C. Brillante, A. Mannarino
Improvement of Aeroelastic Vehicles Performance Through Recurrent Neural Network Controllers
Nonlinear Dynamics, Vol. 84, N. 3, 2016, p. 1479-1495
doi:10.1007/s11071-015-2583-2

Nonlinear Dynamics manuscript No.
(will be inserted by the editor)

Improvement of aeroelastic vehicles performance through recurrent neural networks controllers

Claudio Brillante · Andrea Mannarino

September 9, 2015

Abstract Aeroelastic systems have the peculiarity of changing their behavior with flight conditions. Within such a view, it is difficult to design a single control law capable of efficiently working at different flight conditions. Moreover, control laws are often designed on simple linearized, low-fidelity models. A fact introducing the need of a scheduled tuning over a wide operational range. Obviously such a design process can be time consuming, because of the high number of simulations and flight tests required to assure high performance and robustness. The present work aims at proving the high flexibility of neural network-based controllers, testing their adaptive properties when applied to typical fixed and rotary wing aircraft problems. At first the proposed control strategy will be used to suppress the limit cycle oscillations experienced by a rigid wing in transonic regime. Then as a second example, a controller with the same structure will be employed to reduce the hub vibrations of an helicopter rotor with active twist blades.

Keywords Neural networks · Multibody · Cosimulation · Aeroservoelasticity · Nonlinear behavior.

1 INTRODUCTION

The improvement of aircraft performance through active control systems is a well established research and industrial topic. As different sources state [1–3], the next generation of flight control systems will utilize adaptive and non-deterministic techniques to provide more stable and maneuvrable aircrafts. The study of the interaction of structure compliance, aerodynamic loads and control laws involves many diverse disciplines and, when considering fixed wing aircraft, deals with problems such as wing flutter, buffeting, divergence, control surface effectiveness and reversal, buzz and gust loads [4].

Aeroelastic interactions have to be considered in rotary-wing aircrafts as well. Apart from the classical air resonance and whirl flutter, which is typical for tiltrotors, the challenge in the design of modern and more comfortable helicopters involves the reduction of noise and vibrations due to the asymmetry of the wind perceived by the main rotor. Vibratory loads trasmitted by the main rotor to the fuselage represent nowadays one of the most important problem for rotary wing aircrafts in forward flight condition, because they can cause crew and passenger discomfort and reduce the airframe fatigue life.

The analysis of the stability properties of an aeroelastic system is of utmost importance for any aircraft designer. In fact, these systems change their behavior when some key parameters, e.g. flight speed, Mach number, are changed. Numerous studies that deal with such an issue are present in the literature, considering the aircraft as a whole system, as in [4,5] and references therein, or only its sub-parts, as in [6]. Nonlinear effects may play an important role in aeroelastic response, possibly shaping limit cycle oscillations instead of exponentially diverging responses [7–12]. The control of these unwanted phenomena is essential for preserving the structural integrity of the system and to avoid possible fatigue failures. Different approaches aimed at suppressing aeroelastic vibration can be found in the literature. Classical LQG design [13], robust $\mathcal{H}_\infty$ frame-

C. Brillante
Via La Masa, 34, 20156, Milan, Italy
E-mail: claudio.brillante@polimi.it

A. Mannarino
Via La Masa, 34, 20156, Milan, Italy
E-mail: andrea.mannarino@polimi.it

work [14], input limiting [15,16], feedback linearization [17], sliding mode control [18], immersion and invariance approach [19] and neural networks, both static [20], recurrent [21,22] and within the so called dynamic surface framework [23,24], have been succesfully tested on numerical and experimental models of fixed wing aircraft.

On the other hand, the reduction of vibratory loads in rotary-wing aircraft has been subject of many studies and a suitable strategy aimed at eliminating this problem involves active control to modify the related aerodynamic periodic loads at harmonic frequencies above the rotational frequency. Although the rotor system exhibits a highly nonlinear behavior, due to both the structural dynamics of the blades, which usually require a nonlinear ad hoc formulation as presented in [25], and the aerodynamics involved, the most used approaches rely on linear control theory. The well known Higher-Harmonic-Control [26], both in its classic and adaptive version, considers a quasi-static model of the helicopter rotor and requires the identification of the transfer matrix between the harmonic components of the control signal and the hub loads. More sophisticated techniques exploits linear periodic control theory to better take into account the periodicity of the rotor in forward flight [27–29].

Since aeroelastic systems change their properties in relation to the considered flight condition, an efficient control law should be able to carry out its functions over the whole flight envelope without losing its efficiency. To achieve such a result, two different approaches can be considered: control law scheduling and adaptive control. While the first approach has proved to be very robust and it is currently widely employed in service [30], the latter has not seen any operational deployment yet. This fact leaves open the space for researching of optimal and robust solutions, and to support the ongoing certification process [1].

Actually, there are two possible ways to realize an adaptive control strategy. In one, a learning algorithm is used to compute flight control inputs to augment the controls produced by a non-adaptive flight controller, while the other approach uses a system identification algorithm to compute the gain parameters used by the flight controller.
In this work a neural controller is designed using a recurrent network characterized by on-line learning. The proposed method can be classified within the latter approach mentioned, since the controller is composed by two networks. The first one is used to identify the dynamics of the plant under control and the information so obtained is fed to the second network which acts as a controller. Thanks to its learning capability, the system under control can be identified on-line and no model is required to design the controller. This adaptive black-box approach permits to avoid scheduling procedures to cover the whole flight envelope and reduces the approximations related to linearizations used to describe the dynamic behavior of the system.

Although this work takes the basis from [21,22], several innovations are introduced throughout the paper. First of all the neural network formulation, presented in Section 2, differs deeply from the classical ones found in the literature. It relies on a compact computational framework based on matrix-vector computations, and this permits a simpler analysis and implementation of the control law. Differently from [20], a recurrent formulation is adopted, allowing to use a smaller number of neurons and therefore leading to a smaller computational time. This fact is of great importance when real-time implementations are considered [22].
Moreover, such a controller is tested on quite realistic problems, where the systems dynamics is more complex than the classical two degree-of-freedom system studied in adaptive control demonstrations [31]. The interest in controllers based on neural networks is growing for helicopter applications, for example in [32] a nonlinear adaptive control with a neural network compensator is proposed for trajectory tracking of a model-scaled helicopter. Therefore the application of this kind of controller in the field of helicopters vibrations suppression may also be considered innovative. Thanks to the good results obtained here, such an adaptive nonlinear controller can be seen as a valid alternative to the classical HHC, which works in the frequency domain and it could lose efficacy when strong nonlinearities occur.
Last but not the least, this work set the basis of an integrated aeroservoelastic toolbox, where softwares describing different problems are interfaced. This permits the design of control systems on models with a good level of accuracy, thus considering possible nonlinear effects, and limited computational effort.

The main goal of the present effort is to prove the high flexibility of neural network-based controllers, analyzing their behavior when applied to typical fixed and rotary wing aircraft problems. At first the proposed control strategy will be used to suppress the limit cycle oscillations experienced by a wing in transonic regime, where the motion of the shock waves on the body surface introduces strong nonlinearities in the system. In the second example, a controller with the same structure will be employed to reduce the hub vibrations of an helicopter rotor by active twist blades. In this case, nonlinear effects have been considered through a multibody flexible simulation.

1.1 Software Environment

Appropriate softwares have to be used, while designing an aircraft, so to consider the aeroelastic coupling properly. Unsteady and nonlinear effects in the aerodynamic field are very important for the dynamic response. In particular the flutter instability is typical for linear systems, which predict a destructive behavior beyond a critical free stream velocity. By introducing the nonlinear modeling of the aerodynamics, the instabilities develop in a different way, and often limit cycle oscillations appear.
Other nonlinearities, which have to be considered in some applications, such as that of the helicopter rotor, are related to the structural response. In fact large displacements and complex motion of the structure introduce a severe inertial coupling, demanding an adequate nonlinear structural modeling.
In this section we present a short description of the software later used to perform the simulations.

- **AeroFoam**: is a density-based compressible Unsteady Euler/RANS solver, with the Euler option being selected in this work. It has been developed at Dipartimento di Scienze e Tecnologie Aerospaziali (DSTA), of Politecnico di Milano [33]. Among its features there is an aeroelastic interfacing scheme, based on a moving least square interpolation strategy, providing all the needed functionalities to set the appropriate aerodynamic boundary conditions imposed by a deforming structure, while driving a connected hierarchical mesh deformation within an Arbitrary Lagrangian Eulerian formulation. An extended discussion of its aeroelastic capabilities can be found in [34].
- **MBDyn**: is a general purpose multibody software developed at DSTA [35]. This code deals with Initial Value Problems (IVP) by solving Differential Algebraic Equations (DAE). The equations of motion are based on Newton-Euler principles and the kinematic constraints are enforced through Lagrange's multipliers. It is also possible to model flexible bodies as nonlinear beams and plates with the full constitutive law of the material, complex structural part/components being cared through modal substructures. A simple aerodynamic module based on strip theory and linear induced velocity models is also available.

Different approaches have been used in the following numerical examples to perform closed loop simulations. In the first example the controller is implemented in C++ into the AeroFoam software while in the second one it is implemented in the Simulink environment and subsequently coupled with MBDyn as explained in the next section.

2 CONTROL METHODOLOGY

2.1 Neural networks

A Neural Network (NN) is a massively parallel distributed process made up of simple processing units (the neurons) that have a natural capability to store the knowledge accumulated through experience, making it available for later uses. Knowledge is acquired through a *learning process*, and it is stored in the synaptic connections linking the neurons.
A general network is composed by an input layer, where the input data are stored and passed to the computational units, represented by the neurons, which receive a linear combination of the input, whose coupling coefficients are called *synaptic weights* of the network, and use this input as the argument of a nonlinear, limited function, in this work assumed to be the hyperbolic tangent, located in the *hidden* layer.
Because of the *universal approximation theorem*, a neural network with one hidden layer is sufficient for approximating a nonlinear function with arbitrary accuracy, ensuring that the number of neurons is adequate [36]. In order to represent a dynamic process, the network should be able to learn its evolution path from the input-output data pairs only. The *memory* is included through a time delay applied to the hidden neurons, defining thus the *state* of the network. Basically, a network state is a hidden neuron which is connected to another hidden neuron. Such a network is defined as *Recurrent* (RNN). In this case the input layer is composed by the set of input data, now function of time, and the set of the delayed output of the hidden neurons. A graphical representation of such a recurrent network is given in Figure 1. An RNN presents many advantages over static networks, e.g. a smaller size, reduced computational cost and a faster learning: all characteristics that are very important for real-time control problems [22,21,36].
A mathematical description of such a model can be given in the following compact form:

$$\begin{cases} \mathbf{x}_{n+1} &= \boldsymbol{\phi}\left(\mathbf{W}^a\mathbf{x}_n + \mathbf{W}^b\mathbf{u}_n\right) \\ \mathbf{y}_n &= \mathbf{W}^c\mathbf{x}_n \end{cases} \tag{1}$$

where $\mathbf{x}_n \in \mathbb{R}^{n_x}$ is the network state at time t_n, $\mathbf{u} \in \mathbb{R}^m$ is the network input, $\mathbf{y}_n \in \mathbb{R}^p$ is the network output, $\boldsymbol{\phi} : \mathbb{R}^{n_x} \longrightarrow \mathbb{R}^{n_x}$ is the set of activation functions, $\mathbf{W}^a \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{W}^b \in \mathbb{R}^{n_x \times m}$ and $\mathbf{W}^c \in \mathbb{R}^{p \times n_x}$ are the matrices containing the network synaptic weights; n_x,

