TRANSONIC WING BUFFET LOAD PREDICTION AT STRUCTURAL VIBRATION CONDITIONS

R. Zahn*, V. Völkl*, M. Zieher*, C. Breitsamter*
* Technical University of Munich, Chair of Aerodynamics and Fluid Mechanics, Boltzmannstr. 15, 85748 Garching, Germany

Abstract

This paper presents a reduced-order modeling (ROM) approach based on a hybrid neural network in order to calculate wing buffet pressure distributions due to structural eigenmode-based deformations. For this hybrid ROM a convolutional autoencoder (CNN-AE) and a long short-term memory (LSTM) neural network are connnected in a serial fashion. The NASA Common Research Model (CRM) with the FERMAT structural model is used for forced-motion computational fluid dynamics (CFD) simulations at transonic buffet conditions. Aerodynamic responses are obtained as a result of the eigenmode-based deformations. As eigen shape the first symmetric wing bendig mode is selected. The unsteady simulations are carried out with the triangular adaptive upwind (TAU) solver of the German Aerospace Center (DLR) and the hybrid ROM is trained with this data. When investigating the prediction capability of the hybrid ROM a high accuracy with respect to the forced-motion buffet loads is indicated.

Keywords

Deep Learning; Convolutional Autoencoder (CNN-AE); Long Short-Term Memory Neural Network (LSTM); hybrid Reduced Order Model (ROM), Wing Buffet Aerodynamics; Forced Structural Vibrations; NASA Common Research Model (CRM)

NOMENCLATURE

Nomenclature

α	angle of attack	deg
A_{max}	maximum deflection amplitude	m
AR	aspect ratio	
A_{ref}	wing reference area	m^2
b	wing span	m
$\boldsymbol{b}$	bias vector	
$\boldsymbol{c}$	cell state vector of LSTM cell	
C	channel dimension	
c_L	design lift coefficient	
c_p	pressure coefficient	
Δt	physical time step	s
η	dimensionless spanwise position	
f	reference scaling factor	
$\boldsymbol{f}$	forget gate vector of LSTM cell	
f_t	trip term of SA turbulence model	
f_{t2}	turbulence suppression term of SA turbulence model	
H	data height	
$\hat{y}_i$	c_p value obtained by the neural network	
$\boldsymbol{h}$	hidden state vector of LSTM cell	
$\boldsymbol{i}$	input gate vector of LSTM cell	
k_{red}	reduced frequency	
c_{ref}	mean aerodynamic chord	m
Ma_{cruise}	design cruise Mach number	
Ma_∞	freestream Mach number	
N_S	number of data points	
$\boldsymbol{o}$	output gate vector of LSTM cell	
$\phi_{0.25}$	quarter chord sweep angle	deg
q	displacement vector	
R	specific gas constant	J/(kg · K)
Re	Reynolds number	
SF	scaling factor	
σ	sigmoid activation	
S_M	scale of wind tunnel model	
Sr	Strouhal number	
T_{Buffet}	buffet period	s
TR	taper ratio	
T_0	total temperature	K
W	data width	
$\boldsymbol{W}$	weight matrix	
$\boldsymbol{x}$	model input vector	
y_i	c_p value obtained by the numerical simulation	
$y+$	dimensionless wall unit	
$\boldsymbol{y}$	model output vector	

doi: 10.25967/610103

Acronyms

ADAM	adaptive moment estimation algorithm [1]
BN	batch normalization
BP	backpropagation
BPTT	back-propagation through time [2]
CFL	Courant-Friedrichs-Levy number
CNN-AE	convolutional neural network autoencoder
CNN	convolutional neural network
CRM	NASA Common Research Model
(D)DES	(Delayed) Detached Eddy Simulations
DLR	German Aerospace Center
FC	fully-connected layer
FEM	finite element model
FERMAT	structural model of the CRM developed by Klimmek [3]
HTP	horizontal tail plane
LRZ	Leibniz Supercomputing Center
LSTM	long short-term memory neural network
MSE	mean square error
QCR	quadratic constitutive relation for SA turbulence model [4]
ReLU	rectified linear unit
RNN	Recurrent Neural Network
ROM	reduced order model
SAPRBS	smoothed amplitude-modulated pseudo-random binary signal
SA	Spalart-Allmaras turbulence model
TAU	triangular adaptive upwind solver
TPS	thin-plate splines
URANS	Unsteady Reynolds-averaged Navier-Stokes
VTP	vertical tail plane

1. INTRODUCTION

Determining the limits of the flight envelope as precisely as possible is a main task in the analysis and certification process as well as the aerodynamic and structural optimization of aircraft. Apart the quasi-steady stall limit, the flight envelope limits are determined by dynamic aeroelastic phenomena, in which the coupling of aerodynamic, structural-elastic and inertia forces are present. For so-called "off-design" conditions of commercial aircraft with supercritical wings, structural shaking phenomena can occur as a result of the prevailing shock fluctuations in chord and span direction on the wing. Combined with shock-induced, locally separated flow this is referred to as buffeting and represents a limit of the flight envelope, cf. Iovnovich [5], Raveh [6]. The aerodynamic excitation forces are related to the term "high-speed buffet" [5], while the scenario including the motion induced by the aerodynamic forces and the structural response is referred to as "buffeting" [6].

Transonic buffet/buffeting is characterized by a pronounced shock-boundary-layer interaction, which leads to shock induced, quasi-periodic flow separations, see Paladini [7]. Linked to this is a self-sustaining shock oscillation, which interacts with the dynamically developing separation area. This results in different complex flow conditions on airfoils and swept wings of finite aspect ratio, see Masini [8]. Based on the results of various numerical and experimental studies, buffet is caused by the interaction of various physical mechanisms. Stability analyses based on linearized unsteady Reynolds-averaged Navier-Stokes equations (Unsteady Reynolds-averaged Navier-Stokes, URANS) show the presence of global flow instabilities, see Crouch [9], He [10], Timme [11]. On the one hand, there is a mode associated with the shock movement in chord direction (Sr $\approx$ 0.05 – 0.1; Strouhal number based on mean aerodynamic chord), on the other hand modes associated with the propagation of so-called “buffet cells” in span direction occur (Sr $\approx$ 0.2 – 0.6; propagation speed: u/uref $\approx$ 0.25 – 0.5) [8], [11]. The characteristic frequencies are of more broadband character on the wing of finite aspect ratio compared to those on airfoils, see Ohmichi [12].

For airfoil flows, investigations with specified rigid-body vibrations show that, if the vibration amplitude is sufficiently large, a “lock-in effect” of buffet and vibration frequency occurs, see Raveh [13]. This can also be seen in corresponding scenarios of elastically supported airfoils, see Scharnowski [14]. Overall, therefore, the structural vibrations can influence the buffet-specific fluctuations, see Nitzsche [15].

However, even the application of URANS/(D)DES methods for the treatment of buffet flows is not possible for the large number of parameters to be varied in an aircraft design process due to the computational effort. For the calculation of buffeting, the flow solver and the structural solver must also be coupled, which generates additional computational effort. Therefore, especially with regard to applications in an industrial context, a compromise between the desired physical accuracy and the resulting computational costs has to be reached. One way to reduce computational resources is to use reduced order models (ROMs), which are trained and validated using CFD simulation data or experimental data sets. The trained model can then be applied to other inflow conditions in order to predict characteristic aerodynamic variables. Consequently, the computational effort of aeroelastic investigations can be significantly reduced. A comprehensive overview of ROM approaches in the context of transient aerodynamics and aeroelasticity can be found in Kou and Zhang [16].

