NUMERICAL INVESTIGATION OF 3D DYNAMIC STALL IN ROTATION USING CFD AND NEURAL NETWORKS

A. Spentzos, G. Barakos and K. Badcock
CFD Laboratory, Department of Aerospace Engineering, University of Glasgow, Glasgow G12 8QQ, U.K.
www.aero.gla.ac.uk/Research/CFD

Abstract

The objective of this study is to simulate and model 3D Dynamic Stall. Computational fluid dynamics is the main tool and after careful validation, results for the flow field and the surface pressure distribution were used to obtain a qualitative understanding of the complex flow during dynamic stall. The variety of flow configurations and the interactions between the tip and dynamic stall vortices result in a complex surface loading which was modelled using neural networks. In this work the neural networks were trained using experimental and CFD-generated Cp distributions over the suction side of a ramping wing for a number of reduced pitch rates. Subsequently, the neural network was switched to a predictive mode and produced the temporal and spatial Cp distibutions, for conditions within and outside its training envelope. The obtained results were very encouraging. The training stage was found to be an order of magnitude cheaper than each of the corresponding CFD computations. The predictive stage was found to be even faster thus establishing neural networks as suitable for modelling complex unsteady aerodynamic phenomena.

Nomenclature

Latin

c	Chord length of the airfoil
C_P	Pressure coeff., $C_P = \frac{1}{2\rho U_\infty^2}(P - P_0)$
k	Reduced frequency of oscillation, $k = \frac{\omega c}{U_\infty}$
M	Mach number
Re	Reynolds number, $Re = \rho U_\infty c/\mu$
t	Time
x	Chord-wise coordinate axis (CFD)
y	Normal coordinate axis (CFD)
z	Span-wise coordinate axis (CFD)
U	Local Axial Velocity
U_∞	Free-stream axial velocity

Greek

α	Oscillatory incidence
α_0	Mean incidence for oscillatory cases
α_1	Amplitude of oscillation
α^+	Reduced ramp rate, $\alpha^+ = \frac{d\alpha}{dt}\frac{c}{2U_\infty}$
β	Yaw angle
ρ	Density
ρ_∞	Density at free-stream

Acronyms

AR	Aspect Ratio
CFD	Computational Fluid Dynamics
DS	Dynamic Stall
DSV	Dynamic Stall Vortex
FFBP	Feed-Forward Back-Propagation
NN	Neural Network
RNN	Recursive Neural Network
TEV	Trailing Edge Vortex
TV	Tip Vortex

1 Introduction

Dynamic Stall (DS) is important to rotorcraft, wind turbines as well as highly maneuverable aircraft. The complexity of this highly non-linear, unsteady, three-dimensional (3-D) flow yields aerodynamic loads that are hard to characterise and predict. Experimental and computational work has established that a predominant feature of DS is the presence of large vortical structures on the suction side of the lifting surface, which distort the pressure distribution and produce transient forces that are fundamentally different from their static counterparts [1]. A review up to 1996 of all CFD efforts related to DS has been provided by Ekaterinaris and Platzer [2, 3]. Several other papers have appeared in the literature (see reference [4] amongst others). In addition, the recent paper by Barakos and Drikakis [5] provides an update on CFD prediciton of DS. In contrast to the numerical studies of two-dimensional (2-D) DS, 3-D works are rare. Newsome [6] focused on the laminar flow regime and attempted to simulate the experiments of Schreck and Helin [7]. Newsome's work was the first 3-D investigation to predict the 3-D Dynamic Stall Vortex (DSV). The work by Morgan and Visbal [8] considered the oscillatory motion

of a square wing at laminar flow conditions with end plates at both tips. The objective was to approximate the conditions inside a wind tunnel with the model spanning the test section, and focussed on the development of vorticity near the wing surface. The work of Ekaterinaris [2] dealt to a great extend with 2-D configurations and the 3-D problem was also attempted. The work by Spentzos *et al.*[9], is the most recent work on 3-D DS and to our knowledge, the only CFD investigation, to date, presenting detailed comparisons with experiments.

Overall, CFD studies of 3-D DS produce results of varied levels of accuracy depending on the employed grids, solvers, turbulence and transition models and computations can be expensive in terms of computing time. A variety of alternative approaches have also been undertaken, aiming to produce reduced models that are efficient and capable to characterise the flowfield of DS. Efforts along these lines include the ONERA model [10], the Leishman-Bedoes model [11] as well as a models based on Lagrangian functions devised by Veppa [12].

Past experience has shown [13, 14] that Neural Networks (NN) can recreate histories of unsteady aerodynamic loads on the suction side of pitching aerofoils, following appropriate training. Faller *et al.*[13, 14], used sets of experimental data to train a Recursive NN (RNN) into predicting the Cp readings along three spanwise positions on the upper surface of their wing. He concluded that RNNs are suitable for time-dependent problems. Faller *et al.*[13, 14], trained their network using experimental data for a total of 45 Cp transducers distributed along three spanwise positions on the upper surface of a ramping wing. They had measured data for a total of 8 different pitch rates, 5 of which were used for training and the rest for validation. Their NN included 47 input layer neurons, 45 neurons on the output layer, and two hidden layers with 32 neurons each. Apart from the 45 Cp predictions that where fed back to the input layer, the angle of attack α as well as the angular velocity $d\alpha/dt$ where used to teach the RNN.

In view of the above, the objectives of this work are: a) To explore the qualitative characteristics of DS in realistic rotor motions and b) to assess the suitability of NN models in reproducing the evolutionary characteristics of 3-D DS.

