IV International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2011
M. Papadrakakis, E. Oñate and B. Schrefler (Eds)

NEURAL NETWORKS AS SURROGATE MODELS FOR NONLINEAR, TRANSIENT AERODYNAMICS WITHIN AN AEROELASTIC COUPLING-SCHEME IN THE TIME DOMAIN

K. Lindhorst*, M. C. Haupt* and P. Horst*

*Institute of Aircraft Design and Lightweight Structures (IFL)
Technische Universität Braunschweig
Hermann-Blenk-Straße 35, D-38108 Braunschweig , Germany
e-mail: k.lindhorst@tu-bs.de, www.ifl.tu-braunschweig.de

Key words: Aeroelasticity, Reduced Order Modelling, Neural Networks, NLR7301

Abstract. In this paper the creation of a nonlinear, transient surrogate model is described that can be used within an aeroelastic coupling-scheme in the transonic range. The method is based on the theory of artifical neural networks as well as the autoregressive moving average method (ARMA). It can be shown that the method is able to approximate the nonlinear aeroelastic behaviour of the NLR7301 airfoil. Also limit cycle oscillations can be approximated with acceptable accuracy.

1 INTRODUCTION

Aeroelasticity is an important aspect in modern aircraft design and cannot be neglected in the optimization process of flight performance. The challenge herein is the complexity of an aeroelastic simulation, which is normally separated in two coupled subsystems, the aerodynamic and the structural subsystem. This coupled problem can be solved with high accuracy by combining a computational fluid dynamics solver (CFD) with a computational structure mechanics solver (CSM) within a CFD-CSM-coupling scheme. On the one hand with such a coupling scheme influences due to separation, transition or shocks can be considered, but on the other hand the computational effort is very high.
In general the solution of the aerodynamic subsystem needs much more computational effort than the structural subsystem, especially in simplified systems like a 2D airfoil or 3D wing sections. Furthermore in aeroelasticity only a fracture of the calculated aerodynamic values are needed, for example the global lift and global pitching moment of the airfoil or the pressure distribution on the coupling surface. However the whole aerodynamic system has to be solved completely each timestep.
Therefore the aim within this paper is the creation of a surrogate model for the nonlinear,

transient aerodynamic subsystem that provides only the neccessary data and allows much faster aeroelastic analysis with high accuracy. Such a surrogate model would also allow more complex aeroelastic calculations like manouvers as well as atmospheric disturbances like gusts within an acceptable time.
Reduced order modelling in aeroelasticity became an important field of research in the last years. For flutter boundary prediction methods based on Hopf bifurcation were successfully applied. According to Henshaw et al. [1] harmonic balance (HB), high order harmonic balance (HOHB), center manifold, normal form and numerical continuation methods can be used for bifurcation analysis. Another approach is the eigenvalue realization algorithm (ERA) [1, 6], which identifies a linear state-space-formulation of a given system. Furthermore proper orthogonal decomposition (POD) is widely used for model reduction [12]. Lucia et al. [6] also uses POD in combination with the second order Volterra series to approximate nonlinear behaviour.
Finally different types of neural networks have been used to predict the behaviour of aeroelastic systems. Voitcu for example uses classic multilayer perceptron networks (MLP-ANN) to predict a closed aeroelastic system with structural nonlinearities [9, 10]. In the contrary Won proposes a radial basis function network (RBF-ANN) to approximate the aerodynamic behaviour of the AGARD 445.6 wing [13].
The approach chosen in this paper is based on Won's RBF networks but is modified to allow an efficient multiple input multiple output mapping.

2 ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANN) are mathematical models of biological neural networks, which are naturally also known as brains. An ANN is a quite variable mapping method that can be used in many different fields like physics, biologics, economics and engineering, especially in control engineering. There exist many modifications of artificial neural networks but most of them are based on the multilayer perceptron network (MLP-ANN), which is explained shortly in the following. For more details the reader is referred to Hagan et al. [4].

2.1 Multilayer Perceptron Neural Network

A MLP-ANN consists of one or more layers, which also consist of one or more neurons. In general the architecture[1] is optional but it has strong influence on the networks prediction precision. So the number of layers and neurons that are neccessary for a proper system identification depends on the complexity and nonlinearity of the observed system. In figure 1 a feed forward MLP-ANN with two layers is shown.

Like in biological neural networks the neurons are the backbones of the ANN. A neuron

[1]The architecture is the outer topology meaning the number of layers as well as the inner topology meaning the number of neurons per layer

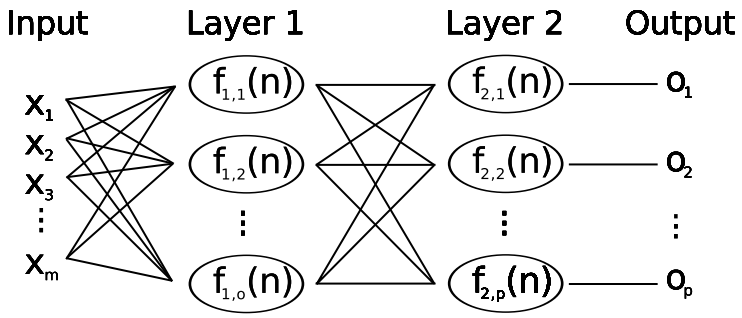


Figure 1: Two layer feed forward MLP-ANN

in a MLP-ANN consists of a bias b, its weighted connections to other neurons w_i and a transfer function $f(n)$ also called activation function. In general the neurons transfer function is user-defined, but often set to linear $f(n) = n$ or hyperbolic functions $f(n) = tanh(n)$. The answer a_n of a single neuron to j given inputs x is calculated with equation 1.

$$a_n = f_n(b_n + \sum_{k=0}^{j} w_{n,k} x_k) \tag{1}$$

An important property of neural networks is that the ability to learn. The learning process of a MLP-ANN is the optimization of the bias values and weight factors in order to match the target output. The network learning process is the difficult part in ANN usage, because the training set as well as the architecture must be well chosen.
An important method for the network training is the backpropagation algorithm, which allows the systematic training of multilayer networks. The backpropagation can be used with several optimization methods like steepest descent, Levenberg-Marquardt or Newtons method. The backpropagation as well as the optimization algorithms are described in Hagan et al. [4].
Well designed networks should be able to predict the behaviour of nonlinear multiple input multiple output (MIMO) systems, but finding the best architecture can be quite difficult.