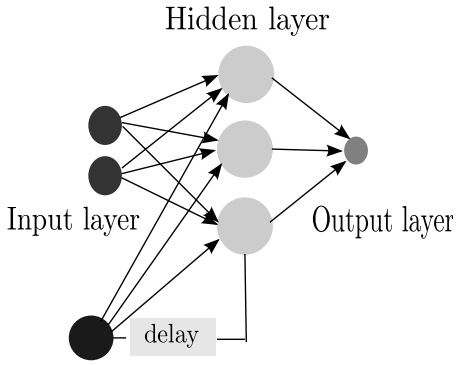


Fig. 1: Example of recurrent neural network

m and p being respectively the state, input and output space dimensions. In practice, n_x represents the number of neurons involved in the network. In general, the network state $\mathbf{x}_n$ has only an internal meaning: no physical interpretation can be associated to it. However, assuming the output matrix $\mathbf{W}^c$ with a fixed structure, able to extract the first p elements of the state, we know that such state elements are actually the physical output of the model, gaining some physical insight of the process at hand [36].

As discussed before, the proposed control approach relies upon two networks, the identifier and the controller.

2.2 Identifier network (ID-RNN)

Let us assume to have available a model of a process capable of producing any sort of output. These are the known, measured, quantities of the model, indicated by $\mathbf{y}_n^M$. Subjected to the same input of the physical model, the ID-RNN should be able to minimize the error between its output and the measured one. The network input and output will be defined on a case by case basis. The training of such a network can be interpreted as an optimization algorithm which minimizes the following cost function:

$$E_n^{\mathrm{ID}} = \frac{1}{2}\mathbf{e}_n^{\mathrm{T}}\mathbf{e}_n, \qquad \mathbf{e}_n = \mathbf{y}_n^{\mathrm{ID}} - \mathbf{y}_n^M \tag{2}$$

Given that $\mathbf{y}_n^{\mathrm{ID}}$ depends on $\mathbf{W}^{a,\mathrm{ID}}$ and $\mathbf{W}^{b,\mathrm{ID}}$, we need an updating strategy for such weights that minimizes the cost function defined in Eq. 2. We use here the Real-Time Recurrent Learning (RTRL) algorithm. Such an algorithm derives its name from the fact that adjustments are made to the synaptic weights of a fully connected recurrent network in real time, i.e. while the network continues to perform its signal-processing function [36,37]. This kind of approach perfectly fits our needs, potentially leading to an accurate description of the plant under control at each sampling time.
Following the algorithm proposed in [36], the synaptic weights of the ID-RNN are so identified. First, let us write the model described by Eq. 1 in the following way:

$$\mathbf{x}_{n+1}^{\mathrm{ID}} = \begin{bmatrix} \phi\left(\mathbf{w}_1^{\mathrm{ID}}\boldsymbol{\xi}_n^{\mathrm{ID}}\right) \\ \phi\left(\mathbf{w}_2^{\mathrm{ID}}\boldsymbol{\xi}_n^{\mathrm{ID}}\right) \\ \vdots \\ \phi\left(\mathbf{w}_j^{\mathrm{ID}}\boldsymbol{\xi}_n^{\mathrm{ID}}\right) \\ \vdots \\ \phi\left(\mathbf{w}_{n_x}^{\mathrm{ID}}\boldsymbol{\xi}_n^{\mathrm{ID}}\right) \end{bmatrix} \tag{3}$$

Where $\mathbf{w}_j^{\mathrm{ID}}$ is the row vector collecting the j-th row of $\mathbf{W}^{a,\mathrm{ID}}$ and $\mathbf{W}^{b,\mathrm{ID}}$, while $\boldsymbol{\xi}$ is the column vector stacking $\mathbf{x}_n^{\mathrm{ID}}$ and $\mathbf{u}_n^{\mathrm{ID}}$:

$$\begin{aligned} \mathbf{w}_j^{\mathrm{ID}} &= \begin{bmatrix} \mathbf{W}_{j,:}^{a,\mathrm{ID}} & \mathbf{W}_{j,:}^{b,\mathrm{ID}} \end{bmatrix} \\ \boldsymbol{\xi}_n^{\mathrm{ID}} &= \begin{bmatrix} \mathbf{x}_n^{\mathrm{ID}} \\ \mathbf{u}_n^{\mathrm{ID}} \end{bmatrix} \end{aligned} \tag{4}$$

Where the subscript $(j,:)$ means 'row j, all columns'.
In order to minimize Eq. 2 by varying the synaptic weights of the identifier network, the simplest choice is a gradient descent approach, that is:

$$\Delta\mathbf{w}_{j,n}^{\mathrm{ID}} = -\eta^{\mathrm{ID}}\frac{\partial E_n^{\mathrm{ID}}}{\partial \mathbf{w}_{j,n}^{\mathrm{ID}}}, \qquad j = 1,2,...,n_x^{\mathrm{ID}} \tag{5}$$

As it can be noted, the weights variation goes in the opposite direction of the gradient of the identification error. The parameter η^{ID} is called learning rate and it is the fundamental parameter governing the ID-RNN training velocity. The learning rate value is limited by stability considerations: too-high values may destabilize the training algorithm, causing the divergence of the synaptic weights. On the other hand, a low learning rate would limit a fast control adaption to system variations. Unfortunately, there is no general theory for predicting the limiting values of η for ID-RNN. So its best value must be tuned through design simulations.
The term $\dfrac{\partial E_n^{\mathrm{ID}}}{\partial \mathbf{w}_{j,n}^{\mathrm{ID}}}$ can be computed explicitly:

$$\begin{aligned} \frac{\partial E_n^{\mathrm{ID}}}{\partial \mathbf{w}_{j,n}^{\mathrm{ID}}} &= \mathbf{e}_n^{\mathrm{T}}\frac{\partial \mathbf{e}_n}{\partial \mathbf{w}_{j,n}^{\mathrm{ID}}} = \mathbf{e}_n^{\mathrm{T}}\frac{\partial \mathbf{y}_n^{\mathrm{ID}}}{\partial \mathbf{w}_{j,n}^{\mathrm{ID}}} = \\ &\mathbf{e}_n^{\mathrm{T}}\mathbf{W}^{c,\mathrm{ID}}\frac{\partial \mathbf{x}_n^{\mathrm{ID}}}{\partial \mathbf{w}_{j,n}^{\mathrm{ID}}} = \mathbf{e}_n^{\mathrm{T}}\mathbf{W}^{c,\mathrm{ID}}\boldsymbol{\Lambda}_{j,n}^{\mathrm{ID}} \end{aligned} \tag{6}$$

Where $\boldsymbol{\Lambda}_{j,n}^{\mathrm{ID}}$ can be interpreted as the network state gradient, i.e. the sensitivity of the state vector $\mathbf{x}_n^{\mathrm{ID}}$ to the weight vector $\mathbf{w}_{j,n}^{\mathrm{ID}}$. Since $\boldsymbol{\Lambda}_{j,n}^{\mathrm{ID}}$ depends on $\mathbf{x}_n^{\mathrm{ID}}$, it is

itself a dynamic quantity governed by a system of difference equations. Such a system can be derived directly from Eq. 3:

$$\frac{\partial \mathbf{x}_{n+1}^{\mathrm{ID}}}{\partial \mathbf{w}_{j,n}^{\mathrm{ID}}} = \mathrm{Diag}\left(..., \phi'\left(\mathbf{w}_j^{\mathrm{ID}} \boldsymbol{\xi}_n^{\mathrm{ID}}\right), ...\right) \cdot \left(\mathbf{W}_n^{a,\mathrm{ID}} \frac{\partial \mathbf{x}_n^{\mathrm{ID}}}{\partial \mathbf{w}_{j,n}^{\mathrm{ID}}} + \begin{bmatrix} \mathbf{0}_{j-1} \\ \boldsymbol{\xi}_n^{\mathrm{ID}^{\mathrm{T}}} \\ \mathbf{0}_{n_x^{\mathrm{ID}}-(j+1)} \end{bmatrix}\right) \tag{7}$$

Where ϕ' is the first derivative of the activation function with respect to its argument and the subscript to the null column vector $\mathbf{0}$ represents its length. Eq. 7 can be compactly rewritten as:

$$\boldsymbol{\Lambda}_{j,n+1}^{\mathrm{ID}} = \boldsymbol{\Phi}_n^{\mathrm{ID}}\left(\mathbf{W}_n^{a,\mathrm{ID}} \boldsymbol{\Lambda}_{j,n}^{\mathrm{ID}} + \mathbf{U}_{j,n}^{\mathrm{ID}}\right) \tag{8}$$

We have now available a complete set of difference equation describing the dynamic evolution of the identifier:

$$\begin{cases} \mathbf{x}_{n+1}^{\mathrm{ID}} & = \boldsymbol{\phi}\left(\mathbf{W}_n^{a,\mathrm{ID}} \mathbf{x}_n^{\mathrm{ID}} + \mathbf{W}_n^{b,\mathrm{ID}} \mathbf{u}_n^{\mathrm{ID}}\right) \\ \boldsymbol{\Lambda}_{j,n+1}^{\mathrm{ID}} & = \boldsymbol{\Phi}_n^{\mathrm{ID}}\left(\mathbf{W}_n^{a,\mathrm{ID}} \boldsymbol{\Lambda}_{j,n}^{\mathrm{ID}} + \mathbf{U}_{j,n}^{\mathrm{ID}}\right) \\ \mathbf{y}_n^{\mathrm{ID}} & = \mathbf{W}^{c,\mathrm{ID}} \mathbf{x}_n^{\mathrm{ID}} \end{cases} \tag{9}$$

where $j = 1, 2, ..., n_x^{\mathrm{ID}}$. With the synaptic weights $\mathbf{W}^{a,\mathrm{ID}}$ and $\mathbf{W}^{b,\mathrm{ID}}$ updated at each sampling time by:

$$\Delta \mathbf{w}_{j,n}^{\mathrm{ID}} = -\eta^{\mathrm{ID}} \mathbf{e}_n^{\mathrm{T}} \mathbf{W}^{c,\mathrm{ID}} \boldsymbol{\Lambda}_{j,n}^{\mathrm{ID}}, \qquad j = 1, 2, ..., n_x^{\mathrm{ID}} \tag{10}$$

The initial conditions of the system described by Eq. 9 are set to $\mathbf{x}_0^{\mathrm{ID}} = \mathbf{0}$ and $\boldsymbol{\Lambda}_{j,0}^{\mathrm{ID}} = \mathbf{0}$, for $j = 1, 2, ..., n_x^{\mathrm{ID}}$. This means that we are not introducing any prior knowledge of the real plant in our ID-RNN. As it will be shown in Section 3, this fact will not cause any instability in the training process, and with a proper tuning of η^{ID} the identifier will replicate accurately the plant output in a few sampling steps.

