APLICATION OF FEEDFORWARD AND RECURRENT NEURAL NETWORKS FOR FLAPPING AND TORSION IDENTIFICATION OF A HELICOPTER BLADE

Luciane de F. Rodrigues de Souza
lfrs@sc.usp.br

Daniela Cristina Rebolho
danielar@sc.usp.br

Alexandre Simião Caporali
caporali@usp.br

Eduardo Morgado Belo
belo@sc.usp.br

Flávio Donizeti Marques
fmarques@sc.usp.br

Aeroelasticity, Flight Dynamics and Control Laboratory – LADinC
School of Engineering of São Carlos – University of São Paulo (USP)
Av. Trabalhador Sancarlense, 400.
CEP: 13560-970 – São Carlos, SP, Brazil.

Abstract. *The identification of non linear systems is an experimental assessment, consisting of the development of techniques for dynamic systems models estimation from experimental data, demanding no previous knowledge of the procces. In this work a comparative study of the bilinear dynamics identification of a helicopter rotating single-blade mathematical model, carried through artificial neural networks, will be presented. The results of the identification are obtained by feedforward multilayer time delay neural networks (TDNN) and by recurrent neural networks. The feedforward multilayer networks that use delayed inputs are applied to problems involving secular processing, since the delayed signals form a memory device. The recurrent networks are networks that possess one or more feedback connections, being local if it is done at a neuron level or global if it encloses one or more complete layers. To illustrate the performance of the networks, the model, first will be executed conventionally and later with the neural networks. To compare the performance of the networks, the signal-noise relations of the estimated values by the networks will be calculated. In order to do a qualitative analysis, the return maps with the simulation results generated by the network will be plotted.*

Keywords. *Identification, helicopter blade, time delay neural network, recurrent neural networks.*

1. Introduction

Aeroelastic instabilities are factors that can limit the capacity of flight of an aircraft and, therefore, must be considered during the planning and development stages of an aircraft project. More and more, modern aircraft are requested to fly at higher speeds and to have less weight what increases its structural flexibility. Therefore, a safe analysis of the fluid-structure interactions must be taken in order to have a dynamic model with all these relevant characteristics (Belo *et al*, 2001).

Helicopters are aircraft with rotating wings which makes them more complex. An example of this complexity is the development of the main rotor. Rotor blades that are slender and flexible can suffer elastic deformations even in normal operation conditions. Such elastic deformations can go beyond the theoretical limits considered by hypothesis for the treatment as linear beams (Celi, 1999). In addition to that, many other factors make the linear models inadequate for the necessary analyses so, their substitution by non linear ones has been done, fact that has been facilitated due to the increasing availability of faster and more powerful computers. However, the mathematical modeling of non linear systems is considerably more complex than linear ones, becoming some times impracticable for short-term and even some times unfeasible.

Due to the difficulty to represent non linear systems by analytical models, there has been an increase of works on identification of systems. The identification of systems is the art of mathematical modeling dynamic systems using only their input and output data, having applications in some areas as modal identification of flexible structures and detection of non-linearities.

Some representations used in non linear systems modelling are: i) Neural networks; ii) Functions of radial base; iii) Series of Volterra; iv) Wavelets; v) polynomial and rational functions and vi) polynomial differential equations. It is worth to note that the bilinear models constitute a special class of non-linear polynomial models (Aguirre, 1998).

According to Cruz (1998), Artificial Neural Networks, due to its learning capacity as well as to its parallel way to process data, appear as another powerful tool, allowing the modeling of the processes by identification using for that their input and output data. Beyond all these advantages, they possess reasonably high processing speed compared to other conventional methods as well as learning capacity in some way similar to the human one.

Narendra and Parthasarathy (1990) were the ones who clearly stated the problem of the applicability of multilayer neural networks for identification and subsequent use of them to control non-linear dynamic systems. Takahashi (1999) presented a multilayer neural network, trained using the backpropagation algorithm to detect the critical aerodynamic loading for the occurrence of flutter and the limit conditions in the structure. Maghami *et al* (2000) presented a new procedure for developing and training artificial neural networks, useful for fast and efficient control as well as analysis of flexible space systems. In Greenwood (1997), the long time performance of the multilayer networks applied to dynamic systems behaviour estimation could be verified. Giannakis *et al* (2001) presented a list of bibliographical references related to identification of non-linear systems and its applications including a good number of existing references until the year 2000. In Tsoi (1998), a reasonably complete review on recurrent neural networks can be found.

In this context, the proposal of this work is the comparison of the results presented by neural networks with delays in the time with those presented by recurrent neural networks during the identification of the bilinear dynamics of a helicopter blade when in hover flight. This blade is considered rotating and presenting the flapping, torsion and stretching deformations, so that one could next propose a technique to reduce and control its vibrations.

2. Artificial Neural Network

The motivation to use artificial neural networks is mainly due to its substantially increased speed reached in the analysis of processes because of its parallel processing capacity, its ability to approach functional relations specifically the non linear ones and its implementation easiness. In order that a neural network is able to catch the dynamics of a system, it must be temporal and for this it must possess memory. There are two ways to place memory in a network: considering delays in the time in the neurons inputs or considering feedback connections. Therefore, systems that possess complex dynamics can be identified by neural networks with appropriate input delays in the time and with feedback connections (Sparks, 1998).

