Technische Universität München
TUM School of Engineering and Design

Deep Learning Approaches for Transonic Aerodynamic Buffet Analysis

Anne Rebecca Carola Zahn

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung einer

Doktorin der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Ilkay Yavrucuk
Prüfende der Dissertation:

1. apl. Prof. Dr.-Ing. Christian W. M. Breitsamter
2. Prof. Dr. Nicolas Gauger
3. Hon.-Prof. Dr.-Ing. Ralf Rudnik

Die Dissertation wurde am 25.10.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 29.03.2024 angenommen.

Typesetting: LaTeX

Acknowledgments

The present thesis was written during and after my work as a research associate at the chair of aerodynamics and fluid mechanics of the Technical University of Munich. At this point, I would like to thank all who have accompanied me during my time at the chair.

First of all, I would like to express my deep gratitude to my advisor and first examiner Prof. Dr. Christian W. M. Breitsamter for giving my the opportunity to conduct my research in a highly interesting topic as well as for the excellent supervision and continuous support he provided over the years. I really appreciate his always open ear, his support in attending national and international conferences as well as publishing journal papers. Further, I am really thankful for his support concerning any kind of professional development. In addition, I would like to thank Prof. Dr. Nicolas Gauger and Prof. Dr. Ralf Rudnik for their functions as the second and third examiner as well as Prof. Dr. Ilkay Yavrucuk for chairing the examination.

Moreover, I would like to thank all current and former members of the chair of aerodynamics and fluid mechanics for their support and the excellent work environment. Special thanks go to Dr. Julie Piquee, Dr. Maximilian Winter, Dr. Vladislav Rosov, Moritz Zieher, Julius Stegmüller, Dominik Sedlacek and Mauricio Jentys. With their help and valuable discussions they essentially contributed to this research and the working atmosphere. Besides, I would like to express my gratitude to my project partners Dr. Andre Weiner, Maximilian Ehrle, Johannes Kleinert and Dr. Thorsten Lutz for the pleasant project-related cooperation and support. Further, I would like to thank the German Research Foundation (DFG) for providing funding that allowed me to work on that research topic.

Finally, I wholeheartedly thank my family, in particular my parents as well as my grandparents for their support; my friends who also supported me as well as my husband Dr. Florian Heckmeier for his continuous support and love. Without all of them, this research work would hardly have been possible.

Munich, July 2024 Rebecca Zahn

Abstract

During flight conditions in the transonic regime, large amplitude and self-sustained shock oscillations as well as boundary layer separation can appear. This instability, commonly referred to as transonic buffet or shock buffet, defines a limit of the flight envelope of modern passenger aircraft. Due to buffet, passenger comfort during flight is drastically decreased and the wing structure is susceptible to material fatigue. Therefore, the determination of buffet loads on the wing structure is crucial for the design and qualification process of an aircraft. However, although considering state-of-the art numerical methods as well as experimental facilities, the determination of buffet loads is time and cost consuming. Therefore, the use of both numerical and experimental methods for buffet load computation is limited, especially in an industrial context.

In order to present a solution for this issue, model-order reduction methods based on long short-term memory (LSTM) neural networks, convolution neural networks (CNN) as well as autoencoders (AE) are developed and applied in the present thesis in order to allow for an accurate and efficient alternative computation of buffet loads. The trained reduced-order models (ROMs) based on a LSTM are able to predict integral buffet characteristics due to pronounced shock motions, varying freestream conditions as well as forced vibrations. In contrast, CNN and AE-based ROMs enable the determination of buffet pressure distributions due to varying flight conditions and structural vibrations.

The proposed ROM methods are tested and validated by considering a two-dimensional and two three-dimensional test cases. For performance evaluation, the results obtained by the ROMs are compared to either full-order reference computational fluid dynamics (CFD) or experimental solutions.

Zusammenfassung

Während des Fluges im transsonischen Bereich können selbsterregte Stoßschwingungen mit großer Amplitude sowie Ablösungen der Grenzschicht auftreten. Diese Instabilität, die allgemein als transonisches Buffet oder Stoß-Buffet bezeichnet wird, stellt eine Grenze für den Flugbereich moderner Passagierflugzeuge dar. Durch das Auftreten von Buffet wird der Komfort der Passagiere während des Fluges drastisch verringert und die Flügelstruktur ist anfällig für Materialermüdung. Daher ist die Bestimmung der Buffetlasten auf die Flügelstruktur für den Entwurfs- und Qualifikationsprozess eines Flugzeugs von entscheidender Bedeutung. Trotz modernster numerischer Methoden und experimenteller Einrichtungen ist die Bestimmung von Buffetlasten jedoch sehr zeit- und kostenaufwendig. Daher ist der Einsatz von numerischen und experimentellen Methoden zur Berechnung von Buffetlasten begrenzt, insbesondere im industriellen Kontext.

Um eine Lösung für dieses Problem zu präsentieren, werden in der vorliegenden Arbeit Modellordnungsreduktionsverfahren auf der Basis von neuronalen Netzen mit Langzeitgedächtnis (LSTM), neuronalen Faltungsnetzen (CNN) sowie Autocodierern (AE) entwickelt und angewendet, um eine genaue und effiziente alternative Berechnung von Buffetlasten zu ermöglichen. Die trainierten Modelle reduzierter Ordnung (ROMs), die auf einem LSTM basieren, sind in der Lage, integrale Buffet-Charakteristika aufgrund ausgeprägter Stoßbewegungen, variierender Anströmbedingungen sowie erzwungener Vibrationen vorherzusagen. Im Gegensatz dazu ermöglichen CNN- und AE-basierte ROMs die Bestimmung von Buffetdruckverteilungen aufgrund variierender Anströmbedingungen und Strukturvibrationen.

Die vorgeschlagenen ROM-Methoden werden anhand eines zweidimensionalen und zweier dreidimensionaler Testfälle getestet und validiert. Zur Leistungsbewertung werden die mit den ROMs erzielten Ergebnisse entweder mit den Ergebnissen der numerischen Strömungsmechanik (CFD) oder mit experimentellen Lösungen verglichen.

Contents

List of Figures

List of Tables

Nomenclature

Roman Symbols	Denotation
A	amplitude
A_{err}	amplitude error
AR	aspect ratio
A_{ref}	reference area
a	activation function
b	wing span
$\boldsymbol{b}$	bias vector
$\boldsymbol{c}$	cell state vector of LSTM cell
C	magnitude squared coherence
C	Sutherland‘s constant
C_L	lift coefficient
C_{err}	comprehensive error
$\boldsymbol{C}_{in}$	input channels
$\boldsymbol{C}_{out}$	output channels
C_{My}	pitching moment coefficient
c_p	pressure coefficient
$\bar{c}_p$	mean of pressure coefficient
c_{ref}	mean aerodynamic chord
E	error function
f_s	sampling frequency
f	scaling factor
f_t	trip term of spalart-allmaras turbulence model
f_{t2}	suppression term of spalart-allmaras turbulence model
$\boldsymbol{f}$	forget gate vector of LSTM cell
$\boldsymbol{h}$	hidden state vector
$\boldsymbol{h}$	bottleneck vector
h	heave displacement degree of freedom
H	height of input matrix
H_k	height of kernel
$\boldsymbol{h}$	hidden state vector of NN
$\boldsymbol{i}$	input gate vector of LSTM cell
k	discrete time step
k_{red}	reduced frequency
$k_{red,Buffet}$	buffet reduced frequency
$k_{red,Ex}$	excitation reduced frequency
L	loss function
$L_{1,2}$	regularization type 1 & 2
Ma_∞	freestream Mach number

N_S	number of samples
p	static pressure
P_{err}	phase error
p_{drop}	dropout factor
$\boldsymbol{o}$	output gate vector of LSTM cell
q	dynamic pressure
Q	fit factor
q/E	dynamic pressure level
R	gas constant
R_λ	regularization term
Re	Reynolds number
s	stride parameter
S	power spectral density estimates
SF	scaling factor
Sr	Strouhal number
t	physical time step
T	temperature
T_{Buffet}	buffet period
U_∞	freestream velocity
U_c	convection velocity
W	width
W_k	width of kernel
$\boldsymbol{W}$	weight matrix
$\boldsymbol{x}$	model input vector
x/c	relative thickness (airfoil)
$\boldsymbol{y}$	model output vector
$\hat{\boldsymbol{y}}$	system output vector
y^+	nondimensional wall distance

Greek Symbols	**Denotation**
α	angle of attack
$\beta_{1,2}$	hyperparameter of ADAM algorithm
β	momentum hyperparameter
β	batch normalization hyperparameter
δ	partial derivative
ε	hyperparameter of ADAM algorithm
ε	batch normalization constant
ε	reparameterization hyperparameter
η	learning rate
η	spanwise position
γ	batch normalization hyperparameter

γ_f	fading factor
θ	pitch angle degree of freedom
λ	wing taper ratio
λ	regularization term hyperparameter
μ	variance
ρ	density
σ	sigmoid function
σ	standard deviation
τ	nondimensional time
ϕ	wing sweep angle
ϕ	phase angle
ω	angular frequency
ω_{Buffet}	angular buffet frequency

Subscripts	**Denotation**
∞	freestream value
in	input
out	output
ref	reference

Abbreviations and Acronyms

ADAM	adaptive moment estimation
Adagrad	adaptive gradient
AE	autoencoder
ANN	artificial neural network
APRBS	amplitude-modulated pseudo-random binary signal
ARMA	auto-regressive with moving average
ARX	auto-regressive with exogenous input
AUSM	advective upstream splitting method
BN	batch normalization
BP	back propagation
BPTT	back propagation through time
CFD	computational fluid dynamics
CFL	courant friedrichs levy
CNN	convolutional neural network
ConvLSTM	convolutional long short-term memory
CRM	common research model
DDES	dettached delayed eddy simulation
DES	dettached eddy simulation
DLR	German Aerospace Center (Deutsches Zentrum für Luft-und Raumfahrt)
DL	deep learning
DMD	dynamic mode decomposition
DNS	direct numerical simulation
DoF	degree of freedom
DTS	dual time stepping
ETW	European Transonic Windtunnel
FC	fully connected
FEM	finite element model
FERMAT	flutter reduced-order model assessment
FFNN	feed-forward neural network
FFT	fast-Fourier transformation
GAN	generative adversarial network
GVT	ground vibration test
HTP	horizontal tail plane
LCO	limit-cycle oscillation
LE	leading edge
LES	large eddy simulation
LN	layer normalization
LRZ	Leibniz Supercomputing Center (Leibniz Rechenzentrum)

LSTM	long short-term memory
LU-SGS	lower-upper symmetric Gauss-Seidel
MAC	mean aerodynamic chord
ML	machine learning
MLP	multilayer perceptron
MSE	mean squared error
MTOW	maximum take-off weight
MUSCL	monotonic upstream-centered schemes for conservation laws
MZFW	maximum zero-fuel weight
NACA	National Advisory Committee for Aeronautics
NARMAX	nonlinear auto-regressive moving average with exogenous input
NARX	nonlinear auto-regressive with exogenous input
NASA	National Aeronautics and Space Administration
NN	neural network
NFM	neuro-fuzzy model
POD	proper orthogonal decomposition
PSD	power spectral density
PSP	pressure sensitive paint
QCR	quadratic constitutive relation
ReLU	rectified linear unit
RBF	radial basis function
RBFNN	radial basis function neural network
RME	relative magnitude error
RMSE	root mean squared error
RMSprop	root mean square propagation
RNN	recurrent neural network
ROM	reduced-order model
SAPRBS	smoothed amplitude-modulated pseudo-random binary signal
SGD	stochastic gradient descent
SST	shear-stress transport
TAU	triangular adaptive upwind
TCN	temporal convolutional network
TE	trailing edge
TUM	Technical University of Munich
UHBR	ultra high bypass ratio
URANS	unsteady Reynolds-averaged Navier-Stokes
VAR-AE	variational autoencoder
VTP	vertical tail plane
ZDES	zonal dettached eddy simulation

1 Introduction

In order to enable the design and certification process of a civil aircraft, an efficient and accurate determination of the flight envelope boundaries is mandatory. Besides engine-related restrictions, the flight envelope is limited by the appearance of unsteady phenomena, which result from the interaction of aerodynamic, elastic and inertia forces. The coupling between these three types of forces is defined by the well-known triangle of forces proposed by Collar [14], as shown in Figure 1.1.

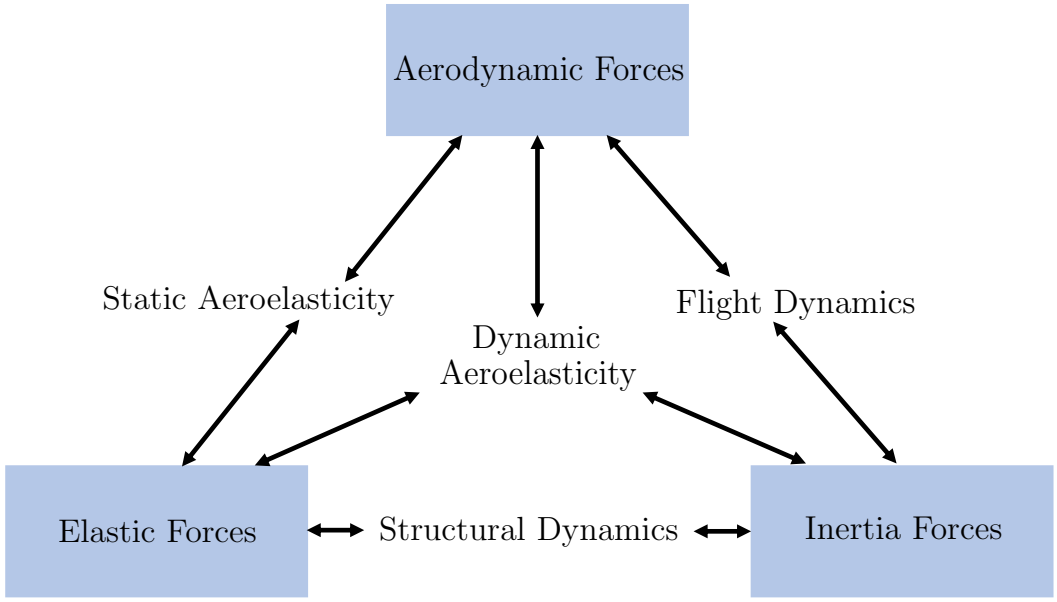


Figure 1.1: Collar's triangle of forces, adapted from [14].

The interaction of aerodynamic and inertia forces leads to rigid body motions, which are summarized by equation of motions related to flight dynamics. The interaction between inertia and elastic forces results in vibrations, which are covered by the discipline of structural dynamics. Static aeroelasticity is defined by the coupling between aerodynamic and elastic forces, whereas the discipline of dynamic aeroelasticity accounts for all three types of forces.

In general, unsteady phenomena resulting from dynamic aeroelasticity can be categorized in stability - and response problems. The stability problem is represented by flutter, whereas response problems are associated to limit-cycle oscillations (LCO), control surface buzz and aerodynamic buffeting. Here, buffeting is defined as the interaction of the wing or tail structure of the aircraft with surface pressure and fluctuations, which result from an instability referred to as buffet. The transonic buffet phenomenon is defined by the interaction between intermittently separated shear layers and shock-waves, which result in large amplitude, self-sustained shock oscillations. Negative effects of this self-sustained unsteadiness range from reduced aerodynamic performance to structural failure due to material fatigue.

1.1 Motivation

Within the framework of this thesis the focus is put on exactly this phenomenon - transonic buffet. In the following, the motivation is given and the research objectives that arise are formulated. Furthermore, an outline of the thesis is presented.

The buffet phenomenon can be differentiated into two types, depending on the flight speed of the aircraft. In the low speed regime, local flow separation due to high lift results in low-speed buffet. The resulting turbulent flow interacts with the wing and/or the tail surface, leading to increased accelerations at the wing and/or the horizontal tail plane (HTP). In contrast, considering transonic flow conditions, the buffet phenomenon occurs due to shock-boundary layer interaction on the upper surface of the wing. In this thesis, the focus is put on the characterization of transonic buffet, which is also commonly referred to as shock buffet.

The unsteady flow characteristics associated to transonic buffet can be investigated by high-fidelity numerical methods such as unsteady Reynolds averaged Navier-Stokes (URANS) simulation, detached-eddy simulations (DES) or large-eddy simulations (LES). In particular, the formation and motion of the shock, as well as the development of the boundary layer and its separation must be adequately captured by the applied numerical solver. However, applying state-of-the-art computational fluid dynamics (CFD) solutions for an aeroelastic buffeting analysis increases computational time and costs drastically. Even with the nowadays available computing capacities, the application of higher fidelity methods such as DES and LES is a challenging task, especially when considering multiple variations of flow conditions.

In addition to the application of numerical methods, experimental investigations are a common tool for the characterization of transonic buffet. However, similar to numerical investigations, the experimental determination is time and cost consuming, due to the high requirements of the aircraft model to be tested, the test equipment as well as the operation of the wind tunnel in order to achieve atmospheric settings and flow similarities (e.g. Reynolds number).

A faster and less expensive alternative to the application of the aforementioned high-fidelity solutions and experimental investigations is given by system identification methods, which are also referred to as reduced-order models (ROM). These models have become increasingly popular in recent years, since they allow for the representation of a given aerodynamic system, which is defined by a certain number of inputs and outputs. Considering the representation of unsteady flow features associated to transonic buffet, the input is commonly defined by external or body related motion of the investigated airfoil or wing. The output is represented by the corresponding integral or local aerodynamic loads like aerodynamic coefficients or pressure distributions. A ROM can either be conditioned by means of numerical and experimental data or by a combination of both data types. If the ROM is accurately trained, it is applicable for the computation

of buffet flow features at flow conditions which are not necessarily included in the training data set. Further, the prediction can be achieved within a fraction of computational time and cost compared to extensive numerical and experimental investigations. Therefore, the application of a ROM for transonic buffet analysis potentially enables a faster and less expensive investigation, especially if various flow parameters are considered.

1.2 Thesis Outline and Research Objectives

In order to enable a fast and cost efficient determination of transonic buffet loads in the earlier stages of the aircraft development process, model-order reduction frameworks based on recurrent neural networks (RNN), convolutional neural networks (CNN) and autoencoders (AE) are developed and applied in this thesis. Therefore, the main objective is the development of fast and accurate deep learning frameworks for the prediction of two - and three-dimensional integral and local buffet characteristics. In particular, the frameworks should be able to model buffet characteristics due to freestream parameter variations, rigid body motions as well as eigenmode-based structural deformations. For the training of the ROM approaches, both full-order CFD solutions or experimental data sets are applied, in order to enable the ROMs to learn the underlying flow physics as accurate as possible. By applying the trained ROMs, a reduction in computational time by several orders of magnitude is pursued, which enables the implementation of the developed methods in early stages of the aircraft design process. Therefore, the following questions can be defined as the overall research objectives approached in this work:

- **Research Objective 1**: To what degree is it possible to predict unsteady integral and local buffet loads due to freestream parameter variations?
- **Research Objective 2**: To what degree is it possible to predict unsteady integral and local buffet loads due to rigid body motions and eigenmode-based deflections?
- **Research Objective 3**: Is it possible to train and apply a ROM by means of experimental data, which are characterized by a high noise content?
- **Research Objective 4**: How much reduction in computational time can be achieved by the application of a trained ROM compared to a full-order CFD or experimental solution?

For demonstration purposes, several well-established test cases are considered, including a basic airfoil geometry (NACA0012) and two transport-aircraft-type configurations. In particular, the Airbus XRF-1 and the NASA Common Research Model (CRM) configuration are considered here. In order to evaluate the performance of the developed and trained ROMs, both the prediction accuracy as well as the computational efficiency compared to full-order CFD simulations and experimental investigations, are applied.

The following thesis is divided into seven chapters. A schematic overview of the research activities is given in Figure 1.2, highlighting the connections between the chapters.

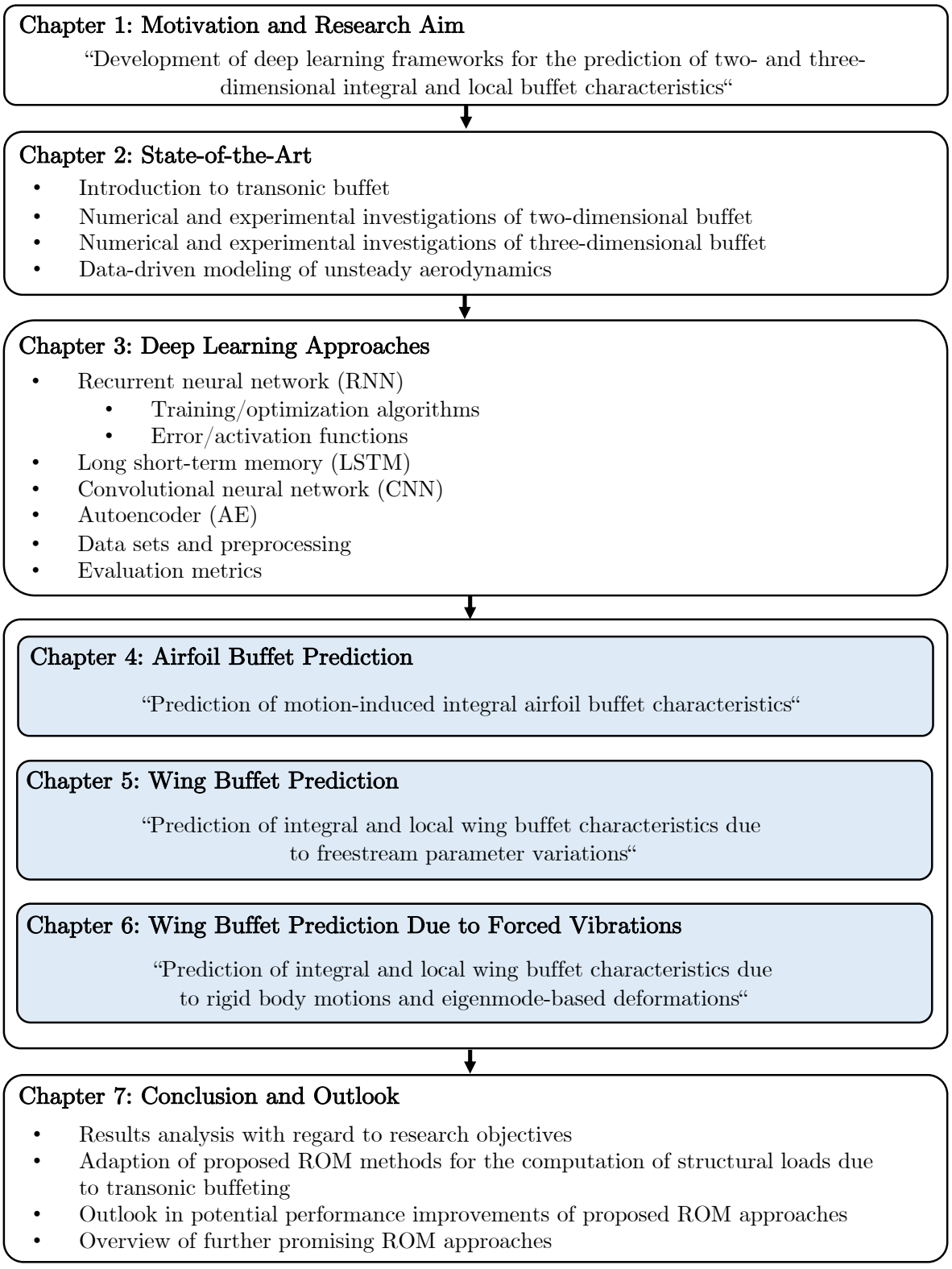


Figure 1.2: Overview of the chapters of this thesis.

In **Chapter 1**, the motivation and outline of the thesis are described. Further, the four main research objectives are introduced, which will be considered in Chapter 4 to Chapter 6.

Chapter 2 gives an overview of the state-of-the-art related to the aim of the thesis. In the first part, the focus is put on a general definition of the transonic buffet instability. Following the general definition, a detailed literature review on both numerical and experimental studies on buffet on airfoils and wings is presented. The third section of the chapter covers an introduction to data-driven modeling, with special focus on unsteady aerodynamic modeling. The chapter is concluded with a brief literature review on the application of data-driven modeling methods for the prediction of transonic buffet aerodynamics.

In **Chapter 3**, the applied data-driven modeling methods, in the following referred to as deep learning approaches, are classified and introduced. In the first section of this chapter, a basic introduction to deep learning is provided. In the second section, the architecture of a RNN is explained. Referring to the RNN architecture, the training algorithm applied in this thesis is outlined. Besides a detailed description of the training algorithm, a comprehensive guide for the selection of error and activation functions as well as hyperparameters is given. Further, applied optimization algorithms and normalization techniques are introduced. In the third part of chapter 3, the architecture of the long short-term memory (LSTM) neural network is described. In the fourth and fifth section, the working principles of CNN and AE are outlined, respectively. In the sixth section, different data sets applied for the training and performance evaluation of a neural network are briefly introduced and differences between the data sets are highlighted. Within the last part of this chapter, different metrics for ROM performance evaluation applied in the present thesis are introduced and explained.

Chapter 4 covers the prediction of airfoil buffet characteristics. The training and application process of a LSTM-based ROM for the computation of lift and pitching moment coefficient trends on the NACA0012 airfoil are outlined. Further, an accurate reproduction of the two-dimensional buffet characteristic lock-in phenomenon is pursued. The prediction accuracy as well as the computational speed-up of the proposed ROM are compared to full-order CFD simulation results.

In **Chapter 5**, the focus lies on an accurate and efficient computation of both integral and local wing buffet characteristics due to freestream parameter variations. Therefore, two different ROM approaches are developed and applied. For training and evaluation purposes, experimental data obtained during a wind tunnel test campaign are used. The wind tunnel model represents the Airbus XRF-1 configuration. Although the experimental data are characterized by a high noise content, an accurate ROM training and application are intended.

In **Chapter 6**, two different ROM approaches for the computation of integral and local wing buffet characteristics due to rigid body motions and eigenmode-based structural deformations are introduced. The ROMs are conditioned by means of high-fidelity numerical data sets. As a test case, the NASA CRM configuration in combination with the FERMAT structural model is chosen.

In **Chapter 7**, the performed studies and their results are analyzed and summarized with regard to the research objectives. Conclusions on the applied ROM methods are given and major outcomes are highlighted and summarized. Further, the chapter includes a discussion of potential performance improvements of the proposed ROM methods as well as the applicability for coupling with a structural solver in order to enable the computation of structural loads due to transonic buffeting. Last, an overview of further promising deep learning approaches for buffet and buffeting analysis is provided.

2 State-of-the-Art

In the following chapter, an introduction to the transonic buffet phenomenon is given. Following the introduction, a review of both numerical and experimental studies of buffet is presented. Here, differences between buffet on airfoils and wing buffet are clearly highlighted. The chapter is concluded with a general definition of data-driven modeling, followed by a literature review of data-driven methods applied for the prediction of unsteady aerodynamic characteristics.

2.1 Introduction to Transonic Buffet

The transonic buffet phenomenon, which is also commonly referred to as shock buffet, was first observed by Hilton and Fowler [48] in 1947. Examining the flow physics of buffet, two distinct types of buffet are identified, defined as Type 1 and Type 2. Type 1 buffet is characterized by shock oscillations on both the lower and upper surface of the airfoil, typically if the airfoil has zero incidence. In contrast, Type 2 buffet occurs at higher angles of attack and is associated with self-sustained cycles of shock movement and flow separation only on the upper airfoil surface. Further, the freestream Mach number as well as the geometry of the airfoil have an influence on the origin of Type 2 buffet. Within the scope of this thesis, the focus will be on Type 2 buffet.

In order to establish a better understanding of the governing physics of the buffet Type 2 instability, several models have been proposed. Lee [67, 68] introduced an acoustic wave-propagation feedback model in order to describe the oscillating shock motion. In Lee's model, pressure waves, which are generated by the moving shock wave, travel down - and upstream and generate a buffet cycle by interacting with each other [53]. Later numerical studies [25, 145] showed good agreement with the assumptions made by Lee [67], however, some studies [36, 55] reported considerable deviations in the buffet characteristics, which led to a suggested revision of Lee's model by Jacquin et al. [55]. The application of the new developed wave-propagation model showed better agreements with the experiments, however, differences in the buffet characteristics still remained [40, 53].

In addition to the study of Lee [67,68], Crouch et al. [16,17] proposed a model based on global mode decomposition. The results indicated that buffet onset is linked to a global mode instability. In contrast to the mechanism proposed by Lee [67, 68], the pressure waves travel around the entire airfoil, instead of oscillating only on the upper airfoil surface. Based on numerical studies by Crouch et al. [17], good agreement with experimental studies [85] was indicated. Following the global mode instability analysis, further numerical studies revealed similar results [62, 112] on Type 2 buffet.

Following the studies mentioned above, Raghunathan et al. [98] and Iovnovich and Raveh [53] revealed flowfield studies of a single buffet cycle on various airfoils. Due to their findings, the buffet phenomenon was described as an unstable interaction of the

shock-wave and flow separation. The buffet cycle can be classified as a transonic pre-stall, self-sustained instability. In Figure 2.1, a shock buffet cycle on the NACA0012 airfoil, divided by four time steps, is visualized. Here, the buffet instability is computed using an URANS approach in combination with the Spalart-Allmaras (SA) turbulence model. For the SA model, the Edwards modification [29] has been activated.

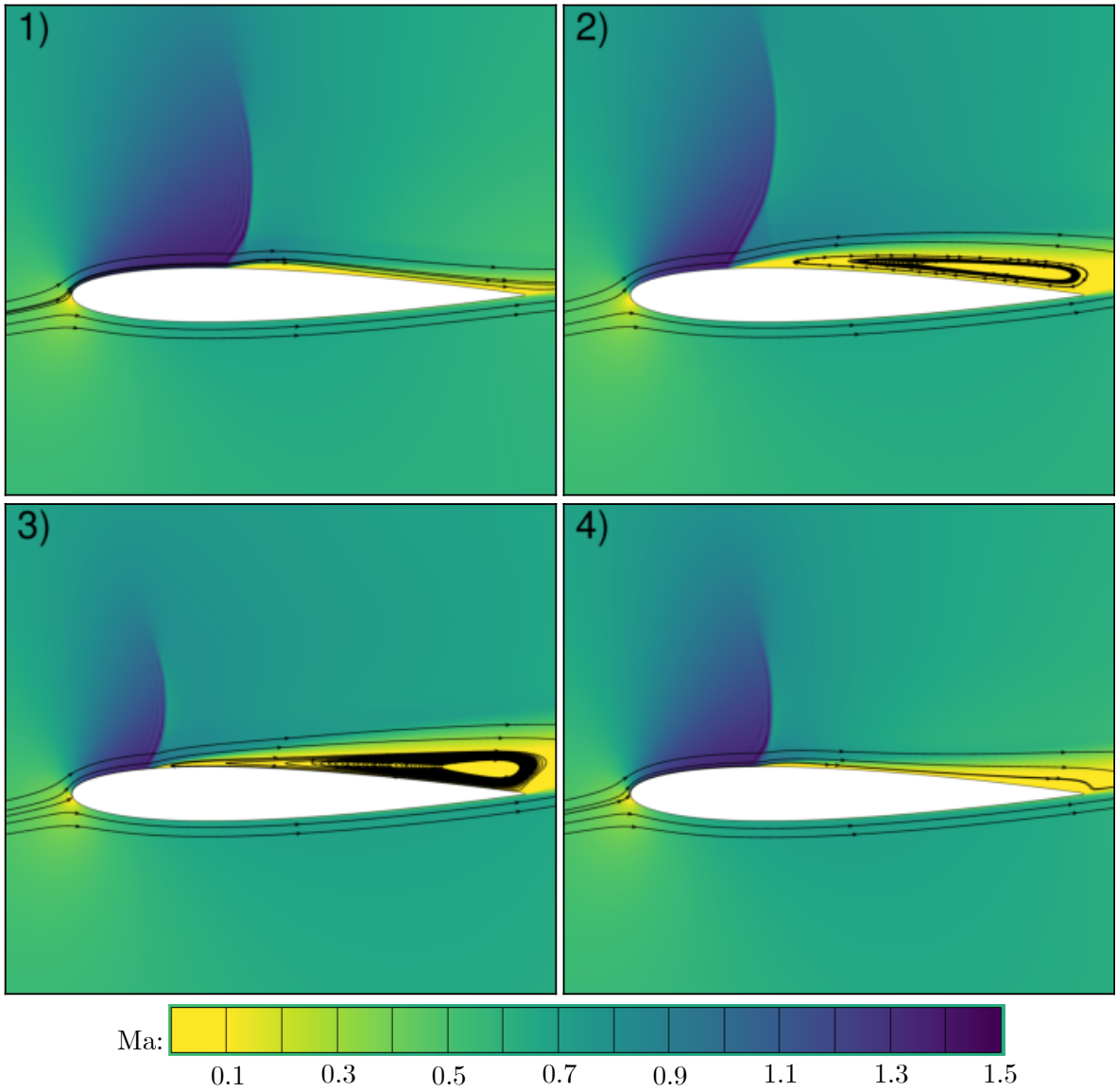


Figure 2.1: Mach number and flow streamline plots of the buffet cycle of the NACA0012 airfoil ($Ma_\infty = 0.72$, Re $= 10^7$, $\alpha = 6°$).

Based on the investigations of the characteristics of the buffet cycle on the NACA0012 airfoil by Iovnovich and Raveh [53], the following stages of the buffet cycle can be summarized: The onset of buffet is typically linked with the formation of a shock at the position of maximum curvature on the suction side of the airfoil (see Figure 2.1 (1)). Downstream of the shock, a flow separation bubble develops, which interacts with the shock. Due to the pressure increase in the area of separated flow, the shock travels upstream, which leads to an increase in shock strength and separated shear layer (see

Figure 2.1 (2)). As the shock weakens, the separated boundary layer behind the shock starts to reattach, while the flow at the trailing edge (TE) is still characterized by large separated areas (see Figure 2.1 (3)) [53]. Due to this large separation area at the TE, the circulation around the airfoil is decreased. As soon as the separated flow starts to reattach, the shock travels back downstream and strengthens (see Figure 2.1 (4)) [53]. As the shock reaches its initial position, the separation bubble behind the shock initiates a new buffet cycle [53].

Due to the repeating cycle of shock movement and flow separation at buffet condition, the corresponding aerodynamic coefficients are characterized by a periodic behavior. Examining the resulting time-series of the aerodynamic coefficients, the buffet oscillations can be characterized by a characteristic frequency, which is referred to as the buffet frequency ω_{Buffet}. In the context of a buffet analysis, the buffet frequency is commonly expressed as a nondimensional, reduced frequency $k_{red,Buffet}$, which is defined as follows:

$$k_{red,Buffet} = \frac{\omega_{Buffet} \cdot c_{ref}}{U_\infty} \tag{2.1}$$

with U_∞ denoting the freestream velocity. C_{ref} is defined as the geometric reference length, which is defined as the mean aerodynamic chord (MAC) for wing buffet investigations or the root chord for investigations on airfoils. Instead of the reduced frequency, another key figure, namely the Strouhal number Sr (see Equation 2.2), is commonly applied for the characterization of the buffet flow physics.

$$Sr = \frac{\omega_{Buffet} \cdot c_{ref}}{U_\infty} \tag{2.2}$$

Typically, the shock-buffet reduced frequency is comparable to low-frequency elastic modes of transport aircraft [40, 53].

2.2 Two-Dimensional Buffet

In the following section, an overview of numerical and experimental studies of buffet on airfoils, is given. Further, a phenomenon referred to as lock-in is addressed, which is important for understanding the influence of external motion on the buffet instability.

2.2.1 Numerical Investigations

In order to capture the unsteady flow features associated to transonic buffet on airfoils, numerical computations including scale-resolving mechanisms are required. Although due to the averaging process of URANS methods only large turbulence scales are resolved, several studies showed that by applying URANS simulations the fundamental buffet flow physics can be represented with a sufficient degree of accuracy [29, 53]. In addition to the application of URANS simulations, scale-resolving methods have also been used for the computation of transonic buffet. Deck [25] investigated the influence

of DES and LES compared to URANS on transonic buffet on the OAT15A airfoil. Further, Deck proposed a novel simulation approach based on the combination of URANS and LES, defined as Zonal Detached-Eddy Simulation (ZDES). By applying ZDES, a more precise prediction of pressure fluctuations and general flow field characteristics was indicated.

Although numerical methods with higher fidelity are available, most of the numerical studies on transonic buffet apply an URANS solver for buffet simulation. However, URANS simulations have shown to be quite sensitive to various simulation parameters, in particular the applied turbulence model, the discretization as well as the numerical scheme [40]. A study by Barakos and Drikakis [4] assessed the influence of several turbulence models, in particular eddy-viscosity models, on the computation of transonic buffet on airfoils. Their study revealed that the application of the Spalart-Allmaras (SA) [120] turbulence model gained the best results in terms of replicating shock unsteadiness and buffet frequency prediction. Further, the Shear-Stress Transport (SST) k-ω turbulence model proposed by Menter [86] also indicated good correlations with experiments. A later study by Goncalves and Houdeville [41] underlined the findings of Barakos and Drikakis [4], indicating good agreement between the SA and Menter SST model and experimental investigations.

Besides the influence of the selected turbulence model, several studies evaluated the influence of the numerical discretization scheme for the convective fluxes [40]. Goncalves and Houdeville [41] revealed that the upwind Roe [104] with Monotone Upstream-Centered Scheme for Conservation Laws (MUSCL) [69] scheme and the Jameson scheme outperform the Advective Upstream Splitting Method (AUSM+) [72] with MUSCL extrapolation scheme concerning the reproduction of characteristic buffet features. A study by Soda and Verdon [119] underlined the theory of Goncalves and Houdeville [41]. However, compared to the influence of turbulence closure, the effects of the applied convective schemes are smaller, with variations in buffet frequency and amplitude of less than 10% [40]. However, as shown by several studies, with an appropriate choice of turbulence model, the buffet instability is reproducable with the Roe, Jamson as well as the AUSM scheme [40].

In addition to the choice of the numerical discretization scheme, special attention has been given to the assessment of the spatial and temporal discretization. Studies by Rouzaud et al. [107] and Rumsey et al. [109] compared an explicit temporal formulation and an implicit Dual Time Stepping (DTS) scheme in terms of computational efficiency and accuracy. Based on those studies, the DTS method indicated better agreement with experiments, while at the same time using less computational resources [40]. Therefore, the implicit formulation has been employed in various numerical studies [41, 62].

2.2.2 Dynamic Interaction Phenomena at Buffet Condition

Besides the buffet instability, transonic aeroelastic response phenomena, such as LCO or single degree-of-freedom (DoF) flutter, are often characterized by the presence of shock-boundary layer interaction [102]. Since shock buffet frequencies are typically in the order of elastic structural frequencies, several studies suggested that they may contribute to the development of LCO on airfoils.

In order to gain a better understanding of the nature of shock oscillations in dynamic systems, a number of experimental and numerical studies investigated the influence of external motion on shock oscillations developed during transonic buffet. Experimental studies by Davis and Malcolm [23] and Despre et al. [26] evaluated the resonance of a NACA64A010 airfoil, excited with pitching motions at varying frequencies. Based on their studies it was shown that the maximum amplitude of the shock-wave oscillations occurred at excitation frequencies close to the buffet frequency [40].

A numerical study by Nitzsche [90] revealed a commonality between shock buffet and a single DoF oscillator. The resonance of shock buffet was shown to be independent of the excitation mode, however, the buffet frequency was encountered when the shock motion changed to a regular mode [40]. Raveh [101] investigated the influence of external harmonic plunge excitations on shock buffet on the NACA0012 airfoil, revealing results that are consistent with findings obtained by Nitzsche [90]. Examining the resulting aerodynamic responses, a phenomenon commonly referred to as lock-in effect, was identified. Here, in the presence of certain excitation frequencies and amplitudes, the buffet flow response synchronizes with the motion of the airfoil. In addition, Raveh [101] suggested that the occurrence of lock-in may encourage LCO. Buffet shock oscillations may excite a damped structural mode near the stability boundary, resulting in increasing amplitude oscillations [40].

Further numerical studies by Raveh and Dowell [102] aimed at analyzing the interaction of pre-buffet response and external motion as well as the influence of lock-in on the flow field around the NACA0012 airfoil [40]. Increasing the angle of attack towards buffet onset yielded an increase in the resonance frequency and decrease in damping, whereas at buffet onset zero damping occurred. At a developed buffet condition, pressure fluctuations at the TE are present, resulting in vortex shedding into the wake region [40]. Further, a dependency between the resulting aerodynamic responses and the level of the excitation frequency was observed. Excitation frequencies higher than the buffet frequency lead to harmonic aerodynamic responses, whereas lower frequencies result in large fluctuations in the responses due to large-scale shock motion. These findings are in good agreement with a more recent study of the OAT15A airfoil by Giannelis and Vio [38]. Continuing investigations on the lock-in phenomenon, Iovnovich and Raveh [53] revealed that pitch excitations result in developed buffet flow and significant

flow fluctuations, supporting earlier findings by Raveh [101]. However, although the lock-in phenomenon has mainly be investigated using numerical methods, an experimental study by Hartmann et al. [44] also indicated lock-in [40].

Besides a general characterization of the lock-in phenomenon, recent studies dealt with the influence of structural parameters, such as mass ratio and structural damping, on the lock-in mechanism [39, 97]. In addition, several studies also found some degree of commonality between shock buffet and other aeroelastic phenomena, such as single DoF flutter [11, 154]. However, within the scope of this thesis, the results of the studies mentioned above are not further discussed.

2.2.3 Experimental Investigations

Although the majority of studies on buffet characterization is based on numerical modeling, a number of comprehensive experimental studies are available in the literature to gain further insight of the underlying buffet flow physics. Early studies by McDevitt et al. [84], Mabey et al. [78] and Finke [33] examined the buffet instability using different airfoil geometries. Further, an extensive wind tunnel test campaign has been undertaken by ONERA, resulting in a fundamental experimental database of the supercritical OAT15A airfoil [21]. The results of the test campaign are presented by Jacquin et al. [55]. Based on a spectral analysis of the unsteady pressure measurements, a dominant frequency was extracted, which supports the global mode instability theory proposed by Crouch et al. [16,17]. In contrast to the results of the spectral analysis, oil flow visualisation indicated the presence of additional three-dimensional structures, which lead to the suggestion of a superposition of both two- and three-dimensional modes [40]. Similar to the wave-propagation model proposed by Lee [67], Jacquin et al. [55] developed a modified version, assuming different disturbance convection timescales. Based on their results, a more precise representation of buffet flow characteristics using the updated propagation model, was indicated. However, the authors concluded that the description of the buffet instability using a single model is a challenging task. Besides the study of Jacquin et al. [55], Hartmann et al. [44] presented another modified propagation model, defining the buffet shock oscillations as an acoustic phenomenon.

2.3 Three-Dimensional Buffet

Considering the studies of transonic buffet on airfoils outlined in the previous section, the buffet instability is essentially a two-dimensional phenomenon. However, although finding some degree of commonality, numerical and experimental investigation of buffet flow on three-dimensional wings yielded flow characteristics that are different compared to two-dimensional buffet [40]. In the following section, an overview of numerical and experimental investigations on the three-dimensional buffet instability, is given.

2.3.1 Numerical Investigations

Due to recent advances in computational resources, three-dimensional effects of the buffet instability have been extensively investigated by URANS or scale-resolving methods. Early studies by Brunet and Deck [10] and Sator and Timme [113, 114] revealed that, similar to airfoil buffet investigation, the application of URANS, DES, DDES and ZDES enables an accurate reproduction of the characteristic shock motion. However, results of DES and DDES computations showed a better agreement with experimental studies by Lawson et al. [64] in terms of mean and root mean squared (RMS) pressure distribution levels as well as the position of the shock. Further, their studies revealed that only using a RANS approach results in lower separation levels, which leads to the assumption that turbulent scales must be accurately resolved in order to model the three-dimensional buffet instability [40].

Based on their studies it was shown that the buffet shock unsteadiness starts to form at the wingtip, moving in spanwise direction towards the wing root with increasing incidence. In addition, with increasing incidence the frequency developed more broadband, while at the same time the dominant buffet frequency reduces [40]. This result highlights a fundamental difference between two- and three-dimensional buffet, since the shock oscillations on airfoils are typically characterized by a periodic behavior representing a single, characteristic frequency. Examining the resulting time histories and spectra of the aerodynamic coefficients, oscillations with low-frequency peaks between 150 and 300 Hz are indicated, which are consistent with experimental studies by Lawson et al. [64] and Koike et al. [59]. With special focus on turbulence modeling, the work by Sator and Timme [113] additionally revealed that both the k-ω Shear-Stress Transport (SST) model and the SA model produced results which are most consistent with experimental studies.

In order to investigate distinct differences between the two-dimensional and three-dimensional buffet instability mechanism, Iovnovich and Raveh [54] conducted a numerical URANS simulation-based study using three different wing geometries. In particular, an infinite-straight, an infinite-swept and a finite-swept wing with constant airfoil section, were considered. Analysis of the infinite-straight wing model yielded shock motion mainly in chordwise direction, with only small disturbances in spanwise direction [40]. Further, the shock motion amplitude and frequency was found to be consistent with two-dimensional buffet simulations. However, with increasing sweep angle ($\lambda > 20°$), an increase in pressure disturbance propagation in spanwise direction from wing root to wing tip, was observed. This spanwise pressure convection is clearly distinctable from the two-dimensional buffet mechanism and is termed as a convection of so-called buffet cells. The spanwise propagation of these buffet cells is linked to lower amplitude shock motions compared to the two-dimensional buffet instability, however, the corresponding oscillation frequencies were found to be higher and developed more broadband, with Sr numbers ranging from 0.2 to 0.7 [20].

Besides a physical description as provided by Iovnovich and Raveh [54], modal decomposition of the three-dimensional buffet instability has been in focus in recent years. A study by Timme and Thormann [128] focused on global mode stability analysis at pre-buffet conditions, considering frequency sweeps at four angles of attack. By applying a low-frequency torsional mode excitation, a distinct peak in the low-frequency range at $Sr = 0.11$ was observed, which potentially indicates a destabilising aerodynamic mode that was also observed in two-dimensional stability analysis of airfoils [16, 17]. Besides this two-dimensional frequency characteristic, peaks at higher frequency levels occurred, with Sr ranging from 0.3 to 0.7. These additional peaks were assumed to reflect the presence of further unstable aerodynamic modes. Another mode-based buffet model was proposed by Crouch et al. [18], including a global stability analysis of swept and unswept wings [30]. For the numerical simulations, an URANS approach was applied. Consistent with findings of Timme and Thormann [128], higher frequency characteristics were observed, which was associated with the occurrence of a global flow instability.

Later studies by Ohmichi et al. [92] and Timme [127] investigated the buffet instability mechanism by means of Proper Orthogonal Decomposition (POD) and Dynamic Mode Decomposition (DMD), concluding that buffet onset is connected with a single, unstable oscillatory eigenmode.

2.3.2 Experimental Investigations

Besides an extensive number of computational investigations, wind tunnel experiments have been conducted for the exploration of the three-dimensional buffet mechanism. Early studies by Roos [105] and Benoit and Legrain [6] revealed chordwise and spanwise pressure perturbations, with large-scale unsteadiness starting at the wing tip. In addition, characteristic frequencies yielded values higher compared to two-dimensional buffet, which correlates well with numerical studies. Further, Steimle et al. [124] applied fast-response pressure sensitive paint (PSP) measurements for buffet investigation. The resulting time sequences of pressure distributions yielded severe shock motion at the wing tip with a highly aperiodic character, which was also validated by numerical studies by Brunet and Deck [10] and Sator and Timme [113, 114].

Similar to numerical studies, mode decomposition methods have been applied for the evaluation of experimental buffet data. Using flow field data obtained from experimental investigations by Lawson et al. [64], Masini et al. [83] applied POD in order to identify dominant buffet flow features. By combining the extracted modes for the reconstruction of the corresponding pressure field, inboard propagating pressure perturbations were found, moving from wing root to wing tip. This phenomenon can be described as the movement of buffet cells proposed by Iovnovich and Raveh [54]. Consistent with the findings of Roos [105] and Benoit and Legrain [6], a study by Dandois [20] revealed a characteristic frequency spectra of the buffet instability, defined by Sr numbers between 0.2 and 0.6.

Using the well-known NASA CRM geometry [131], Koike et al. [59] applied unsteady pressure measurements for buffet investigation. The application of cross-correlation and analysis of coherence provided further insights into the origin of spanwise pressure fluctuations [40]. Considering several incidences, pressure perturbations originated near the wing root and moved to the wing tip. A later study by Sugioka et al. [146] verified the results of Koike et al. [59] through analysis of fast-response PSP buffet data originated from experimental investigations of the NASA CRM.

2.4 Data-Driven Modeling of Unsteady Aerodynamics

Within this section, an introduction to data-driven modeling and a fundamental classification of data-driven methods is given. Following the introduction, a literature overview of data-driven methods applied for unsteady aerodynamic modeling is provided. The section is concluded with a summary of studies focusing on data-driven transonic buffet prediction.

2.4.1 General Introduction

Motivated by the emerging amount of available flow field data, increasing research interest originated in the field of nonlinear system identification and data-driven modeling. The purpose of data-driven modeling is the construction of a ROM, which enables a reduced representation of a given aerodynamic system. Due to the reduction in complexity, the application of a ROM aims for reducing cost and time compared to extensive computational and experimental studies. Therefore, a ROM can be conditioned by means of either numerical and/or experimental flow field data. Commonly, flow field quantities are represented by samples defined in time and space as well as integrated aerodynamic forces and moments and flow field quantities [61].

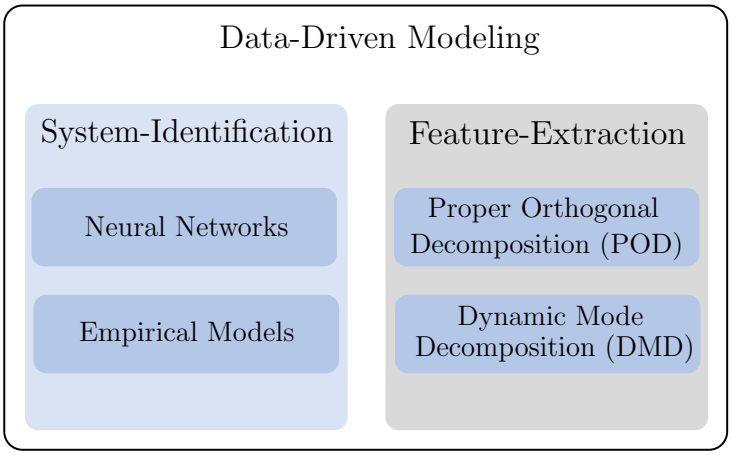


Figure 2.2: Classification of data-driven modeling.

In general, data-driven modeling methods can be differentiated into two types, depending on the type of data applied for training. The first category is defined by system-identification methods, whereas the second category is denoted by feature-extraction methods, as depicted in Figure 2.2.

By applying system-identification methods, the underlying aerodynamic system can be defined as a black-box or grey-box model, constructed by input-output data, without any or partial physical insights of the system [61, 88]. Considering an unsteady aerodynamic system, the inputs are commonly represented by external or self-sustained motions, whereas the outputs are defined by the corresponding integral quantities, such as force and moment coefficients [61]. Based on a numerical and/or experimental data set, the parameters of the system-identification model are determined. For performance evaluation, the trained model is validated using an unknown dataset, including similar system features as used for model training. Based on the structure of the underlying model, system-identification approaches can be further categorized in parametric and non-parametric models [34, 61]. Parametric models are defined by a known mathematical structure, whereas the structure of non-parametric models needs to be defined prior to the training process [61]. Representatives of parametric models are empirical aerodynamic models, whereas non-parametric models are represented by neural networks (NN).

In contrast to system-identification methods, feature extraction approaches are applied for high-dimensional, nonlinear flow field quantities, aiming for the representation of coherent structures and modes in the data [61]. These flow field quantities are defined by velocity, pressure or density. Modal analysis tools for the extraction of flow modes are defined by POD [7, 75] and DMD [116]. Using POD, the characteristic structures are filtered based on their energy content, whereas DMD makes a selection based on the frequency content of the respective modes.

In the following, a literature overview of recently developed unsteady aerodynamic ROM methods is given. Here, both system-identification and feature-extraction approaches are considered.

2.4.2 Application to Unsteady Aerodynamics

In order to accurately capture the unsteadiness of nonlinear flows, early studies proposed several dynamically nonlinear system-identification models, including nonlinear autoregressive with exogenous input (NARX) [155], nonlinear autoregressive moving average with exogenous input (NARMAX) [15, 100], Kriging as well as Volterra series [99] models. In addition, radial basis function neural network (RBFNN) identification approaches gained increasing attention for unsteady aerodynamic modeling. Won et al. [143] and Yao and Liou [147] employed a RBFNN for the representation of motion-induced forces of different wing configurations. Zhang et al. [155] applied a recurrent RBFNN to identify unsteady aerodynamics of the NACA0012 airfoil at large amplitude motions in transonic

flow. Ghoreyshi et al. [37] used a RBFNN for capturing airfoil motion in subsonic flows. Further, Winter and Breitsamter [137] trained a RBFNN using an orthogonal least-squares algorithm [13] and applied the trained model for the prediction of forces and moments acting on an airfoil excited by a defined external motion. Kou and Zhang [60] adapted a RBFNN in order to enable an incorporation of asymmetric wavelet kernels.

Besides RBFNN-based ROMs, several studies covered the application of multi-layer perceptron (MLP) [46] and RNN for unsteady aerodynamic modeling. Studies by Faller and Schreck [31, 32] focused on the application of a recurrent MLP neural network for the prediction of time series trends of aerodynamic coefficients. Similar to the work of Faller and Schreck [31,32], Marques and Anderson [82] applied a feedforward multi-layer neural network for the prediction of unsteady forces on the NACA0012 airfoil. Further, Mannarino and Mantegazza [80] used a recurrent MLP neural network in order to approximate aerodynamic coefficients of the NACA64A010 airfoil. Suresh et al. [125] modeled the lift coefficient of an airfoil using a RNN. Further, Mannarino and Mantegazza [81] adapted a RNN for capturing aeroelastic behavior of different airfoils. In addition to standard RNN, LSTM neural networks have been applied to some extent for unsteady aerodynamic modeling. Zhang et al. [70] developed a LSTM for the prediction of aerodynamic and aeroelastic airfoil characteristics considering flow conditions with varying Mach numbers.

Another field of nonlinear ROM methods, which has been used in the context of unsteady aerodynamic modeling, originated from the fundamentals of fuzzy logic theory. Winter and Breitsamter [138] proposed a neuro-fuzzy (NF) neural network for the prediction of the flutter boundary of the AGARD 445.6 wing, considering multiple Mach numbers. A study by Tatar et al. [77] used a NF-based ROM for capturing dynamic stall conditions of a blade. Based on a series connection of a NF model and a MLP, Winter and Breitsamter [141] proposed a novel method for an accurate representation of unsteady aerodynamics. Further, the hybrid model yielded precise prediction results of aerodynamic coefficients of the NLR7301 airfoil undergoing user-defined pitch and plunge motions in transonic flow [140].

Summarizing the aforementioned ROM approaches, a variety of efficient nonlinear system identification methods is available. However, if the prediction of spatio-temporal characteristics such as unsteady pressure distributions is considered, the application of the aforementioned nonlinear ROM approaches is not feasible. Spatio-temporal data is commonly represented by a grid-like topology, which drastically increases the number of output variables compared to time-series prediction tasks. As a solution for this remedy, recent studies dealt with the application of feature extraction methods such as POD and DMD. Park et al. [94] applied POD in combination with a neural network for wing design optimization tasks. Walton et al. [111] used a hybrid POD-RBF model for modeling the motion of the ONERA M6 wing in unsteady flows, whereas San et al. [91] combined POD with an artificial neural network (ANN) for modeling unsteady flows. Xiao et al. [145]

applied a similar hybrid model for modeling fluid-structure interaction. Further, Lindhorst et al. [56] employed a hybrid model based on POD and a non-recurrent RBFNN for the representation of aeroelastic behavior of the HIRENASD wing-fuselage configuration. Winter and Breitsamter [139] combined POD with a NF model to reconstruct wing pressure distributions on a civil aircraft configuration. Shifting the focus to DMD, Wu et al. [144] applied DMD for transition prediction on an airfoil. Further, Dawson et al. [24] used DMD for predicting force and pressure characteristics of a pitching airfoil in unsteady flow.

In addition to the application of dimensionality reduction techniques, deep learning methods such as CNNs and AEs have been applied to some extent for capturing spatio-temporal characteristics of unsteady, high-dimensional flow field data. A study by Afshar et al. [1] proposed a CNN-based ROM for the prediction of the velocity and pressure field around an airfoil, depending on the shape of the airfoil and corresponding flow parameters. Sekar et al. [117] used a CNN for feature extraction of an airfoil and further processed them as an input for a MLP neural network in order to predict unsteady flow field characteristics.

In order to improve the prediction performance using deep learning architectures, several studies proposed the application of hybrid deep learning models. In particular, series connected ROMs based on CNN and RNN have been used for a more efficient handling of high-dimensional flow field data. By using a CNN-based architecture for the reduction of high-dimensional flow field data and the RNN for the evolution of the reduced flow field, a faster training procedure is achieved. A study by Li et al. [71] applied a convolutional long short-term memory (ConvLSTM) neural network for the prediction of supersonic cascaded channel flow. Hasegawa et al. [45] proposed a series connected ROM based on a convolutional autoencoder (CNN-AE) and a LSTM for the prediction of unsteady-flow characteristics around bluff bodies with different shapes. Further, Nakamura et al. [87] applied the proposed CNN-AE/LSTM model for the prediction of turbulent channel flow.

Modeling of Unsteady Buffet Aerodynamics

Regarding the application of ROM methods for the prediction of buffet flows, only a small amount of studies originated in the last years. Gao et al. [35] applied an ARX-ROM for predicting the aerodynamics of the NACA0012 airfoil and further used them for aeroelastic analysis. Winter and Breitsamter applied the series connected NFM-MLP [141] for the prediction of aerodynamic forces and moments on the NACA0012 airfoil, undergoing user-defined pitching motions beyond the critical buffet angle of attack [142]. Their study revealed that the hybrid ROM is able to capture buffet characteristics on the airfoil, however, the representation of the lock-in effect seems to be a challenge for the applied ROM. A study by Candon et al. [12] compared three deep learning architectures, namely a RNN, a LSTM neural network and a bidirectional LSTM concerning the

prediction of buffet loads. Based on their investigations the bidirectional LSTM yields the best prediction capability regarding structural dynamic responses of a high-agility aircraft during buffet critical maneuvers.

Based on the aforementioned studies it was shown that only a few amount of studies dealt with ROM-based modeling of transonic buffet aerodynamics. So far, none of the ROM approaches was shown to provide an efficient and accurate prediction of integral quantities such as force and moment coefficient trends as well as unsteady pressure distributions. Therefore, in order to close this gap, this thesis focuses on the development of a ROM-based framework for modeling two- and three-dimensional integral and local buffet characteristics.

3 Deep Learning Approaches

In this chapter, the deep learning (DL) approaches applied for modeling buffet aerodynamics are introduced and explained in detail. In the first section, a general introduction and classification on the applied DL methods is given. Following the general introduction, an introduction to the architecture of RNNs is presented in Section 3.2. Based on the structure of a RNN, the training process as well as a guideline for the definition of suitable hyperparameters and activation functions is provided in Section 3.2.1 to Section 3.2.3. Further, an overview of available optimization algorithms and normalization techniques is presented in Section 3.2.4 and Section 3.2.5, respectively. Following the basics of NN theory, a detailed introduction to LSTM neural networks, CNNs and AEs is provided in Section 3.3 to Section 3.5. In Section 3.6, a description and classification of data sets used for training and performance evaluation of NN is given. The chapter is concluded with an overview of metrics applied for performance evaluation of a NN. To some extent, the underlying theory has been explained in publications of the author, see references [149, 150, 152, 153].

3.1 Introduction to Deep Learning

DL is a subfield of Machine Learning (ML), associated with algorithms, structures and capabilities inspired by the human brain. These algorithm and structures are represented by artificial neural networks (ANN), which are also known as NNs.

In general, the structure of a NN is defined as a certain number of processing units, referred to as neurons, which are connected with each other [136]. Further, each NN architecture is composed of three different types of layers, defined as input, hidden and output layer [9, 47], as depicted in Figure 3.1.

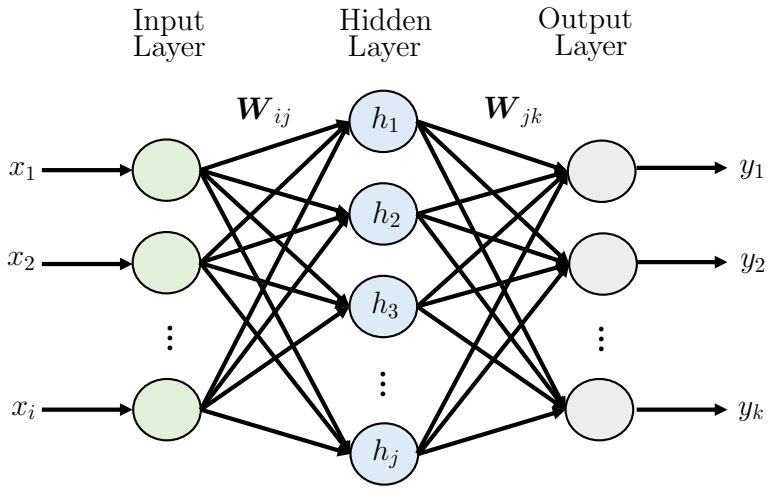


Figure 3.1: Structure of a simple neural network including one hidden layer (h) and connecting several inputs (x_i) to several outputs (y_k).

The input layer $\boldsymbol{x}$ receives all elements in the input data and passes them to the hidden layer $\boldsymbol{h}$. Depending on the network architecture and the underlying mathematical working principle, the data is processed within the hidden layer. Considering a single layer architecture, the data is passed to the output layer $\boldsymbol{y}$, whereas in deep NN, the data is passed to multiple consecutive hidden layers [136]. During the training of the NN, the weights $\boldsymbol{W}$ defining the contribution of each element on the resulting output are updated and optimized.

Based on the training procedure, NNs can be classified into supervised and unsupervised learning methods. The main distinction between the two approaches is the use of datasets with labeled inputs and outputs: While supervised methods use labeled datasets for classification and regression problems, unsupervised methods are trained by unlabeled datasets in order to cluster data or reduce its dimensionality.

In the present thesis, supervised as well as unsupervised DL algorithms are applied. In Figure 3.2, an overview of the DL approaches employed in the present thesis is given.

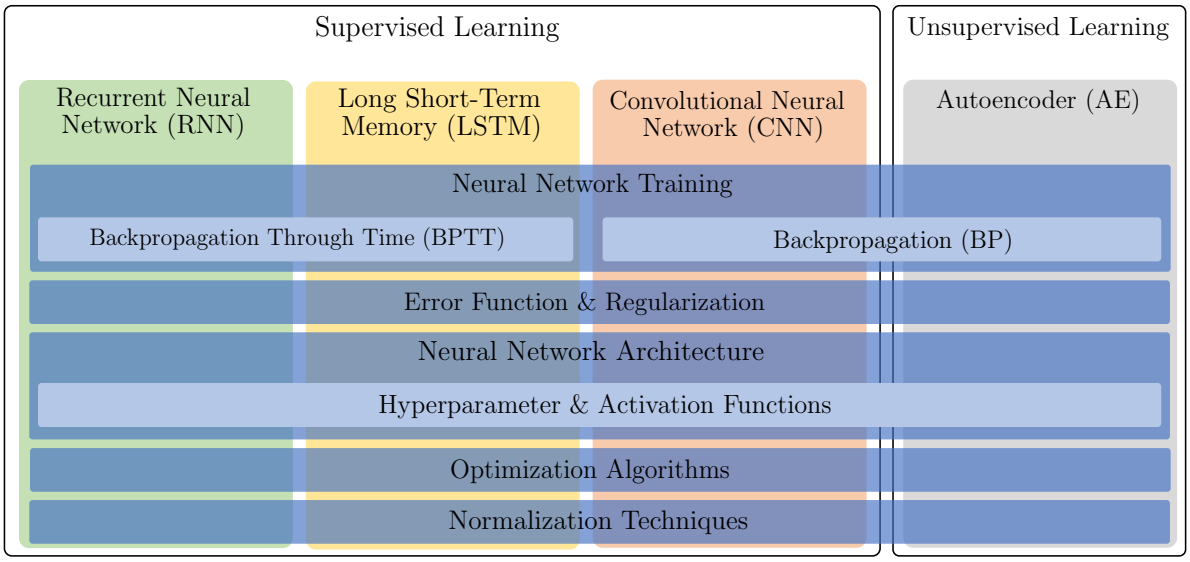


Figure 3.2: Overview and categorization of applied deep learning approaches.

Further, the applied DL methods can be categorized, depending on the structure of the input data and their target task. RNNs and LSTM neural networks, which represent a special type of RNN, are mainly used for representing time-series data or data that involves defined sequences [42]. In contrast, CNNs [66] are commonly applied for processing and predicting data arranged in a grid-like topology, such as images. AE [65] architectures as the main representative of unsupervised learning in this thesis are used for filtering important features of various data structures.

3.2 Recurrent Neural Network (RNN)

A RNN is a special type of an ANN, characterized by internal self-connections, which allow for the representation of time series data [8, 115]. In contrast to ordinary feed forward neural networks (FFNN), which are only applicable for the representation of independent data points, RNNs have the concept of memory that helps them store the states or information of previous inputs to generate the next output of a sequence. At each timestep, previous inputs, the activation of the neurons, as well as past computations define the current output, allowing the network to develop a memory based on several previous timesteps [8].

Similar to an ANN architecture, a common RNN architecture is composed of an input, hidden and output layer, as depicted in Figure 3.3.

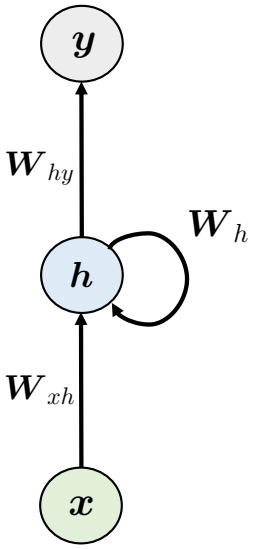


Figure 3.3: Architecture of a RNN including an input, hidden and output layer.

Here, $\boldsymbol{x}$ represents the input vector, $\boldsymbol{y}$ the output vector and $\boldsymbol{h}$ denotes the hidden state vector. $\boldsymbol{W}_{xh}$ is defined as the weight matrix connecting the inputs to the hidden layer, while $\boldsymbol{W}_{hy}$ is the weight matrix connecting the hidden layer to the output layer. $\boldsymbol{W}_h$ represents the weight matrix connecting the hidden state from the previous timestep to the hidden state of the current timestep.