In the present study, structural deflections have been imposed on an already established buffet instability in order to generate a highly nonlinear flow field. As a test case, the NASA Common Research Model (CRM) configuration has been chosen. Based on the FERMAT structural model, a symmetric mode shape is considered for the simulation process. For the generation of the training data set, forced-motion CFD simulations of the mode shape are carried out. In order to evaluate the performance capability of the trained hybrid ROM, it is applied to predict buffet pressure distributions due to harmonic excitations of the considered mode shape.

2. TEST CASE

2.1. NASA Common Research Model (CRM)

The Common Research Model (CRM) is as generic aircraft configuration with a supercritical transonic wing, featuring wing, body and a horizontal-tail with 0° tail setting angle. Figure 1 shows different views of the full-span model of the CRM configuration. For the following investigations, only a half-span model of the CRM configuration is simulated at buffet conditions in order to save computational effort. Table 1 summarizes important design properties of the selected CRM configuration. The absolute dimensions refer to a wind tunnel model, cf. [17]

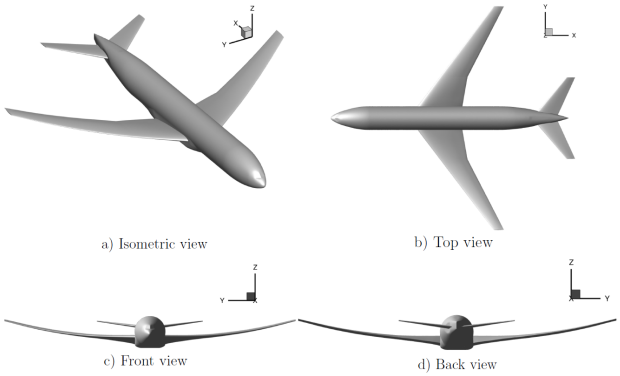

FIG 1. Different views of the CRM configuration.

TAB 1. Design properties of the CRM configuration (wind tunnel model)

Quantity	Symbol	Value
Design cruise Mach number	Ma_{cruise}	0.85
Design lift coefficient	c_L	0.5
Scale of wind tunnel model		1:37
Wing reference area	A_{ref}	0.280 m^2
Wing span	b	1.586 m
Mean aerodynamic chord	c_{ref}	0.189 m
Aspect ratio	AR	9
Taper ratio	TR	0.275
Quarter chord sweep angle	$\phi_{0.25}$	35°

2.2. FERMAT structural model

The FERMAT structural model is a finite element model (FEM) for the CRM, developed by Klimmek [3] to represent overall aircraft characteristics and to enable static and dynamic aeroelastic investigations with the CRM geometry. Therefore, it includes nodes for the fuselage, the wings, the vertical (VTP) and horizontal (HTP) tail planes, the pylons and the engines, see Fig. 2 left side. Two different mass cases are available for the FERMAT model. The C1 case is the maximum zero fuel weight configuration and the C2 case is the configuration with maximum take-off weight.
For the following investigation the C2 case of the FERMAT structural model is used for modeling eigenmode-based structural deformations. This configuration consists of 56 modes, including six rigid body and 50 elastic modes. As the investigated CRM configuration only contains wing, body and a horizontal-tail the structural model is reduced to that as well. Figure 2 shows on the left side the full structural model by Klimmek [3] and on the right hand side the reduced model for this investigation [18].

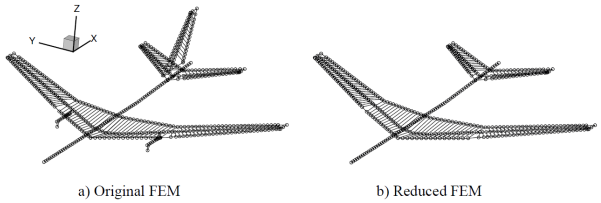

FIG 2. Comparison of the structural FEM by Klimmek [3] (a) and the reduced structural FEM [18] (b)

3. METHODS

3.1. Numerical setup

The numerical investigations are carried out using a hybrid computational mesh generated and applied by Ehrle et al. [17]. The half-model has approximately $36 \cdot 10^6$ elements. The mesh contains unstructured cells in the farfield and structured cells in the proximity of all surfaces. The boundary layer has a sufficient high resolution with a dimensionless wall unit of $y+ < 1$. In order to provide a proper resolution of the area of shock motion and flow separation a block with isotropic hexahedral elements is added on top of the boundary layer elements on the wing's upper surface, see Fig. 3. The cell sizing of the block is done in order to achieve a CFL (Courant-Friedrichs-Levy) number of 1.

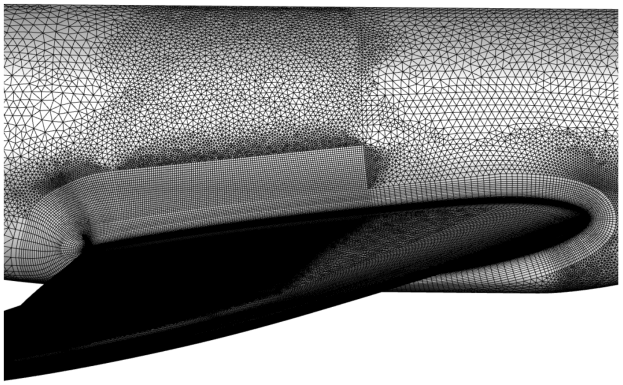

FIG 3. Hybrid numerical mesh of the CRM configuration with a block containing hexahedral elements on the wing's suction side.

The buffet conditions are set to a freestream Mach number of $Ma_\infty = 0.85$, a Reynolds number of $Re = 30 \cdot 10^6$ and an angle of attack of $\alpha = 5°$. According to Ehrle et al. [17] nitrogen with a gas constant of $R = 296.8 \frac{J}{kgK}$ at a total temperature of $T_0 = 115K$ is chosen as working fluid. The sutherland parameters are the same as in [17]. In order to be able to compare the wind tunnel measurements [19] with the numerical simulation, static mesh deformation is applied ahead. The used aeroelastic wing deformation data was gathered via stereo pattern tracking during the wind tunnel testing [19].
URANS simulations are carried out using the DLR-TAU code with the Spalart-Allmaras (SA) turbulence model without trip-term f_t and turbulence suppression term f_{t2}. A previous numerical study at flight Reynolds numbers of the CRM by Illi et al. [20] showed corner flow separation not in

accordance with the experiment. So additionally a quadratic constitutive relation (QCR) [4] extension is activated. As this fusion of an eddy-viscosity turbulence model with the QCR extension is shown by Togiti et al. [21] to cause a reduced corner flow separation.
A central differencing scheme is used for spatial discretization and an implicit backward Euler scheme is used for the time integration of the URANS equations. As a linear solver the LU-SGS scheme is taken and the gradients are reconstructed by the Green Gauss scheme. In order to improve convergence a multigrid approach with a 3v cycle and dual time stepping is chosen. The physical time step is set to $\Delta t = 1 \cdot 10^{-5} s$ with 100 inner iterations per physical time step. For the inner iterations (implicit pseudo timesteps) a CFL number of 2 is chosen. The meanflow fluxes are discretized via the Kok skew symmetric scheme and the turbulence fluxes are discretized via the first order Roe scheme. A reduced numerical dissipation of small scale structures is achieved by a ratio of 0.4 between matrix and scalar dissipation and a fourth order dissipation coefficient of 1/256.
The URANS simulations carried out with these settings are used as training, validation and test data for the Reduced Order Model (ROM) developed in this study.

