Nonlinear Dynamics, Chaos, Control and Their Applications to Engineering Sciences, Vol. 6: Applications of Nonlinear Phenomena, 2002. J. M. Balthazar, P. B. Gonçalves, R. M. F. L. R. F. Brasil, I. L. Caldas, F. B. Rizatto, Editors.
ISBN: 85-900351-6-6

Data Assimilation in Chaotic Dynamics Using Neural Networks

A.G. Nowosad[a] (DCM) A. Rios Neto[b] H.F. de Campos Velho[a] (LAC)

[a] Instituto Nacional de Pesquisas Espaciais (INPE)
Caixa Postal 515
12201-970 – São José dos Campos (SP), BRAZIL,
E-mail: alex@met.inpe.br - haroldo@lac.inpe.br

[b] Instituto de Pesquisa e Desenvolvimento (IP&D)
Universidade do Vale do Paraíba (UNIVAP)
Av. Shishima Hifumi, 2.911 - Urbanova
12245-720 - São José dos Campos (SP), BRAZIL
E-mail: atair@univap.br

Abstract

Multilayer Perceptron Neural Networks are used for data assimilation in two nonlinear dynamic systems: the Hénon and Lorenz systems in chaotic state. The approach "emulates" the Kalman Filter data assimilation method avoiding recalculation of the gain matrix at each instant of assimilation. In the case of Hénon system an Adaptive Extended Kalman Filter was used to provide examples for network training. In the case of Lorenz system the Extended Kalman Filter was used for network training. Preliminary test results obtained are shown.

Keywords: Data assimilation, neural networks, chaotic dynamics.

1 Introduction

The possibility of using Multilayer Perceptron Neural Networks for data assimilation in the context of highly nonlinear systems that exhibit chaotic behaviour, the Hénon and Lorenz systems, is investigated. Ultimately, it is expected that the experience gained from the simple models may guide the application of this new technique in data assimilation for atmospheric numerical models.

In the case of atmospheric continuous data assimilation there are many deterministic and probabilistic methods (Daley, 1991; Todling, 1997). Deterministic methods include *Dynamic Relaxation*, *Variational Methods* and *Laplace Tranform*, whereas probabilistic methods include *Optimal Interpolation* and *Kalman Filtering*. Dynamic Relaxation assumes the prediction model to be perfect, as does

Laplace Transform. Variational Methods and Optimal Interpolation can be regarded as minimum-mean-square estimation of the atmosphere. Kalman Filtering has the advantage of minimizing the error in the assimilation (Nowosad, Rios Neto, Campos Velho, 1999) *plus* propagating itself the error from one data insertion to the next. But it is computationally too expensive for large systems. It is the intention of this work to examine whether neural networks can "emulate" the accuracy of Kalman Filtering with economy in computer time.

The new approach adopted requires training of a multilayer perceptron to emulate a chosen data assimlation method. Kalman filter data assimilation methods are used to generate training examples for the networks. In the case of Hénon system an Adaptive Extended Kalman Filter is used to provide examples for network training. In the case of Lorenz system the Extended Kalman Filter is used for network training. The preliminary results obtained are promising.

In the next sections the paper is organized as follows: in section 2 there is a brief exposition of artificial neural networks (ANN); in section 3 the data assimilation problem is considered in the context of Hénon and Lorenz systems; in section 4 the proposed approach is tested; and section 5 contains a summary of the results and some comments on the new method.

2 Backpropagation neural networks

An artificial neural network (ANN) is an arrangement of units characterized by: a large number of very simple neuron-like processing elements; a large number of weighted connections between these elements, where the knowledge of thea network is stored; highly parallel, distributed processing. The processing element in an ANN is a linear combiner with multiple weighted inputs, followed by an activation function. There are several different architectures of ANN's, most of which directly depend on the learning strategy adopted. It is not the aim of the paper to present an overview on ANN. Instead, a very brief description of the ANN used is focused: the multilayer Perceptron with backpropagation learning (Haykin, 1994).

