theta) = V(\theta) + \delta\|\theta\|^2 \qquad (2)$$

where δ, the weight decay is a small value and   is the original criterion to be minimized.

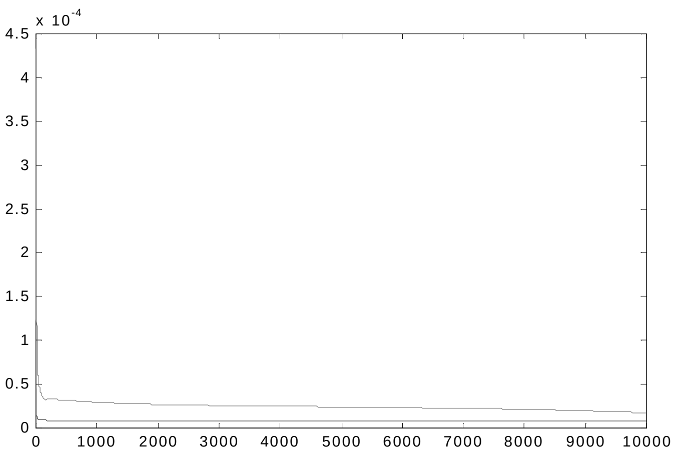

Fig. 20- The test and training error as a function of the iteration for the direct model of the second system with 8 neurons.

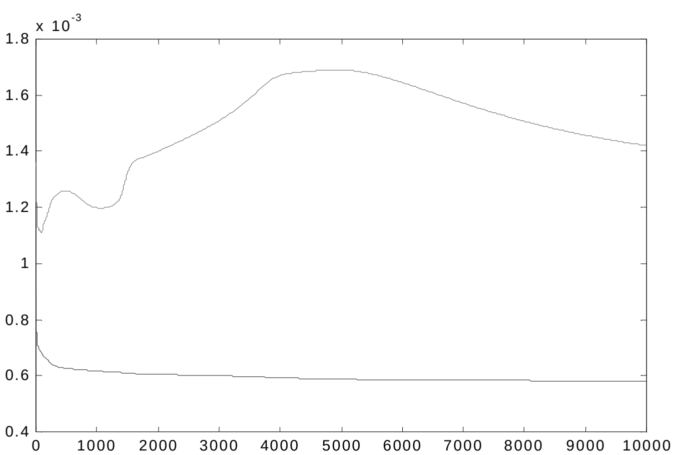

Fig. 21- The test and training error as a function of the iteration for the inverse model of the second system with 8 neurons.

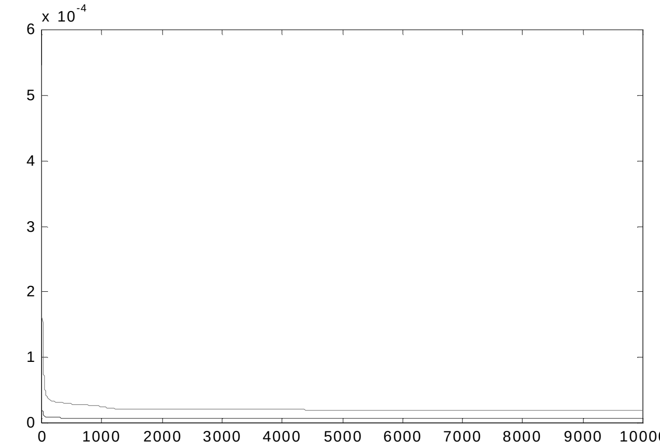

Fig. 22- The test and training error as a function of the iteration for the direct model of the second system with 10 neurons.

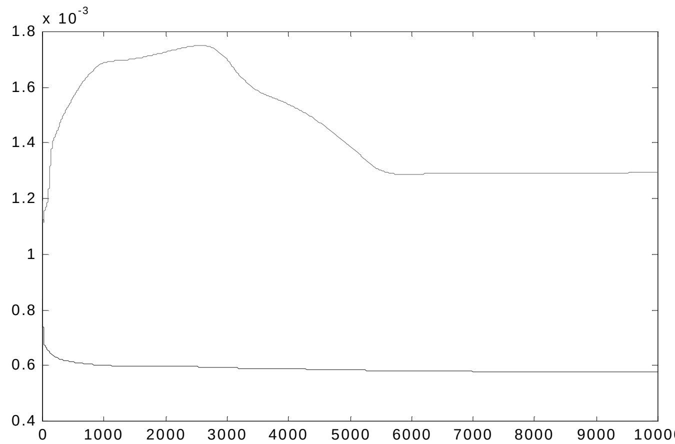

Fig. 23- The test and training error as a function of the iteration for the inverse model of the second system with 10 neurons.