2 Numerical Method

2.1 CFD solver

The CFD solver used for this study is the PMB code developed at the University of Glasgow [15]. The code is capable of solving flow conditions from inviscid to laminar to fully turbulent using the Reynolds Averaged Navier-Stokes (RANS) equations in three dimensions. The use of the RANS form of the equations allows for fully turbulent flow conditions to be calculated with appropriate modelling of turbulence. The turbulence model used for this study has been the standard $k-\omega$ turbulence model [16], however, many other turbulence models are available. To solve the RANS equations, a multi-block grid is generated around the required geometry, and the equations are discritised using the cell-centered finite volume approach. Convective fluxes are discretised using Osher's upwind scheme because of its robustness, accuracy and stability properties. Third order accuracy can be achieved using a MUSCL interpolation technique and viscous fluxes are discretised using central differences. Boundary conditions are set using sets of halo cells. The solution is marched implicitly in time using a second-order scheme and the final system of algebraic equations is solved using a Krylov sub-space method.

2.2 Neural Network Model

Two different types of NNs were developed and assessed, both based on a standard feed-forward back-propagation (FFBP) model: (a) a standard FFBP NN and (b) a FFBP NN with feedback (Recursive NN). The above are based on the assumption that the C_p distribution on the suction side of a moving wing, can be approximated by the continuous mapping: $C_p = C_p(t, \alpha, \alpha^+, x, y)$. The theoretical justification for the suitability of a NN to approximate such a mapping, comes from the existence theorem put forward by Kolmogorov [17]. Various NN topologies have been investigated having 1 to 4 layers of neurons and 10 to 100 neurons per layer. The choice of the number of hidden layers as well as the number of neurons on each hidden layer is not straight forward and depends on the complexity of the problem at hand and the number of patterns used for training. A pattern is a combination of a set of input parameters together with their corresponding target value. It seems that the effect of increasing the number of layers is to make a NN 'smarter' and the effect of increasing the number of neurons per layer, makes a NN more accurate. Figure 1(a) shows the topology of the FFBP NN model. Each neuron is connected to all the neurons of the previous layer and to all the neurons of the next layer. All neurons are exactly the same, the only distinguishing characteristics between them being the number of inputs and outputs they have. Each neuron can be seen as a 'black box' which internally performs two basic operations: a) a combination which computes a linear summation over its input signals and b) a scaling through an activation function which bounds the result of the summation to

a required interval. In the present case, the activation function was chosen to be a sigmoid and the signal at the output of the i^{th} neuron on layer l is given by:

$$output_l^i = \frac{1.0}{1.0 + exp(-\sum_{j=1}^{n_{l-1}} output_{l-1}^j)} \qquad (1)$$

The range of values is the interval $0-1$ and n_{l-1} is the number of neurons on layer $l-1$. In that manner, the NN can be seen as a 'tree' which 'propagates forward' the input signals to the output. Each one of the branches of this tree as seen in figure 1(a), is essentialy a 'synaptic weight' that defines the connection magnitude of each neuron at a given layer to each neuron at the layer ahead. Thus, if all synaptic weights are known, a given combination of input values will yield a single value at the output of the NN. The process described so far is essentially the NN's predictive mode, were all synaptic weights are assumed to be known, hence the resulting Cp is a function of the input parameters. Based on this description, it is evident that the predictive mode is very cheap, as the process from input to output involves a small number of operations.

During training, a set of input parameters for which the resulting Cp values are known, are fed into the NN. An initial random distribution of synaptic weights is assumed, based on which, the predicted Cp values are calculated. Then, the predicted Cp values are compared against the known 'true' Cp values and an error based on this descrepancy is produced. The next stage is to iteratively adjust the synaptic weights in order to minimise the network's error in a back-propagation fashion. The method used here is by 'gradient descent' on the error function, during which all synaptic weights are re-approximated, based on how much the error changes by a small change on each weight. This is done by calculating the partial derivatives of the total error with respect to the individual weights and then shifting the weights by a small amount in the direction that reduces the error. This process is repeated for all patterns many times, until the resulting total error becomes less that 1% of the total initial error based on the initial random weights. The total error is defined as the summation of the absolute value of the errors over each pattern. In each iteration (epoch), all patterns are swept through at random order as this accelerates the learning process and prevents the network from being biased towards the last patterns presented and a particular sequence of pattern presentation.

2.2.1 NN with Feedback

The second NN model used is a variation of the model described in the previous paragraph. A feedback loop feeds the output vector at point t in time into the input layer in view of predicting the output vector at time $t + dt$. Thus time is no longer fed explicitly to the NN via an input node, but it is 'implied' during training since the training patterns are presented in their true time sequence. Another significant difference, is that the values of C_p used for training are presented in a flat manner, *i.e.* output neuron 1 corresponds to the C_p transducer at position x_1, y_1 etc, therefore the information of the spatial relations between the transducers is lost and the network can no longer interpolate or extrapolate in physical space. The RNN model used here can be seen in Figure 1(b)were the values from 15 different transducers where used for training, together with their corresponding α and α^+. The different training datasets were presented in a random fashion as explained in the previous paragraph. The main differences between the straight and recursive implementations, are (a): the target file is now built from the C_p histories used for training such as that $target[{C_p}^i(t)] = {C_p}^i(t + dt)$ and (b) the layer of input values at step p is set to be the layer of output values at step $p-1$ which accounts for the feedback requirement.

The advantage of this method lies in applications where it is convenient to exclude time from the training (and predictive) procedures. The disadvantage is that the spatial association is lost.

3 CFD Computations

In this work, the three-dimensional DS was computed in pitch-yaw and pitch rotation configurations. The wings used where based on the NACA 0012 airfoil with rounded tips and aspect ratios of 10 and 5.