3.2. Structural vibration input

For the above (Section 3.1) described numerical setup of the URANS simulations the structural vibrations have to be implemented as well. This is achieved by mapping the structural deflections of the FEM model onto the surface grid via the thin-plate splines (TPS) algorithm. As the focus of this study is on the wing buffet instability, exemplarily only one eigenmode is considered. The chosen eigenmode is the mode 11 representing the first symmetric wing bending mode. In accordance with the work of [18,22], the structural deflections are scaled proportional to the reference chord length c_{ref} with a scaling factor SF of:

$$SF = f \cdot \frac{c_{ref}}{A_{max}} \tag{1}$$

Here A_{max} denotes the maximum deflection amplitude of the considered deformation and f the reference scaling factor of $f = 0.01 \mathrel{\widehat{=}} 1\%$. f is chosen in order to guarantee stable numerical simulations. The mode 11 is visualized in Figure 4 with increased deflections for clarity.

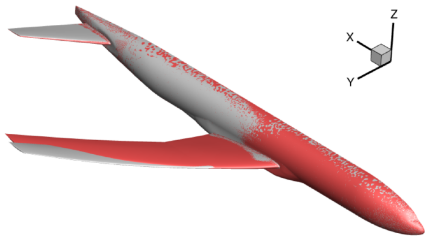

FIG 4. Structural-eigenmode-based surface deformations (increased) of the FERMAT-C2 configuration (red) for the first symmetric wing bendig mode and the non-deformed CRM geometry (grey).

A smoothed amplitude-modulated pseudo-random binary signal (SAPRBS) [23] is used as excitation signature input. This signal contains a high amount of different frequencies and amplitudes. During the generation process of this signal the minimum and maximum excitation amplitude are set equally to a reference scaling factor of $f = 1\%$. The first 2000 time steps of this signal are shown in Figure 5.

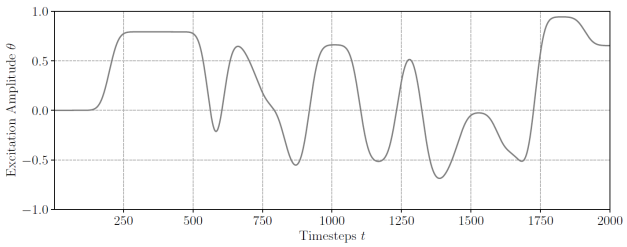

FIG 5. Smoothed SAPRBS time series for the forced-motion excitation of the mode shapes.

3.3. Neural network setup - ROM

In this section the utilized deep learning model is described. The first subsection is about the working principle of CNNs, the second one about the LSTM neural network and the third one about the architecture of the applied hybrid neural network.

3.3.1. Convolutional Neural Network (CNN)

Convolutional neural networks (CNN) are specially designed neural networks for processing data with a grid-like topology [24]. CNNs are mostly used for image data, as those are 2D grids of pixels. If time series are recorded with a fixed time interval, they can also be processed by CNNs as it is then grid-like as well. A CNN consists of three main layers: a convolutional layer, a pooling layer and a fully-connected (FC) layer. The number of convolutional and pooling layers can be adjusted in order to optimize the architecture for the application scenario.
For better explanation a data set with 3D matrix containing spacial information is chosen as example. The data points can be addressed via their indices $(i,\ j,\ o)$. Using the nomenclature of Goodfellow et al. [24] and Rosov and Breitsamter [25] the convolution is then defined like this:

$$\boldsymbol{y}_{k,i,j} = \sum_{l=0}^{C_{in}-1} \sum_{m=0}^{H_w-1} \sum_{n=0}^{W_w-1} \boldsymbol{x}_{l,i\times s+m,j\times s+n} \boldsymbol{W}_{k,l,m,n} + \boldsymbol{b}_k \tag{2}$$

with the output $\boldsymbol{y}_{k,i,j}$ at the indices (k, i, j). The dimension of the input $\boldsymbol{x}$ is $C_{in} \times H_x \times W_x$ with the number of input channels C_{in}, the height H_x and width W_x of the input data. The matrix of the filter weights is $W_{k,l,m,n}$. The filter is also called kernel with the size $C_{in} \times H_w \times W_w \times C_{out}$ and selects important features of the input data. The output channel C_{out} is added to the filter as an additional dimension. The convolution itself is carried out elementwise by the kernel sliding stepwise over the input data, performing element-wise multiplication. By setting the stride s larger than one the input size can be reduced by a factor of s as the data points are picked s-entries apart.
In this study the selected CNN architecture is used as an autoencoder, also called convolutional autoencoder (CNN-AE). The training of the CNN-AE is done in an unsupervised fashion by backpropagation (BP) [26]. Here the CNN-AE encodes a given input and reconstructs it to its output [24]. An autoencoder is built by an encoder, a decoder and a latent space, also called bottleneck, in between. The encoder reduces the high-dimensional data into the low-dimensional latent space and the decoder does the inverse.

3.3.2. Long Short-Term Memory (LSTM)

Hochreiter and Schmidhuber [27] proposed a special type of Recurrent Neural Network (RNN) called Long Short-Term

Memory (LSTM). It is able to predict both long- and short-term dependencies in time series data. One advantage of a LSTM is its robustness against vanishing gradients [28], where the training performance saturates after a small number of training cycles. The LSTM is also capable to capture time-delayed effects existing in unsteady aerodynamic systems.

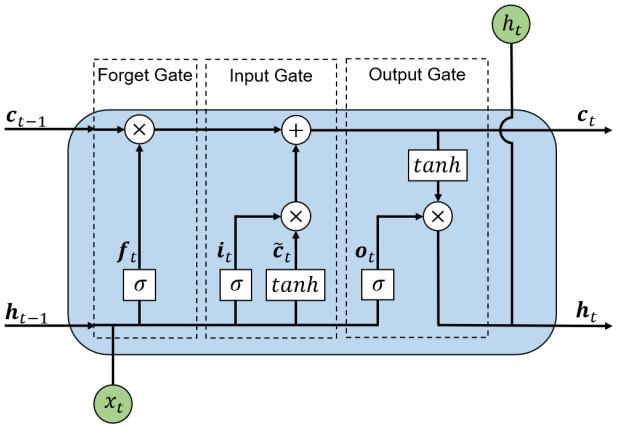

FIG 6. Architecture of the LSTM cell including the characteristic gate structure.