2.1. Feedforward Neural Networks

A feedforward neural network is constituted by processing units called nodes or neurons, since their functioning is similar to the neurons of the living beings. The commonest neuron model is the *perceptron*, in which the input signals are multiplied by the weights (which are a measure of the importance of the connections in the network) and added. Then this result is compared with the *bias* (or *threshold*) value, resulting in the activation potential to which is applied an activation function that produces the neuron output. Among the most useful activation functions are the sigmoid ones, for example the hyperbolic tangent.

The neuron links make a great network, being its main characteristic the parallel and distributed processing ability as already mentioned.

An artificial neural network can have only one layer or multiple layers of neurons, as the example shown in Fig. (1) and the signals can be spread out in only one direction or can also fed back the input layer or another intermediate one.

Various algorithms exist to train a neural network, however one of the most used is *backpropagation* one, which follows the paradigm of supervised learning, in which the neural network must reproduce a output generated by the supervisor, for a certain input (Crivelaro, 1999). Usually, the network inputs and outputs are normalized and defined in a closed interval [0,1] or [-1,1] (Haykin, 1994). This algorithm appeared with the introduction of the multilayer *perceptron*. Its main characteristic is the method of reducing the error (difference between the desired signal and the signal generated by the network) by backward propagating the error signal (method of the delta rule), during the iterations, that define the learning time. In this algorithm, the learning tax parameter specifies the way the weights must be corrected when an error exists. When the quadratic sum of all network errors reaches a global minimum, one can say that the network is trained. Therefore, training is an iterative process, which must converge to a correct solution. However, during training, a period can exist where the process is unstable and does not converge, or converges to a wrong solution for being stopped in a local minimum. This problem can be solved by introducing random disturbances in the calculated corrections.

Once trained, one can assume that the network stored the knowledge supplied to it. However, the knowledge in a neural network is not stored in a particular localization. It depends on its topology and the importance of the weights in the input layer.

The generalization of an artificial neural network is the capacity to reproduce the desired signals for different input signals that have not been used during the network training, or either, that it is able to catch the dynamics of the system being emulated (Saravanan, 1994).

2.2. Recurrent Neural Network

Recurrent networks are neural networks with one or more feedback connections that can be of local or global nature. Feedback allows that the recurrent networks acquire state representations, what make them appropriate devices for different dynamic applications such as: forecast or modeling of non-linear systems, adaptive equalization of communication channels, control of industrial installations, diagnostic of automotive engines and processing of temporal signals as the voice signal (Haykin, 1994).

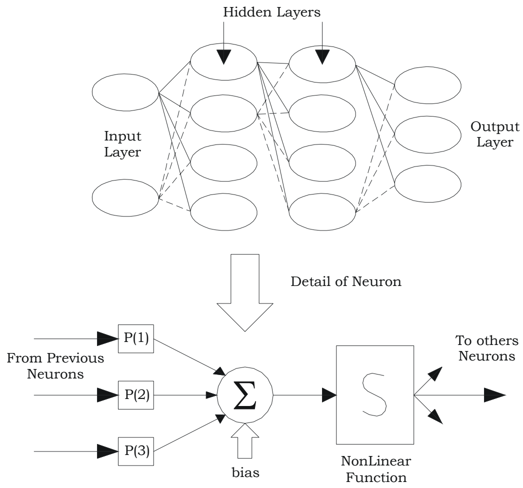


Figure 1. Scheme of a multilayer neural network

Due to the beneficial effects of feedback, the recurrent neural networks can perform better in dynamic applications when compared to the networks with delay in the time without the mentioned feedbacks. This occurs, since the use of feedback has the potential to reduce significantly computer memory requirements.

2.2.1. Types of Recurrent Neural Networks

While there is only a general architecture of multilayer *perceptron* (MLP), various architectures of recurrent neural networks have been proposed by various groups of researchers (Tsoi, 1998). In this work, attention will be given to the recurrent neural networks that map input vectors into output vectors, since these architectures are generally used to identify temporal signals.

2.2.1.1 NARX (Nonlinear Autoregressive with Exogenous Inputs)

NARX network is related as non linear autoregressive model with external inputs (Lin *et al*., 1996).

In this model of recurrent network the output feedback signal is delayed M_r times while the input signal presents M_e delays. The delays of the input signals and feedback ones can be equal or not.

The output of a NARX network is a function of the current external input with its M_e previous inputs together with its M_r previous outputs as one can see in Eq.(1).

$$y(k+1) = F(y(k),...,y(k-M_r);x(k),...,x(k-M_e)) \quad (1)$$

Model NARX encloses an important class of discrete time non linear systems (Leontaritis & Billings, 1985). In the context of neural networks, this model is discussed by Chen *et al*. (1990), Narendra & Partasarathy (1990), Lin *et al*. (1996) and Sieglemann *et al*. (1997).

It was demonstrated that NARX model is well suitable for modeling non-linear systems as heat exchangers (Chen *et al*., 1990), served water treatment plants (Su & McAvoy,1991), non-linear oscillations associated with locomotion by multiple legs in biological systems (Venkataraman, 1994) and grammatical inference (Giles & Horne, 1994).