2.2 Radial Basis Function Artificial Neural Network

Another type of neural network is mentioned by Won [13] called radial basis function networks (RBF-ANN). This network uses radial basis functions, for example gaussian functions as shown in equation 4, in order to approximate nonlinear systems. The network Won proposes is quite a simple summation of the gaussian functions (see eq. 3), which have the ability as universal function approximators.

$$f_n(x) = e^{-\frac{(x-c_n)^2}{\sigma_n^2}} \tag{2}$$

$$o(x) = \sum_{j=1}^{k} e^{-\frac{(x-c_j)^2}{\sigma_j^2}} \tag{3}$$

This kind of networks has a more simple architecture than the MLP-ANN but the mapping of different types of inputs to target outputs is inefficient because the RBF-neurons must cover the whole input range. Furthermore Won proposes a simple training algorithm using the pseudo inverse of the neuron outputs. This training method led to unsatisfying results during the investigations for this paper, which will be explained in section 3.1.
Therefore in this paper a hybrid of both network types is used. Neurons with gaussian RBF are used in a special layer called prelayer to capture nonlinear effects but the network itself is a small MLP-ANN. Furthermore the prelayer is divided into neuron clusters in order to increase the efficiency. Due to this clustering the network is called clustered artificial neural network (CANN).

3 CLUSTERED ARTIFICIAL NEURAL NETWORKS

The clustered artificial neural network is in fact a MLP-ANN with an extra layer placed between the input values and the multilayer perceptron network (cf. fig. 2). The prelayer consists of neuron clusters that are designed for one type of input and which are not involved in the training process. The neurons of each cluster have the same gaussian transfer functions as shown in equation 4. The neuron centers c_n are distributed equidistantly over the range of input values given by the training set. The width σ of the gaussian functions of each cluster is set to $\sigma_n = 2(c_n - c_{n-1})$, so it is defined by the distance between the neuron centers within the current cluster. The number of neurons N per cluster is optional but should be chosen depending on the range of input values as well as the training set.
A given input value is therefore transformed into N values which can be described as an input vector $\underline{C}(x)$. The entries of $\underline{C}(x)$ are values between 0 and 1 whereupon the value depends on the distance between neuron center to the input value: If the input x equals the center c the neurons output is 1, but the larger the distance between x and c is the smaller is the neurons output. This means that the position of the maximum value of the entries depends on the scalar value of x. This vector is then fed into the ANN and mapped to an output.

$$\underline{C}(x) = [f_1(x), f_2(x), ..., f_N(x)] = [e^{-\frac{(x-c_1)^2}{\sigma^2}}, e^{-\frac{(x-c_2)^2}{\sigma^2}}, ..., e^{-\frac{(x-c_N)^2}{\sigma^2}}] \tag{4}$$

3.1 Properties and limitations of CANN

The main advantage of the CANN is its simple architecture. The prelayer replaces the hidden layers of the classic MLP-ANN so that only the output layer remains. Thus it is

possible to use only one neuron with a linear transfer function for each output to cover nonlinearities within the cluster range. In fact for more complex problems the prelayer may also be combined with a real multilayer network but for the current problem one neuron per output was used. According to that the network training can be performed quite easily, because the network only consists of one layer with linear neurons which have to be optimized. In fact this leads to a linear optimization problem which can be solved with the pseudo-inverse as Won proposed [13]. Unfortunatly the input matrix, which has to be inverted is conditioned badly which leads to poor results of the pseudo-inverse training process. For this reason the steepest descent algorithm with conjugate gradient method was used in this paper which is described by Hagan [4]. The investigations showed that the simply structured and easily implemented CANN-method is able to predict nonlinear behaviour with acceptable precision. It is important to say, that a MLP-ANN with a well-designed architecture might be more precise than a CANN but the topology of this well-fitted architecture must be defined first.
A further limitation of the method is the ability of extrapolation. The cluster boundaries, which are set during the training process, define the covered input range. Input values laying out off these cluster boundaries lead to zero valued prelayer output vectors $\underline{v}(i) = \underline{0}$. Thus the input values outside the cluster boundaries are neglected by the prelayer. This aspect is not a real disadvantage, because a reliable extrapolation of nonlinear systems is in fact not possible.