Figure 6 depicts the architecture of a LSTM cell. It consist of three gates, namely the forget gate $\boldsymbol{f}$, the input gate $\boldsymbol{i}$ and the output gate $\boldsymbol{o}$. In order to build a LSTM neural network the hidden layer consists of recurrently connected blocks, also called memory blocks or cells. The forget gate $\boldsymbol{f}$ takes the input $\boldsymbol{x}_t$ of the current time step and the output from the previous time step $\boldsymbol{h}_{t-1}$ (hidden state of the LSTM cell) and computes following equation:

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{W}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right) \tag{3}$$

with the weights $\boldsymbol{W}_f$ and the bias $\boldsymbol{b}_f$. The sigmoid σ acts as an activation and removes parts of the incoming information from the cell. This output is then further processed in the input gate $\boldsymbol{i}$. It takes the same input as the forget gate, namely $\boldsymbol{x}_t$ and $\boldsymbol{h}_{t-1}$:

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{W}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right) \tag{4}$$

Respectively, the weights and bias are $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$, and σ is the sigmoid activation. In a parallel step a new cell state $\tilde{\boldsymbol{c}}_t$ is calculated via:

$$\tilde{\boldsymbol{c}}_t = tanh\left(\boldsymbol{W}_h \boldsymbol{x}_t + \boldsymbol{W}_h \boldsymbol{h}_{t-1} + \boldsymbol{b}_h\right) \tag{5}$$

using a hyperbolic tangent (tanh) activation instead of a sigmoid activation. This new cell state $\tilde{\boldsymbol{c}}_t$ is then used to update the current cell state $\boldsymbol{c}_t$:

$$\boldsymbol{c}_t = \boldsymbol{f}_t \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \tilde{\boldsymbol{c}}_t \tag{6}$$

After this the data is transferred to the output gate $\boldsymbol{o}$. Here the data gets processed in a similar way:

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{W}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right) \tag{7}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \cdot tanh\left(\boldsymbol{c}_t\right) \tag{8}$$

The now updated hidden state $\boldsymbol{h}_t$ and the current cell state $\boldsymbol{c}_t$ are transferred as input to the next hidden layer or the output layer. To be able to predict data the weights and biases have to be set properly. This is achieved by applying back-propagation through time (BPTT) [2] with the adaptive moment estimation (ADAM) [1] algorithm.

3.3.3. The hybrid Reduced Order Model (ROM)

The previous introduced CNN-AE (Section 3.3.1) and the LSTM (Section 3.3.2) are combined into a hybrid ROM architecture so that the forced-motion wing buffet pressure loads can be represented accurately and efficiently. The created hybrid ROM is visualized in Figure 8.
In order to reduce the spatial dimension of the data for the LSTM several levels of the CNN-AE encoder are used. After the LSTM the same number of CNN-AE decoder levels are used to have the same spatial dimension for the output. At each level the encoder performs a multi-channel convolution with a kernel size of 4×4, stride $s = 2$ and padding of one. The input for the hybrid ROM is a sequence of c_p values on the wing suction side from the URANS simulation with forced vibrations (Section 3.2). The data is fed to the encoder as a tensor of size $N_{in} \times C_{in} \times H \times W$, with the number of input time steps N_{in}, the number of input channels C_{in} and H, W the number of grid points in span - and chord-wise direction. For performance reasons those have been chosen by powers of two. At each level of the encoder and the decoder the input data is normalized via batch normalization (BN) and a rectified linear unit (ReLU) activation is applied. In order to have the same output resolution the output channel size of each level is set to two times its input channel size. In the last encoding level the data is flattened and passed through a FC and a tanh activation layer, the inverse is then done in the first decoding level. In the last decoder level the data is rescaled to the value range of [-1,1] via an activation by a tanh layer.
The deformation Δq with respect to structural oscillations or vibrations, respectively, on the other hand is not passed through the CNN encoder, but directly combined with the c_p data before passing the data into the LSTM. This is done via concatenating the scalar deformation values to the end of the c_p vector, as shown in Fig. 7.

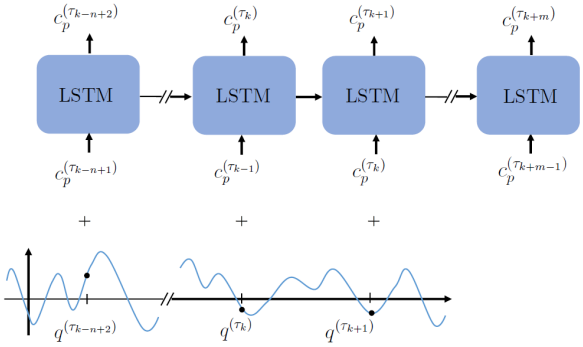

FIG 7. Concatenation of deformation and surface c_p.

The aim of the developed hybrid ROM is to predict the c_p distribution at time steps $k+m$ based on previous snapshots at time steps $k-n+1$ to k. The incidence amplitudes of the deformation are given at time steps $k-n+2$ to $k+1$.

4. RESULTS AND DISCUSSION

In the following section, the flow phenomenology of the buffet cycle will be discussed as well as the application of the hybrid ROM. In the first part, the flow phenomenology of the buffet cycle is briefly discussed. The second subsection will elaborate the data preprocessing and the training of the ROM. In the last subsection the results of the ROM application will be discussed.

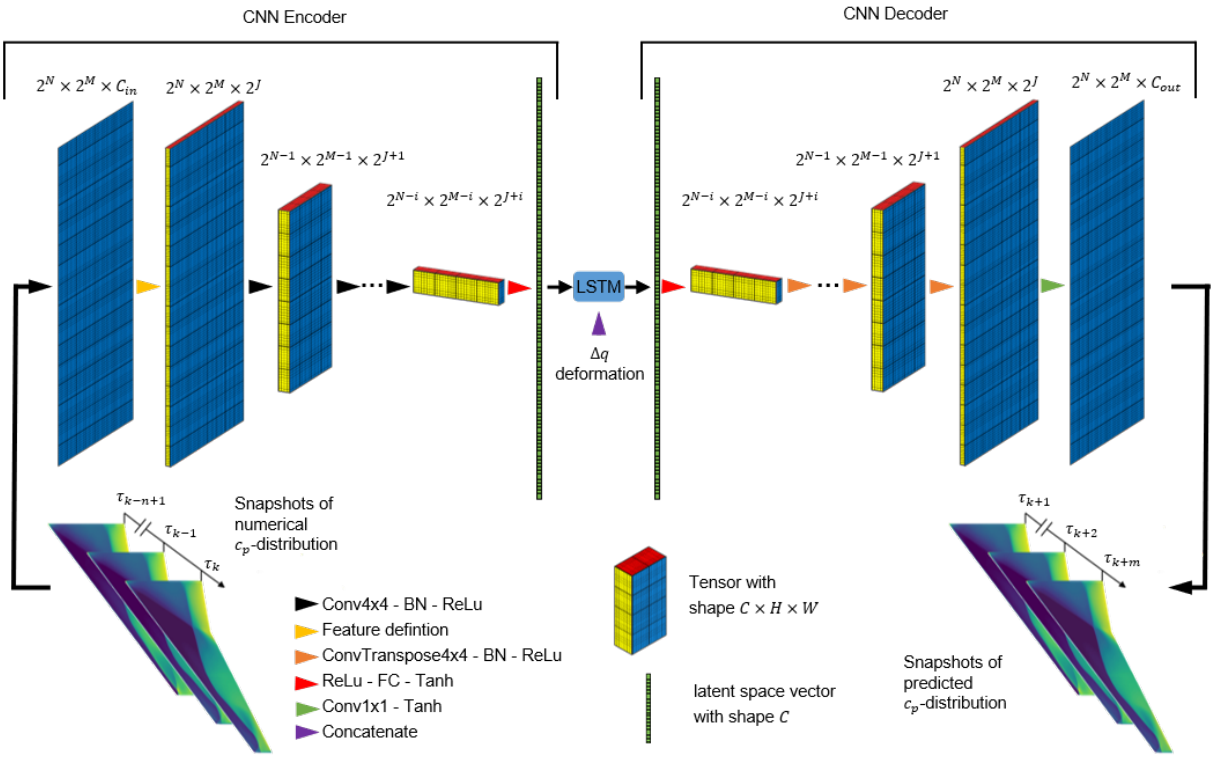

FIG 8. Architecture of the hybrid ROM.