The formation and evolution of the DSV under zero yaw and for a variety of flow conditions has been investigated previously in [9]. Figures 2(a-f) show the DSV cores for three different reduced rates and two different yaw angles, corresponding to conditions 1,3,4,6,8 and 10 of Table 2. The left column of Figure 2 contains the un-yawed conditions, whereas, in the right column, there is an angle of 15^o between the chord and the freestream. Examination of Figures 2 also shows that the main part of the DSV remains parallel to the wing span irrespectively of the magnitude of the yaw angle. As the DSV is primarily energised by the motion of the wing, it tends to follow the particular geometry of the suction side. When the yaw angle is non zero, the parallel to the wing component of the freestream

blows the DSV to the far side of the wing. As a consequence, the DSV-wing junction area approaches the right hand side tip of the wing (Figures 2(b,d and f)) while the distance of the DSV-wing junction at the left hand side of the wing increases.

It was shown in the previous paragraph, that the effect of yaw (for moderate yaw angles and pitching frequencies) in DS is the deformation of the spanwise formation of the DSV; the portion of the DSV close to the near side of the wing departs from the tip, while the portion of the DSV close to the far side of the wing approaches the tip. However, for rotor blades, the yaw angle is a function of the azimuth, therefore the quantity $d\beta/dt$ is non-zero. Figure 3(a) shows the DSV and Tip Vorted (TV) cores for the case 15 of table 2. It is evident that the Ω-shaped part of the DSV is now reduced to a part of the wing closer to the outboard tip. The shape the DSV attains soon after its formation is due to the presence of the tips. The TVs introduce suction in their proximity due to the direction of the induced velocity around the tip of the wing. This means that the DSV is supressed in the near the tip region while freely energised over the rest of the wing's span. This assymetric growth is forcing the DSV to assume a distorted Ω shape. Figure 3(b) shows streamlines on a spanwise plane located at 0.5c superimposed on Figure 3(a). One can observe that despite the inboard tip (right tip in figure) experiencing a lower inflow velocity (due to rotation) than the outboard tip, the TV induced velocity field stretches further away from the tip than for the case of the outboard tip, supressing the growth of the DSV until almost the half-span of the wing. Consequently, the part of the DSV which remains unaffected from the influence of the TVs is located closer to the outboard tip. Another interesting observation is that the DSV is now closer to the TVs [9]. This is the only case in 3D DS where the authors have encountered this vortical topology in the case of a high Re flow.

4 Neural Network Validation

The first step in the present analysis, was to use experimental data provided by Coton & Galbraith [18], in order to check the suitability of the approach described in the previous paragraphs. The measurements of a total of 8 ramping cases were used, the details of which are summarised in table 1. Out of these eight data sets, five were used for training and the remaining three were used for testing the predictions of the NN. Two of these cases were outside the training regime (cases 1 and 8 of table 1) and the third within (case 5 of table 1). This way both the interpolating and extrapolating qualities of the NN model are assessed.

Each ramping case was discretised in time into 150 unsteady steps and for each unsteady step readings from 90 transducers covering the suction side of the wing were used. Therefore the total number of training patterns was equal to $5 \times 150 \times 90 = 67,500$ and around 10^5 epochs were necessary to achieve convergence. The total computational time during training was just under 48 hours on a single processor 2.5 GHz linux box, based on a convergence criterion of the total error being 1% of the total initial error. The variables used to train the NN where $time$, α, α^+ and the x and y coordinates of the C_p transducers on the wing's suction side. The NN was thereafter trained to approximate the functional: $C_p = C_p(t, \alpha, \alpha^+, x, y)$ After training, the NN was switched into the predictive mode and simulated the evolutionary characteristics of a further three ramping cases (cases 1,5 and 8 of Table 1) which were not included in the training dataset. Comparisons between the experimental (left) and predicted (right) surface C_p distributions can be seen at Figure 4 at angles of attack above stall. The comparison is good as both the overall loading and the location of the DSV's footprint have been accurately predicted. The only exception is figure 4(a) were the experiment shows the DSV appearing closer to the wing tip as opposed to the NN prediction which shows the trajectory of the DSV to be closer to the root.

This initial experiment, has shown that the generic NN used, where time has been included in the input variable list rather than being implicitly assumed via the feedback loop of the recursive topology as used in [13, 14], has delivered very promising results. The reduced ramp rate has been the only differentiating parameter between the eight cases listed in table 1. The validation exercise of case 5 (Table 1) which represents a prediction in interpolating conditions regarding the reduced ramp rate clearly provides the best comparison between the experiment and the NN computation. On the left hand side of Figure 4, one can see the Cp contours extracted by the experiments, while shown on the right hand side are the NN predictions for the cases of Table 1. The contour topologies as well as the extrema of the C_p values are virtually identical with 2% being the largest descrepancy of the minimum Cp values. The same comparisons for the extrapolated cases shown in Figures 4(a) and 4(c) reveal very favourable agreement, even if not as good as in the interpolated case. The reason for these small descrepancies, could be either due to a relative inability of the NN to extrapolate away from its training regime or because the specific reduced ramp rate training regime used was inadequate to provide the NN with an 'understanding' of the flow physics outside this regime.

Figure 5 presents the C_p time histories for several chordwise locations on the suction side of the wing at 58% span. Results are shown for both the extrapolated (Figures 5(a) and (c)) and interpolated (Figure 5(b)) cases. On the graphs, the C_p values are shifted by 5 to allow for a clear presentation. The footprint of the DSV is visible in the time histories and can be seen at a dimensionless time of 110. The interpolated case was in excellent agreement with experiments while fair agreement was also obtained for the extrapolated cases. The NN model provided better comparison for the interpolated case (Figure 5(b)) and the RNN compared more favourably in the extrapolated cases (Figures 5(a and c)).

5 Dynamic Stall Modelling

The next step in the exploration of the ability of the NN to model DS, was to increase the complexity of the problem by introducing the yaw angle as an extra variable. The NN was then trained to approximate the functional: $C_p = C_p(t, \alpha, k, \beta, x, y)$. Cases 1-12 of Table 2, refer to sinusoidal pitching motions with an amplitude and mean angle of 10^o for different yaw angles. Various combinations of these cases were used to train and validate the NN, in order to assess it's interpolating as well as extrapolating capabilities.