The input and output layer are defined by feedforward connections, while the hidden layer is characterized by a recurrent connection [8]. In Figure 3.4, an unfolded RNN architecture, representing four consecutive timesteps ($t-2$, ..., $t+1$), is visualized. Based on the description proposed by Bianchi et al. [8], the following processing steps apply for a RNN: At each timestep t, an input $\boldsymbol{x}_t$ is processed by the input layer, summed with a bias vector $\boldsymbol{b}_x$ and multiplied with the input weight matrix $\boldsymbol{W}_{xh}$. The internal state of the network from the previous timestep $\boldsymbol{h}_{t-1}$ is summed with a bias $\boldsymbol{b}_h$ and is multiplied with the corresponding weight matrix of the recurrent connections $\boldsymbol{W}_h$.

Within the hidden layer, the input of the current timestep and the previous network state are combined and processed by the neurons by applying a linear activation [8]. For the computation of the output $\boldsymbol{y}_t$, the current hidden state $\boldsymbol{h}_t$ is multiplied with the matrix including the output weights $\boldsymbol{W}_{hy}$ and summed up with a bias vector $\boldsymbol{b}_y$.

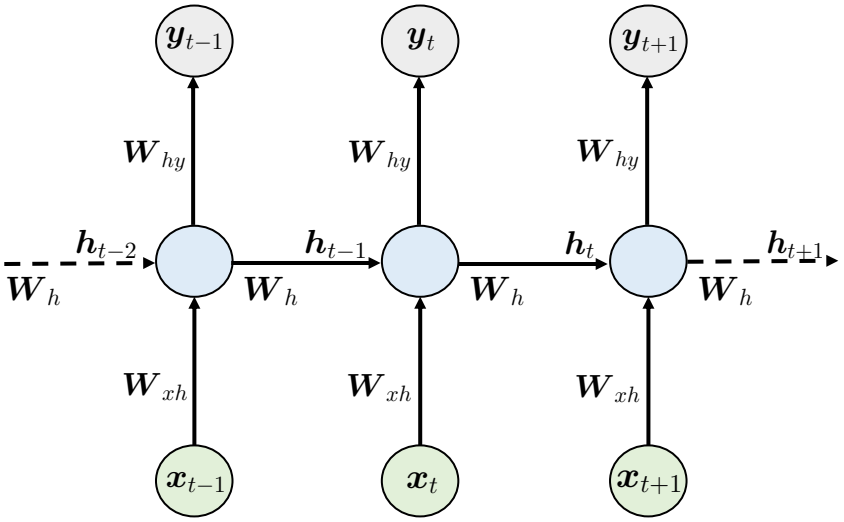


Figure 3.4: Architecture of the unfolded RNN representing three consecutive time steps.

The equations which define the update of the hidden state and the output processing are summarized in the following, with a defining an activation function which is chosen depending on the target task of the RNN.

$$\boldsymbol{h}_t = a(\boldsymbol{W}_{xh}(\boldsymbol{x}_t + \boldsymbol{b}_x) + \boldsymbol{W}_h(\boldsymbol{h}_{t-1} + \boldsymbol{b}_h)) \tag{3.1}$$

$$\boldsymbol{y}_t = a(\boldsymbol{W}_{hy}(\boldsymbol{h}_t + \boldsymbol{b}_y)) \tag{3.2}$$

3.2.1 Training of Neural Networks

The training of a NN is performed by a modification of its parameters, in particular the weights $\boldsymbol{W}$ and biases $\boldsymbol{b}$, by applying a gradient descent (GD) [103] optimization algorithm [8]. Therefore, a loss function L, which quantifies the accuracy of the network, is minimized. The loss function L is typically composed of an error function E, which evaluates the performance of the NN based on the entire training set, as well as an additional regularization term R_λ:

$$L = E(\boldsymbol{y},\hat{\boldsymbol{y}},\boldsymbol{W}) + R_\lambda(\boldsymbol{W}) \tag{3.3}$$

More specifically, the error function E evaluates the difference between the output $\boldsymbol{y}$ that the network should reproduce and the predicted value of the network $\hat{\boldsymbol{y}}$.

The regularization term R_λ depends on the hyperparameter λ, which controls the contribution of the regularization on the total loss the NN produces [8]. In Section 3.2.2, different regularization methods are introduced and discussed in more detail.

During the training of a NN, the GD optimization algorithm performs two consecutive steps, until a desired convergence is reached [8]. In the first step, a set of input data $\boldsymbol{x}$ is processed through the network (forward pass), configured with weights $\boldsymbol{W}$, and the error function at each timestep t is evaluated.

$$E(\boldsymbol{y},\hat{\boldsymbol{y}},\boldsymbol{W}) = \sum_{t=1}^{T} E(\boldsymbol{y}_t,\hat{\boldsymbol{y}}_t,\boldsymbol{W}_t) \tag{3.4}$$

In Equation 3.4, T refers to the total number of considered timesteps. In the second step, at each timestep t, the weights are updated based on the error the current weights produce due to deviation from the target output $\boldsymbol{y}$.

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t + \Delta\boldsymbol{W} \tag{3.5}$$

Therefore, the partial derivative of the error function with respect to all weights $\frac{\partial E}{\partial \boldsymbol{W}}$ is evaluated. By back-propagating [108] the gradient through the NN (backward pass), the error is minimized by finding the updated values of the weights $\Delta\boldsymbol{W}$.

$$\Delta\boldsymbol{W} = \eta(-\frac{\partial E}{\partial \boldsymbol{W}}) \tag{3.6}$$

In Equation 3.6, η is a hyperparameter defined as the learning rate.

By defining and inserting an arbitrary error function, Equation 3.6 can be re-written as follows:

$$\Delta\boldsymbol{W} = \eta(-\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{W}}) \tag{3.7}$$

with $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{W}}$ denoting the partial derivative of the output with respect to each weight. This partial derivative defines the gradient of the weight, which is commonly referred to as δ:

$$\delta = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{W}} \tag{3.8}$$

The elements of the gradient δ are computed by applying the chain rule:

$$\delta = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{W}} = \sum_{i=1}^{n} \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{h}_i}\frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{W}} \tag{3.9}$$

However, when training a RNN or LSTM, the principle of back-propagation (BP) [108] can be applied, but with a small conceptual change. The process is similar to the one used for a standard NN, with the exception that instead of a single timestep several previous timesteps must be considered. Therefore, the training procedure applied for RNN and LSTM is commonly referred to as back-propagation through time (BPTT) [134].

Considering the unfolded structure of the RNN as shown in Figure 3.4, three weight matrices ($\boldsymbol{W}_{xh}$, $\boldsymbol{W}_h$, $\boldsymbol{W}_{hy}$) must be adjusted when training a RNN. The partial derivative of the error function with respect to $\boldsymbol{W}_{hy}$ can be computed by a simple one step chain rule. For an arbitrary timestep t, the gradient calculation for the update of $\boldsymbol{W}_{hy}$ is defined as follows:

$$\frac{\partial E_t}{\partial \boldsymbol{W}_{hy}} = \frac{\partial E_t}{\partial \boldsymbol{y}_t} \frac{\partial \boldsymbol{y}_t}{\partial \boldsymbol{W}_{hy}} \tag{3.10}$$

In contrast, for the calculation of the partial derivative of the error function with respect to $\boldsymbol{W}_h$, all hidden states contributing to the output need to be considered. Therefore, the partial derivatives calculations are accumulated. Mathematically, the adjustment of $\boldsymbol{W}_h$ using BPTT can be written as follows:

$$\frac{\partial E_t}{\partial \boldsymbol{W}_h} = \sum_{i=1}^{t} \frac{\partial E_t}{\partial \boldsymbol{y}_t} \frac{\partial \boldsymbol{y}_t}{\partial \boldsymbol{h}_i} \frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{W}_h} \tag{3.11}$$

Equal to the adjustment of $\boldsymbol{W}_h$, the partial derivatives calculation associated to the update of $\boldsymbol{W}_{xh}$ are defined as follows:

$$\frac{\partial E_t}{\partial \boldsymbol{W}_{xh}} = \sum_{i=1}^{t} \frac{\partial E_t}{\partial \boldsymbol{y}_t} \frac{\partial \boldsymbol{y}_t}{\partial \boldsymbol{h}_i} \frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{W}_{xh}} \tag{3.12}$$

3.2.2 Error Function and Regularization

As already stated in Section 3.2.1, the task of training a NN is to minimize a loss function, which quantifies the prediction performance of a NN. The error function E, which evaluates the difference between a target output $\boldsymbol{y}$ and the output as predicted by the NN $\hat{\boldsymbol{y}}$, can be defined in different ways. The most commonly applied error functions are the mean squared error (MSE) and the root mean square error (RMSE). The MSE is defined as follows:

$$MSE = \frac{1}{N_S} \sum_{i=1}^{N_S} (\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i)^2 \tag{3.13}$$

with N_S representing the number of sample points in the available numerical or experimental data set. Similar to the MSE, the RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{N_S} \sum_{i=1}^{N_S} (\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i)^2} \tag{3.14}$$

For all investigations as presented in this thesis, the MSE is applied as the error function.

The regularization term R_λ in Equation 3.3 introduces a bias in order to improve the prediction performance of the NN by a reduction of overfitting [8]. In the following, the commonly applied regularization methods are introduced and briefly described:

- $\boldsymbol{L_1}$: By applying L_1 regularization, the sums of the absolute values of all weights are added and multiplied with the hyperparameter λ [89]. Therefore, the regularization term in Equation (3.3) is defined as $R_\lambda(\boldsymbol{W}) = \lambda_1(|w_1,...,w_n|)$. L_1 regularization enforces higher sparsity of the network parameters, ensuring that weights which are rarely used are driven to zero [8]. Therefore, L_1 reduces the set of weights, which enables a better performance for feature selection.

- $\boldsymbol{L_2}$: Compared to L_1 regularization, L_2 regularization adds the sum of the square of the weights, resulting in a regularization term of $R_\lambda(\boldsymbol{W}) = \lambda_1(w_1^2,...,w_n^2)$ [89]. L_2 maintains all weights on a homogeneously small level, which makes the regularization more stable for network training. Therefore, the use of L_2 regularization is preferred. However, a simultaneous application of L_1 and L_2 regularization is possible, with different coefficients selected for each network layer. In particular, the input and output weights should be treated differently, if both regularization methods are applied [5].

- **Dropout** [123]: During network training, the contribution of the weights on the training process changes. This results in parts of the network which have a higher influence on the training process than other parts. In order to circumvent this issue, a probability for the activation of a selected neuron can be implemented in the forward pass. More specifically, a probability factor p_{drop}, which is defined by values between [0,1], is applied to the output of the neurons in the hidden layer. After the training process is completed, the activations are scaled by the defined factor in order to regulate the output [8].

3.2.3 Neural Network Architecture

Besides the definition of an error function and a regularization term, the architecture of the NN needs to be defined. Therefore, different hyperparameters as well as layer activation functions are introduced and discussed in this subsection.

3.2.3.1 Hyperparameters

In the following, the hyperparameters which are necessary for applying deep learning models are divided into two groups. The first group is defined by optimizer hyperparameters, representing variables which are more related to the training and optimization process rather than the model itself. This category is represented by the learning rate η, the size of the batches as well as the number of training iterations, also referred to as epochs.

The second group is defined as model hyperparameters, which are applied for defining the structure of the NN. Considering RNN and LSTM neural networks, this group includes the number of hidden layers as well as the number of hidden units, which are also referred to as neurons, in each layer. In the following, a brief introduction of both optimizer and model hyperparameters is given, including a basic guideline how to define these parameters.

Optimizer Hyperparameters

- **Initial learning rate** η: The correct definition of the learning rate is important since it is the parameter with the highest impact on the networks efficiency. Typically, values between 0.001 and 10^{-5} are chosen for training, depending on the complexity of the model [5]. If the learning rate is chosen too large, gradient descent can inadvertently increase rather than decrease the training error. If the learning rate is too small, the training speed is not directly affected, however, it may become permanently stuck in a local optimum associated with a high training error.

  In order to improve the convergence during training without increasing the training time, the learning rate can be adapted during the training process. If training and validation losses do not change over a defined number of epochs, the learning rate can be reduced, which is commonly referred to as *learning rate decay* or *step decay*. Therefore, the learning rate can either be reduced linearly or exponentially after a defined number of iterations.

- **Batch size**: The size of the batches is typically chosen as a value between 2^3 and 2^8, representing powers of two. In general, small values result in the computation of matrix-vector products, whereas larger values enable a faster matrix-matrix multiplication [5]. Therefore, increasing the batch size results in faster computation but requires more input data, since less updates are performed within a single epoch. Further, the use of larger batches provide in general a more accurate approximation of the gradient [42]. In contrast, small batches are characterized by a regularization effect [135] due to the noise they add to the training process. If the batch size is chosen very small, an additional reduction of the learning rate might be necessary in order to maintain stability due to high variance in the gradient estimation [42]. Further, computational time is increased due to the reduced learning rate and the increasing number of iterations.

- **Number of epochs**: This hyperparameter can be optimized using the principle of *early stopping*. By monitoring the development of the validation error during the training process, the training can be finalized if the validation loss does not improve for a user-defined number of epochs.

Model Hyperparameters

- **Number of neurons**: The number of neurons in a multi-layer or stacked NN is the main measure of the NNs learning capacity. Depending on the complexity of the prediction problem, the number of processing units has to be chosen. Defining a size which is too large for the application case typically does not decrease the generalization performance [5], however, computational time is increased. In general, a different number of neurons can be selected for each hidden layer, however, a study by Larochelle et al. [63] has shown that an equal number performs better than individually increasing and decreasing the number of the hidden units in the layers.

- **Number of hidden layers**: Similar to the definition of the number of hidden units, the selection of the hidden layers strongly depends on the complexity of the system the network should reproduce. Therefore, NNs with a deeper architecture commonly result in a better generalization for a variety of tasks [42]. However, in order to find the optimal number of layers, systematic experimentation is required in order identify how many layers are needed for the specified dataset.

3.2.3.2 Activation Functions

The activation function in a NN defines how the weighted inputs are transformed into an output from each node in a layer of the network. More specifically, the activation function is applied to the data which has been internally processed by each neuron of the network. In general, the same activation functions are applied for all neurons included in either the hidden or output layer.

The choice of the activation function applied in the hidden layer has a high impact on the training process, whereas the choice of the activation function in the output layer defines the type of predictions the trained model can make. Therefore, it is common practice to use different activations functions in different parts of the model. However, as already stated above, the hidden layers typically use the same activation function, which differs from the activation function applied in the output layer.

Since NNs are trained by means of BP, which requires the computation of derivatives, the chosen activation function must be differentiable [42]. Typically, a nonlinear, differentiable activation function is applied in the hidden layers, since this enables the NN to learn more complex functions compared to a model trained with a linear activation function. In the following, a short overview of activation functions applied in the present thesis, is given.

- **Sigmoid** ($\boldsymbol{\sigma}$): The nonlinear sigmoid activation function, also sometimes referred to as logistic function, is often applied in NN architectures. The input values are rescaled to values between [0,1].

Therefore, large values are set to one, whereas very small values are defined as zero. The mathematical expression of the sigmoid is defined as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.15}$$

In addition, a visualization of the sigmoid function is provided in Figure 3.5.

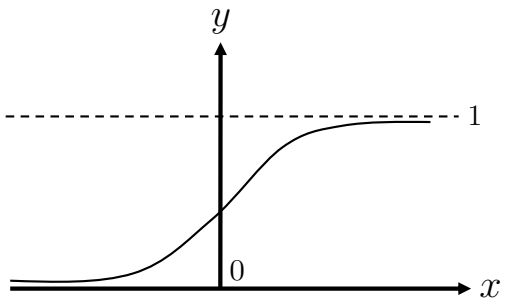


Figure 3.5: Visualization of the sigmoid function.

- **Hyperbolic tangent** (**tanh**): Similar to the sigmoid activation function, tanh rescales the given values between [-1,1]. The larger the input, the closer the output will be to 1, whereas small values are defined close to -1.

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3.16}$$

The hyperbolic tangent is visualized in Figure 3.6.

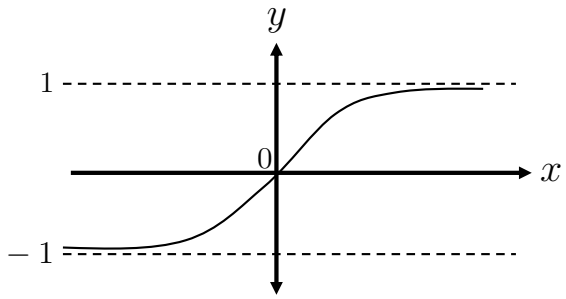


Figure 3.6: Visualization of the hyperbolic tangent function.

By applying both the sigmoid and hyperbolic tangent activation function in RNN architectures, an issue known as vanishing gradient [95] is likely to occur, which results in an insufficient training process after a certain number of training iterations. Since both activation functions limit the given input values, they are susceptible to saturate in the early stages of the training process [42]. Further, they are only sensitive to changes in certain areas of their input. Therefore, due to the limited sensitivity and the saturation, the weights are not updated in an efficient way anymore, which results in a performance loss of the model.

- **Rectified Linear Unit (ReLU)**: The rectified linear activation function, also referred to as ReLU, is the most common activation function used in NN, especially in CNNs. It is defined as a piecewise linear function, since it returns the same value, if the input value x is positive. If the input value x is negative, the return value is zero [42].

$$relu(x) = \begin{cases} x \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases} \tag{3.17}$$

A visualization of the ReLU function is provided in Figure 3.7.

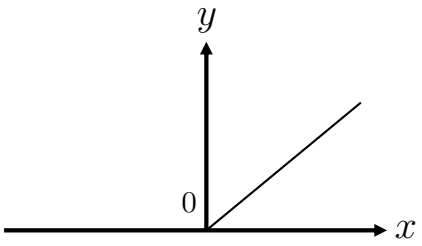


Figure 3.7: Visualization of the rectified linear unit (ReLU) function.

Compared to the sigmoid and tanh activation function, the ReLU function is less susceptible to the vanishing gradient problem [42]. Since the derivative is one if the input values is positive, the application of the ReLU can improve the training significantly. Further, ReLU offers the advantage of computational simplicity, since it does not require the use of an exponential calculation.

However, a drawback of ReLU is that if the activation is zero, a gradient-based optimization is not applicable anymore [42]. Therefore, variations of this activation function such as leaky ReLU and parametric ReLU originated, which enable the use of gradient-based methods. Further, they speed up the learning process and activate each neuron differently.

3.2.4 Optimization Techniques

In general, gradient based optimization algorithms applied for NNs can be divided in two groups. The first group, which uses the entire available training data set, is referred to as batch or deterministic gradient method, since it processes all available training samples simultaneously in one large batch. In contrast, the second group of optimization techniques only processes a small subset of the available data at a time. Therefore, they are referred to as stochastic or online optimization methods. In the framework of this thesis, the focus will be on stochastic optimization algorithms. Therefore, the most commonly used ones are introduced and discussed in detail in the following:

- **Stochastic Gradient Descent (SGD)**: One of the most applied gradient-update method is the SGD. By applying SGD, the available data set is split into small subsets of data, also referred to as mini batches, rather than evaluating the loss function over the entire training data set [8]. Each batch is individually forwarded through the network, the error and its gradient is calculated and back-propagated to update the corresponding weights. Based on the updated weights, the consecutive batch is run through the NN. Following the nomenclature of Bianchi et al. [8], the update equation for using SGD is defined as follows:

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \nabla L_t(\boldsymbol{W}_t) \tag{3.18}$$

  with the term $\nabla L_t(\boldsymbol{W}_t)$ denoting the gradient of the loss function with regards to all weights. However, even if SGD usually represents a safe and efficient optimization technique, its rate of convergence is slow and the gradient computation is likely to get stuck in a local optimum [22, 42].

- **Momentum**: The Momentum method, as proposed by Polyak [96], is defined as a first-order optimization method which introduces a hyperparameter β for the update of the weight. More specifically, the weights $\boldsymbol{W}$ are updated based on the sum of the current gradient $\nabla L_t(\boldsymbol{W}_t)$ and the previous weight update $\boldsymbol{W}_{t-1}$, which is additionally scaled by β [8]:

$$\begin{aligned} \boldsymbol{W}_{t,update} &= \beta \boldsymbol{W}_{t-1} + \eta \nabla L_t(\boldsymbol{W}_t) \\ \boldsymbol{W}_{t+1} &= \boldsymbol{W}_t - \boldsymbol{W}_{t,update} \end{aligned} \tag{3.19}$$

  Therefore, updates associated to previous timesteps are accounted for in the current weight update process. However, weight updates from timesteps that happened a longer time ago are less weighted than the ones happened recently. The values of β are defined between [0,1], however, a common choice is to set β =0.9 [42].

- **Adaptive Gradient (Adagrad)**: In comparison to the previous discussed approaches, the Adagrad [27] approach adaptively scales the learning rate for each parameter. It performs smaller updates on parameters which are frequently updated and larger updates for parameters associated to less frequently updates [42]. Given the updated weights from previous iterations $\nabla L_t \boldsymbol{W}_j$ ($j \in \{0,1,...,t\}$), each parameter of the weight matrix is updated as follows [8]:

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \frac{\nabla L_t(\boldsymbol{W}_t)}{\sqrt{\sum_{j=0}^{t} \nabla L_t(\boldsymbol{W}_j)^2 + \varepsilon}} \tag{3.20}$$

  In Equation 3.20 ε defines a hyperparameter which is introduced in order to avoid a division by zero.

However, the main weakness of Adagrad is the accumulation of the squared gradients over time, causing the learning rate to shrink and becoming infinitesimal small [42].

- **Root Mean Square Propagation (RMSprop)**: In order to solve the issue of the diminishing learning rate encountered when using Adagrad optimization, Tieleman and Hinton [126] introduced a method called RMSprop. RMSprop uses an exponential decaying average of the square gradients, which leads to decreasing shrinkage of the learning rate [8]:

$$\begin{aligned} \boldsymbol{v}_t &= \begin{cases} \delta \cdot \boldsymbol{v}_{t-1} + (1-\delta)\nabla L_t\left(\boldsymbol{W}_t\right)^2 & \text{if } \nabla L_t\left(\boldsymbol{W}_t\right) > 0 \\ \delta \cdot \boldsymbol{v}_{t-1} & \text{otherwise} \end{cases} \\ \boldsymbol{W}_{t+1} &= \boldsymbol{W}_t - \eta \boldsymbol{v}_t \end{aligned} \tag{3.21}$$

According to Equation 3.21, the learning rate is reduced by the factor $1-\delta$ if there are large variations in the gradient updates [8, 126]. Otherwise, the learning rate is increased by the decay rate δ, which is commonly defined as $\delta = 0.01$.

- **Adaptive Moment Estimation (Adam)**: Another approach called Adam, which was proposed by Kingma and Ba [57], combines the principles of the Adagrad and Momentum optimizer. In addition to applying an exponentially decaying average of the square gradients, Adam also keeps an exponentially decaying average of the gradients [8, 57]. The update equation using Adam are defined as follows [8]:

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \frac{\eta}{\sqrt{\hat{\boldsymbol{v}}_t + \varepsilon}} \hat{\boldsymbol{m}}_t \tag{3.22}$$

To estimate the moments, Adam utilizes exponentially moving averages, computed based on the gradient of the current mini-batch [8]:

$$\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\nabla L_t\left(\boldsymbol{W}_t^{(i)}\right) \tag{3.23}$$

$$\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\nabla L_t\left(\boldsymbol{W}_t^{(i)}\right)^2 \tag{3.24}$$

In Equation 3.23 and 3.24 β_1, β_2 and ϵ are hyperparameters with values $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$. Due to the initialization of $\boldsymbol{m}$ and $\boldsymbol{v}$ as zero vectors, they diverge to zero during the first training epochs [8]. In order to avoid this issue, both terms are corrected as $\hat{\boldsymbol{m}}$ and $\hat{\boldsymbol{v}}$ [8]:

$$\hat{\boldsymbol{m}}_t = \frac{\boldsymbol{m}_t}{1-\beta_1} \tag{3.25}$$

$$\hat{\boldsymbol{v}}_t = \frac{\boldsymbol{v}_t}{1 - \beta_2} \tag{3.26}$$

The methods discussed above only considered first-order derivatives of the loss function, which could lead to reduced optimization and a slow training progress. However, second-order methods include the computation of the Hessian matrix, which increases computational time even for small and medium size networks [8,42]. Therefore, they are not considered and discussed in detail within the frame of the present thesis.

3.2.5 Normalization Techniques

During the training of NNs, the weighted input distribution into each layer changes during the training process [52]. The inputs to each layer are highly affected by the output parameters of the previous layers, resulting in an increasing sensitivity towards the network parameters. This effect further intensifies with an increasing depth of the network. Therefore, an adaption of the learning rate as well as a more conservative parameter initialization is required, which increases training time and reduces the performance quality of NN, in particular those representing large nonlinearities [52]. This phenomenon is also referred to as internal covariate shift [118]. In order to solve this issue, a common approach is the normalization of the layer inputs during the training process. The two most commonly applied methods are briefly described in the following:

- **Batch Normalization (BN)**: BN was proposed by Ioffe and Szegedy [52] and defines a normalization technique applied to the input of each layer, in particular to each training batch. More specifically, BN standardizes each summed input $\boldsymbol{x}_i$ of the current batch B using its mean μ_B (see Equation 3.27) and standard deviation σ_B (see Equation 3.28) across the available training data [52].

$$\mu_B = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{x}_i \tag{3.27}$$

$$\sigma_B = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (\boldsymbol{x}_i - \mu_B)^2} \tag{3.28}$$

  In Equation 3.27 and 3.28, m is defined as the number of values included in the batch B. Following the computation of the mean and variance, the input is normalized by means of the batch statistics [52]:

$$\hat{\boldsymbol{x}}_{\boldsymbol{i}} = \frac{\boldsymbol{x}_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \tag{3.29}$$

In Equation 3.29, ε is a constant added to the variance in order to ensure numerical stability [52]. In the last step, the normalized input $\hat{\boldsymbol{x}}$ is transformed into an output $\boldsymbol{y}_i$ [52]:

$$\boldsymbol{y}_i = \gamma \hat{\boldsymbol{x}}_{\boldsymbol{i}} + \beta = BN_{\gamma,\beta}(\boldsymbol{x}_i) \tag{3.30}$$

In order to adapt the mean and standard values to each layer, additional learnable parameters (γ, β) are introduced in Equation 3.30.

By applying BN, small changes to the parameters which result in large changes in the activations in gradients, are prevented. Therefore, BN allows for the usage of higher learning rates, enabling a faster and more regularized training process [52]. In addition, a larger decay rate for the adaption of the learning rate can be applied. Further, the stability during the training makes the NN less sensitive to the selection of the weight initialization method. If BN is applied, the use of dropout for regularization is not recommended, since the random dropping may cause noisy batch statistics [52].

However, although BN enables a more robust training process, some disadvantages are given. Since BN calculates the statistics of batches at every training epoch, large batch sizes are required for an effective approximation of the batch statistics [3]. Therefore, the application of BN is limited to models which do not include very small batches. Another drawback of BN is the limited use in RNN and LSTM. During training of RNN, the inputs fed to the recurrent neurons of the hidden layer often vary depending on the selected sequence length, which requires different statistics for different time-steps [3]. Therefore, γ and β need to be adapted for each time-step, which instead adds complexity rather than simplicity to the network training process.

- **Layer Normalization (LN)**: Due to the above mentioned disadvantages of BN, Ba et al. [3] proposed a normalization technique called LN. Compared to BN, LN computes the statistics of the summed inputs to the neurons within a hidden layer instead of the batch statistics. Therefore, the computation of the mean μ_{LN} and the standard deviation σ_{LN} over all hidden units h in a single layer is defined as follows:

$$\mu_{LN} = \frac{1}{h} \sum_{i=1}^{h} \hat{\boldsymbol{x}}_{\boldsymbol{i}} \tag{3.31}$$

$$\sigma_{LN} = \sqrt{\frac{1}{h} \sum_{i=1}^{h} (\hat{\boldsymbol{x}}_{\boldsymbol{i}} - \mu_{LN})^2} \tag{3.32}$$

In contrast to the computation of μ_{BN} and σ_{BN}, all hidden units in a single layer share the same statistics μ_{LN} and σ_{LN} [3]. However, different training cases are associated with different statistics. Therefore, the application of LN does not impose a constraint on the size of the batches [3]. Further, since the normalization terms only depend on the layer inputs of the current time-step, LN is easier applicable to RNN and LSTM.

3.3 Long Short-Term Memory (LSTM)

The LSTM neural network architecture has been proposed by Hochreiter and Schmidhuber [50] and represents a special type of a RNN, which has proven to be powerful for time-series prediction tasks. While a standard RNN only provides the capability of a short-term memory, which allows the use of previous information to some extend, the LSTM architecture additionally enables the prediction of long-term dependencies in time-series data. Considering the application for unsteady aerodynamic modeling, the representation of time-delayed effects can also be captured by the LSTM. Further, the application of the LSTM solves for the problem known as vanishing gradient [95], where the training of the network saturates in the early stages of the training procedure.

The hidden layer of a LSTM neural network consist of recurrently connected blocks, which are referred to as memory blocks or cells. Each cell is defined by a characteristic gate structure, which processes the incoming information in several steps. Each cell of the LSTM hidden layer is defined by three gates. These three gates, as shown in Figure 3.8, are defined as the forget gate $\boldsymbol{f}$, the input gate $\boldsymbol{i}$ and the output gate $\boldsymbol{o}$.

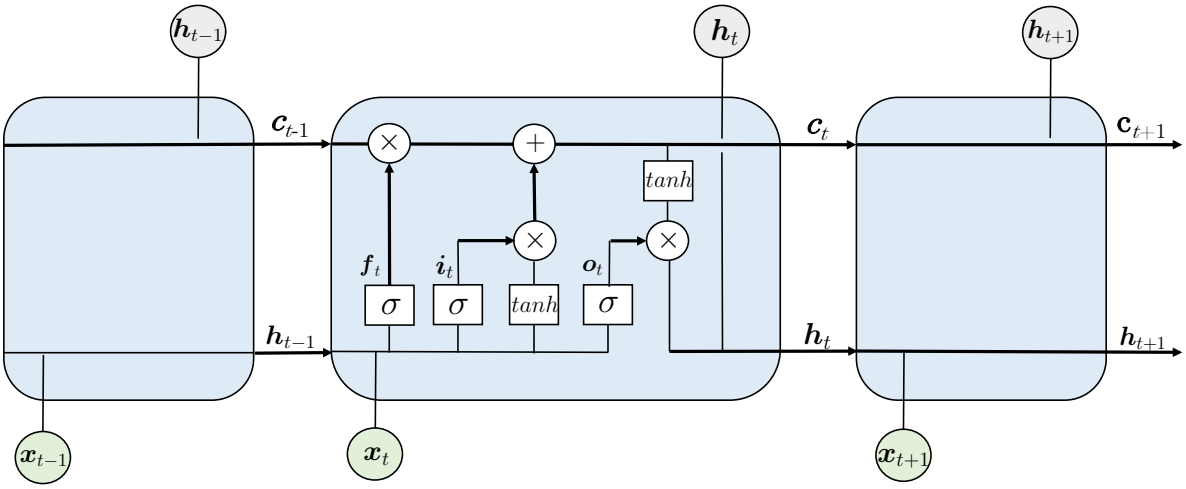


Figure 3.8: Architecture of a LSTM memory cell representing the characteristic gate structure.

The forget gate $\boldsymbol{f}$ collects the input of the current time step $\boldsymbol{x}_t$ as well as the vector representing the output from the previous time step $\boldsymbol{h}_{t-1}$:

$$\boldsymbol{f}_t = a(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{W}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \tag{3.33}$$

The output from the previous time step $\boldsymbol{h}_{t-1}$ is also referred to as the hidden state of the LSTM cell. Both inputs of the forget gate are multiplied with a set of weights $\boldsymbol{W}_f$ and a bias $\boldsymbol{b}_f$ is added. By means of an activation function a, which is typically implemented as the sigmoid σ function (see Section 3.2.3.2), the forget gate decides which information is discarded from the cell.

Equal to the forget gate $\boldsymbol{f}$, the current time step $\boldsymbol{x}_t$ as well as the hidden state from the previous time step $\boldsymbol{h}_{t-1}$ are combined and processed in the input gate $\boldsymbol{i}$. The mathematical formulation can be written as follows:

$$\boldsymbol{i}_t = a(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{W}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \tag{3.34}$$

with $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$ defining the weight matrix and the bias of the input gate, respectively. Further, an activation a is applied, which is also commonly defined by a σ activation.

Based on the selected inputs, a new cell state vector $\tilde{\boldsymbol{c}}_t$ is created:

$$\tilde{\boldsymbol{c}}_t = a(\boldsymbol{W}_h \boldsymbol{x}_t + \boldsymbol{W}_h \boldsymbol{h}_{t-1} + \boldsymbol{b}_h) \tag{3.35}$$

The activation function a in Equation 3.35 is commonly implemented as a hyperbolic tangent ($tanh$) activation. $\boldsymbol{W}_h$ and $\boldsymbol{b}_h$ defining the weight matrix and the bias of the hidden state, respectively.

Based on the new cell state $\tilde{\boldsymbol{c}}_t$, the current cell state $\boldsymbol{c}_t$, which defines the long-term memory of the LSTM neural network, is updated with new information. Therefore, the cell state from the previous time step $\boldsymbol{c}_{t-1}$ is multiplied with the forget gate vector $\boldsymbol{f}_t$ and the current cell state is updated with the input gate vector $\boldsymbol{i}_t$:

$$\boldsymbol{c}_t = \boldsymbol{f}_t \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \tilde{\boldsymbol{c}}_t \tag{3.36}$$

After passing the input gate, the data of the current input $\boldsymbol{x}_t$, the previous hidden state $\boldsymbol{h}_{t-1}$ as well as the current cell state $\boldsymbol{c}_t$ are processed by a sigmoid and tanh activation:

$$\begin{aligned} \boldsymbol{o}_t &= \sigma(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{W}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \\ \boldsymbol{h}_t &= \boldsymbol{o}_t \cdot tanh(\boldsymbol{c}_t) \end{aligned} \tag{3.37}$$

with $\boldsymbol{W}_o$ and $\boldsymbol{b}_o$ denote the weight matrix and the bias of the output gate, respectively.

The new cell state $\boldsymbol{c}_t$ denotes the previous cell state $\boldsymbol{c}_{t-1}$ for the consecutive LSTM cell, whereas the updated hidden state $\boldsymbol{h}_t$ becomes the previous hidden state $\boldsymbol{h}_{t-1}$. These gate processing steps are repeated until all sequences of the input data are processed by all including LSTM cells.

3.4 Convolutional Neural Network (CNN)

A CNN [66] is a type of NN mainly developed for processing data represented by a grid-like topology [42]. The underlying mathematical principle is defined by a convolution operation. Although CNNs are applicable to time-series data, they are mainly used for image data, which is characterized as a 2D or 3D grid of pixels.

In general, in addition to the input and output layer, a CNN includes three layers: a convolutional layer, a pooling layer and a fully-connected (FC) layer, as shown in Figure 3.9. Each CNN architecture includes a convolutional layer as the layer following the input layer and a FC layer as the last layer before the output layer. Between these two layers, additional convolution and pooling layers can be added in order to increase the complexity of the model. In the earlier layers, basic features such as colors and edges are identified, while the subsequent layers focus on extracting larger parts of the input data [42]. However, the convolutional layer represents the main building block of each CNN architecture.

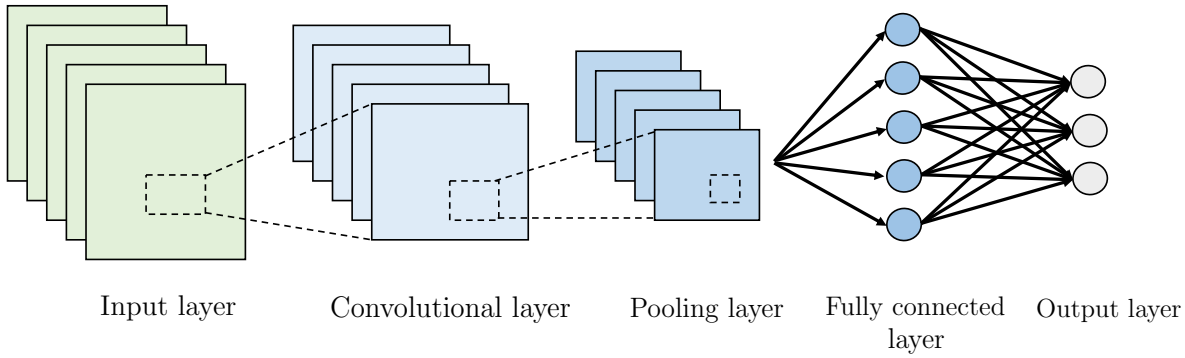


Figure 3.9: Architecture of a basic convolutional neural network (CNN).

Assuming an input data set defined by 2D arrays, the elements are stored in a grid-like layout and the position of each element can be defined by an index (i,j) [106]. Following the nomenclature of Goodfellow et al. [42] and Rosov and Breitsamter [106], the corresponding convolution operation can be written as follows:

$$\boldsymbol{y}_{i,j} = \sum_{m=0}^{\boldsymbol{H}_k-1} \sum_{n=0}^{\boldsymbol{W}_k-1} \boldsymbol{x}_{i+m,j+n} \boldsymbol{W}_{m,n} + \boldsymbol{b}_k \tag{3.38}$$

with $\boldsymbol{x}_{i,j}$ denoting the input at index (i,j) of the two-dimensional input $\boldsymbol{x}$. $\boldsymbol{W}_{m,n}$ is defined as the weights at point (m,n) of the corresponding filter, which selects important features of the input data. Commonly, the filter is also referred to as kernel and is defined by a size of $\boldsymbol{H}_k \times \boldsymbol{W}_k$. Depending on the size of the input data, the height (H_k) and width (W_k) of the kernel are defined by a matrix of size 2×2 or 3×3. The kernel slides stepwise over the input data, performing element-wise multiplications with each entry of the input data. At each step, the results are summed up and stored as a new output. Therefore, the original input data is converted from a 2D matrix into

another, reduced 2D matrix of features, which essentially represent the weighted sums of the input feature. The output of the 2D convolution operation is defined by $\boldsymbol{y}_{i,j}$.

Besides a 2D data set, the input data can be defined by a 3D matrix [42]. Therefore, the 2D input representing both spatial dimensions is extend with a third dimension, which is referred to as the channel dimension $\boldsymbol{C}$ [106]. The convolution operation assuming a three-dimensional input is defined as follows [106]:

$$\boldsymbol{y}_{i,j,o} = \sum_{l=0}^{\boldsymbol{C}_{in}-1} \sum_{m=0}^{\boldsymbol{H}_k-1} \sum_{n=0}^{\boldsymbol{W}_k-1} \boldsymbol{x}_{i+m,j+n,o} \boldsymbol{W}_{m,n,l,o} + \boldsymbol{b}_k \tag{3.39}$$

The size of the input $\boldsymbol{x}$ is defined by the number of input channels, the height and the width ($\boldsymbol{C}_{in} \times \boldsymbol{H}_x \times \boldsymbol{W}_x$), whereas the size of the corresponding kernel is defined as ($\boldsymbol{C}_{in} \times \boldsymbol{C}_{out} \times \boldsymbol{H}_k \times \boldsymbol{W}_k$). $\boldsymbol{C}_{in}$ represents the number of input channels, whereas $\boldsymbol{C}_{out}$ denotes the number of outputs arranged along the channel dimension in the output $\boldsymbol{y}$ [106]. Equation 3.39 can be further extended by a parameter referred to as stride s, which defines the number of elements the filter selects at each step:

$$\boldsymbol{y}_{i,j,o} = \sum_{l=0}^{\boldsymbol{C}_{in}-1} \sum_{m=0}^{\boldsymbol{H}_k-1} \sum_{n=0}^{\boldsymbol{W}_k-1} \boldsymbol{x}_{l,i\times s+m,j\times s+n} \boldsymbol{W}_{o,l,m,n} + \boldsymbol{b}_k \tag{3.40}$$

By applying the striding technique, the output size can be decreased compared to the input size. The idea of the stride is to skip some of the slide locations of the kernel. By using a stride of one, slides are picked an entry apart, so basically every single slide, acting as a standard convolutional operation (see Equation 3.39). If the stride is chosen as two, slides two entries apart are picked, reducing the size of the input data by a factor of two and so on. Therefore, the higher the size of the stride is chosen, the smaller the corresponding output.

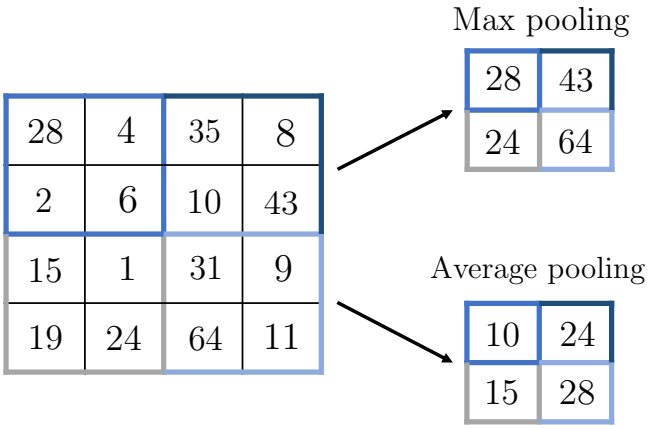


Figure 3.10: Schematic of max and average pooling.

Similar to the convolutional layer, the application of a pooling layer aims at reducing the spatial size of the input data. The main reason for using a pooling operation is the reduction of computational time and power for processing the data. Further, it is

useful for the extraction of data features which are invariant against translation and rotation [42]. In general, two types of pooling are applicable: Max pooling and average pooling. Max pooling returns the maximum value of the image part covered by the kernel, whereas average pooling return the average of all elements. In Figure 3.10, the schematic of max and average pooling is visualized.

For some applications, the size of the filter does not fit the size of the input data. Therefore, a technique referred to as zero padding, is applicable. Considering a standard convolution, the data points on the edge of the input matrix get trimmed off, if the filter does not have the correct size. Since these trimmed data points are at the edges of the matrix, the output size reduces compared to the input size, which is not beneficial in some cases. Padding solves this issue by adding extra pixels (usually of value 0, which often refers to as zero-padding) at the edges of the input matrix. By doing so, the original edge pixels are allowed to be at the center of the kernel, producing an output the same size as the input.

As the last processing step in a CNN, a FC layer is applied in order to connect the data to the output layer. This layer performs the classification part based on the features extracted from previous layers and their corresponding filters. However, in comparison to the convolutional and pooling layer, which mostly use ReLU activation functions, FC layers apply tanh or sigmoid activation functions [42].

3.5 Autoencoder (AE)

An AE [49, 65] is a type of NN generally used for data denoising, dimensionality reduction or feature learning [42]. AEs are trained in order to reconstruct a given input to its output in an unsupervised way, using the principle of BP. The architecture of an AE is defined by three parts, the encoder, decoder and a latent representation, also referred to as bottleneck. In Figure 3.11, the basic architecture of an AE is visualized.

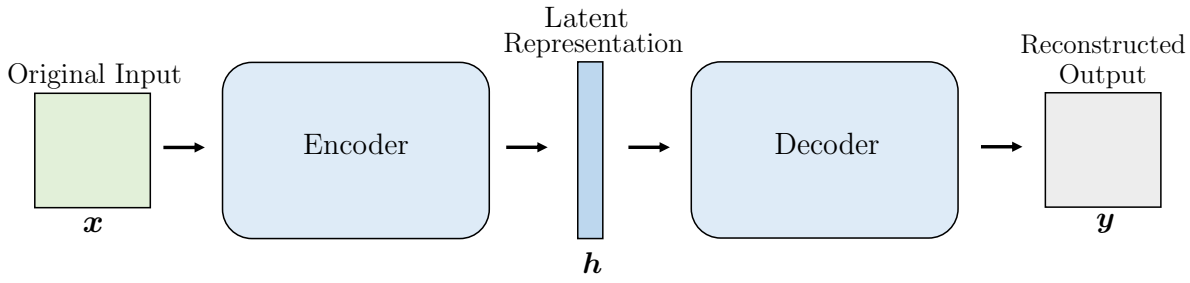


Figure 3.11: Architecture of an autoencoder (AE).

The encoder compresses the input $\boldsymbol{x}$ into a low dimensional representation, also referred to as latent representation $\boldsymbol{h}$ ($\boldsymbol{h} = f(\boldsymbol{x})$). The latent representation contains the important, filtered content of the input data. After passing the encoder, the decoder reconstructs an output $\boldsymbol{y}$ ($\boldsymbol{y} = f(\boldsymbol{h})$) from the most important input features extracted by the encoder.

In general, AEs can be differentiated into AEs composed of either convolutional or FC layers. AEs with FC layers are represented by single or multilayer AEs, including one or more hidden layers, respectively. However, within the frame of this thesis, the focus will be on convolutional AE (CNN-AE). Similar to a classical CNN architecture, both the encoder and decoder of a CNN-AE are composed of one or several convolutional and pooling layers as well as selected activation functions.

Besides a classification of AEs based on their layer structure, AEs can be further differentiated by the way the data is encoded and decoded. The most popular representatives are defined by sparse, denoising and variational AEs [42]. In the following, a more detailed description of variational AEs (VAR-AE) is given.

A VAR-AE is defined as an AE with a regularised training procedure, aiming to avoid overfitting and improving the latent representation of the input data [42]. In comparison to a standard CNN-AE, the input is encoded as a distribution over the latent space, rather than a single data point. In practice, a normal distribution is selected, defining the mean μ and variance σ of the latent state distribution as the output. After the encoding process, a data point from the latent representation is randomly sampled from the input distribution and fed into the decoder. In Figure 3.12, the working principle of a VAR-AE is depicted.

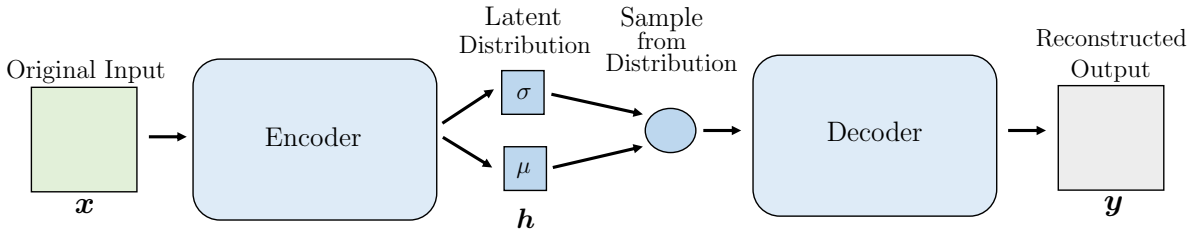


Figure 3.12: Architecture of a variational autoencoder (VAR-AE).

Since the data from the defined distribution is randomly selected, a small modification in the training process is required, which is commonly referred to as reparameterization. By applying reparameterization, a random value ϵ sampled from a unit Gaussian is shifted by the mean and scaled by the variance of the latent distribution:

$$\boldsymbol{h} = \mu + \sigma \cdot \varepsilon \tag{3.41}$$

Applying this reparameterization step enables a random selection, which is independent of the parameter optimization during training.

3.6 Data Sets and Preprocessing

In order to accurately train a NN and verify its performance quality, different data sets including different system features are required. In the following, three different data sets are introduced and characterized, which are typically applied for training, validation and testing of NNs.

- **Training Data**: In order to train a NN, an initial set of data, referred to as training data, is required. This data set, which determines the parameters of the applied model, acts as a baseline for further application and utilization of a NN. Based on the training data, the algorithm recognizes patterns in the dataset, which should cover all important system features. Therefore, the quality as well as the quantity of the training data set determine the accuracy and the performance of the model. The amount of training data which is needed to build an accurate ML model depends on the complexity of both the system as well as the structure of the algorithm itself. One possible way to test how much training data is required is building the model and evaluate the performance based on the available data. If the model is applied for the prediction of the output of the training case, the model should be able to reproduce the desired information. Otherwise, the training data set includes too few data for an adequately description of the underlying system. Further, also the model structure including the characteristic parameters could be inefficient.

- **Validation Data**: Besides the training data set, an additional data set, which is considerably smaller than the training data set, is introduced to the NN during the training process. This data set is commonly referred to as validation data set and is used to evaluate the performance of the model during the training process.

  On the one hand, the validation data is used to avoid overfitting. If the model is overfitted, it includes the system characteristics and the noise in the training data to an extent that the performance of the model on an unknown data set is negatively impacted. On the other hand, the validation data set is applied to monitor the loss of the error function during the training. If the progress of the training is only monitored by means of the error function evaluated on the training data, the error commonly decreases with increasing number of training iterations [42, 136]. In contrast, the error computed based on the validation data set does not necessarily show a comparable convergence improvement [136]. Therefore, in general the convergence of the validation data set error is applied as a measure for the termination of the overall training process. Here, the model with the smallest validation error is selected for further applications. Consequently, since the validation data has already been introduced to the model during training, it should not be used for further performance assessment of the trained model [136].

- **Test Data**: In order to assess the prediction quality of the trained model, a third data set, referred to as test data, is applied. The test data set contains information which was neither applied for defining the model structure nor for hyperparameter tuning. However, the test set should include similar system features in order to be representative for the investigated system. If the test set differs to a large extent from the training and validation data, the prediction performance of the trained model might decrease.

Within the present thesis, the available numerical or experimental full dataset is divided into training, validation and test dataset. Prior to the segmentation, each data set is preprocessed. In practice, it is nearly always advantageous to apply preprocessing transformations to the data before it is presented to the NN. Generally, data preprocessing is accomplished by either normalization or standardization. In the present work, data normalization is applied. The normalization technique rescales the data from the original range in a way that all vales are within a defined range, i.e. [0,1] or [-1,1]. It should be noted for completeness that the predicted data must be re-normalized in order to obtain consistent output values.

3.7 Evaluation Metrics

In order to provide a quantitative measure between the results predicted by a trained NN applied to a test data set and the reference CFD - or experimental solution, different evaluation metrics are applied within this thesis.

For the computation of integral buffet characteristics such as force and moment coefficient trends, both the deviation in amplitude and phase between the original and predicted time-series must be considered. Therefore, an evaluation metric as proposed by Russel [110], which accounts for an amplitude and phase error in transient data, is applied. The proposed error metric is defined by a comprehensive error C_{err}, which is defined as follows:

$$C_{err} = \sqrt{\frac{\pi}{4}(A_{err}^2 + P_{err}^2)} \tag{3.42}$$

The comprehensive error is composed of two terms, defining the amplitude error A_{err} and a phase error P_{err}. The definition of the phase and amplitude error are given in Equation 3.43 and Equation 3.44, respectively.

$$P_{err} = \frac{1}{\pi} cos^{-1} \left(\frac{\sum_{i=1}^{N_S} \hat{\boldsymbol{y}}_i \boldsymbol{y}_i}{\sqrt{\sum_{i=1}^{N_S} \hat{\boldsymbol{y}}_i^2 \sum_{i=1}^{N_S} \boldsymbol{y}_i^2}} \right) \tag{3.43}$$

$$A_{err} = sign(RME) Log_{10}(1 + |RME|) \tag{3.44}$$

In addition, the computation of a relative magnitude error (RME), as defined in Equation 3.45, is necessary for the definition of the A_{err}. In Equation 3.42 to Equation 3.45, the reference CFD or experimental solution is denoted by $\hat{\boldsymbol{y}}$, whereas the sample predicted by the NN is represented by $\boldsymbol{y}$. N_S defines the number of samples included in the applied data set.

$$RME = \frac{\sum_{i=1}^{N_S} \boldsymbol{y}_i^2 - \sum_{i=1}^{N_S} \hat{\boldsymbol{y}}_i^2}{\sqrt{\sum_{i=1}^{N_S} \boldsymbol{y}_i^2 \sum_{i=1}^{N_S} \hat{\boldsymbol{y}}_i^2}} \tag{3.45}$$

Following the definition of Russel [110], the phase error is defined by values between 0 and 1. Here, a phase error of $P_{err} = 0$ indicates two signals without a phase error, whereas $P_{err} = 1$ corresponds to two signals being completely out of phase. In contrast, a RME value of $RME = 1$ indicates that the magnitude of a signal s_2 is ten times larger than the magnitude of a signal s_1 ($s_1 = 10 \cdot s_2$), whereas a $RME = 2$ defines $s_1 = 100 \cdot s_2$. With increasing RME, the value of the amplitude error also increases. Consequently, a large phase and amplitude error result in an overall increasing comprehensive error. Therefore, in general, the smaller the comprehensive error, the smaller is the phase shift and the amplitude shift between the reference and modeled data.

In addition to the application of the C_{err}, the fit factor Q as introduced by Ljung [74] is applied in this thesis. The fit factor between a reference solution $\boldsymbol{y}$ and the results $\hat{\boldsymbol{y}}$ modelled by the NN is defined as follows:

$$Q = 100\% \cdot \left(1 - \frac{\sqrt{\sum_{s=1}^{N_S} (\boldsymbol{y}(s) - \hat{\boldsymbol{y}}(s))^2}}{\sqrt{\sum_{s=1}^{N_S} (\boldsymbol{y}(s) - \bar{\boldsymbol{y}})^2}} \right) \tag{3.46}$$

In order to compute the fit factor, the mean response $\bar{\boldsymbol{y}}$ (see Equation 3.47) of the NN is inserted as the model output $\hat{\boldsymbol{y}}$. In general, a fit factor of 100% indicates an exact agreement between the reference solution and the NN response [74, 136].

$$\bar{\boldsymbol{y}} = \frac{1}{N_S} \sum_{s=1}^{N_S} \hat{\boldsymbol{y}} \tag{3.47}$$

In order to provide a better classification of the proposed error metrics, three sine signals representing different amplitudes and phases, are compared in the following. The signals s are defined as follows:

$$s(t) = A \cdot sin(\omega t - \phi) \tag{3.48}$$

with ω and ϕ defining the angular frequency and the phase angle, respectively.

Focusing on the amplitude error, Figure 3.13 shows three different sine signals with amplitudes $A = [0.5, 1, 1.25]$, while the differences in the phase is $\phi = 0$. In contrast, Figure 3.14 depicts three sine signals representing an amplitude of $A = 1$, whereas the phase is varied as follows: $\phi = [0, 0.5\pi, \pi]$. In Figure 3.15, sine signals with both varying amplitudes and phases compared to the reference sine signal ($A = 1$, $\phi = 0$) are depicted. Here, the amplitudes are chosen as $A = [0.5, 1.25]$, whereas the corresponding phases are defined as $\phi = [-0.5\pi, 0.3\pi]$.

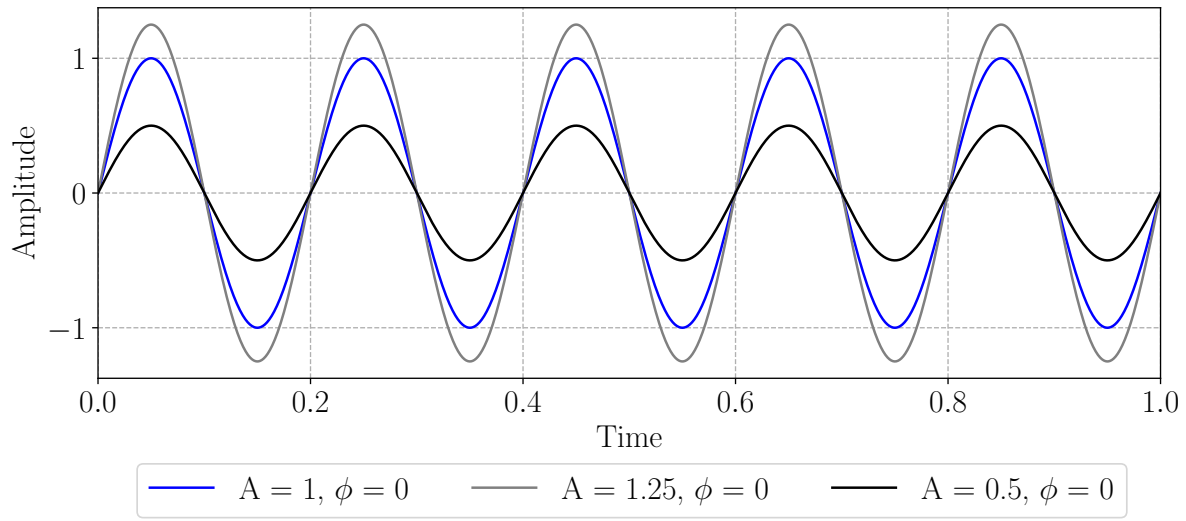


Figure 3.13: Sine signals representing three different amplitudes A = [0.5, 1, 1.25] at a constant phase of $\phi = 0$.

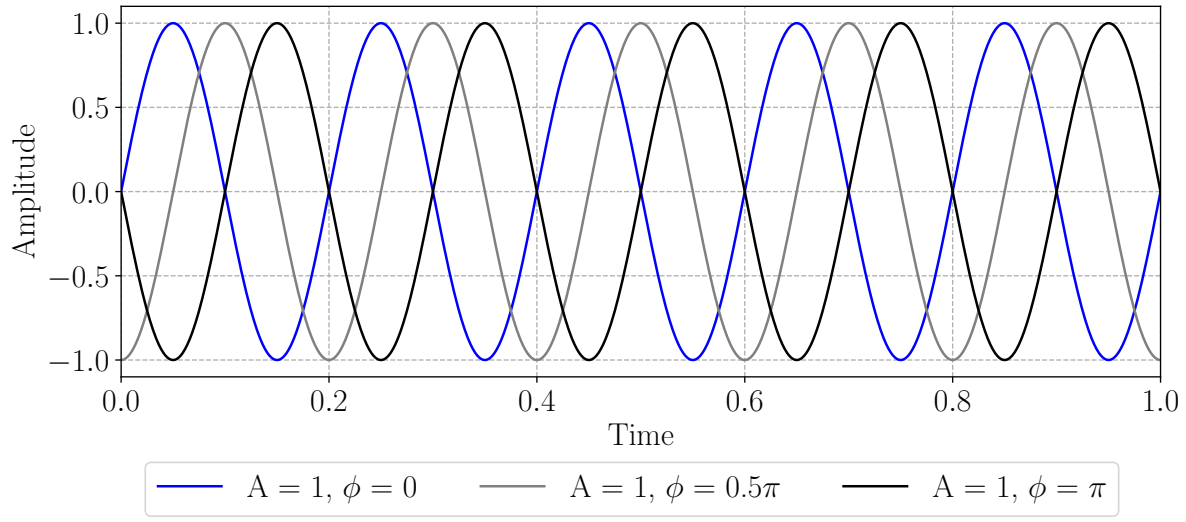


Figure 3.14: Sine signals representing three different phases $\phi = [0, 0.5\pi, \pi]$ at a constant amplitude of $A = 1$.

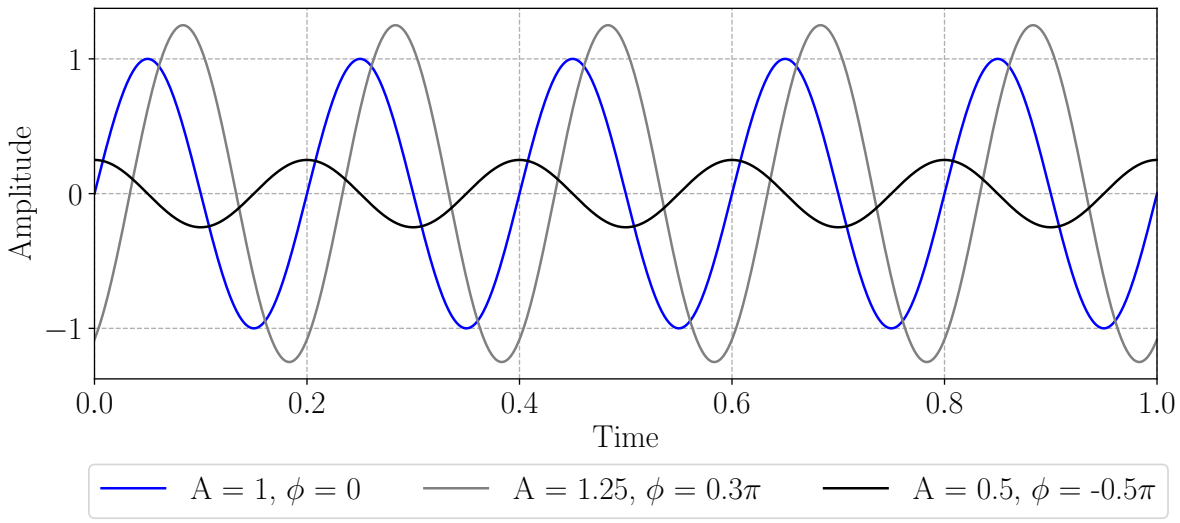


Figure 3.15: Sine signals representing three different amplitudes A = [0.5,1,1.25] and phases.

In Table 3.1, the computed error metrics for the considered sine signals with respect to the reference signal ($A = 0$, $\phi = 0$) are summarized. Considering the signals with a variation in the amplitude it is shown that an increase in amplitude results in a positive A_{err}, whereas a negative A_{err} with respect to the reference signal indicates a reduction in the amplitude. Depending on the magnitude of the A_{err}, the C_{err} changes accordingly. Further, the fit factor Q reduces with increasing C_{err}.

Based on the computed P_{err} for the signals with a phase variation it becomes clear that with increasing phase shift the P_{err} also increases. As already mentioned above, if $P_{err} = 1$, both signals are completely out of phase [110]. With increasing P_{err}, C_{err} also increases. In contrast to Q computed for the single amplitude variation, Q is represented by negative values if the signals are out of phase. With increasing P_{err} the fit factor becomes more negative, indicating the increasing out-of-phase behavior of both signals.

Focusing on the sine signal with both varying amplitude and phase it is shown that C_{err} increases with increasing A_{err} and P_{err}. Since the signals are phase shifted with respect to the reference signal, Q is defined by negative values.

Amplitude A	Phase ϕ	A_{err}	P_{err}	C_{err}	Q
1.25	0	0.16	0	0.14	75%
0.5	0	-0.39	0	0.35	50%
1	0.5π	0	0.5	0.44	-41%
1	π	0	1	0.88	-100%
1.25	0.3π	0.16	0.33	0.32	-14%
0.51	-0.5π	-0.67	0.5	0.74	-3%

Table 3.1: Amplitude-, phase-, comprehensive error and fit factor between sine signals representing different amplitudes and phases.

4 Airfoil Buffet Prediction

In this chapter, a LSTM neural network architecture as introduced in Section 3.3 is applied for the prediction of aerodynamic forces and moments at transonic buffet condition. As a test case, the NACA0012 airfoil is chosen, which is excited by forced pitch and simultaneous pitch and plunge motions beyond the critical buffet angle of attack. In this regard, the training and application procedures of the LSTM-based NN are presented. The content of this chapter is mainly based on the author's publications, see references [153] and [149].

In the first section, the geometric properties of the NACA0012 airfoil are defined and a brief analysis of the buffet flow characteristics of the test case are given. The second section covers the numerical setup of the conducted URANS simulations for the training, validation and test data sets. Subsequent, the generation of the CFD-based training and validation data set is discussed in Section 4.3, followed by a description of the training and validation procedure (see Section 4.4). In Section 4.5 and Section 4.6, the prediction results of the LSTM-ROM are presented. In this regard, the LSTM is used for the prediction of buffet forces and moments due to harmonic motions, covering different reduced frequencies and amplitudes. The results of the trained LSTM are compared to the full-order reference CFD solutions. The chapter is concluded with a comparison of the computational effort of the full-order reference CFD simulations and the training and application of the LSTM-ROM.

4.1 Test Case: NACA0012 Airfoil

For demonstrating the performance of the LSTM neural network, the NACA0012 airfoil at buffet condition is chosen. According to a numerical study by Raveh [101], the buffet condition of the NACA0012 airfoil is defined by a freestream Mach number of $Ma_\infty = 0.72$, a Reynolds number of $Re = 10^7$ and an angle of attack of $\alpha = 6°$. The geometrical properties of the NACA0012 airfoil are defined by a chord length of $c_{ref} = 1$ m, whereas the axis for the considered pitch and plunge motion is set at 25% of the chord length of the airfoil.

In Figure 4.1, the flow field of the NACA0012 airfoil at buffet condition is visualized by means of Mach number contour plots. Here, four timesteps within a single buffet period T_{Buffet} are shown. In addition, the distribution of the pressure coefficient (c_p) for each respective timestep of the buffet cycle is illustrated. As shown in Figure 4.1, the selected flow condition leads to a cyclic change in the shock position. Consistent with studies by Raghunathan et al. [98] and Iovnovich and Raveh [53], the formation of the shock starts almost at maximum thickness ($\approx 0.3\ x/l$) of the airfoil. By moving fore and aft within a buffet cycle, the shock position varies approximately between 20% and 40% of the airfoil chord length. In addition, the shock intensity only slightly varies around $\Delta c_p = 0.6 - 0.8$ within one buffet period.