The Multilayer Perceptron with backpropagation learning, also called the backpropagation neural network, is a feedforward network composed of an input layer, an output layer, and a number of hidden for extracting high order statistics from the input data (Haykin, 1994). In order to make the network more flexible to solve nonlinear problems, the activation functions for the hidden layer are sigmoid functions. Here $\tanh(x)$ is used as activation function at the hidden layers and the identity function is used at the output layer. Mathematically, a feedforward network simply maps input vectors of real values onto output vector of real values. The connections have associated weights that are adjusted during learning process, thus changing the performance of the network. There are two distinct phases in the usage of an ANN: the training phase (learning process) and the running phase (activation of the network). In the training phase, the weights are adjusted for the best performance of the network in establishing the mapping of many input-output vector pairs. Once trained, the weights are fixed and new inputs can be presented to the network for it to compute corresponding outputs, based on what it has learned.The training phase of a backpropagation network is controlled by a supervised learning algorithm, which differs from unsupervised learning. The main difference is that the latter uses only information contained in the input data, whereas the former requires both input and output (desired) data, which permits the calculation of the error of the network as the difference between the calculated

output and the desired vector. Adjustment of the network's weights is conducted by backpropagating such error through the network. The weight change rule is a development of the Perceptron learning rule. Weights are changed by an amount proportional to the error at that unit, times the output of the unit feeding into the weight. Equation (1) shows the general weight correction according to the so-called delta rule

$$\Delta w_{ij} = \eta \, \delta_j \, y_i \tag{1}$$

where δ_j is the local gradient, y_i is the input signal of the neuron j, and η is the learning rate parameter that controls the strength of change.

3 Data assimilation in nonlinear systems

The data assimilation process can be described as a procedure that uses observational data to improve a prediction made by an inaccurate mathematical model. For example, suppose a computational model where many properties are only expressed approximately, like turbulent fluxes. Typically, the assimilation process can be outlined as a two step process:

$$\text{Forecast step:} \quad w_n^f = F[w_{n-1}^a] \; ; \qquad \text{Analysis step:} \quad w_n^a = w_n^f + \mathbf{d}_n \; ;$$

where w_n represents model state variable at n time step, $F[.]$ is the mathematical (forecast) model, superscripts f and a denote forecasted and analyzed values respectively, and $\mathbf{d}_n$ is the inovation. Several methods of data assimilation have been developed for air quality problems (Zannetti, 1990), numerical weather prediction (Daley, 1991), and numerical oceanic simulation (Bennet, 1992). One of these assimilation techniques is based on the Kalman filter, where the analysis inovation $\mathbf{d}_n$ is computed as a linear function of the misfit between observation (denoted with superscript o) and forecast:

$$\mathbf{d}_n = G_n \left(w_n^o - H_n \, w_n^f \right) \; , \tag{2}$$

where G_n is a weighting (gain) matrix, w_n^o is the with error observed value of w_n and H_n is an observation matrix. An adaptive extended Kalman filter has been tested in strongly nonlinear dynamical systems for assimilation procedure: the Lorenz chaotic system, and DYNAMO meteorological model - a simplified version of the shallow water equations (Nowosad et al., 1999). The goal of the present study is to introduce a new method to compute an assimilation function, where such function is implemented by an ANN: $w_n^a = F_{\text{ANN}}(w_n^f, w_n^o)$.

This process of adding new observation data to the current integration of a numeric prediction model is known as *Data Assimilation*. A similar problem may occur when disturbances are inserted in the computer simulation of small-dimensional chaotic nonlinear synamical systems. Two such examples are the Hénon and the Lorenz systems in chaotic state. The Hénon discrete-time system is (Thompson and Stewart, 1986):

$$X_{n+1} = 1 + Y_n - aX_n^2, \quad Y_{n+1} = bX_n \quad . \tag{3}$$

This mapping will be evolved adopting $a = 1.4$ and $b = 0.3$ so that the system will be in chaotic state. The unit of time, though unnecessary, will be taken as *second*, in the sense that $\Delta n = (n+1) - n = 1 \; s$.