2.3 Controller network (CO-RNN)

Starting with a basic interpretation, the CO-RNN input will be fed by the outcomes of the ID-RNN, giving a new output value of the control input, which will minimize a certain control cost function. The network dynamics is described by the usual model:

$$\mathbf{x}_{n+1}^{\mathrm{CO}} = \begin{bmatrix} \phi\left(\mathbf{w}_1^{\mathrm{CO}} \boldsymbol{\xi}_n^{\mathrm{CO}}\right) \\ \phi\left(\mathbf{w}_2^{\mathrm{CO}} \boldsymbol{\xi}_n^{\mathrm{CO}}\right) \\ \vdots \\ \phi\left(\mathbf{w}_j^{\mathrm{CO}} \boldsymbol{\xi}_n^{\mathrm{CO}}\right) \\ \vdots \\ \phi\left(\mathbf{w}_{n_x}^{\mathrm{CO}} \boldsymbol{\xi}_n^{\mathrm{CO}}\right) \end{bmatrix} \tag{11}$$

with the obvious meaning of $\mathbf{w}_j^{\mathrm{CO}}$ and $\boldsymbol{\xi}_n^{\mathrm{CO}}$.
Similarly to the ID-RNN case, the training of the controller can be interpreted as an optimization algorithm which minimizes the following cost function:

$$E_n^{\mathrm{CO}} = \frac{1}{2} \mathbf{e}_n^{\mathrm{T}} \mathbf{e}_n + \frac{1}{2} \rho\, \mathbf{y}_n^{\mathrm{CO}^{\mathrm{T}}} \mathbf{y}_n^{\mathrm{CO}}, \qquad \mathbf{e}_n = \mathbf{y}_n^{\mathrm{ID}} - \mathbf{y}_n^{\mathrm{ref}} \tag{12}$$

Where $\mathbf{y}_n^{\mathrm{ref}}$ is a reference output. The parameter ρ is defined as the control penalization parameter and is used to limit the control effort in a way similar to what is done in a classical LQG controller. In an unstable system, all zeros in the right half of the complex plane may be transformed into unstable poles of the controller by a straight system inversion. In this case, the penalty term is necessary to avoid the divergence of the control [22, 21]. The same approach to network training employed for the ID-RNN is used here. However, care must be taken to consider that the dynamics of the ID-RNN and CO-RNN are coupled.
Once again, the gradient descent algorithm is used to minimize the cost function described by Eq. 12:

$$\Delta \mathbf{w}_{j,n}^{\mathrm{CO}} = -\eta^{\mathrm{CO}} \frac{\partial E_n^{\mathrm{CO}}}{\partial \mathbf{w}_{j,n}^{\mathrm{CO}}}, \qquad j = 1, 2, ..., n_x^{\mathrm{CO}} \tag{13}$$

The gradient of the cost function can be derived analytically:

$$\frac{\partial E_n^{\mathrm{CO}}}{\partial \mathbf{w}_{j,n}^{\mathrm{CO}}} = \mathbf{e}_n^{\mathrm{T}} \mathbf{W}^{c,\mathrm{ID}} \frac{\partial \mathbf{x}_n^{\mathrm{ID}}}{\partial \mathbf{w}_{j,n}^{\mathrm{CO}}} + \rho \left(\mathbf{W}^{c,\mathrm{CO}} \mathbf{x}_n^{\mathrm{CO}}\right)^{\mathrm{T}} \mathbf{W}^{c,\mathrm{CO}} \frac{\partial \mathbf{x}_n^{\mathrm{CO}}}{\partial \mathbf{w}_{j,n}^{\mathrm{CO}}} \tag{14}$$

where the presence of the cross term $\frac{\partial \mathbf{x}_n^{\mathrm{ID}}}{\partial \mathbf{w}_{j,n}^{\mathrm{CO}}}$ makes it explicit the coupling between the two networks. Such a sensitivity can be computed directly from the difference equations which describe the ID-RNN dynamics:

$$\frac{\partial \mathbf{x}_n^{\mathrm{ID}}}{\partial \mathbf{w}_{j,n}^{\mathrm{CO}}} = \mathrm{Diag}\left(..., \phi'\left(\mathbf{w}_j^{\mathrm{ID}} \boldsymbol{\xi}_n^{\mathrm{ID}}\right), ...\right) \cdot \left(\mathbf{P}\mathbf{W}^{b,\mathrm{ID}}\right) \mathbf{W}^{c,\mathrm{CO}} \frac{\partial \mathbf{x}_n^{\mathrm{CO}}}{\partial \mathbf{w}_{j,n}^{\mathrm{CO}}} \tag{15}$$

The quantity $\frac{\partial \mathbf{x}_n^{\mathrm{CO}}}{\partial \mathbf{w}_{j,n}^{\mathrm{CO}}}$ is a dynamic variable, and its evolution can be derived directly from Eq. 11:

$$\frac{\partial \mathbf{x}_{n+1}^{\mathrm{CO}}}{\partial \mathbf{w}_{j,n}^{\mathrm{CO}}} = \mathrm{Diag}\left(..., \phi'\left(\mathbf{w}_{j,n}^{\mathrm{CO}} \boldsymbol{\xi}_n^{\mathrm{CO}}\right), ...\right) \cdot \left(\mathbf{W}_n^{a,\mathrm{CO}} \frac{\partial \mathbf{x}_n^{\mathrm{CO}}}{\partial \mathbf{w}_{j,n}^{\mathrm{CO}}} + \begin{bmatrix} \mathbf{0}_{j-1} \\ \boldsymbol{\xi}_n^{\mathrm{CO}^{\mathrm{T}}} \\ \mathbf{0}_{n_x^{\mathrm{CO}}-(j+1)} \end{bmatrix}\right) \tag{16}$$

The matrix $\mathbf{P}$ is a boolean matrix which extracts only the terms of $\mathbf{W}^{b,\mathrm{ID}}$ which multiply the control input $\mathbf{y}_n^{\mathrm{CO}}$ from $\mathbf{u}_n^{\mathrm{ID}}$. In fact, as it will be seen in details in the application section, the input to the ID-RNN may not contain only the control input but also other measured quantities.

Defining $\boldsymbol{\Lambda}_{j,n}^{\mathrm{CO}} = \dfrac{\partial \mathbf{x}_n^{\mathrm{CO}}}{\partial \mathbf{w}_{j,n}^{\mathrm{CO}}}$, Eqs. 15 and 16 can be rewritten in a more compact form:

$$\frac{\partial \mathbf{x}_n^{\mathrm{ID}}}{\partial \mathbf{w}_{j,n}^{\mathrm{CO}}} = \boldsymbol{\Phi}_n^{\mathrm{ID}} \left(\mathbf{P}\mathbf{W}^{b,\mathrm{ID}}\right) \mathbf{W}^{c,\mathrm{CO}} \boldsymbol{\Lambda}_{j,n}^{\mathrm{CO}} \tag{17}$$

$$\boldsymbol{\Lambda}_{j,n+1}^{\mathrm{CO}} = \boldsymbol{\Phi}_n^{\mathrm{CO}} \left(\mathbf{W}_n^{a,\mathrm{CO}} \boldsymbol{\Lambda}_{j,n}^{\mathrm{CO}} + \mathbf{U}_{j,n}^{\mathrm{CO}}\right) \tag{18}$$

By exploiting these compact definitions, Eq. 13 can now be rewritten as:

$$\begin{aligned} \Delta \mathbf{w}_{j,n}^{\mathrm{CO}} = & -\eta^{\mathrm{CO}} \mathbf{e}_n^{\mathrm{T}} \mathbf{W}^{c,\mathrm{ID}} \boldsymbol{\Phi}_n^{\mathrm{ID}} \left(\mathbf{P}\mathbf{W}^{b,\mathrm{ID}}\right) \mathbf{W}^{c,\mathrm{CO}} \boldsymbol{\Lambda}_{j,n}^{\mathrm{CO}} + \\ & -\eta^{\mathrm{CO}} \rho \left(\mathbf{W}^{c,\mathrm{CO}} \mathbf{x}_n^{\mathrm{CO}}\right)^{\mathrm{T}} \mathbf{W}^{c,\mathrm{CO}} \boldsymbol{\Lambda}_{j,n}^{\mathrm{CO}} \end{aligned} \tag{19}$$

where $j = 1, 2, ..., n_x^{\mathrm{CO}}$. We have now available a complete set of difference equations that describes the dynamic evolution of the controller:

$$\begin{cases} \mathbf{x}_{n+1}^{\mathrm{CO}} & = \boldsymbol{\phi} \left(\mathbf{W}_n^{a,\mathrm{CO}} \mathbf{x}_n^{\mathrm{CO}} + \mathbf{W}_n^{b,\mathrm{CO}} \mathbf{u}_n^{\mathrm{CO}}\right) \\ \boldsymbol{\Lambda}_{j,n+1}^{\mathrm{CO}} & = \boldsymbol{\Phi}_n^{\mathrm{CO}} \left(\mathbf{W}_n^{a,\mathrm{CO}} \boldsymbol{\Lambda}_{j,n}^{\mathrm{CO}} + \mathbf{U}_{j,n}^{\mathrm{CO}}\right) \\ \mathbf{y}_n^{\mathrm{CO}} & = \mathbf{W}^{c,\mathrm{CO}} \mathbf{x}_n^{\mathrm{CO}} \end{cases} \tag{20}$$

where $j = 1, 2, ..., n_x^{\mathrm{CO}}$. With the synaptic weights $\mathbf{W}^{a,\mathrm{CO}}$ and $\mathbf{W}^{b,\mathrm{CO}}$ updated at each sampling time by Eq. 19. Similarly to the ID-RNN, the initial conditions of the system described by Eq. 20 are set to $\mathbf{x}_0^{\mathrm{CO}} = \mathbf{0}$ and $\boldsymbol{\Lambda}_{j,0}^{\mathrm{CO}} = \mathbf{0}$, for $j = 1, 2, ..., n_x^{\mathrm{CO}}$. The previously discussed considerations are still valid in this case. At this point the control framework composed by the interconnection of ID-RNN and CO-RNN is ready to be implemented. The adaptivity of such a control system is achieved by keeping the networks training active, so that the control adapts itself to system variations by appropriately modifying the networks weights.