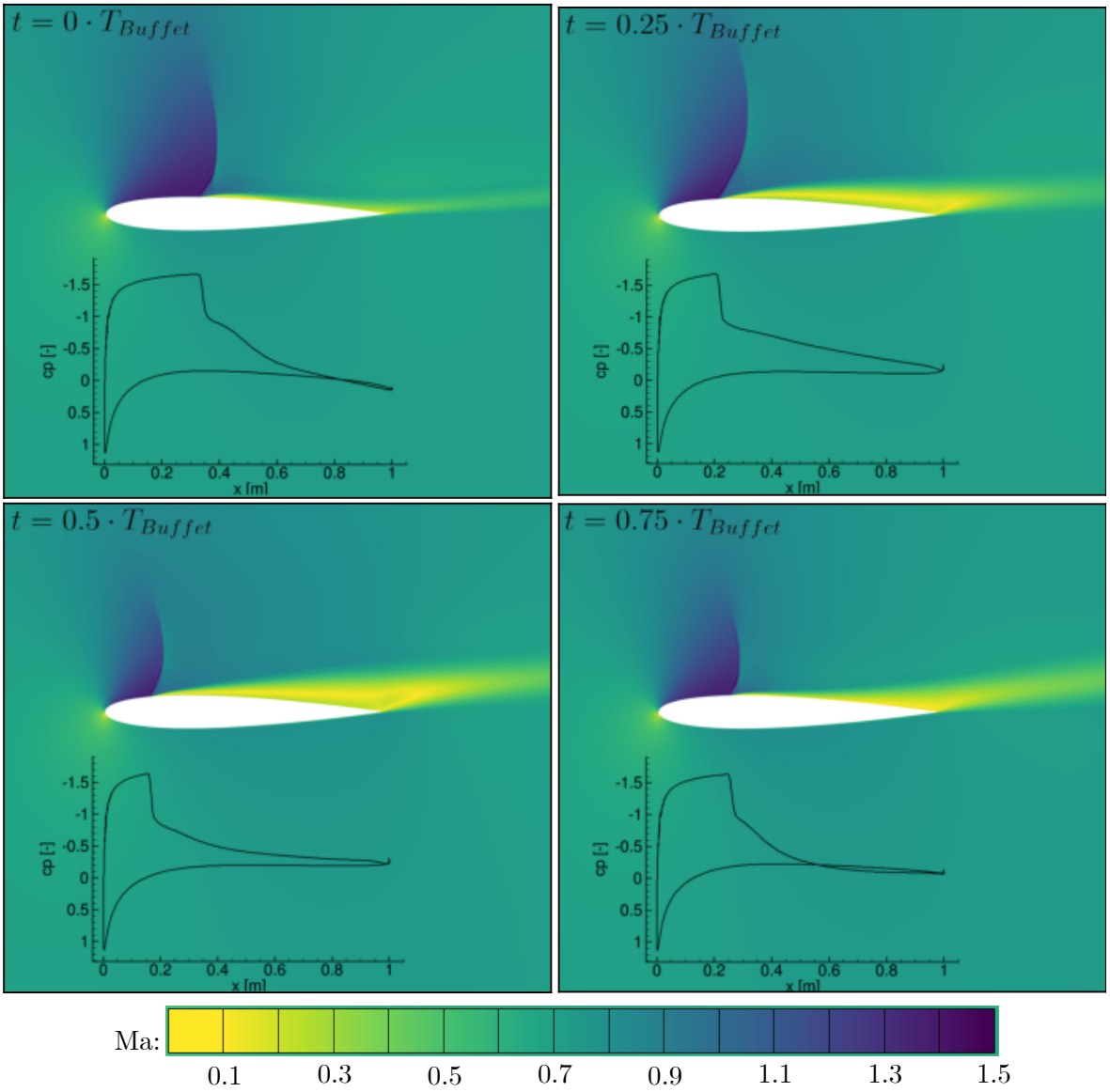


Figure 4.1: Mach number contour plots showing the buffet cycle of the NACA0012 airfoil ($Ma_\infty = 0.72$, Re $= 10^7$, $\alpha = 6°$, no external excitation, URANS, SA turbulence model). In addition, the corresponding distribution of the pressure coefficient (c_p) for each timestep is visualized. T_{Buffet} refers to the buffet period.

Due to the movement of the shock, the boundary layer downstream of the shock separates and exhibits a periodic thickening and thinning. At the beginning of the buffet period, only a small boundary layer separation is visible. With the shock moving towards the leading edge (LE) of the airfoil, the separation intensity increases and the area of separation extends towards the trailing edge (TE). As the shock travels back downstream, the area of separated flow at the TE decreases. By examining the corresponding response of the lift coefficient, the characteristic reduced frequency of the buffet condition of the NACA0012 airfoil is identified as $k_{red,Buffet} = 0.43$.

4.2 Computational Setup

In order to capture the buffet characteristic shock motion and boundary layer separation, the numerical simulations of the NACA0012 airfoil are conducted using an URANS approach. In the present investigation, the URANS simulations are performed using the Triangular Adaptive Upwind (TAU) [19] flow solver developed by the German Aerospace Center (DLR). With regard to the computation of the training, validation and test data sets, the same computational setup is selected.

Using TAU, the URANS equations are solved in conservation form using a shock-capturing finite volume scheme. Further, a dual grid approach is used for the unstructured mesh. In the present investigation, the temporal integration is performed using a backward Euler implicit scheme, while the embedded pseudo-time solution is computed by means of a lower-upper symmetric Gauss-Seidel (LU-SGS) algorithm. In addition, a multigrid approach is applied to accelerate convergence. The spatial discretization is accomplished by a central scheme with matrix dissipation. For the discretization of the convective fluxes, a second-order central scheme is used, whereas the reconstruction of the gradients is accomplished by means of a Green Gauss scheme.

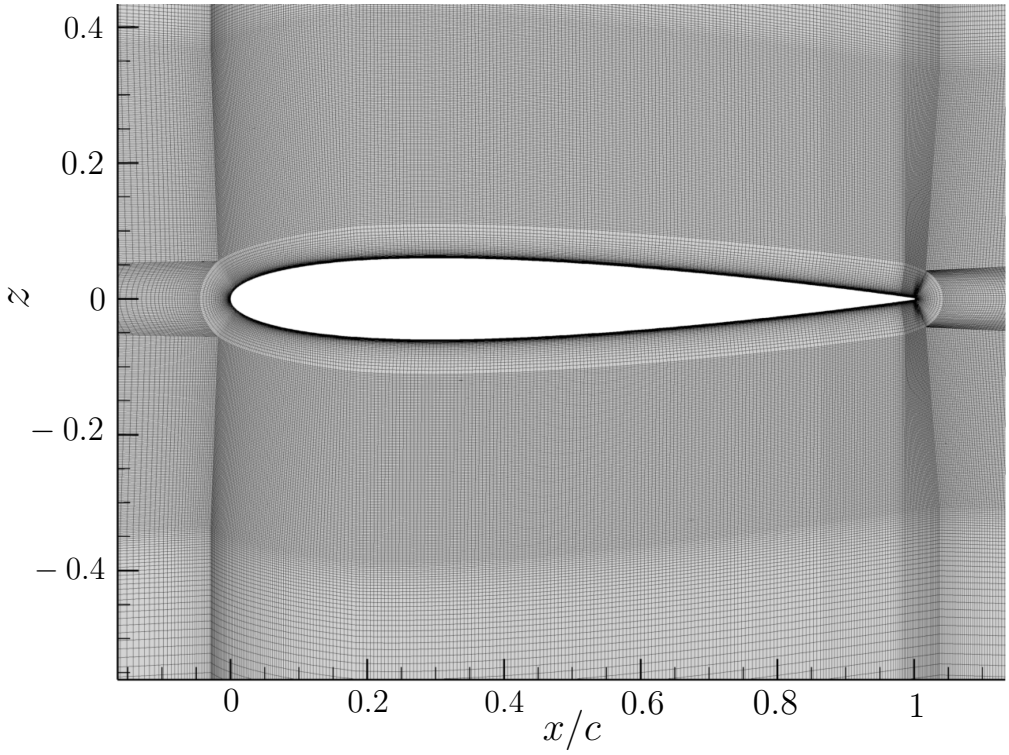


Figure 4.2: Geometry of the NACA0012 airfoil embedded within a block-structured grid for the CFD-based simulations.

Following the numerical studies by Raveh [101] and Barakos and Drikakis [4], the SA model is applied for turbulence modeling. More specifically, the SA model with Edwards modification [28] is used in order to improve the near-wall resolution.

The mesh deformation for the respective single and simultaneous pitch and plunge motion of the NACA0012 airfoil is implemented by the application of a RBF-based grid deformation using a TAU-Python interface.

With respect to the viscid CFD simulations, a block-structured computational grid applied in a previous study by Winter and Breitsamter [142], is used. The grid has been generated using the commercial meshing software ANSYS ICEM CFD [2]. The computational grid as shown in Figure 4.2 is composed of $8.8 \cdot 10^5$ elements arranged in an O-type topology. The airfoil surface is resolved with 500 cells and the height of the first surface cell is defined to $5 \cdot 10^{-4}$ m in order to define a dimensionless wall distance of $y^+ < 1$.

In order to ensure the independence of the solution from the grid resolution, a steady as well as an unsteady grid sensitivity study are performed. Therefore, five grid levels with the respective halved and doubled number of edge nodes relative to the basis grid are considered. In Figure 4.3, the time-series of the lift coefficient at a developed buffet condition is presented for the applied grid levels. Since the relative error between the grid levels including $1.7 \cdot 10^6$ and $8.8 \cdot 10^5$ number of cells is given as 0.13% and 0.12% with respect to the minimum and maximum amplitude of C_L, the grid with $8.8 \cdot 10^5$ elements is chosen to be adequate for the following computations. In addition, a steady state grid study is performed, resulting in the same sufficient grid resolution. Based on an additional convergence study, the physical timestep for the simulation is defined by $\Delta t = 0.0005$ s, which corresponds to a nondimensional timestep ($\tau = (t \cdot U_{\infty,Buffet}) \,/\, c_{ref}$)) of $\Delta\tau = 0.11$. Here, $U_{\infty,Buffet}$ represents the freestream velocity at buffet condition, which is defined as $U_{\infty,Buffet} = 238$ m/s.

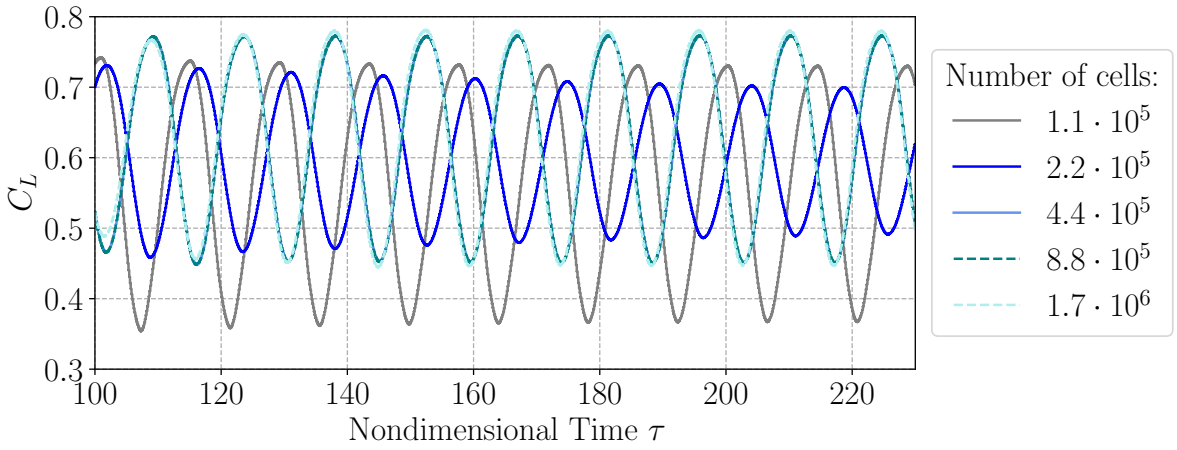


Figure 4.3: Results of the grid sensitivity study based on the time series of the unsteady lift coefficient C_L at a developed buffet condition.

4.3 CFD-Based Data Set Generation

For the generation of a sufficient training, validation and test data set, forced motion CFD simulations are performed, in order to model the pitch and plunge deflection of the NACA0012 airfoil at buffet condition. The resulting aerodynamic response of the airfoil is recorded by means of the lift (C_L) and pitching moment coefficient (C_{My}).

The excitation of the aerodynamic system is accomplished by several smoothed amplitude modulated pseudo random binary signals (SAPRBS) [88], which are generated for a single and simultaneous excitation of the pitch and plunge degree of freedom (DoF). This type of signal is chosen for the following investigation, since it is characterized by a high information content per signal length. Hence, it is able to cover a large range of various frequencies and amplitudes [88, 136]. This aspect is in particular important concerning the limitations of computational time and costs.

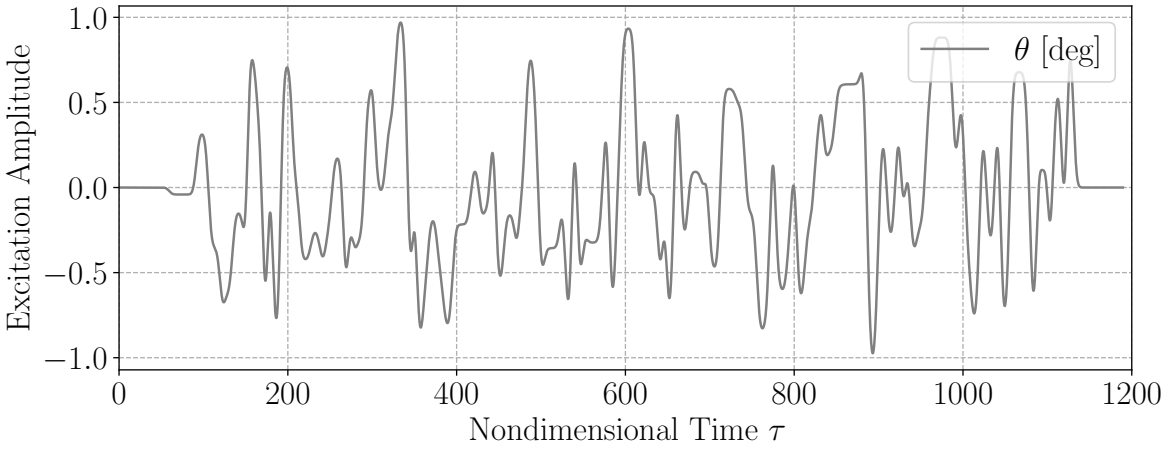


Figure 4.4: SAPRBS for the prescribed excitation of the pitch degree of freedom (NACA0012, $\Delta\tau = 0.11$).

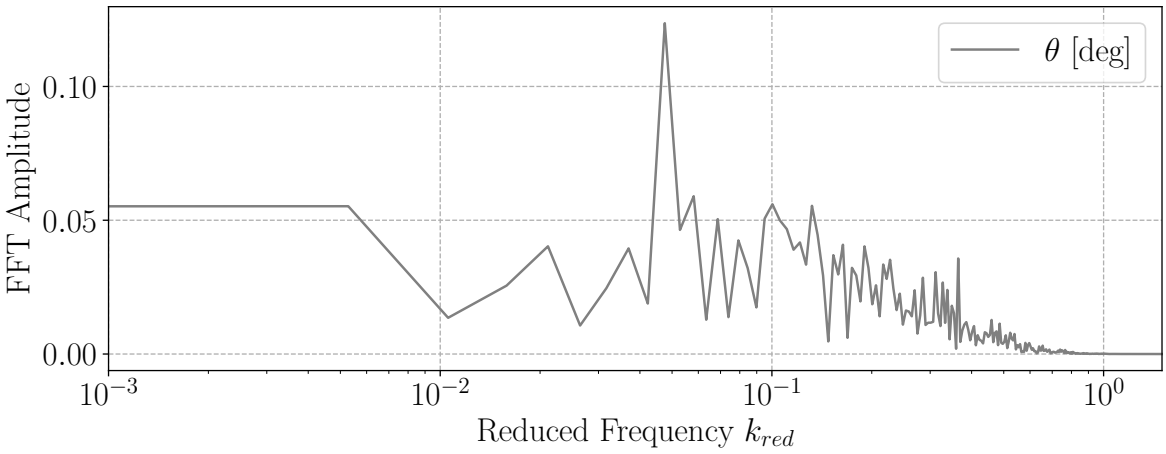


Figure 4.5: Amplitude spectrum of the SAPRBS for the prescribed excitation of the pitch degree of freedom (NACA0012, $\Delta\tau = 0.11$).

Considering only the excitation of the pitch DoF, ten different SAPRBS including 10000 data points each, are randomly generated. Therefore, the overall number of available data points for training the LSTM is summed up to $1 \cdot 10^5$. Since all signals are smoothed, the minimum and maximum pitch amplitude are defined as $\theta_{min,max} = \pm 1°$. For smoothing the signals, Gauss filtering as described in Winter and Breitsamter [136, 141], is applied. In Figure 4.4, one of the applied SAPRBS is visualized. Therefore, the excitation amplitude θ is plotted over the nondimensional time τ. In addition to the time-domain representation, the amplitude spectrum of the SAPRBS shown in Figure 4.4 is visualized in Figure 4.5. As shown, the plateaus of the signal are designed in order to cover reduced frequencies in the range from $0 < k_{red} < 1$.

In contrast, considering the excitation of both the pitch and plunge DoFs, three different SAPRBS are randomly defined, including 15000 time steps each. Analogous to the single pitch excitation, the pitch amplitude is limited to $\theta = \pm 1°$, whereas the plunge amplitude is defined as $h = \pm 1 \cdot c_{ref}$. In Figure 4.6, the time-domain response of one of the three applied SAPRBS for the simultaneous excitation of the pitch and plunge motion, is visualized. As shown, the first and last part of the SAPRBS are constructed for a separate excitation of the pitch and plunge DoF, respectively. In between these two time instances, an overlapped excitation of both DoFs is accomplished in order to capture the nonlinear interactions caused by a combined input. The separate excitation for the pitch and plunge DoF is defined by 6000 time steps each, whereas the combined excitation includes 4500 time steps.

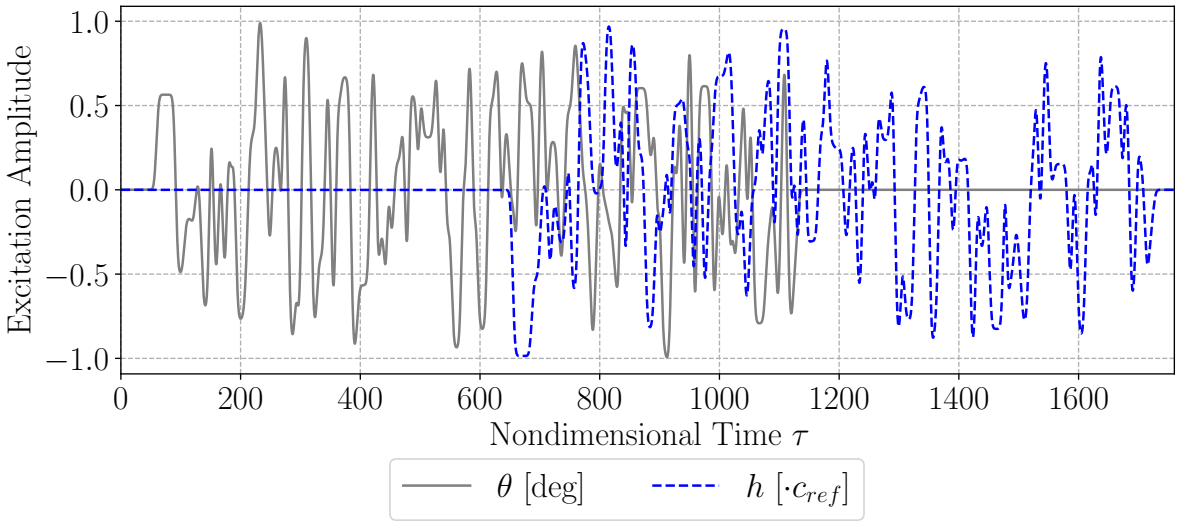


Figure 4.6: SAPRBS for the prescribed excitation of the pitch and plunge degrees of freedom (NACA0012, $\Delta\tau = 0.11$).

The amplitude spectra of the SAPRBS shown in Figure 4.6 are visualized in Figure 4.7. As it is the case for the SAPRBS for the pitching motion, the signals are designed in order to represent reduced frequencies in the range from $0 < k_{red} < 1$.

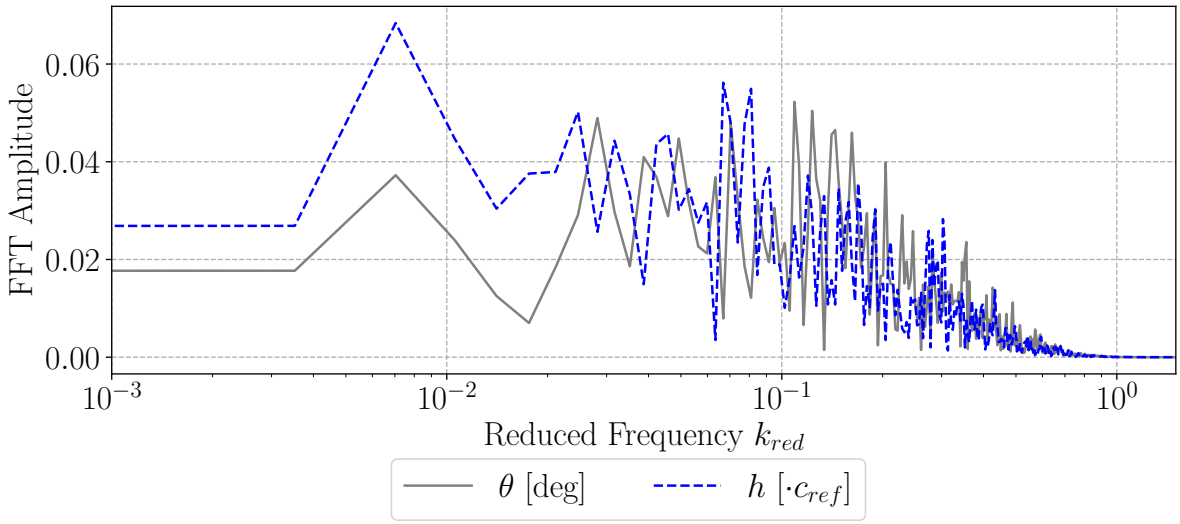


Figure 4.7: Amplitude spectra of the SAPRBS for the prescribed excitation of the pitch and plunge degrees of freedom (NACA0012, $\Delta\tau = 0.11$).

4.4 Training of the LSTM-ROM

Given the training data set including the buffet forces and moment characteristics of the pitch and plunge excited NACA0012 airfoil, the LSTM-ROM training is performed. Considering the simulated CFD data set in Section 4.3, the system inputs consist either of the pitch (θ) or the pitch and plunge (h) excitation amplitudes, whereas the system outputs are given by the lift and pitching moment coefficient. With t defining the time dependency of each amplitude, inputs and outputs of the LSTM can be defined as follows:

$$\begin{aligned} \boldsymbol{x}_t &= [\theta(t), (h(t))]^T \\ \boldsymbol{y}_t &= [C_L(t), C_{M_y}(t)]^T \end{aligned} \tag{4.1}$$

Prior to the training process, the available CFD data set is segmented. Considering the application case including the individual excitation of the pitch DoF, seven out of the ten simulated SAPRBS are used for training, whereas the remaining three are used for validation. Hence, 70% of the data points are chosen for the training of the ROM, whereas the remaining percentage is used for validation purposes. Considering the application of the LSTM for the computation of harmonic motions covering individual and simultaneous pitch and plunge motions, two of the three available SAPRBS are applied for training. Therefore, 66% of the data is applied for training, whereas 33% are used as the validation data set.

For both individual training processes of the LSTM neural network, several hyperparameters are identified based on a detailed parameter study. In the following, all parameters of each LSTM neural network are summarized.

Therefore, $\text{LSTM}_{\text{single}}$ is referred to the LSTM-ROM applied for modeling harmonic pitch motions, whereas $\text{LSTM}_{\text{combined}}$ is defined as the LSTM-ROM applied for harmonic pitch and plunge motions:

- $\text{LSTM}_{\text{single}}$: The number of hidden layers is varied between two and four, however, three hidden layers are chosen for the following application. A two-layered LSTM indicates a lower prediction quality, whereas the implementation of four hidden layers intensively increases the training time. For each of the three selected hidden layers, the following number of neurons are tested: [128, 256, 512]. Since the implementation of 512 neurons results in a training time increase without any considerable performance improvement, the number of neurons per layer is set to 256. The initial learning rate for the optimizer is defined as $\eta = 1 \cdot 10^{-4}$. An increase of the learning rate to $\eta = 1 \cdot 10^{-3}$ results in an acceleration of convergence, however, the training performance decreases. Reducing the learning rate to $\eta = 1 \cdot 10^{-5}$ leads to a deceleration of convergence and therefore a considerable increase in training time. The size of the input sequence has been varied between 32, 64, 128 and 256. Based on the results the signals are divided in sequences including 128 timesteps each, with a batch size defined as one.

- $\text{LSTM}_{\text{combined}}$: Analogous to the $\text{LSTM}_{\text{single}}$, three hidden layers have been implemented, including 256 neurons each. Further, the initial learning rate is set to $\eta = 1 \cdot 10^{-4}$. The batch size is defined as one, with each sequence in the batch including 128 timesteps.

In Table 4.1, the final hyperparameters for both LSTM-ROMs are summarized.

	$\text{LSTM}_{\text{single}}$	$\text{LSTM}_{\text{combined}}$
Hidden layers	3	3
Neurons per layer	256	256
Batch size	1	1
Sequence length	128	128
Initial learning rate	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
State activation	$tanh$	$tanh$
Gate activation	σ	σ

Table 4.1: Hyperparameters of the $\text{LSTM}_{\text{single}}$ and $\text{LSTM}_{\text{combined}}$.

As the state activation function, a $tanh$ is selected, while the gate activation function is chosen to be the sigmoid function σ. In Figure 4.8, the convergence trends of the training and validation losses of both LSTM neural networks are shown. Therefore, the decrease in MSE (see Equation 3.13) is plotted over the number of epochs. As shown, the training process of the $\text{LSTM}_{\text{single}}$-ROM is terminated after 1500 epochs, whereas the $\text{LSTM}_{\text{combined}}$ is trained for 800 epochs. Examining the convergence of the validation losses, no overfitting occurs during the training process.

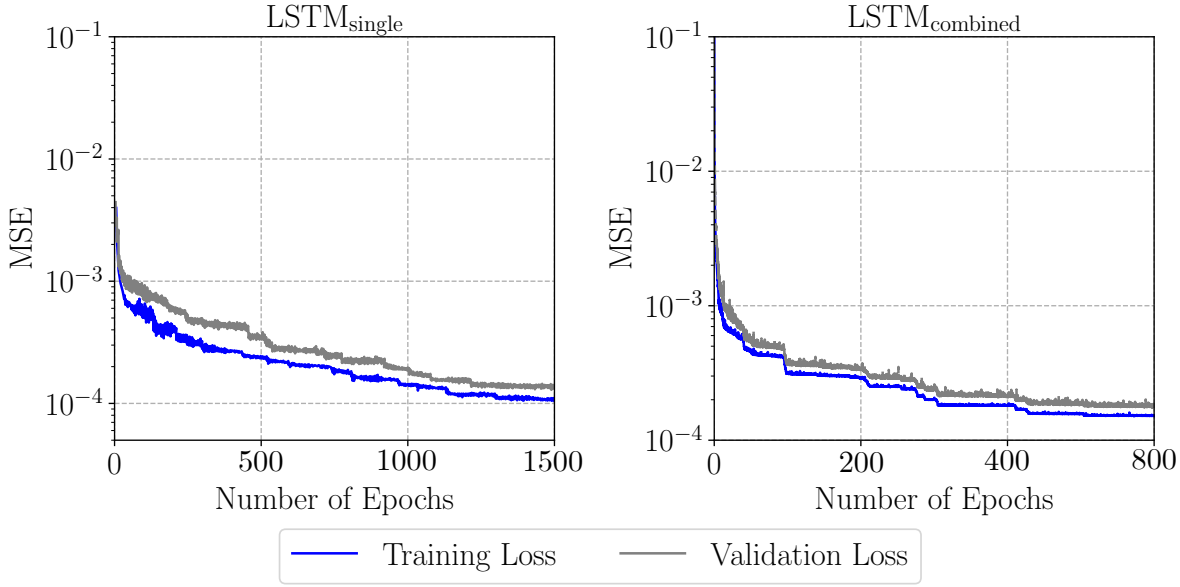


Figure 4.8: Convergence trends of training and validation losses of the LSTM$_{\text{single}}$ (left) and LSTM$_{\text{combined}}$ (right).

Multi-Step Predictions

Prior to the application of the trained LSTM-ROMs to a test data set, the trained models are evaluated based on the validation data set by performing recurrent multi-step predictions. For the initialization of the multi-step mode, the first 32 time steps of the SAPRBS response are provided to the trained LSTM. As the prediction proceeds, the numerical data is successively substituted by the data points predicted by the LSTM.

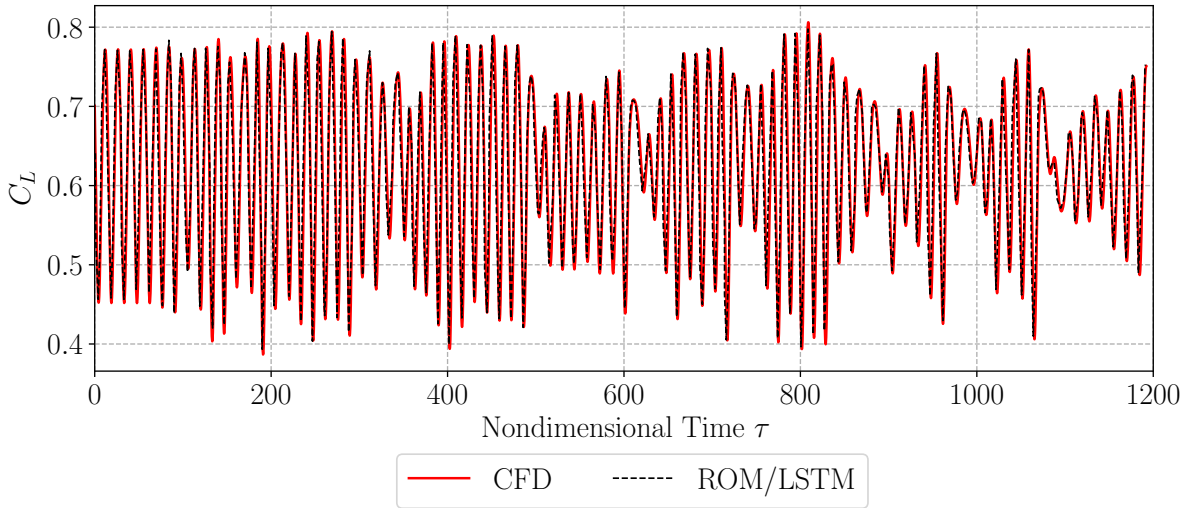


Figure 4.9: Lift coefficient response induced by a validation SAPBRS. The simulation result of the LSTM-ROM is compared to the CFD reference solution (NACA0012, $Ma_\infty = 0.72$, Re $= 10^7$, $\alpha = 6°$).

In the following, only the results of the trained LSTM$_{\text{single}}$ are presented. Therefore, the aerodynamic responses due to one of the three SAPBRS applied for validation,

are considered. In Figure 4.9 and Figure 4.10, the lift and pitching moment coefficient trends as obtained by the LSTM compared to the CFD reference solution, are shown, respectively.

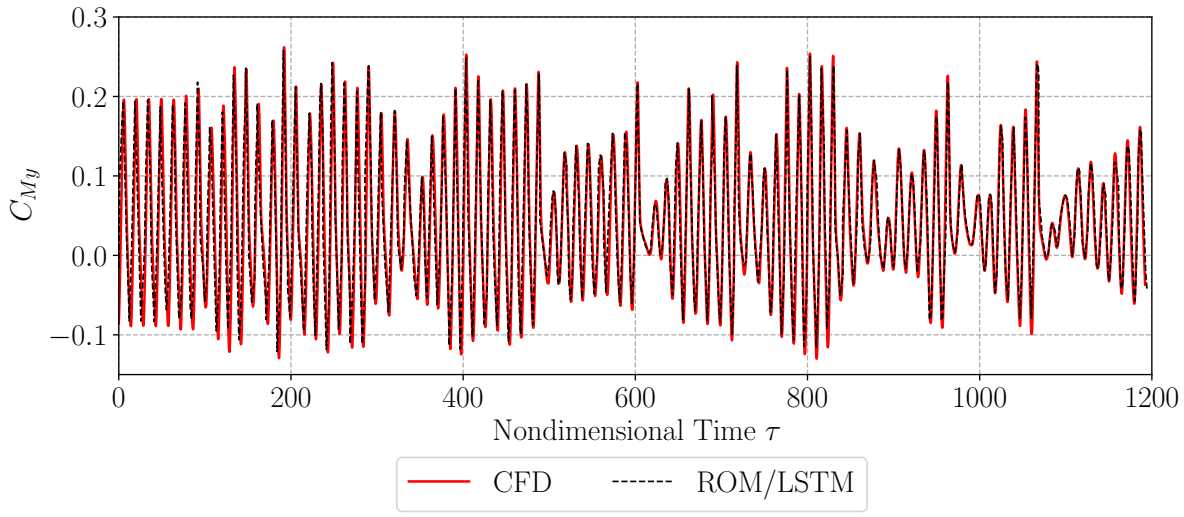


Figure 4.10: Pitching moment coefficient response induced by a validation SAPBRS. The simulation result of the LSTM-ROM is compared to the CFD reference solution (NACA0012, $Ma_\infty = 0.72$, Re $= 10^7$, $\alpha = 6°$).

In addition to the multi-step predictions using the validation data set, the trained $\text{LSTM}_{\text{single}}$-ROM is applied to reproduce the buffet characteristics in the absence of any external excitation of the airfoil. In Figure 4.11, the resulting lift coefficient response as obtained by the LSTM compared to the reference CFD solution, is shown. As indicated, the LSTM is able to accurately capture the lift coefficient response at buffet condition.

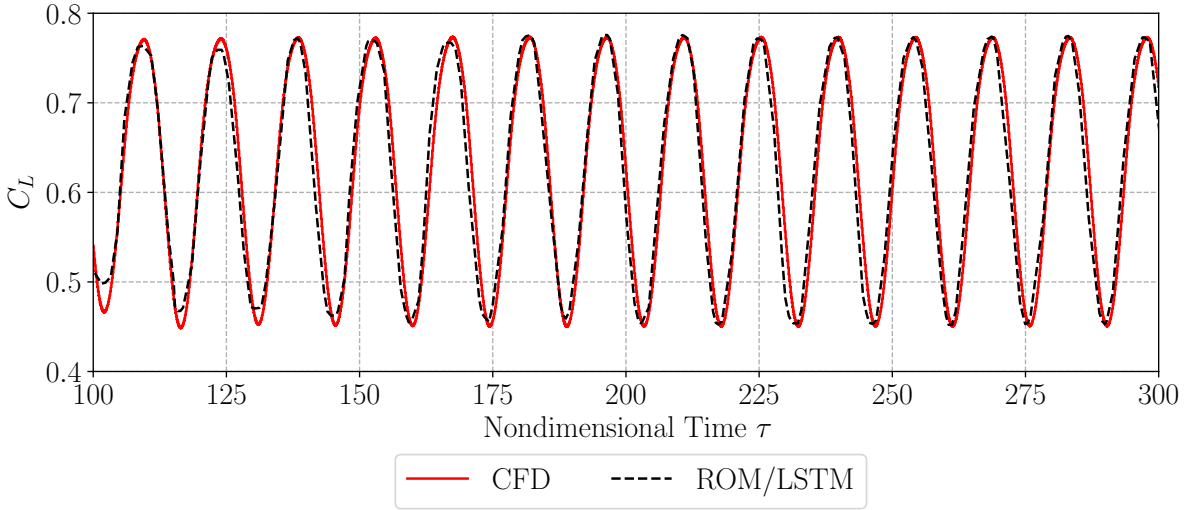


Figure 4.11: Lift coefficient response at buffet condition. The result of the LSTM-ROM is compared to the CFD reference solution (NACA0012, $Ma_\infty = 0.72$, Re $= 10^7$, $\alpha = 6°$, no excitation).

In order to quantify the prediction performance of the $\text{LSTM}_{\text{single}}$, the evaluation metrics introduced in Section 3.7, are applied. In Table 4.2, the errors for both the validation data set and buffet condition application, are summarized. Based on the computed errors it is shown that the error in amplitude and phase shift is rather small, resulting in an overall fit of up to 97%.

Prediction mode	Aerodynamic coefficient	A_{err}	P_{err}	C_{err}	Q
Validation data	C_L	-0.023	0.012	0.023	97%
	C_{My}	-0.028	0.016	0.028	96%
Buffet condition	C_L	-0.019	0.027	0.029	95.5%

Table 4.2: Evaluation metrics computed for the validation data set and buffet condition prediction ($\text{LSTM}_{\text{single}}$-ROM).

4.5 Performance Evaluation: Single Sinusoidal Excitation

Following the application of the trained LSTM-ROM on the validation data set, the final capability test is conduced by the prediction of time-series trends of lift and pitching moment coefficient due to harmonic pitch motions. Further, the focus of the ROM application is on the representation of the lock-in effect (see Section 2.2.2). Therefore, harmonic excitations covering different frequencies and amplitudes, are considered.

4.5.1 Sinusoidal Excitation with Varying Frequencies

In a first step, the NACA0012 airfoil is excited with harmonic motions covering four different reduced frequencies: $k_{red,Ex} = [0.2,\ 0.4,\ 0.6,\ 0.8]$. Analogous to the excitation using the SAPRBS (see Figure 4.4), the pitch amplitude is defined as $\theta = \pm\ 1°$. The reference full-order CFD simulations for the considered harmonic motions are conducted using the TAU solver setup as described in Section 4.2. In order to guarantee a fully developed solution without any initial transient influence, 30 excitation periods are computed with both the CFD solver and the LSTM-ROM for each of the considered harmonic motions. Here, a fully developed buffet condition is applied for initialization of the harmonic computations. For the initialization of the recurrent multi-step predictions, a sequence of 32 time steps originated from the CFD solution is provided to the trained LSTM.

For a clear comparison and evaluation of the results, frequency-domain as well as time-domain responses are presented in the following. The frequency-domain responses are visualized using a fast-Fourier transformation (FFT) of the aerodynamic coefficients (C_L, C_{M_y}), plotted over the reduced frequency k_{red}. Evaluating the frequency- and time-domain responses of the lift coefficient (see Figure 4.12 and Figure 4.14), a nonlinear and

frequency-dependent interaction between the airfoil motion and the buffet instability is clearly indicated. This frequency dependent interaction is also visible in the selected frequency - and time-domain responses of the pitching moment coefficient, as shown in Figure 4.13 and Figure 4.15.

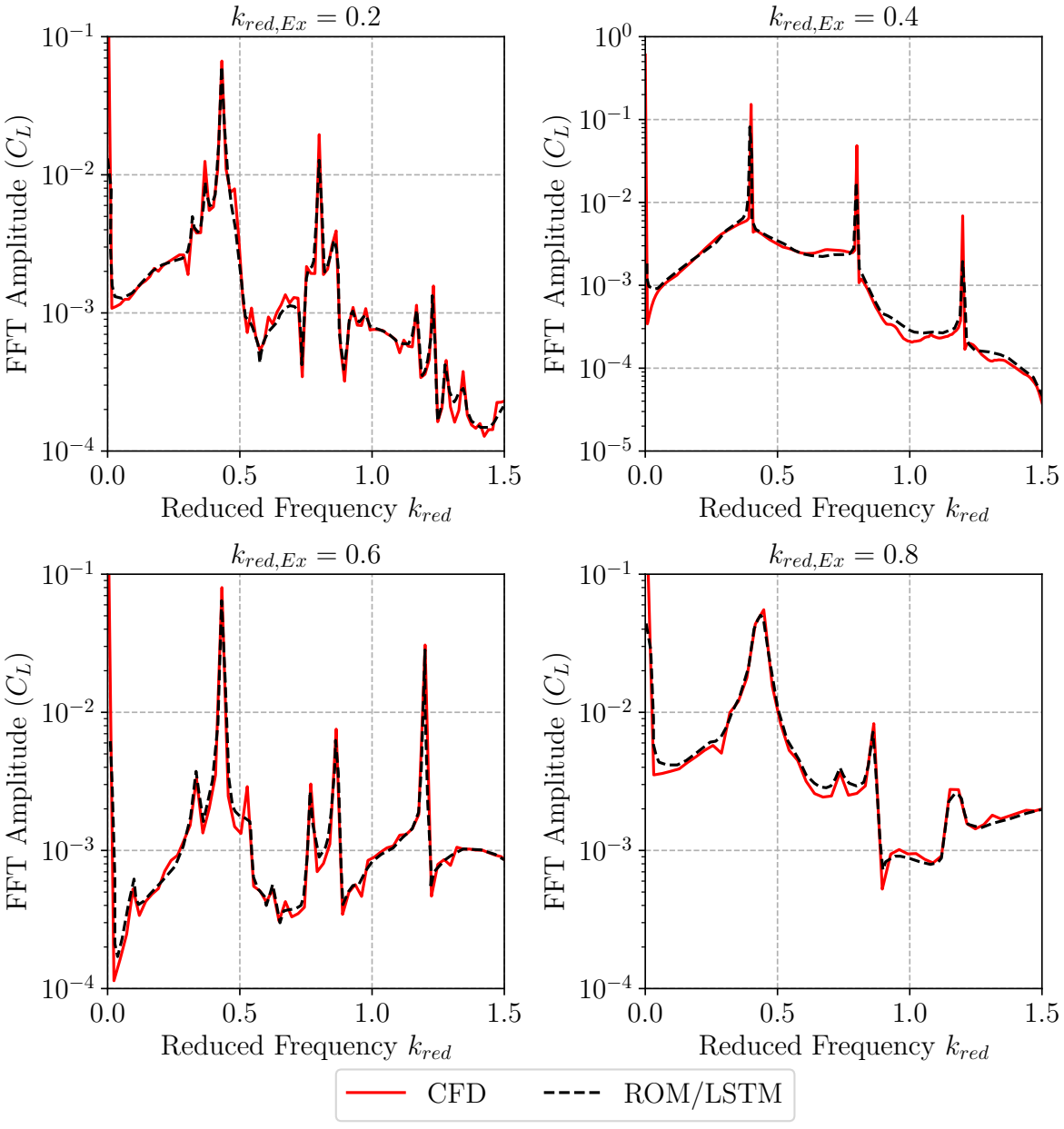


Figure 4.12: Frequency domain responses of the lift coefficient resulting from harmonic pitch motions with $k_{red,Ex} = [0.2, 0.4, 0.6, 0.8]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.72$, Re $= 10^7$, $\alpha = 6°$, $\theta = \pm 1°$).

Considering the reduced excitation frequency $k_{red,Ex} = 0.2$, which is smaller than the buffet frequency of the NACA0012 airfoil ($k_{red,Buffet} = 0.43$, without pitching motion), the buffet characteristics are to a certain extent disturbed by the pitch excitation.

In the corresponding spectra of the lift and pitching moment coefficient shown in Figure 4.12 and Figure 4.13, no peak at $k_{red,Ex} = 0.2$ is visible. However, a distinct peak representing the buffet frequency and a peak at $k_{red,Ex} = 0.8$ are indicated.

Examining the excitation with a reduced frequency $k_{red,Ex} = 0.4$, which almost represents the buffet frequency, the buffet flow response synchronizes with the pitch motion of the airfoil. Therefore, the presence of lock-in is clearly indicated in the frequency responses of the lift and pitching moment coefficient.

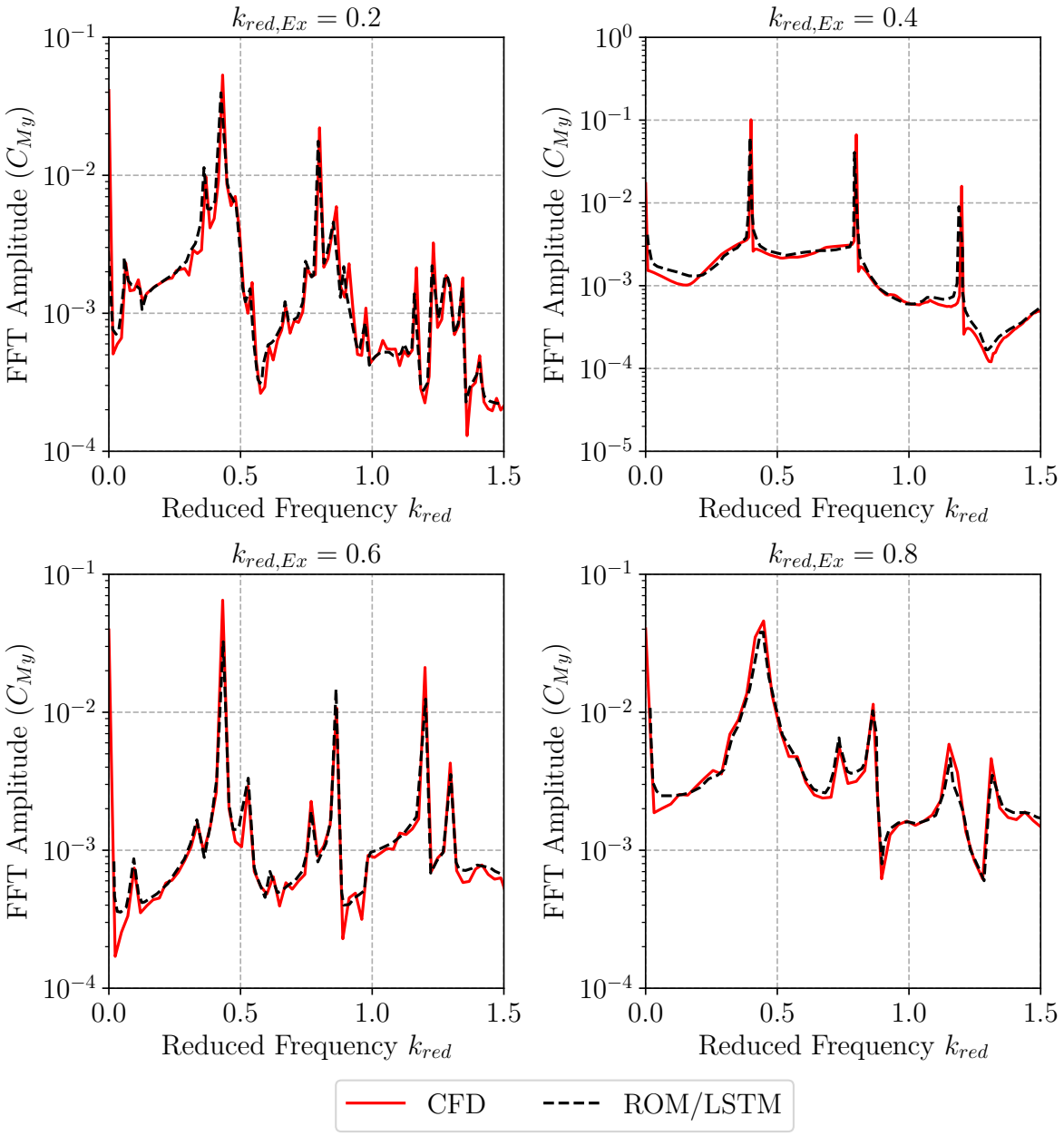


Figure 4.13: Frequency domain responses of the pitching moment coefficient resulting from harmonic pitch motions with $k_{red,Ex} = [0.2, 0.4, 0.6, 0.8]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.72$, $\mathrm{Re} = 10^7$, $\alpha = 6°$, $\theta = \pm 1°$).

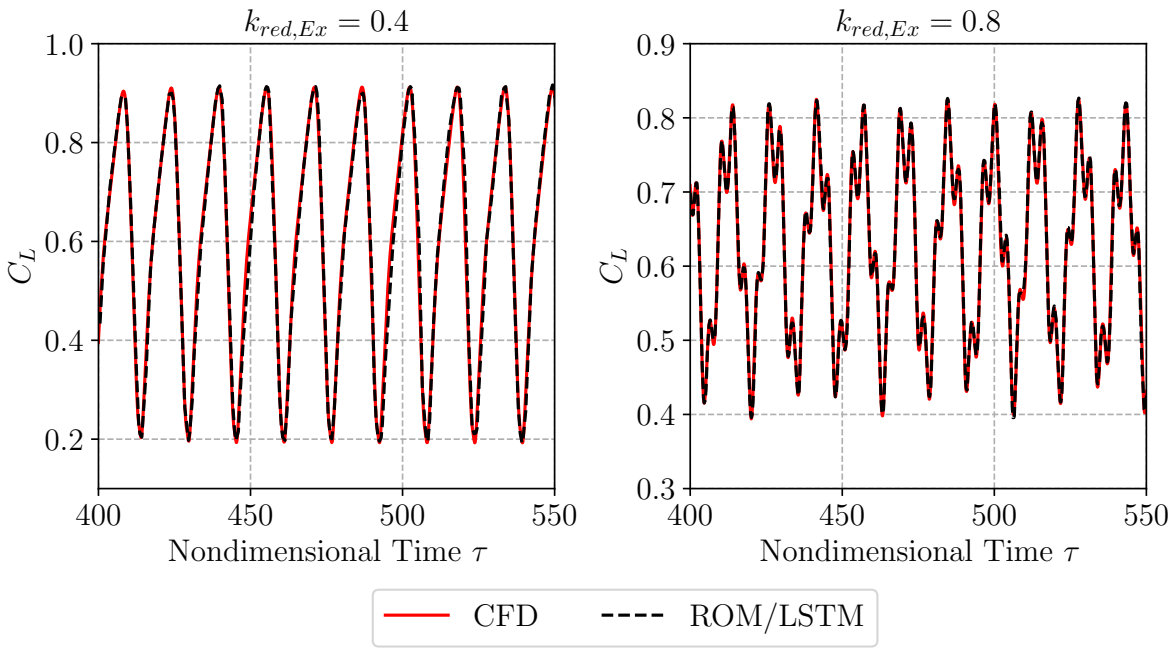


Figure 4.14: Time domain responses of the lift coefficient resulting from harmonic pitch motions with $k_{red,Ex} = [0.4, 0.8]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.72$, Re $= 10^7$, $\alpha = 6°$, $\theta = \pm 1°$).

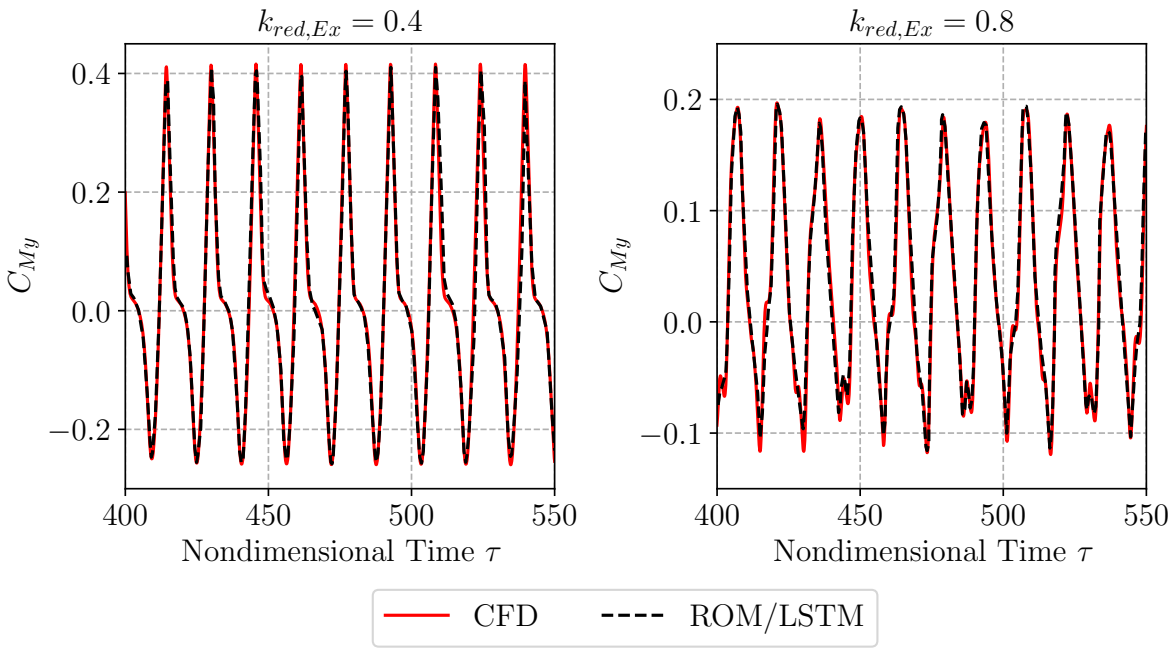


Figure 4.15: Time domain responses of the pitching moment coefficient resulting from harmonic pitch motions with $k_{red,Ex} = [0.4, 0.8]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.72$, Re $= 10^7$, $\alpha = 6°$, $\theta = \pm 1°$).

Further, the time-domain responses of the aerodynamic coefficients (see Figure 4.14 and Figure 4.15) also clearly show a harmonic response for $k_{red,Ex} = 0.4$. Consequently, the influence of the buffet frequency is considerably smaller compared to the other frequency cases and the system is predominantly influenced by the pitch motion of the airfoil. Considering the harmonic motion test cases with frequencies above the buffet frequency ($k_{red,Ex} = [0.6, 0.8]$), effects due to the buffet itself as well as the pitch excitation are again clearly differentiable.

Focusing on the harmonic motion test case with $k_{red,Ex} = 0.6$, only a small peak at $k_{red} = 0.6$ is visible in Figure 4.12 and Figure 4.13. However, as shown in the spectra of $k_{red,Ex} = 0.2$, a large peak around the buffet frequency and the second harmonic of the buffet frequency are visible. Further, a distinct peak at the second harmonic of the pitch excitation ($k_{red} = 1.2$) is indicated.

Compared to the test cases with $k_{red,Ex} = [0.2, 0.6]$, the test case with $k_{red,Ex} = 0.8$ shows a less significant interaction between both buffet and airfoil motion frequency responses. Further, the peak at buffet frequency appears to be more broadband compared to the other test cases. In addition, a peak at $k_{red} = 0.8$ is shown.

By comparing the responses of the trained LSTM with the reference CFD solution it is shown that the LSTM is able to capture the lift - and pitching moment trends due to the harmonic motions. Distinct peaks and their amplitudes in all spectra are correctly represented by the trained ROM. Further, the LSTM is able to reproduce the lock-in effect in the presence of external motions. In order to emphasize the performance of the LSTM for all considered test cases, the corresponding evaluation metrics are summarized in Table 4.3. Based on the computed errors it is shown that both the error in amplitude and phase is considerably small, resulting in a fit of about 92% to 93%.

Excitation Frequency ($k_{red,Ex}$)	Aerodynamic Coefficient	A_{err}	P_{err}	C_{err}	Q
0.2	C_L	-0.048	0.011	0.043	92.7%
	C_{My}	-0.041	0.013	0.038	93.3%
0.4	C_L	-0.053	0.013	0.048	92%
	C_{My}	-0.05	0.016	0.046	92.3%
0.6	C_L	-0.043	0.015	0.04	92.9%
	C_{My}	-0.051	0.017	0.047	92.1%
0.8	C_L	-0.048	0.013	0.044	92.5%
	C_{My}	-0.045	0.017	0.042	92.9%

Table 4.3: Amplitude, phase, comprehensive error and fit factor of the LSTM-ROM applied to harmonic pitch motions covering varying excitation frequencies.

4.5.2 Sinusoidal Excitation with Varying Amplitudes

Following the excitation of the NACA0012 airfoil considering different excitation frequencies, the trained LSTM-ROM is applied for the prediction of harmonic pitch motions covering varying amplitudes: $\theta = [\pm\ 0.1°, \pm\ 0.5°, \pm\ 1°]$. In addition, an excitation amplitude of $\theta = \pm\ 1.5°$ is considered in order to investigate the prediction performance of the LSTM-ROM outside the training range. For each harmonic motion, the excitation frequency is defined as the buffet frequency ($k_{red,Ex} = k_{red,Buffet} = 0.43$). Analogous to Section 4.5.1, 30 excitation periods are computed with the TAU solver and the LSTM for each selected harmonic motion. Further, for the start of the recurrent multi-step predictions, a batch size of 32 timesteps is defined.

Evaluating the frequency-domain responses of the lift and pitching moment coefficient, as shown in Figure 4.16 and Figure 4.17, a synchronization between the airfoil motion and the buffet frequency response is indicated. In all spectra, a distinct peak at the buffet frequency $k_{red,Buffet} = 0.43$ is visible. Further, second and third harmonic responses of the buffet frequency are indicated. Considering the responses due to harmonic excitations included in the range of the training amplitude ($\theta_{Train} = \pm\ 1°$), a good agreement between the CFD reference solution and the prediction of the LSTM-ROM is given. In contrast, considering the test case with an excitation amplitude of $\theta = \pm\ 1.5°$, the prediction seems to be a challenge for the trained LSTM. As shown, the frequency amplitudes as well as the phase of the lift and pitching moment coefficient responses are not correctly reproduced by the LSTM. These higher deviations in the responses are also highlighted by the amplitude and phase errors summarized in Table 4.4.

Considering the test cases covering the smaller excitation amplitudes, error values and an overall fit of about 92% similar to the results presented in Section 4.5.1 are given. In contrast, deviations of approximately 70% are indicated for the high amplitude test case. Therefore, the test case outside the training range clearly represents a challenge for the trained LSTM-ROM.

Excitation Amplitude (θ)	Aerodynamic Coefficient	A_{err}	P_{err}	C_{err}	Q
$\pm 0.1°$	C_L	-0.042	0.011	0.038	93.3%
	C_{My}	0.044	0.011	0.04	92.9%
$\pm 0.5°$	C_L	0.043	0.01	0.039	93%
	C_{My}	0.041	0.013	0.038	93.3%
$\pm 1°$	C_L	0.057	0.015	0.052	91.2%
	C_{My}	0.06	0.016	0.055	89.4%
$\pm 1.5°$	C_L	0.56	0.41	0.61	32%
	C_{My}	-0.61	0.44	0.66	28.6%

Table 4.4: Amplitude, phase, comprehensive error and fit factor of the LSTM-ROM applied to harmonic pitch motions covering varying excitation amplitudes.

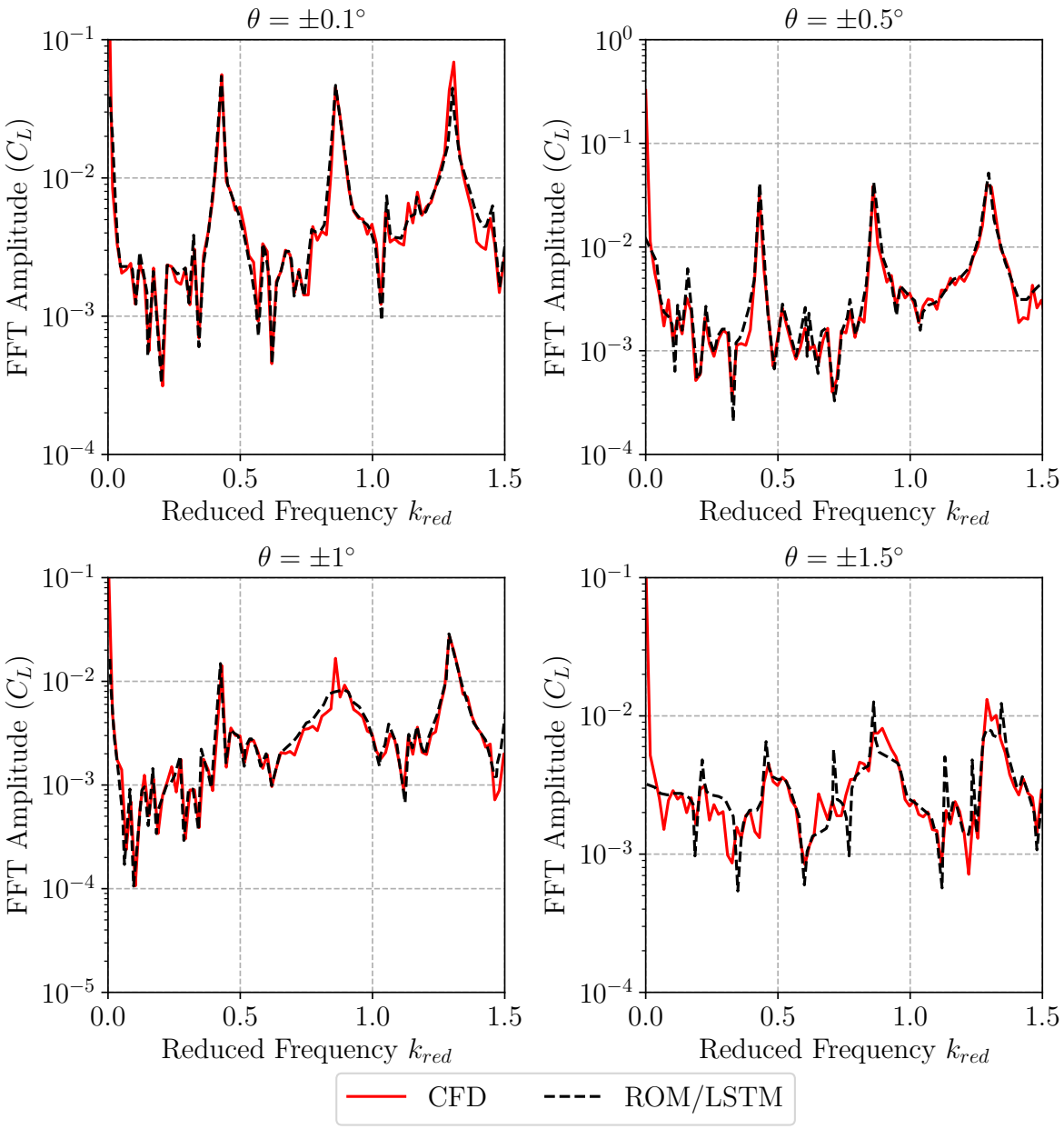


Figure 4.16: Frequency domain responses of the lift coefficient resulting from harmonic pitch motions with $\theta = [\pm 0.1°, \pm 0.5°, \pm 1°, \pm 1.5°]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.72$, $\mathrm{Re} = 10^7$, $\alpha = 6°$, $k_{red,Ex} = 0.43$).

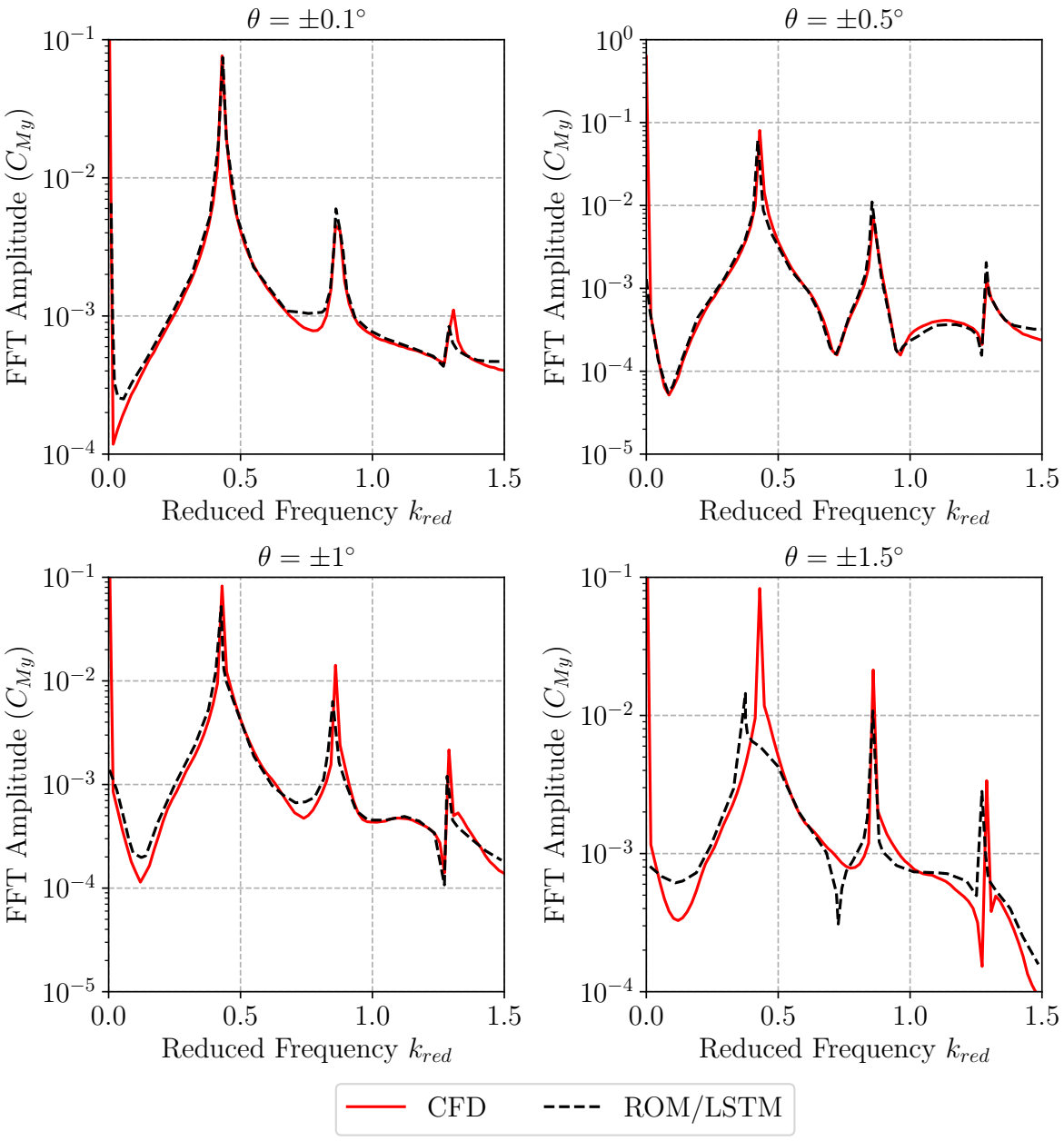


Figure 4.17: Frequency domain responses of the pitching moment coefficient resulting from harmonic pitch motions with $\theta = [\pm 0.1°, \pm 0.5°, \pm 1°, \pm 1.5°]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.72$, Re $= 10^7$, $\alpha = 6°$, $k_{red,Ex} = 0.43$).

4.6 Performance Evaluation: Combined Sinusoidal Excitation

Besides the prediction of coefficient time-series trends due to harmonic pitch motions, the trained $\text{LSTM}_{\text{combined}}$-ROM is applied for modeling time-series trends due to an individual and simultaneous pitch and plunge motion. Therefore, harmonic motions covering the same reduced excitation frequencies as in Section 4.5.1, are applied. In order to avoid any initial transient influence, 13 excitations periods are computed for each individual DoF with both the TAU solver and the trained LSTM. Further, the excitation period including a simultaneous excitation of the pitch and plunge DoF is represented by seven cycles. Analogous to the training signal, the pitching amplitude is defined as $\theta = \pm\ 1°$, however, the plunge motion is limited to $h = \pm\ 0.1 \cdot c_{ref}$ in order to ensure a stable numerical computation. In Figure 4.18, the signal for an individual and simultaneous excitation of both DoFs is exemplary shown, representing a reduced excitation frequency of $k_{red,Ex} = 0.2$.

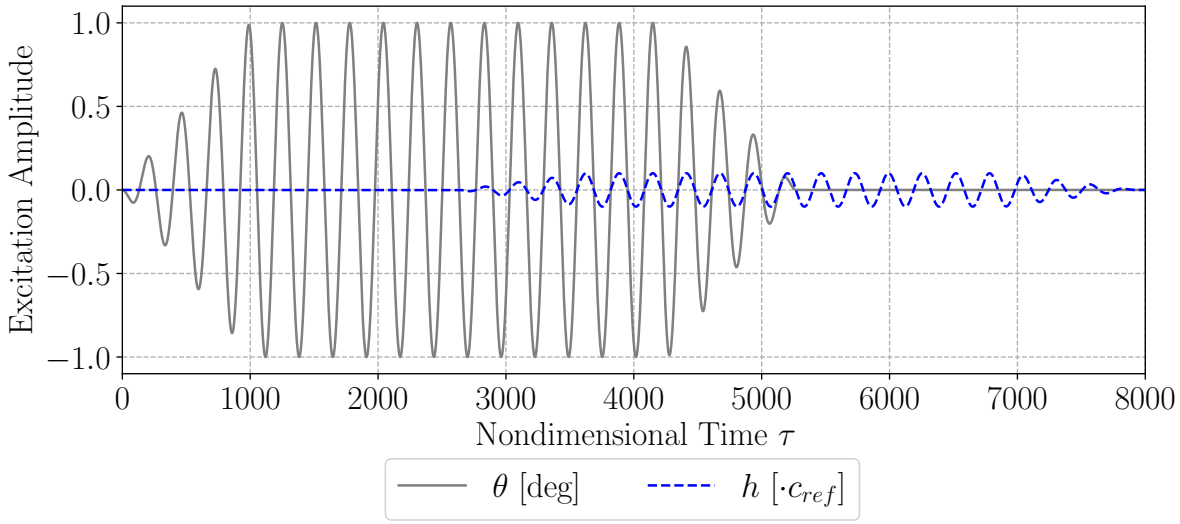


Figure 4.18: Harmonic signal for the prescribed excitation of the pitch and plunge DoF ($k_{red,Ex} = 0.2$).