3.2 Usage as Transient Predictor

Until now a nonlinear mapping method for MIMO systems was described. In order to apply the method to transient, timediscretized systems an approach similar to the autoregressive moving average (ARMA) is used, like it is proposed by Won [13]. The ARMA method predicts the response of a linear, timediscretized single input single output system at the time t with the sum of m past inputs x and n past outputs o like it is shown in equation 5. The coefficients a_j and b_k have to be determined so that the model error is minimized.

$$o(t) = \sum_{j=0}^{m} a_j x(t-j) + \sum_{k=1}^{n} b_k o(t-k) \tag{5}$$

This approach is a simple neural network with a single linear neuron without a bias. Hence this method is also applicable to more complex networks like MLP-ANNs as well as CANNs. In contrast to ARMA in this paper the gradients of the past n inputs are observed instead of the outputs, because the usage of past outputs has a destabilizing effect.
Furthermore m is set to 1 so only the input of the current timestep is used and the time history of motion is only respected by the gradients. The advantage herein is that the

CANN is able to learn easily the approximation of the static as well as the transient behaviour of the system, because in the static case the gradients are set to zero and only the remaining input is mapped to the target output. Another aspect is, that the gradients contain the information of the time step size implicitly.
With these modifications and the assumption that there exists one layer behind the prelayer with one neuron per target output the approximation of $\underline{o}(t)$ can be described with equation 6.

$$\underline{o}(t) = \underline{a}_0 \underline{C}_x(x(t)) + \sum_{k=0}^{n} \underline{b}_k \underline{C}_{\dot{x}}(\dot{x}(t-k)) \tag{6}$$

The degrees of freedom of the observed two dimensional, rigid aeroelastic model are the pitch angle α and the plunge excitation h. Regarding that only the velocity of plunge motion $\dot{h}$ and not the plunge excitation h itself influences the aerodynamics, h is neglected as input parameter. Furthermore the gradients are scaled with the freestream flow velocity u_∞ to cover the aerodynamic dependency of the freestream flow conditions. For the same reason the dimensionless lift and pitching moment coefficient are used as target outputs. So the input parameters are α, $\frac{\dot{\alpha}}{u_\infty}$, $\frac{\dot{h}}{u_\infty}$ and the target output parameters are the lift coefficient C_L and the pitching momentum coefficient C_M. A scheme of the used CANN is shown in figure 2.

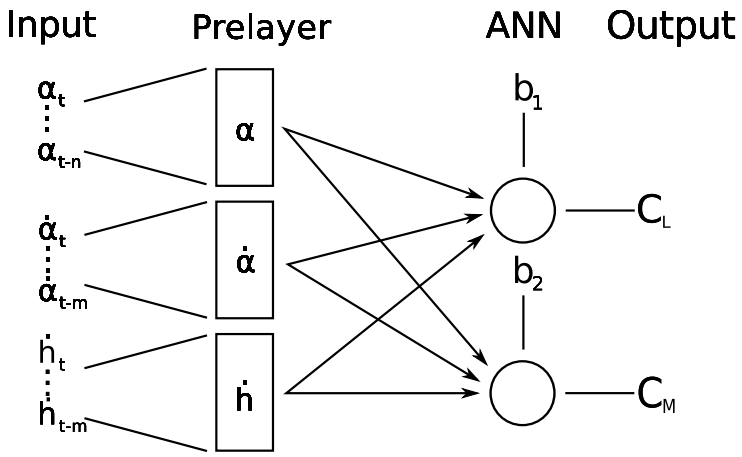


Figure 2: CANN-scheme used for transient aeroelastic systems

4 RESULTS

The method is demonstrated on the NLR7301 airfoil which is shown in figure 3. The CFD data are calculated with the *TAU*-code of the *German Aerospace Center (DLR)* [2, 3]. Weber [11] as well as Tang [7] showed that viscous effects have to be respected

for proper limit cycle calculation. Both of them suggest the Spalart-Allmaras turbulence model which is therefore also used in this paper. To neglect influences due to the transition location a fully turbulent flow is assumed.
Furthermore the number of neurons per cluster in the prelayer is set to $N = 30$ and the number of observed former gradients is set to $m = 75$, which is chosen due to parameter studies.

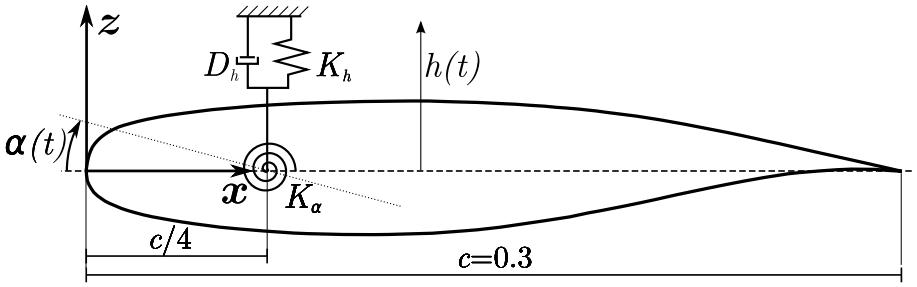


Figure 3: Geometry of the investigated NLR7301 airfoil

4.1 Network Training

In the following pitch and plunge motion of the CANN-CSD-coupling are compared with the results of the CFD-CSD-coupling at various Mach numbers (see Tab. 2). It is important to say that a network is identified at a fixed Mach number, so for each of the investigated Mach numbers at $Ma = 0.753$, $Ma = 0.7$ and $Ma = 0.65$ an according network was trained. On the other hand the freestream flow parameters pressure p_∞, velocity u_∞ and density ρ_∞ change with the temperature T_∞ at a fixed Mach as well as a fixed Reynolds number. The free stream parameters are connected with the Sutherland model, the ideal gas law, the Mach number definition and the Reynolds number definition. According to that the flow conditions can be manipulated even if the Mach and Reynolds number is fixed, which is why different free stream temperatures T_∞ are investigated with the same network at each Mach number.