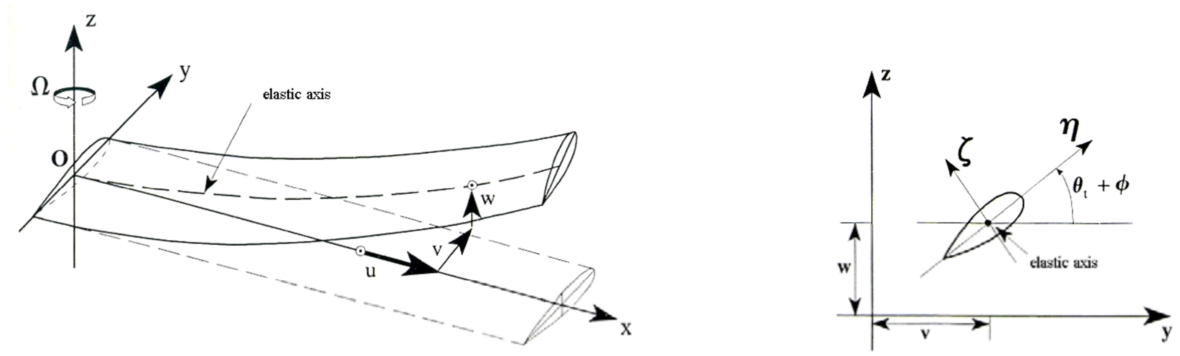
NARX model is also referred to as Non-Linear Auto Regressive Moving Average (NARMA), with the moving average related to the inputs.

3. The helicopter blade non-linear mathematical model

3.1. Mathematical Modeling

The blade studied here is modeled as a rotating cantilever beam with length R, undergoing the coupling motions of flapping, lead-lagging, axial stretching and torsion and it was based on Marques (1993). A pretwist angle θ_t adopted in the model is considered null in the blade root and varying linearly through the span. It is also supposed that elastic and mass axes are noncoincidents.

The main coordinate systems of the blade model are shown in Figure (2a) and (2b). Figure (2a) shows the main coordinate system *x*, *y* and *z*, that is fixed in the blade root with its origin in the intersection of blade root cross-section and elastic axis. When the blade is not deformed the *x*-axis is exactly coincident with the elastic axis. Figure (2a) also shows the deformed blade and elastic displacement u, v and w, in the x, y and z directions, respectively. Figure (2b) shows an arbitrary blade cross-section and its local coordinate system η and ζ. The torsional deflection ϕ, due to the blade deformation can also be seen.



(a) (b)

Figure 2 – (a) Blade coordinate system and elastic displacement and (b) cross-sectional coordinate system.

3.1.1. Strain and kinetic energy

The strain energy, considering a rotating beam undergoing axial stress, shear in the lead-lagging plane and in the flapping plane, is given by:

$$U = \frac{1}{2}\int_0^R \left\{ EAu'^2 + EI_z\left(v''\cos\theta_t + w''\sin\theta_t\right)^2 + EI_y\left(-v''\sin\theta_t + w''\cos\theta_t\right)^2 + GJ\phi'^2 + F_c v'^2 + F_c w'^2 \right\} dx \quad (2)$$

where EA, EI_y, EI_z and GJ are the axial, lead-lagging, flapping and torsional stiffness, respectively. The term F_c is the centrifugal effect and is a function of the mass (m) and the blade rotational speed (Ω):

$$F_c = \int_x^R \Omega^2 m x dx \quad (3)$$

To obtain the kinetic energy expression, the approach presented by Magari *et al*. (1988) is also used here.

$$T = \frac{1}{2}\int_0^R \left\{ \iint_A \left(\rho d\eta d\zeta\right) \left(\frac{d\vec{r}}{dt} \cdot \frac{d\vec{r}}{dt} \right) \right\} dx \quad (4)$$

where, $\frac{d\vec{r}}{dt}$ is the velocity vector of an arbitrary point in the blade cross-section.

3.1.2. Aerodynamic Loading

The steady aerodynamic approach was adopted to yield the expressions of lift (L), drag (D) and aerodynamic moment (M) in the hovering flight condition. Some simplifications were adopted. The first one is neglecting the induced velocity, which yields a free airflow velocity parallel to the y-axis. The small displacement consideration results in the assumption that the blade cross-section remains parallel to the yz plane. There is no coincidence between mass and elastic axes, but the aerodynamic center is taken at the same point of the elastic axis and cross-section intersection. The profile NACA 0015 was assumed; therefore the aerodynamic and the pressure center of the blade cross-section are the same.

A blade element of dx length was taken and the corresponding load element was calculated. Considering that the blade elastic displacements in the free air flow and supposing an operational region of the blade angle of attack, an expression representing the aerodynamic loading results as follows:

$$\begin{bmatrix} D \\ L \\ M \end{bmatrix} = \frac{1}{2}\rho_{ar} c \begin{bmatrix} C_{D\alpha} \\ C_{L\alpha} \\ C_{L\alpha} e \end{bmatrix} \int_x^R \left(\theta_P + \Theta_0 + \theta_t + \phi\right)\left\{\left[\dot{u} - \Omega v\right]^2 + \left[\dot{v} + \Omega\left(x + u\right)\right]^2 + \dot{w}^2\right\} dx \quad (5)$$

where, e is the offset between elastic and mass axis; ρ_{ar} is the air mass density; c is the blade cross-section chord; θ_P is the command pitch angle; Θ_0 is the nominal value of pitch angle in the operational region (10° in this work).