5.0.2 Interpolation in Reduced Frequency and Yaw

The training cases were (a) case 4 ($k = 0.1, \beta = 0^o$), (b) case 5 ($k = 0.1, \beta = 10^o$), (c) case 8 ($k = 0.15, \beta = 0^o$) and (d) case 9 ($k = 0.15, \beta = 10^o$), while the validation case was case 7 ($k = 0.12, \beta = 8^o$) of Table 2. Figure 6(a) presents the comparison of the CFD simulation and NN prediction for case 7 of Table 2, at 20^o of angle of attack during upstroke. The comparison is outstanding, as both the magnitude and the finer details of the C_p contour topology have been captured by the NN. From this results it becomes evident that at purely interpolating conditions, the smallest possible number of datasets, namely 4 for 2 interpolating parameters, are adequate for prediction, provided that the values of these parameters are reasonably close for the flow physics to be similar.

5.0.3 Interpolation in Reduced Frequency and Extrapolation in Yaw

The training cases were (a) case 5 ($k = 0.1, \beta = 10^o$), (b) case 6 ($k = 0.1, \beta = 15^o$), (c) case 9 ($k = 0.15, \beta = 10^o$) and (d) case 10 ($k = 0.15, \beta = 15^o$), while the validation case was case 7 ($k = 0.12, \beta = 8^o$) of Table 2. Figure 6(b) presents the comparison of the CFD simulation and NN prediction for case 7 of Table 2, at 20^o of angle of attack during upstroke. The comparison remains very favourable, as both the magnitude and the overall details of the C_p contour topology have been captured by the NN. Again, both the magnitude as well as the location of the DSV shedding are in excellent agreement between the CFD simulation and the NN prediction. Analysis of the CFD data obtained at yawed conditions indicated that yaw has a mild effect on 3-DDS and this may be the reason good results are obtained when the NN is asked to extrapolate predictions at yaw angles outside its training envelope.

5.0.4 Extrapolation in Reduced Frequency and Interpolation in Yaw

The training cases were: (a) case 1 ($k = 0.05, \beta = 0^o$), (b) case 3 ($k = 0.05, \beta = 15^o$), (c) case 4 ($k = 0.1, \beta = 0^o$) and (d) case 6 ($k = 0.1, \beta = 15^o$), while the validation case was case 7 ($k = 0.12, \beta = 8^o$) of Table 2. Figure 6(c) compares the CFD simulation and NN prediction for case 7 of Table 2, for an angle of attack of 10^o during upstroke. Variation of the reduced frequency produced in this case a non linear effect to the C_p and training based on only two different values of reduced frequency is evidently insufficient to provide an accurate prediction. However, one would expect the descrepancy between the actual and predicted values during the formation stage of the DSV to be closer, since the reduced frequency of $k = 0.12$ is not far from the reduced frequencies used for training of $k = 0.05$ and $k = 0.1$.

Following the previous remark, the training dataset was augmented with case 8 of Table 2 and the case chosen for validation was case 11. Figure 6(d) shows the comparison of the CFD simulation and NN prediction for case 11 of Table 2, again at 20^o during upstroke. The comparison is now is very good, since the physics incorporated in the training datasets were adequate for an accurate extrapolating prediction.

Figures 7((a) and (b)) show the C_p histories for the cases described in paragraphs 5.0.2 and 5.0.4 respectively for transducers placed on the wing's mid span. The interpolating case shown in Figure 7(a) shows an excellent agreement between the CFD computations and NN prediction, whereas the extrapolating case shown in Figure 7(b) shows a slight phase shift between CFD and NN in the location of the DSV. The same conclusion can also be drawn by examining Figure **??**(c) where the NN predicted DSV footprint is shown slightly closer to the wing's trailing edge.

6 Conclusions

CFD computations were used to provide insight in the flow topology during 3-D DS combining rotation and pitch or pitch at fixed yaw angles. The resulted flow configuration revealed a distorted omega-shaped vortex with a strong influence on the surface pressure of the wing. The omega-shaped 3D-DSV was also found to interact with the tip vortices. Neural Networks were then used to model the wing loading using CFD and experimental results for training. It was found that the NN can provide very accurate interpolations within their training envelope as well as adequate extrapolations provided enough cases are used during the training phase.

Overall the NN was found to be more effective in interpolating results than the RNN. On the other hand the RNN was better when extrapolations were needed. The training phase of the NN was found to be faster than for the RNN.

Acknowledgements

This work is sponsored by the Engineering and Physical Sciences Research Council (EPSRC) of UK under the grant GR/R7965/01. The authors are gratefull to Profs. F. Coton and R. Galbraith of Glasgow University for allowing the use their experimental data during this project.

References

[1] McCROSKEY, W.J., CARR, L.W., McALISTER, K.W., "Dynamic Stall Experiments on Oscillating aerofoils", *AIAA Journal*, **14**, (1), pp. 57-63, 1976.

[2] EKATERINARIS, J.A., *et al.*, "Present Capabilities of Predicting Two-Dimensional Dynamic Stall", AGARD Conference Papers CP-552, Aerodynamics and Aeroacoustics of Rotorcraft, 1995.

[3] EKATERINARIS J.A. and Platzer M.F., "Computational Prediction of Airfoil Dynamic Stall", *Progress in Aerospace Sciences*, **33**(11-12), pp. 759-846, 1997.

[4] VISBAL M.R., "Effect of Compressibility on Dynamic Stall", *AIAA Paper 88-0132*, 1988.

[5] BARAKOS G.N. and DRIKAKIS D., "Computational Study of Unsteady Turbulent Flows Around Oscillating and Ramping Aerofoils", *Int. J. Numer. Meth. Fluids*, **42**(2), pp. 163-186, 2003.