As shown in Figure 4.18, both the excitation of the pitch and plunge DoF are defined by a ramp-up amplitude, in order to enable a stable numerical simulation.

Analogous to the previous section, both frequency-domain responses of the lift (see Figure 4.19) and pitching moment coefficient (see Figure 4.20) indicate a clear interaction between the responses of the buffet motion and the excitation of the airfoil. However, compared to the responses of the single pitch excitation, a more broadband range of frequencies is present in each of the spectra. Considering the response of the harmonic motion with $k_{red,Ex} = 0.2$, the spectra of both coefficients are dominated by the pitch and plunge excitation. Compared to the spectra shown in Figure 4.12 and Figure 4.13,

a distinct peak at $k_{red} = 0.2$ and $k_{red} = 0.6$ is indicated. Further, peaks at the buffet frequency and its second harmonic are present.

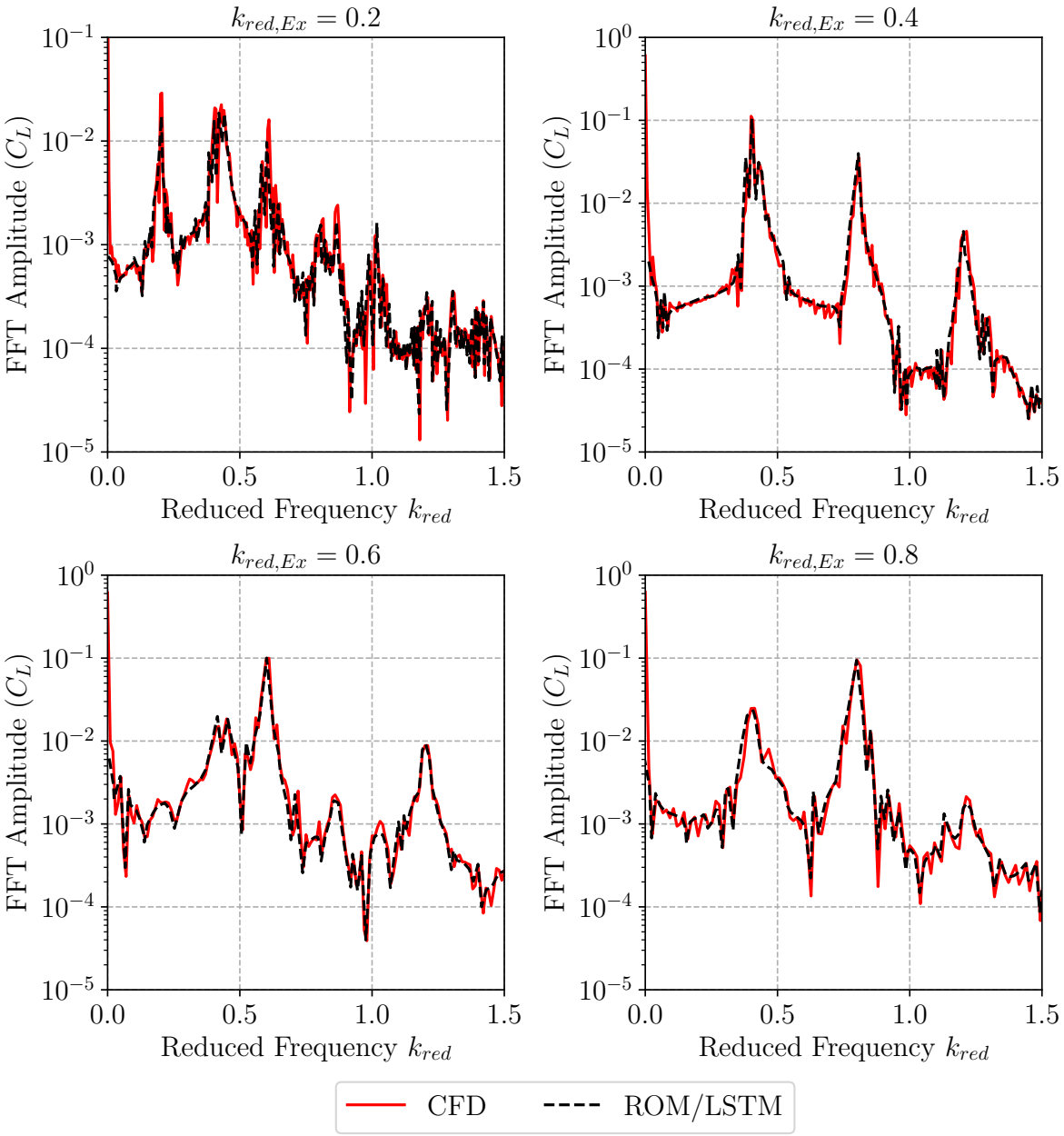


Figure 4.19: Frequency domain responses of the lift coefficient resulting from harmonic pitch and plunge motions with $k_{red,Ex} = [0.2, 0.4, 0.6, 0.8]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.72$, $\mathrm{Re} = 10^7$, $\alpha = 6°$, $\theta = \pm 1°$, $h = \pm 0.1 \cdot c_{ref}$).

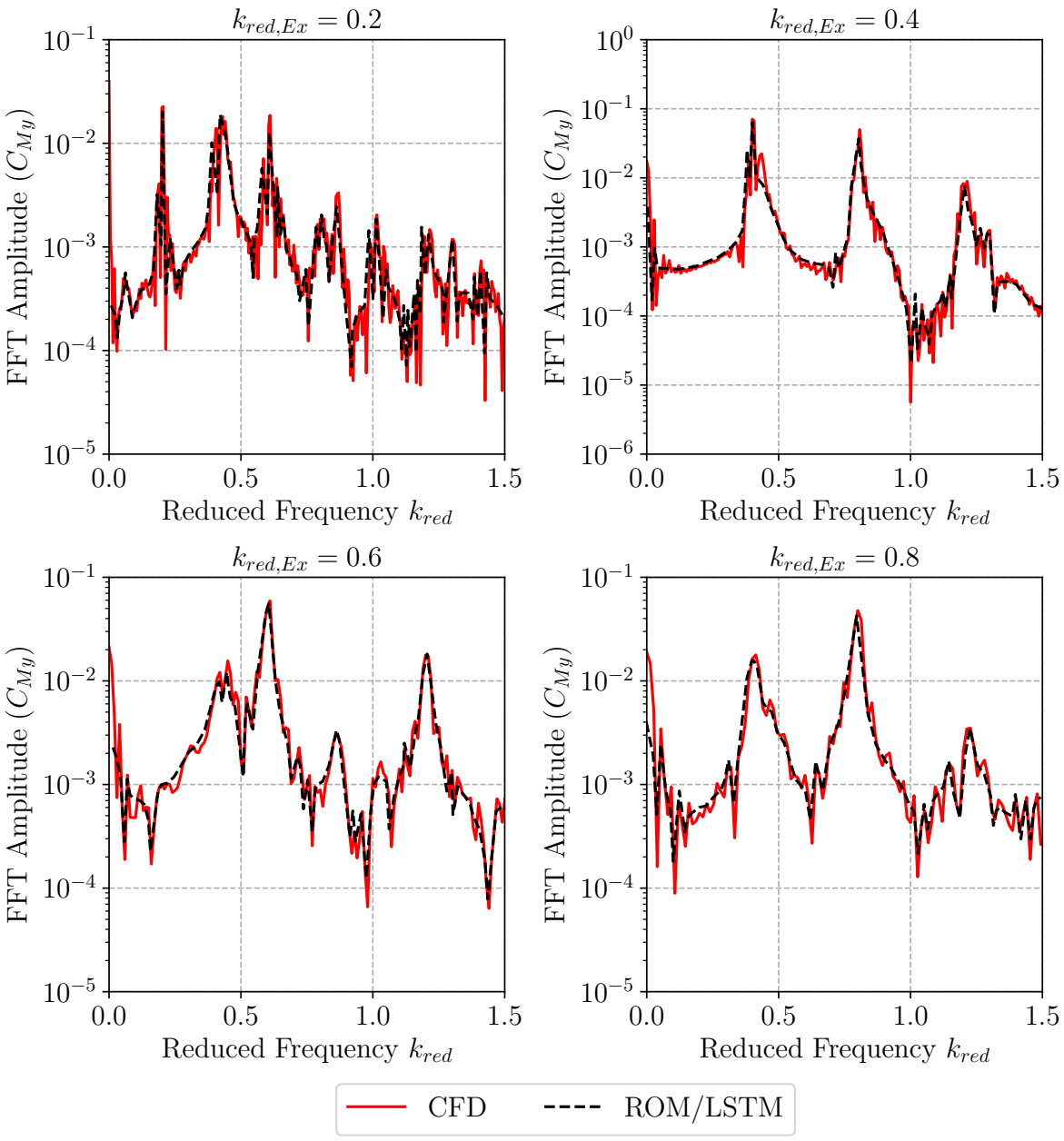


Figure 4.20: Frequency domain responses of the pitching moment coefficient resulting from harmonic pitch and plunge motions with $k_{red,Ex} = [0.2, 0.4, 0.6, 0.8]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.72$, Re $= 10^7$, $\alpha = \ 6°$, $\theta = \ \pm 1°$, $h = \pm 0.1 \cdot c_{ref}$).

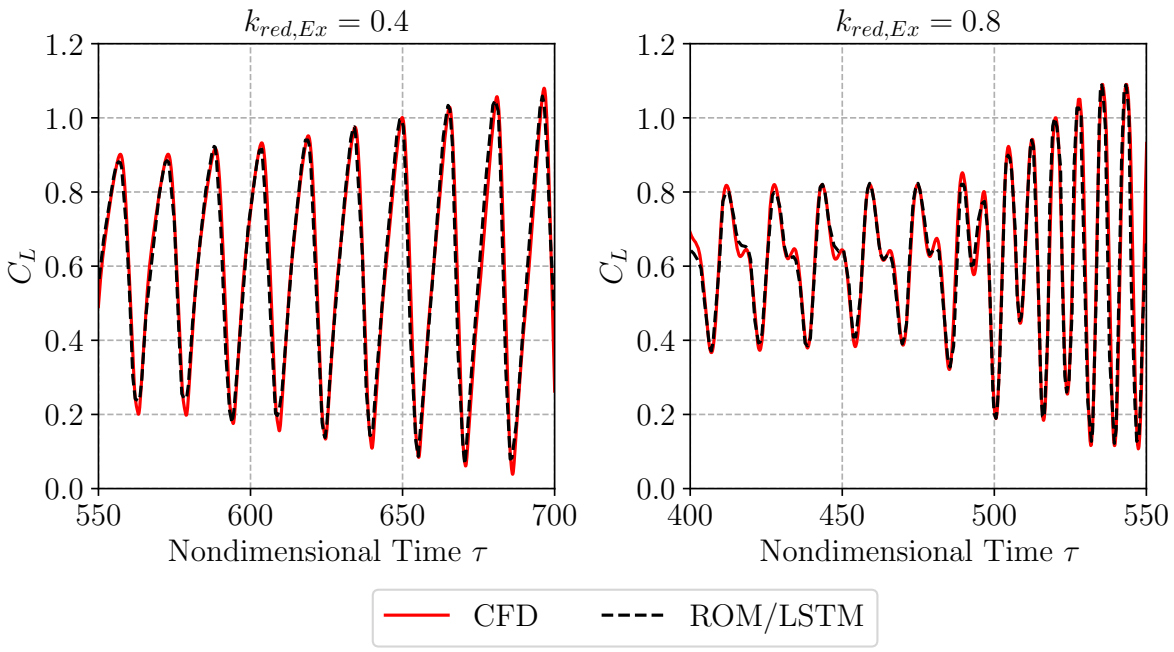


Figure 4.21: Time domain responses of the lift coefficient resulting from harmonic pitch and plunge motions with $k_{red,Ex} = [0.4,\ 0.8]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.72$, Re $= 10^7$, $\alpha = 6°$, $\theta = \pm 1°$, $h = \pm 0.1 \cdot c_{ref}$).

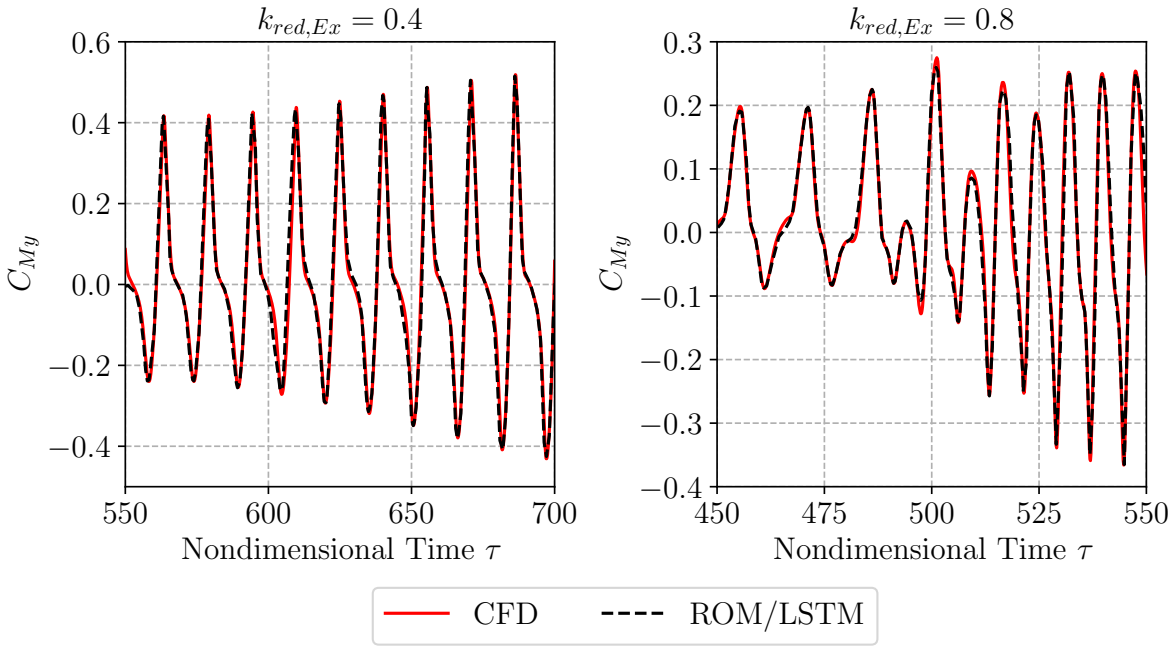


Figure 4.22: Time domain responses of the pitching moment coefficient resulting from harmonic pitch and plunge motions with $k_{red,Ex} = [0.4,\ 0.8]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.72$, Re $= 10^7$, $\alpha = 6°$, $\theta = \pm 1°$, $h = \pm 0.1 \cdot c_{ref}$).

However, compared to the responses of the single pitch excitation, the peaks appear to be more broadband, which is assumed to result from the additional plunge excitation. Examining both the frequency - and time-domain responses (see Figure 4.19 - Figure 4.22) of the harmonic motion with $k_{red,Ex} = 0.4$, lock-in is clearly indicated. Further, some small peaks at various frequencies are shown in the FFT spectra. Considering the spectra with $k_{red,Ex} = [0.6, 0.8]$, effects due to buffet and the harmonic excitation are visible.

Analogous to the results presented in Section 4.5.1, it is shown that the trained LSTM-ROM is able to model the time-series coefficient trends due to harmonic pitch and plunge motions. In addition, the lock-in effect is correctly reproduced. Comparing the evaluation metrics in Table 4.3 and Table 4.5 an overall increase of about 2-3% in the error metrics is presented for the results due to the combined DoF excitation. It is assumed that the loss in performance is due to the increased nonlinear behavior which must be covered by the trained LSTM-ROM. However, in general the errors are still small, underlying the good performance quality of the LSTM-ROM. Here, an overall fit of about 89% to 92% is achieved.

Excitation Frequency ($k_{red,Ex}$)	Aerodynamic Coefficient	A_{err}	P_{err}	C_{err}	Q
0.2	C_L	-0.053	0.018	0.049	91.7%
	C_{My}	-0.055	0.017	0.051	91.2%
0.4	C_L	-0.05	0.019	0.047	92.1%
	C_{My}	-0.053	0.02	0.05	90.6$
0.6	C_L	-0.058	0.021	0.055	89.4%
	C_{My}	-0.058	0.02	0.054	90%
0.8	C_L	-0.055	0.023	0.052	91.2%
	C_{My}	-0.055	0.02	0.051	90.4%

Table 4.5: Amplitude, phase, comprehensive error and fit factor of the LSTM-ROM applied to harmonic pitch and plunge motions covering varying excitation frequencies.

4.7 Efficiency Evaluation

Last but not least, the reduction in computational time due to the application of the LSTM-ROM, compared to a reference full-order CFD analysis, is summarized.

Each fully resolved numerical simulation with the DLR-TAU code has been performed on the SuperMUC-NG of the Leibniz Supercomputing Centre (LRZ). Here, six nodes with 48 cores each have been applied, resulting in a total number of 288 applied cores per simulation. In contrast, the training, validation and test computations of the LSTM-ROM have been performed on a workstation equipped with an Intel Xeon 2.2 GHz processor. For comparability, only a single CPU core has been used.

As a first step, the numerical effort related to the training of both the $\text{LSTM}_{\text{single}}$-ROM and $\text{LSTM}_{\text{combined}}$-ROM is outlined. The computation of each SAPRBS designed for the excitation of the pitch DoF required approximately 50 hours on the SuperMUC-NG. Considering the number of training signals (= 10) and the number of applied cores, an overall computational time of 144000 ($= 10 \cdot 50 \cdot 288$) CPU hours results. The computation of the three SAPRBS selected for the individual and simultaneous excitation of the pitch and plunge DoF required approximately 60 CPU hours, resulting in a total computational time of 51840 ($= 3 \cdot 60 \cdot 288$) CPU hours. In contrast, the training of the $\text{LSTM}_{\text{single}}$-ROM and $\text{LSTM}_{\text{combined}}$-ROM were each conducted within approximately ten and eight CPU hours on the workstation, respectively. Therefore, the overall computational training costs sum up to 144010 and 51848 CPU hours, with the CFD-based forced-motion SAPBRS simulations holding the highest share on the overall training procedure.

In the framework of an aerodynamic airfoil buffet analysis, the application of the trained ROM for various test cases, covering different frequencies and amplitudes, is of interest. Therefore, in the following, a comparison between the application of the LSTM in contrast to the fully resolved URANS simulations, is given. To summarize the CPU time of the LSTM, the SAPRBS simulations, the training of the LSTM, the CFD computations of the initial LSTM input solutions as well as the LSTM application on the test data set are required. In contrast, the CPU time of the URANS computations includes the CFD simulations of all required test cases.

Considering the excitation of only the pitch DoF, the application of the LSTM includes the amount of CPU hours for the SAPRBS simulations (= 144000), ten CPU hours for the training, a batch of initial time steps for the multi-step predictions for each harmonic motion and about 0.015 CPU hours for the application of the trained $\text{LSTM}_{\text{single}}$ itself. In contrast, the URANS simulation sum up to approximately 8640 ($= 30 \cdot 288$) CPU hours for each harmonic pitch motion. Since the oscillation period is dependent on the excitation frequency, a different number of computed time steps is set for each harmonic motion. Therefore, an averaged computation time of 30 hours required with the DLR-TAU solver is assumed. With an average of 8640 CPU hours, 30 excitation cycles with an average number of 3000 time steps are assumed. Therefore, for the computation of the initialization batch (= 32 time-steps), approximately 92 CPU hours are required.

Shifting the focus to the excitation of both the pitch and plunge DoF, the LSTM application is defined by 51840 CPU hours for the SAPBRS simulation, eight CPU hours for the training, 32 initial time-steps for the LSTM initialization and approximately 0.015 CPU hours for LSTM application itself. Assuming an averaged computation time of 30 hours for each test case, the fully resolved CFD simulations for each combined DoF excitation also sum up to approximately an average of 8640 CPU hours. Therefore, the simulation time for the initial test sample batch is also defined by approximately 92 CPU hours

In Table 4.6, the number of CPU hours for the application of both the $\text{LSTM}_{\text{single}}$-ROM and $\text{LSTM}_{\text{combined}}$-ROM are summarized. Further, the CPU hours for the fully resolved URANS simulations are included. Here, the CPU hours for the computation of a single harmonic motion test case are considered.

		$\text{LSTM}_{\text{single}}$	$\text{LSTM}_{\text{combined}}$
ROM	SAPRBS simulation	144000	51840
	ROM training	10	8
	ROM initialization	92	92
	ROM application	0.015	0.015
Fully resolved	URANS simulation	8640	8640

Table 4.6: Comparison of CPU hours for the $\text{LSTM}_{\text{single}}$ and $\text{LSTM}_{\text{combined}}$ and the fully resolved URANS simulations.

Considering a fair amount of harmonic motions for a detailed analysis, covering the frequency (e.g. $k_{red,Ex} = [0.2{:}0.1{:}0.8]$) and amplitude range of interest (e.g. $\theta = [\pm 0.1°, \pm 0.5°, \pm 1°]$), a number of 24 simulations needs to be performed. Therefore, the number of ROM related CPU hours for the $\text{LSTM}_{\text{single}}$ sum up to approximately 144000 for the SAPRBS simulation and 2208 ($= 24 \cdot 92$) CPU hours for the computation of the initialization batch for each harmonic test case. In contrast, the fully resolved CFD simulations would take up to 207360 ($= 24 \cdot 30 \cdot 288$) CPU hours. For the $\text{LSTM}_{\text{combined}}$, approximately 51840 CPU hours for the SAPBRS simulation as well as 2208 ($= 24 \cdot 92$) CPU hours for the ROM initialization solutions are mandatory. In contrast, the CPU hours for the URANS simulation are also summed up to 207360 CPU hours.

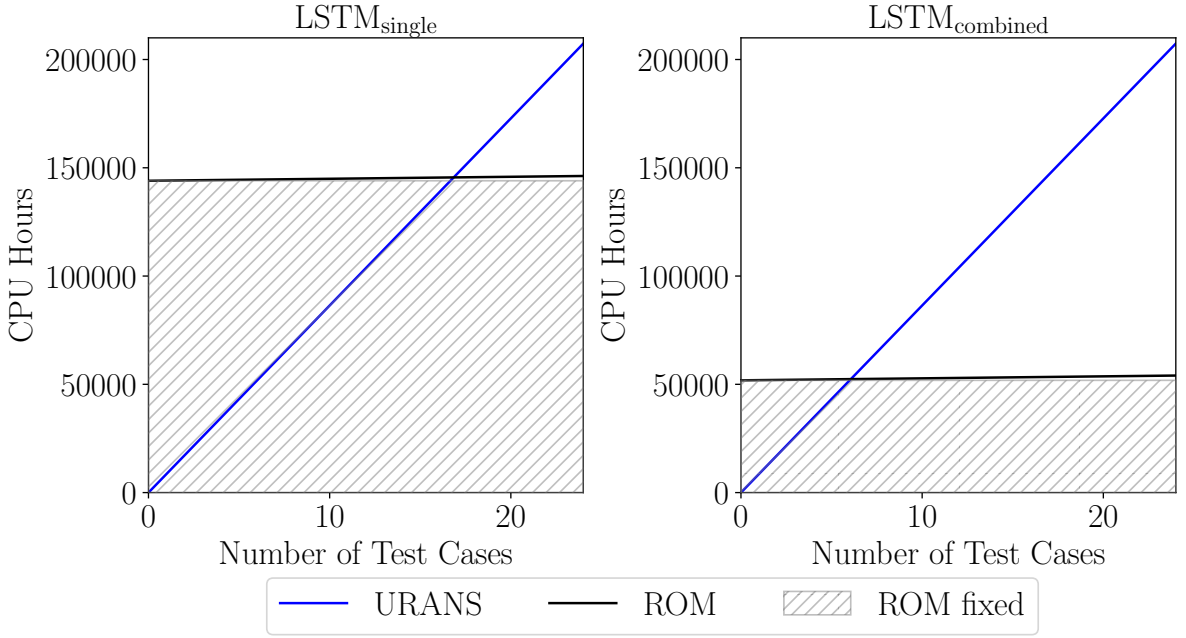


Figure 4.23: Comparison of CPU hours required for the fully resolved URANS simulation and the LSTM-ROM simulations ($\text{LSTM}_{\text{single}}$ (left) and $\text{LSTM}_{\text{combined}}$ (right)).

In order to visualize the reduction in computational costs, a comparison between the required CPU hours of the full-order URANS simulations and the application of both LSTM-ROMs for the considered test cases are provided in Figure 4.23. Here, the grey shaded area marks the number of CPU hours required for the training of the LSTM-ROM. The black line defines the additional CPU hours for the computation of the initialization solution as well as the application of the LSTM-ROM to the test data set. In contrast, the blue line denotes the number of CPU hours required for the fully resolved URANS simulations of the considered test cases. As shown, after the computation of a couple of test cases, the application of an URANS approach drastically increases the CPU hours compared to the application of a trained LSTM-ROM.

In order to further reduced computational time and costs, the number of training samples included in the SAPRBS could be decreased. Here, a reduction in the size of three to four orders of magnitude is assumed, depending on the number of considered test cases.

4.8 Summary

Within this chapter, a LSTM neural network-based ROM has been applied for the prediction of buffet force and moment coefficient time-series trends on the NACA0012 airfoil. Therefore, the airfoil has been excited with user-defined harmonic pitch and simultaneous pitch and plunge motions beyond the critical buffet angle of attack. The harmonic excitations have been defined in order to cover different frequencies and amplitudes. For modeling both the coefficient time-series trends due to single and simultaneous pitch and plunge motions, two different LSTM neural networks have been trained.

For the training, an optimized set of hyperparameters has been defined. In particular, the number of hidden layers, the number of neurons in each layer, the sequence length as well as the initial learning rate have been varied in order to find the best combination of parameters. The number of hidden layers and neurons in each layer should be chosen in order to enable a good prediction performance, while at the same time keeping the training time as low as possible. The size of the sequence is defined in order to provide a sufficient number of training samples to the network. With increasing sequence size, more information is fed into the neural network at the same time, however, the training time is also increased. Besides the definition of hidden layers, neurons and the length of the sequence, the initial learning rate must be set. Here, starting with a learning rate of $\eta = 1 \cdot 10^{-4}$ is the preferred choice. In order to adapt the learning rate, the convergence trends of training and validation loss should be monitored. If the convergence is very slow, the learning rate could be increased to speed up the process, however, overfitting can appear more easily. In contrast, if the convergence saturates at a rather high value, decreasing the learning rate can be helpful.

By comparing the time-series trends computed by CFD and the results of the LSTM-ROM, an accurate prediction is indicated for all considered test cases. Considering the different test cases, an overall fit between 89% and 93% has been achieved. Further, the LSTM is able to capture the lock-in effect as well as strong nonlinearities due to a simultaneous excitation of the pitch and plunge DoF. Besides the good prediction performance, the developed ROMs gained significant numerical cost and time savings compared to the full-order reference CFD solutions. Consequently, a robust and accurate framework for modeling integral airfoil buffet characteristics has been proposed.

5 Wing Buffet Prediction

This chapter deals with the ROM-based prediction of buffet characteristics due to varying freestream conditions on a civil aircraft wing. Therefore, the Airbus XRF-1 configuration is chosen as a test case. For modeling the wing buffet characteristics, a LSTM-based ROM and a hybrid ROM are applied. The hybrid ROM is defined by a series connection of a convolutional variational autoencoder (CNN-VAR-AE) and a LSTM. Experimental data obtained by two measurement campaigns in the European Transonic Wind Tunnel (ETW) is used for the training and performance evaluation of the proposed ROM methods. The content of this chapter is based on the author's publications, see references [150] and [152], and is structured as follows:

In the first section, the Airbus XRF-1 configuration as well as the experimental setup are introduced. Following the general introduction, a detailed buffet flow characterization based on the experimental data sets is presented in Section 5.2. In Section 5.3, the training and application procedure of a LSTM-based ROM for modeling local pressure characteristics at varying flow conditions and wing positions, is presented. In Section 5.4, the hybrid CNN-VAR-AE/LSTM is employed to predict surface pressure distributions at varying buffet conditions. For performance evaluation, the results of both ROM methods are compared to the reference experimental data.

5.1 Test Case and Experimental Setup: Airbus XRF-1

As a test case for the following investigations, a wind tunnel model representing the Airbus XRF-1 configuration is applied. The XRF-1 model represents a long range, twin engine aircraft test case. Similar to a modern transonic transport aircraft, the XRF-1 configuration is defined by a design Mach number of $Ma = 0.83$ and a design lift coefficient of $C_L = 0.5$ [43, 79].

During several consecutive wind tunnel test campaigns, two different XRF-1 configurations were considered. Both configurations include a fixed vertical tailplane (VTP) and a horizontal tail plane (HTP) as well as adjustable ailerons. The HTP was installed with a fixed incidence angle of α = - 2°. The wing of the model is defined by an aspect ratio of $AR = 9.302$, a mean aerodynamic chord of $c_{ref} = 0.1965$ m and a leading edge sweep of $\phi = 30°$ [132]. In Table 5.1, basic geometric properties of the XRF-1 wind tunnel model are summarized.

Quantity	**Symbol**	**Value**
Aspect Ratio	AR	9.302
Mean aerodynamic chord	c_{ref}	0.1965 m
Sweep angle	ϕ	30°

Table 5.1: Geometric properties of the Airbus XRF-1 wind tunnel model.

In addition to the tail planes and ailerons, each wing includes four flap track fairings, as depicted in Figure 5.1. Furthermore, one of the models is equipped with ultra high bypass ratio (UHBR) engine nacelles [122], as shown in Figure 5.2.

Figure 5.1: Front view of the Airbus XRF-1 wind tunnel model with clean wings (©Airbus/ETW).

Figure 5.2: Front view of the Airbus XRF-1 wind tunnel model with UHBR engine nacelles installed (©Airbus/ETW).

The wind tunnel model is mounted on a straight sting entering the lower fuselage with a cavity. The offset of the model/sting configuration is defined by an angle of 5°.

The experimental buffet data has been obtained during several experimental campaigns in the cryogenic wind tunnel ETW [132] in Cologne, Germany. The wind tunnel facility enables the installation of both half-span and full-span models. Besides integral force and moment measurements, several further measurement techniques can be applied in the ETW. The ETW can be operated in a Mach number range of 0.15-1.35 and a Reynolds number up to 50 Mio per meter. Further, an independent variation of Mach and Reynolds numbers as well as dynamic pressures is possible. These conditions are set by pressurizing the test section up to 450 kPa and operating at low temperatures around 110 K due to the injection of liquid nitrogen [132].

Dynamic Data Acquisition

During the wind tunnel tests, both steady and unsteady data were acquired. However, in the following the focus lies on dynamic pressure data acquisition. Therefore, the wing and HTP were instrumented with unsteady pressure transducers, referred to as Kulites (KUP), located at different span - and chordwise positions on the upper wing surface. Further, unsteady pressure transducers were mounted on the lower wing surface, however, they are not considered in the following. In Figure 5.3, the positions of the unsteady pressure transducers on the wing upper side are depicted.

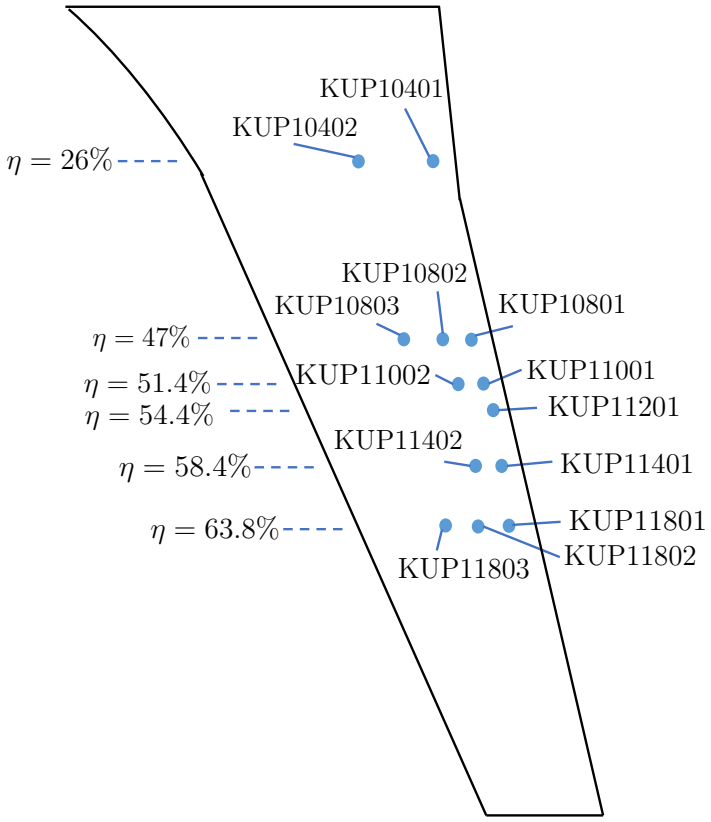


Figure 5.3: Pressure sensor positions on the wing upper side.

The sensors are located at six spanwise positions η=[26%, 47%, 51.4%, 54.4%, 58.4%, 63.8%]. The installation of the pressure sensors aims for capturing shock movement and flow separation at different flow conditions and incidence angles. The pressure signals are recorded with a high speed data acquisition system with a sampling rate of $fs = 10000$ Hz. All measurements are synchronized with a common time stamp. Further, the data is low-pass filtered at a frequency of $f = 4000$ Hz, which is lower than the Nyquist frequency f_N ($f_N < \frac{fs}{2}$). The length of each sensor signal was defined by the duration of additional pressure sensitive paint (PSP) measurements, with the pressure sensor being active during each run [132]. Therefore, the time for each run was set to 3.86 seconds, resulting in 38600 samples included in each signal.

In addition to the pressure transducers, several accelerometers are placed inside the model. In order to provide a separate signal for the x-,y-, and z-direction, a three-component accelerometer is located in the balance. Further, the nose, fuselage rear part, the port as well as the flap track fairings beneath the wings are equipped with devices measuring the acceleration in z-direction.

Pressure Sensitive Paint Measurements

In addition to the pressure sensor data acquisition, optical measurements using the pressure sensitive paint (PSP) technique and time-resolved PSP, in the following referred to as iPSP [73, 148], were conducted. The measurements were obtained by the DLR and the ETW test teams. Here, the unsteady pressure sensor signals define reference measurements for iPSP data. With the iPSP setup, upper wing surface pressure image acquisition with a sampling frequency of 1000 or 2000 Hz was possible. All iPSP measurements were conducted in a pitch/pause mode, enabling a fixed incidence at each run.

Buffet Flow Conditions

During the tests, different buffet conditions were considered. The measured flow conditions were defined in order to enable the analysis of isolated effects due to changes in Mach and Reynolds number as well as the angle of attack [132]. The Mach number was varied between Ma_∞ = [0.84, 0.87, 0.9], whereas the Reynolds number was set to Re = [3.3 Mio., 12.9 Mio., 25 Mio.]. Further, two levels of dynamic pressure ratios q/E = [0.2 $\cdot 10^{-6}$, 0.4 $\cdot 10^{-6}$] were applied. In the following, flow conditions defined by Ma_∞ = 0.78 and Re = 3.3 Mio. are not considered, since no unsteady data was acquired at these conditions. In Table 5.2, the flow conditions considered for the investigations in this thesis are summarized.

	$Ma_\infty = 0.84$ $Re = 12.9$ Mio.	$Ma_\infty = 0.84$ $Re = 25$ Mio.	$Ma_\infty = 0.9$ $Re = 12.9$ Mio.	$Ma_\infty = 0.9$ $Re = 25$ Mio.
α [°]	3, 3.5, 4	3, 3.5, 4, 4.5, 5	5, 6, 6.5, 6.9, 7	2.5, 4, 5, 6

Table 5.2: Investigated buffet flow conditions in the ETW for the XRF-1 configuration.

5.2 Buffet Flow Characterization

In the following section, a comprehensive characterization and analysis of the buffet instability on the XRF-1 configuration is presented. In the first part of the section, the flow topology of the buffet flow is discussed. Here, differences between the flow characteristics of the clean wing and the wing with the UHBR nacelle are highlighted. Following the general characterization, the data obtained by the pressure sensors is examined and presented. In the third subsection, a shock-motion analysis is conducted by means of cross spectra and coherence of the unsteady pressure samples.

5.2.1 Flow Topology

In the following section, iPSP surface pressure data is analyzed. Due to Airbus disclosure restrictions, the visualizations of contours can be solely presented non-dimensional and without the corresponding legend. Further, it has to be noted that the LE and TE of the wing are subject to higher measurement errors due to the iPSP measurement limitations, which result in flawed data. Therefore, the data near the LE and TE are neglected in the following investigation. In all figures, the surface pressure distribution is visualized using the mean of the pressure coefficient $\bar{c}_p$, computed based on 500 c_p-snapshots of each flow condition. The $\bar{c}_p$ is defined as follows

$$c_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty U_\infty^2}, \qquad \bar{c}_p = \sum_{i=1}^{500} c_{p_i} \tag{5.1}$$

with p_∞, ρ_∞ and U_∞ denoting freestream static pressure, density and freestream flow velocity, respectively.

Prior to a detailed flow characterization considering different Mach and Reynolds numbers as well as angle of attacks α, a general characterization of the buffet flow topology is presented. In Figure 5.4, the mean pressure distribution on the wing suction side at a flow condition of $Ma_\infty = 0.84$, $Re = 25$ Mio. and an angle of attack of $\alpha = 4°$, is depicted. Here, the clean wing without UHBR nacelle is considered. Further, it has to be noted that the large green dots on the entire wing represent areas where no data has been obtained by the optical iPSP measurement. In contrast, the smaller green dots arranged in lines on the TE are defined as the position of the unsteady pressure sensors.

As shown in Figure 5.4, a distinct λ-shaped two-shock pattern across the entire wing span is clearly visible. The main shock, which is marked in yellow, spreads over the rear part of the chord from wing root (inboard region) to the wing tip (outboard region), whereas a second shock (marked in orange) is visible at the inboard region of the wing. Here, it originates near the LE at the root of the wing towards the midspan region of the wing. Behind the main shock, the flow is separated along the whole span. The area of separated flow is visualized by a black dotted line in Figure 5.4.

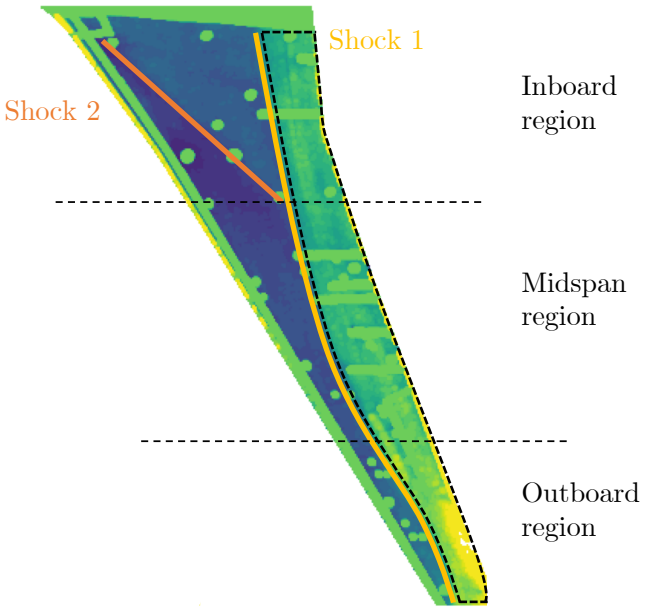


Figure 5.4: Mean surface $\bar{c}_p$ on the clean wing suction side at $Ma_\infty = 0.84$, $Re = 25$ Mio. and $\alpha = 4°$ (iPSP).

Following the general characterization of the flow topology at buffet condition on the clean wing configuration, varying flow conditions are considered in the following. In Figure 5.5, the mean pressure distributions on the wing suction side at a flow condition of $Ma_\infty = 0.84$, $Re = 25$ Mio. and varying angles of attack $\alpha = [3°, 3.5°, 4°, 4.5°, 5°]$ are visualized.

For each flow condition, the characteristic λ-shaped two-shock pattern is visible, however, clear changes due to varying α are indicated. With increasing incidence, the main shock position at the inboard region slightly shifts aft, while the shock position in the midspan and outboard region moves towards the LE. The sweep angle of the second shock increases and the intersection point of both shocks moves inboard. Further, the intensity of both shocks increases with increasing incidence.

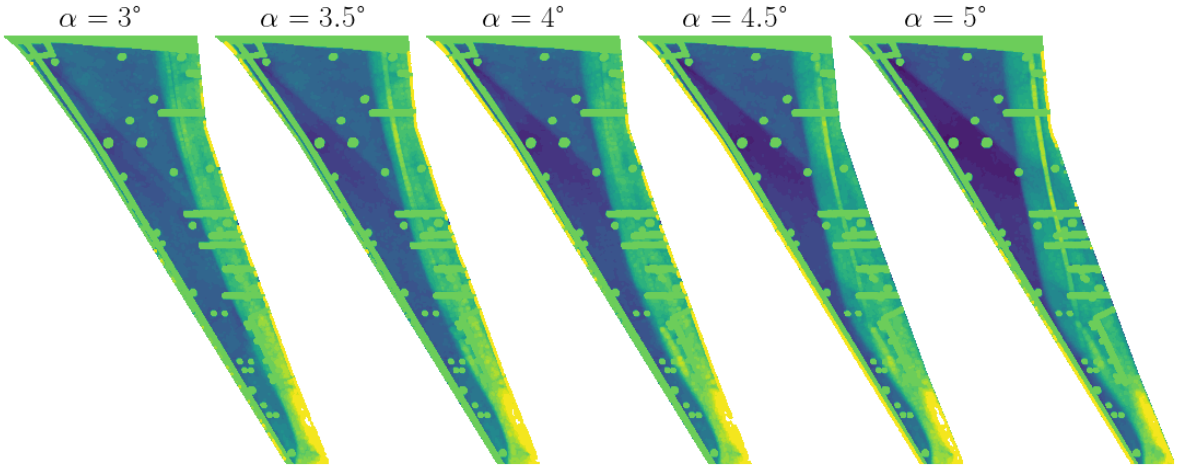


Figure 5.5: Comparison of surface c_p at varying angles of attack (clean wing, $Ma_\infty = 0.84$, $Re = 25$ Mio., iPSP)

Considering a flow condition with a higher Mach number $Ma_\infty = 0.9$ and $Re = 25$ Mio. kept constant, as shown in Figure 5.6, similar changes in the span-wise two-shock pattern become apparent. With increasing α, the intensity and position of both shock changes, however, the main shock movement in the midspan and outboard region is less pronounced compared to the lower Mach number flows visualized in Figure 5.5.

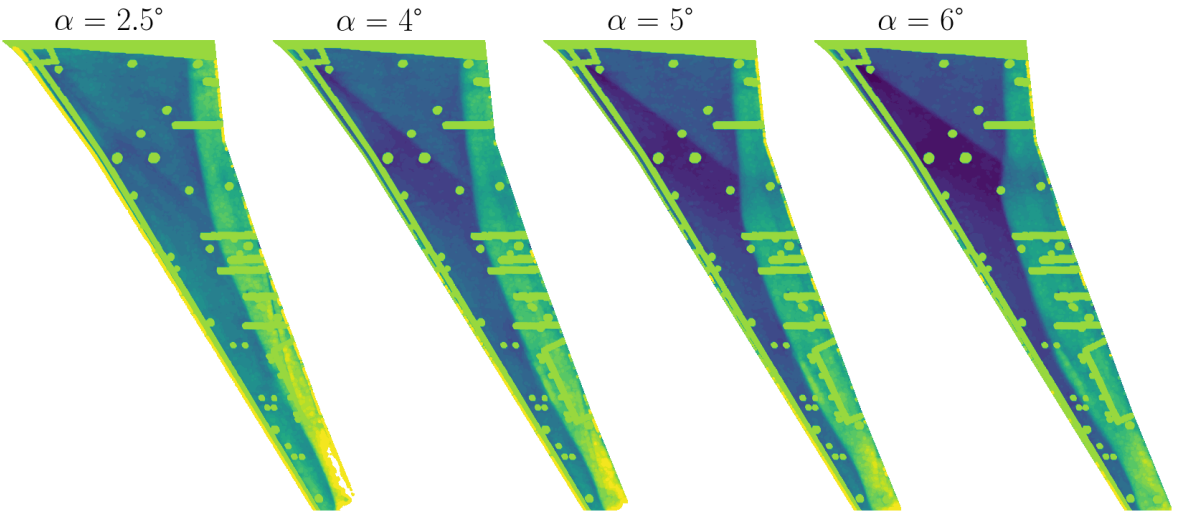


Figure 5.6: Comparison of surface c_p at varying angles of attack (clean wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., iPSP)

Besides the influence of increasing α, the influence of changes in the Re number on the pressure distribution are investigated in the following. In Figure 5.7, wing suction side pressure distributions for two Re numbers $Re = [12.9, 25]$ Mio. for two specific Mach and α combinations ($Ma_\infty = [0.84, 0.9]$, $\alpha = [4°, 5°]$) are exemplary visualized. As shown, compared to the influence of varying angles of attack, changes in the Re number do not affect the position and the intensity of the spanwise two-shock pattern.

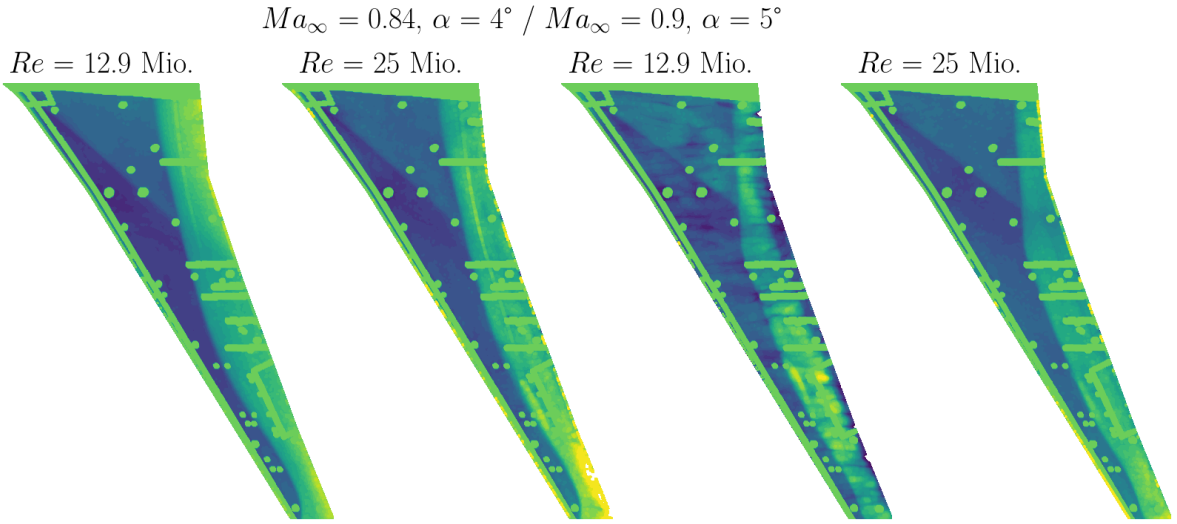


Figure 5.7: Comparison of surface c_p at varying Reynolds numbers $Re = [12.9, 25]$ Mio., $Ma_\infty = [0.84, 0.9]$ and $\alpha = [4°, 5°]$ (iPSP).

Shifting the focus to the surface pressure distribution of the wing with the UHBR nacelle installed, a general characterization of the flow topology is visualized in Figure 5.8. Here, a flow condition with $Ma_\infty = 0.84$, $Re = 25$ Mio. and $\alpha = 4°$ is considered. Similar to the flow topology of the clean wing (see Figure 5.4), the characteristic λ-shaped two-shock pattern along the wing span is again clearly visible. In addition, a third shock (marked in red) originates from the position of the UHBR nacelle to the root of the wing and intersects with the other inboard shock.

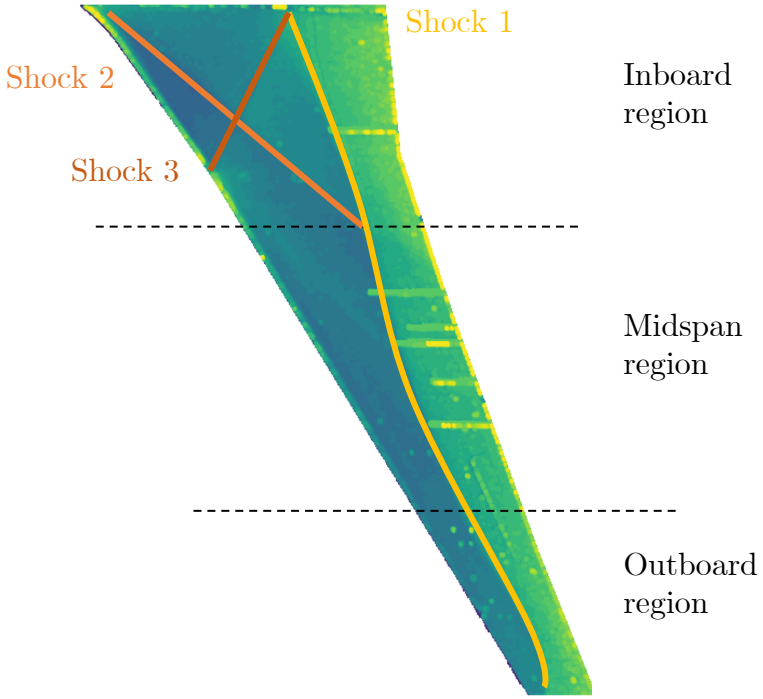


Figure 5.8: Mean surface $\bar{c}_p$ on the UHBR wing suction side at $Ma_\infty = 0.84$, $Re = 25$ Mio. and $\alpha = 4°$ (iPSP).

Following the general presentation of the UHBR wing buffet flow topology, changes in surface pressure due to varying flow conditions are analyzed in the following. In Figure 5.9, changes in surface pressure due to changes in the angle of attack at $Ma_\infty = 0.84$ and $Re = 25$ Mio., are shown. Consistent with findings referred to the flow topology on the clean wing, with increasing α the terminating inboard shock position moves towards the TE, while the shock position in the midspan and outboard region is shifted forward towards the LE. However, compared to the clean wing configuration, the movement of the inboard shock position is less distinct. Further, the magnitude of the main and second shock increases with increasing incidence, while the strength of the third shock remains unchanged. In addition, focusing on the intersection of shock 1 and shock 2, a similar behavior as on the clean wing is indicated. With increasing α, the intersection of both shocks moves towards the inboard region.

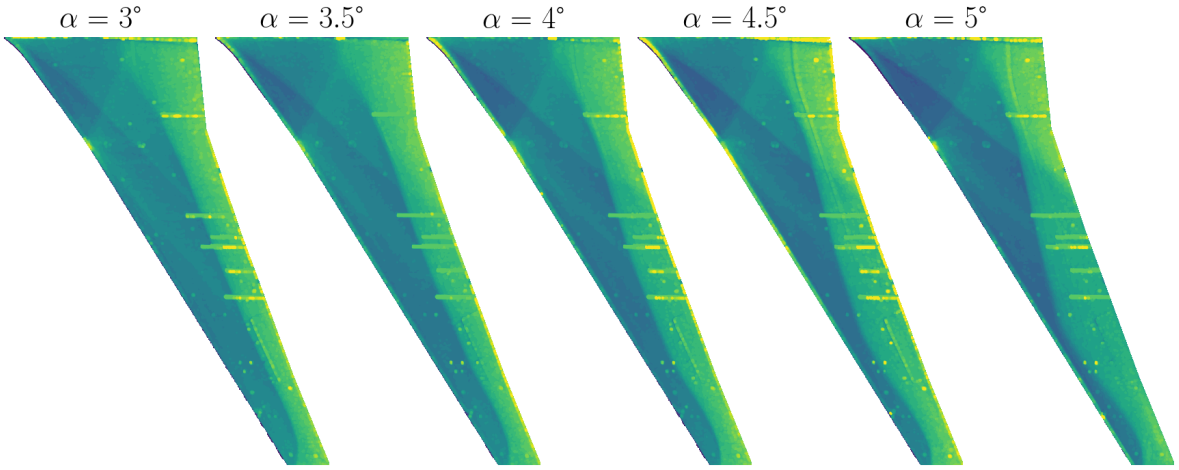


Figure 5.9: Comparison of surface c_p at varying angles of attack (UHBR wing, $Ma_\infty = 0.84$, $Re = 25$ Mio., iPSP).

In Figure 5.10, the surface pressure distributions at $Ma_\infty = 0.9$, $Re = 25$ Mio. and varying $\alpha = [2.5°, 4°, 5°, 6°]$ are depicted.

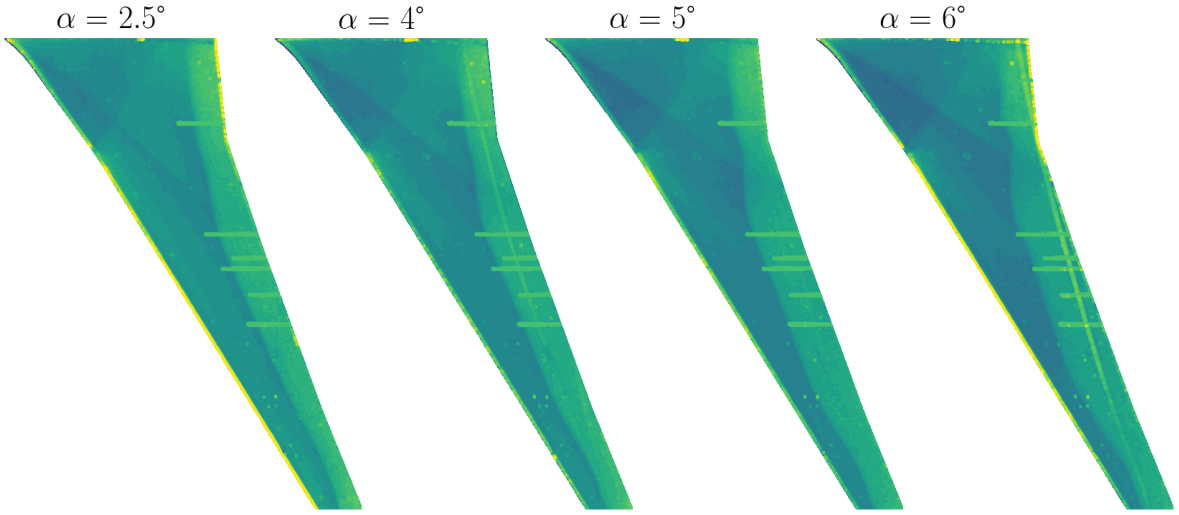


Figure 5.10: Comparison of surface c_p at varying angles of attack (UHBR wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., iPSP).

Analogous to the lower Mach number cases, the inboard shock position moves towards the TE, whereas in the midspan and outboard area the main shock moves to the LE. However, compared to the surface pressures shown in Figure 5.9, the shift towards the LE is less pronounced than for the lower Mach number pressure distributions.

To summarize the results presented within this section in Figure 5.5 to Figure 5.10, distinct buffet flow characteristics are captured by the optical iPSP measurements. Further, clear differences between the flow topology of both the clean wing and UHBR wing configuration, are presented.

5.2.2 Unsteady Pressure Sensor Analysis

In order to identify characteristic features of the buffet instability, pressure sensor data is analyzed in the following. Here, only pressure data measured on the clean wing configuration are considered for simplification. For the analysis, flow conditions at $Ma_\infty = [0.84, 0.9]$ and $Re = 25$ Mio. are investigated. Further, the pressure spectra at different span - and chordwise positions are evaluated. Analogous to the data presented in Section 5.2.1, the following figures are shown without the corresponding legend on the y-axis due to Airbus closure restrictions.

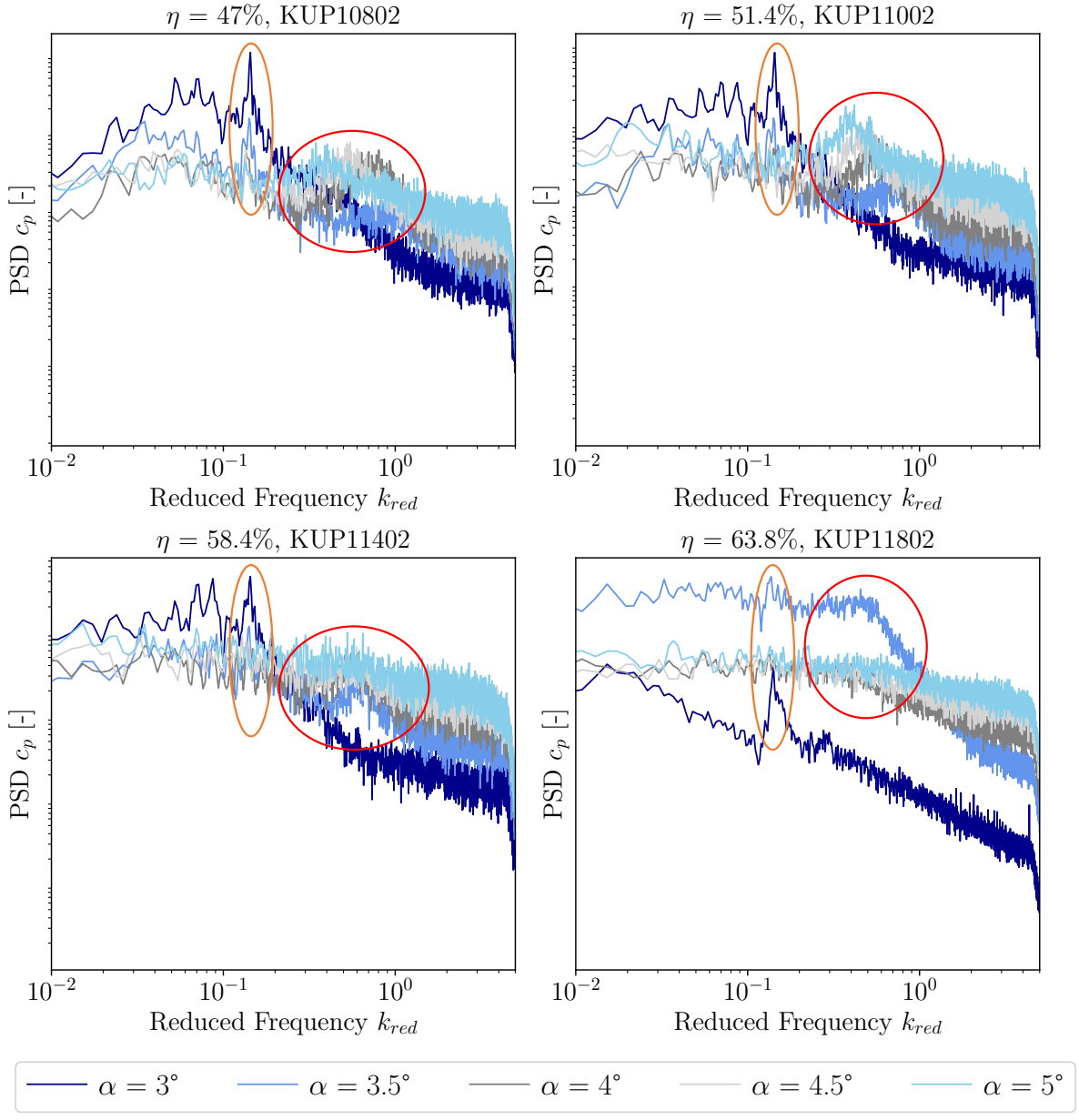


Figure 5.11: Power spectral densities of c_p at varying α obtained by dynamic pressure sensors at spanwise positions $\eta = [47\%, 51.4\%, 58.4\%, 63.8\%]$ ($Ma_\infty = 0.84$, $Re = 25$ Mio.).

In Figure 5.11, the power spectral densities (PSD) of the pressure data at $Ma_\infty = 0.84$ and $Re = 25$ Mio are plotted over the reduced frequency, considering different spanwise positions $\eta = [47\%, 51.4\%, 58.4\%, 63.8\%]$. In all spectra, a variety of peaks at different reduced frequencies, are visible. At low frequencies around $0.01 < k_{red} < 0.1$, higher amplitude peaks dominate the spectrum, especially at lower angles of attack of $\alpha = [3°, 3.5°, 4°]$. With increasing incidence, the amplitude peaks at lower frequencies are less pronounced. Besides the low frequency peaks, a significant peak at a reduced frequency of $k_{red} \approx 0.13$ (marked in orange) is present for almost all angles of attack. Especially for lower angles of attack, this peak dominates the spectra.

Apart from the narrowband, low frequency peaks, broadband fluctuations occur at higher frequencies with a wide peak (marked in red) emerging. At a spanwise position of $\eta = 47\%$ and $\alpha = 3.5°$, this peak occurs at roughly $k_{red} = 0.8$ and shifts to lower frequencies of $k_{red} = 0.4$ at $\alpha = 5°$. A similar trend is visible at $\eta = 51.4\%$, with the broadband frequency bump shifting from $k_{red} = 0.7$ at $\alpha = 3.5°$ to $k_{red} = 0.4$ at $\alpha = 5°$. However, the amplitude of the broadband frequency bump is larger compared to the frequency bump at $\eta = 47\%$. Consistent with findings of recent numerical and experimental studies [54, 59, 93], these broadband fluctuations are assumed to be related to aerodynamic buffet oscillations. At $\alpha = 3°$, almost no broadband frequency peak is visible in the spectra, which might results from buffet onset at this angle of attack [132].

With increasing spanwise position, the amplitude of the broadband frequency bump starts to decrease. Further outboard at $\eta = 63.8\%$, the broadband frequency range is not as easily distinctable from the remaining higher frequency content. There is some indication of a high frequency bump at lower incidences, however, it is less distinct than at the other spanwise positions.

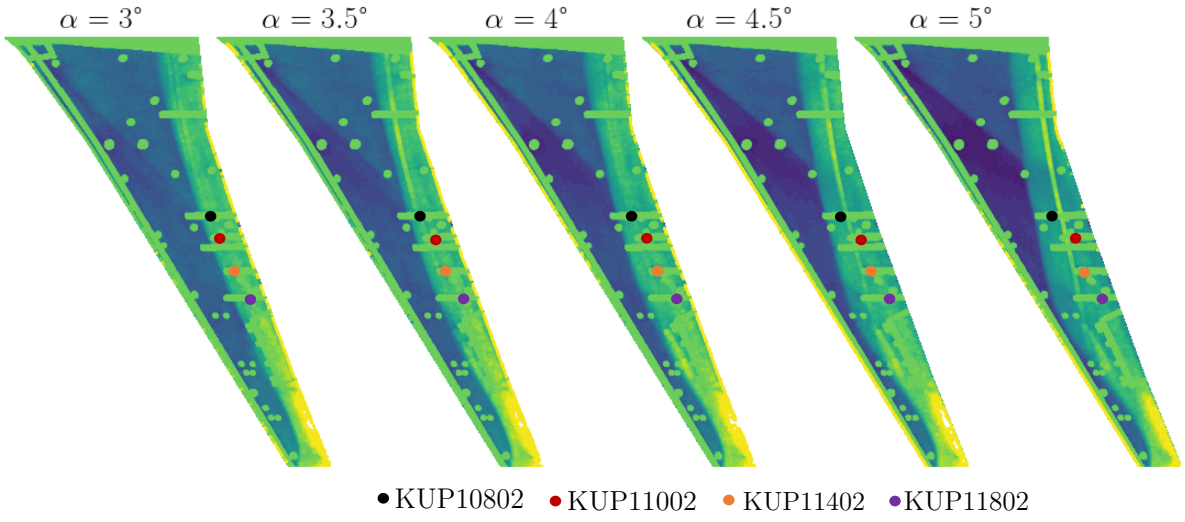


Figure 5.12: Mean pressure distribution c_p on the upper wing surface at varying angles of attack $\alpha = [3°, 3.5°, 4°, 4.5°, 5°]$ ($Ma_\infty = 0.84$, $Re = 25$ Mio.).

Examining the shock position at each respective flow condition relative to the sensor position at KUP11802 ($\eta = 63.8\%$), as shown in Figure 5.12, a larger distance between

shock position and sensor position is visible for increasing angles of attack. Due to the increased distance, the buffet oscillations might be less visible in the spectrum compared to the other sensor positions.

Shifting the focus to the pressure data obtained at $Ma_\infty = 0.9$ and $Re = 25$ Mio., as shown in Figure 5.13, a similar but less pronounced characteristic, as presented in Figure 5.11, is visible.