Proper linearization can be achieved by supposing, for instance, small displacements and neglecting higher order terms. Nonetheless, it is desired to maintain some degree of nonlinearity. Here, mildly nonlinear effect can be attained by keeping coupling terms, in special, those relating the states and input variables. Such hypothesis provides a typical bilinear model representation of a dynamic system. A bilinear system is a nonlinear system representation that presents linear behavior of the states and linear behavior of the control inputs, but not linear when states and control inputs are considered coupled. Perhaps bilinear systems are the simplest class of non-linear systems (Marques, 1999).

The blade mathematical model was developed applying the finite elements method, due to its efficiency and proven applicability in modeling flexible structures like helicopter blades. The final mathematical model results in the following equation of motion in matrix form:

$$\mathbf{M\ddot{q}} + \mathbf{G\dot{q}} + \mathbf{Kq} = \mathbf{Q(q, \dot{q}, \theta_P, q\theta_P)} \tag{6}$$

Then the equation of motion can be conveniently transformed into state space representation, given by:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\{\mathbf{Q_1 x}(t) + \mathbf{Q_2 u}(t) + \mathbf{Q_3 u}(t)\mathbf{x}(t)\} \tag{7}$$

After grouping some terms to place the equation in the most traditional state space representation form, Eq. (8) is obtained:

$$\dot{\mathbf{x}}(t) = \mathbf{A_1}\ \mathbf{x}(t) + \mathbf{B_1}\ \mathbf{u}(t) + \mathbf{N}\ \mathbf{u}(t)\,\mathbf{x}(t)\,, \tag{8}$$

where, $\mathbf{A_1} = \mathbf{A} + \mathbf{B\,Q_1}$, $\mathbf{B_1} = \mathbf{B\,Q_2}$, $\mathbf{N} = \mathbf{B\,Q_3}$, **x** is the states vector, **u** is the control inputs vector, **A** is the dynamic matrix, **B** is the control matrix and **N** is the bilinear couplings between states and inputs of control matrix, or either, the matrix of bilinear terms.

4. Simulation of the Model

4.1. Blade simulation

The models closest to real systems, such as the non-linear ones, do not possess direct solution. In this case, the usual procedure is to build a simulation model of the system to get its time responses by integrating the system motion equations (using, for example, Runge-Kutta methods). There are various simulation packages, such as *Simulink*® that can generate the once required responses for analysis in the time domain.

In this work, *Simulink*® was used to simulate flapping and torsion at the blade tip which presented a good performance. In the beginning, there has been some problems due to simulation time, since the model was quite extensive. However, aiming optimization of the process, the blade was discretized using 5 finite elements, instead of 10 elements, since another technique for non-linear model reduction would be difficult to immediately obtain. Comparisons between the simulation results of the two models (with 5 and 10 finite elements) have been made, and the conclusion was that there was no considerable loss of precision when considering only 5 finite elements.

Figure (3) presents a block diagram developed under *Simulink*® environment.

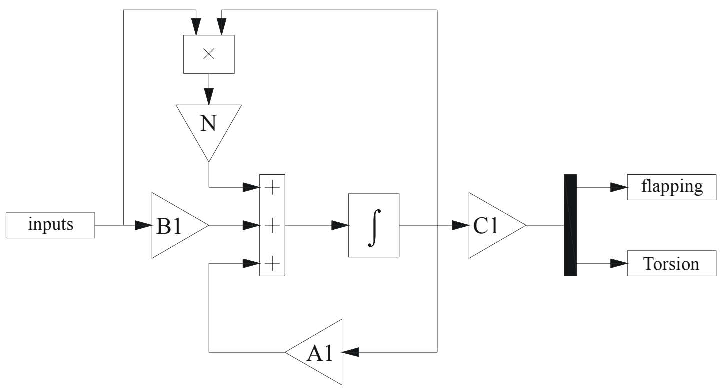


Figure 3. Block diagram used for simulation under S*imulink*® environment

4.2. Modeling of the blade by artificial neural networks

Focusing the non-linear behaviour identification of a single-blade mathematical model of a helicopter under hover flight, two topologies of artificial neural networks have been trained. The mathematical model was represented in bilinear form, implemented and simulated in *Matlab/Simulink®* environment. Firstly, a feedforward multilayer neural network with delays in the time was trained, being its structure presented in Fig. (4).