[6] NEWSOME R. W., "Navier-Stokes Simulation of Wing-Tip and Wing-Juncture Interactions for a Pitching Wing", *AIAA Paper 94-2259*, 24th AIAA Fluid Dynamics Conference, Colorado Springs, Colorado, USA, June 20-23 1994.

[7] SCHRECK S.J. and HELIN H.F., "Unsteady Vortex Dynamics and Surface Pressure Topologies on a Finite Wing", *Journal of Aircraft*, **31**(4), pp. 899-907, Jul-Aug 1994.

[8] MORGAN, P.E. and VISBAL M.R., "Simulation of Unsteady Three-Dimensional Separation on a Pitching Wing", *AIAA paper 2001-2709*, 31st AIAA Fluid Dynamics Conference and Exhibit, Anaheim, CA, 11-14 June 2001.

[9] SPENTZOS, A. *et al.*, "Investigation of Three-Dimensional Dynamic Stall using Computational Fluid Dynamics", *AIAA Journal*, **43**, (5), pp. 1023-1033, 2005.

[10] TANG, D.M. and DOWELL, E.U.,"Comments on the ONERA Stall Aerodynamic Model and its Impact on Aeroelastic Stabiliry", *Journal of Fluids and Structures*, **10**, pp. 353-366, 1996.

[11] BEDDOES, T.S. and LEISHMAN, J.G., "A Generalised Model for Airfoil Unsteady Aerodynamoc Behaviour and Dynamic Stall Using the Indicial Method", 42nd Annual Forum of the American Helicopter Society, 1986.

[12] VEPA, R., "Non-Linear Aerodynamics Modelling for Modal Analysis of Aircraft Dynamics", Paper 94, CEAS Aerospace Aerodynamics Conferences, Cambridge, UK, 10-12 June 2002.

[13] FALLER, W.E., SCHRECK S.J. and HELIN, H.E., "Real-Time Model of Three Dimensional Dynamic Reattachment Using Neural Networks", *Journal of Aircraft*, **32**, (6), pp. 1177-1182, 1995

[14] FALLER, W.E. and SCHRECK S.J., "Unsteady Fluid Mechanics Applications of Neural Networks", *AIAA Journal*, **34**, (1), pp. 48-55, 2005

[15] BADCOCK, K.J., RICHARDS, B.E. and WOODGATE, M.A. "Elements of Computational Fluid Dynamics on Block Structured Grids Using Implicit Solvers", *Progress in Aerospace Sciences*, **36**(5-6), pp. 351-392, 2000.

[16] WILCOX D.C.,"Reassessment of the Scale-Determining Equation for Advanced Turbulence Models", *AIAA Journal*, **26**(11), pp. 1299-1310, 1988.

[17] KOLMOGOROV A.T., "On the representaton of continuous functions of many variables by superposition of continuous functions of one variable and addition"', *Translated in American Mathematical Society Translations* **28** (2), pp. 55-59, 1963.

[18] COTON F.N. and GALBRAITH McD, "An Experimental Study of Dynamic Stall on a Finite Wing", *The Aeronautical Journal*, **103** (1023), pp. 229-236, 1999.

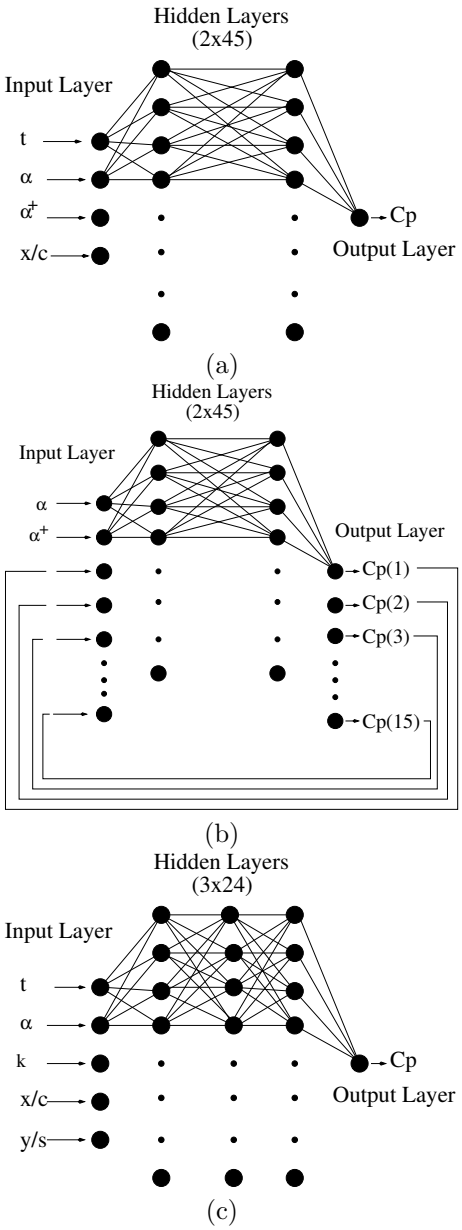
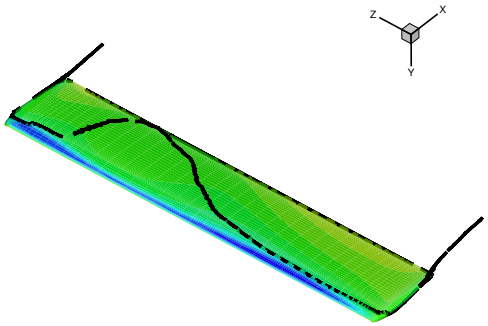
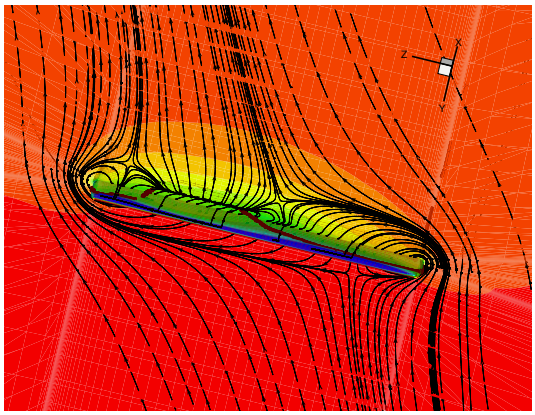


Figure 1: Schematics of employed NN models: (a) 2-layer, no feedback, (b) 2-layer with feedback, (c) 3-layer, no feedback.