In application the network must correctly react on different frequencies and amplitudes. This flexibility must be learned during the training which is why it is neccessary to provide as different training sets as possible to the network. The three networks in this paper were trained with a standardized bundle of forced motion training sets to show the practicability of the method. It is important that the training sets include different amplitudes and frequencies to ensure the required flexibility.
An example for a bundle of forced motion training sets is shown in figure 4. In addition to the shown transient sets a small set of 33 static calculations with $\alpha_{Static} = -8°...8°$ is also involved in the network training to ensure correct approximation of static cases.

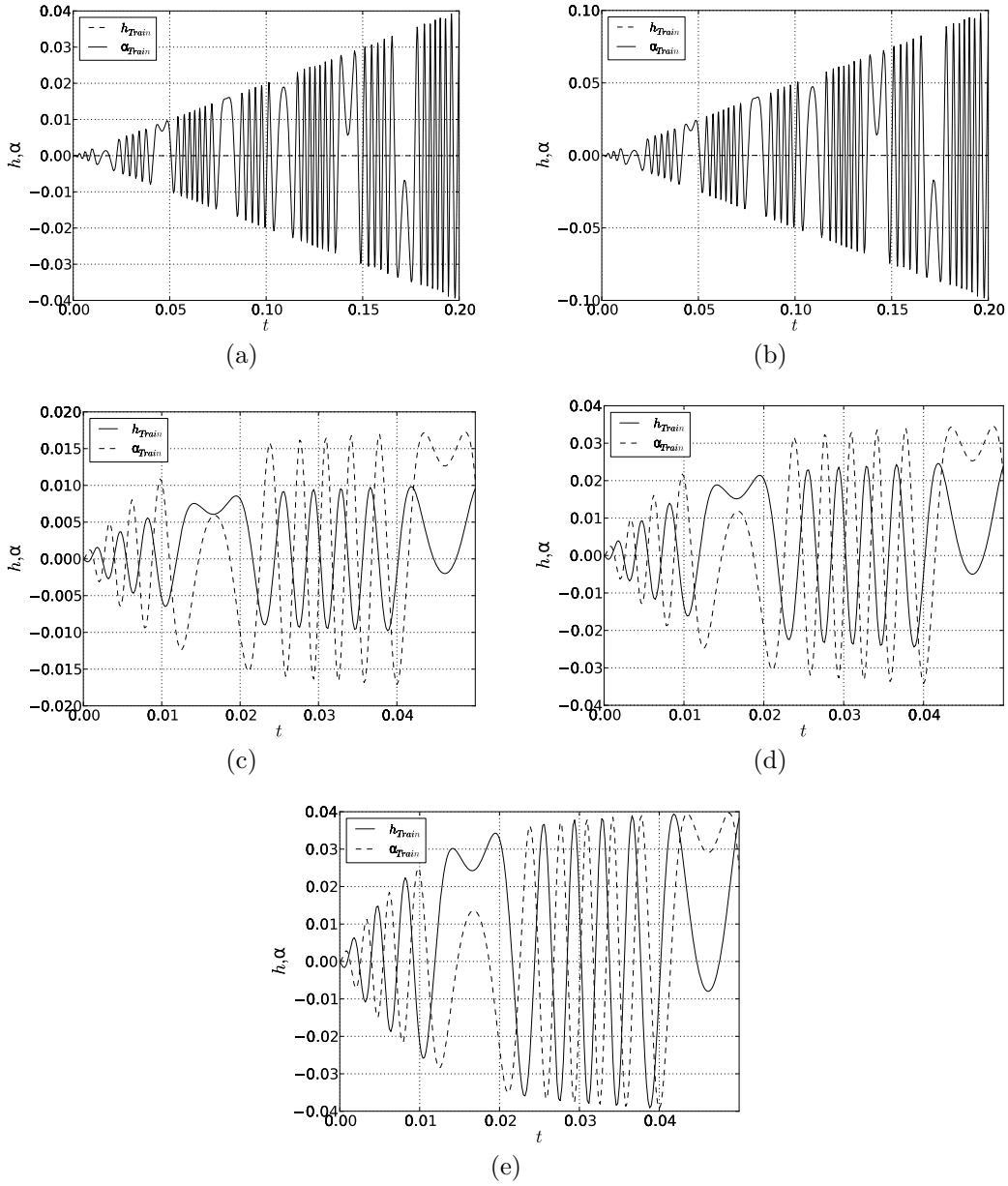


Figure 4: Example of a bundle of training set for one Mach number

4.2 Usage as Surrogate Model

After the network was trained it can be used in a two degree of freedom coupling scheme with a structure model. In this paper the structure model of Tang [7] is used for the coupled investigations of which the parameters are given in table 1. The governing equation of motion is given in 7.

Table 1: Parameters of the structure model by Tang [7]

m	$26.64kg$	$I_{c/4}$	$0.086m^2kg$
s_α	$0.378mkg$	K_h	$1.21 \cdot 10^6 N/m$
K_α	$6.68 \cdot 10^3 Nm/rad$	D_h	$82.9kg/s$
D_α	$0.197m^2kg/rad/s$		

The deformation and load transfer between the structure and the surrogate model is quite trivial: The output values of the structure model h and α are used as network input and the network output values C_L and C_M are given back to the structure model. The lift $L(t)$ and the pitching moment $M(t)$ can then be recalculated from C_L and C_M with the current flow conditions.
Furthermore an iterative staggered coupling with Aitken relaxation and first order predictor is used as it is described by Unger [8]. For the time integration of the structural system the Newmark method is used (see Hughes [5]).