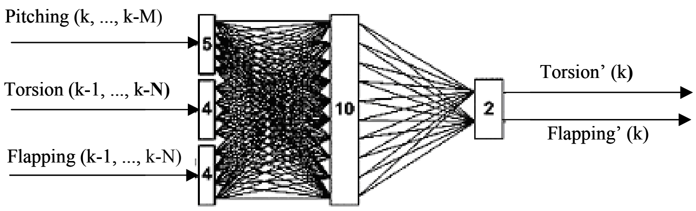


Figure 4. Feedforward multilayer neural network with delays in the time used to foresee the torsional and flapping motion values at the helicopter blade tip

To train the network, the current and previous signals of the blade rotation, M, as well as 4 previous flapping and torsion signals, N, at the blade tip were used to estimate the flapping and torsion ones in the current instant (k). Eq. (9) shows a representation of the input vector containing the values of the rotation angle, the blade tip flapping and torsion at their respective instants of time. The instants α and β are random values for the training of the network and vary from M to 4000.

$$\text{Input} = \begin{bmatrix} \text{Time}(1) & \text{Time}(2) & \dots & \text{Time}(\alpha) & \dots & \text{Time}(\beta) & \dots & \text{Time}(4000) \\ \text{Pitch}(1) & \text{Pitch}(2) & \dots & \text{Pitch}(\alpha) & \dots & \text{Pitch}(\beta) & \dots & \text{Pitch}(4000) \\ \text{Flapping}(1) & \text{Flapping}(2) & \dots & \text{Flapping}(\alpha) & \dots & \text{Flapping}(\beta) & \dots & \text{Flapping}(4000) \\ \text{Torsion}(1) & \text{Torsion}(2) & \dots & \text{Torsion}(\alpha) & \dots & \text{Torsion}(\beta) & \dots & \text{Torsion}(4000) \end{bmatrix} \tag{9}$$

The input pattern for training is given by:

$$\text{Pattern} = \overbrace{\begin{bmatrix} \text{Pitch}(\alpha) & \cdots & \text{Pitch}(\beta) \\ \text{Pitch}(\alpha-1) & \cdots & \text{Pitch}(\beta-1) \\ \vdots & \cdots & \vdots \\ \text{Pitch}(\alpha-\text{M}) & \cdots & \text{Pitch}(\beta-\text{M}) \\ \text{Flapping}(\alpha-1) & \cdots & \text{Flapping}(\beta-1) \\ \vdots & \cdots & \vdots \\ \text{Flapping}(\alpha-\text{N}) & \cdots & \text{Flapping}(\beta-\text{N}) \\ \text{Torsion}(\alpha-1) & \cdots & \text{Torsion}(\beta-1) \\ \vdots & \cdots & \vdots \\ \text{Torsion}(\alpha-\text{N}) & \cdots & \text{Torsion}(\beta-\text{N}) \end{bmatrix}}^{\text{Nt patterns}}$$

A recurrent NARX network, shown in the Fig (5) was also implemented. It can be observed that the output of a NARX network is a function of the current external input together with its previous inputs and outputs. The same data set, used to train the already shown above network with delays in the time, was used to train this one, adding N=4 previous values (taken from the estimated ones) to the network input configuring in this way a recurrent network.

5. Results

To train the network with delays in the time, 800 standards have been used (Nt) each one containing 13 samples (rotation, flapping and torsion) and each standard was obtained from the matrix of inputs from an α variable randomly chosen between M and 4000.

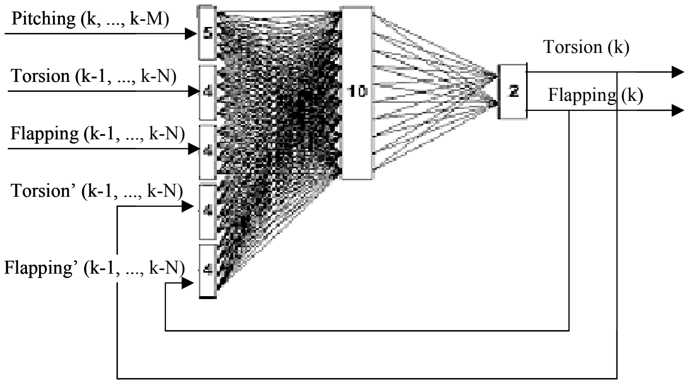


Figure 5. Scheme of the recurrent NARX network used to foresee the torsional and flapping values at the helicopter blade tip.

To verify the generalization of the network all the possible standards the inputs matrix can supply have been applied, i.e. Nt= 4000-M standards and in this case time sequentially ordered. Again, each standard contains 13 samples with information about the helicopter blade rotation angle, flapping and torsion.

The sequence of the target values used for the network training can be represented as:

$$\text{Target} = \overbrace{\begin{bmatrix} \text{Flapping}(\alpha) & \text{Flapping}(\beta) \\ \text{Torsion}(\alpha) & \text{Torsion}(\beta) \end{bmatrix}}^{800 \text{ samples}}$$

The presented results have been sufficiently satisfactory and in Fig. (6) some results of the generalization tests are presented. In Fig. (7), the results obtained by the recurrent neural network are presented.