Figure 2: cores for three different reduced rates and two different yaw angles, corresponding to conditions 1 (a), 3 (b), 4 (c), 6 (d), 8 (e) and 10 (f) of table 2 respectively. The angle of attack is 20^o.



(a)



(b)

Figure 3: Vortex cores (a) and a spanwise 2D slice at the wing's half chord corresponding to condition 15 of table 2 respectively. The angle of attack is 14^o.

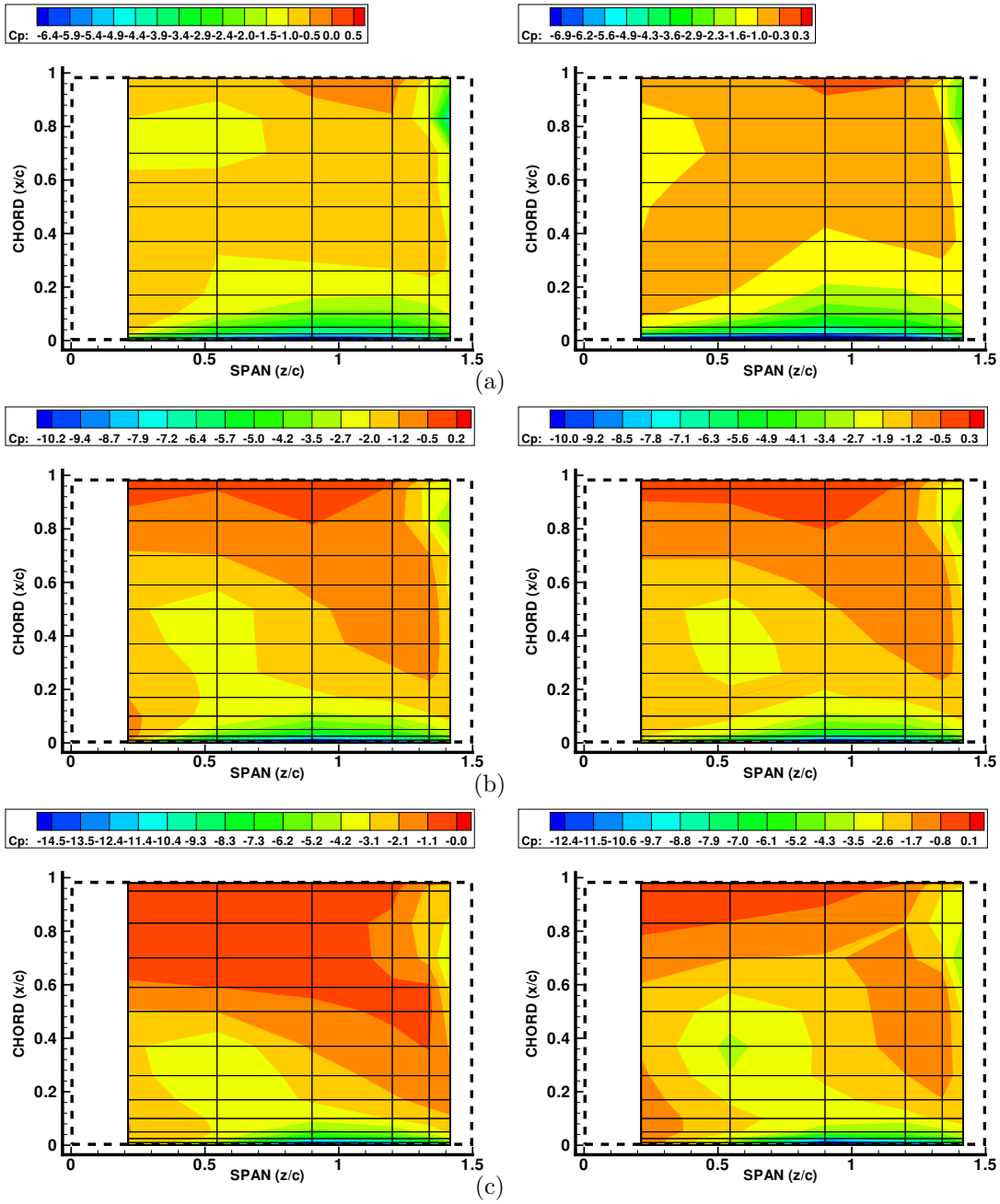
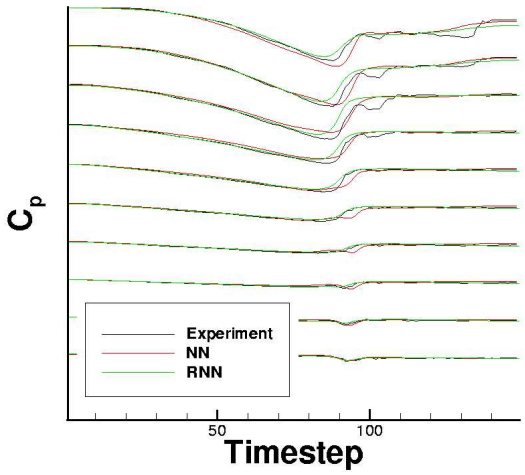
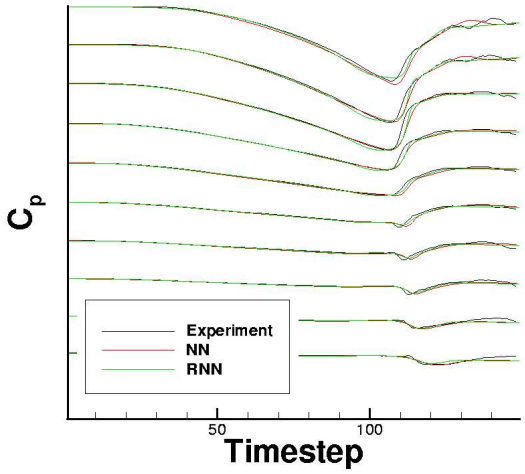