3 Sample applications

3.1 BACT wing

The Benchmark Active Controls Technology (BACT) project was part of NASA Langley Research Center's Benchmark Models Program for studying transonic aeroservoelastic phenomena. The BACT wind-tunnel model was developed to collect high quality unsteady aerodynamic data (pressures and loads) near transonic flutter conditions and to demonstrate the potential of designing and implementing active control systems for flutter suppression using flaps and spoilers [38]. Therefore, it is a well known, easy to use, detailed and fully validated aeroservoelastic model [14,38,39], which has become an often referred benchmark application for verifying nonlinear aerodynamic analyses and active controls design methods. It is composed by an elastically constrained rigid rectangular wing model, with NACA 0012 sections, equipped with a trailing-edge control surface and upper and lower-surface spoilers, which can be controlled independently through well performing hydraulic actuators. Its dynamic behavior is very similar to a classical typical section but, because of its low aspect ratio, it displays a not so simple three-dimensional transonic flow. However, it has been shown, e.g. [39], that the related nonlinear aerodynamic behavior is mild enough to produce slowly growing limit cycle oscillations, which can be verified through high fidelity CFD simulations [40,41]. Because of the above remark, the literature related to the design of active controllers for the BACT wing presents many instances of effective, experimentally validated, applications of linear design techniques, such as: classical and min-max [39], robust $\mathcal{H}_\infty$ and $\mu-$synthesis [14], robust passification [42] and static neural networks [20].

For the design of the active control system it is convenient to adopt an approximated and efficient numerical model, providing satisfactory results within the frequency range of interest, rather than choosing an over-detailed and expensive numerical model. To this end, the linear model proposed in [38], with a quasi-steady approximation of the aerodynamic loads is employed. The need to include more sophisticated aerodynamic modelling options, such as unsteady kernel approximations or time accurate CFD, is albeit mitigated by the fact that the flutter reduced frequency for the BACT wing is relatively low, approximately $k = \dfrac{\omega\, c}{2V_\infty} \approx 0.05$. The aerodynamic derivatives have been estimated through wind tunnel testing, with the exception of the unsteady ones, obtained from computational aerodynamic analysis. Including also the actuator dynamics, as derived in [43], the model so obtained can be written in the following form, where the subscript SE represent the servoelastic part of the model, meanwhile the subscript A represent its aerodynamic part:

$$\begin{aligned} & \mathbf{M}_{\mathrm{SE}} \ddot{\mathbf{q}} + \mathbf{C}_{\mathrm{SE}} \dot{\mathbf{q}} + \mathbf{K}_{\mathrm{SE}} \mathbf{q} = \\ & \left(\frac{c}{2\,V_\infty}\right)^2 \mathbf{M}_{\mathrm{A}} \ddot{\mathbf{q}} + \frac{c}{2\,V_\infty} \mathbf{C}_{\mathrm{A}} \dot{\mathbf{q}} + \mathbf{K}_{\mathrm{A}} \mathbf{q} + \mathbf{B}_s \beta_C \end{aligned} \tag{21}$$

where c is the aerodynamic chord, V_∞ the flight speed and the matrices are detailed as:

$$\mathbf{q} = \begin{Bmatrix} h \\ \theta \\ \beta \\ z \end{Bmatrix} \quad \mathbf{M}_{\mathrm{SE}} = \begin{bmatrix} m & S_{h\theta} & S_{h\beta} & 0 \\ S_{h\theta} & J_{\theta\theta} & I_{\theta\beta} & 0 \\ S_{h\beta} & I_{\theta\beta} & J_{\beta\beta} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{22a}$$

$$\mathbf{C}_{\mathrm{SE}} = \begin{bmatrix} c_{hh} & 0 & 0 & 0 \\ 0 & c_{\theta\theta} & 0 & 0 \\ 0 & 0 & c_{\beta\beta} & 0 \\ 0 & 0 & 0 & 2\xi_{\mathrm{act}}\omega_{0_{\mathrm{act}}} \end{bmatrix} \quad \mathbf{K}_{\mathrm{SE}} = \begin{bmatrix} k_{hh} & 0 & 0 & 0 \\ 0 & k_{\theta\theta} & 0 & 0 \\ 0 & 0 & k_{\beta\beta} & -k_{\beta\beta} \\ 0 & 0 & 0 & \omega^2_{0_{\mathrm{act}}} \end{bmatrix} \tag{22b}$$

$$\mathbf{B}_s = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega^2_{0_{\mathrm{act}}} \end{bmatrix} \quad \mathbf{M}_{\mathrm{A}} = q_\infty \frac{2S}{c} \begin{bmatrix} -C_{L,\dot{\alpha}} & -l\,C_{L,\dot{\alpha}} & 0 & 0 \\ c\,C_{M,\dot{\alpha}} & lc\,C_{M,\dot{\alpha}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{22c}$$

$$\mathbf{C}_{\mathrm{A}} = q_\infty \frac{2S}{c} \begin{bmatrix} -C_{L,\alpha} & -l\,C_{L,\alpha} & -C_{L,\dot{\beta}} & 0 \\ c\,C_{M,\alpha} & lc\,C_{M,\alpha} & c\,C_{M,\dot{\beta}} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{22d}$$

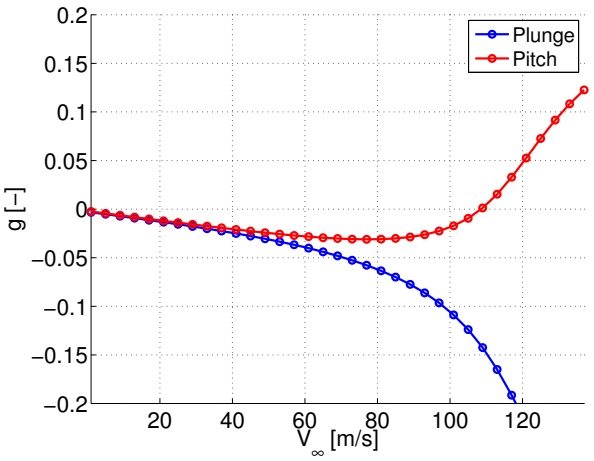
$$\mathbf{K}_A = q_\infty S \begin{bmatrix} 0 & -C_{L,\alpha} & -C_{L,\beta} & 0 \\ 0 & c\,C_{M,\alpha} & c\,C_{M,\beta} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{22e}$$

The related data are available in [38]. The actuator state is here represented by the variable z, $q_\infty = 1/2\rho_\infty V_\infty^2$ is the dynamic pressure, S is the wing surface, l is the relative distance between the elastic axis and the aerodynamic center of the wing and β_C is the control input. The actuator dynamics is included in the model considering also its compliance, according to the frequency-domain formulation:

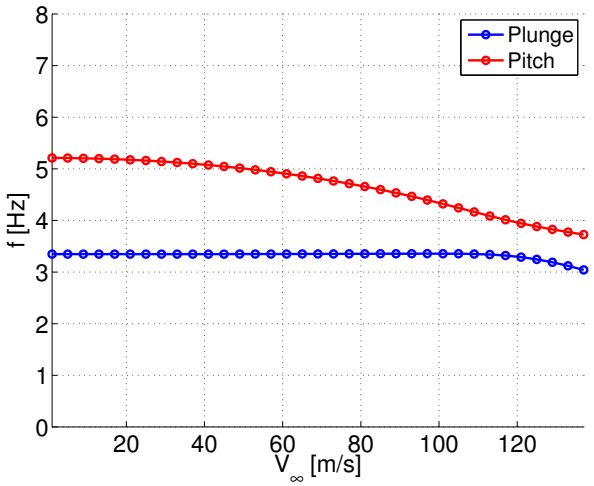
$$\beta(s) = \frac{\omega^2_{0_{\mathrm{act}}}}{s^2 + 2\xi_{\mathrm{act}}\omega_{0_{\mathrm{act}}} s + \omega^2_{0_{\mathrm{act}}}} \beta_C(s) - \frac{M_\beta}{k_{\beta\beta}} \tag{23}$$

M_β being the hinge moment acting on the aileron. The compliance $1/k_{\beta\beta}$ has been set to $k_{\beta\beta} = 2\,k_{\theta\theta}$ to maintain a good level of positioning precision.
Two accelerometers are placed on the wing, one on the leading edge and the other close to the control surface hinge. Their dynamic model is assumed to be represented by a second order model, characterized by an adequate bandwidth, i.e. $\xi = 1$, $\omega_0 = 160$, rad/s. The signals coming from these accelerometers will be our measured output, $\mathbf{y}^{\mathrm{M}}$.

First of all, a linear flutter analysis is performed by varying the flight speed and tracking the pitch and plunge modes. All the simulations are performed at a Mach number $M_\infty = 0.77$, in the heavy gas $R-12$ [38]. The related results are shown in Figure 2. The flutter speed is predicted to be $V_\infty = 108.58$ m/s with an error of 1.3 % with respect to the experimental value [38].
Following the interpretation of [1] we define some standard testing cases for our control law: response to initial conditions, response to input pulses, response to



(a) Aeroelastic damping versus flight speed



(b) Aeroelastic frequency versus flight speed

Fig. 2: Linear flutter analysis

random input with time varying flow velocity and any response evaluated for a slightly different model (variations in mass, inertia, stiffness, etc.). Of particular interest is also the effectiveness of the controller when it is switched on only a certain amount of time after the simulation starts. We choose to employ the first two cases for tuning the control law, while the others are considered in the verification phase. Extensive numerical simulations have been carried out to understand the behavior of the ID-RNN and CO-RNN and to determine the correct networks topology in terms of number of neurons and value of the various parameters involved. In this case, the ID-RNN input is $\mathbf{u}_n^{\mathrm{ID}} = \left\{\mathbf{y}_n^{\mathrm{M}^{\mathrm{T}}}, \beta_{c,n}, -1\right\}^{\mathrm{T}}$, while its output are the identified accelerations. With regard to the CO-RNN, its input is $\mathbf{u}_n^{\mathrm{CO}} = \left\{\mathbf{y}_{n+1}^{\mathrm{ID}^{\mathrm{T}}}, \beta_{c,n}, -1\right\}^{\mathrm{T}}$ and the output is the next control effort $\beta_{C,n+1}$ to be applied to the system. The

bias term -1 is added to each input to improve the network performance in following an eventual constant reference [36]. The results of this numerical campaign are summarized in Table 1.