$$\begin{bmatrix} m & s_\alpha \\ s_\alpha & I_{c/4} \end{bmatrix} \begin{pmatrix} \ddot{h} \\ \ddot{\alpha} \end{pmatrix} + \begin{bmatrix} D_h & 0 \\ 0 & D_\alpha \end{bmatrix} \begin{pmatrix} \dot{h} \\ \dot{\alpha} \end{pmatrix} + \begin{bmatrix} K_h & 0 \\ 0 & K_\alpha \end{bmatrix} \begin{pmatrix} h \\ \alpha - \alpha_0 \end{pmatrix} = \begin{pmatrix} L(t) \\ M(t) \end{pmatrix} \quad (7)$$

LCO case

First the LCO case at $Ma = 0.753$ is observed. Weber as well as Tang performed their LCO investigations without a structural damper and with an initial angle of attack of $\alpha_0 = 0.6°$. In order to match their results no damper is used in this case, so $D_h = D_\alpha = 0$ and the same initial angle of attack is applied. Here the first time steps are critical, because the surrogate model needs the gradients of former time steps (see equation 6), which are not available during this phase. Therefore the unknown gradients are set to zero.

In figure 5 the predicted motion of the CANN-CSM-model is compared with the calculated motion of the CFD-CSM-scheme. The amplitudes of the predicted motion increases faster than the reference solution between $t = 0.2$ and $t = 0.8$ (see figure 5 on the left hand side) but the final LCO amplitude is predicted quite well as it can be seen in figure 5 on the right hand side. The temporal offset between the predicted and the calculated motion can be explained with small differences in the frequencies. In figure 6 (a) and (b) the corresponding C_L and C_M coefficients are shown. In both diagrams the nonlinear effects can be recognized due to the nonelliptic shape of the hysteresis loop.

The computational effort of the CFD-CSM-scheme for the presented LCO-case is about 92 hours on two Intel i7 cores with 2.8 GHz. In contrast to that the surrogate model needs

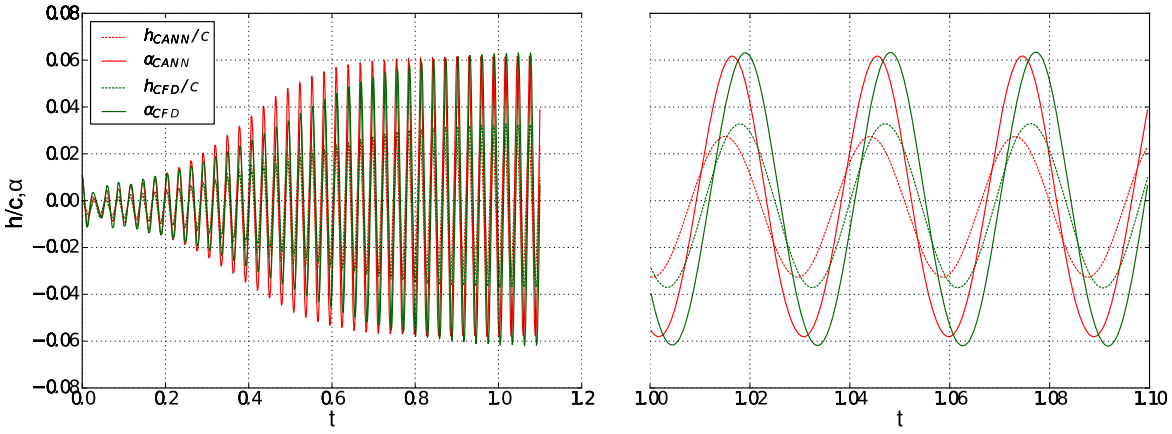
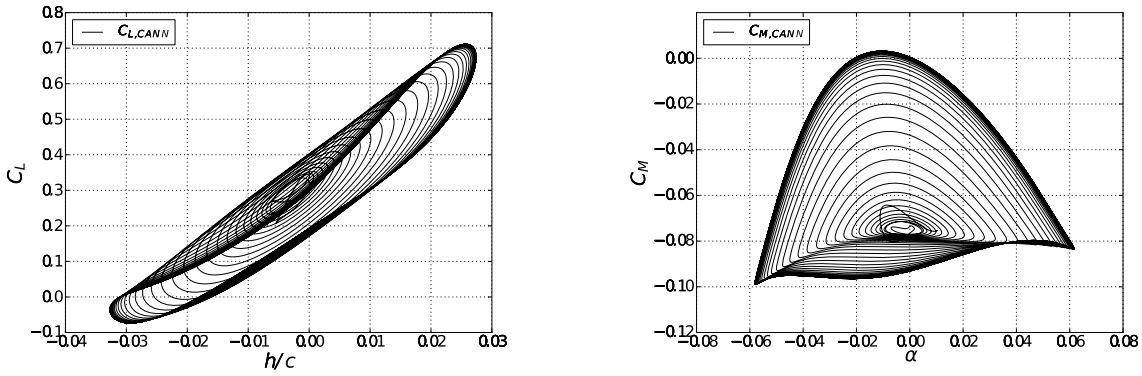


Figure 5: Comparison of CFD- and CANN-results of the LCO-case



(a) (b)

Figure 6: a) Approximated lift coefficient vs. h ; b) Approximated pitching moment coefficient

for the same calculation 6 minutes and 13 seconds on a single Intel i7 core with 2.8 GHz. So the surrogate model needs 0.1% of the time with half of the processor power to predict the LCO behaviour.
It is important to say, that the creation of the surrogate model takes also about 23 hours per Mach number, because training sets have to be calculated (ca. 21h) and the network has to be trained as well (ca. 2h).