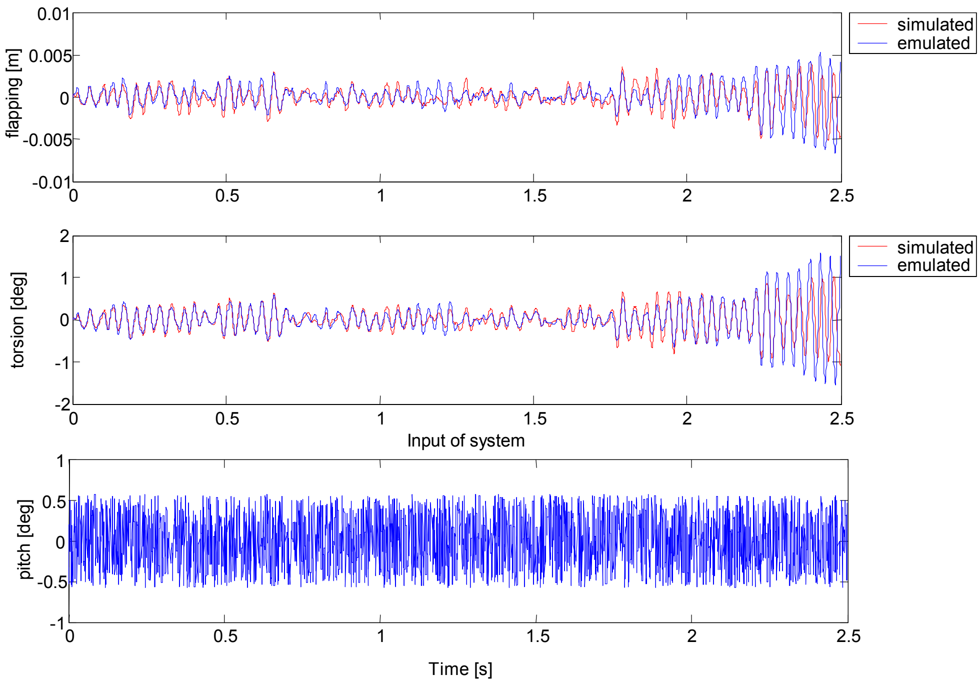


Figure 6. Flapping and torsion results of the neural network with delays in the time generalization test.

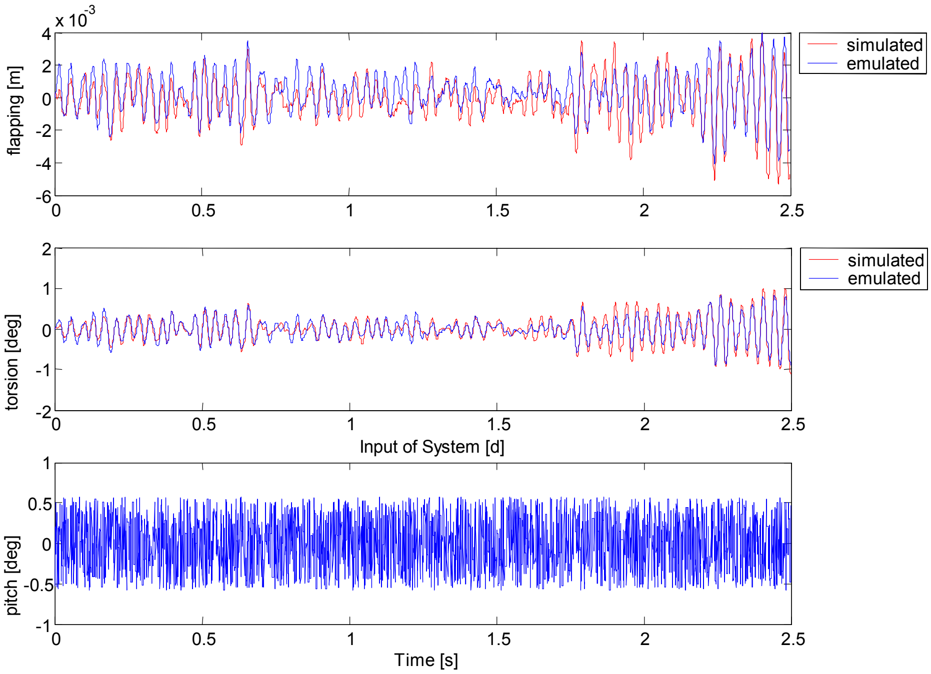


Figure 7. Flapping and torsion results of the recurrent neural network generalization test.

5.1. A qualitative analysis of the identification process

5.1.1 Iterative maps

According to Greenwood (1997), a dimension map is a function of the form:

$$x^{(n+1)} = M[x^{(n)}] \tag{10}$$

where $x^{(n)}$ is the value of x for time n. The sequence $\{x^{(0)}\ x^{(1)}\ x^{(2)}\ldots\}$ is called orbit of $x^{(n)}$. $x^{(n+1)}$ can recursively be defined by the the map iteration for n values of time if $x^{(0)}$ is known, being able to be denoted by $M^{n}[x^{(0)}]$ which can be calculated as follows:

$$M^{n}[x(0)] = \underbrace{M[M(\ldots\{M[x^{(0)}]\}\ldots)]}_{\text{n tempos}} \,. \tag{11}$$

There are many types of orbits, such as the fixed point orbit which is one of the most important and must satisfy the condition $x_0=M^n[x^{(0)}]$. Another type of orbit is the *periodic orbit*. A periodic orbit of period p has points that repeat all iterations p. The set of points to which an arbitrary orbit eventually converges is called *atrator*. *Atratores* are called strange when they possess an intriguing geometry and not entire dimension.