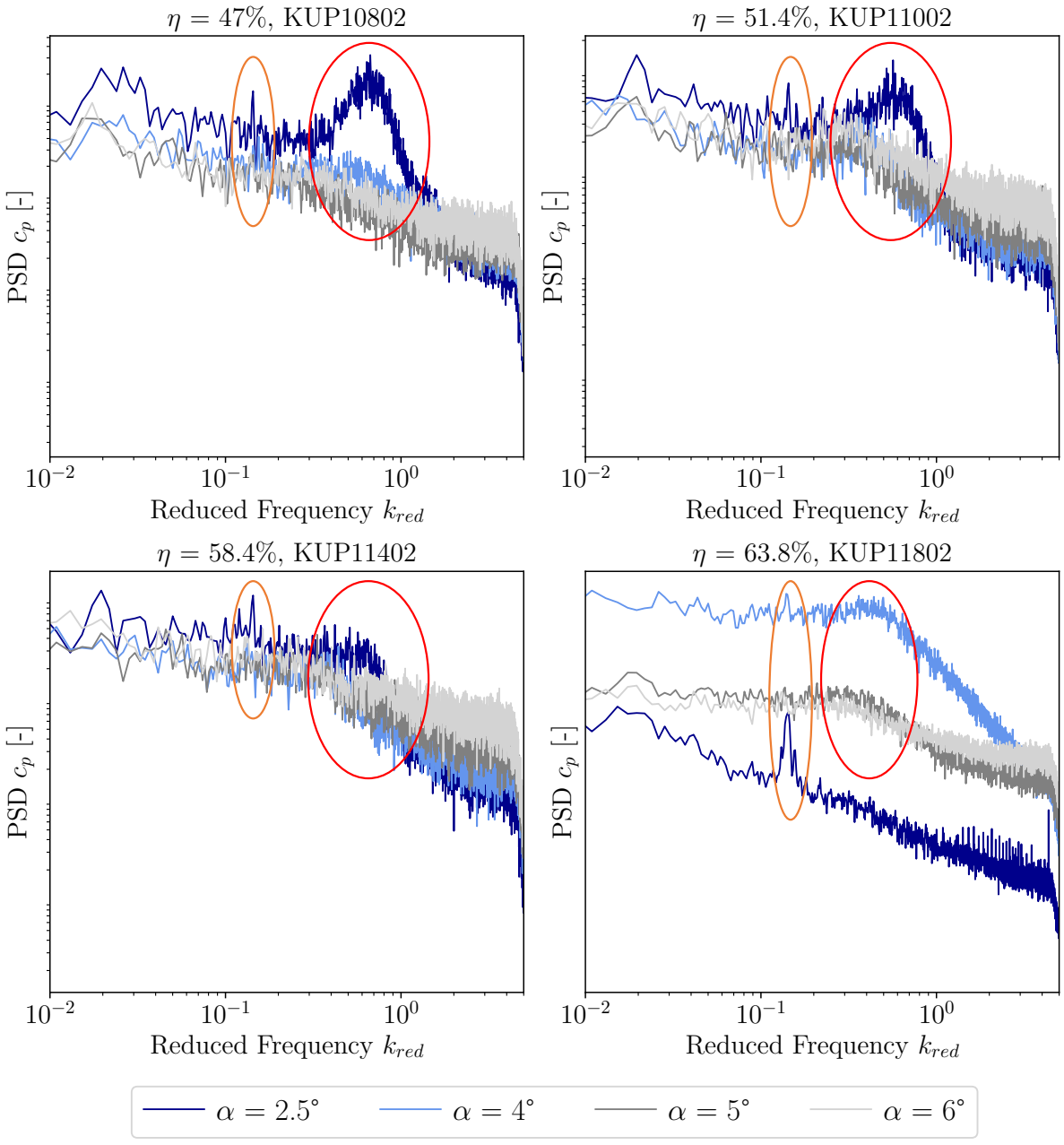


Figure 5.13: Power spectral densities of c_p at varying α obtained by dynamic pressure sensors at spanwise positions $\eta = [47\%, 52.4\%, 58.4\%, 63.8\%]$ ($Ma_\infty = 0.9$, $Re = 25$ Mio.).

At low frequencies ($0.01 < k_{red} < 0.1$), high amplitude peaks are present, with the amplitudes reducing with increasing angles of attack. The characteristic peak at $k_{red} \approx 0.13$ (marked in orange) is also indicated in all spectra. At a lower angle of attack $\alpha = 2°$,

a broadband frequency bump (marked in red) at $0.3 < k_{red} < 0.8$ is visible, with a comparatively larger amplitude at $\eta = [47\%, 51.4\%]$ than at $Ma_\infty = 0.84$. As shown in Figure 5.14, at $\alpha = 2.5°$ the sensors are located close to the shock, which might cause the larger increase in the amplitude. Similar to the spectra at $Ma_\infty = 0.84$, the frequency bump slightly shifts towards lower frequencies with increasing incidence, however, the overall fluctuations are more broadband.

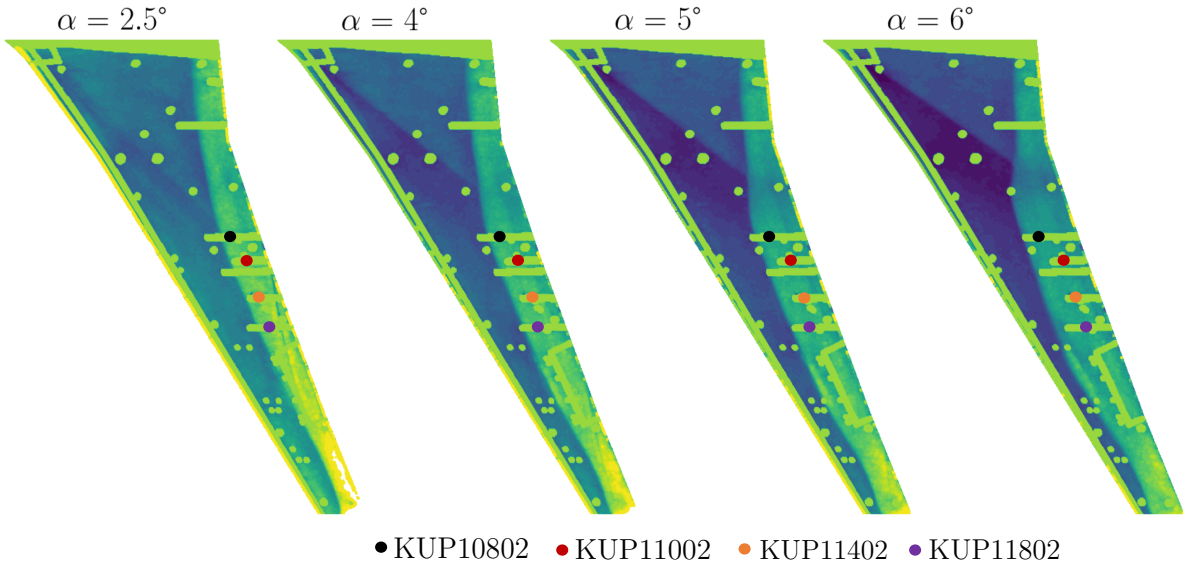


Figure 5.14: Mean pressure distribution c_p on the upper wing surface at varying angles of attack $\alpha = [2.5°, 4°, 5°, 6°]$ ($Ma_\infty = 0.9$, $Re = 25$ Mio.).

Besides the visualization of the sensor spectra at varying angles of attack, a comparison of the spectra at $Ma_\infty = [0.84, 0.9]$, $Re = 25$ Mio. is depicted in Figure 5.15. Here, angles of attack of $\alpha = [4°, 5°]$ measured at the same sensor position as shown in Figure 5.11 and Figure 5.13, are considered. By comparing the spectra it becomes clear that at $Ma_\infty = 0.84$, the overall level of PSD at the sensor positions KUP10802, KUP11002 and KUP11402 is higher over the entire reduced frequency bandwidth. In contrast, the spectra at KUP11802 show an overall higher PSD level for $Ma_\infty = 0.9$. Further, as already shown in Figure 5.11 and Figure 5.13, for the lower Mach number spectra, the broadband buffet frequency peak is more pronounced and shifted towards higher reduced frequencies.

Since the experiments were conducted using a full aircraft model, which is mounted on a sting, structural vibrations need to be taken into account if frequency spectra are analyzed. Besides aerodynamic phenomena, pressure fluctuations and corresponding frequencies due to structural eigenfrequencies need to be identified.

A ground vibration test (GVT) is preferred to give insights into the structural behavior of the model. Based on the GVT, a number of modes and their corresponding frequencies have been identified and provided by the DLR and ETW test teams. In Table 5.3, the modes and frequencies are summarized. In order to determine the origin and

differences of various peaks in the pressure and acceleration sensor spectra, the spectra of the acceleration signals are investigated in the following.

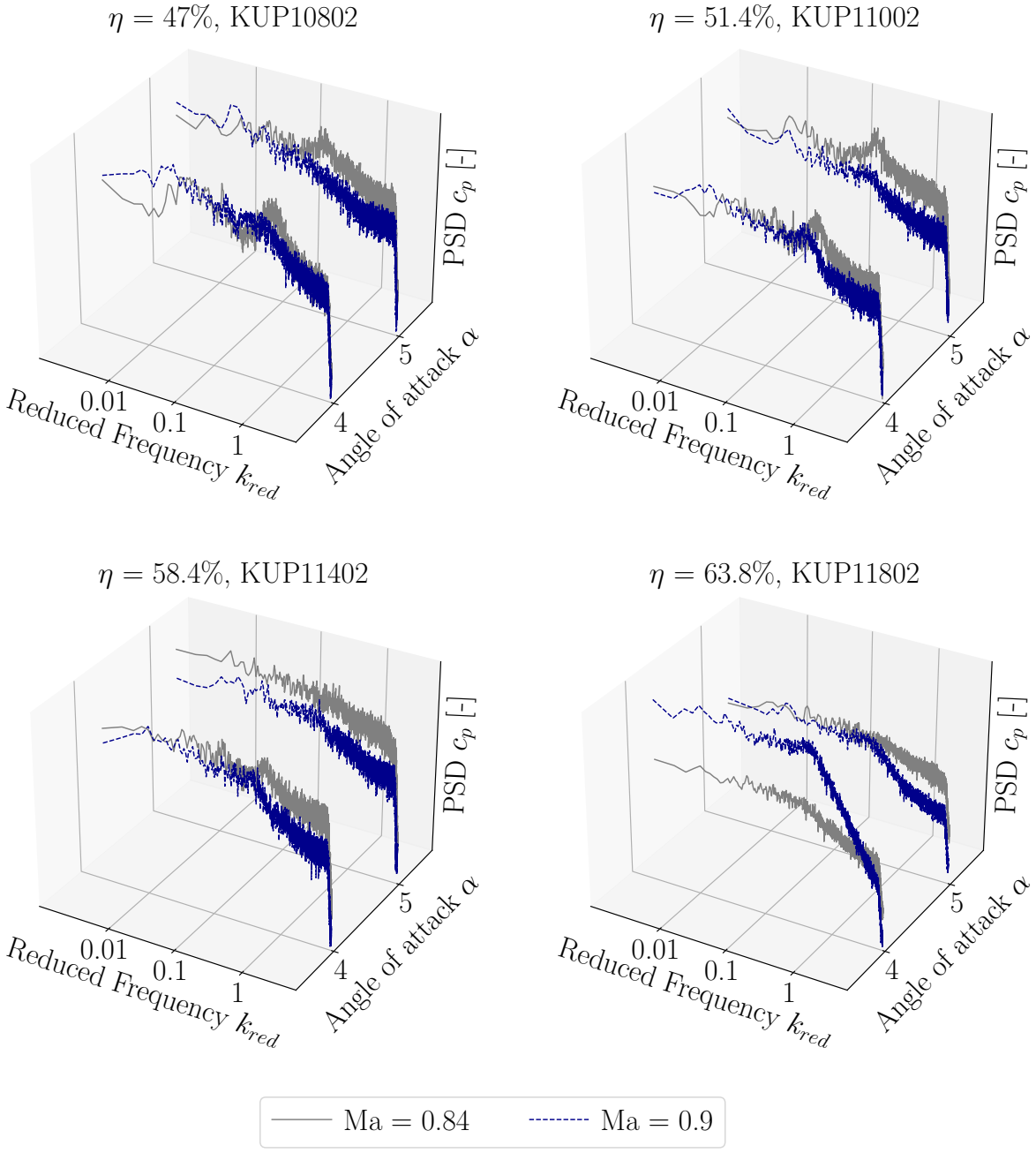


Figure 5.15: Power spectral densities of c_p at $\alpha = [4°, 5°]$ and $Ma_\infty = [0.84, 0.9]$ obtained by dynamic pressure sensors at spanwise positions $\eta = [47\%, 52.4\%, 58.4\%, 63.8\%]$ ($Re = 25$ Mio.).

In Figure 5.16, the spectra of the nose-, rear-, starboard- as well as the triax balance accelerometers are depicted, considering $Ma_\infty = [0.84, 0.9]$, $Re = 25$ Mio. and $\alpha = 4°$. Although different flow conditions are considered, similar spectra results, with distinct peaks at a wide range of frequencies. However, most dominant is the peak at $k_{red} = 0.13$, that is also visible in most of the pressure spectra of the sensor signals shown in Figure 5.12 and Figure 5.14.

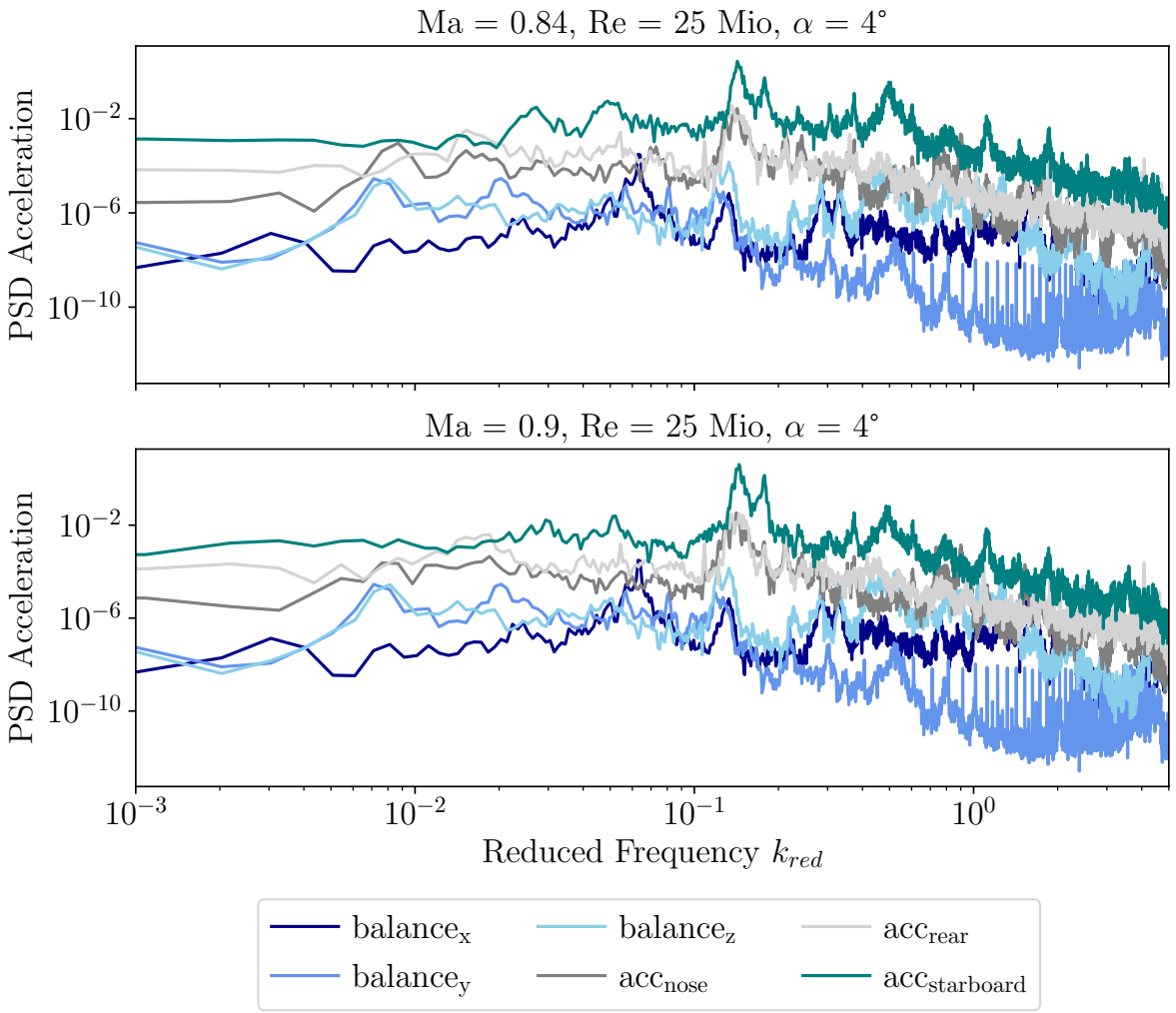


Figure 5.16: Power spectral densities of accelerometer signals obtained at $Ma_\infty = [0.84, 0.9]$, $Re = 25$ Mio. and $\alpha = 4°$.

Name	Frequency [Hz]	Damping [%]
A/C asymmetric y-translation	7.39	0.61
A/C symmetric heave	8.121	0.25
A/C asymmetric yaw	10.59	0.34
A/C symmetric pitch	13.15	0.23
A/C asymmetric roll	24.96	0.12
A/C symmetric x-translation	61.14	0.76
Asymmetric sting bend	25.71	1.34
Symmetric 2^{nd} wing bending	42.12	0.24
Symmetric sting bending	45.41	2.04
Asymmetric 3^{rd} wing bending	63.34	0.23
Asymmetric 2^{nd} sting bending	70.84	1.42
Asymmetric lateral sting bending	107.08	1.17
4^{th} wing bending	126.72	0.31
Asymmetric 5^{th} wing bending	153.68	0.76
Asymmetric fuselage bending	196.04	1.31
Symmetric fuselage bending	205.19	1.31
Asymmetric 1^{st} wing inplane	237.61	1.18
Symmetric 2^{nd} wing inplane	249.25	0.65
Symmetric 6^{th} wing bending	309.25	0.72
Asymmetric 7^{th} wing bending	330.39	0.55
Asymmetric 2^{nd} sting bend	365.69	1.12
Symmetric 3^{rd} fuselage bend	382.86	1.91
Asymmetric wing torsion	421.26	0.85
Symmetric wing torsion	425.98	0.70
Asymmetric 3^{rd} fuselage bend	467.75	0.74
Symmetric 8^{th} wing bending	486.19	1.00

Table 5.3: Modes and corresponding frequencies of the XRF-1 wind tunnel model as obtained by a ground vibration test (GVT).

In order to compare the frequency peaks of both acceleration and pressure sensor signals, the spectrum of the starboard accelerometer signal and the pressure sensor KUP11002 are shown in Figure 5.17. For a better assignment of the GVT modes listed in Table 5.3, the spectra are represented using the frequency instead of the reduced frequency. As indicated, several frequencies related to GVT modes can be identified in both spectra.

In the lower frequency range at 8-13 Hz, frequencies related to sting induced heave and pitching oscillations are indicated. At frequencies of 20-45 Hz, oscillations and symmetric wing bending frequencies are visible in both spectra. At a frequency of 127 Hz, a distinct peak is shown in both spectra, which corresponds to the peak at $k_{red} = 0.13$, visible in most of the pressure sensor spectra. Therefore, this peak clearly shows a structural related frequency associated to wing bending. Further, a second larger peak at a frequency of 154 Hz is visible, which is assumed to be additionally related to wing bending.

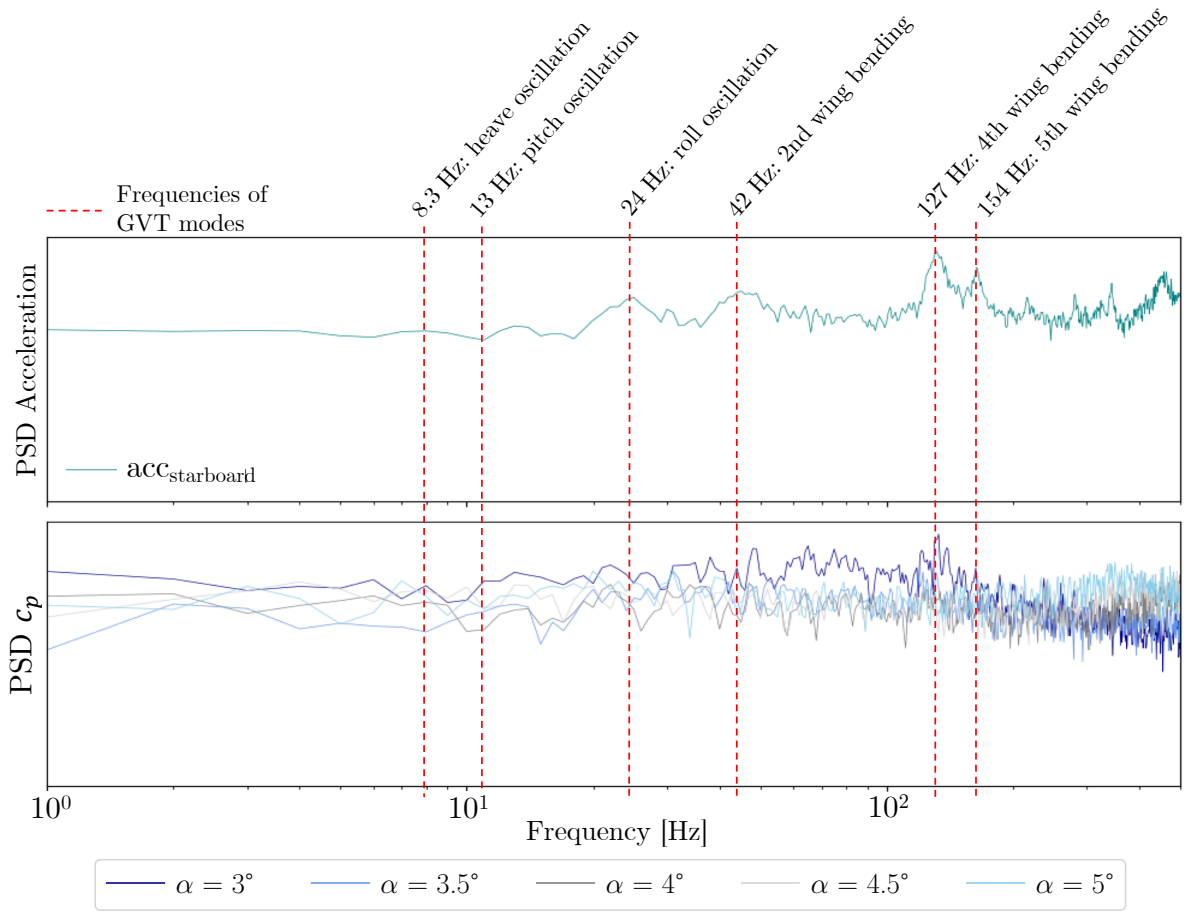


Figure 5.17: Power spectral density of the starboard accelerometer signal (top) and the pressure sensor KUP11002 (bottom) obtained at $Ma_\infty = 0.84$, $Re = 25$ Mio. and $\alpha = 4°$.

As a conclusion, based on the comparison of the accelerometer and pressure sensor spectra, frequencies related to structural vibrations as well as buffet frequencies can be identified and distinguished. The lower frequency range covers clear structural frequencies, whereas the buffet frequencies are present at a higher frequency range with a larger bandwidth compared to the structural frequencies.

5.2.3 Shock Motion Analysis

Following the investigation of the buffet characteristics, cross-spectra analysis is performed in order to quantify the buffet wave-propagation speed in spanwise direction. Following the studies by Koike et al. [59] and Masini et al. [83], data measured by the unsteady pressure sensors is used to determine the buffet propagation speed.

Considering two pressure signals x_t and y_t obtained at buffet condition and including a certain number of timesteps t, the statistical relation between the signals can be expressed by their magnitude squared coherence C_{xy} at a defined sampling frequency f [83]:

$$C_{xy}(f) = \frac{|S_{xy}(f)|^2}{S_{xx}(f)S_{yy}(f)} \tag{5.2}$$

with $S_{xx}(f)$ and $S_{yy}(f)$ denoting the PSD estimates of each signal, calculated by means of Welch's method. $S_{xy}(f)$ is defined as the cross-spectrum, which is computed using the cross-correlation of both signals. The phase shift $\phi_{xy}(f)$ between the two signals can be evaluated as follows [83]:

$$\phi_{xy}(f) = arctan\left(\frac{Re(S_{xy}(f))}{Im(S_{xy}(f))}\right) \tag{5.3}$$

Based on the physical distance between the pressure sensors Δs, the frequency range Δf at high coherence as well as the phase shift at the frequency range $\phi_{xy}(\Delta f)$, the convection velocity U_c can be computed as follows [83]:

$$U_c = 2\pi\Delta s\frac{\Delta f}{\phi_{xy}(\Delta f)} \tag{5.4}$$

For the wave speed computation, data obtained at $Ma_\infty = 0.84$ and $Re = 25$ Mio., is selected. In Figure 5.18 and Figure 5.19, five snapshots of the pressure distribution on the upper wing surface are shown for $Ma_\infty = 0.84$, $Re = 25$ Mio. and $\alpha = 4°$ and $Ma_\infty = 0.9$, $Re = 25$ Mio. and $\alpha = 4°$, respectively. These five snapshots represent a buffet period.

By comparing the change in pressure distribution along the span, a more pronounced motion of the shock front for $Ma_\infty = 0.84$ is indicated. Especially at a spanwise position between roughly 60% to 80%, the motion of the buffet instability is visible. For clearance, the area of shock motion is marked with a black circle. Further inboard and outboard, almost no buffet oscillations are indicated. At $Ma_\infty = 0.9$, a buffet cell motion is indicated at 70% spanwise position, however, the shock motion is less pronounced.

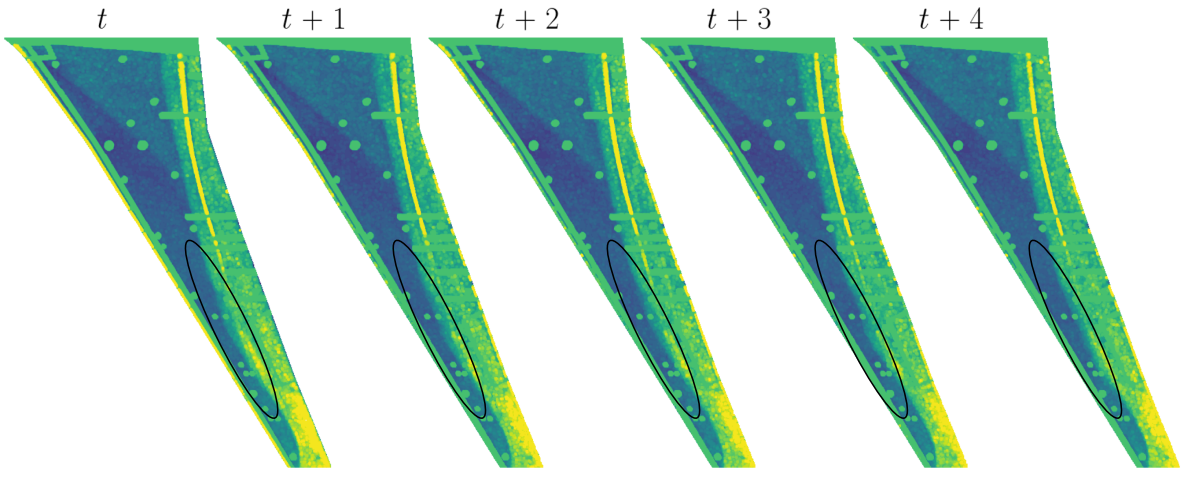


Figure 5.18: Surface pressure distribution c_p representing a buffet period on the wing suction side at $Ma_\infty = 0.84$, $Re = 25$ Mio. and $\alpha = 4°$ (iPSP).

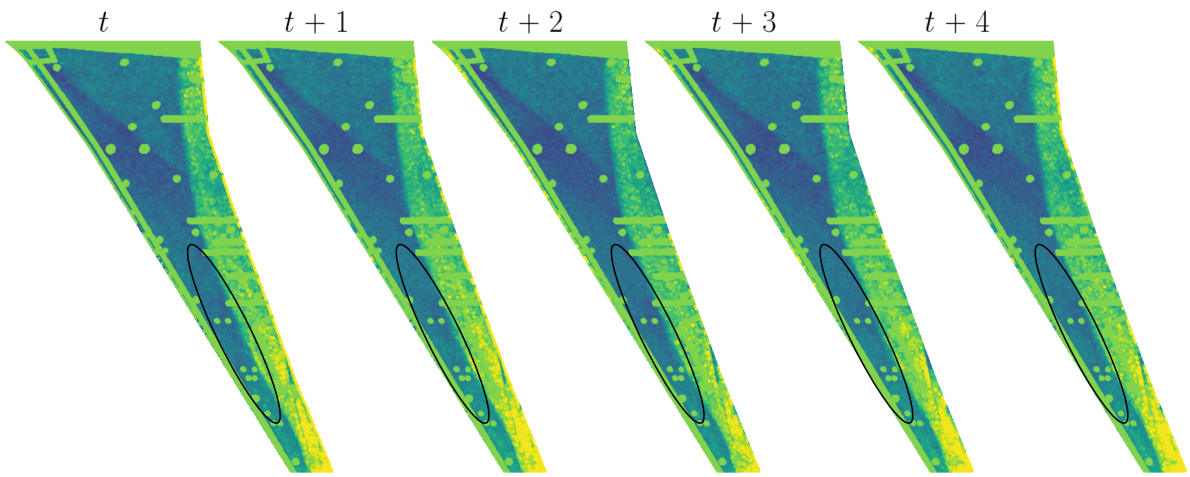


Figure 5.19: Surface pressure distribution c_p representing a buffet period on the wing suction side at $Ma_\infty = 0.9$, $Re = 25$ Mio. and $\alpha = 4°$ (iPSP).

Since the largest buffet motion is indicated at $Ma_\infty = 0.84$ between 60% and 80% spanwise position, the signals obtained by the sensors KUP11002 and KUP11802 are used in the following for the computation of the spanwise shock motion. Therefore, the magnitude squared coherence C (see Equation 5.2) and the phase angle ϕ (see Equation 5.3) between both sensor signals are computed. In Figure 5.20 (left), the coherence and the phase angle between the sensors at $Ma_\infty = 0.84$, $Re = 25$ Mio. and $\alpha = 4°$ are visualized. For comparison, coherence and phase angle at $Ma_\infty = 0.84$, $Re = 25$ Mio. and $\alpha = 5°$ between both sensors are additionally depicted in Figure 5.20 (right).

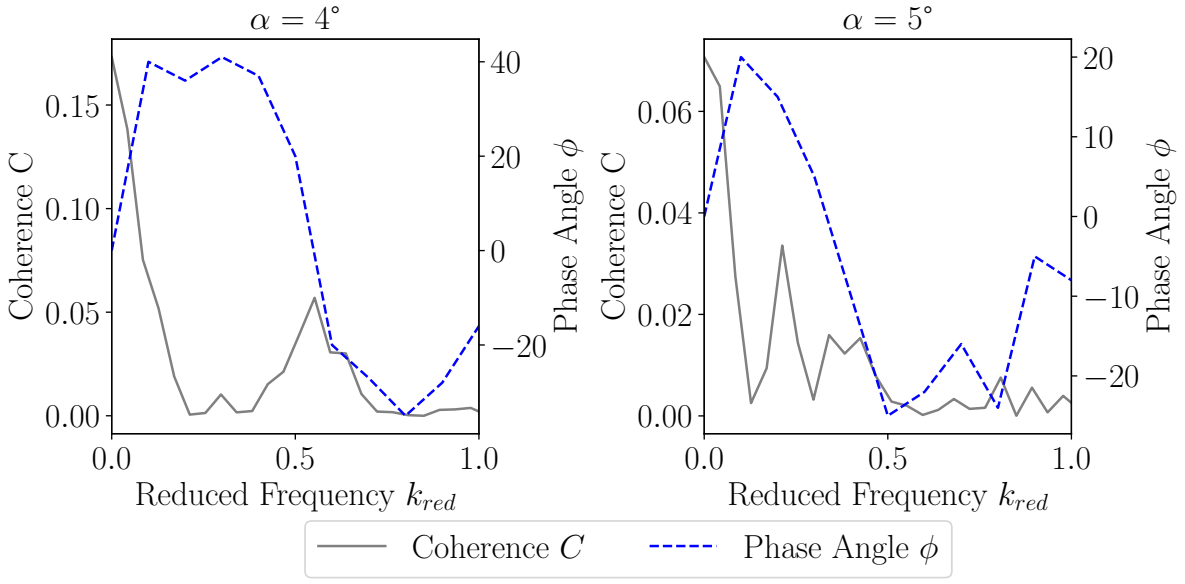


Figure 5.20: Magnitude squared coherence C and phase angle ϕ between pressure sensors KUP11002 and KUP11802 ($Ma_\infty = 0.84$, $Re = 25$ Mio., $\alpha = [4°,5°]$).

The range of frequencies, which are distinct for the propagation of the buffet motion, are defined by regions of high coherence [83]. For $\alpha = 4°$, high coherence is given at $k_{red} \approx 0.4$ - 0.6, whereas at $\alpha = 5°$ the highest coherence occurs at $k_{red} \approx 0.25$ - 0.5. At

the selected frequency ranges, the phase shift shows an almost linear behavior, indicating a constant buffet propagation speed. Since the slope of the phase angle is negative, the shock propagates outboard towards the wing tip.

Applying Equation 5.4, the propagation speed in spanwise direction is $U_s = 48.6$ m/s $= 0.27\ U_\infty$ for $\alpha = 4°$ and $U_s = 59.4$ m/s $= 0.33\ U_\infty$ for $\alpha = 5°$. Comparing the computed velocities with recent findings of numerical and experimental studies, a good agreement is indicated. Paladini et al. [93] evaluated velocities in the range of $U_s/U_\infty \approx 0.23$ - 0.26 on different wings with moderate sweep. More specifically, two different half wing/-fuselage body configurations based on the OAT15A airfoil, a half wing/fuselage body configuration from Dassault Aviation as well as a generic half wing/fuselage body model from the FLIRET (Flight Reynolds number testing) project have been investigated. A study by Masini et al. [83] obtained a similar value of $U_s/U_\infty = 0.26$, considering the RBC12 aircraft configuration. Sugioka et al. [146] derived a propagation speed of $U_s/U_\infty = 0.53$, studying a 80% scaled NASA CRM configuration. Further studies on the NASA CRM by Timme [127] and Ehrle et al. [30] revealed propagation speeds of $U_s/U_\infty = 0.26$ to 0.32 and $U_s/U_\infty = 0.24$ to 0.28, respectively. In Table 5.4, convection speeds as obtained by the studies mentioned above, are summarized.

Publication	Propagation Speed U_s/U_∞	Test Case
Paladini et al. [93]	0.26	OAT15A
Paladini et al. [93]	0.24	OAT15A
Paladini et al. [93]	0.23	Dassault model
Paladini et al. [93]	0.26	FLIRET model
Masini et al. [83]	0.26	RBC12 wing
Sugioka et al. [146]	0.53	80% CRM
Timme et al. [127]	0.26-0.32	CRM
Ehrle et al. [30]	0.24-0.28	CRM

Table 5.4: Overview of buffet wave-propagation speeds obtained by recent numerical and experimental studies.

Besides the computation of the buffet propagation speed in spanwise direction, the propagation speed in chordwise direction is additionally investigated. Therefore, the signals obtained by the sensor KUP10803 and KUP10802 are applied. Analogous to the investigation of the spanwise propagation speed, data obtained at $Ma_\infty = 0.84$, $Re = 25$ Mio. and $\alpha = [4°, 5°]$ are used. In Figure 5.21, coherence and phase angle between both sensors for $\alpha = 4°$ (left) and $\alpha = 5°$ (right) are depicted. As shown, for $\alpha = 4°$ an increase in coherence is indicated at $k_{red} \approx 0.2$ - 0.5, whereas at $\alpha = 5°$ the highest coherence occurs at $k_{red} \approx 0.3$ - 0.5. In addition, the phase shift at the frequencies with the highest coherence values is almost linear, which indicates a propagation speed in chordwise direction. Applying Equation 5.4, the propagation speed in chordwise direction is computed as $U_c = 43.2$ m/s $= 0.24\ U_\infty$ and $U_c = 52.2$ m/s $= 0.29\ U_\infty$ for $\alpha = 4°$ and $\alpha = 5°$, respectively.

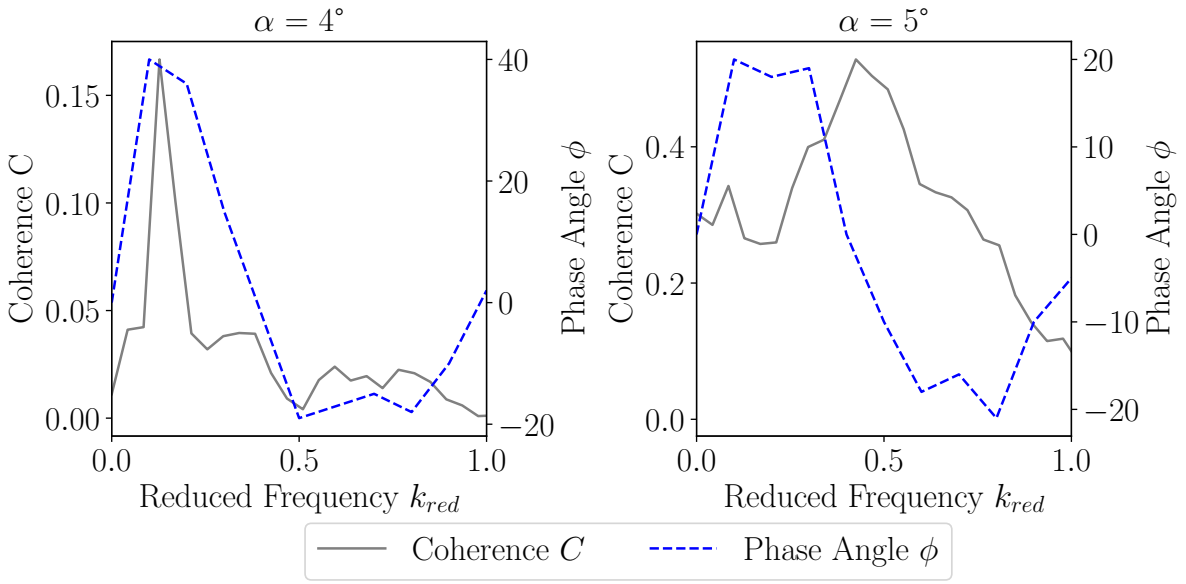


Figure 5.21: Magnitude squared coherence C and phase angle ϕ between pressure sensors KUP10802 and KUP10803 ($Ma_\infty = 0.84$, $Re = 25$ Mio., $\alpha = [4°,5°]$).

In Figure 5.22, a schematic summary of the propagation speeds as well as the related reduced frequencies in chord - and spanwise direction, is visualized.

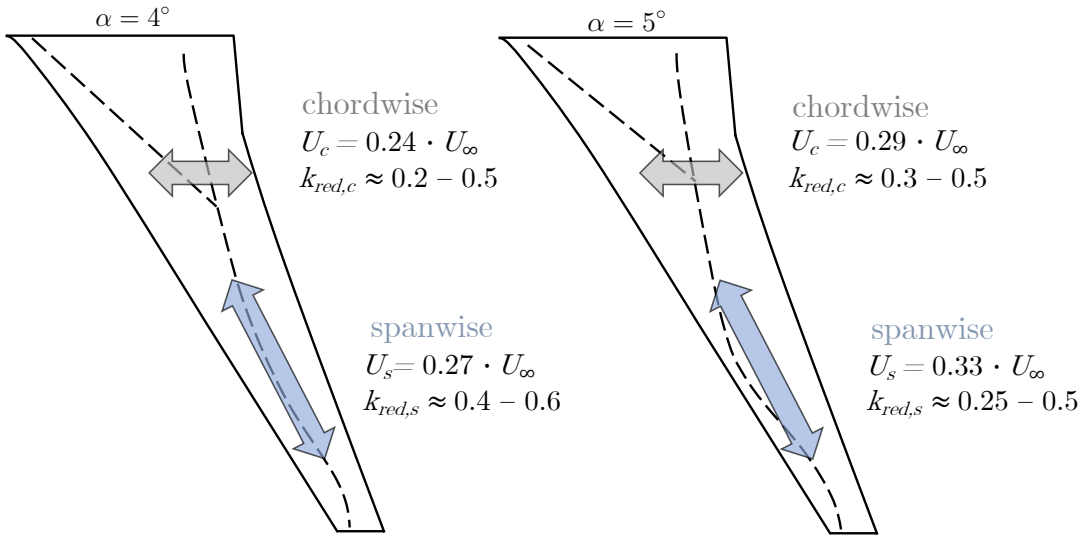


Figure 5.22: Buffet propagation speed and frequency in chord - and spanwise direction for $\alpha = 4°$ (left) and $\alpha = 5°$ (right) ($Ma_\infty = 0.84$, $Re = 25$ Mio.).

5.3 Prediction of Local Pressure Characteristics

Following the introduction of the Airbus XRF-1 wind tunnel model and the flow analysis, the following section presents the training and application process of a LSTM neural network for the prediction of local pressure characteristics measured by the unsteady pressure sensors. In the first part of this section, the preprocessing steps of the experimental data are outlined. The second section includes a detailed description of the LSTM training procedure as well as the selection of the hyperparameters. In the third part, the trained LSTM is applied for the prediction of unsteady pressure data, which is not included in the training data set. Here, pressure data obtained at different flow conditions and sensor positions are used for performance evaluation.

5.3.1 Data Preprocessing

Prior to the training of the LSTM, the experimental data sets need to be preprocessed. Based on the measurements in the ETW, the signals of each sensor are provided as sensor output voltage time-series. Therefore, in the first step, the data is processed into pressure readings, using calibration coefficients provided by the ETW test team.

In the second preprocessing step, the corresponding coefficient c_p measured at a location P and time t is calculated as follows

$$c_p(P,t) = \frac{p(P,t) - p_\infty}{\frac{1}{2}\rho_\infty U_\infty^2} \tag{5.5}$$

with test section static pressure, density and flow velocity p_∞, ρ_∞ and U_∞, respectively. Subsequent, the mean pressure $\bar{c}_p$ is subtracted from all samples, resulting in the pressure fluctuation coefficient $c'_p(P,t)$:

$$c'_p(P,t) = c_p(P,t) - \bar{c}_p(P,t) \tag{5.6}$$

In the last step, all data sets are normalized using the minimum and maximum values of the pressure fluctuations $c'_{p,min}(P,t)$ and $c'_{p,max}(P,t)$ as calculated from the data set. Therefore, all data values are rescaled to [-1,1].

5.3.2 Training of the LSTM-ROM

The aim of the LSTM neural network is the prediction of local pressure fluctuations at different span - and chordwise positions. For the training of the LSTM, pressure data obtained at two different flow conditions at a single sensor position is used. In particular, data measured at $Ma_\infty = 0.84$, $Re = 25$ Mio., and $\alpha = [4°, 5°]$ by the sensor KUP11002 ($\eta = 51.4\%$) is used. These flow conditions and the sensor position are selected since the spectra (see Figure 5.11, top right) clearly represented characteristic buffet frequencies, while at the same time the influence of structural frequencies is moderate.

Prior to the training process, the experimental data sets are split into two parts for training and validation purposes. Consistent as in Section 4.3, 70% of the data points included in each data set are taken for the training of the LSTM, whereas the remaining points are applied for validation. During the training, the pressure samples are fed into the LSTM in sequences, including 256 data points each. The batch size is defined as one. A stacked LSTM with three consecutive layers is applied, with each layer including 200 neurons per layer. Analogous to previous application cases of the LSTM, a hyperbolic tangent ($tanh$) is selected as the state activation function, while the gate activation function is chosen as sigmoid (σ). The initial learning rate is set to 10^{-4} and the training of the LSTM is terminated after 600 epochs. During the training, the MSE between the experimental pressure samples and the LSTM predictions, is minimized. In Figure 5.23, the convergence trends of the training and validation loss are visualized.

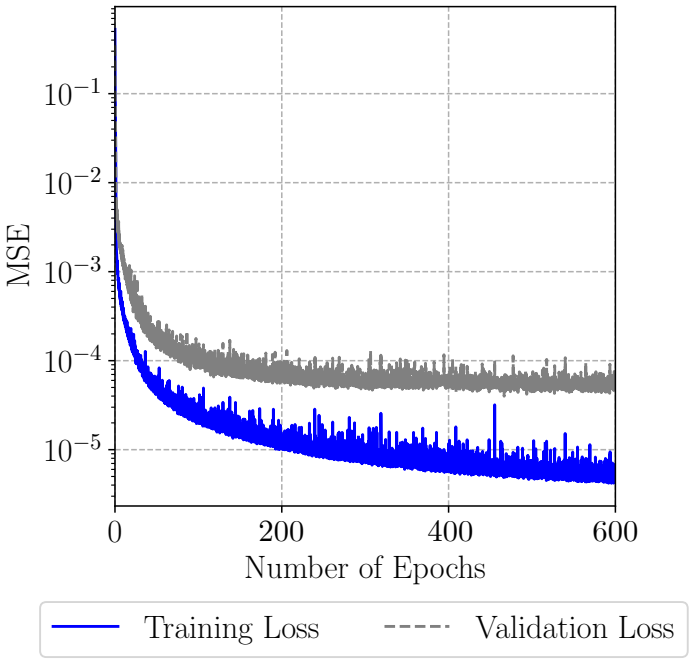


Figure 5.23: Convergence trends of training and validation loss.

In Table 5.5, the optimized hyperparameters for the training of the LSTM are summarized.

Hidden layers	3
Neurons per layer	200
Sequence length	256
Batch size	1
Initial learning rate	$1 \cdot 10^{-4}$
State activation	$tanh$
Gate activation	σ

Table 5.5: Hyperparameters of the LSTM.

5.3.3 Performance Evaluation

As a first step for the performance evaluation of the trained LSTM-ROM, the LSTM is applied for the prediction of unknown pressure samples obtained from sensor KUP11002 at $Ma_\infty = 0.84$, $Re = 25$ Mio. and $\alpha = [3.5°, 4.5°]$. Therefore, the LSTM is applied in a recurrent multi-step prediction mode. To start the iteration of the recurrent multi-step predictions, the first 100 time steps of the unknown sensor data are provided to the LSTM for initialization. As the multi-step prediction proceeds, the experimental data set input is successively substituted by pressure samples predicted by the LSTM itself.

In Figure 5.24 (left) and Figure 5.24 (right), a comparison of the experimental data and the data obtained by the LSTM at $\alpha = 3.5°$ and $\alpha = 4.5°$ is visualized, respectively. Therefore, the data is presented by means of the PSD of c_p, plotted over the reduced frequency k_{red}.

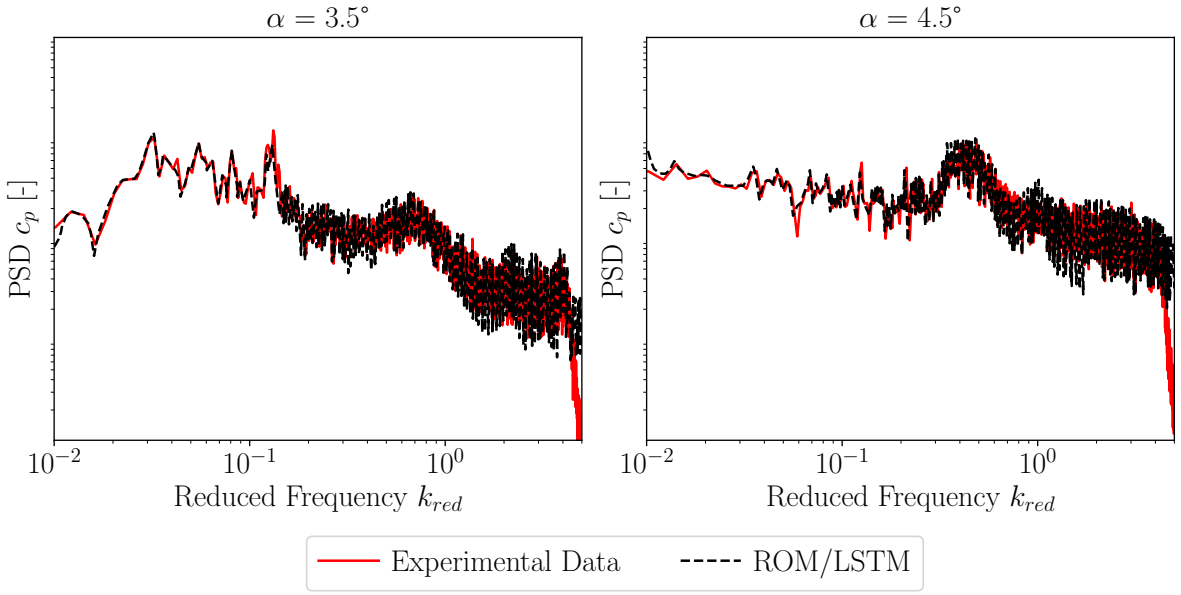


Figure 5.24: Comparison of experimental and predicted PSD of c_p at $\alpha = 3.5°$ (left) and $\alpha = 4.5°$ (right) ($Ma_\infty = 0.84$, $Re = 25$ Mio., KUP11002).

As shown, the trained LSTM is able to capture the trends of the pressure data at both flow conditions. Almost all frequency peaks are correctly represented, however, there is a slight shift towards lower amplitudes of the predicted data at higher frequencies. Further, the broadband frequency bump at $0.3 < k_{red} < 0.8$, which is related to buffet oscillations, is captured by the LSTM-ROM at both conditions.

In order to quantify the error between the experimental data and the data computed by the LSTM-ROM, the error metrics as introduced in Section 3.7 are applied. In Table 5.6, amplitude, phase and comprehensive error as well as the corresponding fit factor for the considered test cases are summarized. As shown, an overall fit of about 80% is achieved by the trained LSTM-ROM.

α [°]	A_{err}	P_{err}	C_{err}	Q
3.5	-0.098	0.084	0.11	80.4%
4.5	-0.087	0.092	0.11	80.4%

Table 5.6: Amplitude, phase, comprehensive error and fit factor of the LSTM-ROM applied to $\alpha = [3.5°, 4.5°]$ ($Ma_\infty = 0.84$, $Re = 25$ Mio., KUP11002)

Prediction of Pressure Characteristics at Varying Flow Conditions

Besides the application of the trained LSTM-ROM for the prediction of pressure data at different angles of attack, the LSTM-ROM is used for multi-step predictions of sensor data obtained at different flow conditions. Therefore, pressure samples obtained at $Ma_\infty = 0.9$, $Re = 25$ Mio, $\alpha = [4°,5°]$ and $Ma_\infty = 0.84$, $Re = 12.9$ Mio, $\alpha = [3°, 4°]$ are provided as an input for the multi-step computations.

As already outlined, the first 100 data points of each data set are provided as input for the LSTM-ROM predictions. In Figure 5.25 and Figure 5.26, the experimental result are compared to the LSTM-ROM predictions. Similar to the results presented in Figure 5.24, the trained LSTM-ROM is able to capture the pressure trends at each considered test condition. However, compared to the previous test cases, some frequency peaks are not captured correctly. Further, the amplitude shift between the experimental data and the predicted data points is larger, especially in the high frequency region. However, the broadband frequency bump related to aerodynamic buffet oscillations is captured for each of the considered test cases. Examining the corresponding errors of the selected test cases summarized in Table 5.7, slightly higher errors compared to the previous test cases result. However, the decrease in the fit factor is only around 2-3%.

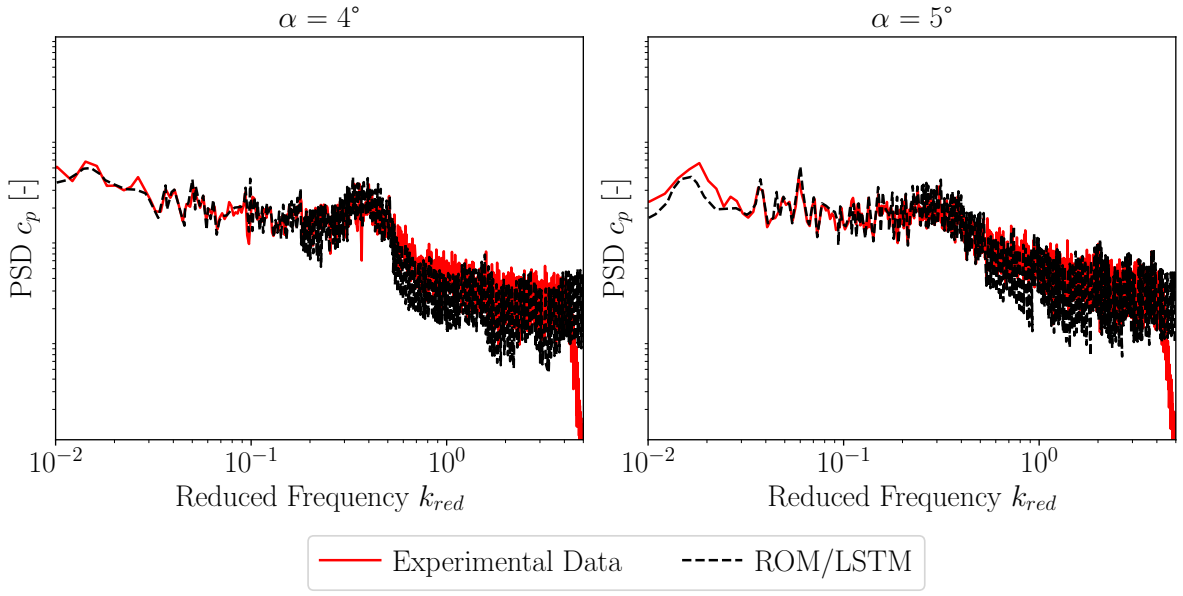


Figure 5.25: Comparison of experimental and predicted c_p at $\alpha = 4°$ (left) and $\alpha = 5°$ (right) ($Ma_\infty = 0.9$, $Re = 25$ Mio., KUP11002).

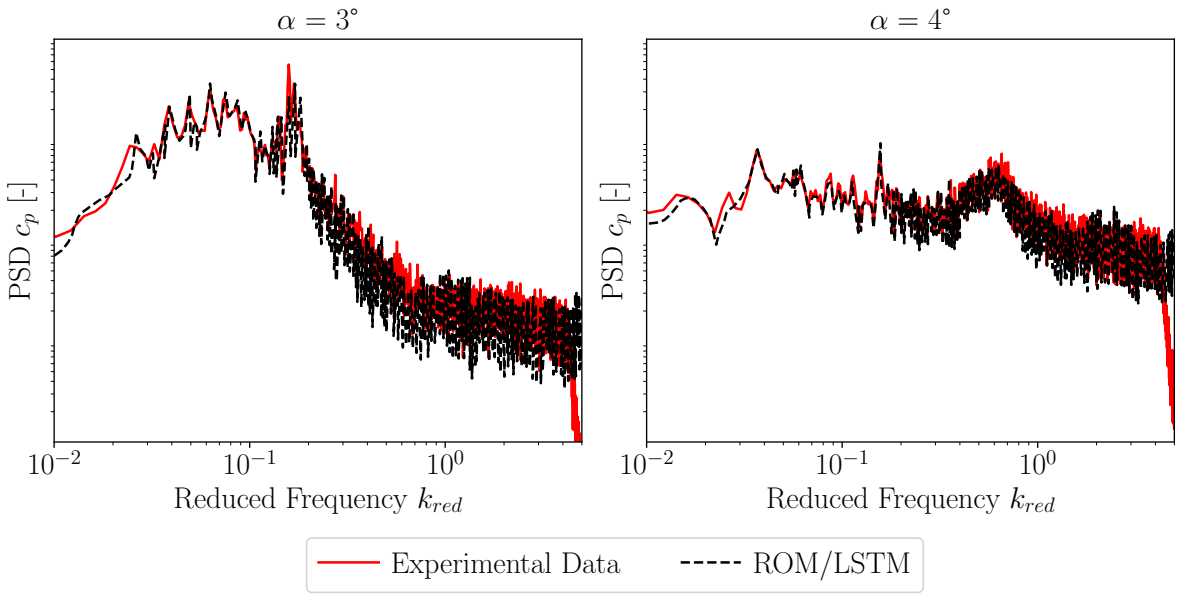


Figure 5.26: Comparison of experimental and predicted c_p at $\alpha = 3°$ (left) and $\alpha = 4°$ (right) ($Ma_\infty = 0.84$, $Re = 12.9$ Mio., KUP11002).

Ma_∞	Re [Mio.]	α [°]	A_{err}	P_{err}	C_{err}	Q
0.9	25	4	-0.102	0.087	0.12	79.3%
		5	-0.097	0.088	0.12	79.3%
0.84	12.9	3	-0.098	0.09	0.13	78.5%
		4	-0.097	0.094	0.12	79.3%

Table 5.7: Amplitude, phase, comprehensive error and fit factor of the LSTM-ROM applied to varying flow conditions at a fixed sensor position KUP11002.

Prediction of Pressure Characteristics at Varying Sensor Positions

In addition to the application of the trained LSTM towards test cases with varying flow conditions, the LSTM is applied for the prediction of pressure data as obtained at different sensor positions. Therefore, sensors which are located at the same spanwise position ($\eta = 51.4\%$) as well as sensors which are located closer to the wing root and wing tip, are selected. In particular, the following sensor positions are selected for performance evaluation: KUP11001, KUP10801, KUP11402, KUP11802. The flow condition of the test cases is selected as $Ma_\infty = 0.84$, $Re = 25$ Mio. and $\alpha = [4°, 5°]$. In Figure 5.27 and Figure 5.28, power spectra of the original sensor signals and the signals predicted by the LSTM are compared for each considered sensor position at $\alpha = 4°$ and $\alpha = 5°$, respectively. In addition, the mean pressure distribution of the considered flow condition is visualized, with a black and red marker indicating the selected sensor position and the reference training position, respectively.

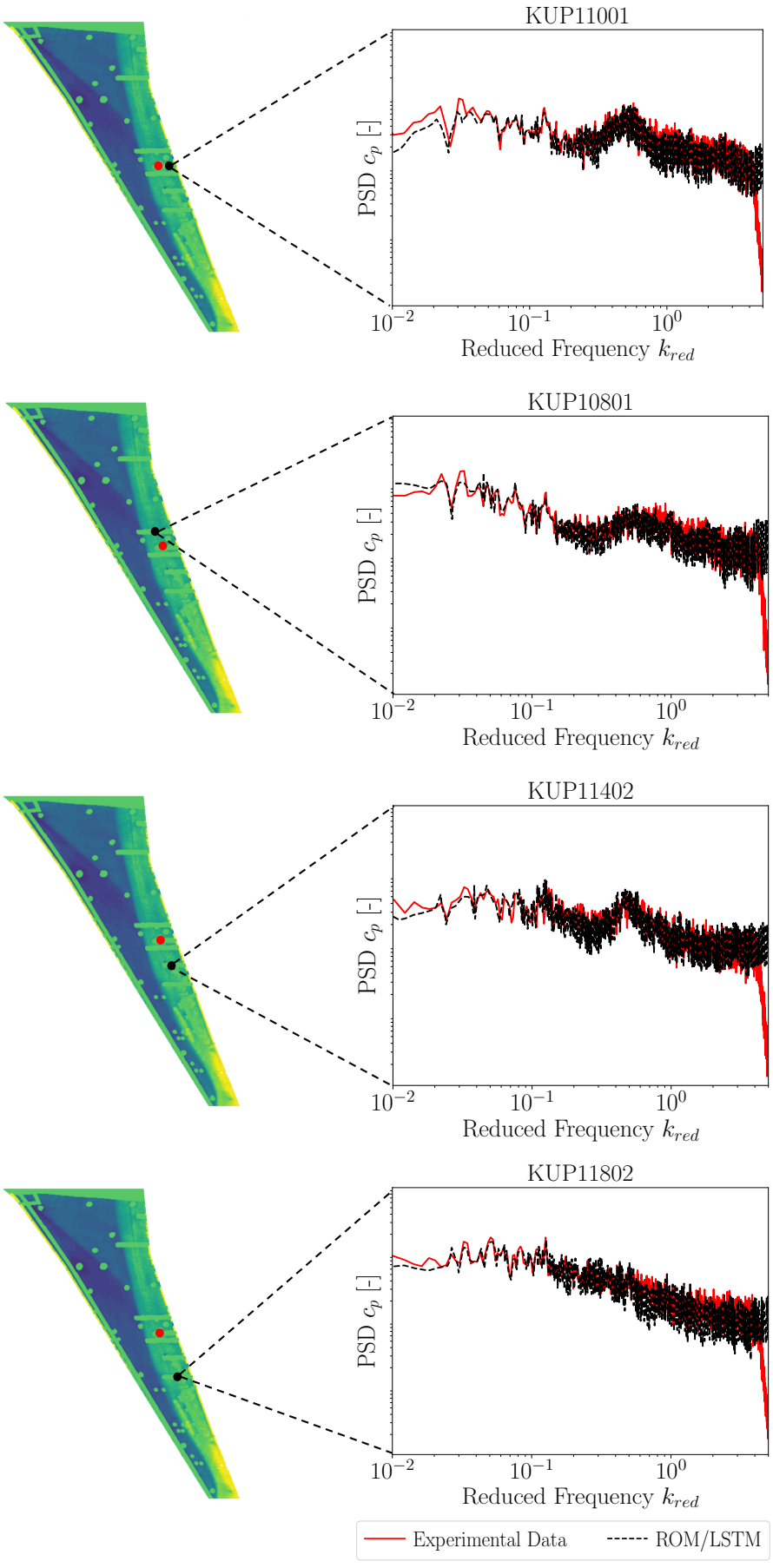


Figure 5.27: Comparison of experimental and predicted power spectral densities of c_p at varying sensor positions (black marker) ($Ma_\infty = 0.84$, $Re = 25$ Mio., $\alpha = 4°$). The reference training sensor location is marked in red.

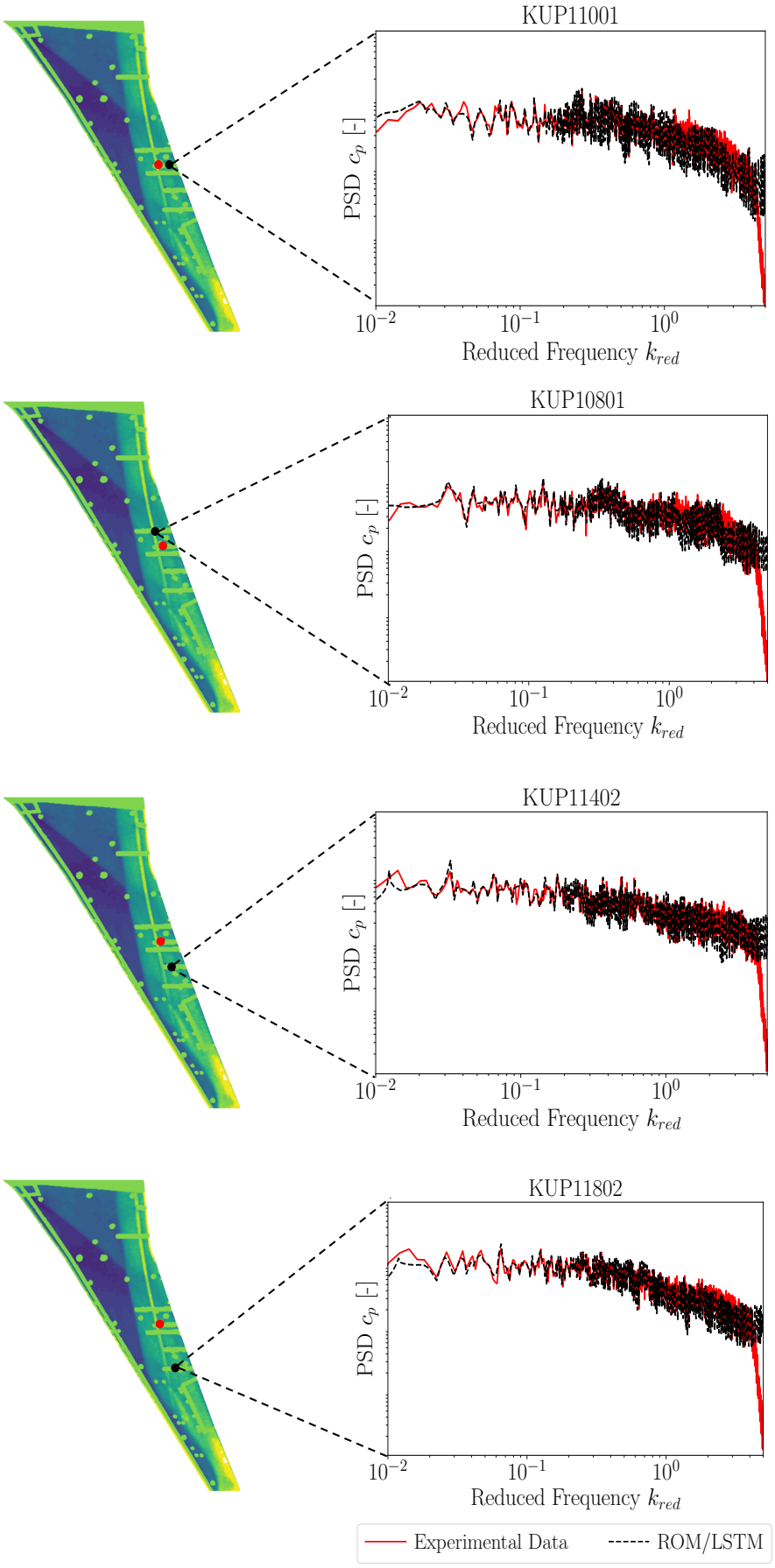


Figure 5.28: Comparison of experimental and predicted power spectral densities of c_p at varying sensor positions (black marker) ($Ma_\infty = 0.84$, $Re = 25$ Mio., $\alpha = 5°$). The reference training sensor location is marked in red.

Examining the results, an overall good agreement between the original and predicted signals is indicated. Compared to previous application cases, several peaks in the low frequency range are not correctly captured by the LSTM-ROM. However, the characteristic broadband buffet frequency bump is represented by the LSTM. Further, there is some indication of an amplitude shift in the high frequency range, as already shown for previous test cases. In Table 5.8, the errors and fit factors for each test case are summarized. Compared to the previous test cases, a slight decrease of about 3-4% in the overall fit in indicated.

α [°]	Sensor Position	A_{err}	P_{err}	C_{err}	Q
4	KUP11001	-0.11	0.093	0.127	77,5%
	KUP10801	-0.109	0.099	0.13	76,8%
	KUP11402	-0.115	0.1	0.135	75,8%
	KUP11802	-0.108	0.095	0.127	77,5%
5	KUP11001	-0.099	0.096	0.122	77,9%
	KUP10801	-0.11	0.102	0.132	76,2%
	KUP11402	-0.113	0.098	0.132	76,2%
	KUP11802	-0.114	0.1	0.134	76%

Table 5.8: Amplitude, phase, comprehensive error and fit factor of LSTM-ROM application for pressure data obtained at different sensor positions ($Ma_\infty = 0.84$, $Re = 25$ Mio., $\alpha = [4°, 5°]$).

Keeping in mind, that the LSTM-ROM has been trained using data sets including different frequency content, the overall prediction performance of the LSTM applied to various test cases is assumed to be accurate and efficient. Although some frequency peaks in both the low and high frequency range are not correctly captured, the overall trends of the sensor spectra are well covered. Further, the relevant broadband frequency peaks related to aerodynamic buffet oscillations are captured by the proposed ROM.

5.4 Prediction of Buffet Surface Pressure Distributions

Following the prediction of local buffet pressure characteristics on the basis of the measured unsteady pressure data, this section deals with the prediction of surface pressure distributions obtained by the iPSP measurements. Here, iPSP data from both the clean wing and the UHBR wing configuration are considered. For this task, a hybrid model based on the combination of a convolutional variational autoencoder (CNN-VAR-AE) and a LSTM neural network is developed and applied. In the first part, the architecture of the hybrid neural network is introduced. The second part covers a description of the preprocessing steps applied to the iPSP data sets, followed by the training process of the hybrid ROM. In the last part, application results of the trained hybrid ROM are presented and compared to the reference experimental data.