Investigation of other Mach numbers

Furthermore two other Mach numbers with different free stream conditions are investigated, which can found in table 2. In all observed cases the Reynolds number and

Reynolds length are constant at $Re = 1.723 \cdot 10^6$ and $L_{Re} = 0.3m$. In these investigations the dampers D_h and D_α are set to the values of table 1 and the initial angle to $\alpha_0 = 0°$. Furthermore during the first 100 steps ($t_{100} = 0.02$) a forced motion is applied to the coupled model with $\alpha(t) = \frac{\pi}{180}\sin(200t)$ and $h(t) = 0$. The training sets for both networks were calculated at a free stream temperature of $T = 273.15K$.

Table 2: Aerodynamic parameters of the observed cases

Parameter	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Ma	0.7	0.7	0.7	0.65	0.65	0.65
$T_\infty[K]$	173.15	273.15	373.15	173.15	273.15	373.15
$u_\infty[m/s]$	184.635	231.902	271.047	171.447	215.337	251.687

In figures 7, 8 and 9 the predicted and the calculated airfoil motion is compared. In the beginning of each time series during the transient non-periodic phase the prediction matches the calculation with quite good accuracy. Even the dependency of the free stream conditions are covered in an acceptable manner. But it is also noticeable that after the transient starting phase the difference between the predicted and calculated amplitude increases with the time. This means that the stability limit of the surrogate model does not match the stability limit of the CFD-system. This lack of accuracy may be fixed with a better chosen training set.
Nevertheless it can be summarized, that the presented method is able to represent the aerodynamic system behaviour and can therefore substitute the CFD code within a CFD-CSM-coupling scheme for efficient aeroelastic analysis.

5 CONCLUSIONS

A method is presented, which is able to substitute the CFD-code within a CFD-CSM-coupling scheme with two degrees of freedom. Also nonlinear, transient effects can be covered by the method, which is demonstrated at the LCO-case as well as several examples at different transonic Mach numbers. Even the dependency of different free stream flow conditions at a fixed Mach number can be regarded, but the surrogate model also shows a lack of accuracy respective to the stability behaviour of the system.
Another important aspect is the computational effort needed for the creation of the surrogate model. It can be shown that the system behaviour can be identified with several not specially chosen trainings sets. For more complex aeroelastic investigations like manouvers or influences caused by gusts the choice of training sets should be optimized for a proper approximation of the stability behaviour.

6 ACKNOWLEDGMENTS

As part of the "ComFliTe" project the partial funding of this research under contract 20A0801D of the Federal Ministry of Economics and Technology through the German Aerospace Center is greatfully acknowledged. The authors thank J. Brink-Spalink of Airbus, F. Selimefendigil and S. Goertz of DLR for valueable discussions and their support.

REFERENCES

[1] M. de C. Henshaw, K. Badcock, G. Vio, C. Allen, J. Chamberlain, I. Kaynes, G. Dimitriadis, J. Cooper, M. Woodgate, A. Rampurawala, D. Jones, C. Fenwick, A. Gaitonde, N. Taylor, D. Amor, T. Eccles, and C. Denley. Non-linear aeroelastic prediction for aircraft applications. *Progress in Aerospace Sciences*, 43(4-6):65 – 137, 2007.

[2] M. Galle, T. Gerhold, and J. Evans. Parallel computation of turbulent flows around complex geometries on hybrid grids with the DLR-TAU code. In *Ecer, A., Emerson, D. R. (eds.), Proc. 11th Parallel CFD Conf., May 23-26 1999, Williamsburg, VA, North Holland*, 1999.

[3] H. V. Gerhold, T. and D. Schwamborn. On the validation of the DLR-TAU code. In *New Results in Numerical and Experimental Fluid Mechanics, Notes on Numerical Fluid Mechanics*. Vieweg, 1999.

[4] M. T. Hagan, H. B. Demuth, and M. Beale. *Neural Network Design*. PWS Publishing, 1996.

[5] T. J. Hughes and J. McCulley. *The Finite Element Method*. Prentice-Hall International Editions, 1987.

[6] D. J. Lucia, P. S. Beran, and W. Silva. Aeroelastic system development using proper orthogonal decomposition and volterra theory. In *44th AIAA Structures, Structural Dynamics and Materials Conference*, 7-10 April 2003.

[7] L. Tang, R. E. Bartels, P.-C. Chen, and D. D. Liu. Numerical investigation of transonic limit cycle oscillations of a two-dimensional supercritical wing. *Journal of Fluids and Structures*, 17:29–41, 2003.

[8] R. Unger, M. Haupt, and P. Horst. Coupling techniques for computational non-linear transient aeroelasticity. *Journal of Aerospace Engineering*, 222:435–447, 2008.

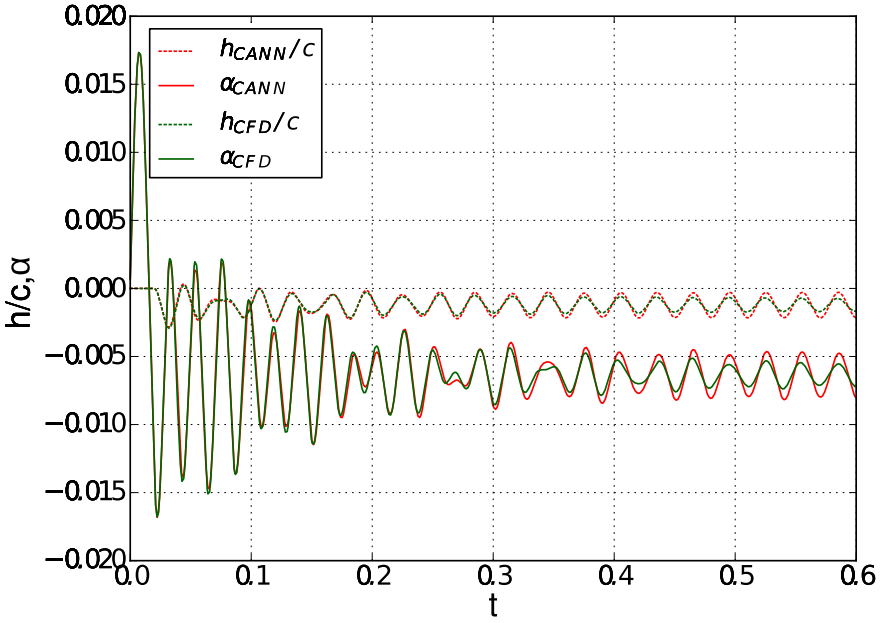
[9] O. Voitcu and Y. S. Wong. A neural network approach for nonlinear aeroelastic analysis. In *43th AIAA Structures, Structural Dynamics and Materials Conference, 22-25 April, 2002, Denver, Colorado*, 2002.