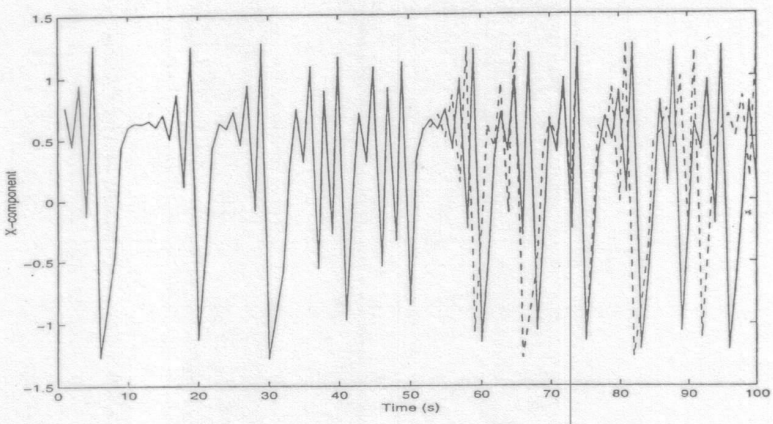

Figure 1: Temporal corruption of disturbed variable X.

Initial conditions are $[X_0 \;\; Y_0]^T = [0.9 \;\; 0.9]^T$. Two simulations will be made, in the second one inserting at $t = 50\ s$ the disturbances $[\Delta X,\ \Delta Y]^T = \left[\sqrt{10^{-3}},\ \sqrt{10^{-3}}\right]^T$. The result is illustrated by figure 1. The dashed line represents the simulation without disturbance and the solid line the simulation with disturbance. The insertion of the disturbances seriously damages the prediction of X, from $t = 53\ s$ on. Here modification of time-dependent variables X and Y or their substitution for inappropriates estimates during the integration can generate undesired solutions. A similar experiment will be made with the Lorenz dynamic system (Miller et al, 1994):

$$dX/dt = -\sigma(X - Y), \quad dY/dt = R\,X - Y - X\,Z, \quad dZ/dt = X\,Y - b\,Z \quad . \tag{4}$$

This system will be integrated using the Euler predictor-corrector method adopting $\Delta t = 0.001\ s$, $\sigma = 10$, $b = 8/3$, $R = 28$, with these values the system shows a chaotic dynamics. Initial conditions are $[X_0 \;\; Y_0 \;\; Z_0]^T = [1.508870 \;\; -1.531271 \;\; 25.46091]^T$. Again, two simulations will be made, in the second one inserting at $t = 7,5\ s$ the disturbances $[\Delta X,\ \Delta Y,\ \Delta Z]^T = [10^{-2},\ -10^{-2},\ 10^{-2}]^T$. The result is illustrated by figure 2. The dashed line represents the simulation without disturbance and the solid line the simulation with disturbance. As is known, the insertion of inappropriate estimates, even if containing only small perturbations, during the integration may cause complete loss of the original dynamics of the process.

4 Testing the new approach

Having presented in the previous section a new approach for the Assimilation of Data, using a multilayer perceptron, it is necessary to test it. The first test was made using the Hènon system and the second was made using the Lorenz system. Both systems were quoted in section 3.

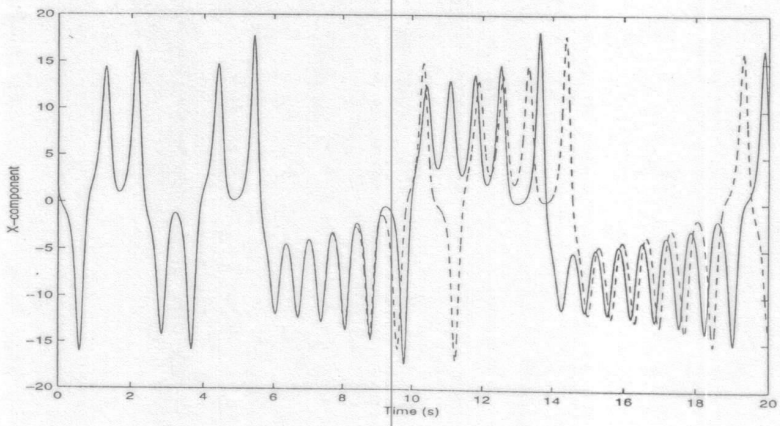

Figure 2: Temporal corruption of disturbed variable X.