5.4.1 Hybrid ROM Architecture

The hybrid ROM applied for the following investigation is defined by the combination of a CNN-VAR-AE and a LSTM neural network. Here, the CNN-VAR-AE enables a reduction of the high-dimensional flow field data into a reduced latent space, whereas the LSTM is applied in order to predict the temporal evolution of the buffet instability. Therefore, the hybrid ROM is used for the prediction of buffet pressure distributions at time steps $k+1$ to $k+m$ based on several previous snapshots of the pressure distribution at time steps $k-n+1$ to k. Here, n denotes the number of training samples applied for the prediction, whereas m defines the number of time steps ahead which should be predicted by the trained model. The architecture of the hybrid model is illustrated in Figure 5.29.

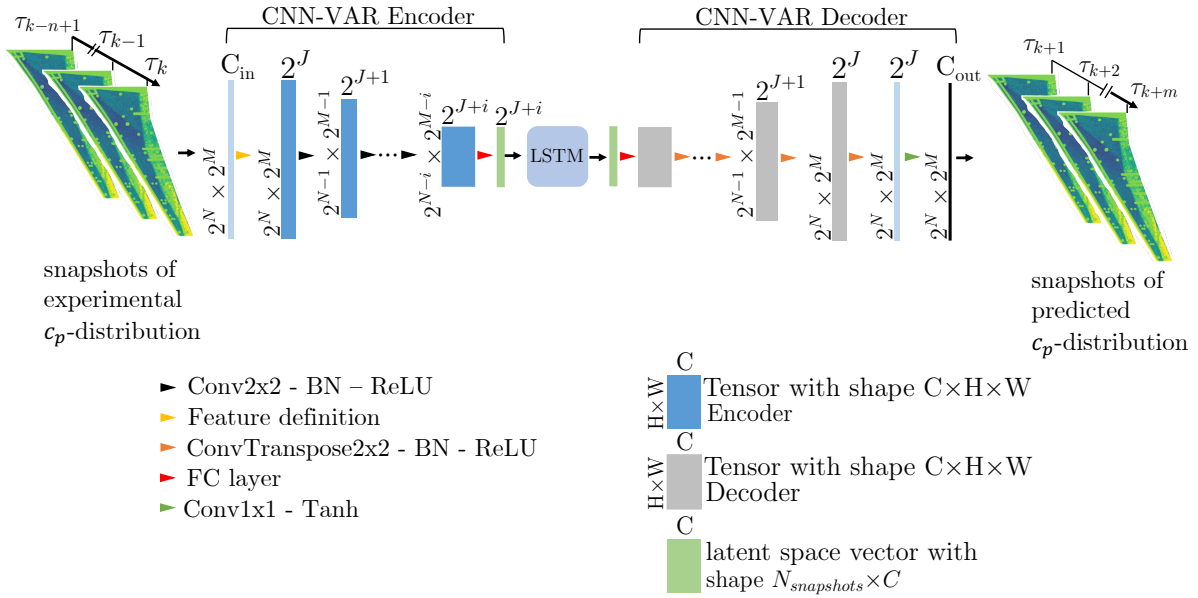


Figure 5.29: Architecture of the hybrid deep learning model.

The CNN-VAR-AE is defined by an encoder, decoder and a latent space vector. Both the encoder and decoder of the CNN-VAR-AE are divided into a pre-defined number of levels. At each level, several operations are performed on the input data, which is fed into the encoder. Defining the input of the CNN-VAR-AE, snapshots of pressure distribution $c_p(\tau,x,y)$ at each measured time step τ are combined into an array along the channel dimensions C. Therefore, the size of the input array is defined as $C \times H \times W$. H and W denote the spatial resolution of each c_p snapshot, which are defined by powers of two.

As the first processing step of the encoder, the number of input features, which is also referred to as the input channel dimension C_{in}, is upsampled to 2^J. The initial channel dimension is defined as three, including the time step τ and the spatial resolution (x,y) of each pressure coefficient snapshot $(c_p(\tau,x,y))$. J is set as an integer value in order to define each following channel dimension by a power of two. After the definition of the input features, a multi-channel convolution with a kernel size of 2×2 and stride $s = 2$ is performed. Following the convolution, the input is normalized by applying batch normalization (BN) (see Section 3.2.5). In the last encoder step, an activation function is applied to each element of the incoming array. Within the scope of this thesis, a rectified linear unit (ReLU) is used.

The sequence of operations described above is applied at each level of the encoder. Therefore, the output of each encoder level has twice as many channels as the input. In addition, the size of the spatial dimensions (H,W) is reduced by a factor of two at each level of the encoder. As the last encoder processing step, the data is fed into a FC layer in order to reduce the channel dimension. Since a variational AE is used, the resulting reduced latent features are sampled from a normal distribution.

After the input data has been processed by the encoder, the data is reshaped and fed into the LSTM. In Figure 5.30, the schematic of the multiple time steps ahead prediction obtained by the LSTM model is depicted.

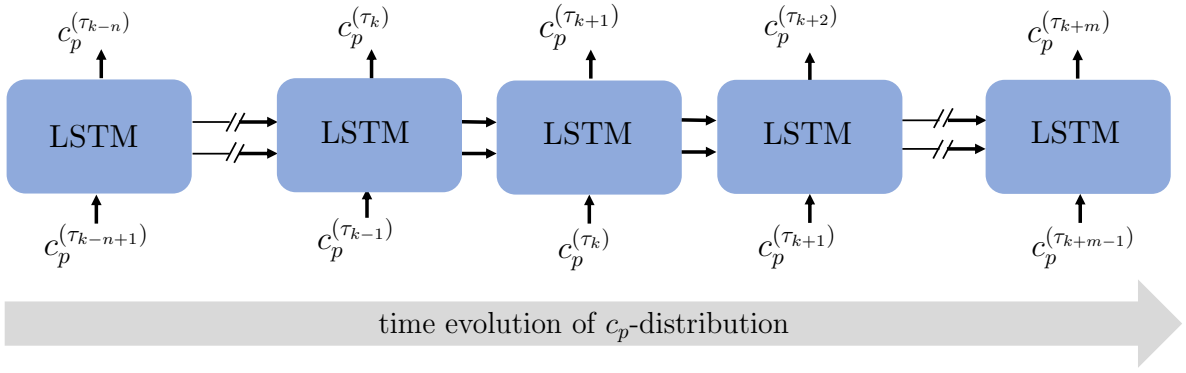


Figure 5.30: Prediction of time evolution of the c_p-distribution as obtained by the LSTM neural network.

As soon as the prediction of the LSTM is finalized, the output is fed into the decoder. Therefore, the low-dimensional data is reconstructed to the original high-dimensional flow field. At each level of the decoder, a transposed convolution is applied to upscale the spatial dimension. The corresponding kernel size and stride values are chosen equally to those of the encoder. In addition, BN is applied at each level of the decoder. In order to obtain the final prediction, an activation function is applied to the output of the last decoder level. Therefore, a hyperbolic tangent ($tanh$) is chosen, which reshapes all predicted elements to [-1,1].

5.4.2 Data Preprocessing

In order to feed the experimental data obtained from iPSP measurements into the deep learning model, the data needs to be preprocessed accordingly. All preprocessing steps are accomplished using the Python library flowTorch [133]. Based on the geometry of the XRF-1 wind tunnel model and the iPSP measurement technique, the pressure distribution on the upper wing surface is discretized by 465×159 data points.

Due to measurement errors during the wind tunnel measurement campaign, the data set includes a small amount of non physical c_p values. Therefore, in the first preprocessing step, the experimental data is cleaned by applying a weight mask, which defines values of $c_p \geq 1.5$ as 1 and values of $c_p \leq -1.5$ as 0.

In the second step, the number of data points representing the pressure distribution is downscaled by linear interpolation from 465×159 to 256×128 ($2^8 \times 2^7$). In Figure 5.31, a comparison of the pressure distribution on the clean wing configuration represented by the original amount of data points (left) and the interpolated data points (right) is provided.

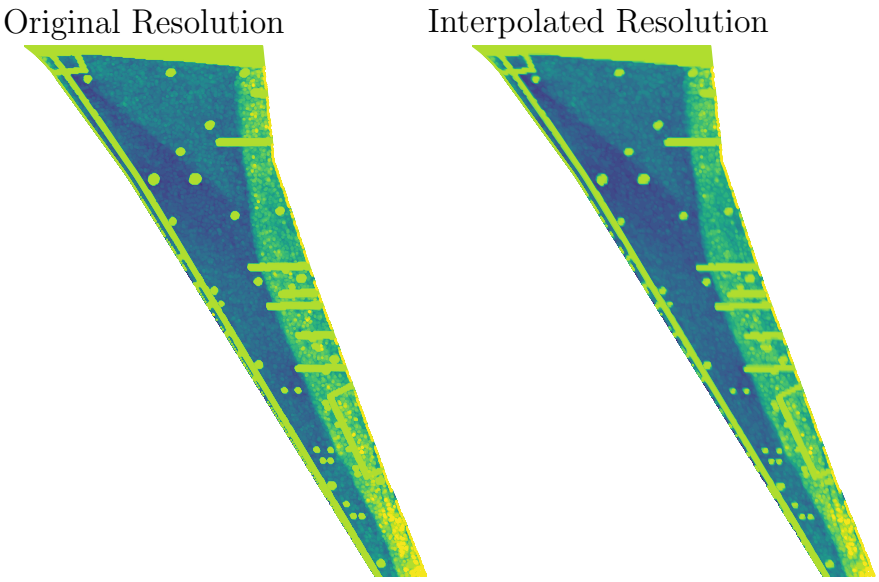


Figure 5.31: Original (left) and interpolated (right) number of data points representing the buffet pressure load distribution on the upper wing surface of the XRF-1 configuration (clean wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., $\alpha = 4°$).

The reduced resolution still maintains a high level of detail of the spatial resolution. The characteristic λ-shaped two shock pattern along the wingspan is still clearly visible.

In the final step, the data set is normalized based on the minimum and maximum pressure values in the data set ($c_{p,min}$,$c_{p,max}$). Therefore, the resulting value range of all c_p-snapshots is rescaled to [-1,1].

5.4.3 Training of the Hybrid ROM

The training of the hybrid deep learning model includes two consecutive steps. In the first step, the CNN-VAR-AE is trained independently from the LSTM, using c_p - snapshots representing one or more flow conditions. In the second step, the LSTM is trained based on a set of reduced c_p-snapshots, which have been encoded by the trained CNN-VAR-AE.

Since the application of the trained hybrid ROM aims for the prediction of buffet pressure distributions at unknown flow conditions, snapshots representing several flow conditions are applied for the training. Within the scope of this thesis, the focus is on the prediction of buffet pressure characteristics considering a flow condition measured at a different angle of attack as applied for the training of the model.

In order to evaluate the performance of the hybrid ROM for the computation of various unsteady flow fields, iPSP data obtained from both the clean wing and the UHBR wing configuration are applied for ROM training and evaluation. Therefore, two hybrid deep learning models are individually trained and applied.

The CNN-VAR-AE is trained using snapshots representing two different flow conditions, defined by $Ma_\infty = 0.9$, $Re = 25$ Mio. and $\alpha = [4°, 6°]$. For the training of the CNN-VAR-AE, in total 2000 c_p-snapshots are considered, including 1000 c_p-snapshots for each flow condition. For hyperparameter tuning and validation, in total 400 c_p-snapshots are used. During the training, the c_p-snapshots are fed into the encoder in batches, including 128 time steps each. For both the encoder and decoder, four convolution levels are applied. By means of the convolution operation, the input array size is reduced from 256×128 to 32×16, while the channel size is increased from 3 to 512. By passing the FC layer, the channel size is downscaled from 512 to 256 features. The initial learning rate is set to 10^{-4} and the training of the CNN-VAR-AE for the clean wing and UHBR wing data is terminated after 1000 and 1500 epochs, respectively. In order to achieve an appropriate model performance, the MSE between the reference experimental data and the predictions, is minimized. In Figure 5.32, training and validation losses of the CNN-VAR-AEs trained with the clean wing data (left) and the UHBR wing data (right) are visualized. In Table 5.9, the hyperparameters for the training and application of both CNN-VAR-AEs are summarized.

Besides the evaluation of the corresponding convergence trends, the training performance of both CNN-VAR-AEs is assessed based on a visual comparison of an original

c_p-snapshot included in the validation data set and the corresponding c_p-snapshot predicted by the CNN-VAR-AE.

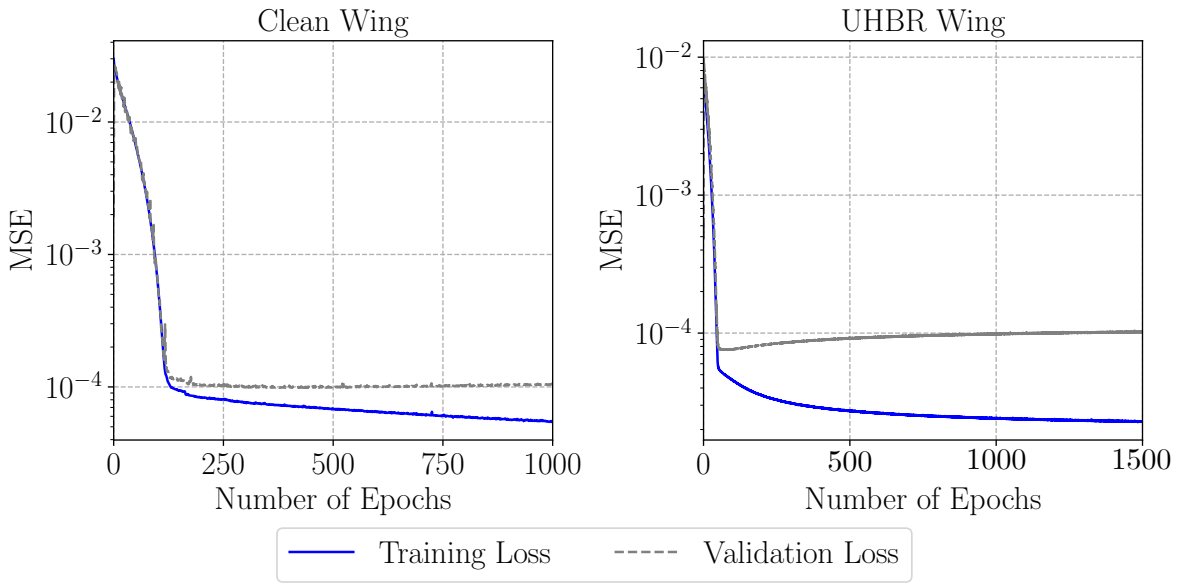


Figure 5.32: Convergence trends of training and validation losses of the individually trained CNN-VAR-AEs (left: clean wing, right: UHBR wing).

Encoder convolution layers	4
Decoder convolution layers	4
Latent dimension	256
Kernel size	2x2
Stride	2
Batch size	128
Initial learning rate	$1 \cdot 10^{-4}$
Activation function	ReLU

Table 5.9: Hyperparameters for the training of the CNN-VAR-AEs.

Therefore, one original snapshot and the corresponding predicted snapshot are exemplary visualized in Figure 5.33 and Figure 5.34 for the clean wing and UHBR wing configuration, respectively. Here, an angle of attack of $\alpha = 4°$ ($Ma_\infty = 0.9$, $Re = 25$ Mio.) is considered for both cases. In addition to the c_p - distribution, the MSE between the original and predicted c_p - distribution is visualized in Figure 5.33 and Figure 5.34 on the right wing surface. Here, the legend only refers to the MSE. Based on a visual comparison and the corresponding level of the MSE, which is almost zero on the entire wing, an overall sufficient training is indicated.

Equal to the training of the CNN-VAR-AE, 2000 snapshots are used for training the LSTM. The overall amount of training snapshots is divided in sequences of 128 snapshots, while the batch size is defined as one. For the training, a stacked LSTM with two layers is applied, with each layer including 256 neurons.

Analogous to the training of the CNN-VAR-AE, the initial learning rate is defined as 10^{-4}. The LSTM is trained for 5000 epochs, until a sufficient convergence is reached. In Figure 5.35, the convergence trends for the individually trained LSTMs are visualized. In Table 5.10, the hyperparameters for the training and application of both LSTM-ROMs are summarized.

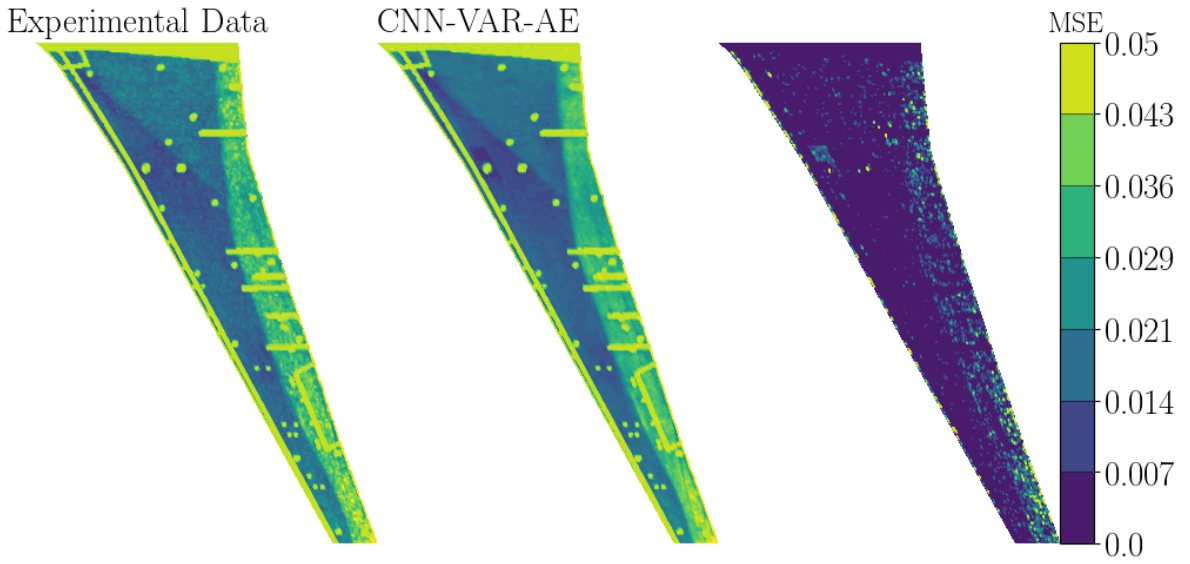


Figure 5.33: Comparison of an original validation c_p-snapshot and a c_p-snapshot predicted by the trained CNN-VAR-AE (clean wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., $\alpha = 4°$).

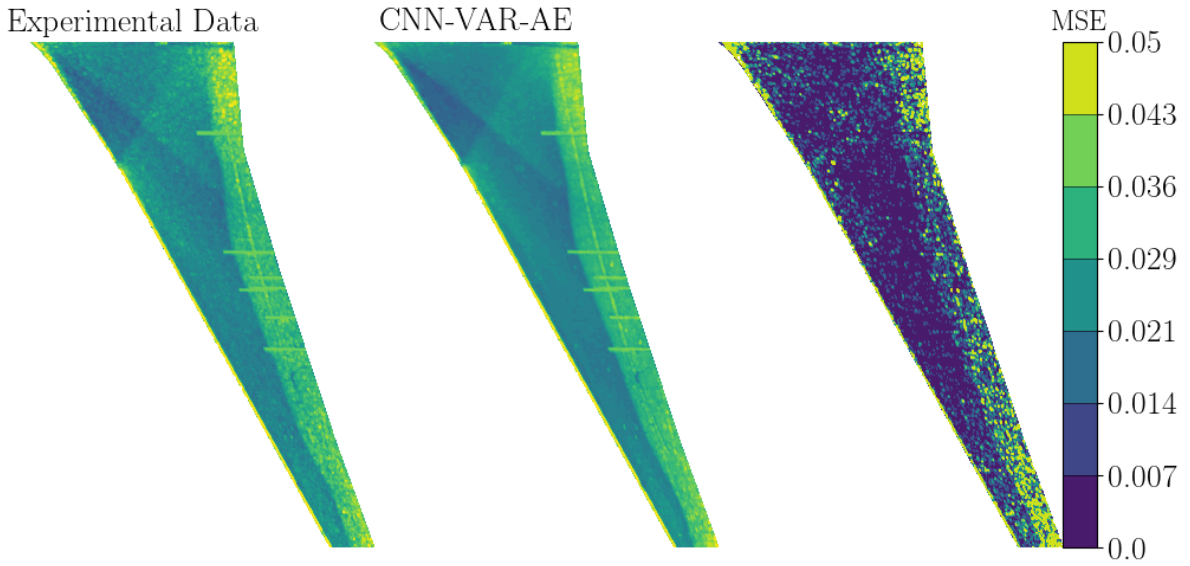


Figure 5.34: Comparison of an original validation c_p-snapshot and a c_p-snapshot predicted by the trained CNN-VAR-AE (UHBR wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., $\alpha = 4°$).

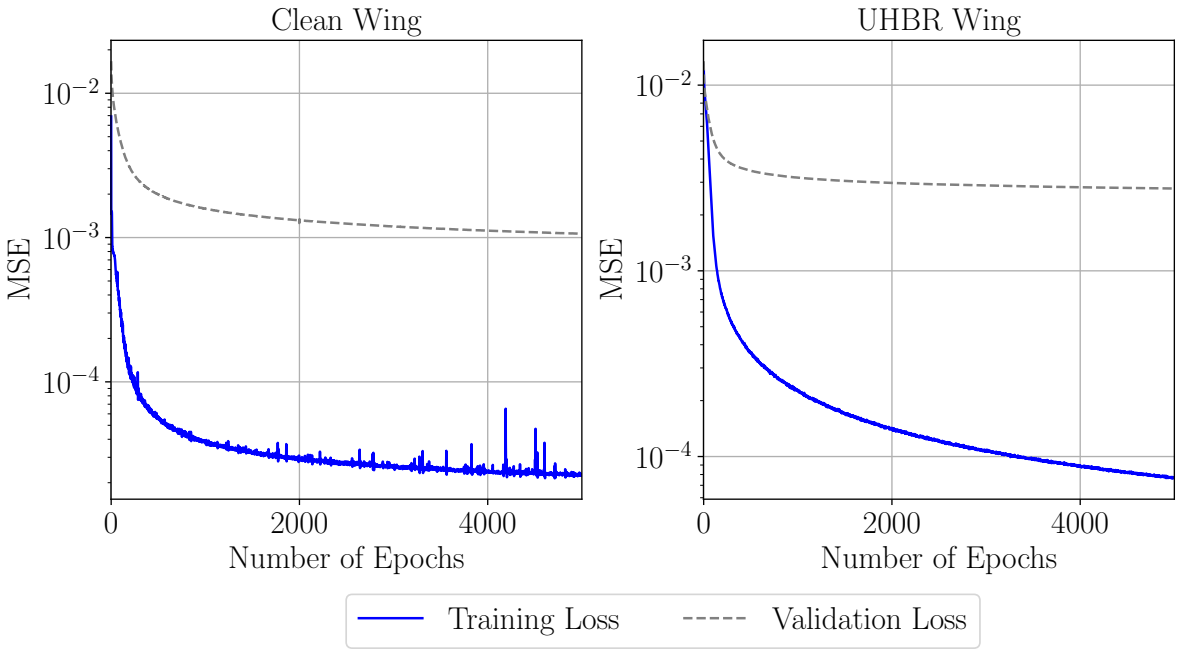


Figure 5.35: Convergence trends of training and validation losses of the individually trained LSTM (left: clean wing, right: UHBR wing).

Hidden layers	2
Neurons per layer	256
Batch size	128
Initial learning rate	$1 \cdot 10^{-4}$
State activation	$tanh$
Gate activation	σ

Table 5.10: Hyperparameters for the training of both LSTM-ROMs.

5.4.4 Performance Evaluation

In order to evaluate the performance of the individually trained hybrid ROMs, the ROMs are applied for the prediction of buffet pressure distributions at flow conditions which are not included in the training data set. Since c_p-snapshots obtained at $Ma_\infty = 0.9$, $Re = 25$ Mio. and $\alpha = [4°, 6°]$ are used for training, c_p-snapshots representing $Ma_\infty = 0.9$, $Re = 25$ Mio. and $\alpha = 5°$ are applied for performance evaluation. Analogous to the training procedure, the performance evaluation of the hybrid ROM is performed in two consecutive steps. First, the trained CNN-VAR-AEs are applied for the reconstruction of the unknown buffet pressure loads. In the second step, the pressure samples are encoded and used as an initialization input for the LSTM. In Figure 5.36 and Figure 5.37, a comparison between an original and reconstructed c_p-snapshot of the clean wing and UHBR wing configuration, is visualized, respectively. In addition, the MSE between the original and predicted c_p-distribution is presented in both figures.

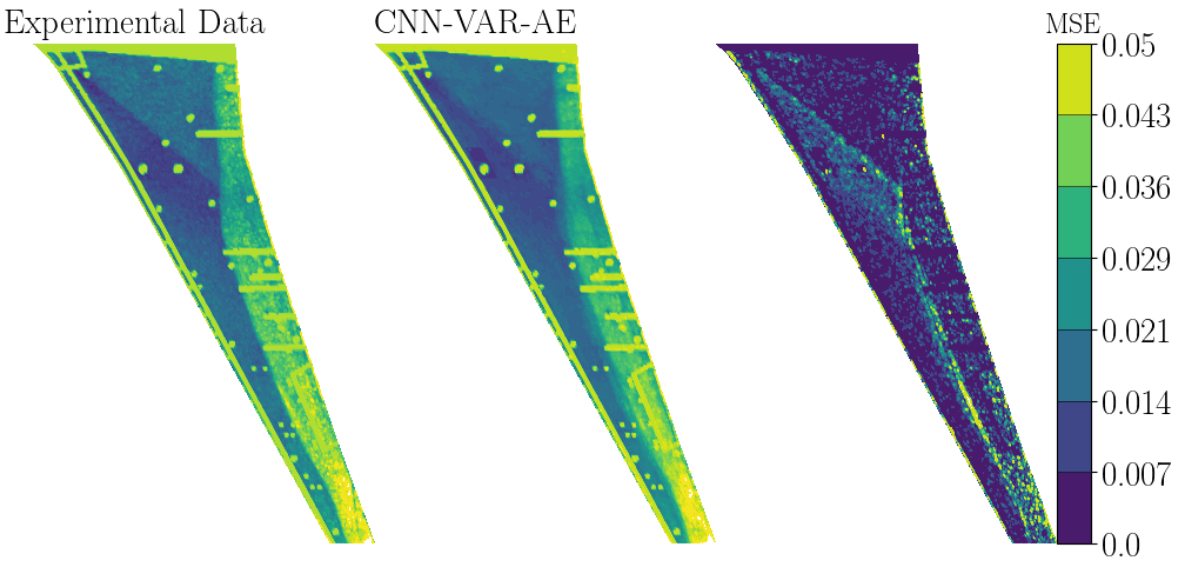


Figure 5.36: Comparison of an original c_p-snapshot and a c_p-snapshot predicted by the trained CNN-VAR-AE (clean wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., $\alpha = 5°$).

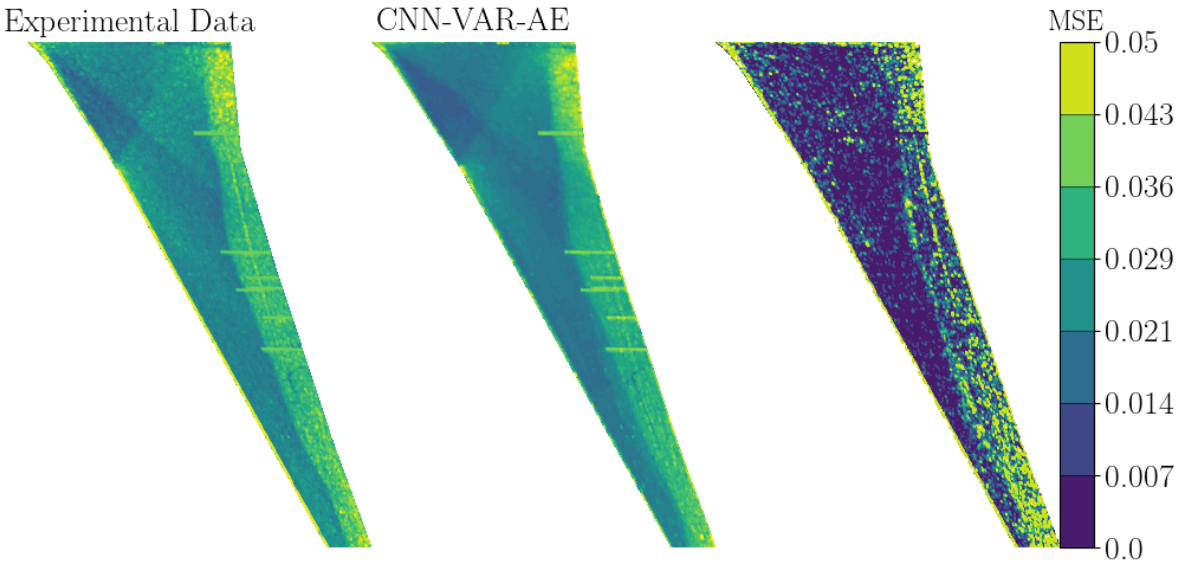


Figure 5.37: Comparison of an original c_p-snapshot and a c_p-snapshot predicted by the trained CNN-VAR-AE (UHBR wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., $\alpha = 5°$).

Examining the c_p-distribution as computed by the trained CNN-VAR-AEs, an overall good agreement is indicated. The characteristic λ-shaped shock patterns, as discussed in Section 5.2.1, are correctly captured by the CNN-VAR-AE for both experimental configurations. Although the chord - and spanwise position of the shock is in good agreement with the original data, larger MSE values for both test cases yield slight deviations along the shock position, especially in spanwise direction. However, it has to be emphasized that the application of the ROM aims for representing the characteristics buffet flow

physics, such as the propagation speed in chord- and spanwise direction as well as dynamic loads in terms of c_p, rather than a completely correct local representation of the shock position on the wing.

In order to identify if the trained CNN-VAR-AEs are able to reproduce the main buffet flow physics to some extent, both experimental and computed c_p-snapshots are compared by applying a proper orthogonal decomposition (POD). By using POD, important modes of the buffet instability are extracted. The order of occurrence of the modes yields the level of contribution to the buffet flow. In Figure 5.38 and Figure 5.39, power spectra of the first six POD modes of the experimental clean wing and UHBR wing data and the corresponding predictions of the trained CNN-VAR-AEs, are visualized, respectively. For the POD, 500 c_p-snapshots are applied. As shown in Figure 5.38 and Figure 5.39, the trained CNN-VAR-AEs are able to capture both the low and high frequency peaks of each mode with a high degree of accuracy. At higher modes, some peaks are not correctly represented. However, the overall trend is predicted by the trained CNN-VAR-AEs.

In order to evaluate the performance of both the trained CNN-VAR-AE and LSTM, the ROM is applied for the prediction of surface pressure distributions, which represent the temporal evolution of the buffet instability. Therefore, the test data sets are encoded by the trained CNN-VAR-AEs and processed as an input to the LSTM. For the initialization of the recurrent multi-step predictions, the first 32 encoded timesteps are provided to the LSTM. The LSTM is applied for the prediction of 250 timesteps ahead.

Analogous to the performance evaluation of the CNN-VAR-AEs, POD is applied for the performance assessment of the hybrid ROM. Therefore, a comparison between the experimental and predicted modes of the clean wing and UHBR wing data is depicted in Figure 5.40 and Figure 5.41, respectively. Here, only the first six modes are visualized, since deviations between the experimental and ROM solutions increase for higher modes. Examining the resulting spectra, a good agreement between the reference and ROM-based data is indicated. Similar to the results presented in Figure 5.38 and Figure 5.39, peaks in both the low and high frequency range are captured by the hybrid ROM. Besides the mode comparison, a visual comparison between experimental and predicted c_p-snapshots at two timesteps t = [150,200] obtained by the recurrent multi-step prediction mode is presented. In Figure 5.42 and Figure 5.43, the results of the clean wing data are presented, whereas Figure 5.44 and Figure 5.45 show the results of the UHBR wing data. As shown, besides the replication of the modes, the hybrid ROM is able to capture the surface pressure distribution on the wing, with only minor deviations in the position of the shock.

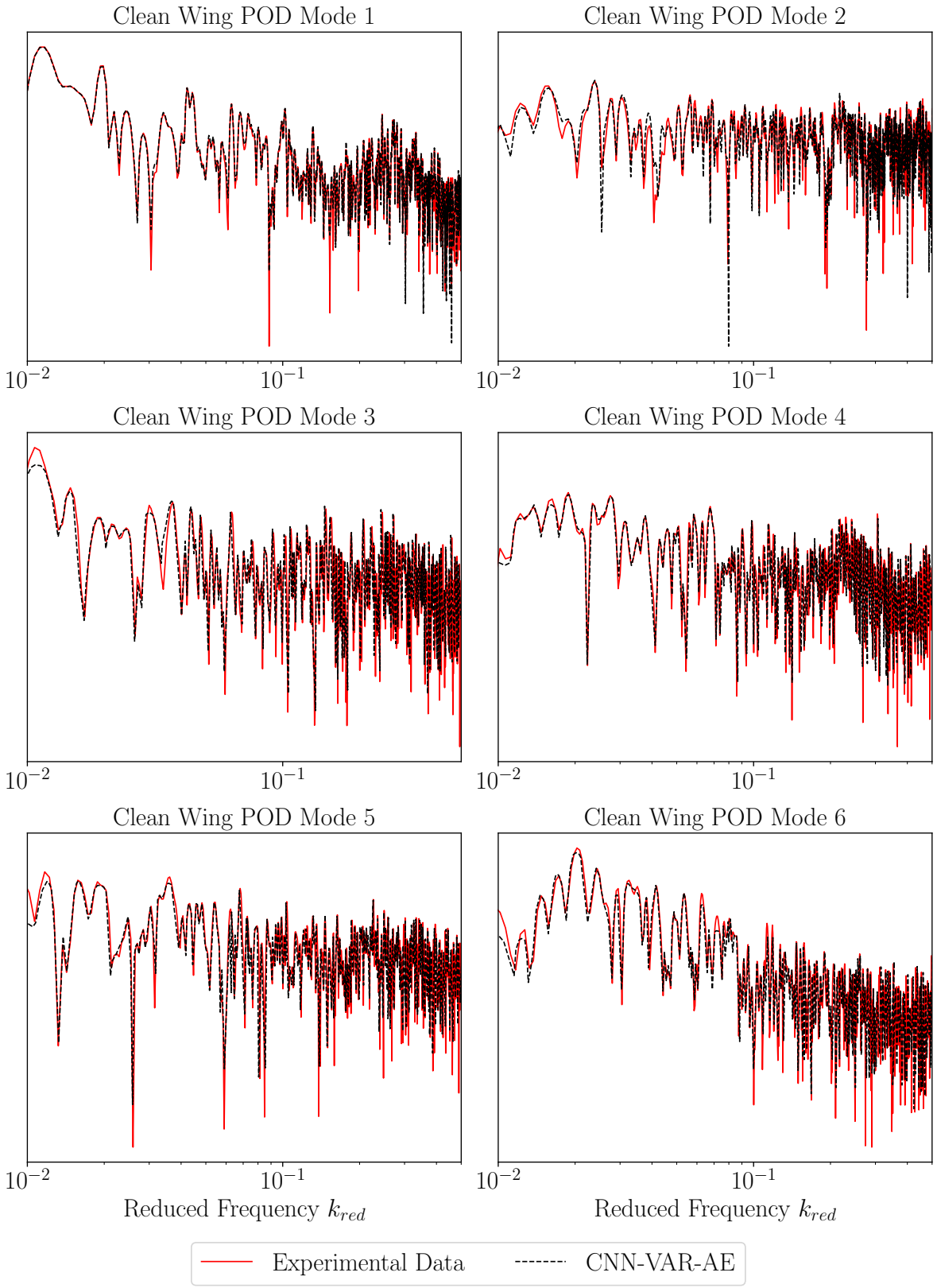


Figure 5.38: Power Spectra of the first six POD modes of the buffet cycle (clean wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., $\alpha = 5°$). The experimental results are compared to the results predicted by the CNN-VAR-AE.

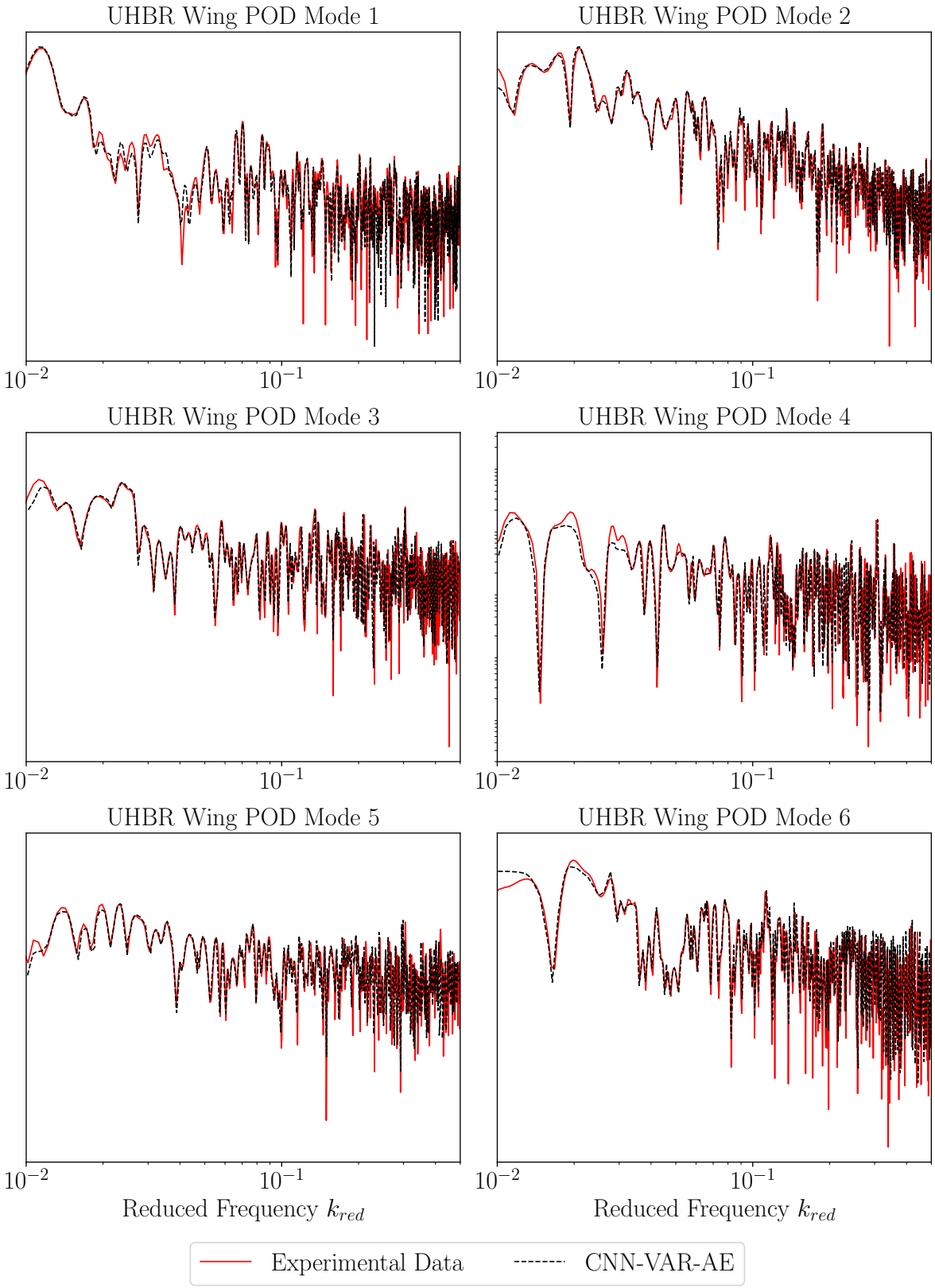


Figure 5.39: Power Spectra of the first six POD modes of the buffet cycle (UHBR wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., $\alpha = 5°$). The experimental results are compared to the results predicted by the CNN-VAR-AE.

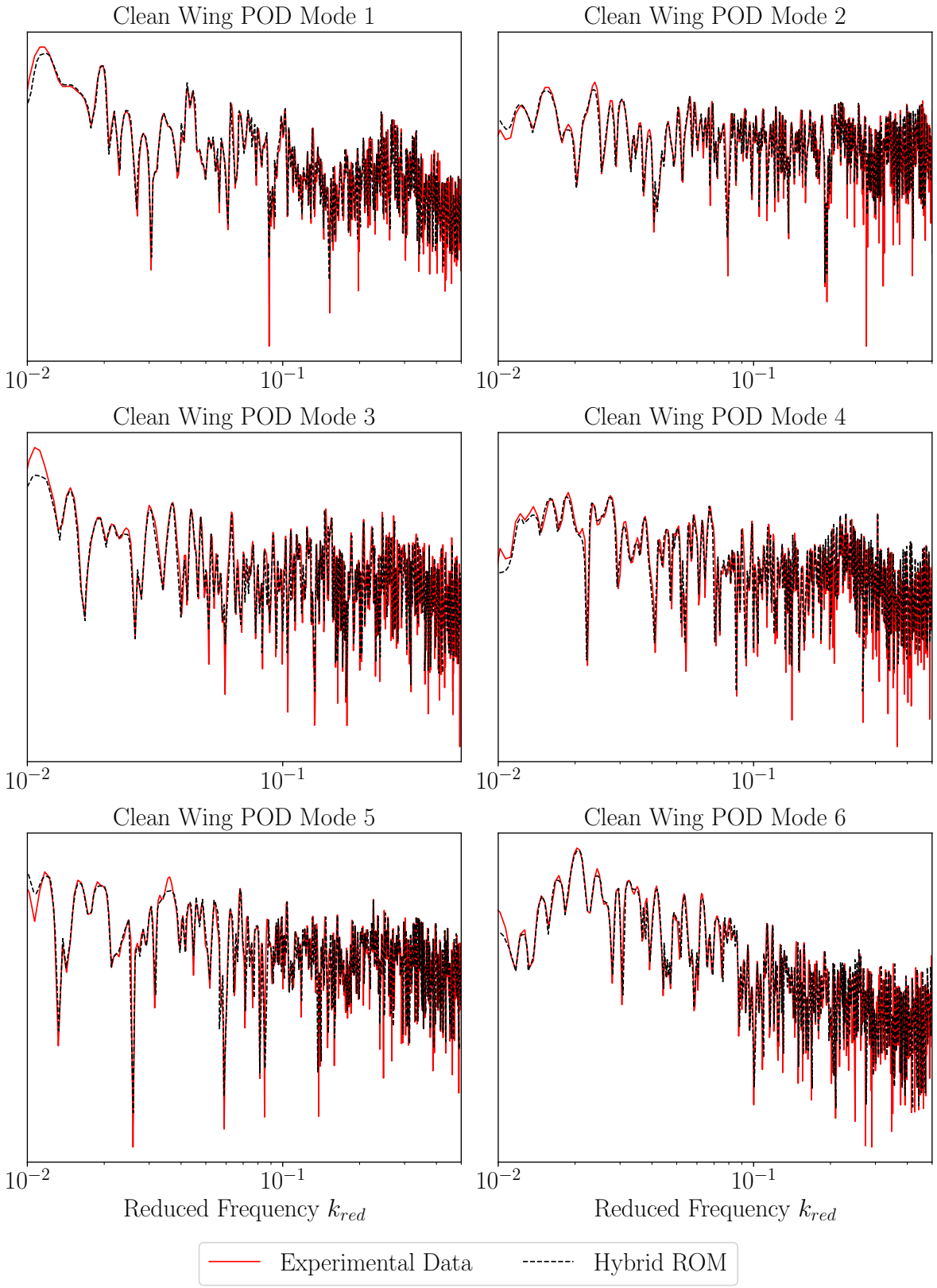


Figure 5.40: Power spectra of the first six POD modes of the buffet cycle (clean wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., $\alpha = 5°$). The experimental results are compared to the results predicted by the hybrid ROM.

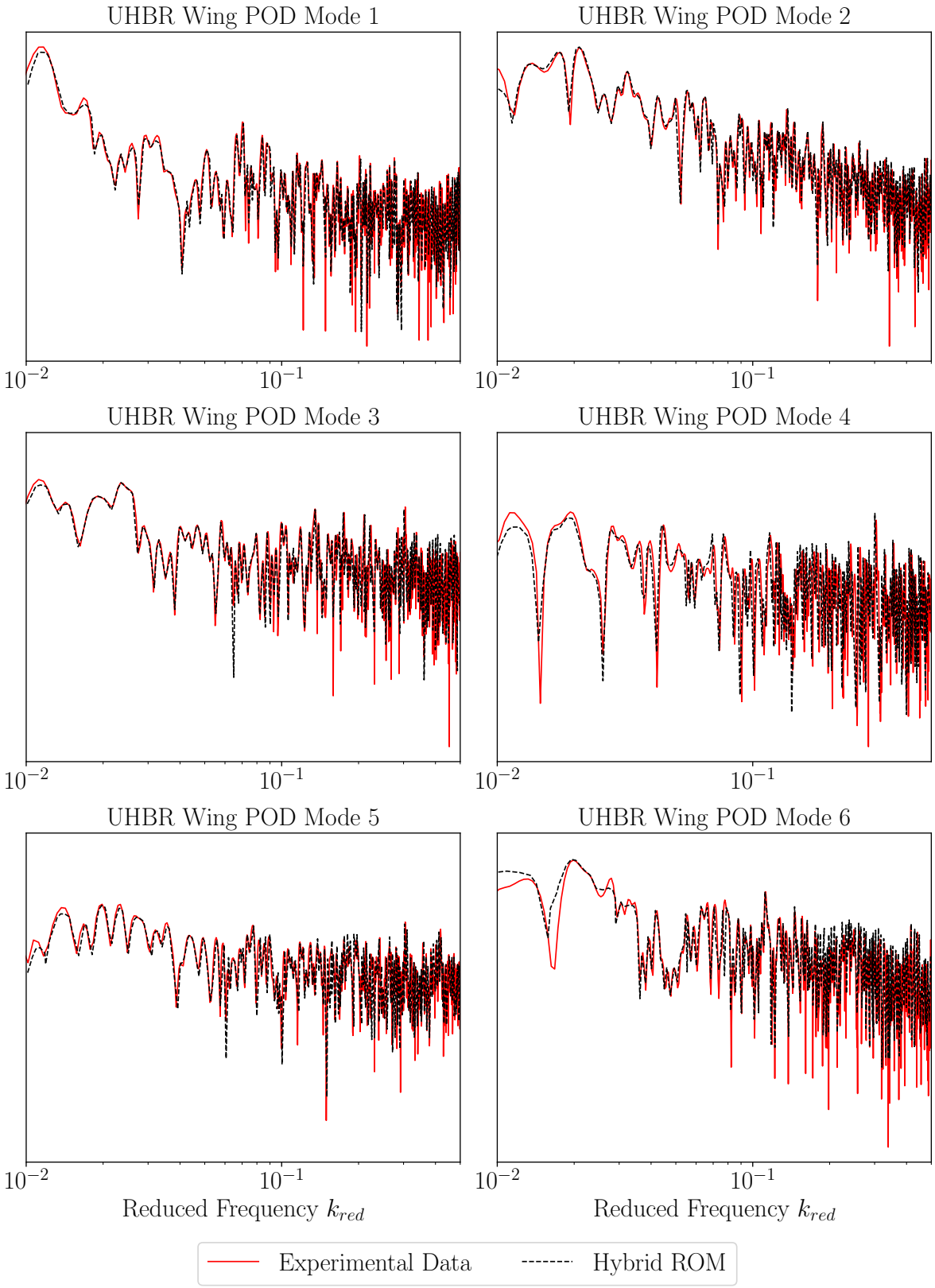


Figure 5.41: Power spectra of the first six POD modes of the buffet cycle (UHBR wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., $\alpha = 5°$). The experimental results are compared to the results predicted by the hybrid ROM.

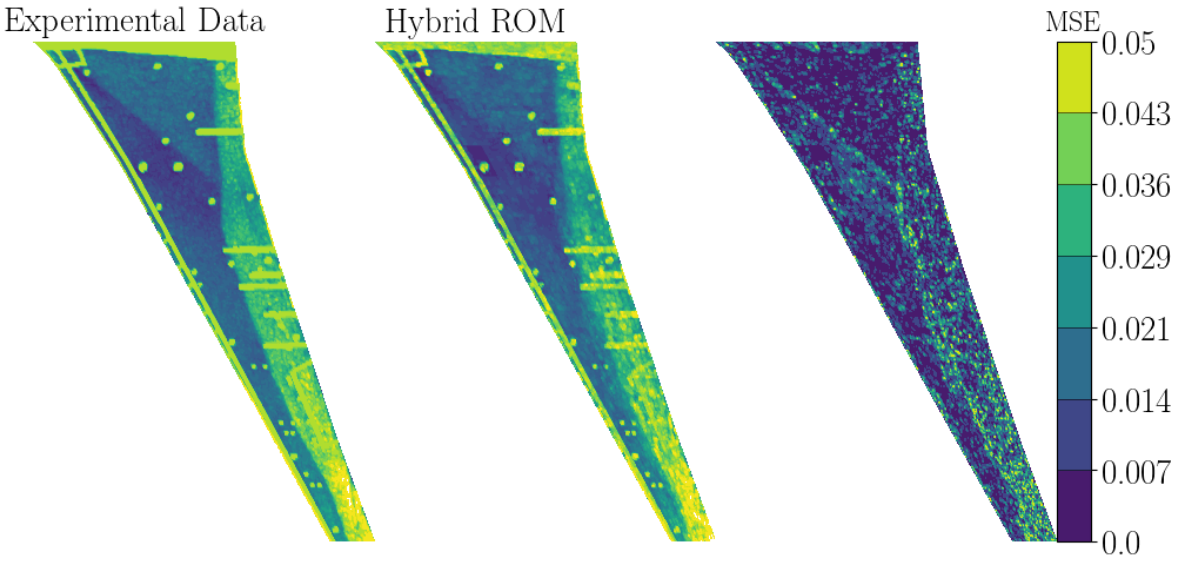


Figure 5.42: Comparison of an original c_p-snapshot and a c_p-snapshot predicted by the hybrid ROM at timestep $t = 150$ (clean wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., $\alpha = 5°$).

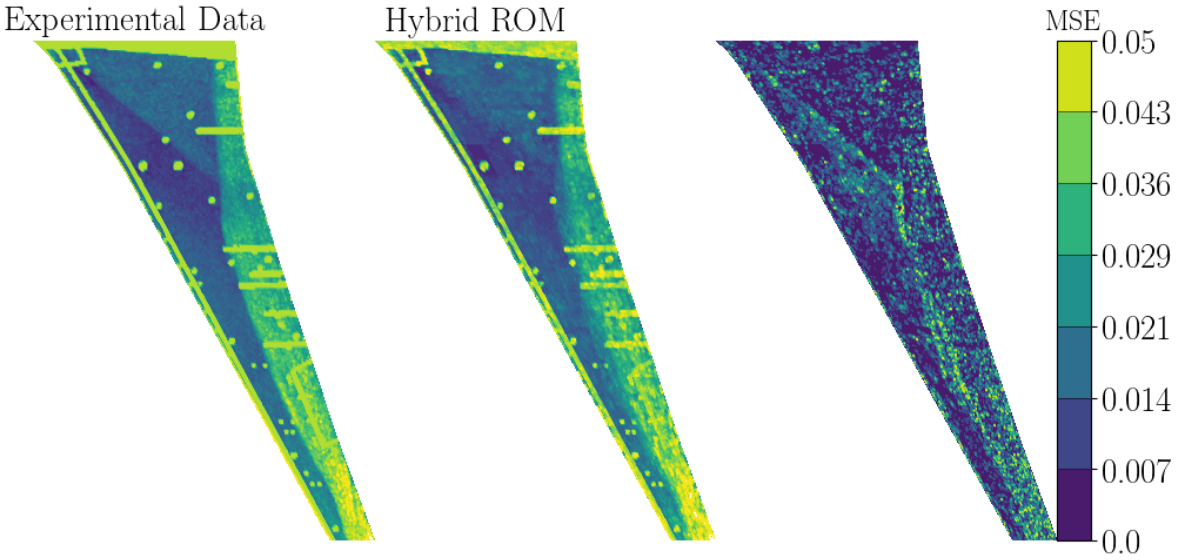


Figure 5.43: Comparison of an original c_p-snapshot and a c_p-snapshot predicted by the hybrid ROM at timestep $t = 200$ (clean wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., $\alpha = 5°$).

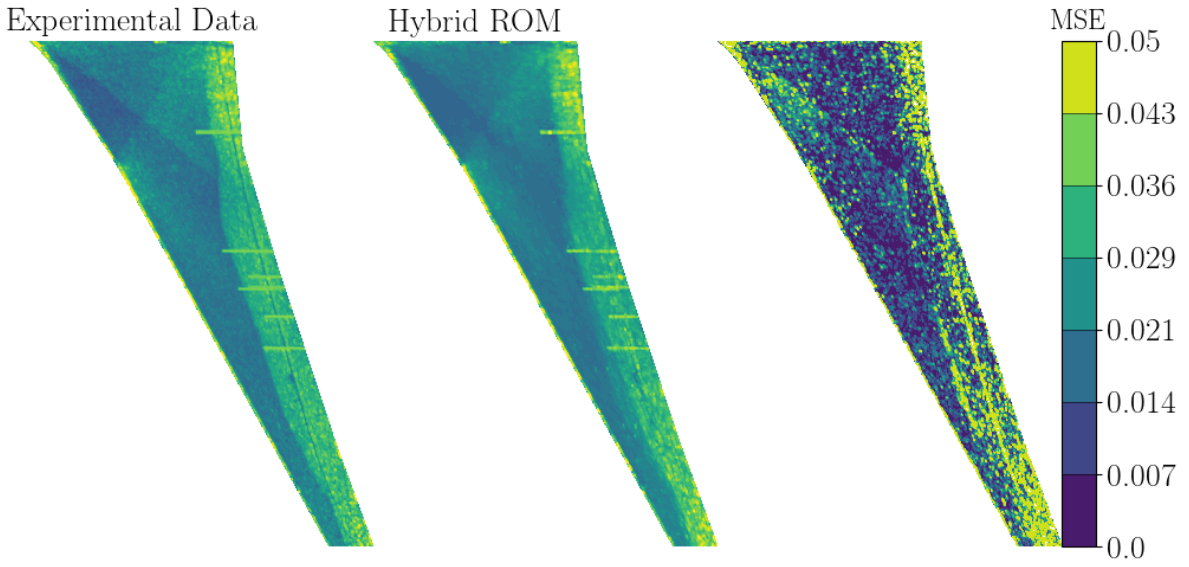


Figure 5.44: Comparison of an original c_p-snapshot and a c_p-snapshot predicted by the hybrid ROM at timestep $t = 150$ (UHBR wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., $\alpha = 5°$).

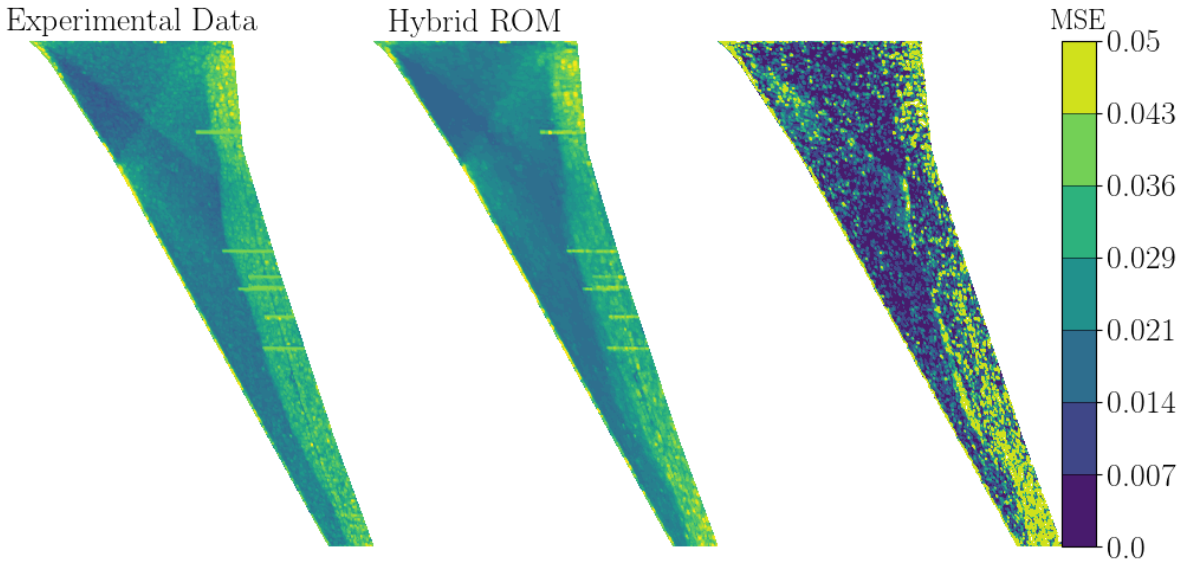


Figure 5.45: Comparison of an original c_p-snapshot and a c_p-snapshot predicted by the hybrid ROM at timestep $t = 200$ (UHBR wing, $Ma_\infty = 0.9$, $Re = 25$ Mio., $\alpha = 5°$).

5.5 Summary

In the first part of this chapter, an overview and analysis of the buffet instability on the Airbus XRF-1 configuration was given. For the analysis, experimental pressure sensor and iPSP data obtained during two measurement campaigns in the ETW were applied. Based on the results, it was shown that the buffet phenomenon can be captured by both the unsteady pressure sensors and the iPSP measurements at varying freestream conditions. Examining the spectra of the sensor signals, frequency peaks related to structural excitations were found and separated from peaks due to aerodynamic effects, like buffet pressure oscillations. In order to determine the propagation speed of the buffet instability in spanwise direction, cross-spectra analysis was applied using the signals of the pressure sensors. The computed velocities agree very well with earlier findings of comparable numerical and experimental studies.

Following the general characterization of the buffet instability on the XRF-1 configuration, ROM methods for both the prediction of three-dimensional integral and local buffet characteristics have been proposed. Therefore, in the second part of this chapter, the training and application of a LSTM neural network for the prediction of pressure signals obtained by the pressure sensors was presented. For the training of the LSTM, data obtained at different flow conditions have been applied. Analogous to Chapter 4, an optimized set of hyperparameters has been defined. Due to the large number of samples in the training data set, a three layered LSTM with 200 neurons in each layer has been applied. Further, the input sequence has been defined as 256 in order to fed an appropriate amount of information into the LSTM. An examination of the convergence trends revealed that a learning rate of $\eta = 1 \cdot 10^{-4}$ is the preferred choice. For performance evaluation, the trained LSTM has been used for the prediction of pressure signals at varying freestream and sensor positions. A good agreement with the reference data was indicated, with an overall fit between 75% and 80%. Characteristic peaks in the pressure spectra related to structural vibrations and aerodynamic oscillations have been correctly captured by the proposed ROM method for all applied test cases.

Focusing on the computation of local buffet characteristics, the last section of this chapter described the application of a hybrid ROM for the prediction of buffet pressure distributions on the XRF-1 wing. The hybrid ROM is composed of a series connection of a CNN-VAR-AE and a LSTM. For training and performance evaluation, surface pressure data measured by iPSP was used. Prior to the training of the hybrid ROM, a tailored preprocessing routine has been applied to the experimental data sets. In order to highlight the performance capability of the proposed ROM for modeling high-dimensional flow field data, both the iPSP data measured on the clean wing and the UHBR wing configuration have been examined. Therefore, two different hybrid ROMs were trained and applied.

For training the hybrid ROM, several hyperparameters have been defined. Considering the CNN-VAR-AE, the number of convolutional levels, the kernel size of each convolution, the batch size as well as the initial learning rate have been set. The number of convolutional levels as well as the kernel size are chosen in order to extract the most important features of the input data by reducing the dimensionality of the data. However, if too many convolutions are applied, important features of the data get lost. The batch size must be defined in order to provide a sufficient amount of data to the ROM, while at the same time keeping the training time as low as possible. Further, concerning the initial learning rate, $\eta = 1 \cdot 10^{-4}$ is chosen.

For the evaluation of the performance of the proposed ROM, the results have been compared using a POD. A comparison of the experimental and predicted modes shows that the proposed ROM method is able to reproduce the first six POD modes with a high degree of accuracy, covering the buffet instability to a sufficient extent. Both the low and high frequency content are correctly represented by the hybrid ROM.

6 Wing Buffet Prediction Due to Forced Vibrations

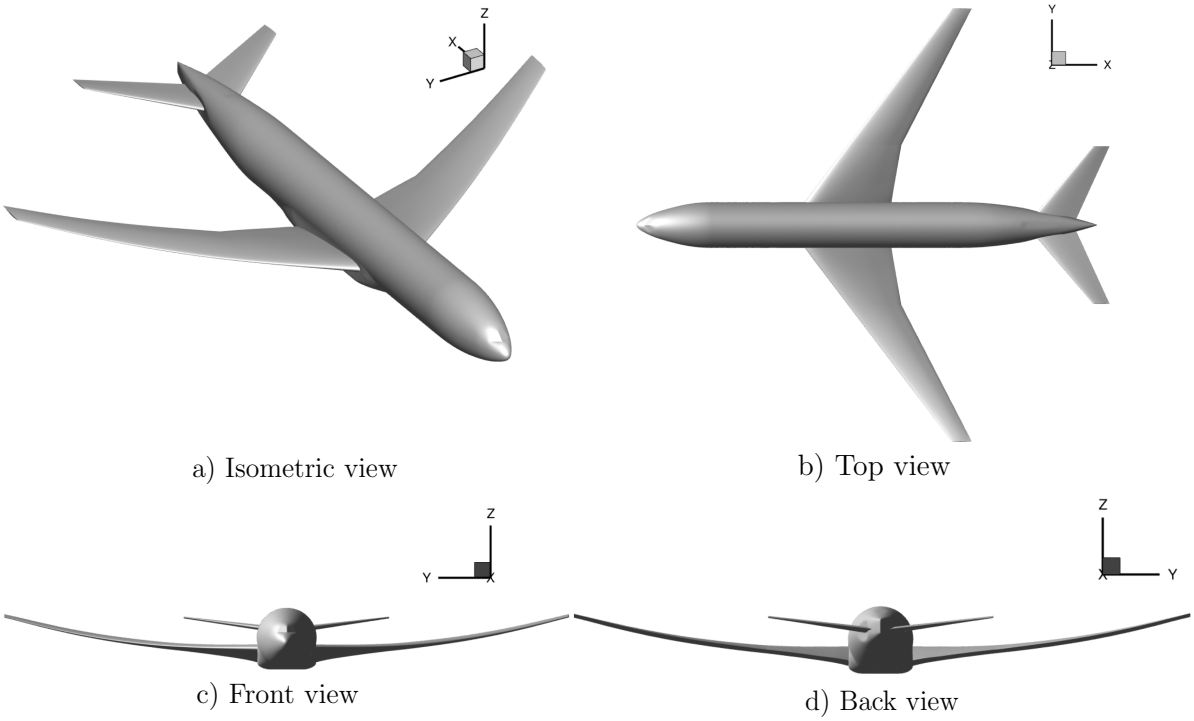
In the preceding chapter, both integral and local buffet characteristics of the XRF-1 configuration due to varying freestream conditions have been computed using two different ROM approaches. In this chapter the focus lies on the prediction of both integral and local wing buffet characteristics due to rigid-body motions and eigenmode-based deformations. The NASA Common Research Model (CRM) is selected as the test case in this chapter, since a structural model is available for this aircraft configuration. Analogous to the previous chapters, a LSTM-ROM is applied for the prediction of integral characteristics, whereas a hybrid ROM is used for the prediction of the buffet pressure distributions. Similar to the previous chapter, the hybrid ROM is defined by a combination of a convolutional autoencoder (CNN-AE) and a LSTM-ROM. For the training and generalization of both ROMs, numerical data computed with the DLR-TAU code, are applied. Some content of this chapter is based on the authors publication, see references [151].

The chapter is structured as follows: In the first section, the selected configuration of the NASA CRM is introduced, followed by an overview of the numerical setup given in Section 6.2. Further, the developed buffet instability on the CRM is briefly analyzed. In Section 6.3, the training and application of a LSTM-based ROM for the computation of integral wing buffet characteristics due to forced rigid-body motions, is presented. In Section 6.4, a hybrid CNN-AE/LSTM-ROM is employed for the computation of unsteady pressure distributions due to eigenmode-based deflections. Therefore, both symmetric and asymmetric mode shape deflections are implemented in the DLR-TAU solver and the aerodynamic responses are simulated. In order to evaluate the performance quality of both ROM methods, a comparison with reference high-fidelity numerical solutions is provided.

6.1 Test Case: NASA Common Research Model

For demonstrating the performance of the selected ROM approaches for capturing wing buffet characteristics due to forced motions, the NASA CRM is chosen. The CRM configuration, which represents a modern commercial transport aircraft configuration, has been developed by NASA's wing aerodynamics technical working group in collaboration with the committee of the drag prediction workshops [130, 131]. The design cruise condition of the model is defined by a freestream Mach number of $Ma_\infty = 0.85$ in combination with a design lift coefficient of $C_L = 0.5$. Considering the investigation of the three-dimensional buffet instability, the CRM configuration has already been applied in several numerical [30, 59, 128] and experimental studies [76].

The CRM configuration investigated in the present thesis represents a wing/body model with a horizontal tail plane (HTP). For the studies in this thesis, both a half- and a full-span model of the CRM configuration are used. In Figure 6.1, the geometry of the full-model is visualized using an isometric, a top as well as a front and a back view.



a) Isometric view b) Top view

c) Front view d) Back view

Figure 6.1: Geometry of the selected NASA CRM configuration.

6.2 Computational Setup

The computational grid employed for all following simulations has already been applied in a numerical buffet study by Ehrle et al. [30]. The purpose of the simulations is the replication of conditions of a corresponding wind tunnel test campaign in the ETW facility [76]. Based on the 2.7% scaled wind tunnel model, the wing is defined by a mean aerodynamic chord of $c_{ref} = 0.189\ m$. The aspect ratio AR is defined as 9, whereas the sweep angle of the wing is given as $\phi = 35°$. Further, the reference area of the wing and the wing span are defined by 0.280 m^2 and 1.586 m, respectively. The geometrical parameters of the CRM configuration, which are based on the wind tunnel model, are summarized in Table 6.1.

Quantity	**Symbol**	**Value**
Aspect Ratio	AR	9
Mean aerodynamic chord	c_{ref}	0.189 m
Sweep angle	ϕ	35°
Wing reference area	A_{ref}	0.280 m^2
Wing span	b	1.586 m

Table 6.1: Geometric properties of the NASA CRM configuration.