	ID-RNN	CO-RNN
n_x	6	5
η	$6 \cdot 10^{-2}$	$8 \cdot 10^{-2}$
ρ	-	$4 \cdot 10^{-2}$
Sampling frequency	180 Hz	

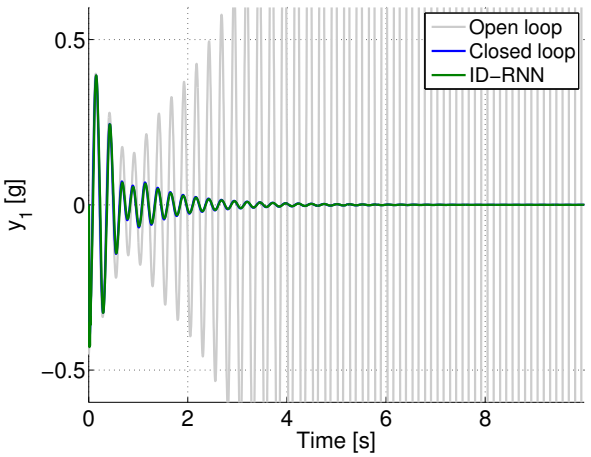
Table 1: Networks parameters

As also reported in [21], the number of neurons affects the networks memory. Increasing n_x^{ID}, the robustness and accuracy of the identified output is increased, as well as the related computational cost is, while the learning rate should be decreased to avoid possible unstable effects. The learning rates have been chosen as a trade-off between fast learning (high η^{ID}) and its stability (low η^{ID}). On the other hand, the behavior of the CO-RNN facing variations of n_x^{CO} is much more complicated, and this is probably due to the coupling between the two networks, well detailed in the training algorithm section. Nevertheless, in the authors' experience, good results are obtained by tuning the networks with the same number of neurons, i.e. $n_x^{\mathrm{ID}} = n_x^{\mathrm{CO}}$, with the same learning rate, i.e. $\eta^{\mathrm{ID}} = \eta^{\mathrm{CO}}$ and a very low value of the control penalty, i.e. $\rho = 1 \cdot 10^{-4}$. The parameters are then varied until a satisfactory solution is achieved.

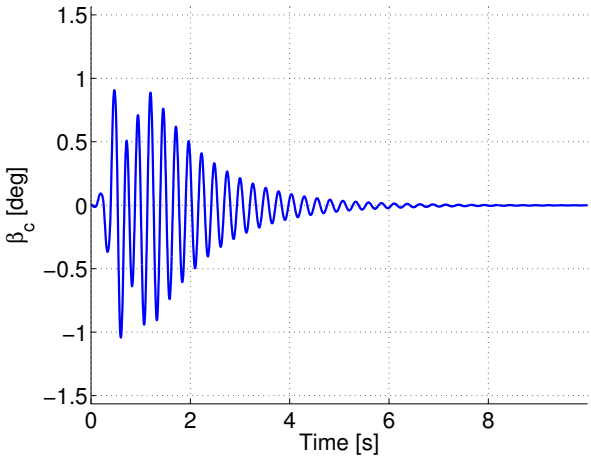
The results obtained in the first two test cases mentioned are displayed in Figures 3 and 4, where the response of the accelerometer placed at the leading edge is shown along with the related control effort. The design speed is $V_\infty = 120$ m/s, well beyond the open loop flutter limit. The initial condition assumed in the first test case is $\mathbf{q}_0 = (0.1, 0, 0, 0)^{\mathrm{T}}$, while in the second case the initial condition is homogenuous and a pulse of amplitude 5 deg is given to the actuator, at time $t = 0.1$ s. Notice that the green line in the plots, which represents the ID-RNN response, perfectly replicates the real plant response.

The performance indexes considered in the design are the settling time, which should be as small as possible, i.e. $T_{\mathrm{set}} \leq 5$ s, and a limited control effort, i.e. $\beta_{C,max} \leq 5$ deg. For the linear model these requirements are satisfied with ample margin.

The high fidelity model is composed by the previous servoelastic model coupled to the CFD solver AeroFoam. After a convergence analysis performed on steady state cases, a computational mesh composed by 103040 cells is employed. The effect of a moving aileron is included by a smooth mesh deformation algorithm based on radial basis functions [34]. At the same design test point as that of the previous analysis, a sample of the pressure field and leading edge acceleration is given in Figure 5. As can be seen from the figures reported above, the effect of a nonlinear aerodynamic model is somewhat beneficial to the dynamics of the whole aeroservoelastic system, because beyond the stability limit the wing oscillations are bounded by a limit cycle and an exponential divergence, typical of linear systems, does not occour. The control law designed in the previous case is now applied without changing any parameter. However, the networks training is kept active in order to achieve the required level of adaptiveness in these verification simulations also. The results relative to the response to non-null initial condition and input noise with varying flight speed are shown in Figures 6 and 7, while the response to an input impulse at

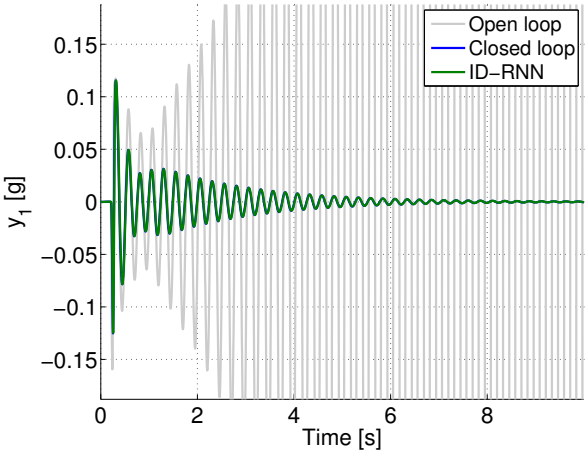


(a) Response at the leading edge

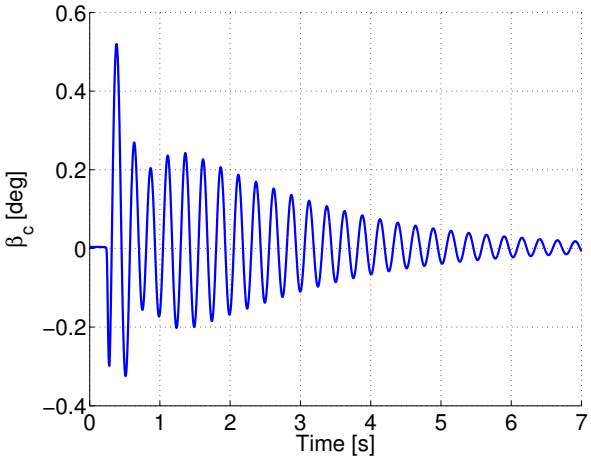


(b) Control effort

Fig. 3: Response to initial condition

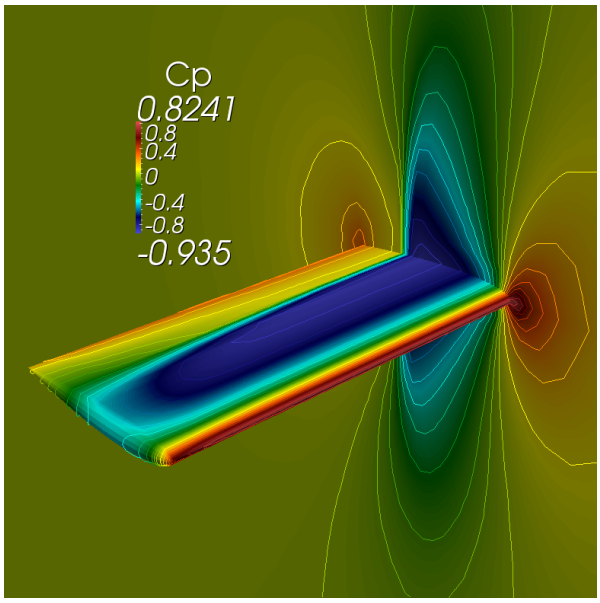


(a) Response at the leading edge

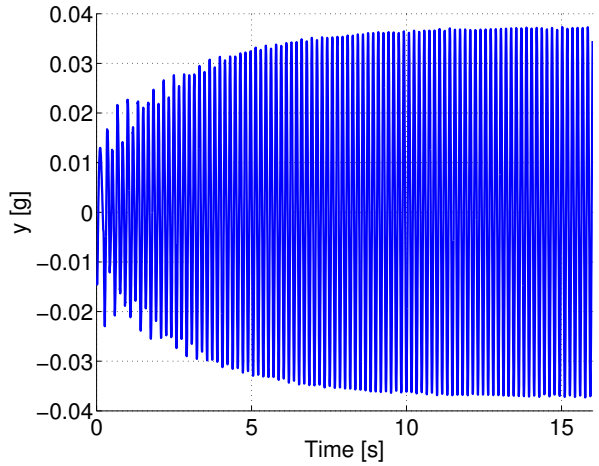


(b) Control effort

Fig. 4: Response to input pulse



(a) Sample of numerical solution



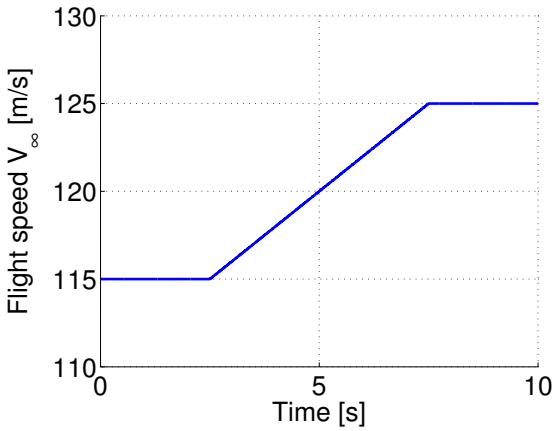
(b) Limit cycle oscillation at $V_\infty = 120$ m/s

Fig. 5: High fidelity representation of the BACT wing

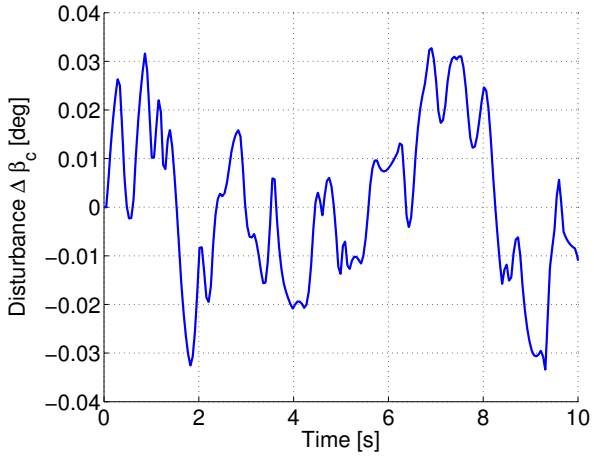
$V_\infty = 125$ m/s, switching on the controller at $t = 0.5$ s, are shown in Figure 8. As it can be seen, the controller designed on the low fidelity model is still able to stabilize efficiently the nonlinear model under operational conditions that were not seen before, a really appreciable feature of neural network-based controllers. To conclude we present a comparison of response and control effort for the linear and nonlinear models, carried out at $V_\infty = 130$ m/s. The case considered is the response to a 1 deg input pulse, applied at $t = 0.2$ s. The results are shown in Figure 9. The two responses are quite different, this may be due to the effects of the nonlinear, unsteady aerodynamic versus a linear, quasi-steady approximation. As a result, the control effort is greater at the first response peaks in the nonlinear case, but with a decay that is much more faster than its linear counterpart. However, the stabilization task is carried out very efficiently in both cases, with a maximum control effort well below the quite conservative limit of 5 deg. Further numerical simulations have been carried out, resulting in a new flutter speed of 134 m/s (25 % greater than the nominal case), achieved while respecting the previously cited limit on the control effort.