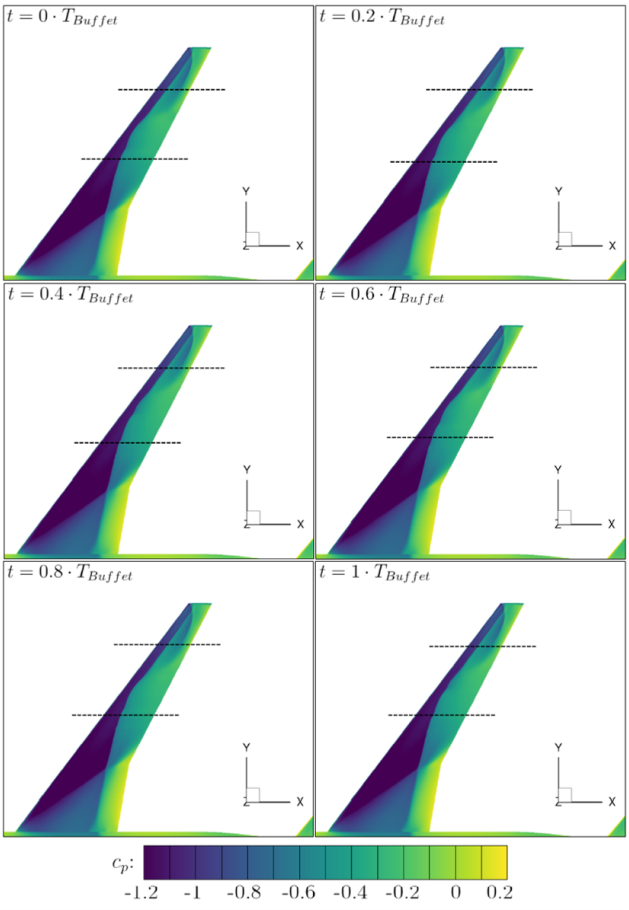

FIG 9. Pressure coefficient (c_p) contour plots of one buffet cycle with the buffet period T_{Buffet} ($Ma_\infty = 0.85$, $Re = 30 \cdot 10^6$, $\alpha = 5°$).

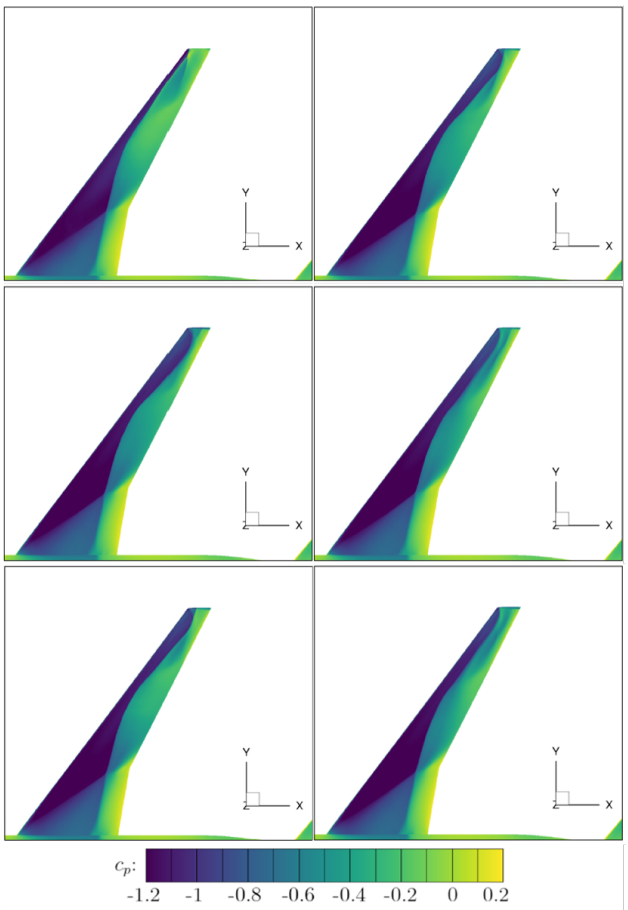

FIG 10. Six consecutive pressure coefficient (c_p) contour plots of the wing suction side for the SAPRBS forcing (Mode 11, $\Delta t = 1 \cdot 10^{-5} s$, $Ma_\infty = 0.85$, $Re = 30 \cdot 10^6$, $\alpha = 5°$).

4.1. Flow phenomenology - buffet cycle

URANS simulations have been carried out with and without structural vibrations. The used numerical setups are described in Sections 3.1 and 3.2, respectively. First the results without structural vibrations will be discussed.
Figure 9 shows six c_p contour plots of a buffet cycle with a buffet period of $T_{Buffet} = 0.0065s$. Close to the wing root (at the bottom of each snapshot) a characteristic λ-shaped two-shock pattern can be seen. Further outboard the pressure distribution changes, indicated by two black dotted lines. These indicate the convection of so called buffet cells which is in alignment with previous numerical studies [5]. The mean c_p [17] from the URANS simulations also agree well in the shock region with the experimental data [19].
The influence of the forced excitation and the resulting structural deflections can be seen in Figure 10. Figure 10 shows six consecutive c_p snapshots. Especially in the wing tip region stronger fluctuations of the shock positions are visible.

4.2. Data processing and ROM training

The data has to be preprocessed in order for the hybrid ROM to be able to process it. The surface mesh on the wing suction side is represented by 168686 data points. This resolution is too fine, therefore the size of the data is reduced by a factor of approximately 2.5 to $128 \times 512 (2^7 \times 2^9)$ data points. This data reduction is done by linear interpolation. For a good representation of the flow field, the wing is divided into two parts, see Fig. 11. Each part has a resolution of 256 nodes in spanwise and 128 nodes in chordwise direction. Afterwards the minimum and maximum pressure values ($c_{p,min}$, $c_{p,max}$) of each data set are used to normalize the data set to [-1,1].

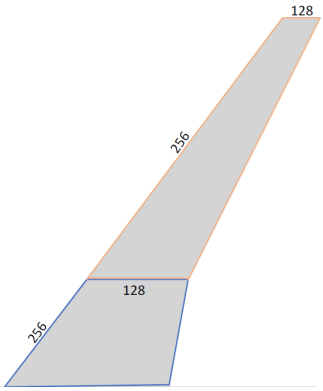

FIG 11. Sketch of the surface mesh interpolation for the hybrid ROM.