The grid has been generated using the commercial meshing software Pointwise. The half-span model grid is composed of approximately $36 \cdot 10^6$ elements, whereas the full-span model grid includes around $70 \cdot 10^6$ elements. The cells around the wing, fuselage and HTP have a structured topology, whereas the farfield is composed of unstructured cells. The farfield is defined by a hemisphere with a radius equal to approximately 50 times of the model length. In addition, a proper boundary layer resolution with a minimum wall distance of the first cell of $y^+ < 1$ is accomplished on the fuselage and the surface of the wing. In addition, a refinement block containing uniform hexahedral elements is set above the boundary layer elements on the wing suction side (see Figure 6.2). This refinement ensures a sufficient spatial resolution of the area of the estimated shock motion and flow separation. The size of the block elements was defined according to a local convective Courant-Friedrichs-Levy (CFL) number of 1.

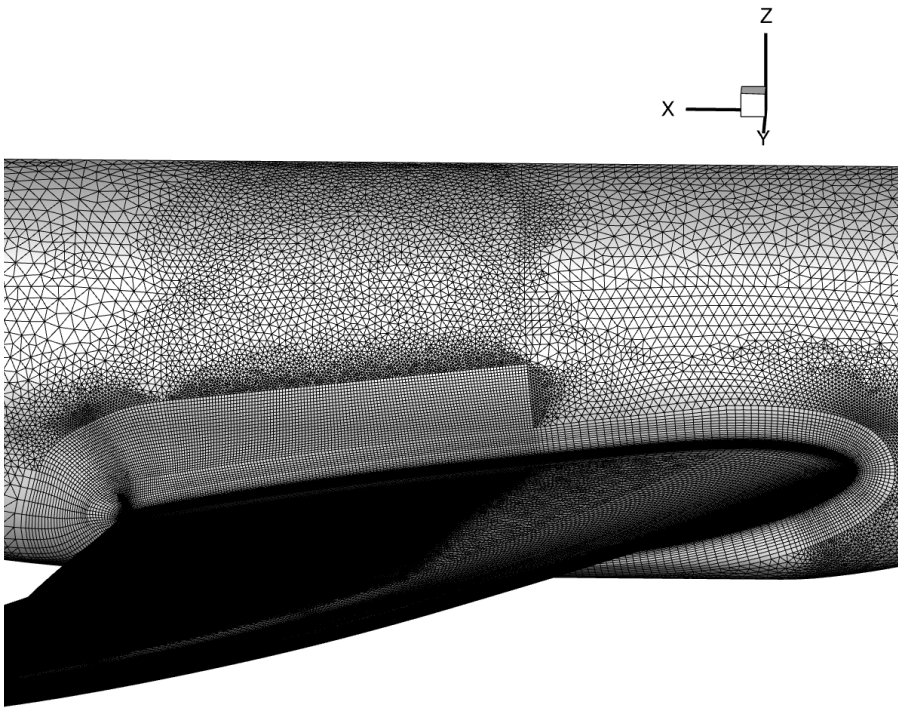


Figure 6.2: Hybrid numerical grid of the CRM configuration with a refinement block containing hexahedral elements on the upper wing surface.

According to the experimental investigations in the ETW facility [76], the buffet condition of the applied CRM configuration is defined by a freestream Mach number of $Ma_\infty = 0.85$, a Reynolds number of $Re = 30 \cdot 10^6$ and an angle of attack of $\alpha = 5°$. In order to approximate the conditions as experienced in the corresponding wind tunnel test campaign, nitrogen including a viscosity correction based on Sutherland's law is chosen as the working fluid. The corresponding parameters for the application of Sutherland's law are summarized in Table 6.2.

Quantity	Symbol	Value
Gas constant	R	$296.8\ \frac{J}{kgK}$
Sutherland's constant	C	$111\ K$
Reference temperature	T_{ref}	$300.55\ K$
Reference viscosity	μ_{ref}	$17.81 \cdot 10^{-6}\ Pa \cdot s$

Table 6.2: Sutherlands parameters for nitrogen.

During the wind tunnel tests, aeroelastic deformations were measured on the model due to aerodynamic forces acting on the model. In order to match the deformations, a twist of 2° and an upward bending of 30 mm near the wingtip are incorporated in the numerical grid by means of a static mesh deformation prior to the simulations.

All simulations of the CRM configuration in the present study are performed using the DLR-TAU code and an URANS approach. Following the study of Ehrle et al. [30], the following settings are applied: The spatial discretization is accomplished by a central differencing scheme. In order to reduce numerical dissipation of small scale structures, a ratio of 0.4 between matrix and scalar dissipation in combination with a fourth order dissipation coefficient of 1/256 are chosen [30]. A Kok skew symmetric scheme is applied for the discretization of the mean flow fluxes, whereas the discretization of the turbulence fluxes is accomplished by a first order Roe scheme. A Green Gauss scheme is applied for the reconstruction of the gradients. Time integration is achieved using an implicit backward Euler scheme, whereas a lower-upper symmetric Gauss-Seidel (LU-SGS) scheme is chosen as a linear solver. Further, a $3v$ multigrid cycle is applied to the grid in order to accelerate convergence [30].

A dual time stepping approach is applied for all simulations with a physical time step of $1 \cdot 10^{-5}$ s and 100 inner iterations per time step. Further, a CFL number of the implicit pseudo time steps of two is chosen. For turbulence modeling, the SA model without trip-term f_t and turbulence suppression term f_{t2} is employed. In addition, a quadratic constitutive relation (QCR) [121] extension is activated for the SA model. As shown by Togiti et al. [129], the combination of an eddy-viscosity turbulence model and the QCR leads to a reduced corner flow separation, which was observed by a numerical study at flight Reynolds number of the CRM by Illi et al. [51].

The applied computational setup indicates a clear buffet instability on the wing suction side, as represented by the series of c_p-snapshots shown in Figure 6.3. Therefore, a buffet cycle (T_{Buffet}), which is defined by a time period of 0.00665s, is divided by six timesteps. Consistent with findings of previous numerical studies [54, 127], a variation in surface pressure in spanwise direction, as defined by the black dotted lines (see Figure 6.3) is clearly visible, which indicates the convection of buffet cells. Further, a characteristic λ-shaped two shock pattern develops on the wing surface. The URANS simulation reveal a characteristic buffet Strouhal number of $Sr = 0.25$ - 0.65 [30], which is also consistent with findings of numerical and experimental studies summarized in Section 2.3.

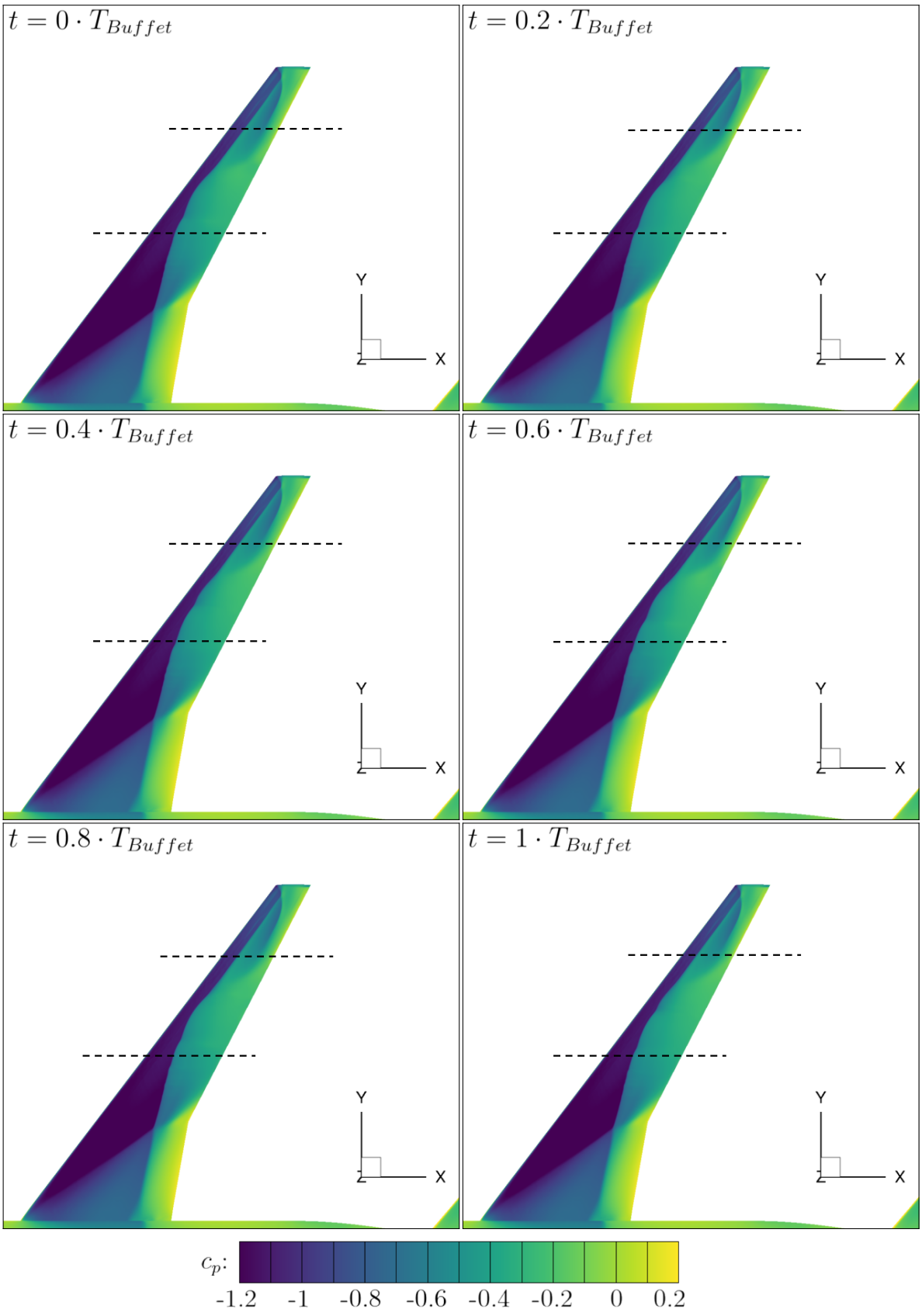


Figure 6.3: Pressure coefficient (c_p) contour plots showing the buffet cycle of the NASA CRM ($Ma_\infty = 0.85$, $Re = 30 \cdot 10^6$, $\alpha = 5°$). T_{Buffet} refers to the buffet period.

Further, a comparison of the mean pressure coefficient $\bar{c}_p$ obtained by the URANS simulation with experimental data [76] showed a good agreement in the region of the shock. In Figure 6.4, a comparison between numerical and experimental $\bar{c}_p$ at a spanwise position of $\eta = 60\%$ is visualized. As shown, the URANS result represent the experimental data with sufficient accuracy.

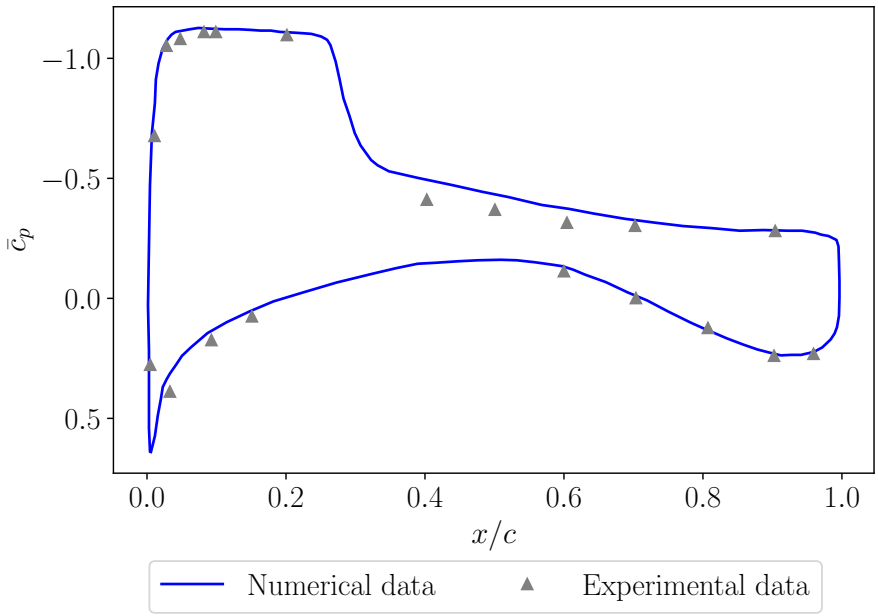


Figure 6.4: Comparison of numerical and experimental $\bar{c}_p$ at a spanwise position of $\eta = 60\%$ ($Ma_\infty = 0.85$, $Re = 30 \cdot 10^6$, $\alpha = 5°$).

Further, an examination of the resulting time-series of the lift coefficient, as shown in Figure 6.5, reveals an aperiodic behavior. Compared to the lift coefficient time-series of an airfoil at buffet condition (see Chapter 4, Figure 4.3), the change in C_L is less pronounced for the three-dimensional buffet instability.

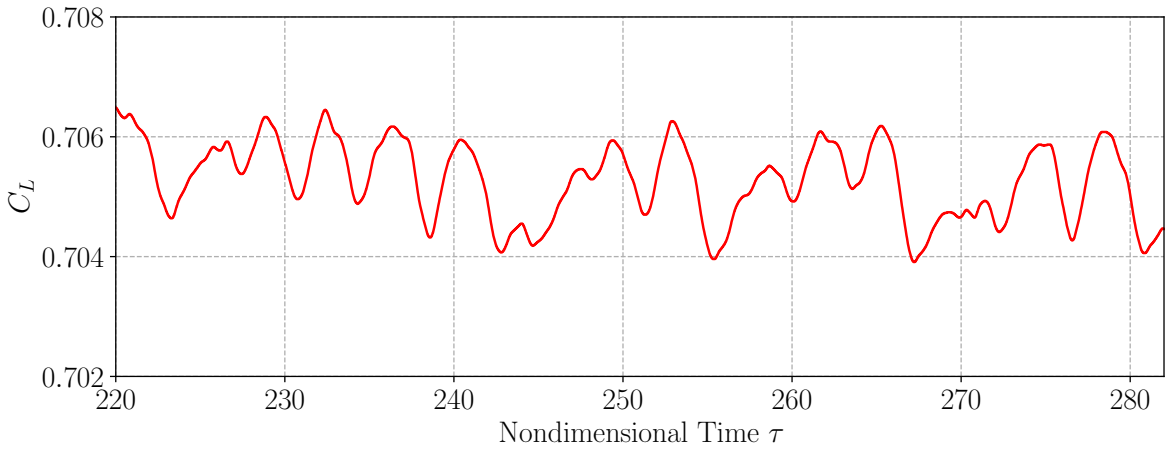


Figure 6.5: Time-series of the lift coefficient C_L of the NASA CRM at a developed buffet condition ($Ma_\infty = 0.85$, $Re = 30 \cdot 10^6$, $\alpha = 5°$).

6.3 Prediction of Motion-Induced Integral Buffet Characteristics

In the following section, a LSTM-based ROM is trained and applied for the prediction of motion-induced integral buffet characteristics of the CRM configuration. In particular, the aerodynamic characteristics of the wing are considered. In the first part of this section, the generation of the training and validation data set as well as the selection of the LSTM hyperparameters are outlined. The second part covers the performance evaluation of the selected ROM. Therefore, the trained LSTM-ROM is applied for the prediction of time-series coefficient trends due to harmonic motions with varying frequencies and amplitudes. The section is concluded with an examination of the reduction in computational time compared to the simulation results obtained by CFD.

6.3.1 CFD-Based Data Set Generation

For the generation of the training and validation data set, forced-motion CFD simulations are performed. Analogous to the deformations of the NACA0012 airfoil grid (see Chapter 4), a RBF-based grid deformation using Python is applied. Further, similar to Section 4.3, a smoothed APRBS (SAPRBS) is used for the excitation of the pitch degree of freedom (DoF) of the wing structure. However, due to limitations of computational time, for the following ROM training only a single SAPRBS covering 10000 timesteps is used. The applied excitation signal is visualized in Figure 6.6.

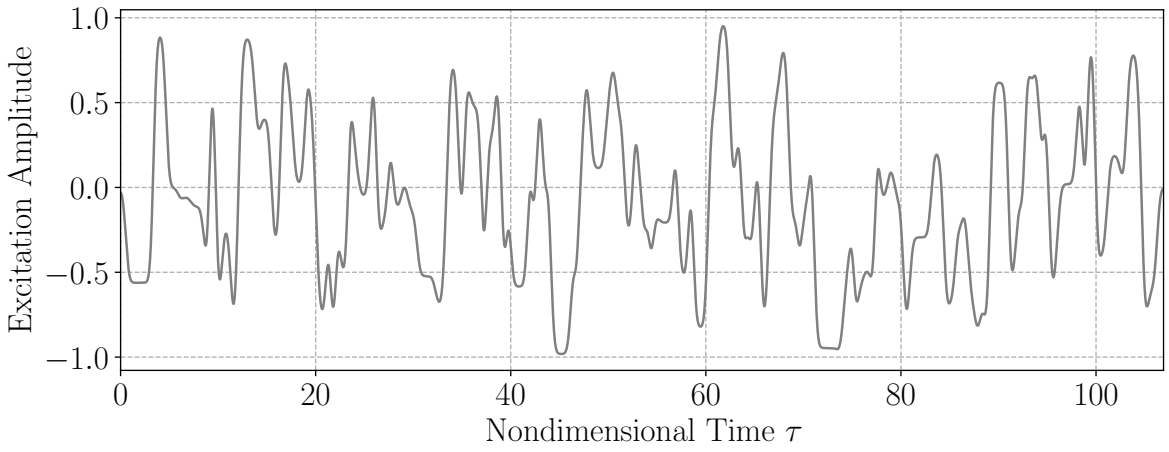


Figure 6.6: SAPRBS for the prescribed excitation of the pitch degree of freedom of the wing structure.

In addition to the time-domain representation, the spectrum of the SAPRBS visualized in Figure 6.6 is presented in Figure 6.7. Based on the frequency content it is shown that the signal covers reduced frequencies in the range of $0.1 < k_{red} < 10$.

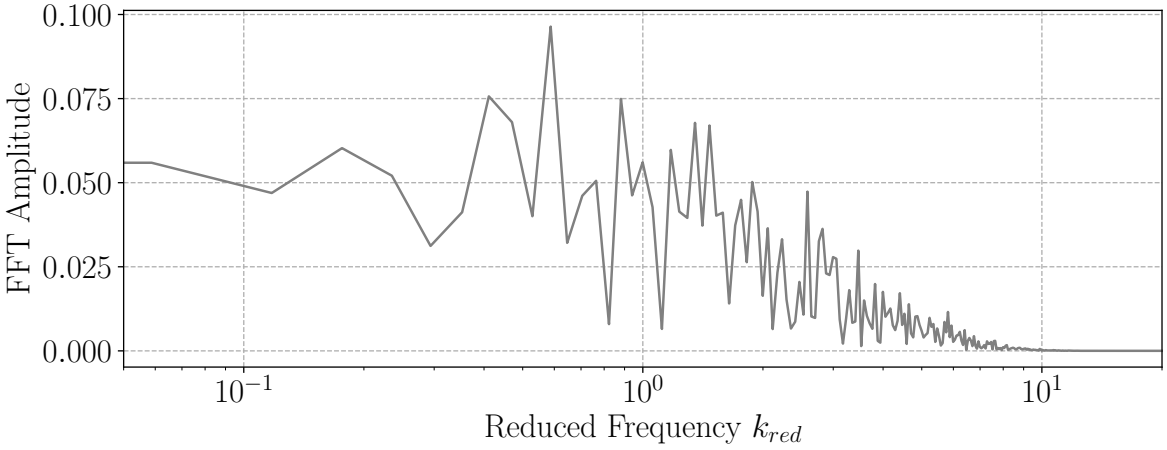


Figure 6.7: Amplitude spectrum of the SAPRBS for the prescribed excitation of the pitch degree of freedom of the wing structure.

Based on the information of the system excitation and the corresponding coefficient time-series trends, the training of the LSTM-ROM is performed. Therefore, the system input is represented by the pitch excitation of the wing structure, whereas the system outputs are defined by the corresponding lift and pitching moment coefficient responses:

$$\begin{aligned} \boldsymbol{x}_t &= [\theta(t)] \\ \boldsymbol{y}_t &= [C_L(t), C_{M_y}(t)]^T \end{aligned} \tag{6.1}$$

6.3.2 Training of the LSTM-ROM

The computed CFD data set is divided into two data sets prior to the training of the LSTM-ROM. The first data set, including 80% of the data points, is applied for training, whereas the remaining data points are used for validation and hyperparameter tuning.

Based on a detailed parameter study, the following hyperparameters are selected: The number of hidden layers is defined as three, with 128 neurons in each layer. The data is fed into the network in sequences, including 64 samples each. Analogous to the previous applications of the LSTM, the batch size is defined as one. The initial learning rate is defined as $\eta = 1 \cdot 10^{-4}$. Analogous to Chapter 4, the state activation is chosen as $tanh$, while the gate activation is defined as sigmoid. In Table 6.3, the final hyperparameters applied for the training of the LSTM are summarized.

Hidden layers	3
Neurons per layer	128
Sequence length	64
Batch size	1
Initial learning rate	$1 \cdot 10^{-4}$
State activation	$tanh$
Gate activation	σ

Table 6.3: Hyperparameters for the training of the LSTM.

The training is terminated after 750 epochs, reaching a sufficient convergence of the training and validation loss. In Figure 6.8, the convergence trends of the training and validation loss are visualized.

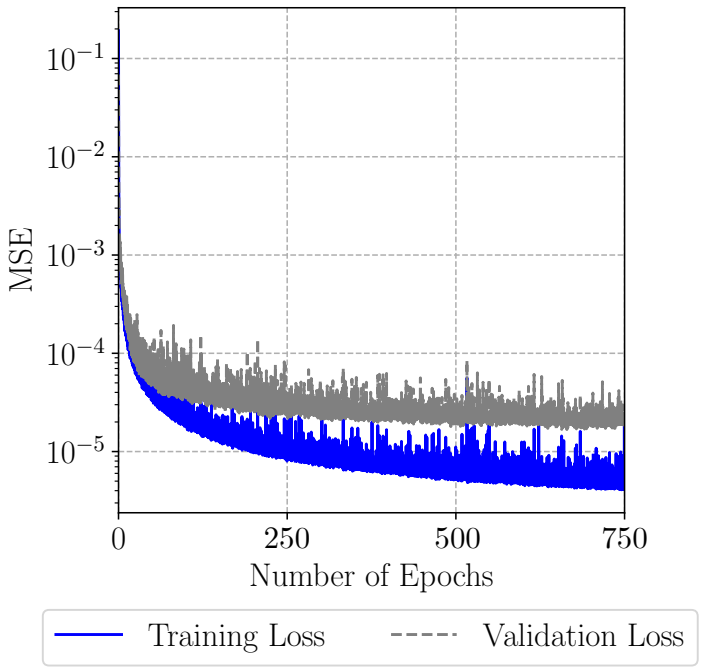


Figure 6.8: Convergence trends of training and validation loss of the LSTM-ROM.

Prior to the performance evaluation of the trained LSTM-ROM using an unknown test data set, the LSTM is applied for the prediction of the lift coefficient data points included in the validation data set in a recurrent multi-step prediction mode. In Figure 6.9, a comparison between the reference CFD solution and the data predicted by the LSTM, is visualized. Applying the error quantification metrics as introduced in Section 3.7, a good agreement between the CFD solution and the prediction of the LSTM is indicated. The corresponding error values and the fit factor for the multi-step prediction mode on the validation data set are summarized in Table 6.4. As shown, an overall fit of almost 78% is accomplished for both the lift and pitching moment coefficient trends.

	A_{err}	P_{err}	C_{err}	Q
C_L	-0.109	0.089	0.124	77.8%
C_{My}	-0.111	0.093	0.128	77.2%

Table 6.4: Amplitude, phase, comprehensive error and fit factor of the multi-step prediction mode on the validation data set.

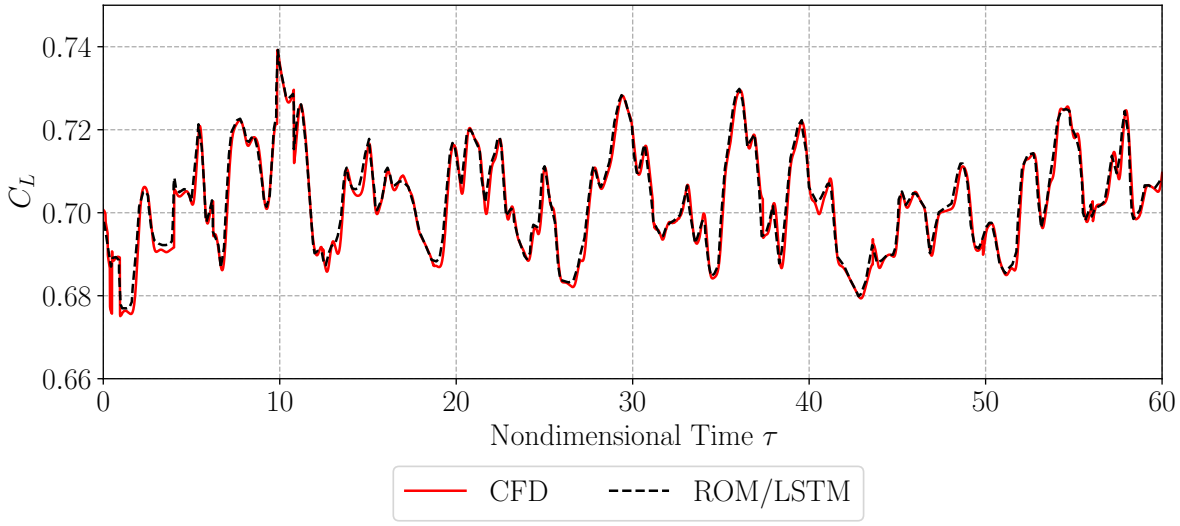


Figure 6.9: Lift coefficient response due to the SAPRBS excitation (CRM, $Ma_\infty = 0.85$, $Re = 30 \cdot 10^6$, $\alpha = 5°$, $\theta = \pm 1.0°$). Besides the CFD reference solution, the simulation results of the LSTM-ROM are shown.

6.3.3 Performance Evaluation

For performance evaluation, the trained LSTM-ROM is applied for the prediction of lift and pitching moment coefficient responses due to harmonic wing pitching motions, covering different frequencies and amplitudes. The reduced excitation frequencies are chosen as $k_{red,Ex} = [0.25, 0.6]$, which corresponds to $Sr_{Ex} = [0.25, 0.6]$. Further, for both selected frequencies, two different excitation amplitudes are considered: $\theta = [\pm 0.5°, \pm 1°]$.

For a comparison of the results, four excitation periods are computed with the TAU solver for the respective harmonic motions. The number of excitation cycles is limited due to the large computational time necessary for the mesh deformation mechanism included in every computed time step. For the initialization of the recurrent multi-step predictions, 128 numerical timesteps are provided as an input to the LSTM.

In Figure 6.10 and Figure 6.11, both time - and frequency-domain responses of the lift coefficient as obtained by the trained LSTM-ROM compared to the reference CFD solution are visualized. In addition, in Figure 6.12 and Figure 6.13, the results of the pitching moment coefficient trends are shown. As it can be seen, the LSTM is able to reproduce the trends of both coefficients with sufficient accuracy. Further, focusing on the frequency spectra, it is shown that the trained LSTM is able to capture most of the frequency content. Based on the corresponding error values summarized in Table 6.5, a good agreement between the CFD and ROM solution is emphasized. An overall fit of about 73% is achieved with the application of the trained LSTM-ROM.

Further, in Figure 6.14 and Figure 6.15 the LSTM-based modeled lift and pitching moment coefficient trends compared to the reference CFD solution with an excitation amplitude of $\theta = \pm 0.5°$ are shown. Similar to the results of the larger amplitude test cases, a good prediction performance of the trained LSTM is indicated.

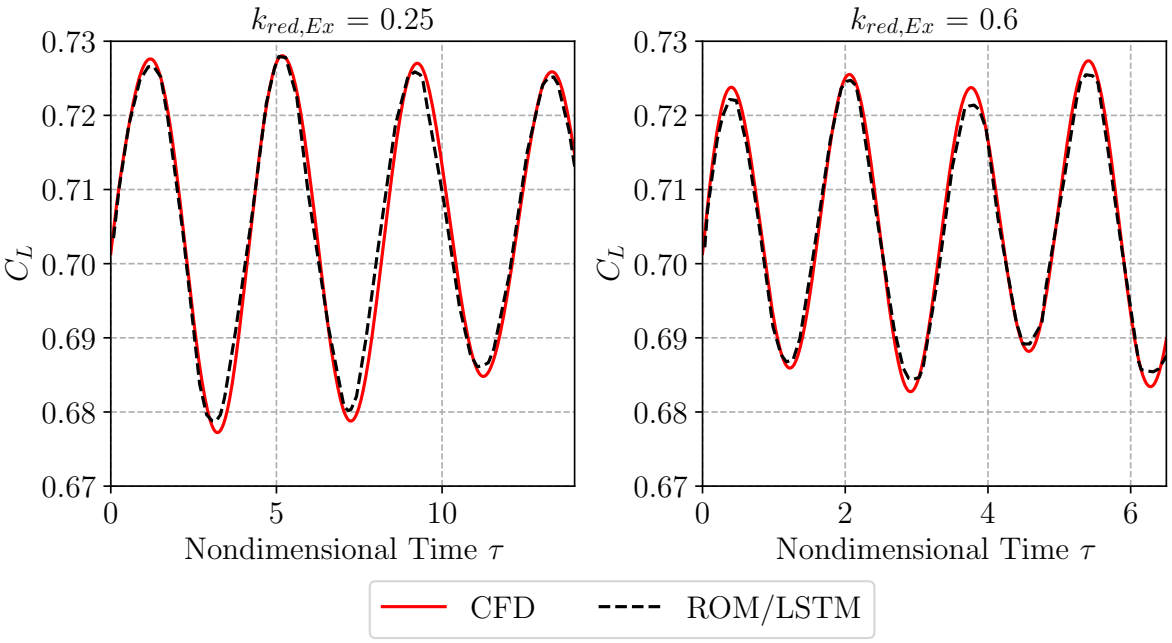


Figure 6.10: Time domain responses of the lift coefficient resulting from harmonic pitching motion with $k_{red,Ex} = [0.25, 0.6]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.85$, $Re = 30{\cdot}10^6$, $\alpha = 5°$, $\theta = \pm 1°$).

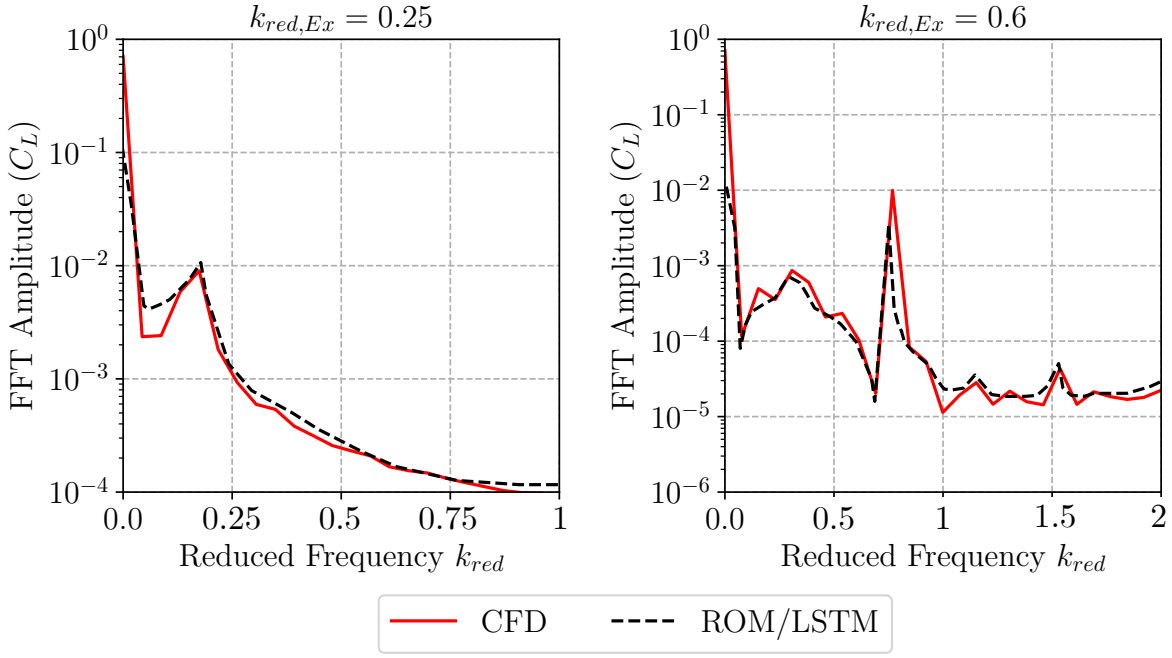


Figure 6.11: Frequency domain responses of the lift coefficient resulting from harmonic pitching motion with $k_{red,Ex} = [0.25, 0.6]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.85$, $Re = 30 \cdot 10^6$, $\alpha = 5°$, $\theta = \pm 1°$).

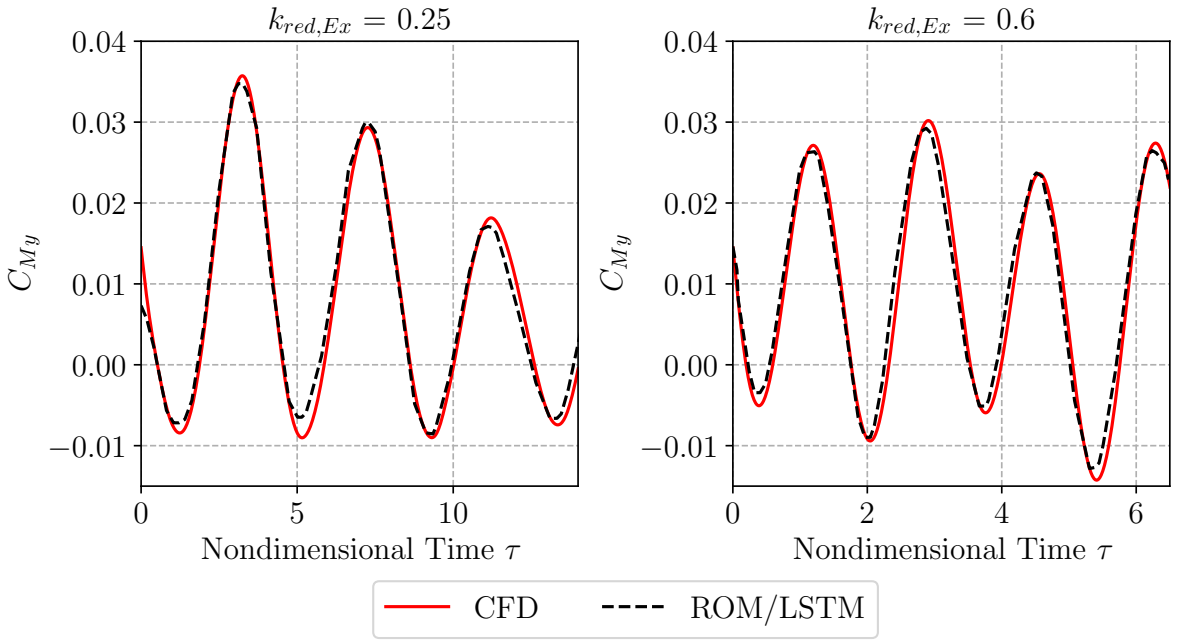


Figure 6.12: Time domain responses of the pitching moment coefficient resulting from harmonic pitching motion with $k_{red,Ex} = [0.25,\ 0.6]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.85$, $Re = 30 \cdot 10^6$, $\alpha = 5°$, $\theta = \pm 1°$).

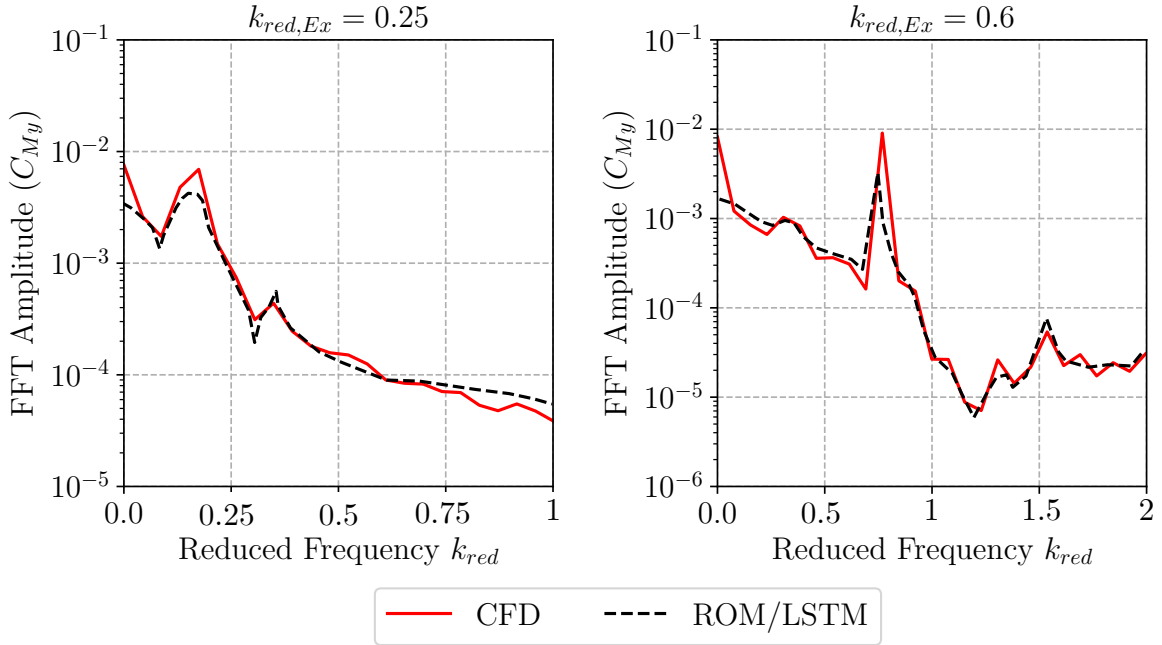


Figure 6.13: Frequency domain responses of the pitching moment coefficient resulting from harmonic pitching motion with $k_{red,Ex} = [0.25,\ 0.6]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.85$, $Re = 30 \cdot 10^6$, $\alpha = 5°$, $\theta = \pm 1°$).

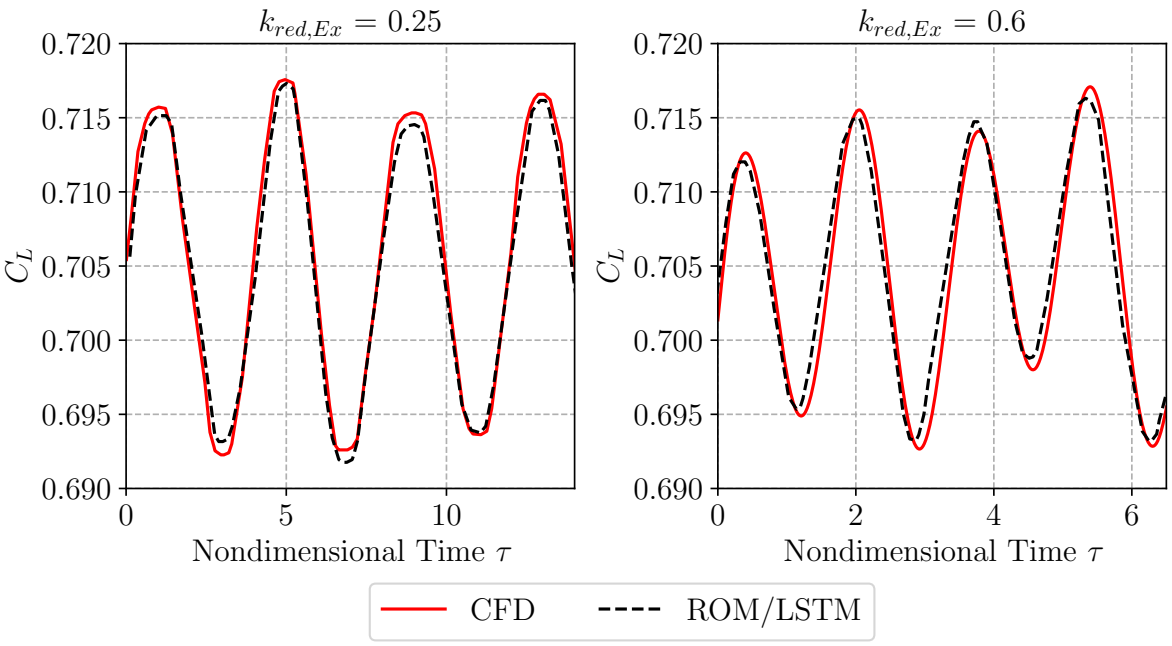


Figure 6.14: Time domain responses of the lift coefficient resulting from harmonic pitching motion with $k_{red,Ex} = [0.25,\ 0.6]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.85$, $Re = 30{\cdot}10^6$, $\alpha = 5°$, $\theta = \pm 0.5°$).

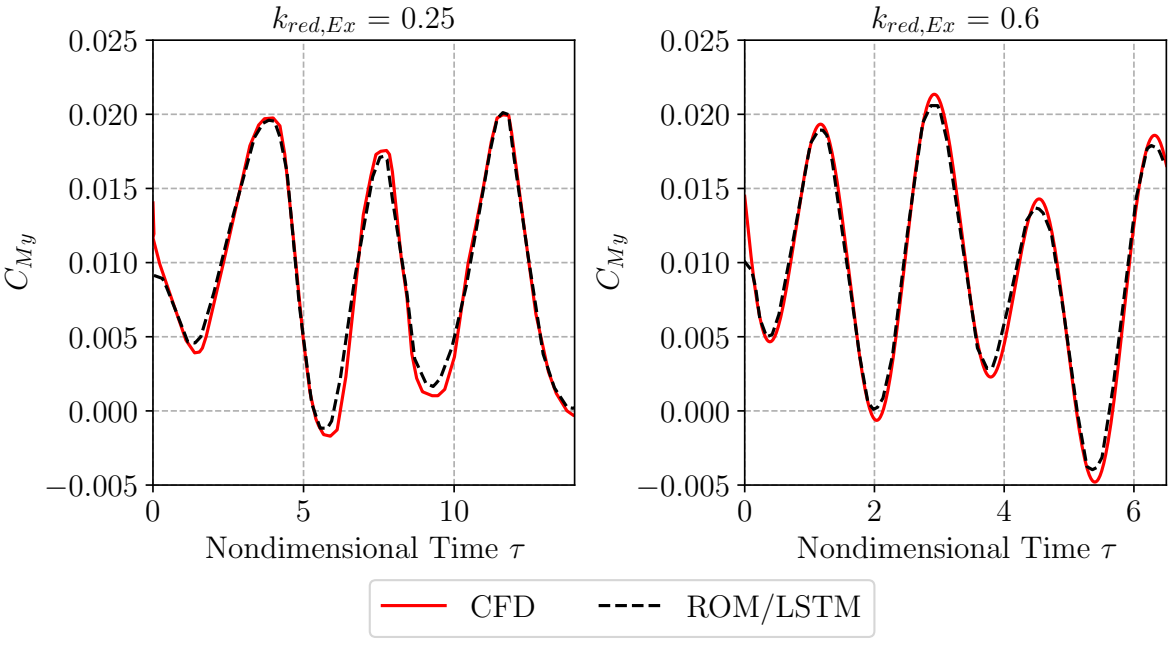


Figure 6.15: Time domain responses of the pitching moment coefficient resulting from harmonic pitching motion with $k_{red,Ex} = [0.25,\ 0.6]$. The results of the LSTM-ROM are compared to the reference CFD solution ($Ma_\infty = 0.85$, $Re = 30 \cdot 10^6$, $\alpha = 5°$, $\theta = \pm 0.5°$).

However, slightly higher deviations around 3-4% are indicated when comparing the corresponding error values with the errors from the previous test cases.

Excitation amplitude	Excitation frequency	Aerodynamic coefficient	A_{err}	P_{err}	C_{err}	Q
$\pm 0.5°$	0.25	C_L	-0.131	0.12	0.157	69.5%
		C_{My}	-0.128	0.119	0.154	70.1%
$\pm 1°$	0.25	C_L	-0.118	0.11	0.143	73.5%
		C_{My}	-0.119	0.117	0.147	72.9%
$\pm 0.5°$	0.6	C_L	-0.125	0.118	0.152	70.3%
		C_{My}	-0.127	0.12	0.154	70.1%
$\pm 1°$	0.6	C_L	-0.123	0.108	0.145	73%
		C_{My}	-0.121	0.109	0.144	73.3%

Table 6.5: Amplitude, phase, comprehensive error and fit factor of the harmonic excitation with varying reduced frequencies and amplitudes.

6.3.4 Efficiency Evaluation

Within the last part of this section, the gain in computational efficiency by applying the LSTM-ROM for motion-induced integral wing buffet characteristic prediction compared to a full-order CFD analysis, is quantified.

Each fully resolved numerical simulation with the DLR-TAU solver has been performed on the SuperMUC-NG of the LRZ. Since the numerical grid of the CRM configuration includes a large number of elements, 10 nodes with 48 CPU cores each are applied, resulting in a total number of 480 CPU cores. Analogous to the training, validation and test computations of the LSTM-ROM performed in Chapter 4, a workstation equipped with an Intel Xeon 2.2 GHz processor has been used. In order to provide the CPU hours as wall clock times, only a single CPU core has been selected on the workstation.

In the first step, the numerical effort due to the training procedure of the LSTM-ROM is analyzed. The computational time required for the SAPRBS simulation is defined by approximately 80 hours on the SuperMUC-NG. Considering the number of applied cores, an overall computational time of 38400 ($= 80 \cdot 480$) CPU hours results. Besides the computational time required for the SAPRBS simulation, the training of the LSTM-ROM on the workstation was accomplished within approximately six CPU hours. Therefore, the total computational effort for the training of the LSTM-ROM sums up to 38406 CPU hours, with the full-order SAPRBS simulation defining the highest share on the overall training costs.

Considering a comprehensive motion-induced wing buffet study, various frequencies and amplitudes for the excitation of the wing structure are of interest. Therefore, the efficiency evaluation is provided in terms of a comparison of the application time of the trained LSTM-ROM and the computational costs of the fully resolved URANS simulations. As already summarized in Section 4.7 of Chapter 4, the CPU time of the

LSTM-ROM is defined by the simulation of the SAPRBS, the training of the LSTM on the workstation, the computations of the initial numerical LSTM input solutions as well as the application of the LSTM on a set of test samples. In contrast, the CPU time of the URANS simulations is defined by the fully resolved CFD computations of all required test cases.

Considering the training and application of the LSTM-ROM as provided in the present chapter, the following CPU hours apply: The application of the LSTM includes the CPU hours for the SAPRBS simulation (= 38400 CPU hours), six CPU hours for training the LSTM, a batch of test samples for the initialization of the recurrent multi-step prediction as well as 0.01 CPU hours for the application of the trained LSTM itself. In contrast, the CFD simulations sum up to approximately 9600 ($= 20 \cdot 480$) CPU hours for each harmonic excitation, assuming an averaged computational time of 20 hours. With an average of 9600 CPU hours, a number of approximately 2000 timesteps is accomplished. Therefore, the computation of the timesteps for the LSTM initialization, which are defined by a sequence length of 128, required approximately 614 CPU hours.

In Table 6.6, the number of CPU hours required for the application of the LSTM in contrast to the CPU time of the URANS computations, are summarized. Here, the CPU hours for the computation of a single harmonic motion test case are considered.

ROM	SAPRBS simulation	38400
	ROM training	6
	ROM initialization	614
	ROM application	0.01
Fully resolved	URANS simulation	9600

Table 6.6: Comparison of CPU hours for the LSTM and the fully resolved URANS simulations considering a single harmonic motion test case.

In the framework of a more detailed buffet analysis, harmonic motions covering different frequencies and amplitudes are of interest. Considering the harmonic motion test cases as defined in Section 4.7 in Chapter 4, a number of 24 simulations is required. Therefore, the number of CPU hours related to the application of the LSTM sum up to 38400 CPU hours for SAPRBS simulation and 14736 ($= 24 \cdot 614$) CPU hours for the computation of the initialization test samples for each harmonic test case. In contrast, $23 \cdot 10^4$ ($= 24 \cdot 9600$) CPU hours are required for the full-order URANS simulations of all considered test cases. In order to emphasize the reduction in computational time by applying the trained LSTM for a detailed wing buffet investigation, a comparison of the CPU hours for the full-order URANS simulation and the application of the LSTM-ROM is visualized in Figure 6.16.

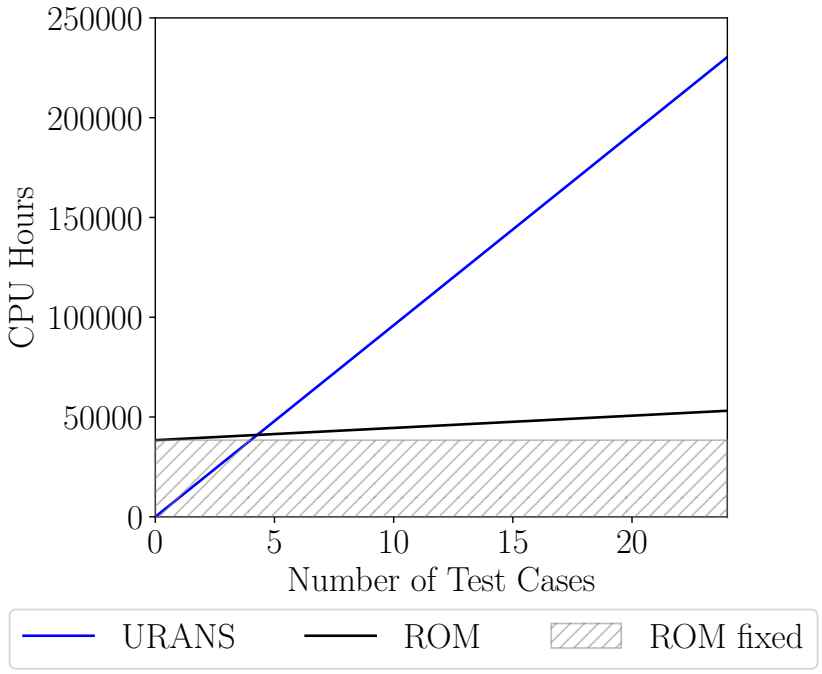


Figure 6.16: Comparison of CPU hours required for the fully resolved URANS simulation and the LSTM-ROM simulations.

Analogous to Figure 4.23 presented in Section 4.7, the grey area defines the computational effort for the training of the LSTM-ROM, whereas the black line represents the additional CPU hours required for the computation of the initial test samples. The blue line indicates the amount of CPU hours required for the full-order CFD simulations of all test cases. As shown in Figure 6.16, with increasing number of test cases the number of CPU hours for the URANS simulation drastically increases compared to the CPU hours required for the LSTM application. By applying the trained LSTM, a reduction in computational time by one order of magnitude is possible.

6.4 Prediction of Surface Pressure Distributions Due to Structural Eigenmode-Based Deformations

Within this section, the training and application process of a hybrid ROM for the prediction of unsteady buffet surface pressure distributions due to structural eigenmode-based deformations is presented. The architecture of the hybrid ROM is similar to the hybrid ROM introduced in Section 5.4.1, however, there are some conceptual changes in the setup. In the first part of this section, a comprehensive overview of the architecture of the hybrid ROM is given. The second part covers a description of the applied structural model and the implementation of symmetric and asymmetric eigenmode-based deformations. In the third and fourth section, the generation of the training and validation data set is outlined and the preprocessing steps applied to the numerical data sets are briefly summarized, respectively. Following the description of the data preprocessing, the training procedure of the hybrid ROM is outlined and the performance of the ROM is evaluated in detail. This section is concluded with an analysis of the computational efficiency of the hybrid ROM compared to the high-fidelity reference CFD simulations.

6.4.1 Hybrid ROM Architecture

The architecture of the hybrid ROM applied in the following is defined by a convolutional autoencoder (CNN-AE) and a LSTM. Analogous to the hybrid CNN-VAR-AE/LSTM introduced in Section 5.4.1, the CNN-AE is used for the reduction of the flow field data, whereas the LSTM is applied in order to predict changes in the surface pressure distribution due to the eigenmoden-based structural deformations. The architecture of the hybrid ROM is visualized in Figure 6.17.

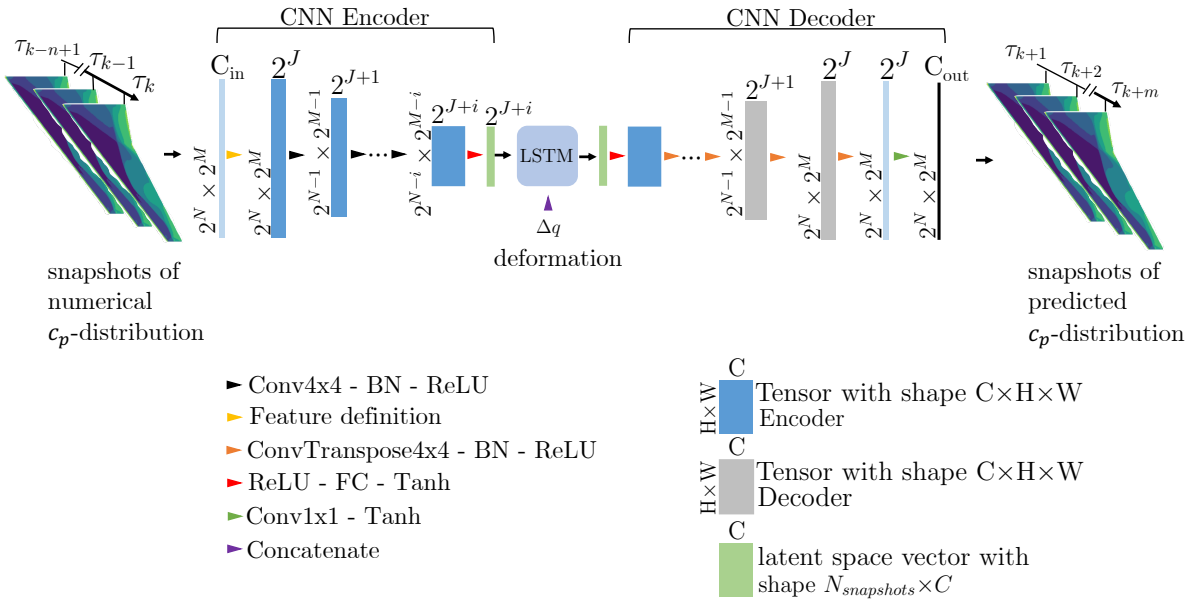


Figure 6.17: Architecture of the hybrid deep learning model.

Similar to the architecture described in Section 5.4.1, the encoder and decoder of the CNN-AE are divided into several convolution and de-convolution levels. The model receives a sequence of numerically computed c_p values of the wing suction side, which is defined by a tensor of $N_{in} \times C_{in} \times H \times W$. Here, N_{in} the defined as the number of input time steps, C_{in} denotes the number of input channels and H and W represent the number of grid points in span - and chordwise direction, respectively. Following common practice, C_{in}, H and W are defined by powers of two.

At each level of the encoder, a multi-channel convolution with a kernel size of 4×4, a stride $s = 2$ and padding of one is applied. Following the convolution operation, the input data is normalized using batch normalization (BN) [52]. Subsequent, a ReLU activation is applied to each element of the input data. Due to this sequence of operations, the output channel size at each level is defined as two times the input channel size of the respective convolution level. In contrast, the size of the spatial dimensions (H,W) is reduced by a factor of two at each level of the encoder. After passing the different encoder levels, a second ReLU activation is applied to the input data. In the last steps, the data is flattened and passed through a FC layer and a *tanh* activation layer.

Before passing the encoded data into the LSTM, the information concerning the deformation q and the c_p data are combined. The aim is the prediction of pressure distributions at timesteps $k+m$ based on several previous c_p-snapshots at timesteps $k-n+1$ to k. Here, n defines the number of timesteps used for the prediction operation, whereas m denotes the number of timesteps to be predicted ahead. Concerning the deformation, incidence amplitudes at timesteps $k-n+2$ to $k+1$ are considered. The scalar deformation values are concatenated to the end of the vector containing the c_p values, as shown in Figure 6.18. The resulting latent vectors including both the c_p-distribution and the deformation information are passed sequentially into the LSTM.

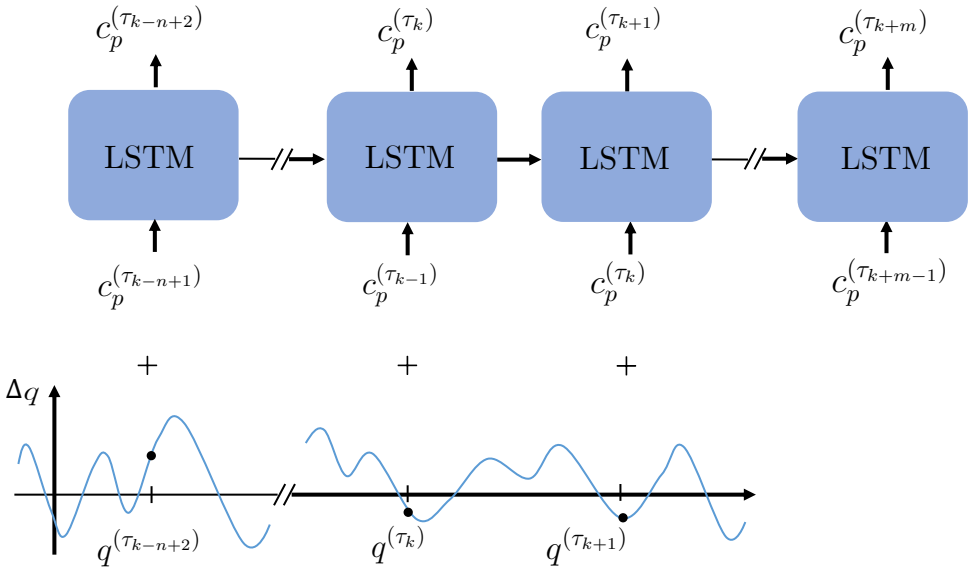


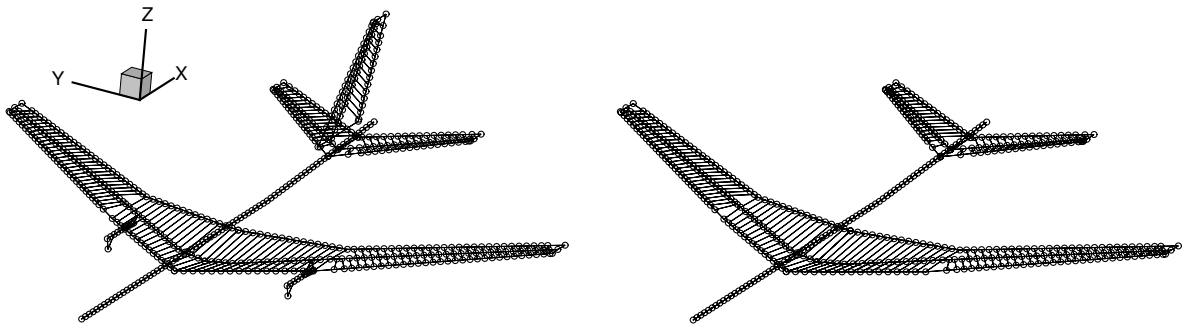
Figure 6.18: Concatenation of deformation q and surface c_p.

After applying the LSTM, the predicted data is fed back into the decoder. At each level of the decoder, a transposed convolution is applied to upscale the spatial dimension. The corresponding kernel, stride and padding are chosen equally to those of the encoder. Further, a BN is applied at each decoder level. At the final level, the decoded data is activated by a *tanh* function.

6.4.2 Implementation of Eigenmode-Based Deformations

For structural modeling, the FERMAT configuration proposed by Klimmek [58] is applied. The finite element model (FEM) was developed in order to allow static and dynamic aeroelastic investigations on the basis of the CRM geometry, referring to a full scale aircraft. The FERMAT model allows the representation of two different set-ups, namely the C1 case representing the maximum zero fuel weight configuration, as well as the C2 case, defining the configuration with maximum take-off weight (100% fuel). For the following investigation, the C2 mass configuration is applied. The C2 configuration includes in total 56 modes, containing six rigid body modes and 50 elastic eigenmodes.

The FEM of the FERMAT configuration includes nodes modeling the fuselage, the wing, the VTP, the HTP as well as the pylons. Since the numerical grid of the CRM does only include the fuselage, wing and HTP, the FEM is reduced for the surface interpolation between the structural and the aerodynamic surface grid. In Figure 6.19, a comparison of the original (left) and reduced (right) structural FEM is depicted.



a) Original FEM b) Reduced FEM

Figure 6.19: Comparison of the original structural FEM (a) and the reduced structural FEM (b) [136].

Prior to the interpolation between the numerical and structural grid, the aerodynamic grid ($grid_a$) needs to be upscaled in order to match the dimensions of the structural grid:

$$grid_{a,scaled} = grid_a \cdot \frac{y_{s,max}}{y_{a,max}} \tag{6.2}$$

In Equation 6.2, $y_{a,max}$ and $y_{s,max}$ define the maximum wing spans of the aerodynamic and structural grid, respectively.

The deformation of the CFD grid is performed by a nearest neighbor search. Each node of the structural grid is linked to a node of the aerodynamic grid located at the minimum euclidean norm between both grid nodes. The interpolation of the model deflections onto the surface grid is achieved using a thin-plate splines algorithm. Since particular focus is on the deformation of the wing structure, the intersection between the wing root and the fuselage needs a special treatment. In order to avoid any deformation of the fuselage and the generation of negative volume cells, interpolated node deflections at spanwise position $y_{node} < y_{limit}$ (with $y_{limit} = 0.085$ m) are defined as zero. Further, the wing area close to the fuselage is divided into five sections in spanwise direction. Within this sections, which are depicted in Figure 6.20, the interpolated deflections δ_{interp} are linearly faded in using a dimensionless fading factor γ_f, resulting in the final deflections δ_{final} of the nodes:

$$\delta_{final} = \delta_{interp} \cdot \gamma_f \tag{6.3}$$

In Figure 6.20, the selected areas close to the wing root are depicted. In addition, the corresponding fading factors γ_f for each section are listed.

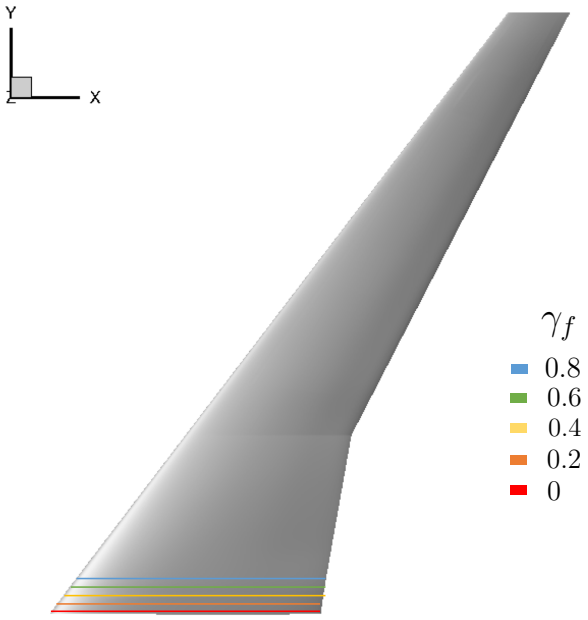


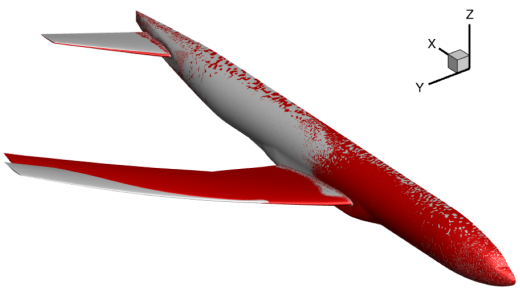
Figure 6.20: View of the upper wing surface with marked fading sections.

Following the work of Winter and Breitsamter [136, 139], the mode deflections are scaled proportional with respect to the reference chord length c_{ref}. The resulting scaling factor SF is defined as follows:

$$SF = f \cdot \frac{c_{ref}}{A_{max}} \tag{6.4}$$

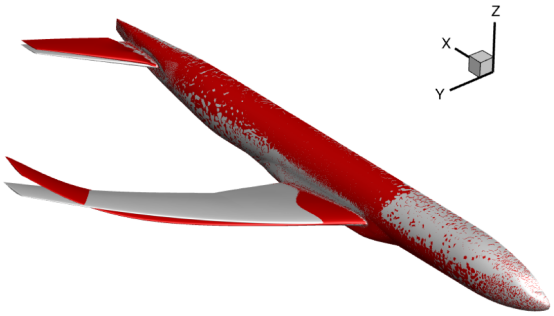
with A_{max} defining the maximum deflection amplitude of the considered mode and f denoting the reference scaling factor, which is chosen as $f = 0.01$ (1%). The factor is defined in order to guarantee a stable numerical simulation.

Based on the developed deformation framework, symmetric as well as asymmetric mode shapes of the wing have been implemented and simulated. In Figure 6.21, an overview of some of the selected, implemented wing modes shapes is given.

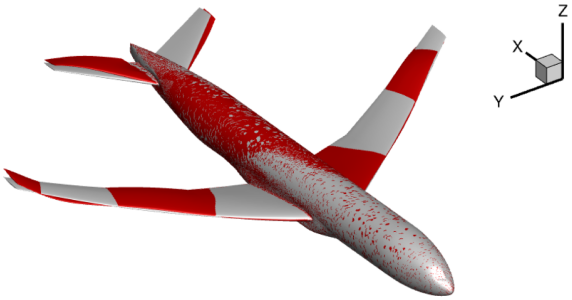


Elastic mode 11
1st symmetric wing bending

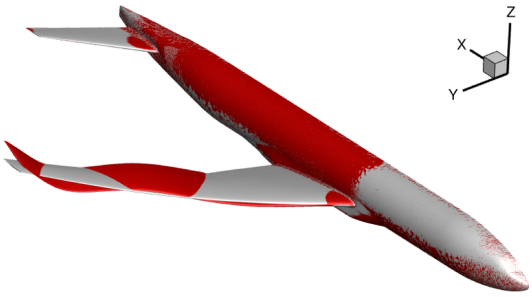
Elastic mode 14
1st asymmetric wing bending



Elastic mode 13
2nd symmetric wing bending



Elastic mode 25
2nd asymmetric wing bending



Elastic mode 27
3rd symmetric wing bending

Elastic mode 28
1st asymmetric wing torsion

Figure 6.21: Overview of selected structural-eigenmode-based surface deformations of the FERMAT-C2 configuration (red) and the non-deformed aircraft geometry (grey). For a clear comparison, increased deflections are visualized.

It has to be mentioned that the implemented simulation framework allows for the simulation of further symmetric and asymmetric wing shapes. However, the maximum deflection amplitude needs to be chosen individually for each mode in order to guarantee a stable numerical simulation.