4.1 Implementation of multilayer perceptron

The backpropagation training algorithm in batch mode was adapted from MATLAB Neural Network Toolbox (MNNT) (Demuth and Beale, 1994) to FORTRAN. The algorithm used constant learning rate η. The toolbox features a method for initialization of weights w_{ij} called *Nguyen-Widrow method* (1990) which was also adapted. In what follows the training error at each iteration m is defined as

$$e_m = \sum_{k=1}^{N} \left\| F_{\text{ANN}}(w_k^i, w_k^o, W, \mu, m) - w_k^a \right\|_2 \tag{5}$$

where W and μ are the weights and biases of the network, k represents each example and N the number of examples in the training set.

4.2 Results obtained with the chaotic Hénon system

The first numeric experiment consisted in using an Adaptive Extended Kalman Filter (Rios Neto and Kuga, 1985; Nowosad et al, 1999; Gelb, 1978) to train the neural newtwork. The Filter was implemented using:

$$w_{n+1} = f(w_n), \quad F_n \doteq \left.\frac{\partial f}{\partial w_n}\right|_{w_n = w_n^a}, \quad H_{n+1} \doteq I, \quad Q_0 = 0 \tag{6}$$

$$R_n \doteq \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.001 \end{bmatrix}, \quad P_0^a \doteq 10 \times \begin{bmatrix} X_0^2 & 0 & 0 \\ 0 & Y_0^2 & 0 \\ 0 & 0 & Z_0^2 \end{bmatrix} \tag{7}$$

$$P_0^{qa} = 10 \times R_0^q, \quad \gamma = 1, \quad \alpha = 3 \quad . \tag{8}$$

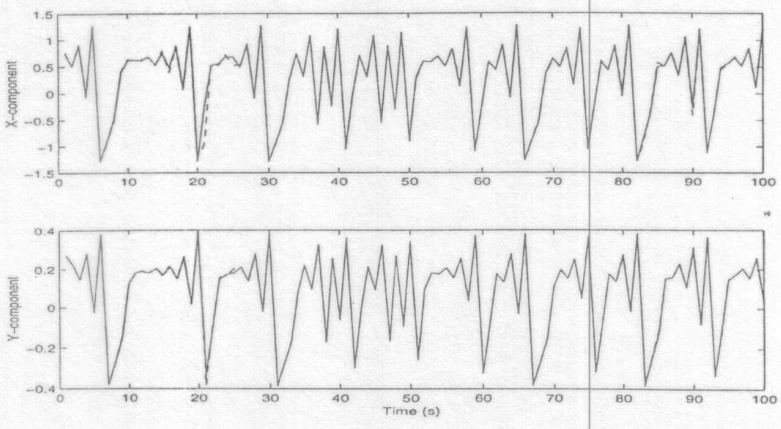

Figure 3: Data assimilation in Hénon system using Adaptive Extended Kalman Filter.

$f(.)$ describes the dynamics of the system, H_{n+1} is the observation matrix, Q_0 the initial covariance of modelling error, R_n the covariance of observation error and P_0^a the initial covariance of assimilation error. P_0^{qa}, γ and α are parameters adjusted for the adaptation process, in which Q_n is assumed to be constant for $n > 0$. The result inserting data at *all* timesteps of the simulation is shown in figure 3. After generating the examples a network consisting of