Figure 12 depicts the result of the data reduction and normalization for two selected time steps. The difference between the reduced and original resolution can be evaluated by comparing Figure 10 and Figure 12. Still the characteristic λ-shaped two-shock pattern is clearly visible. In spanwise direction a high level of spatial resolution is maintained too.
The next step after the preprocessing is the training of the hybrid ROM. The set of all interpolated and normalized surface c_p snapshots on the wing suction side is now referred to as data set. This is done in two consecutive steps, first the CNN-AE is trained and afterwards the LSTM is trained with the reduced data set provided by the CNN-AE. In order to provide validation data, the data set is split into 80% training data and 20% validation data. The validation data is also used for hyperparameter tuning.

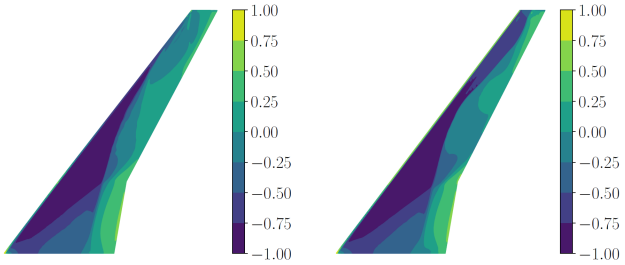

FIG 12. Interpolated and normalized surface c_p resolution on the wing suction side at two selected time steps (symmetric mode 11, SAPRBS excitation).

The CNN-AE is trained in batch mode, with each batch containing 128 c_p snapshots. The encoder and decoder consist of four convolution levels, thereby reducing the spatial resolution of the data from 128×512 to 8×32. As a result the channel sizes are increased from 4 to 512. The FC layer reduces the latent size from 512 to 256 features. The CNN-AE is trained for 15000 epochs with an initial learning rate of 10^{-4}. The objective of the training was to minimize the mean square error (MSE, see Eq. (9)) between the reference numerical data and the corresponding predictions.

$$MSE = \frac{1}{N_S} \sum_{i=1}^{N_S} (\hat{y}_i - y_i)^2 \tag{9}$$

Here N_S is the number of data points in the data set, $\hat{y}$ the value obtained by the neural network and y the value from the numerical simulation. The development of the training and validation losses over the 15000 training epochs of the CNN-AEs is visualized in Figure 13.

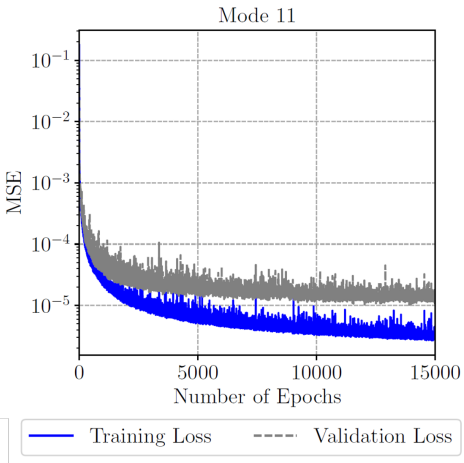

FIG 13. Training and validation losses over the 4000 training epochs of the CNN-AEs.

The result of the training can be seen in Figure 14 where an exemplary snapshot is visualized. On the left side is shown the c_p-snapshot from URANS, in the middle the c_p-snapshot obtained by the trained CNN-AE and on the right the error between both c_p-distributions depicted by the MSE on the wing surface. The MSE is close to zero on almost the entire surface, which indicates a good performance quality of the CNN-AE.
The chosen LSTM architecture contains three layers, with each layer including 256 neurons. The LSTM is also trained in batch mode, with batches of 32 data points each. The sequence length is defined as one. It is trained for 3000

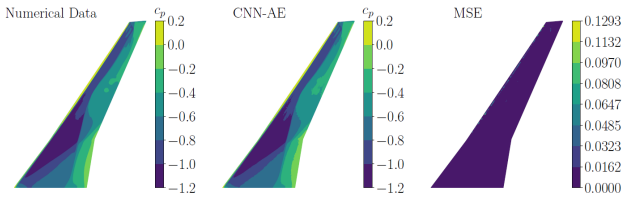

FIG 14. Comparison of a numerical c_p-snapshot (left) and a c_p-snapshot obtained by the trained CNN-AE (middle) (Mode 11, SAPRBS excitation). The corresponding MSE is shown on the right wing surface.

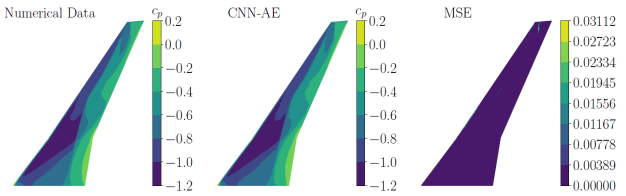

FIG 16. Comparison of a numerical c_p-snapshot (left) and a c_p-snapshot obtained by the trained CNN-AE (middle) (Mode 11, harmonic excitation). The corresponding MSE is shown on the right wing surface.

epochs with an initial learning rate of 10^{-4}. The performance evaluation of the trained hybrid ROM is done via recurrent multi-step predictions using the validation data set. As initializer for the trained model three consecutive c_p-snapshots are used. Afterwards the trained ROM replaces those with predicted c_p-snapshots, as it advances.

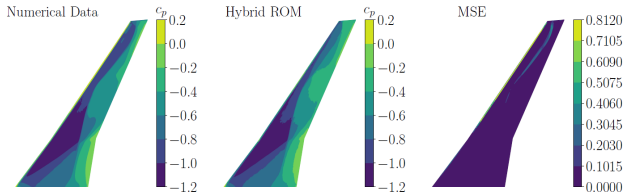

FIG 15. Comparison of a numerical c_p-snapshot (left) and a c_p-snapshot obtained by the trained hybrid ROM (middle) (Mode 11, SAPRBS excitation). The corresponding MSE is shown on the right wing surface.

The performance of the hybrid ROM is visualized in Figure 15. The left side shows a c_p-snapshot from the URANS simulation. In the middle the corresponding prediction of the hybrid ROM is shown. The MSE (right) is close to zero on almost the entire surface, with exception of the leading edge and the shock in the region close to the wing tip. Nevertheless this indicates a good performance quality of the hybrid ROM.

4.3. ROM application and data analysis

After training and validating, the prediction performance of the hybrid ROM is now investigated in more detail. Therefore a new testcase is chosen. Rozov et al. [25] states that a harmonic signal is a well-suited testcase for a ROM trained with an SAPRBS, as the SAPRBS differs significantly from a harmonic signal. The chosen harmonic signal has a reduced frequency of $k_{red} = 0.6$. Two different amplitudes with a reference scaling factor of $f = [0.5\%, 1\%]$ are used for the external forcing of the wing. The URANS simulation and the trained ROM were run for one oscillation period of each harmonic signal.

In order to better evaluate the prediction quality of the trained ROM, first the prediction accuracy of the CNN-AE is investigated. Figure 16 shows one snapshot of URANS results with the harmonic excitation (left), the corresponding result of the trained CNN-AE (middle) and the squared difference between both c_p-distributions as the MSE on the wing surface (right). The overall small MSE on the wing suction side pinpoints the good prediction capability of the trained CNN-AEs.

Subsequently to the performance evaluation of the trained CNN-AE, the performance of the hybrid ROM is evaluated. This is done in a recurrent multi-step prediction mode. Therefore an initializer of 32 c_p-snapshots from the URANS simulation with the corresponding deformation data is used.