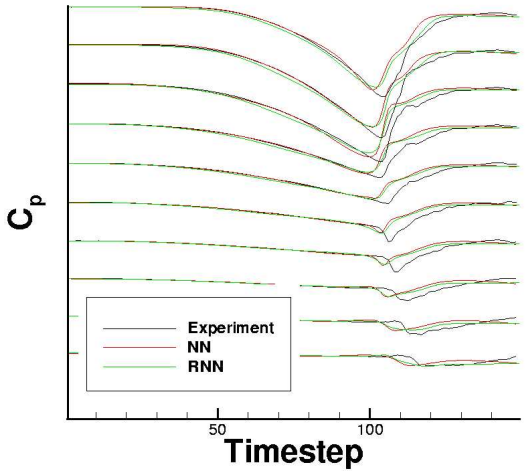
Figure 4: Comparison between experiments by Coton and Galbraith [18] (left) and NN predictions (right) for the surface pressure distribution on the suction side of the square NACA-0015 wing with rounded tips. Ramping motion between -5 and 40 degrees of incidence, $Re = 1.47 \times 10^6$ and $M = 0.16$. (a) case 1, AoA=23.84^o, (b) case 5, AoA=28.74^o and (c) case 8, AoA=35.55^o of Table 1.



(a)



(b)



(c)

Figure 5: Comparison between the predictions of the NN (red) and RNN (green) models against experimental and CFD data for cases (a) 1, (b) 5 and (c) 8 of Table 1 respectively.

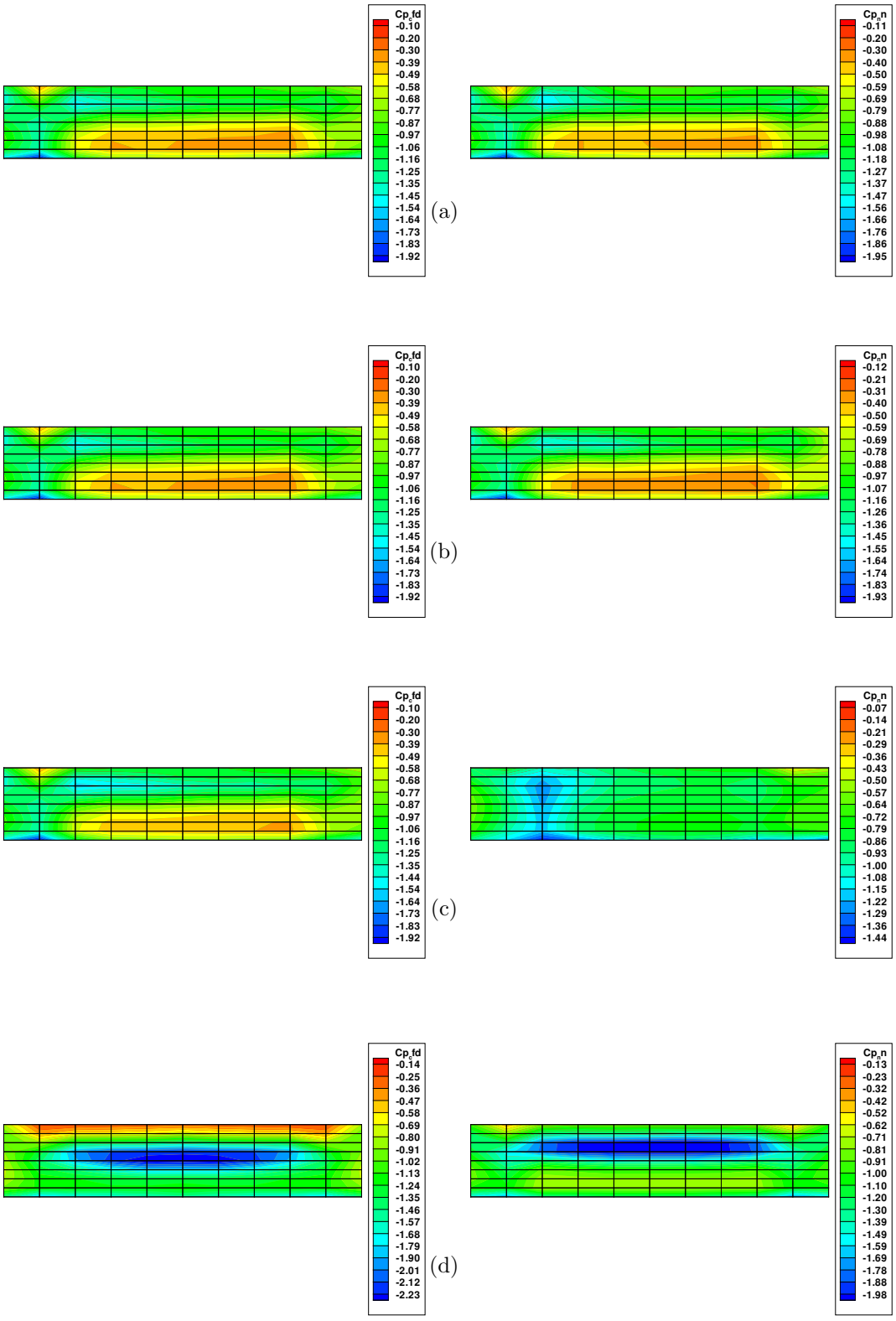
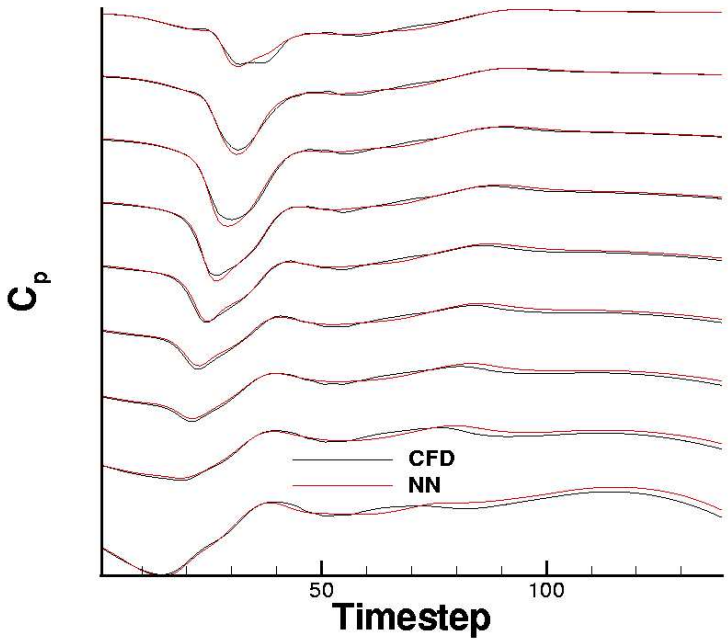
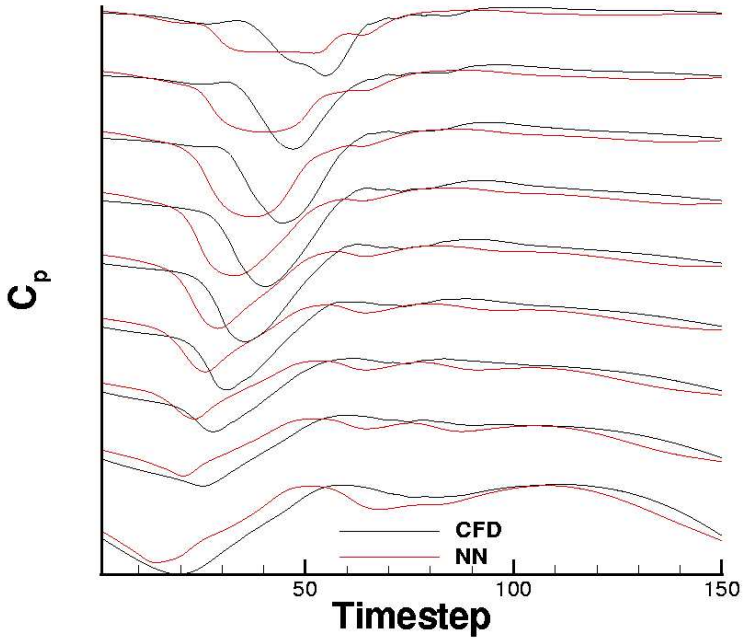