- inputs w_1^f, w_2^f, w_1^o, w_2^o;
- 1 input layer having 2 neurons with activation function $f(x) = \tanh(x)$;
- 1 hidden layer having 2 neurons with activation function $f(x) = \tanh(x)$;
- 1 output layer having 2 neurons with activation function $f(x) = x$;
- outputs w_1^a, w_2^a;

was trained by attaching to its input and output layers respectively the pairs (x_n^i, w_n^a) formed by the vectors

$$x_n^i = \left[\begin{array}{cccc} w_1^f(n) & w_2^f(n) & w_1^o(n) & w_2^o(n) \end{array} \right]^T, \quad w_n^a = \left[\begin{array}{cc} w_1^a(n) & w_2^a(n) \end{array} \right]^T \quad . \tag{9}$$

for all n. The network was trained with constant learning rate $\eta = 10^{-4}$ until $e_m < 0.01$. To test the new assimilation method one attaches to the input of the trained network the vectors

$$x_n^i = \left[\begin{array}{cccc} X_n^f & Y_n^f & X_n^o & Y_n^o \end{array} \right]^T \tag{10}$$

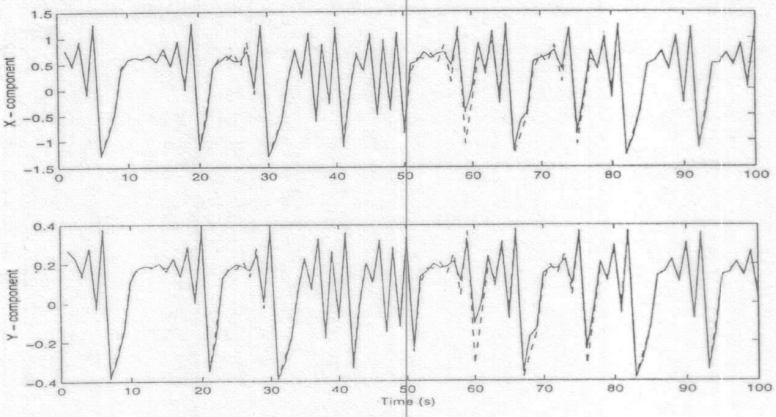

Figure 4: Data assimilation in Hénon system using neural network.

where X_n^f and Y_n^f are predicted by the Hénon equations and X_n^o and Y_n^o are observations. Then one reads at the output layer the assimilation vectors

$$\hat{w}_n^a = \left[\begin{array}{cc} \hat{X}_n^a & \hat{Y}_n^a \end{array} \right]^T \quad . \tag{11}$$

The result inserting data at each $\Delta t = 2\ s$ can be seen in figure 4. The dashed lines represent the true signal, the solid lines the predicted signal. Although the assimilation was not perfect the chaotic deviation seen in previous figure 1 appears to have been avoided.

4.3 Results obtained with the chaotic Lorenz system

Writing the Lorenz system as

$$\frac{dw}{dt} = g(w) \quad , \tag{12}$$

to calculate $w_{n+1} = f\,(w_n)$ model (12) is integrated using the Euler explicit method:

$$w_{n+1} = w_n + \Delta t \cdot g(w_n) = f(w_n) \quad .$$

The filter was implemented using:

$$w_{n+1} \;=\; f\,(w_n)\,, \quad F_n \doteq \left.\frac{\partial f}{\partial w_n}\right|_{w_n = w_n^a} \quad , \quad H_{n+1} \doteq I, \quad Q_n \doteq 0 \tag{13}$$

$$R_n \;\doteq\; \left[\begin{array}{ccc} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{array} \right], \quad P_0^a \doteq 10 \times \left[\begin{array}{ccc} X_0^2 & 0 & 0 \\ 0 & Y_0^2 & 0 \\ 0 & 0 & Z_0^2 \end{array} \right] \quad . \tag{14}$$