5.1.2 Return Maps

A particularly simple example and for this reason one of the most two-dimensional maps used, is the Return Map. Given the data $x^{(n)}$ of any temporal series, $x^{(n)}$ is plotted against $x^{(n+1)}$. The evolution will show the richness and complicateness, or either, how much complex is the system behavior shown by the trajectory of the points with time passing.

The Return maps of the two outputs considered in this work were plotted. Fig. (8) presents a comparison between the Return maps of flapping at the helicopter blade tip generated by simulation with, respectively, the results generated by the neural network with delays in the time and the recurrent one. In the same way, Fig. (9) presents the Return maps of the torsion results.

One can observe that, either the maps plotted with the simulation results or the maps plotted with the results generated by the networks, are quite similar, what means that identification quality was good, i.e., the neural networks made a good identification.

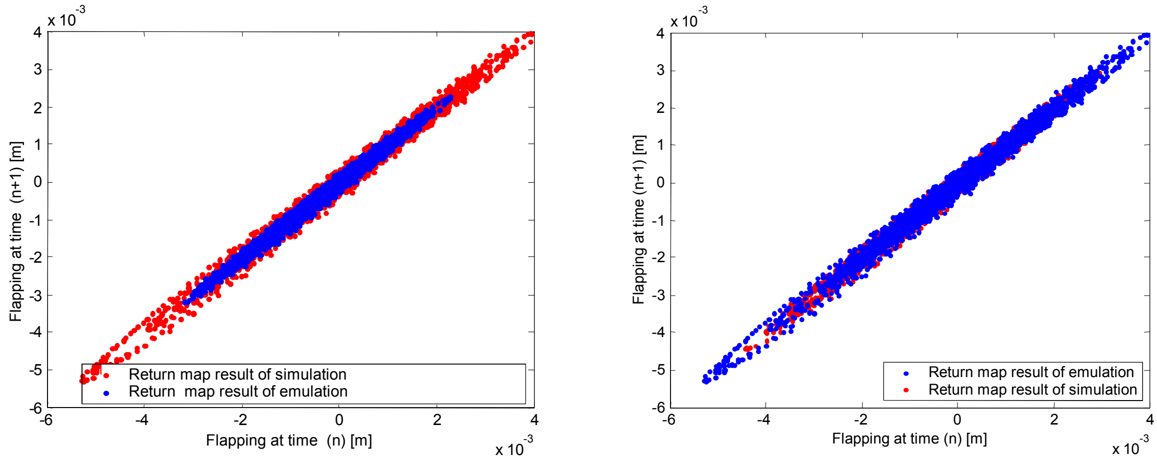


Figure 8. Return maps built with the flapping results at the helicopter blade tip obtained from simulation and from networks emulations; the left map with results from the network with delays in the time and the right one with results from the recurrent network.

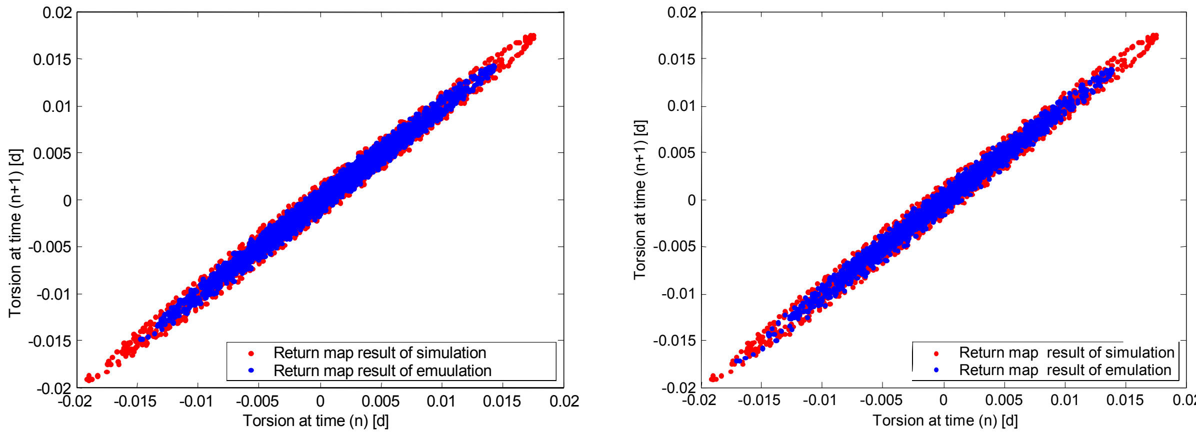


Figure 9. Return maps built with the torsion results at the helicopter blade tip obtained from simulation and from networks emulations; the left map with results from the network with delays in the time and the right one with results from the recurrent network.

6. Conclusions

This work has presented a specific application of artificial neural networks for fast and efficient identification of non-linear flexible dynamic systems. In particular, neural networks with delays in the time and recurrent ones, both with an intermediate layer of neurons, had been designed to approach the bilinear dynamics of a helicopter blade.

After adequately trained, the neural networks supplied satisfactory approaches for both desired outputs, which are flapping and torsion at the helicopter blade tip, noting only that the recurrent neural network resulted in a better identification of the system. Some generalization tests have been carried out and the results were satisfactory, but to certify that the network identified well the system, the return maps with simulated and emulated results were plotted, having been verified the proximity of these, proving the satisfactory result of the identification process.