For a better comparison the results of the hybrid ROM with the URANS results, the c_p values in chordwise direction are calculated for one selected spanwise positions $\eta = 80\%$. This position is chosen for its distinct changes in surface c_p. The result depicted in Figure 17 showcases that the hybrid ROM is able to predict the overall trend of the c_p values. The deviation of the hybrid ROM from the numerical solution increase with an increasing number of time steps predicted. The chosen spanwise position is $\eta = 80\%$. For the advanced time step larger deviations are shown. Especially at the higher forcing amplitude a deviation in the shock intensity is visible. Despite the deviations between the URANS and the hybrid ROM solutions, the overall trend of the shock position and the overall spanwise c_p-distribution are captured by the used ROM.

The deviations are also shown by the MSE values of the surface c_p for the timestep $t = 2$, see Figs. 18 and 19. Here moderate deviations in the overall MSE are indicated. In conclusion, the hybrid ROM is able to predict the c_p-distributions to a sufficient extend.

After evaluating the prediction quality of the hybrid ROM the computational efficiency is investigated. All numerical simulations have been carried out with the DLR-TAU solver on the SuperMUC-NG of the Leibniz Supercomputing Center (LRZ). For the calculation 10 nodes with 48 cores each have been applied, resulting in a total number of 480 cores. The training, validation and test computations of the hybrid ROM on the other hand where conducted with one core of an Intel Xeon W-2295 3 GHz processor.

For comparison the needed time for every simulation will be given in CPU hours, which are the wall clock time of a simulation times the number of processors used. For the training of the hybrid ROM a total of 18244 CPU hours was needed. This is composed out of 18240 CPU hours for running the SAPRBS simulations for the generation of the training data and 4 CPU hours for training the hybrid ROM.

The generation of the harmonic motion test data set took 7200 CPU hours each. In contrast, the application of the trained hybrid ROM took around 0.3 CPU hours. So the reduction in computational time for the already trained hybrid ROM is by three orders of magnitude for the considered test cases.

Exceeding a number of three different harmonic excitation signals to be investigated the effort of training the hybrid ROM has already payed off. In an industrial context for an aeroelastic analysis a set of approximately 40 different modes would have to be investigated. Done solely by URANS simulations this would result in an overall computational effort of approximately 1 Mio. CPU hours. If further variations should be investigated like different flow conditions even more computational time is needed. Depending on the number of test cases and performed simulations, using hybrid ROMs like this has a potential to save several orders of magnitude in computational time.

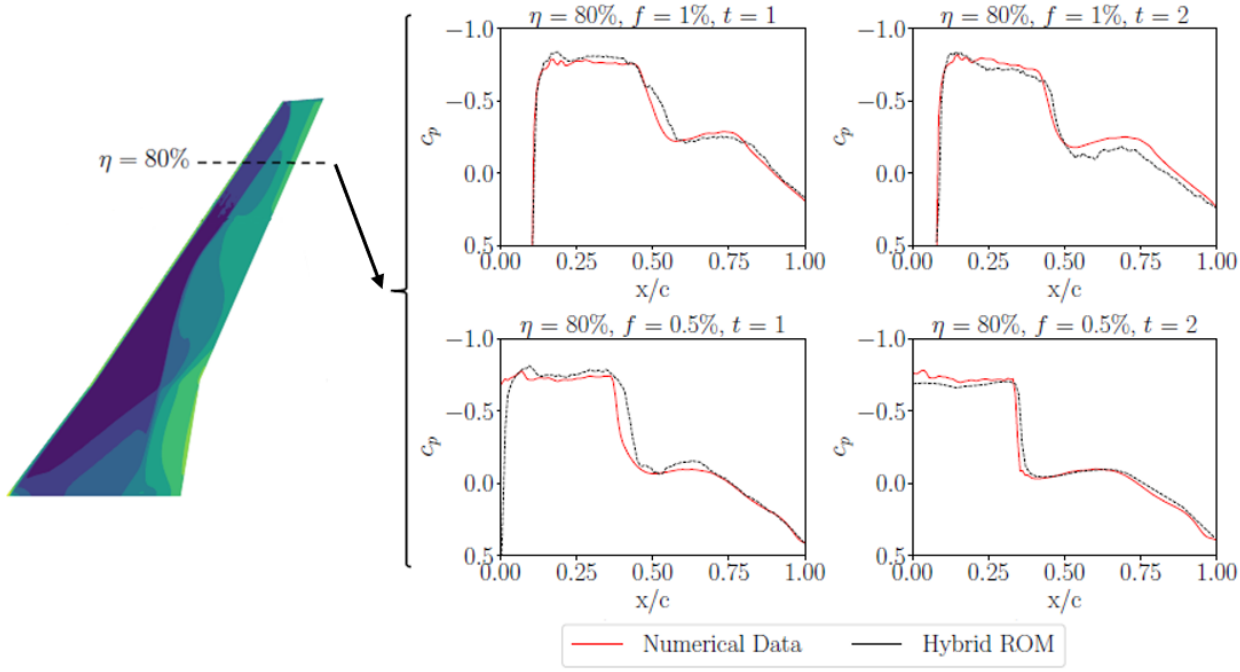

FIG 17. Comparison of original and predicted c_p values at a spanwise positions $\eta = 80\%$ for two timesteps t of the multi-step prediction (harmonic excitation, $f = [1\%, 0.5\%]$).

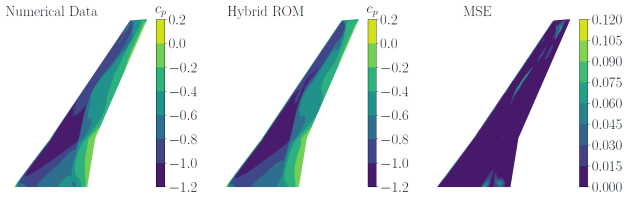

FIG 18. Comparison of a numerical c_p-snapshot (left) and a c_p-snapshot obtained by the trained hybrid ROM (middle) (Mode 11, harmonic excitation, $f = 1\%, t = 2$). The corresponding MSE is shown on the right wing surface.

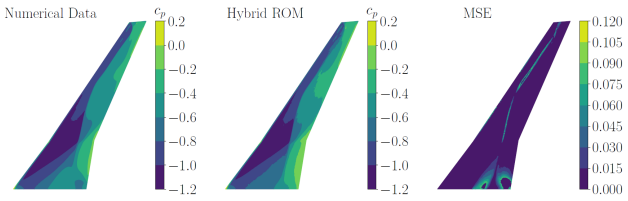

FIG 19. Comparison of a numerical c_p-snapshot (left) and a c_p-snapshot obtained by the trained hybrid ROM (middle) (Mode 11, harmonic excitation, $f = 0.5\%, t = 2$). The corresponding MSE is shown on the right wing surface.