3.2 Helicopter vibrations reduction

The neural controller is then applied to an available rotor model of the MBB Bo105 [44] to reduce hub vibrations. Among all solutions for individual blade control (IBC), we focus on active twist design [45,46]. The Bo105 original blades are then replaced with actively twisted ones, in such a way to include macro-fiber composite (MFC) piezoelectric actuators distributed along the blade span. They are oriented at $\pm 45°$ relative to

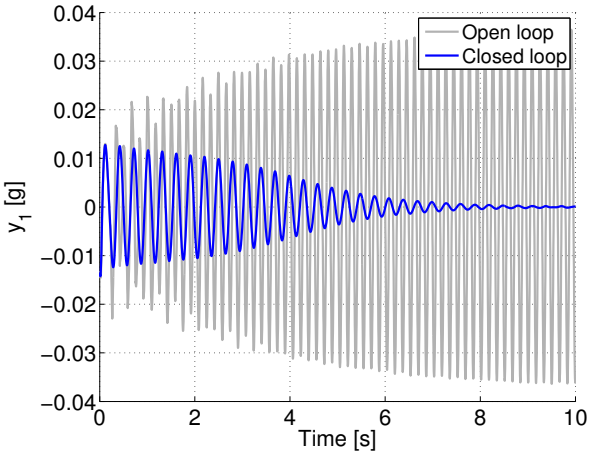


(a) Flight speed ramp

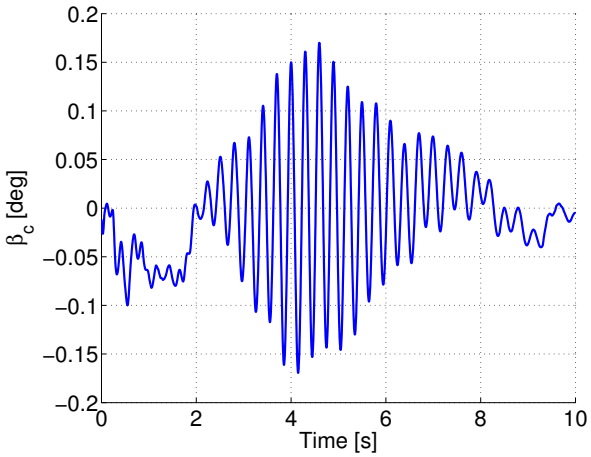


(b) Input noise

Fig. 6: Flight conditions



(a) Response at leading edge



(b) Control effort

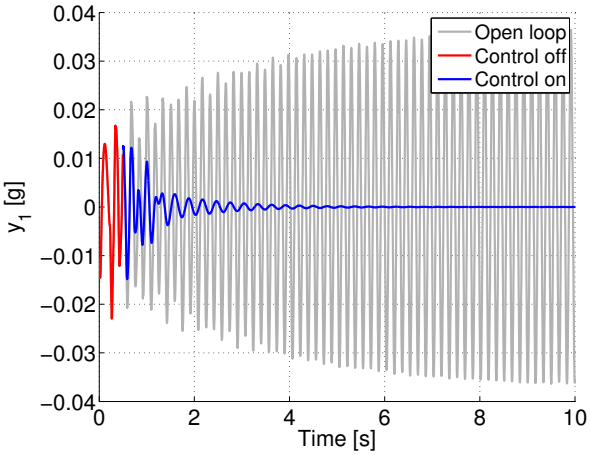
Fig. 7: Response to noise with varying flight speed

the blade spanwise axis, so to generate a high torque control. The flex-beams of the original hingeless rotor are left unchanged.

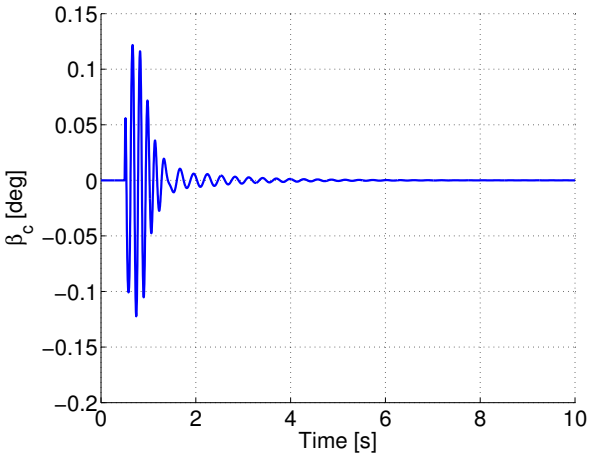
To represent the structural nonlinearities and to reproduce the complex kinematic movement of the rotor system, the multibody flexible model is built using MBDyn [35]. The only flexible bodies are the blades, which are modeled with 5 nonlinear geometrically exact finite volumes beam elements [47]. To compute the beam properties of the piezoelectric actuated blade, the semi-analytical approach, described in [48] and [49], is employed. The beam problem is decomposed into the one-dimensional domain of the classical beam model and a two-dimensional domain of the beam section, which is solved by a finite element based analysis, as shown in Figure 10. The blade section is designed through an optimization procedure to satisfy the constraints related to the center of mass and the elastic axis, while maximizing the actuation power, as described in [50]. Rotor data and the first blade frequencies are shown in Table 2, where R is the radius, p is the pitch link location behind the blade axis, ϑ_p is the precone angle, c is the blade chord, ϑ_{tw} is the linear twist, Ω is the rotor angular velocity, α is the shaft angle, ν_β, ν_η and ν_θ are the nondimensional first flap, lag and pitch frequency of the blade respectively.

The aerodynamic model is based on the blade element theory combined with Drees' inflow model, along with the sectional aerodynamic characteristics of a NACA 23012 airfoil. This simple aerodynamic theory is accurate enough to reproduce the main charachteristics of the rotor in forward flight and can be used in a preliminary design.

Due to the simple aerodynamic theory, there is no interference among the blades and a single controller can be designed for one blade only. The control strategy is implemented as a stabilization problem, where a certain measure has to be mantained as small as possible by the controller. We are interested in minimizing the 4/rev harmonic of the blade root shear force F_z, so

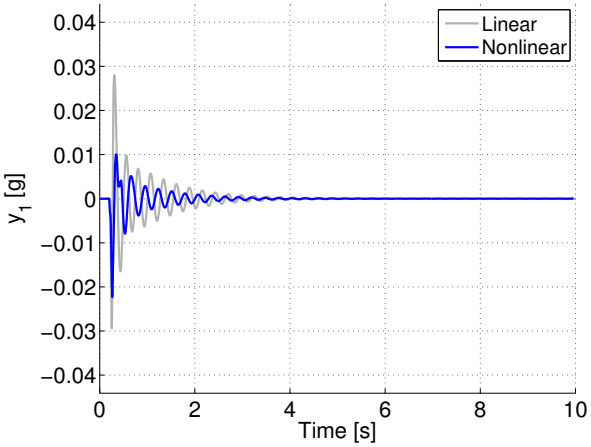


(a) Response at leading edge

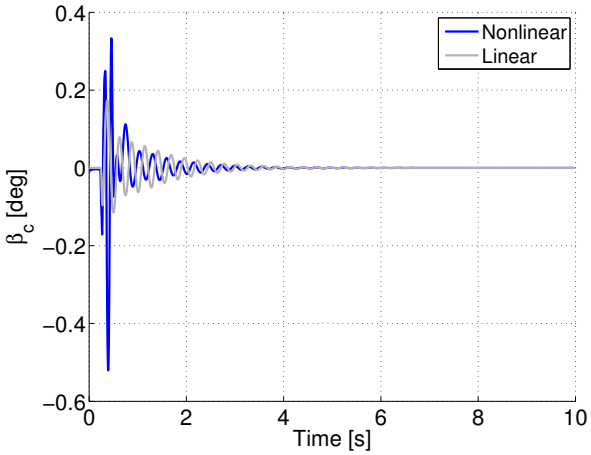


(b) Actuator output

Fig. 8: Control switch on/off



(a) Response at leading edge



(b) Control effort

Fig. 9: Comparison between linear and nonlinear model

Rotor data	Bo105 blade	Piezoelectric blade
R	4.9 m	-
p	0.23 m	-
ϑ_p	2.5°	-
c	0.3025 m	-
ϑ_{tw}	−8°	-
Ω	44.4 rad/s	-
α	3°	-
ν_β	1.11	1.1
ν_η	0.69	0.73
ν_θ	3.63	3.89

Table 2: Bo105 data with original and piezoelectric blade.

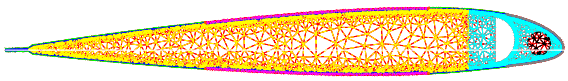


Fig. 10: Blade section discretization

to reduce the vibratory load of the shear force at the hub. To reduce the vibrations related to the two hub moments, we decide to minimize the 3/rev harmonic of the blade root shear force F_z as well, so acting indirectly on the blade root moment also. Therefore we decided to apply a pass-band filter to the blade force so to consider only the response related to the 3/rev and the 4/rev harmonics of interest.

The identifier approximates on-line the nonlinear map between the applied voltage on the blade V and the filtered blade root shear force F_z, while the controller computes the voltage on the blade minimizing its response. Therefore, $\mathbf{u}_n^{\mathrm{ID}} = \left\{F_{z,n}^{\mathrm{M}}, V_n, -1\right\}^{\mathrm{T}}$ and $y_n^{\mathrm{ID}} = F_{z,n}^{\mathrm{ID}}$ for the ID-RNN, while $\mathbf{u}_n^{\mathrm{CO}} = \left\{F_{z,n+1}^{\mathrm{ID}}, V_n, -1\right\}^{\mathrm{T}}$ and $y_n^{\mathrm{CO}} = V_{n+1}$ for the CO-RNN.
As mentioned before, we design only a single controller for the blade at azimuth $\Psi = 0°$. Taking into account the periodicity of the system, the same control signal is

delayed and applied to the other blades.
MBDyn can be easily coupled to external codes using bidirectional socket based communication protocols. In this example, MBDyn is linked to Simulink to perform closed loop simulations. The interface is managed through Matlab MEX-functions and the controller can be entirely developed within the Simulink enviroment.