[10] O. Voitcu and Y. S. Wong. An improved neural network model for nonlinear aeroelastic analysis. In *44th AIAA Structures, Structural Dynamics and Materials Conference, 7-10 April, 2003, Norfolk, Virginia*, 2003.

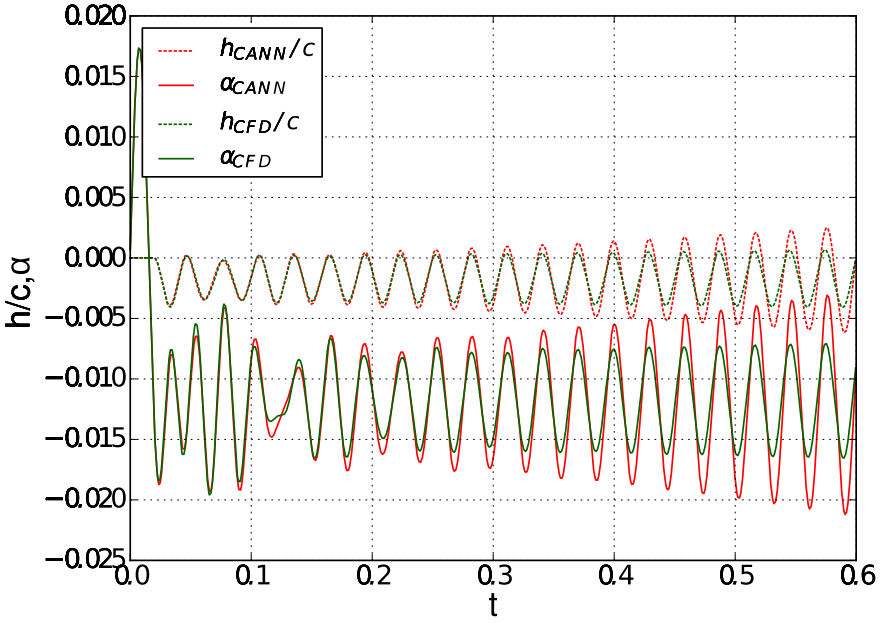
[11] S. Weber, K. D. Jones, J. A. Ekaterinaris, and M. F. Platzer. Transonic flutter computations for the nlr 7301 supercritical airfoil. *Aerospace Science and Technology*, 5:293–304, 2001.

[12] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, pages 2323–2330, 2002.

[13] K. Won, H. Tsai, M. Sadeghi, and F. Liu. Non-linear impulse methods for aeroelastic simulations. In *AIAA-2005-4845, presented at the 23rd AIAA Applied Aerodynamics Conference, Toronto, Ontario, June 6-9, 2005*, 2005.

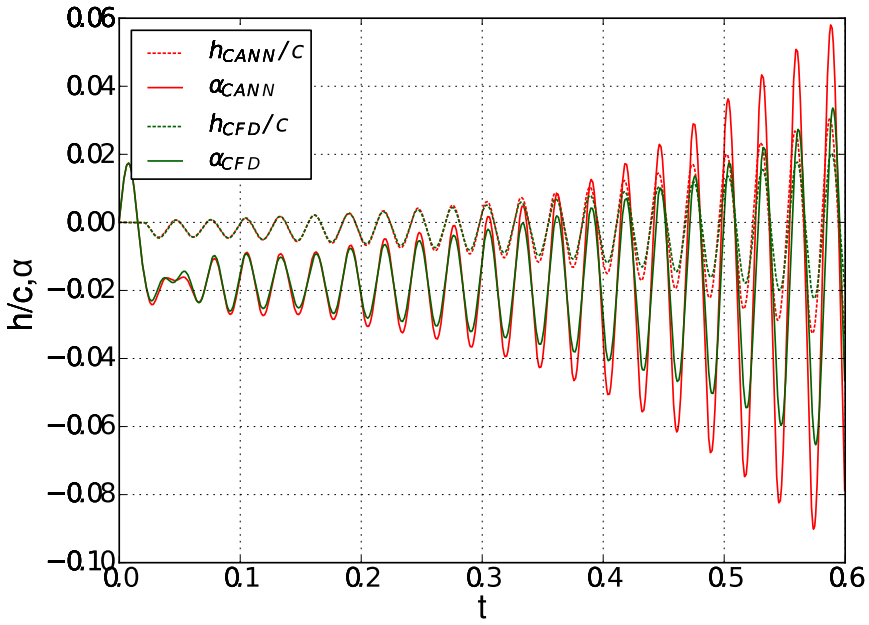


(a)