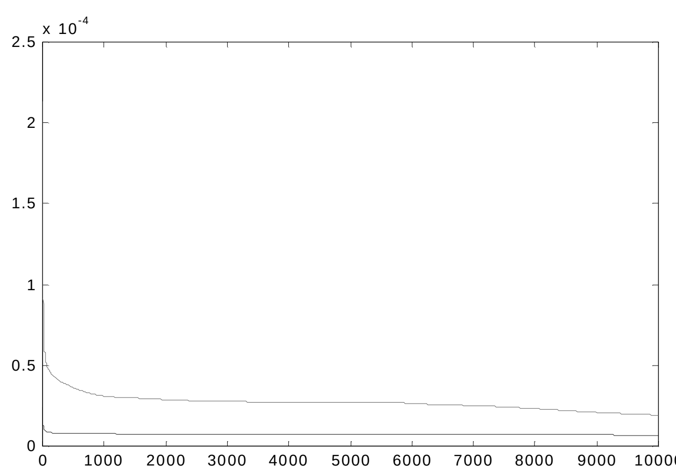

Fig. 24- The test and training error as a function of the iteration for the direct model of the second system with 15 neurons.

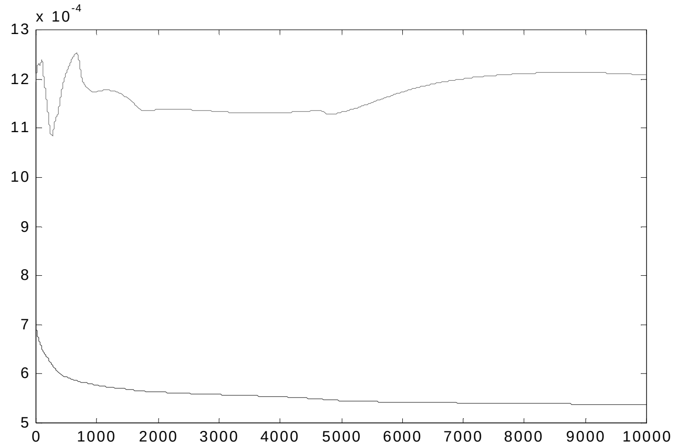

Fig. 25- The test and training error as a function of the iteration for the inverse model of the second system with 15 neurons.

The idea is to eliminate the so called second phase in learning where parameters with small influence are updated by introducing a trend towards zero in the parameters.

The difficulty is to determine the appropriate value of δ for performing regularization.

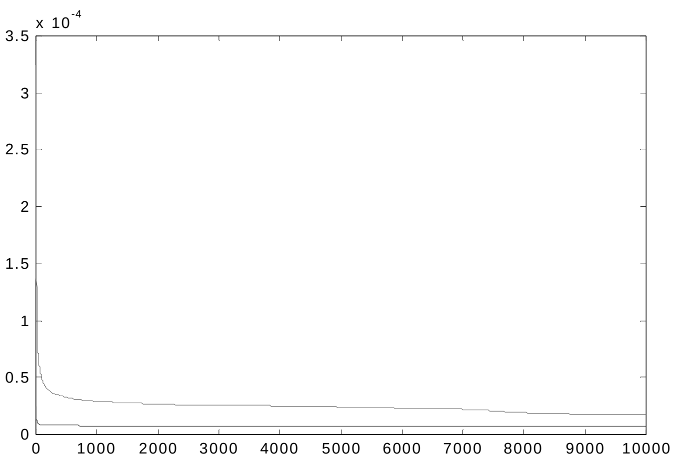

Fig. 26- The test and training error as a function of the iteration for the direct model of the second system with 20 neurons.

B. *Early Stopping*

Another way to avoid the overtraining, called early stopping, which is quite intuitive, consists in stopping training before the second phase of training starts but after the first one is concluded so that the characteristics of the system are learned.

Clearly the difficulty here is to find the exact number of iterations for performing the training.

Both solutions have been proved to be formally equivalent in [2]. Nevertheless it is important to take into account the difficulties to determine the regularization parameter for explicit regularization or the number of iterations to use for early stopping. One example of comparison of both techniques in an automated procedure can be found in [7].

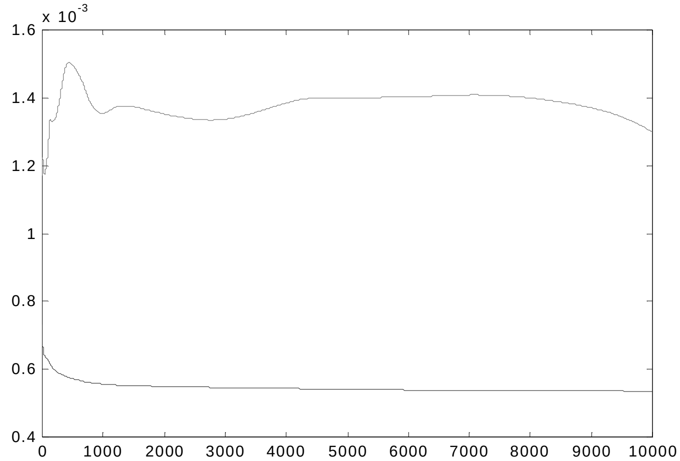

Fig. 27- The test and training error as a function of the iteration for the inverse model of the second system with 20 neurons.