Two trim configurations, at advancing parameters of $\mu = 0.23$ and $\mu = 0.33$, are analysed. The isolated rotor is trimmed so that its mean values of the rotor hub thrust and moments match the ones of table 3, with a shaft angle of α=3 deg. We choose to use 120 time steps per rotor revolution for both simulations. The

Rotor forces	$\mu = 0.23$	$\mu = 0.33$
T_z, N	20010	20200
M_x, Nm	740	1030
M_y, Nm	-85	-1400

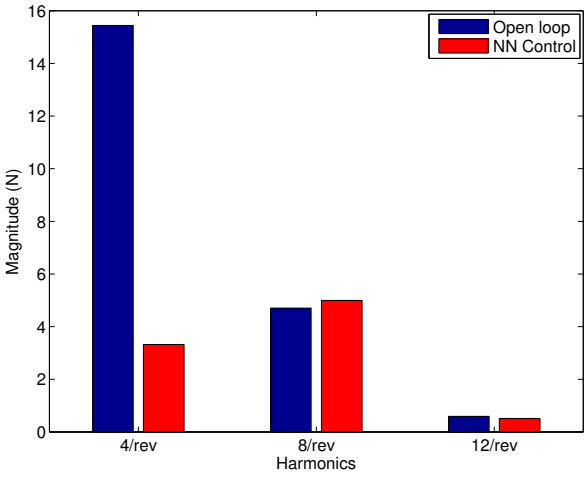
Table 3: Rotor trim data.

controller parameters, which are shown in Table 4, are tuned through simulations for the higher advancing velocity case, where vibrations are significantly stronger. To show the potentiality of the proposed adaptive controller, the same parameters are used for other flight conditions. Results for both cases are shown in Figures 11 and 13.

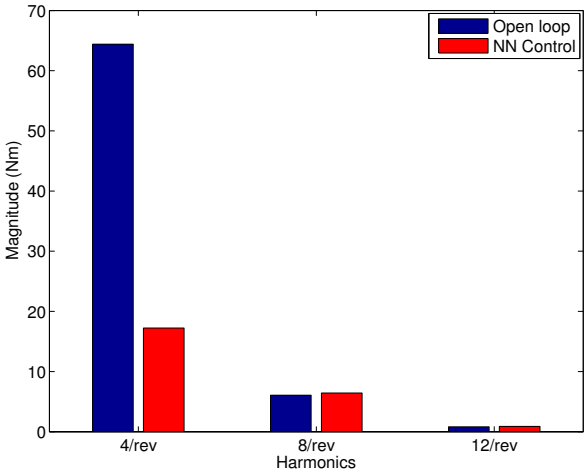
	ID-RNN	CO-RNN
n_x	4	3
η	$2 \cdot 10^{-1}$	$1 \cdot 10^{-1}$
ρ	-	$1 \cdot 10^{-4}$
Sampling frequency	800 Hz	

Table 4: Networks parameters for Bo105 case

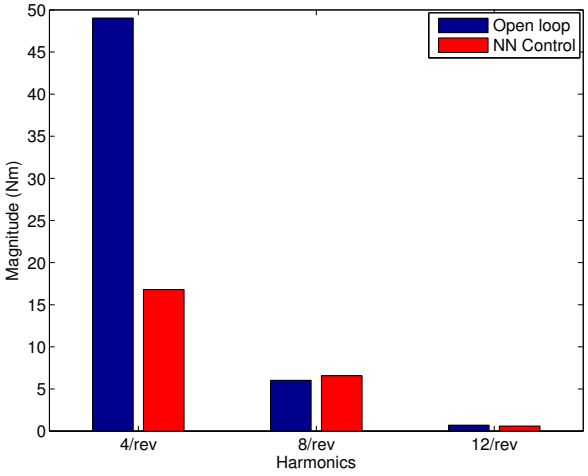
As can be seen in such figures, a great reduction of vibratory loads is achieved thanks to the controller activity and a reduction up to 90% is obtained in the flight condition where the controller parameters have been tuned. It is interesting to notice that even for the flight condition at $\mu = 0.23$ the results are good and the drop of vibratory loads is satisfactory. Figures 12 and 14 show the filtered response of the blade root shear force and the applied potential on the blades for both flight conditions. The maximum amplitude of the electric potential remains within reasonable levels and it doesn't exceed peaks of 300 V. This is important for a possible real implementation of such a solution.



(a) T_z hub

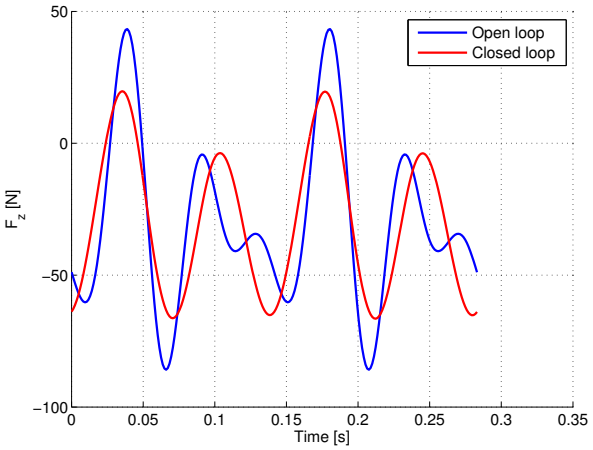


(b) M_x hub

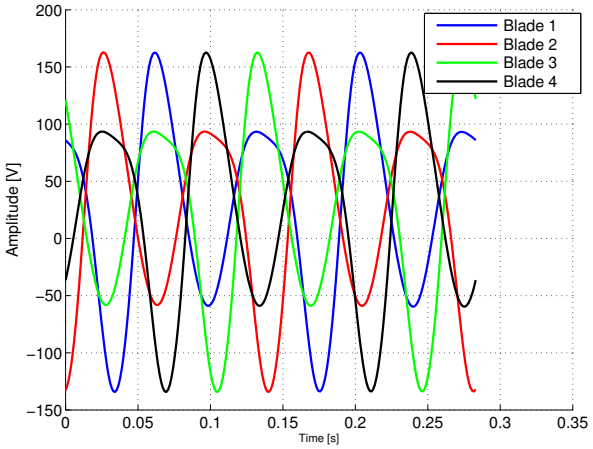


(c) M_y hub

Fig. 11: Vibrations suppression at μ=0.23. Harmonic comparison

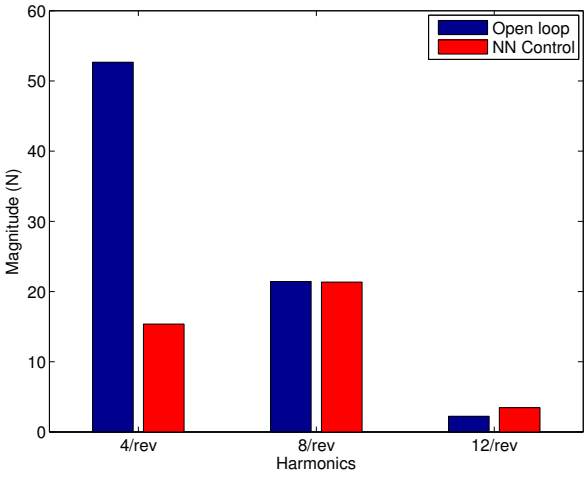


(a) Blade root shear force F_z (filtered signal).

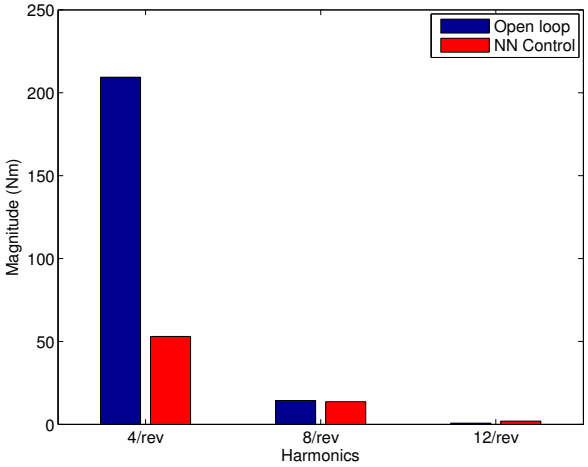


(b) Applied potential

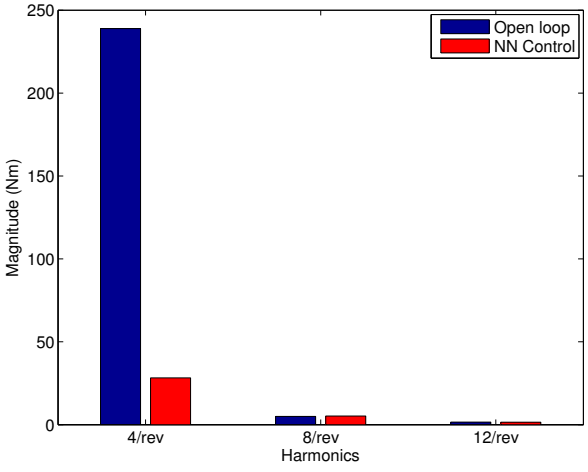
Fig. 12: Vibrations suppression at μ=0.23. Blade shear force reduction and control activity.



(a) T_z hub

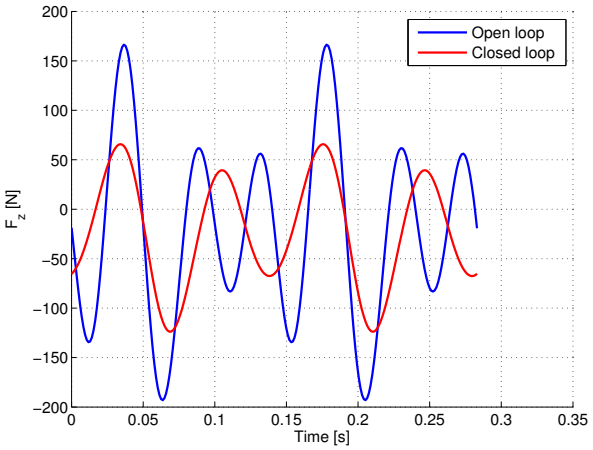


(b) M_x hub

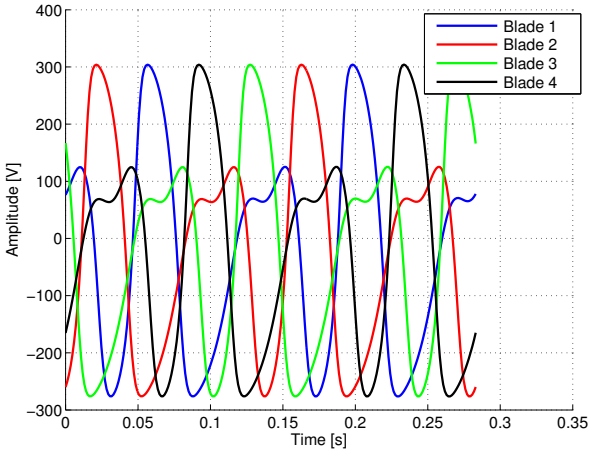


(c) M_y hub

Fig. 13: Vibrations suppression at μ=0.33



(a) Blade root shear force F_z (filtered signal).



(b) Applied potential

Fig. 14: Vibrations suppression at μ=0.33. Blade shear force reduction and control activity.