Figure 6: CFD computations (left) and NN predictions (right). All cases refer to Table 2. The NN was trained using cases (a) 4,5,8 and 9 (validation case was case 7), (b) 5,6,9 and 10 (validation case was case 7), (c) 1,3,4 and 6 (validation case was case 7) and (d) cases 1,3,4,6 and 10 (validation case was case 11). Note that the AR has been compressed for illustration purposes.



(a)



(b)

Figure 7: Comparison between the predictions of the NN (red) and RNN (green) models against experimental and CFD data for cases (a) 7 and (b) 11 of Table 2 respectively.

Case	Reduced Ramp Rate	Ramp Rate (deg/s)	Used for Training/Prediction
1	0.0044	67.34	Prediction
2	0.0061	91.26	Training
3	0.0095	143.36	Training
4	0.0120	184.16	Training
5	0.0140	213.85	Prediction
6	0.0190	283.97	Training
7	0.0250	371.97	Training
8	0.0300	429.02	Prediction

Table 1: Details of the experimental data used [18] for the training and validation of the NN. $Re = 1.4710^6$, $M = 0.16$ and the range of angles was $-5^o - 39^o$.

Case	Reduced Frequency	Yaw/Rotation	β	M	AR	μ
1	0.05	Y	0^o	0.3	10	-
2	0.05	Y	5^o	0.3	10	-
3	0.05	Y	15^o	0.3	10	-
4	0.10	Y	0^o	0.3	10	-
5	0.10	Y	10^o	0.3	10	-
6	0.10	Y	15^o	0.3	10	-
7	0.12	Y	8^o	0.3	10	-
8	0.15	Y	0^o	0.3	10	-
9	0.15	Y	10^o	0.3	10	-
10	0.15	Y	15^o	0.3	10	-
11	0.20	Y	0^o	0.3	10	-
12	0.40	Y	0^o	0.3	10	-
13	0.10	Y	0^o	0.25	5	-
14	0.10	Y	0^o	0.55	5	-
15	0.15	R	-	0.15	5	0.33

Table 2: Details of the CFD performed. The wings used were based on a NACA0012 aerofoil section with rounded tips. The pitching component of the motion, where appicable, was a sinusoidal oscillation with both mean angle and amplitude of 10^0, while the Reynolds number was always $Re = 10 \times 10^6$. In the runs that involved rotational motion, the rotating frequency was the same as the pitching frequency.