5. CONCLUSION AND OUTLOOK

This study presented a hybrid deep learning model based on a convolutional autoencoder (CNN-AE) and a long short-term memory (LSTM) neural network. This hybrid reduced order model (ROM) has been applied to predict wing buffet pressure distributions under forced vibrations in the structural eigenmodes. By imposing this structural deflections during an already established buffet instability a highly nonlinear flow field has been generated. The NASA Common Research Model (CRM) with the FERMAT structural model has been used as test case. The first symmetric wing bending mode shape (mode 11) has been simulated. The training data for the hybrid ROM was computed via forced-motion URANS simulations with the DLR-TAU solver. A smoothed amplitude-modulated pseudo-random binary signal (SAPRBS) was used for the training data. The performance of the trained hybrid ROM was evaluated by using the ROM to predict buffet pressure distributions due to harmonic excitations of the considered mode shapes. When comparing the buffet loads from the reference numerical solution with the prediction of the hybrid ROM good agreement could be achieved. Therefore, the proposed ROM is able to capture buffet pressure loads due to deflections with a high degree of accuracy. By applying the ROM, computational time and with that computational costs can be reduced by several orders of magnitude compared to the full-order reference solution.
Future work will focus on applying the hybrid ROM architecture to different eigenmodes of the wing. Besides the prediction quality of the hybrid ROM also the extrapolation abilities onto different eigenmodes of the wing should be investigated.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for funding this work in the framework of the research unit FOR 2895 (Unsteady flow and interaction phenomena at high speed stall conditions), subproject TP7, grant number BR1511/14-1; BR1511/14-2. The Gauss Centre of Supercomputing e.V. is gratefully acknowledged for funding this project by providing computing time on the SuperMUC-NG at Leibniz Supercomputing Centre. Further, Kai Eberl is gratefully acknowledged for his support with the implementation of the neural network architecture. In addition, the authors thank the Institute of Aerodynamics and Gas Dynamics of the University of Stuttgart, in particular Maximilian Ehrle, for providing the numerical grid of the CRM.

Contact address:

rebecca.zahn@tum.de

References

[1] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

[2] P. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990. DOI: 10.1109/5.58337.

[3] T. Klimmek, E. Dowell, and S. Ricci. Parametric set-up of a structural model for fermat configuration for aeroelastic and loads analysis. *Journal of Aeroelasticity and Structural Dynamics*, 3(2), 2014.

[4] P. Spalart. Strategies for turbulence modelling and simulations. *International Journal of Heat and Fluid Flow*, 21(3):252–263, 2000. DOI: 10.1016/S0142-727X(00)00007-2.

[5] M. Iovnovich and D. Raveh. Numerical study of shock buffet on three-dimensional wings. *AIAA Journal*, 53(2):449–463, 2015. DOI: 10.2514/1.J053201.

[6] D. Raveh and E. Dowell. Aeroelastic responses of elastically suspended airfoil systems in transonic buffeting flows. *AIAA Journal*, 52(5):926–934, 2014. DOI: 10.2514/1.J052185.

[7] E. Paladini, J. Dandois, D. Sipp, and J.-Ch. Robinet. Analysis and comparison of transonic buffet phenomenon over several three-dimensional wings. *AIAA Journal*, 57(1):379–396, 2019. DOI: 10.2514/1.J056473.

[8] L. Masini, S. Timme, and A. Peace. Analysis of a civil aircraft wing transonic shock buffet experiment. *Journal of Fluid Mechanics*, 884:A1, 2020. DOI: 10.1017/jfm.2019.906.

[9] J. Crouch, A. Garbaruk, and M. Strelets. Global instability in the onset of transonic-wing buffet: Corrigendum. *Journal of Fluid Mechanics*, 901, 2020. DOI: 10.1017/jfm.2020.557.

[10] W. He and S. Timme. Triglobal infinite-wing shock-buffet study. *Journal of Fluid Mechanics*, 925, 2021. DOI: 10.1017/jfm.2021.678.

[11] S. Timme. Global instability of wing shock-buffet onset. *Journal of Fluid Mechanics*, 885:A37, 2020. DOI: 10.1017/jfm.2019.1001.

[12] Y. Ohmichi, T. Ishida, and A. Hashimoto. Numerical investigation of transonic buffet on a three-dimensional wing using incremental mode decomposition. In *AIAA SciTech Forum*, 2017. DOI: 10.2514/6.2017-1436.

[13] D. Raveh and E. Dowell. Frequency lock-in phenomenon for oscillating airfoils in buffeting flows. *Journal of Fluids and Structures*, 27(1):89–104, 2011. DOI: 10.1016/j.jfluidstructs.2010.10.001.

[14] S. Scharnowski, K. Kokmanian, C. Schäfer, T. Baur, A. Accorinti, and C. Kähler. Shock-buffet analysis on a supercritical airfoil with a pitching degree of freedom. *Experiments in Fluids*, 63(6), 2022. DOI: 10.1007/s00348-022-03427-4.

[15] J. Nitzsche, J. Otte, C. Kaiser, and H. Hennings. The effect of shock control bumps on the transonic flutter and buffeting characteristics of a typical wing section. In *International Forum on Aeroelasticity and Structural Dynamics (IFASD) 2022*, 2022.

[16] J. Kou and W. Zhang. Data-driven modeling for unsteady aerodynamics and aeroelasticity. *Progress in Aerospace Sciences*, 125:100725, 2021. DOI: 10.1016/j.paerosci.2021.100725.

[17] M. Ehrle, A. Waldmann, T. Lutz, and E. Krämer. Simulation of transonic buffet with an automated zonal des approach. *CEAS Aeronautical Journal*, 11(4):1025–1036, 2020. DOI: 10.1007/s13272-020-00466-7.

[18] M. Winter. *Nonlinear Aerodynamic Reduced-Order Modeling Using Neuro-Fuzzy Approaches*. Dissertation, Technical University of Munich, 2021. DOI: 10.13140/RG.2.2.29909.09440.

[19] T. Lutz. Going for experimental and numerical unsteady wake analyses combined with wall interference assessment by using the NASA CRM-model in ETW. In *Aerospace Sciences Meetings*, 2013. ISBN: 978-1-62410-181-6. DOI: 10.2514/6.2013-871.

[20] S. Illi, C. Fingskes, T. Lutz, and E. Kraemer. Transonic tail buffet simulations for the common research model. In *31st AIAA Applied Aerodynamics Conference*, 2013. DOI: 10.2514/6.2013-2510.

[21] V. Togiti, B. Eisfeld, and O. Brodersen. Turbulence model study for the flow around the nasa common research model. *Journal of Aircraft*, 51(4):1331–1343, 2014. DOI: 10.2514/1.C032609.

[22] M. Winter and C. Breitsamter. Application of unsteady aerodynamic reduced-order modeling techniques to a complex configuration. In *International Forum on Aeroelasticity and Structural Dynamics*, 2017.

[23] O. Nelles. *Nonlinear system identification: From classical approaches to neural networks and fuzzy models*. Engineering online library. SPRINGER, Berlin and Heidelberg, 2001. ISBN: 9783540673699.

[24] I. Goodfellow, A. Courville, and Y. Bengio. *Deep learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Massachusetts, 2016. ISBN: 0262337371.

[25] V. Rozov and C. Breitsamter. Data-driven prediction of unsteady pressure distributions based on deep learning. *Journal of Fluids and Structures*, 104:103316, 2021. DOI: 10.1016/j.jfluidstructs.2021.103316.

[26] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. DOI: 10.1038/323533a0.

[27] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[28] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies: A field guide to dynamical recurrent neural networks. *A Field Guide to Dynamical Recurrent Neural Networks*, 2003.