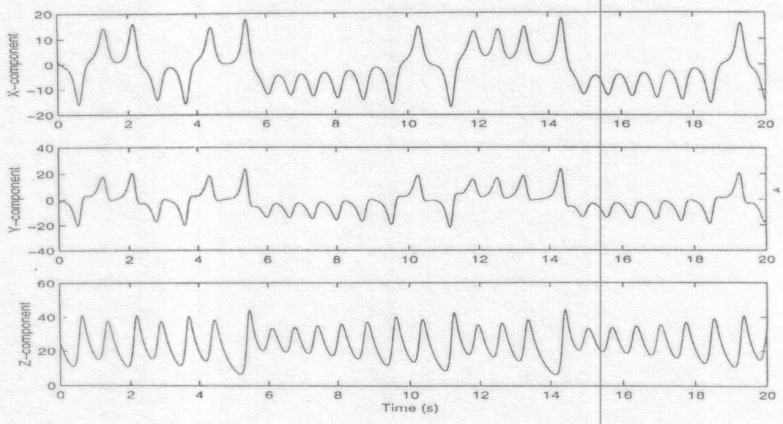

Figure 5: Data assimilation in Lorenz system using Extended Kalman Filter.

To calculate $w_{n+1} = f(w_n)$ the prediction model (12) is integrated using Euler predictor-corrector. The result inserting data at *all* timesteps of the simulation is shown in figure 5. After generating the examples a network composed of

- inputs w_1^f, w_2^f, w_3^f, w_1^a, w_2^a, w_3^a;
- 1 input layer having 2 neurons with activation function $f(x) = \tanh(x)$;
- 1 hidden layer having 2 neurons with activation function $f(x) = \tanh(x)$;
- 1 output layer having 2 neurons with activation function $f(x) = x$;
- outputs w_1^a, w_2^a, w_3^a;

was trained by attaching to its input and output layers respectively the pairs (x_n^i, w_n^a) formed by the vectors

$$x_n^i = \left[\begin{array}{cccccc} w_1^f(n) & w_2^f(n) & w_3^f(n) & w_1^a(n) & w_2^a(n) & w_3^a(n) \end{array} \right]^T \cdot \frac{1}{40} \tag{15}$$

$$w_n^a = \left[\begin{array}{ccc} w_1^a(n) & w_2^a(n) & w_3^a(n) \end{array} \right] \cdot \frac{1}{40} \quad . \tag{16}$$

for all n. The vectors are scaled so that the input and output signals are in the range $[-1, 1]$. The network was trained with constant $\eta = 10^{-4}$ until $e_m < 10$. To test the new assimilation method one attaches to the input of the trained network the vectors

$$x_n^i = \left[\begin{array}{cccccc} X_n^f & Y_n^f & Z_n^f & X_n^a & Y_n^a & Y_n^a \end{array} \right]^T \cdot \frac{1}{40} \tag{17}$$

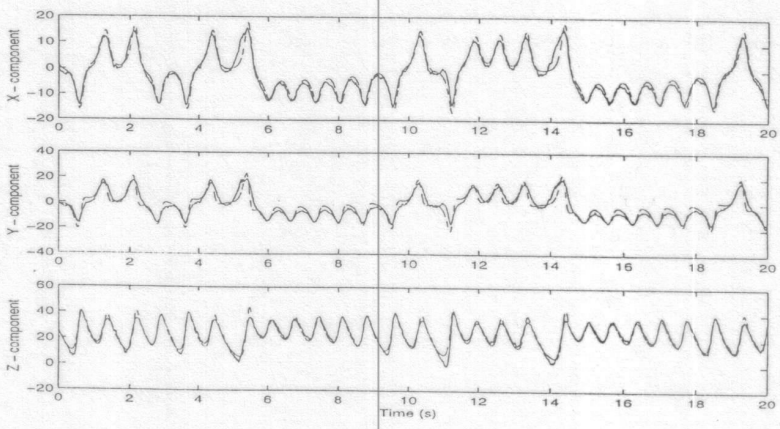

Figure 6: Data assimilation in Lorenz system using neural network.

and reads at the output layer the vectors

$$\hat{w}_n^a = \left[\begin{array}{cc} \hat{X}_n^a & \hat{Y}_n^a \end{array} \right]^T \quad . \tag{18}$$

X_n^f and Y_n^f are predicted by the Lorenz equations and X_n^o and Y_n^o are observations. The assimilation output will actually be

$$\tilde{X}_n^a = 40 \times \hat{X}_n^a \quad , \quad \tilde{Y}_n^a = 40 \times \hat{Y}_n^a \quad , \quad \tilde{Z}_n^a = 40 \times \hat{Z}_n^a \quad . \tag{19}$$

The result inserting data at each $\Delta t = 12\ s$ can be seen in figure 6. The dashed lines represent the true signal, the solid lines the predicted signal. Although the assimilation was not perfect the chaotic deviation seen in previous figure 2 appears to have been avoided.