7. Acknowledgement

The authors wish to acknowledge the financial support of the Brazilian Research Agencies CNPq and CAPES for this research.

8. References

Aguirre, L. A.; Rodrigues, G. G. and Jácome C. R. F., 1998, “Identificação de Sistemas não Lineares Utilizando Modelos NARMAX Polinomiais – Uma Revisão e Novos Resultados”, SBA Controle e Automação, vol. 9, No 2, pp 90-106.

Belo, E. M. and Souza, L. F. R., 2001, “Identificação do Comportamento Não Linear de um Aerofólio Flexível”, In: 22nd Iberian Latin-American Congress on Computational Methods in Engeering, Campinas,Brazil, (CD ROM).

Celi, R., 1999, “Recent Applications of Design Optimization of Rotorcraft”, A survey. 55^{nd} Annual Forum at the American Helicopter Society, Montréal, Canada, May 25-27.
Chen, S.; Billings, S. and Grant, P., 1990, “Nonlinear System Identification Using Neural Networks”, International Journal of Control, Vol.51, No. 6, pp.1191-1214.
Crivelaro, K. C. O., 1999, “Identificação de Regimes de Escoamento Bifásico Horizontal com Auxílio de Redes Neurais”, Dissertação de Mestrado, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos,Brazil.
Cruz, J.C.G., 1998 “Identificação de Uma Torre de Retificação de Águas Ácidas Usando Redes Neurais Artificiais”,. Dissertação de Mestrado, Universidade Federal de Minas Gerais, Minas Gerais, Brazil.
Giannakis, G. B.; Serpedin, E., 2001, “A bibliography on Nonlinear System Identification”, Elsevier Science, Vol.81, pp.533-580.
Giles, C.L.and Hornes, B.G., 1994, “Representation of Learning in Recurrent Neural Network Architectures”, *Proc.* Eighth Yale Workshop on Adaptive and Learning Systems, pp. 128-134, Yale University, New Haven, Ct.
Greenwood, G. W., 1997, “Training Multiple-Layer Perceptrons to Recognize Attractors”, IEEE Transactions on evolutionary computational, Vol.1, No.4, p. 244-248.
Haykin, S., 1994, “Neural Network a Comprehensive Foundation”, Macmillan College Publishing Company, New York.
Leontaritis, I. and Billings, S., 1985, “Input- Output Parametric Models for Nonlinear Systems: Part I: Deterministic Nonlinear Systems”, International Journal of Control, Vol. 41, pp. 303-328.
Lin, T.; Horne, B.G.; Tino, P. and Giles, C.L., 1996, “Learning Long Term Dependencies in NARX Recurrent Neural Networks”, IEEE Trans on Neural Networks, Vol. 7, pp. 1329-1338.
Maghami, P. G. and Sparks, D. W., 2000, “Design of Neural Networks for Fast Convergence and Accuracy”, IEEE Transactions on Neural networks, Vol.11, n.1, pp.113-123.
Marques, F. D.,1993, “Controle de Vibrações em uma Pá de Helicóptero”, Dissertação de Mestrado. Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, Brazil.
Marques, F.D. and Belo, E. M., 1999, “Bilinear System Representation of an Aircraft Dynamics”, DINAME 99 Applied Mechanics in the Americas, Vol.8, p.1393-1396.
Narendra, K.P. and Partasarathy, K., 1990, “Identification and Control of Dynamical Systems using Neural Networks”, IEEE Trans Neural Networks, Vol. 1, pp. 4-27.
Saravanan, N.and Duyar, A., 1994, “Modeling Space Shuttle Main Engine Using Feed-Forward Neural Networks”,. Journal of Guidance, Control and Dynamics, v. 17, n.4, pp.641-648.
Sieglemann, H.; Horne, B.; Giles, L., 1997, “Computational Capabilities or Recurrent NARX Neural Networks”, IEEE Trans System, Man and Cybernetics. Part B”, Vol 27, pp. 208-218.
Souza, L. F. R.; Gruppioni, E. M.; Belo and E. M.; Marques, F. D., 2002, “Identificação da Dinâmica Não Linear de uma Pá de Helicóptero via Redes Neurais”, In: II Congresso Nacional de Engenharia Mecânica, João Pessoa, (CD ROM).
Sparks, D. W. and Maghami, P. G., 1998, “Neural Networks for Rapid Design and Analysis”, AIAA Journal, Vol.39, pp. 20-23.
Su, H.T.; MacAvoy, T. and Werbos, P., 1991, “Long Term Prediction of Chemical Progress using Recurrent Neural Network: A parallel Training Approach”, Industrial Engineering and Chemical Research, Vol. 31, pp. 1338-1352.
Takahashi, I. 1999, “Identification for Critical Flutter Load and Boundary Conditions of a Beam using Neural Networks”, Journal of Sound and Vibration, Vol.228, n.4, pp.857-870.
Tsoi, A.C. 1998, “Recurrent Neural Network Architectures an Overview”, In: Cirles, C.L. & Ciani, M. Ed. Adaptive Processes of Sequences and Data Structure, Springe Velas Berlin Heidelberg, New Yale. pp 1-26.