4 Concluding Remarks

In this work, a recurrent neural network based nonlinear adaptive controller has been investigated for complex aeroelastic problems. Thanks to its general formulation and to the black-box approach, it can be easily employed in a large class of problems. The two examples in this paper, being different in what concerns both the physics and the tools used for their modeling, show the RNN control capability to improve the performance of flying machines.

In the first example, nonlinear limit cycle oscillations of the BACT wing occurring above the flutter speed have been suppressed, while in the second one, the vibratory loads of the Bo105 rotor have been substantially reduced in two flight conditions retaining the same controller parameters.

Despite the potential of using a neural network approach, the choice of the controller parameters, such as the number of neurons and the learning rate, is not trivial and the search for a general rule remains an open problem. When considering a practical implementation of such a well performing controller, these issues should be carefully addressed.

It is believed that to fully verify the strength and weaknesses of a non linear adaptive RNN controller there remains the need of focusing on more complex and realistic applications, integrating true design specifications related to stability and response performances, while trying to make the tuning process as self contained as possible through the use of appropriate optimization techniques.

References

1. S. A. Jacklin. Closing the Certification Gaps in Adaptive Flight Control Software. In *AIAA Guidance, Navigation and Control Conference, Honolulu, AIAA Paper 2008-6988*, 2008.
2. B. Taylor, M. Darrah, and C. Moats. Verification and validation of neural networks: a sampling of research in progress. *Intelligent Computing: Theory and Applications*, 5103:8–16, 2003.
3. J. Schumann, P. Gupta, and S. Nelson. On Verification and Validation of Neural Network Based Controllers. In Berlin Springer, editor, *Proceedings of the Conference on Engineering applications in Neural Networks (EANN 03)*, 2003.
4. V. Mukhopadhyay. Historical Perspective on Analysis and Control of Aeroelastic Responses. *Journal of Guidance, Control and Dynamics*, 26(5):673–684, 2003.
5. E. Livne. Future of airplane aeroelasticity. *Journal of Aircraft*, 40(6):1066–1092, 2003.
6. J. Murua, P. Martínez, H. Climent, L. van Zyl, and R. Palacios. T-tail flutter: Potential-flow modelling, experimental validation and flight tests. *Progress in Aerospace Sciences*, 71:54–84, 11 2014.
7. S. N. Singh and M. Brenner. Limit cycle oscillation and orbital stability in aeroelastic systems with torsional nonlinearity. *Nonlinear Dynamics*, 31(4):435–450, 2003.
8. P. S.. Beran, T. W. Strganac, K. Kim, and C. Nichkawde. Studies of store-induced limit-cycle oscillations using a model with full system nonlinearities. *Nonlinear Dynamics*, 37(4):323–339, 2004.
9. E. H. Dowell. Some recent advances in nonlinear aeroelasticity: Fluid-structure interaction in the 21st century. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, 2010.
10. A. Abdelkefi, R. Vasconcellos, A. H. Nayfeh, and M. R. Hajj. An analytical and experimental investigation into limit-cycle oscillations of an aeroelastic system. *Nonlinear Dynamics*, 71(1-2):159–173, 2013.
11. H. Guo and Y. Chen. Supercritical and subcritical hopf bifurcation and limit cycle oscillations of an airfoil with cubic nonlinearity in supersonic\hypersonic flow. *Nonlinear Dynamics*, 67(4):2637–2649, 2012.
12. W. Zhang, C. Gao, Y. Liu, Z. Ye, and Y. Jiang. The interaction between flutter and buffet in transonic flow. *Nonlinear Dynamics*, Published online: 15 August 2015.
13. C. Gang, S. Jian, and L. Yueming. Active flutter suppression control law design method based on balanced proper orthogonal decomposition reduced order model. *Nonlinear Dynamics*, 70(1):1–12, 2012.
14. M. R. Waszak. Robust Multivariable Flutter Suppression for the Benchmark Active Control Technology (BACT) Wind-Tunnel Model. In *11th Symposium on Structural Dynamics and Control, Blacksburg, VA*, 1997.
15. C. Rubillo and P. Marzocca. Active Aeroelastic Control of Lifting Surfaces Via Jet Reaction Limiter Control. *International Journal of Bifurcation and Chaos*, 16:2559–2574, 2006.
16. K. A. Wise, E. Lavretsky, and N. Hovkimyan. Adaptive Control of Flight: Theory, Applications, and Open Problems. In *Proceedings of the 2006 American Control Conference*, 2006.
17. J. Ko, T. W.. Strganac, and A. J. Kurdila. Adaptive feedback linearization for the control of a typical wing section with structural nonlinearity. *Nonlinear Dynamics*, 18(3):289–301, 1999.
18. C.-L. Chen, C.-W. Chang, and H.-T. Yau. Terminal sliding mode control for aeroelastic systems. *Nonlinear Dynamics*, 70(3):2015–2026, 2012.
19. K. W. Lee and S. N. Singh. Immersion- and Invariance-Based Adaptive Control of a Nonlinear Aeroelastic System. *Journal of Guidance, Control and Dynamics*, 32(4):1100–1110, 2009.
20. R.C. Scott and L.E. Pado. Active Control of Wind Tunnel Model Aeroelastic Response Using Neural Networks. *Journal of Guidance, Control and Dynamics*, 23:1100–1108, 2001.
21. F. Bernelli-Zazzera, P. Mantegazza, G. Mazzoni, and M. Rendina. Active Flutter Suppression Using Recurrent Neural Networks. *Journal of Guidance, Control and Dynamics*, 23:1030–1037, 2000.
22. M. Mattaboni, G. Quaranta, and P. Mantegazza. Active flutter suppression for a three surface transport aircraft by recurrent neural networks. In *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, HI*, 2007.
23. Q. Zong, F. Wang, B. Tian, and R. Su. Robust adaptive dynamic surface control design for a flexible air-breathing hypersonic vehicle with input constraints and uncertainty. *Nonlinear Dynamics*, 78(1):289–315, 2014.

24. B. Xu. Robust adaptive neural control of flexible hypersonic flight vehicle with dead-zone input nonlinearity. *Nonlinear Dynamics*, 80(3):1509–1520, 2015.
25. M. Ghorashi. Nonlinear analysis of the dynamics of articulated composite rotor blades. *Nonlinear Dynamics*, 67:227–249, 2012.
26. D. Patt, L. Liu, J. Chandrasekar, D. S. Bernstein, and P. P. Friedmann. Higher-harmonic-control algorithm for helicopter vibration reduction revisited. *Journal of Guidance, Control and Dynamics*, 28(5), September-October 2005.
27. P. Arcara, S. Bittanti, and M. Lovera. Periodic control of helicopter rotors for attenuation of vibrations in forward flight. *IEEE Transaction on Control Systems Technology*, 8(6), 2000.
28. F. D. Ulker. *A New Framework For Helicopter Vibration Suppression; Time-Periodic System Identification and Controller Design*. PhD Thesis, Ottawa-Carleton Institute for Mechanical and Aerospace Engineering, April, 2011.
29. C. Brillante, M. Morandini, and P. Mantegazza. H2 periodic control on active twist rotor for vibration reduction. *AHS 70th Annual Forum and Technology Display, Montreal, Canada*, May 20-22, 2014.
30. B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation*. Wiley, 2003.
31. N. Bhoir and S. N. Singh. Output feedback modular adaptive control of a nonlinear prototypical wing section. *Nonlinear Dynamics*, 37(4):357–373, 2004.
32. B. Zhu. Nonlinear adaptive neural network control for a model-scaled unmanned helicopter. *Nonlinear Dynamics*, 78:1695–1708, 2014.
33. G Romanelli, E. Serioli, and P. Mantegazza. A "Free" Approach to Computational Aeroelasticity. In *48th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2010-0176, Orlando, FL*, 2010.
34. G. Romanelli, M. Castellani, P. Mantegazza, and S. Ricci. Coupled CSD/CFD non-linear aeroelastic trim of free-flying flexible aircraft. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, HI*, 2012.
35. P. Masarati, M. Morandini, and P. Mantegazza. An efficient formulation for general-purpose multibody/multiphysics analysis. *ASME J. Comput. Nonlinear Dyn.*, 2013.
36. S. Haykin. *Neural Networks and Learning Machines: A Comprehensive Foundation*. Prentice Hall, 2008.
37. R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1(2):270–280, June 1989.
38. M. R. Waszak. Modeling the Benchmark Active Control Technology Wind-Tunnel Model for Active Control Design Applications. Technical report, NASA, 1998.
39. V. Mukhopadhyay. Transonic Flutter Suppression Control Law Design and Wind-Tunnel Test Results. *Journal Of Guidance, Control and Dynamics*, 23:930–937, 2000.
40. W. Zhang, B. Wang, and Z. Ye. High efficient numerical method for limit cycle flutter analysis based on nonlinear aerodynamic reduced-order-model. In *51st AIAA, ASME, ASCE, AHS, ASC Structures, Structural Dynamics and Material Conference, Orlando, FL*, 2010.
41. A. Mannarino and P. Mantegazza. Nonlinear aeroelastic reduced order modeling by recurrent neural networks. *Journal of Fluids and Structures*, 48(0):103 – 121, 2014.
42. A. G. Kelkar and S. M. Joshi. Passivity-Based Robust Control with Application to Benchmark Active Controls Technology Wing. *Journal of Guidance, Control and Dynamics*, 23:938–947, 2000.
43. M.R. Waszak and J. Fung. Parameter estimation and analysis of actuators for the BACT wind-tunnel model. In *21th AIAA Atmospheric Flight Mechanics Conference, San Diego,CA*, 1996.
44. O. Dieterich, J. Götz, B. DangVu, H. Haverdings, P. Masarati, M. Pavel, M. Jump, and M. Genaretti. Adverse rotorcraft-pilot coupling: Recent research activities in europe. *34th European Rotorcraft Forum (ERF)*, September 2008.
45. M. L. Wilbur and W. K. Wilkie. Active-twist rotor control applications for uavs. *24th US Army Science Conference*, 2004.
46. H. P. Monner, J. Riemenschneider, S. Opitz, and M. Schulz. Development of active twist rotors at the German Aerospace Center (DLR). *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 2011, Denver, CO*, 2011.
47. G. L. Ghiringhelli, P. Masarati, and P. Mantegazza. Multibody implementation of finite volume C0 beams. *AIAA Journal*, 38(1), 2000.
48. G. L. Ghiringhelli, P. Masarati, and P. Mantegazza. Characterisation of anisotropic, non-homogeneous beam sections with embedded piezo-electric materials. *Journal of Intelligent Material Systems and Structures*, 8:842–858, 1997.
49. M. Morandini, M. Chierichetti, and P. Mantegazza. Characteristic behavior of prismatic anisotropic beam via generalized eigenvectors. *International Journal of Solids and Structures*, 47:1327–1337, 2010.
50. G. L. Ghiringhelli, P. Masarati, M. Morandini, and D. Muffo. Integrated aeroservoelastic analysis of induced strain rotor blades. *Mechanics of Advanced Materials and Structures*, 15:291–306, 2008.