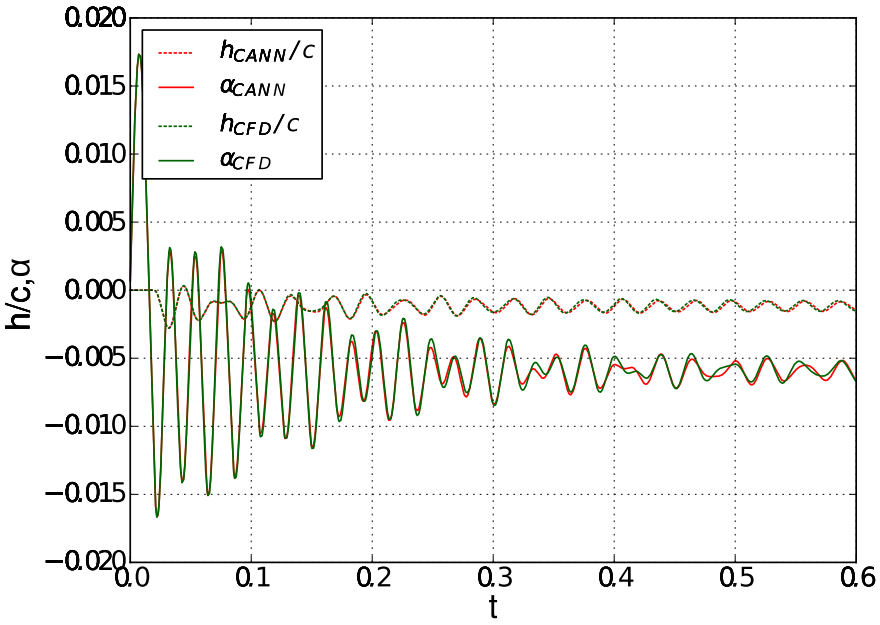


(b)

Figure 7: CANN vs. CFD Cases 1 & 2

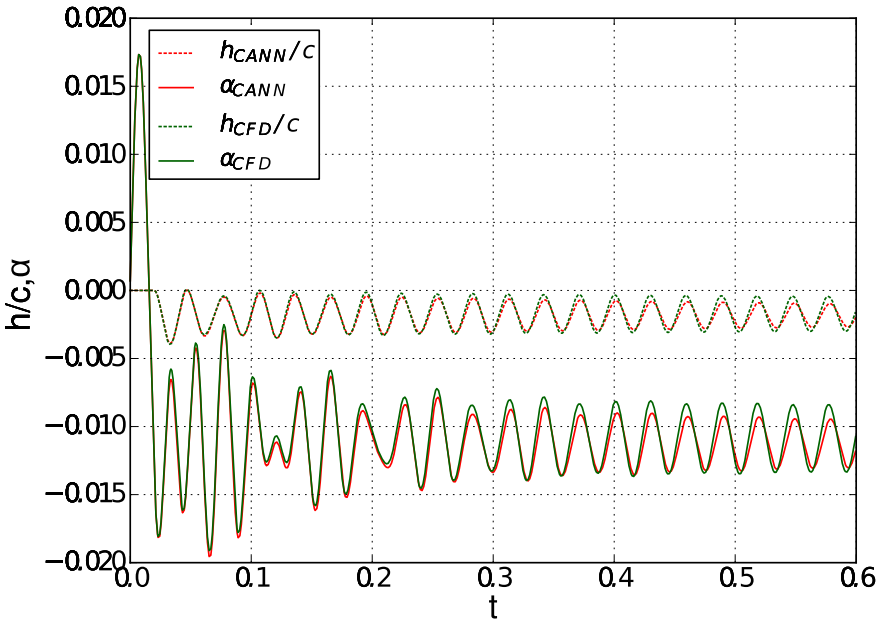


(a)

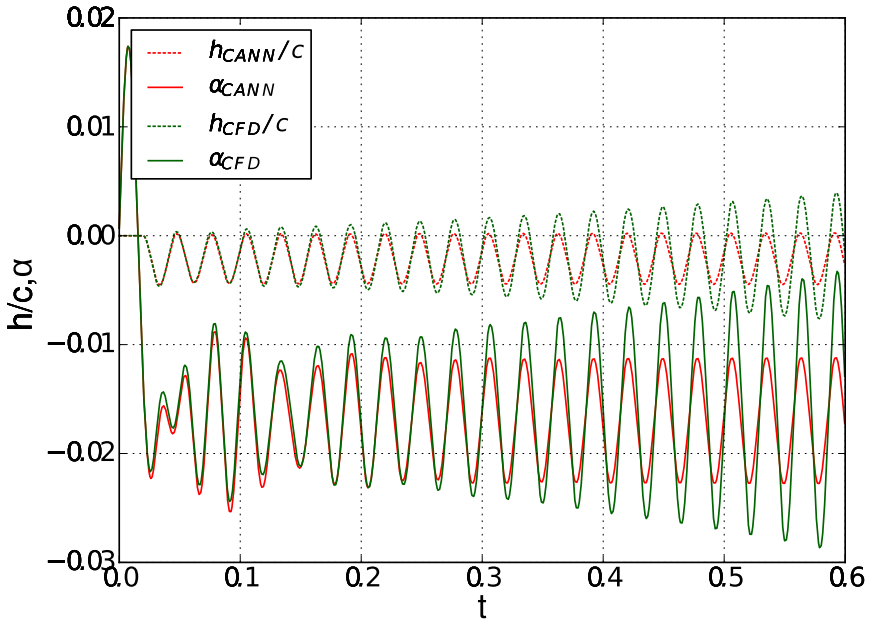


(b)

Figure 8: CANN vs. CFD Cases 3 & 4



(a)



(b)

Figure 9: CANN vs. CFD Cases 5 & 6