Suppose now that a network with m_0 inputs and L layers of m_1 neurons has been trained to emulate a Kalman Filter. The assimilation function implemented by the network has complexity of order (Terada, 1982)

$$m_1 O\left(m_0\right) + (L-1) m_1 O\left(m_1\right) \quad . \tag{20}$$

If it is assumed that $O\left(m_0\right) = O\left(m_1\right)$ and that $L \ll m_0$ then the algorithm to calculate $w_n^a = F_{\mathrm{ANN}}(w_n^f, w_n^o)$ will be of complexity $O(m_0^2)$. On the other hand, the complexity of a standard Kalman Filter with m_0 state variables and m_0 observables is $O(m_0^3)$ due to the matrix products at every step.

5 Conclusions

Multilayer Perceptron Neural Networks are used for data assimilation in two nonlinear dynamic systems: the Hénon and Lorenz systems in chaotic state. This aproach emulated Kalman Filter data assimilation

methods avoiding recalculation of the gain matrix at each instant of assimilation. In the case of Hénon system an Adaptive Extended Kalman Filter was used to provide examples for network training. In the case of Lorenz system the Extended Kalman Filter was used for network training. The preliminary results obtained were promising. It was also shown that the complexity of the resulting assimilation function is smaller than that of the standard Kalman Filter. Kalman Filters provided the training sets for the networks, but other assimilation methods can also be used to train multilayer perceptrons.

6 Acknowledgements

This research has been partially supported by a FAPESP grant.

References

[1] A.F. Bennet (1992): *Inverse Methods in Physical Oceanography*, Cambridge University Press.

[2] R. Daley (1991): *Atmospheric Data Analysis*, Cambridge University Press, Cambridge, EUA.

[3] A. Gelb (1978): *Applied Optimal Estimation*, Fourth edition, The M.I.T. Press, Cambridge, MA, EUA.

[4] S. Haykin (1994): *Neural Networks: A Comprehensive Foubdation*, Macmillan, New York.

[5] R.N. Miller, M. Ghil, F. Gauthiez (1994): Advanced Data Assimilation in Strongly Nonlinear Dynamical Systems, *Journal of the Atmospheric Sciences*, 51(8), pp. 1037-1056.

[6] A.G. Nowosad, A. Rios Neto, H.F. Campos Velho (1999): Data Assimilation Using an Adaptative Kalman Filter and Laplace Transform, *Workshop on Physics of the Planetary Boundary Layer and Dispersion Process Modelling*, 23-26 November, Santa Maria (RS), Brasil - Proceedings will be published as special issue of the journal Hybrid Methods in Engineering.

[7] A. Rios Neto, H.K. Kuga (1985): Kalman Filtering State Noise Adaptive Estimation, *2nd IASTED International Conference on Telecommunication and Control* (TELECON'85), Rio de Janeiro, Brazil, pp. 210-213, 10-13 December.

[8] R. Terada (1982): *Desenvolvimento de Algoritmos e Complexidade de Computação*, Terceira Escola de Computação, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.

[9] R. Todling (1997): *Estimation Theory and Foundations of Data Assimilation*, Course notes, National Laboratory of Scientific Computation, 22-29 September, Rio de Janeiro (RJ), Brasil.

[10] J.M.T. Thompson, H.B. Stewart (1991): *Nonlinear Dynamics and Chaos: Geometric Methods for Engineers and Scientists*, John Wiley & Sons.

[11] P. Zannetti (1990): Air Pollution Modeling, Computational Mechanics Publications, UK.