VI. Conclusion

Two systems were presented where the behavior of the test and training error is not the monotonic decreasing usually pointed out in the literature.

The objective is to show that for systems subject to noise or actuation delay it is quite common to find a behavior different from simulated systems.

The differences found in the test error for ANNs with more parameters are also relevant since this larger set of adjustable weights can lead to a better network, but only if the test error is carefully evaluated.

These examples suggest that the choice of the models' characteristics must be careful to avoid getting a worst model after a larger training period. This urges for the use of the Early Stopping and Regularization techniques that can be very helpful both for manual and automatic model selection.

VII. References:

[1] M. Nørgaard, "System Identification and Control with Neural Networks", PhD Thesis, Department of Automation, Technical University of Denmark, 1996.

[2] Sjöberg, J. & Ljung, L., "Overtraining, regularization, and searching for minimum with application to neural nets", International Journal of Control 62(6), 1391–1407, 1995.

[3] K. Levenberg, "A method for the solution of certain problems in least squares", Quart. Appl. Math., 2:164—168, 1944.

[4] D. Marquardt, "An algorithm for least -squares estimation of nonlinear parameters", SIAM J. Appl. Math., 11:431—441, 1963.

[5] Lutz Prechelt, "Automatic early stopping using cross validation: Quantifying the Criteria", Neural Networks, 1998.

[6] F. Morgado Dias and A. Mota, "Comparison between Different Control Strategies using Neural Networks", Mediterranean Control Conference, Dubrovnik, 2001.
[7] F. Morgado Dias, A. Antunes and A. Mota, "Automating the Construction of Neural Models for Control Purposes using Genetic Algorithms", 28th Annual Conference of the IEEE Industrial Electronics Society, Sevilha, 2002.
[8] F. Morgado Dias, A. Antunes and A. Mota, "A new hybrid direct/specialized approach for generating inverse neural models", 6th WSEAS Int. Conf. on Algorithms, Scientific Computing, Modelling and Computation (ASCOMS'04), Cancun, México, 2004.
[9] A. Antunes, F. Morgado Dias, A. Mota, "Influence of the sampling period in the performance of a real-time distributed system under jitter conditions", 6th WSEAS Int. Conf. on Telecommunications and Informatics (TELE-INFO'04), Cancun, 2004.
[10] J. Vieira, F. Morgado Dias, A. Mota,"Neuro-Fuzzy Systems: A Survey", 5th WSEAS NNA International Conference on Neural Networks and Applications, Udine, Italia, 2004.

Fernando Morgado Dias received his engineering degree in Electronics and Telecomunications (1994) from the University of Aveiro, Portugal, his *Diplôme D'Études Approfondies* in Microelectronics from the University Joseph Fourier in Grenoble, France (1995) and his PhD from the University of Aveiro, Portugal in 2005. His main research field is artificial neural

He is a former teacher from the Superior School of Technology of Setubal, Portugal and is currently Assistant professor at the University of Madeira, Portugal. He has published over 40 scientific papers in the fields of control and neural networks.
Prof. Dias has received two awards, the best paper in session CCS-07 Robust Controllers I, in the 28th Annual Conference of the IEEE Industrial Electronics Society conference, with the paper "Additive Internal Model Control: an Application with Neural Models in a Kiln", held in Sevilha, Spain, 2002 and second place in the best design student project category with the integrated circuit "AMBROSIO: Advanced Microchip for Broadband System Improvement and Optimization", in the 5th Eurochip Workshop, Dresden, Germany 1994.

Ana Antunes received her engineering degree in Electronics and Telecomunications (1994) from the University of Aveiro, Portugal, her *Diplôme D'Études Approfondies* in Microelectronics from the University Joseph Fourier in Grenoble, France (1995) and she is preparing her PhD degree at the University of Aveiro, Portugal. Her main research field is distributed control systems.

She currently teaches at the Superior School of Technology of Setúbal, Portugal and has published over 30 scientific papers in the fields of distributed control and neural networks.
Prof. Antunes has received three awards: best paper in the conference ANIPLA 2006- Methodologies for Emerging Technologies in Automation with the paper "Dynamic Rate Adaptation in Distributed Computer Control Systems", held in Roma, Italy; best paper in session CCS-07 Robust Controllers I, in the 28th Annual Conference of the IEEE Industrial Electronics Society conference, with the paper "Additive Internal Model Control: an Application with Neural Models in a Kiln", held in Sevilha, Spain, 2002 and second place in the best design student project category with the integrated circuit "AMBROSIO: Advanced Microchip for Broadband System Improvement and Optimization", in the 5th Eurochip Workshop, Dresden, Germany 1994.