6.4.3 CFD-Based Data Set Generation

For the training and the performance evaluation of the hybrid ROM, pressure data obtained from simulations due to one symmetric and one asymmetric mode shape deflection are considered. Therefore, mode shape 11 and 14 (see Figure 6.21, top left and right) are selected. For the generation of the training and validation data set, the modes shapes are excited, with the amplitude defined by a SAPRBS as depicted in Figure 6.22. As shown, the minimum and maximum deformation amplitude of the modes are chosen as one. Due to the limitation of computational resources, the SAPRBS sequence is limited to 2000 timesteps, with a timestep equal to the one of the numerical simulation.

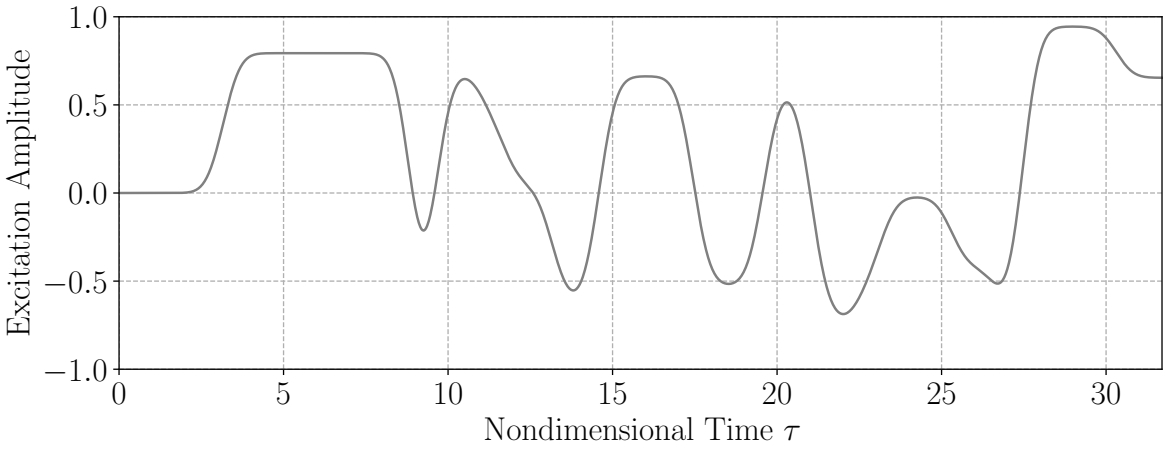


Figure 6.22: SAPRBS for the prescribed deformation of the mode shapes.

In addition to the SAPRBS time-domain signal, the frequency content of the SAPRBS shown in Figure 6.22 is visualized in Figure 6.23.

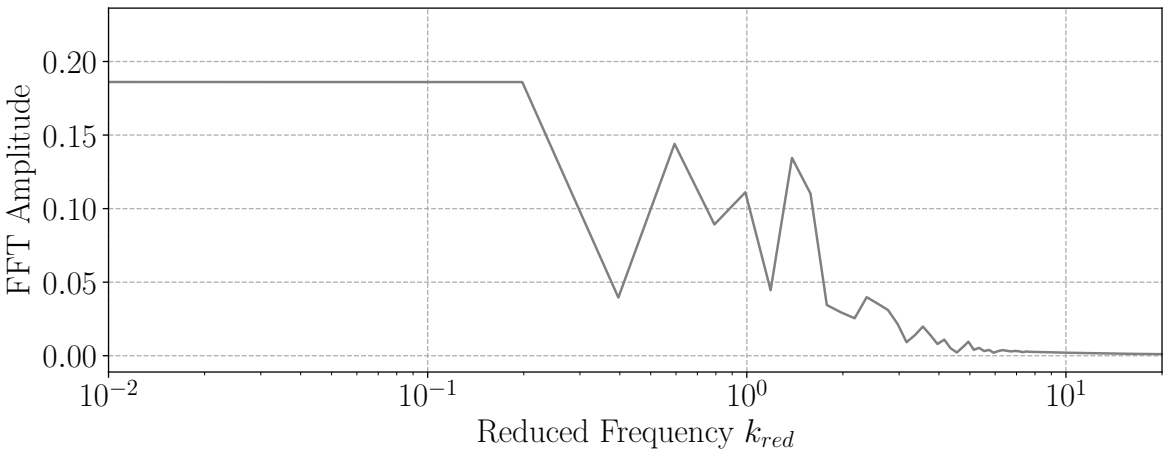


Figure 6.23: Amplitude spectrum of the SAPRBS for the prescribed deformation of the mode shapes.

Due to the structural deformation, distinct changes in the surface c_p on the wing suction side are present. In Figure 6.24 and Figure 6.25, a series of six consecutive

c_p-snapshots of the simulation results of deflections with mode shapes 11 and 14 are depicted, respectively.

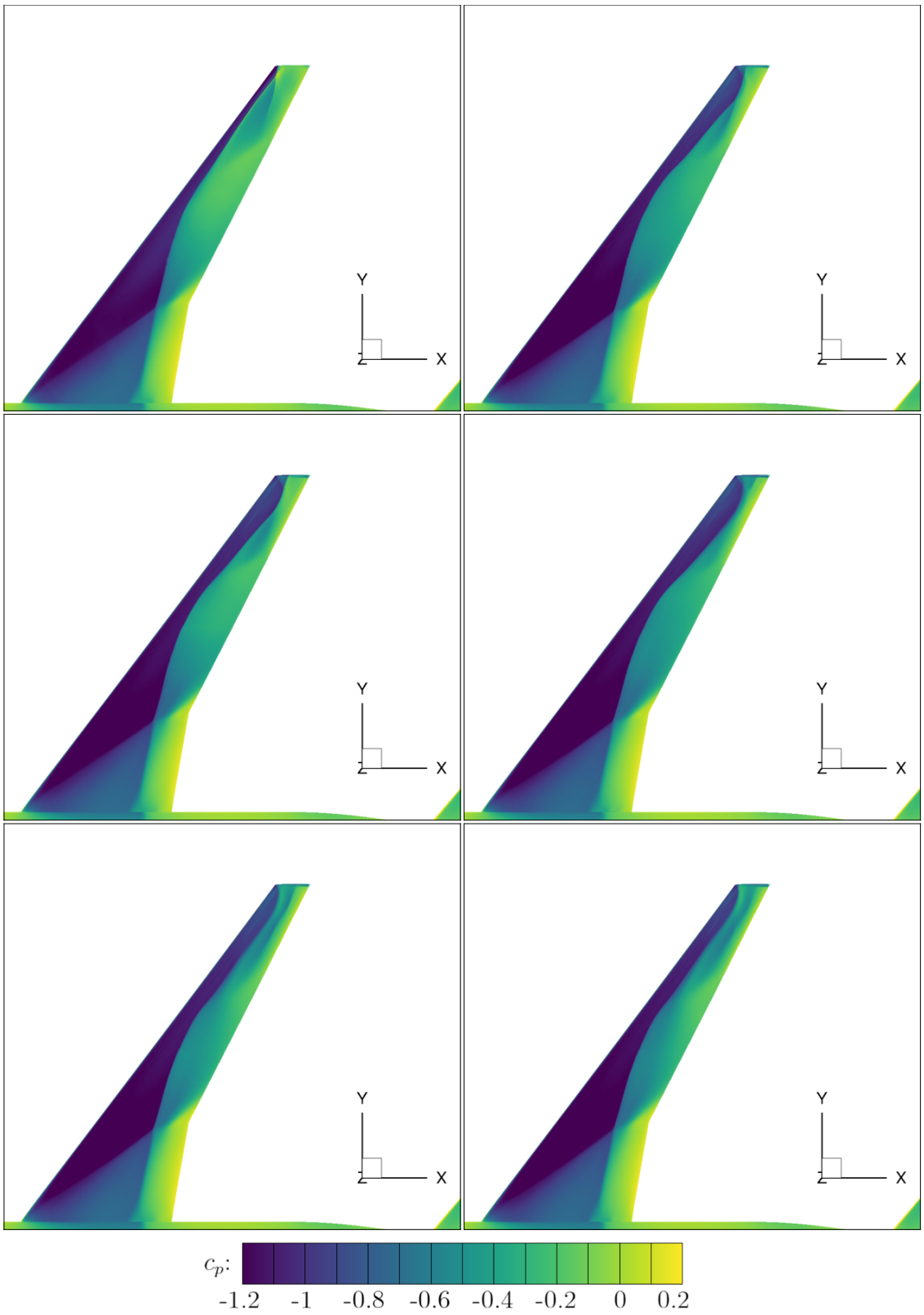


Figure 6.24: Pressure coefficient (c_p) contour plots of mode shape 11 due to the SAPRBS excitation ($Ma_\infty = 0.85$, $Re = 30 \cdot 10^6$, $\alpha = 5°$).

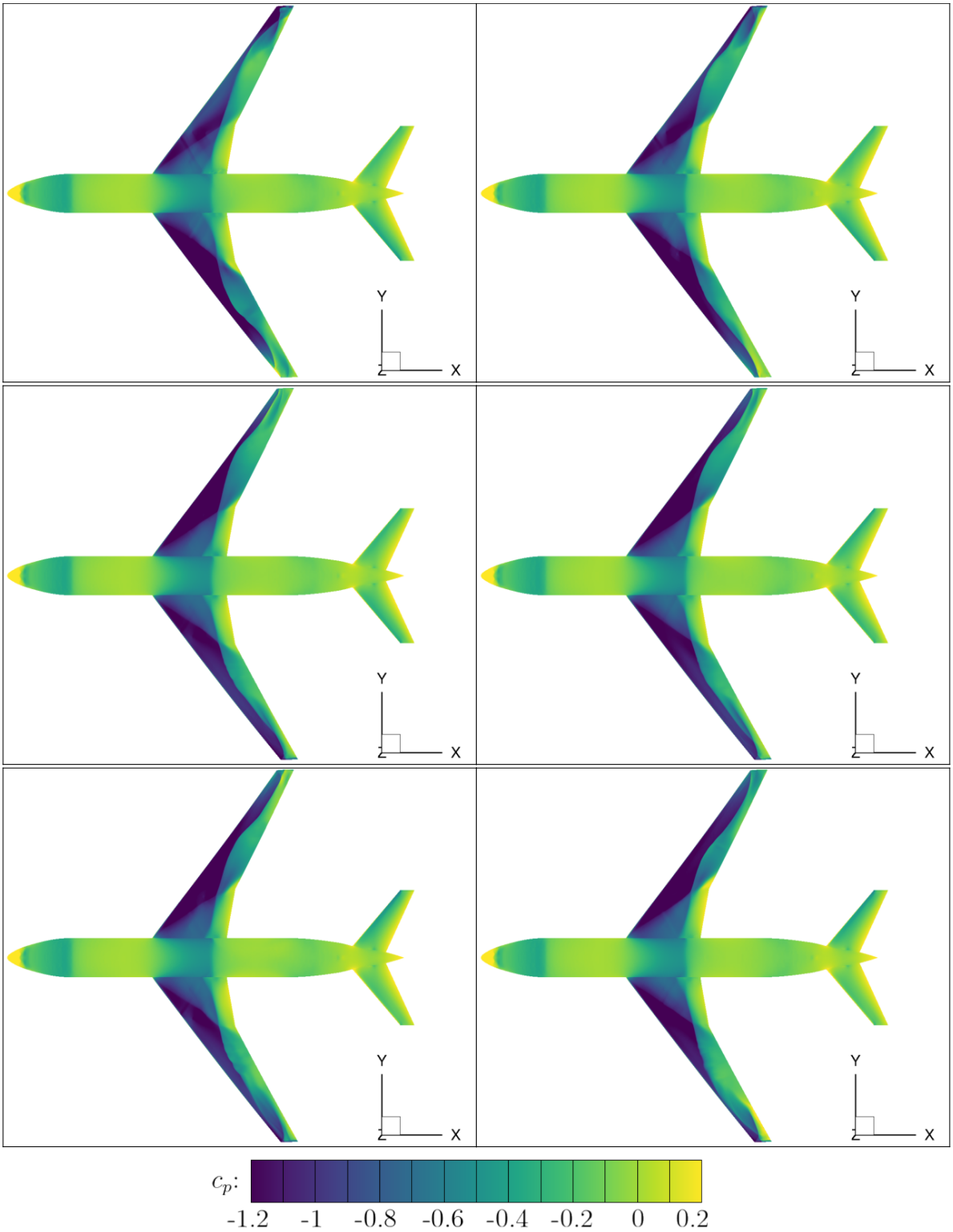


Figure 6.25: Pressure coefficient (c_p) contour plots of mode shape 14 due to the SAPRBS excitation ($Ma_\infty = 0.85$, $Re = 30 \cdot 10^6$, $\alpha = 5°$).

Compared to the pressure distribution at buffet condition (see Figure 6.3), larger fluctuations of the pressure in span - and chordwise direction are clearly indicated. Especially at the wing tip and the mid-span area, the largest fluctuations in the surface c_p are visible.

6.4.4 Data Preprocessing

Prior to the training of the hybrid ROM, two preprocessing steps are applied to the numerical data. Due to the discretization of the wing suction side, the c_p-distribution on the upper wing is represented by 168686 data points. Therefore, in the first step, the size of the data is reduced by linear interpolation. The surface of the wing is divided into two parts, as shown in Figure 6.26.

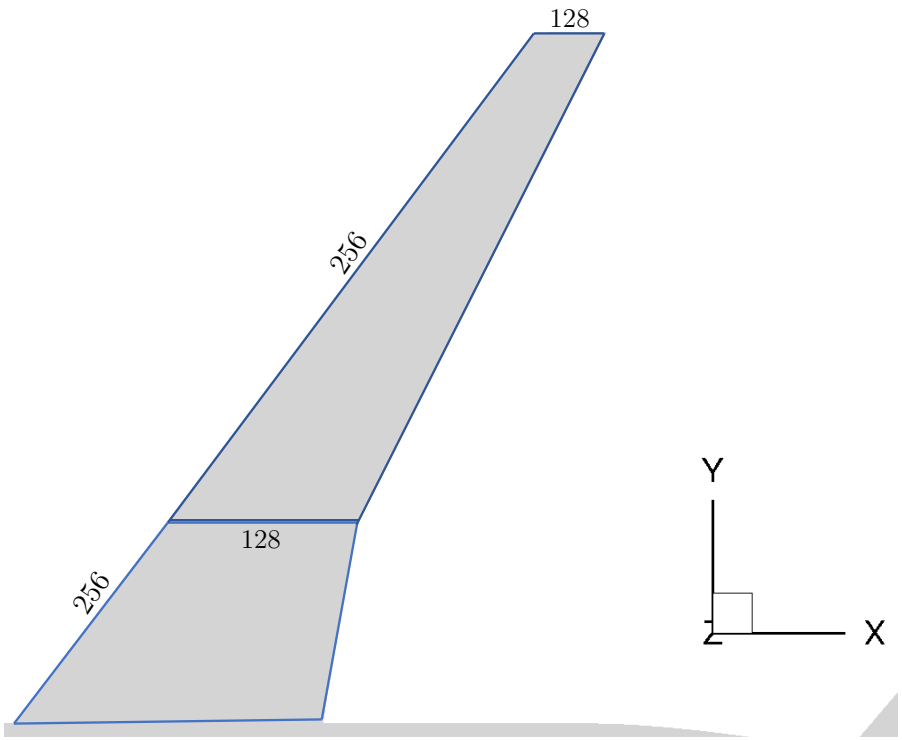


Figure 6.26: Division of the wing suction side for pressure distribution interpolation.

For each part, a grid with 256 nodes in spanwise and 128 nodes in chordwise direction, is defined. The pressure distribution is linearly interpolated on the new grids. By stacking both grids together, the resulting pressure distribution resolution on the surface is defined by 128 $\times$ 512 ($2^7 \times 2^9$) data points. Equal to the data representing the half-span model, both wings of the full-model are separated in two parts. Therefore, the resulting pressure distribution on the full-span model is defined by 128 $\times$ 1024 ($2^8 \times 2^{10}$) data points.

In Figure 6.27, the wing pressure distribution due to mode shape 11 in combination with the SAPRBS excitation is depicted at two different timesteps. As shown, although the interpolation has been applied, a high level of spatial resolution is maintained. The characteristic λ-shaped two-shock pattern as well as changes in the spanwise pressure distribution, in particular in the mid-span and tip region are still clearly visible.

In the second step, the data set is normalized to [-1,1] using the minimum and maximum pressure values of the numerical data set ($c_{p,min}$, $c_{p,max}$).

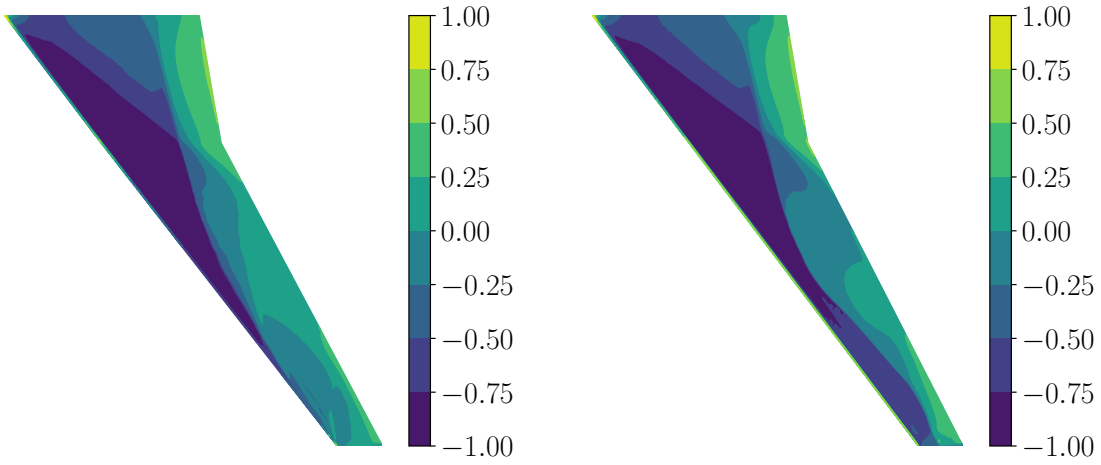


Figure 6.27: Interpolated and normalized surface c_p resolution on the wing suction side at two selected timesteps (mode 11, SAPRBS excitation).

6.4.5 Training of the Hybrid ROM

Analog to the training procedure of the hybrid CNN-VAR-AE/LSTM, as described in Section 5.4.3, the CNN-AE and LSTM are trained separately in two consecutive steps. In the following, two separate models for the symmetric and asymmetric mode shape c_p-distributions are trained.

Prior to the training, the data set is split into a training and validation data set. Therefore, 80% of the c_p-snapshots are used for training, whereas the remaining 20% are applied for validation and hyperparameter tuning. The CNN-AE is trained using batches, including 32 c_p-snapshots each. For both the encoder and decoder, four convolution levels are applied. Considering both the data representing the symmetric and asymmetric mode shape pressure distributions, the spatial dimension of the input data is reduced from 128×512 to 4×16 and 128×1024 to 4×32, respectively. The channel size is increased from 3 to 512. By passing the FC layer, the latent size is reduced from 512 and 1024 to 256 features, respectively. The initial learning rate is defined as 10^{-4}. In Table 6.7, a summary of the hyperparameters of the CNN-AE training are provided.

Encoder convolution layers	4
Decoder convolution layers	4
Latent dimension	256
Kernel size	4x4
Stride	2
Batch size	32
Initial learning rate	$1 \cdot 10^{-4}$
Activation	ReLU

Table 6.7: Hyperparameters for the training of the CNN-AE.

The training of both CNN-AEs is terminated after 15000 epochs. In Figure 6.28, the trends of the training and validation losses of both CNN-AEs are visualized. As

shown, the training and validation loss of mode 11 show a clear convergence, whereas the validation loss of mode 14 stays constant after an initial number of epochs.

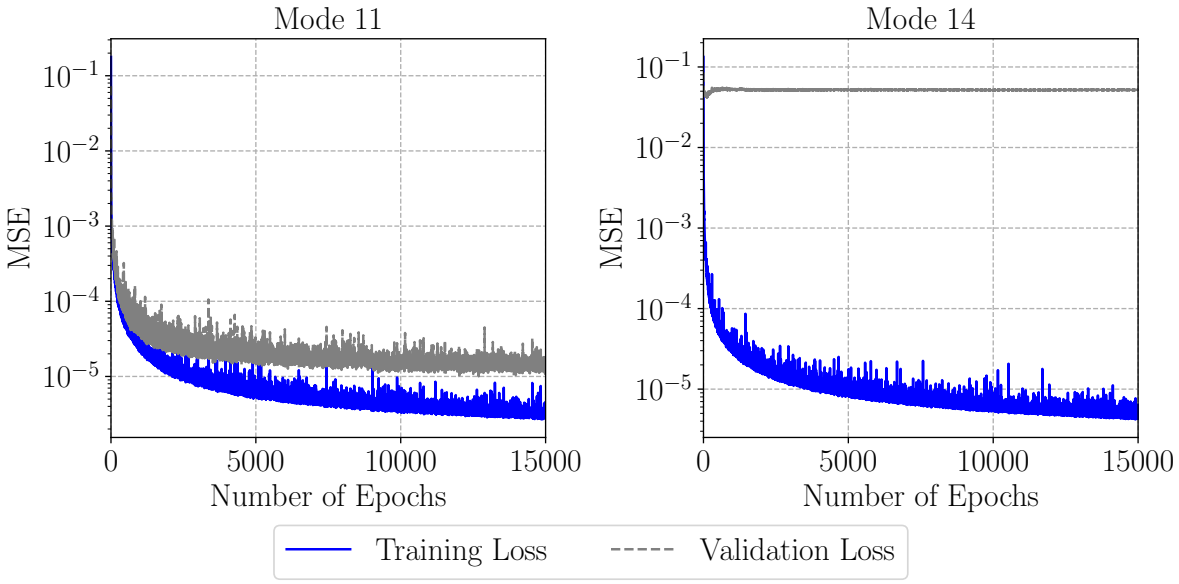


Figure 6.28: Convergence trends of training and validation losses of both trained CNN - AEs (mode 11 (left) and mode 14 (right)).

In addition to monitoring the convergence trends of the training and validation losses, a visual comparison between a numerical c_p-snapshot from the validation data set and the corresponding CNN-AE prediction is provided. In Figure 6.29, a numerical c_p-snapshot (left) and the corresponding prediction of the CNN-AE (middle) are exemplary visualized for mode shape 11. Based on the distribution of the MSE (right), which is almost zero on the entire wing surface, a precise reconstruction quality of the trained CNN-AE is indicated.

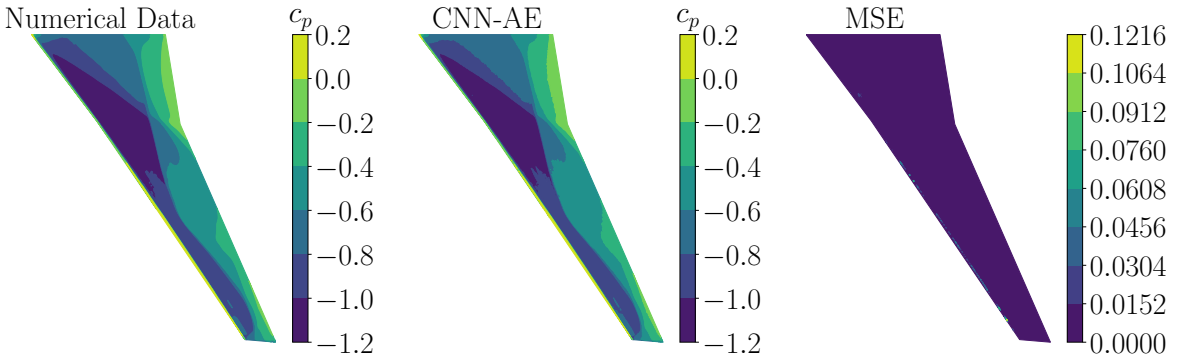


Figure 6.29: Comparison of a numerical validation c_p-snapshot (left) and a c_p-snapshot obtained by the trained CNN-AE (middle) (mode 11, SAPRBS excitation). The corresponding MSE is shown on the right wing surface.

In Figure 6.30, a comparison between the numerical and CNN-AE solution for mode shape 14 is visualized. Although the convergence of the validation loss saturates in the

early stages of the training process, the trained CNN-AE is able to reconstruct the numerical pressure distribution. Based on the MSE (Figure 6.30 (right)), which is almost zero on the entire wing, a sufficient reconstruction capability is indicated.

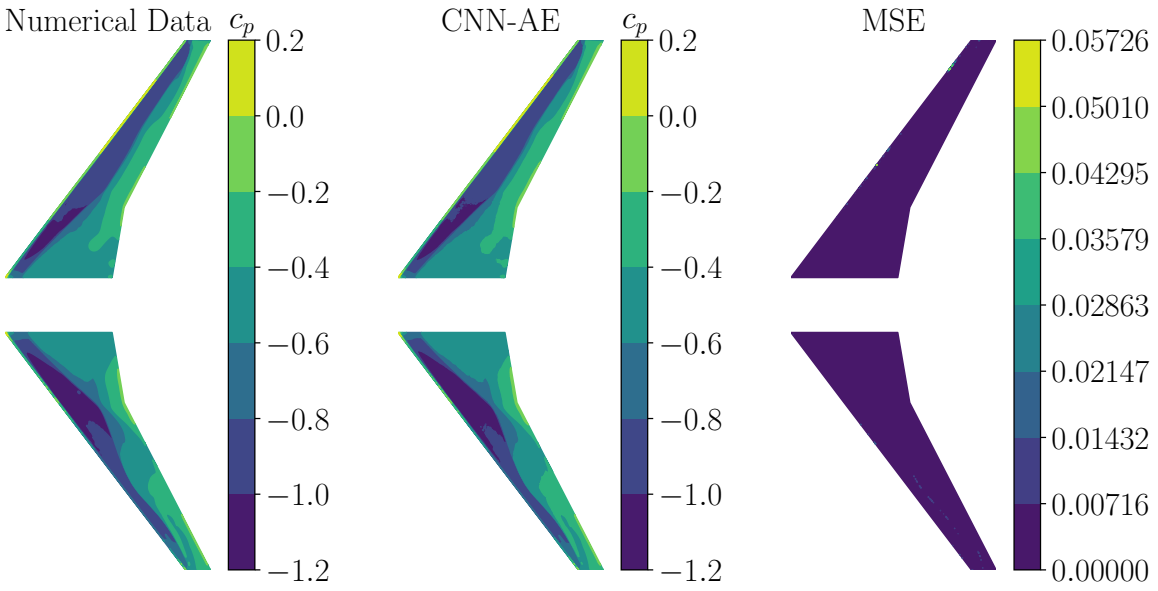


Figure 6.30: Comparison of a numerical validation c_p-snapshot (left) and a c_p-snapshot obtained by the trained CNN-AE (middle) (mode 14, SAPRBS excitation). The corresponding MSE is shown on the right wing surface.

Equal to the training of the CNN-AE, 80% of the data points are applied for the training of the LSTM. The data is fed into the LSTM in sequences, including 32 data points each, whereas the batch size is defined as one. A stacked LSTM with three layers is applied, containing 256 neurons per layer. Analogous to the training of the CNN-AE, the initial learning rate is defined as 10^{-4}. The training of the LSTM is terminated after 15000 epochs. In Table 6.8, a summary of the LSTM hyperparameters is provided.

Hidden layers	3
Neurons per layer	256
Sequence length	32
Batch size	1
Initial learning rate	$1 \cdot 10^{-4}$
State activation	$tanh$
Gate activation	σ

Table 6.8: Hyperparameters for the training of the LSTM.

Prior to the application of the trained hybrid ROM to an unknown test data set, the ROM is applied on the validation data set in a recurrent multi-step prediction mode. Therefore, an initial sequence of 32 c_p-snapshots is fed into to the trained model. In Figure 6.31, a comparison between a c_p-snapshot of the fully resolved CFD simulation (left) and the corresponding prediction of the hybrid ROM (middle) for mode shape 11

is provided. Compared to the pressure distribution as modelled by the CNN-AE (see Figure 6.29), minor deviations between the numerical and ROM solution are indicated. The position of the shock is captured by the hybrid ROM, however, the predicted magnitude of the shock is less pronounced compared to the numerical solution. The deviation in shock intensity is defined by slightly higher MSE values.

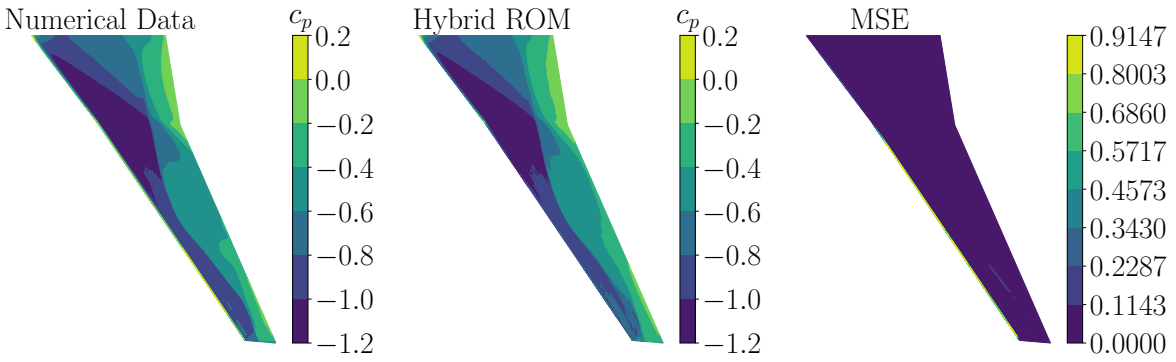


Figure 6.31: Comparison of a numerical validation c_p-snapshot (left) and a c_p-snapshot obtained by the trained hybrid ROM (middle) (mode 11, SAPRBS excitation). The corresponding MSE is shown on the right wing surface.

Similar to mode shape 11, a comparison between a numerical c_p-snapshot of mode 14 and the corresponding prediction of the hybrid ROM as presented in Figure 6.32 shows minor deviations at the shock position. Examining the corresponding MSE (Figure 6.32, right), increased MSE values along the shock position in the midspan and tip region, especially on the starboard wing, are present.

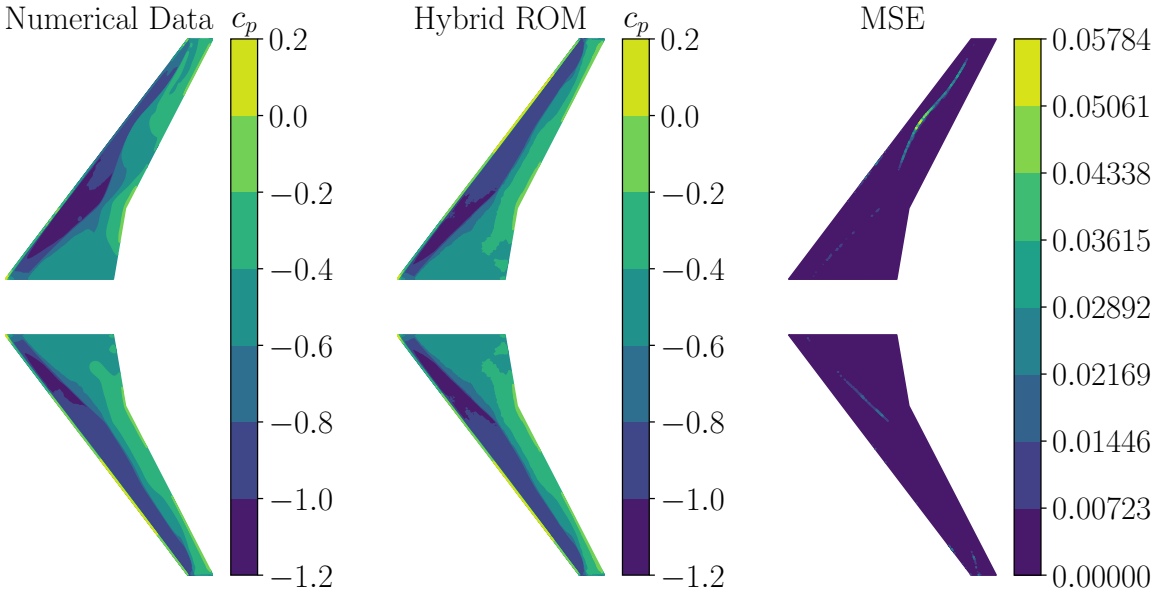


Figure 6.32: Comparison of a numerical validation c_p-snapshot (left) and a c_p-snapshot obtained by the trained hybrid ROM (middle) (mode 14, SAPRBS excitation). The corresponding MSE is shown on the right wing surface.

6.4.6 Performance Evaluation

In order to evaluate the performance of the trained hybrid ROM on an unknown data set, the ROM is applied for modeling the surface c_p-distribution at buffet condition in combination with a mode shape deformation excited by a harmonic oscillation. In order to represent a typical wing buffet frequency of the CRM configuration, the reduced frequency of the harmonic oscillation is defined as $k_{red} = 0.6$. Further, two deformation amplitudes, which are equal to reference scaling factors of $f = [0.5\%, 1\%]$, are defined for the harmonic excitation. Due to computational constraints concerning the fully resolved simulations, only a single oscillation period is computed with both the CFD solver and the trained ROM for each harmonic motion.

In the first step, the trained CNN-AEs are applied in order to reconstruct pressure distributions due to the harmonic deformations. In Figure 6.33, a numerical c_p-snapshot (left) included in the test data and the corresponding, reconstructed c_p-snapshot (middle) for mode shape 11 are compared. Here, the test case with a deflection amplitude of $f = 1\%$ is considered. Similar to the results presented in Figure 6.29 it is shown that the trained CNN-AE is able to reconstruct the pressure distribution with a high degree of accuracy. As depicted in Figure 6.33 (right), the MSE on the entire wing surface is almost zero.

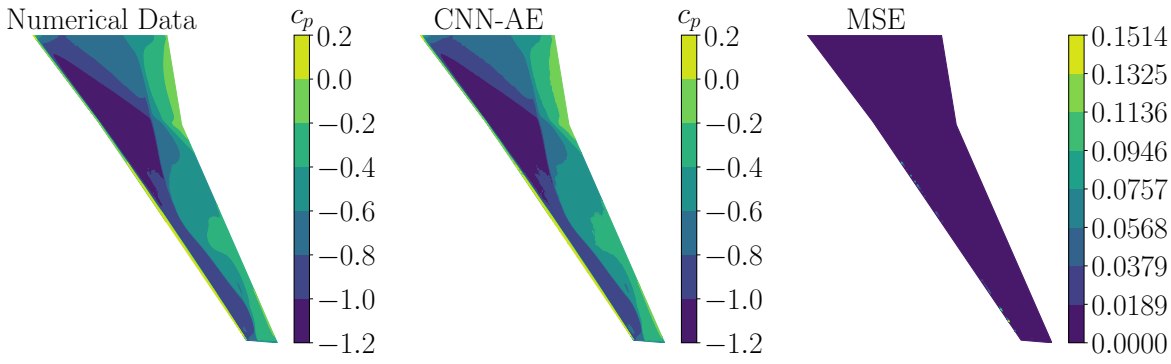


Figure 6.33: Comparison of a numerical c_p-snapshot (left) and a c_p-snapshot obtained by the trained CNN-AE (middle) (mode 11, harmonic deformation, $f = 1\%$). The corresponding MSE is shown on the right wing surface.

In Figure 6.34, a comparison between the numerical and reconstructed solution of mode shape 14 is provided. Analogous to Figure 6.33, the test data set based on the scaling factor of $f = 1\%$ is applied. As shown, the MSE on the port side indicates minor deviations along the shock position, whereas the MSE on the starboard side is almost zero on the entire surface. However, although some disagreement between the numerical and CNN-AE solution are indicated, the CNN-AE is able to capture the overall pressure distribution on both the starboard and port wing.

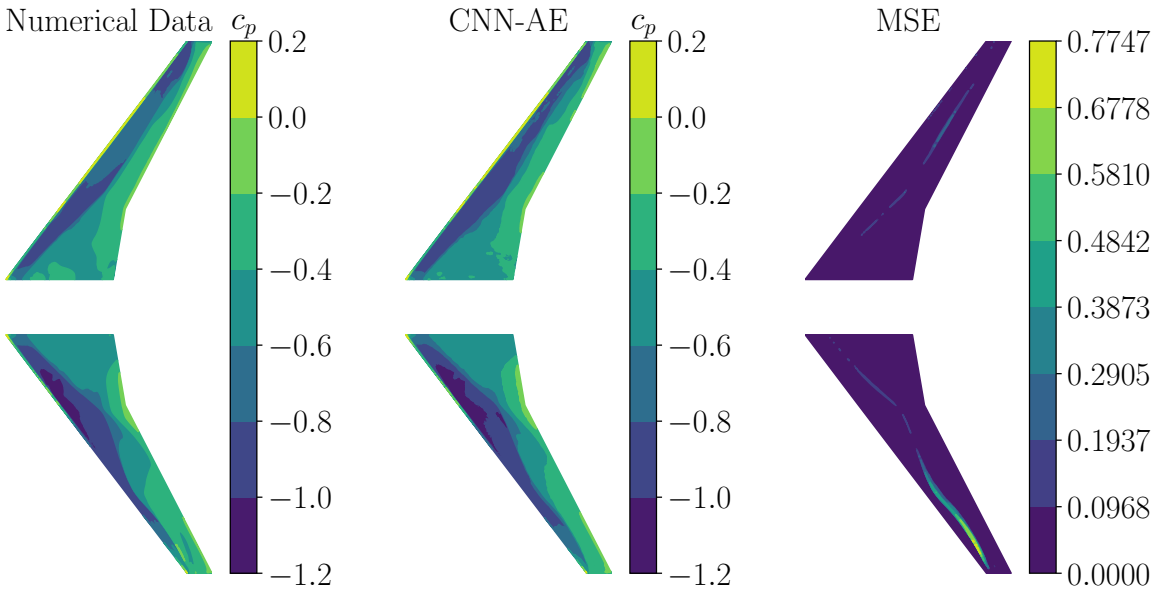


Figure 6.34: Comparison of a numerical c_p-snapshot (left) and a c_p-snapshot obtained by the trained CNN-AE (middle) (mode 14, harmonic deformation, $f = 1\%$). The corresponding MSE is shown on the right wing surface.

For the performance evaluation of the hybrid ROM, the c_p-samples representing both test data sets are encoded and applied as an input to the trained LSTM. The LSTM is applied in a recurrent multi-step prediction mode. For the initialization of the multi-step predictions, a sequence of ten encoded numerical c_p-snapshots is provided to the LSTM. In addition, the corresponding initial 10 timesteps of the harmonic deformations signal are concatenated with the pressure samples. Based on the initialization solution, the trained ROM is applied for the prediction of 150 timesteps ahead.

In order to compare the c_p-distributions as modelled by the hybrid ROM with the reference numerical data, chordwise c_p-distributions at two different spanwise positions η = [70%, 80%] are considered. These positions are chosen since they mark locations with distinct changes in the surface pressure distribution. At each position, the c_p-distribution at four timesteps included in the multi-step predictions is evaluated. The timesteps are set as $t_n = t_0 + \Delta t \cdot n$, with Δt defined by 30 timesteps. In Figure 6.35, a comparison of the numerical and predicted c_p-distributions of mode shape 11, simulated with a scaling amplitude of $f = 0.5\%$, is provided. Considering a spanwise position of $\eta = 80\%$, a good agreement between the numerical and ROM solution is indicated for timesteps t_0, t_1 and t_2. As the multi-step prediction advances, increased deviations in both the magnitude of c_p and the shock position, as shown at timestep t_3, occur. Similar to the data obtained at $\eta = 80\%$, an overall good agreement of the reference and modelled data at $\eta = 70\%$ is indicated. Although increased deviations including the representation of a double shock front at timesteps t_2 and t_3 are shown, the hybrid ROM is able to capture the overall trends of the pressure distributions.

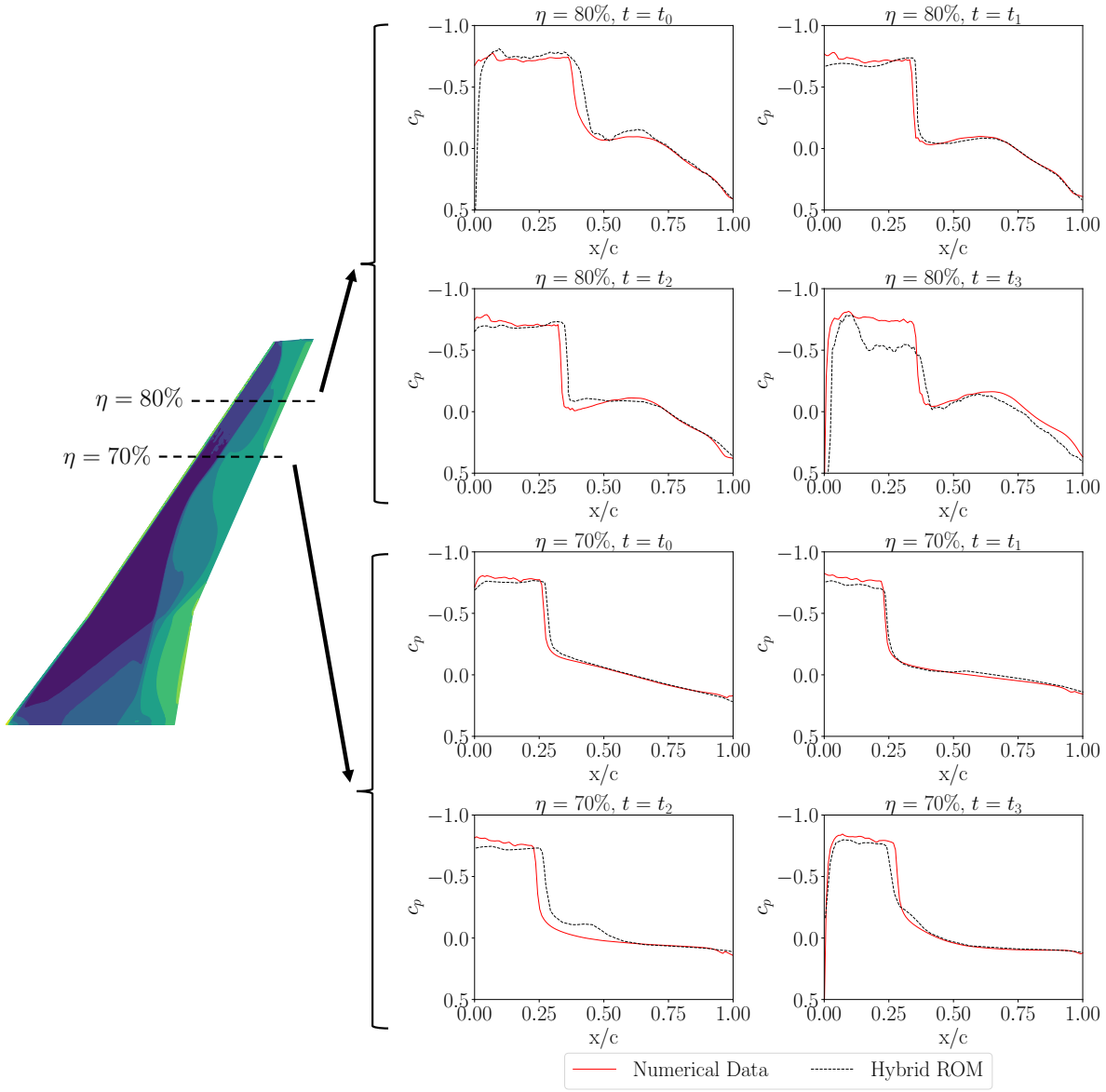


Figure 6.35: Comparison of numerical and predicted c_p-distributions at two spanwise positions $\eta = [70\%, 80\%]$ (mode 11, harmonic excitation, $f = 0.5\%$).

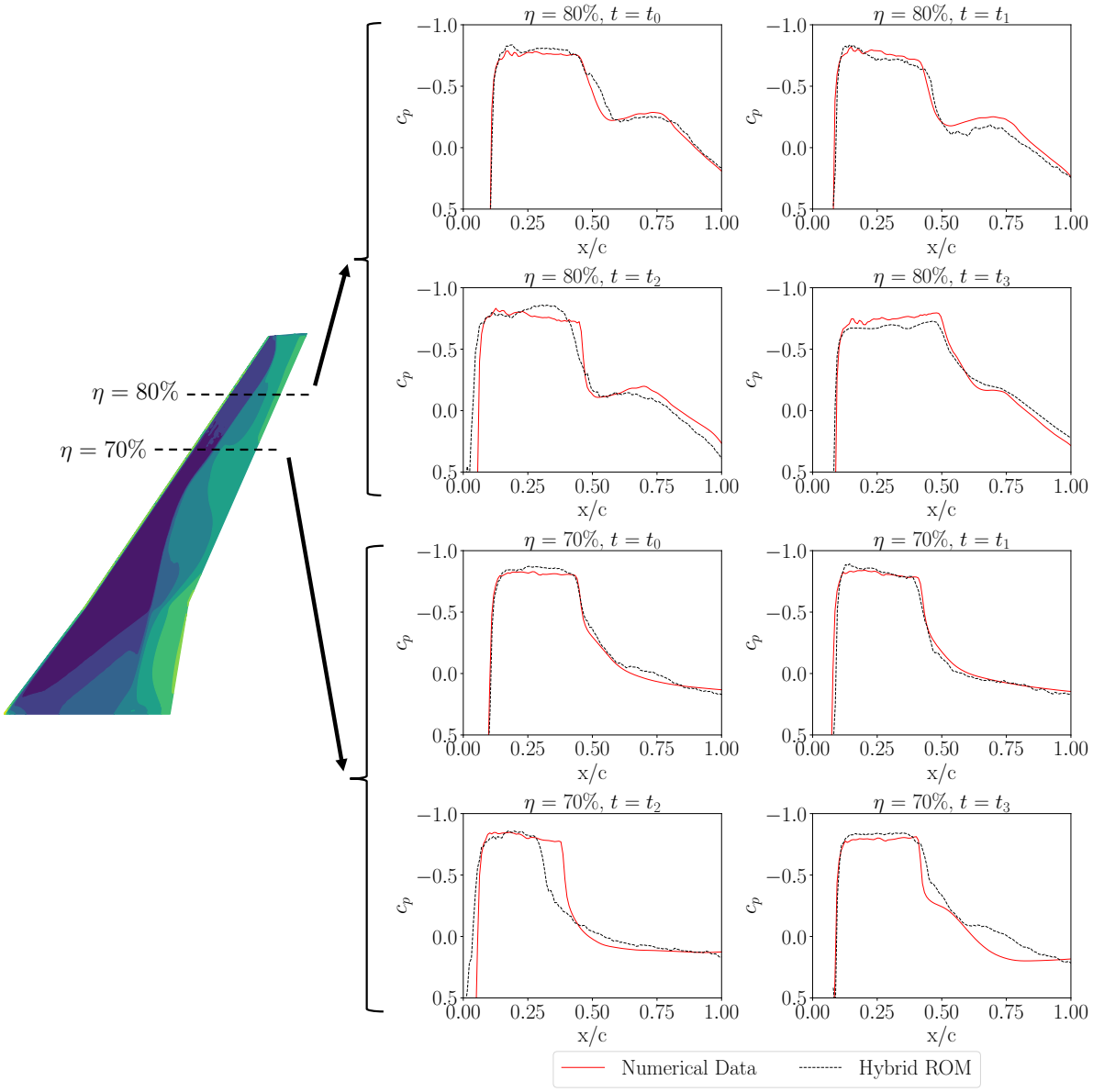


Figure 6.36: Comparison of numerical and predicted c_p-distributions at two spanwise positions $\eta = [70\%, 80\%]$ (mode 11, harmonic excitation, $f = 1\%$).

In addition, in Figure 6.36, a comparison of the numerical and predicted mode shape 11 pressure data, simulated with a scaling factor of $f = 1\%$, is presented. Similar to the results presented in Figure 6.35, the hybrid ROM is able to capture the c_p-distributions at $\eta = 80\%$ obtained at all timesteps. Here, only minor deviations in the c_p-magnitude are visible, whereas the shock position is correctly captured by the trained ROM. Considering the spanwise position $\eta = 70\%$, the c_p-distribution modelled by the hybrid ROM at timesteps t_0 and t_1 agree very well with the numerical solution. In contrast, the prediction of the c_p-distribution at the advanced timesteps seems to be a challenge for the trained ROM. Here, both deviations in the shock position as well as in the c_p-magnitude are indicated.

Shifting the focus to the test data sets representing mode shape 14, in Figure 6.37, a comparison of numerical pressure distributions and the corresponding predictions is provided. Here, pressure data obtained with a scaling factor of $f = 0.5\%$ is considered. At both spanwise positions, an overall good agreement in terms of shock position and c_p-magnitude at timesteps t_0 and t_1 is indicated. In contrast, considering the increased timesteps t_2 and t_3, the hybrid ROM is not able to capture either the magnitude of the c_p-distribution nor the position of the shock in a correct way. In Figure 6.38, a comparison between the numerical and ROM solutions, computed with a deformation factor of $f = 1\%$, is provided. Similar to the results presented in Figure 6.37, the hybrid ROM provides a good prediction performance for the c_p-distribution at timestep $t = [t_0,\ t_1]$ at both selected spanwise positions. As the timestep advances, the error of the multi-step predictions accumulates, resulting in increased deviations between the reference and modelled c_p-distributions.

In addition to a comparison of the chordwise c_p-distributions at the selected spanwise positions, a comparison of the surface c_p is provided for all four test cases at timestep t_2, as depicted in Figure 6.35 - 6.38. In Figure 6.39 and Figure 6.40, a comparison of the surface c_p due to the mode shape test cases with $f = 0.5\%$ and $f = 1\%$ is visualized, respectively. As shown, the hybrid ROM is able to capture the pressure distribution on the wing with a high degree of accuracy. However, consistent with the predicted chordwise c_p-distributions shown in Figure 6.35 and Figure 6.36, minor deviations in the shock position compared to the reference numerical solution are indicated. This is also emphasized by the increased MSE values at the position of the shock.

In Figure 6.41 and Figure 6.42, a visualization of the numerical and modelled c_p-distributions of the mode shape 14 test cases with $f = 0.5\%$ and $f = 1\%$ is provided, respectively. Consistent with the findings presented in Figure 6.37 and Figure 6.38, larger deviations at the shock position on the starboard side are indicated by the increased MSE values. However, besides the deviations in the c_p-distribution in the tip region, the hybrid ROM is able to model the c_p-distribution close to the wing root. Further, it has to be mentioned that both the c_p-distribution on the starboard and port side are captured by the hybrid ROM. However, the prediction of the c_p-distribution imposes a challenge on the proposed ROM compared to the prediction of mode shape 11 c_p-distribution.

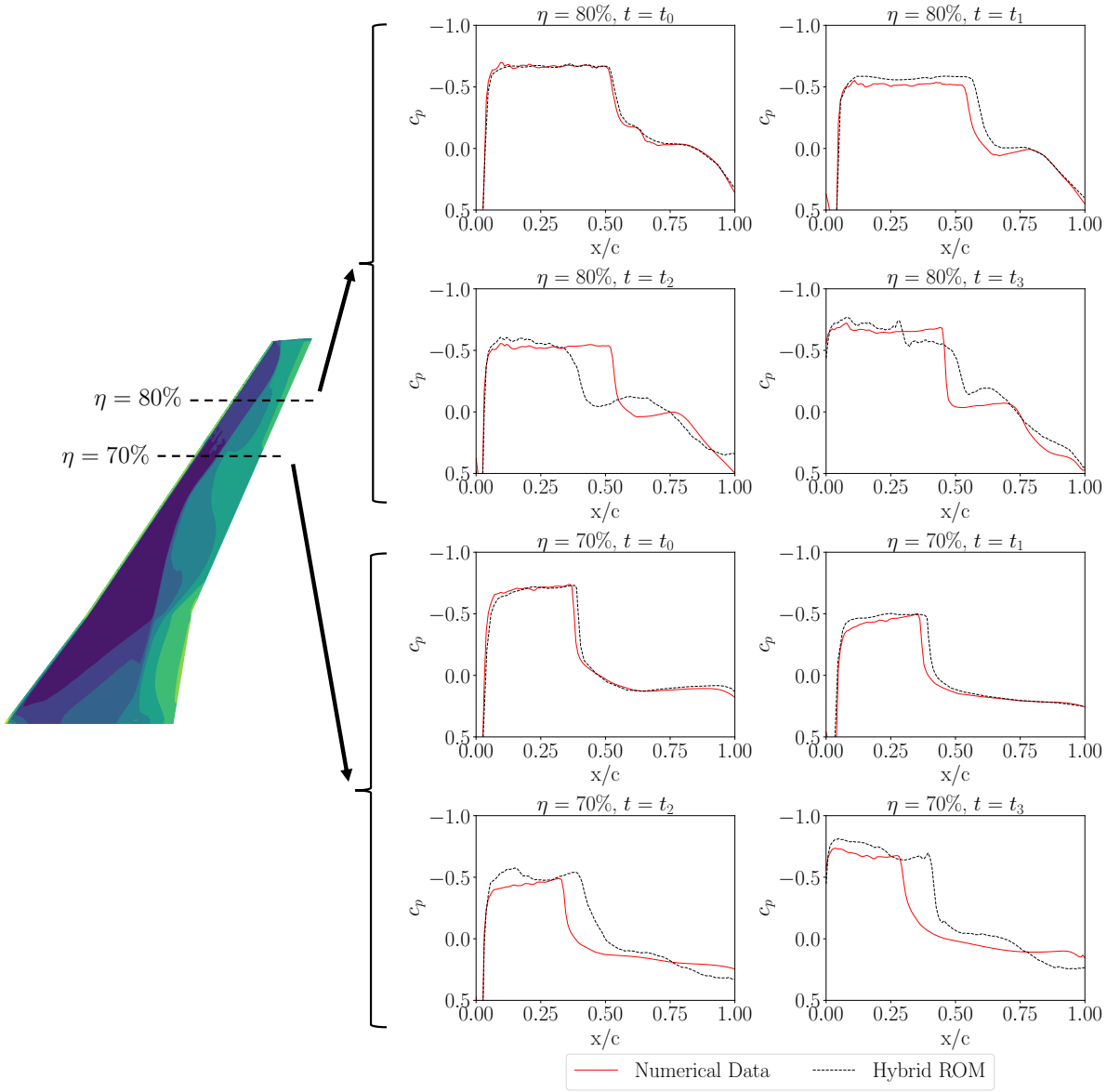


Figure 6.37: Comparison of numerical and predicted c_p-distributions at two spanwise positions $\eta = [70\%, 80\%]$ (mode 14, harmonic excitation, $f = 0.5\%$).

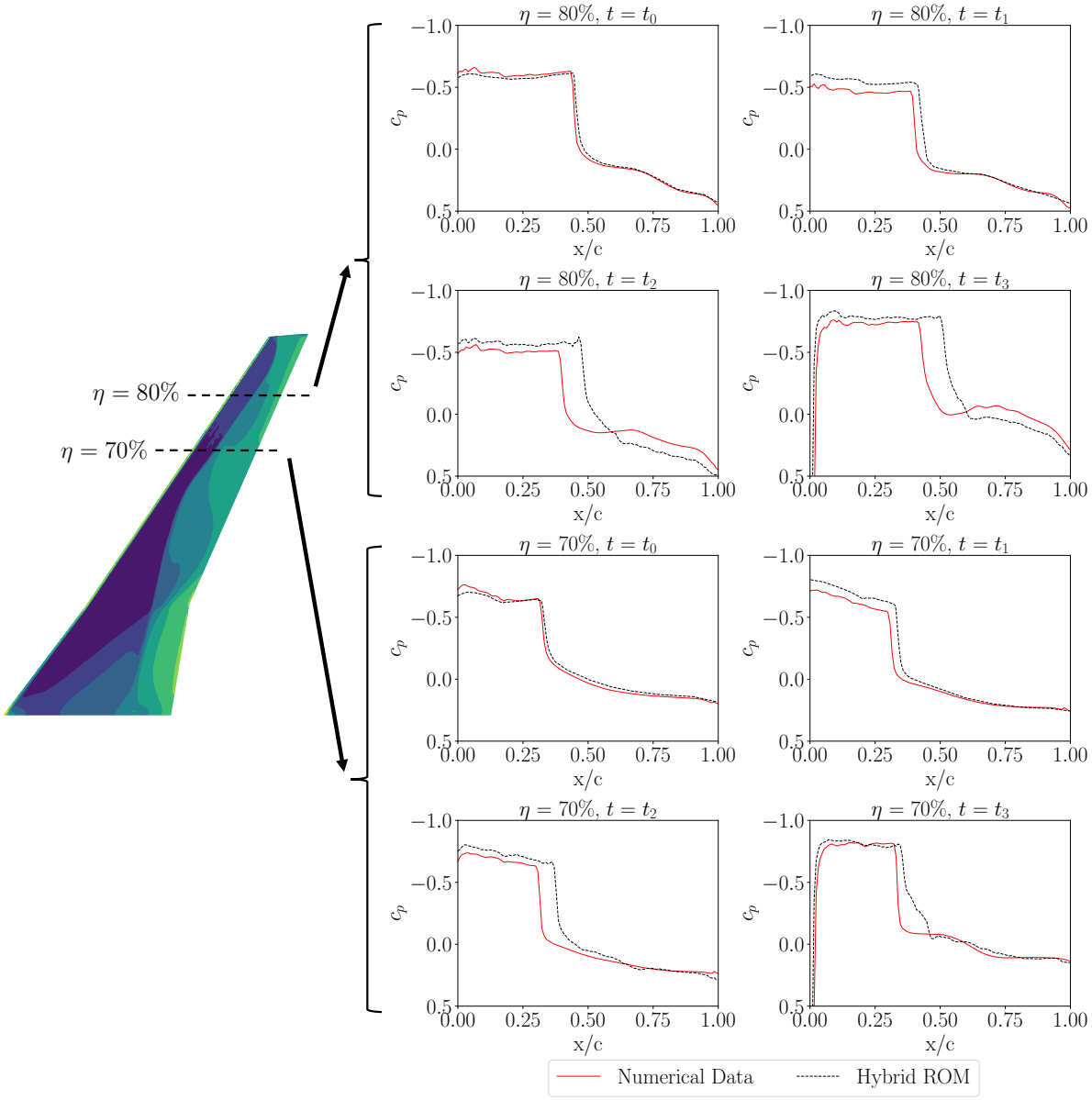


Figure 6.38: Comparison of numerical and predicted c_p-distributions at two spanwise positions $\eta = [70\%, 80\%]$ (mode 14, harmonic excitation, $f = 1\%$).

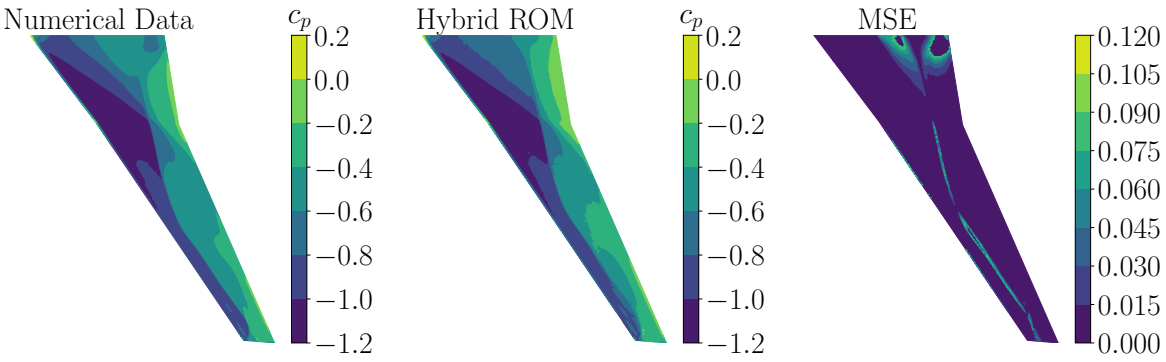


Figure 6.39: Comparison of a numerical c_p-snapshot (left) and a c_p-snapshot obtained by the trained hybrid ROM (middle) (mode 11, harmonic excitation, $f = 0.5\%$). The corresponding MSE is shown on the right wing surface.

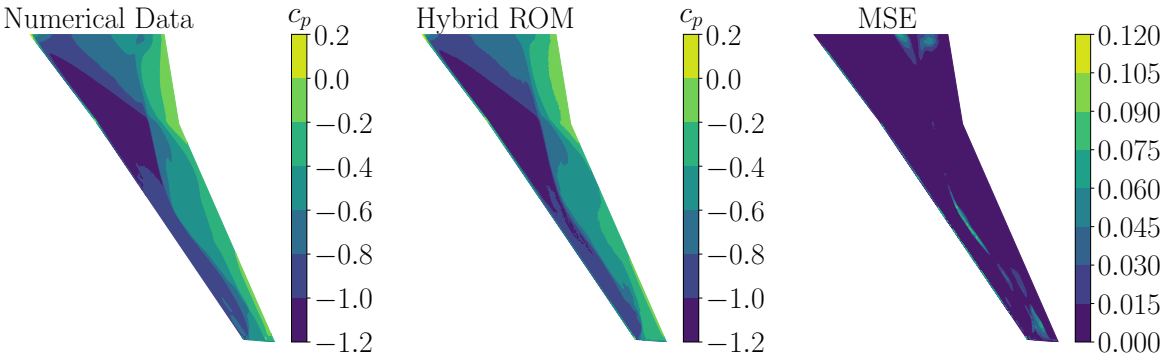


Figure 6.40: Comparison of a numerical c_p-snapshot (left) and a c_p-snapshot obtained by the trained hybrid ROM (middle) (mode 11, harmonic excitation, $f = 1\%$). The corresponding MSE is shown on the right wing surface.

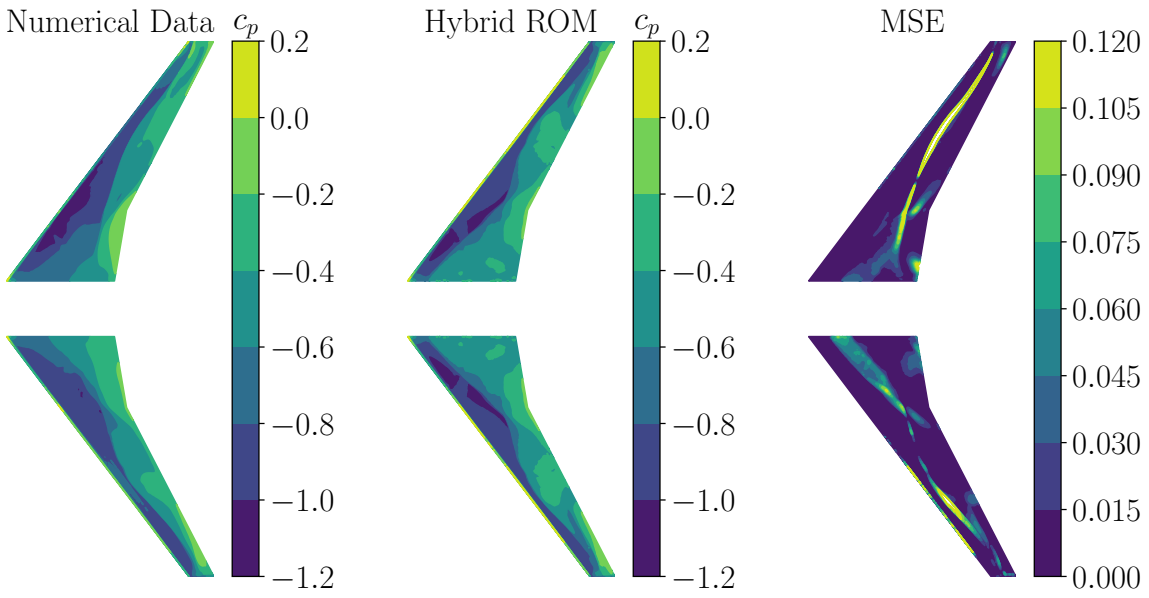


Figure 6.41: Comparison of a numerical c_p-snapshot (left) and a c_p-snapshot obtained by the trained hybrid ROM (middle) (mode 14, harmonic excitation, $f = 0.5\%$). The corresponding MSE is shown on the right wing surface.

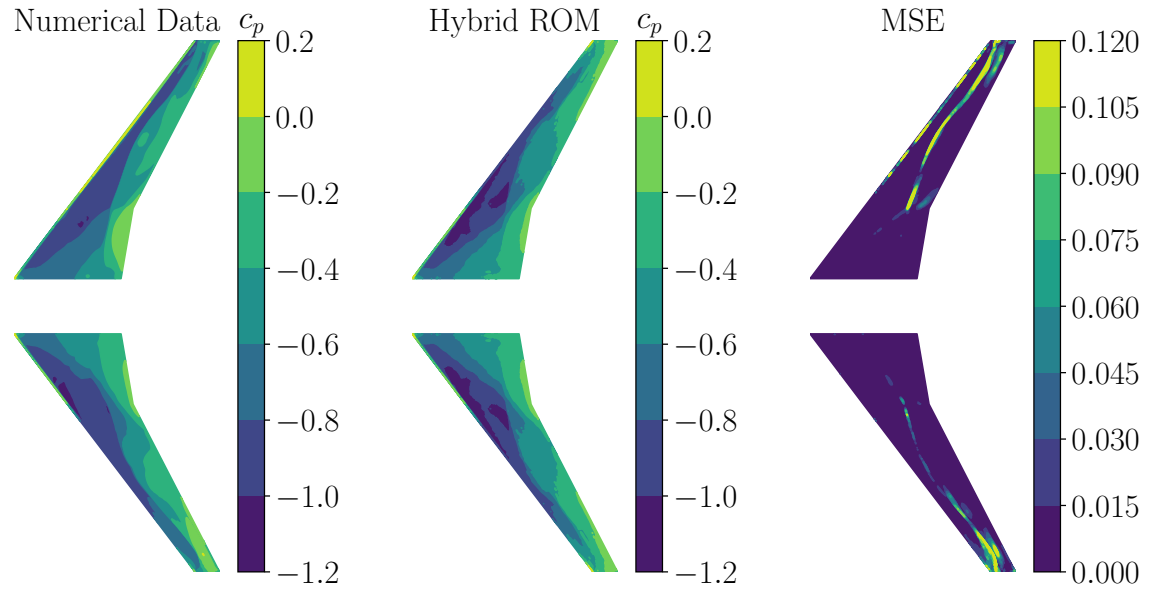


Figure 6.42: Comparison of a numerical c_p-snapshot (left) and a c_p-snapshot obtained by the trained hybrid ROM (middle) (mode 14, harmonic excitation, $f = 1\%$). The corresponding MSE is shown on the right wing surface.

6.4.7 Efficiency Evaluation

Within the following section, the reduction in computational costs by applying the proposed hybrid ROM in the framework of an aeroelastic analysis compared to URANS simulations, is quantified.

Analogous to previous numerical simulations presented in this thesis, the SuperMUC-NG of the LRZ has been used for the URANS computations with the DLR-TAU code. Equal to the computations presented in Section 6.3, 10 nodes with 48 cores are selected for the half-span model, resulting in a total number of 480 cores for each simulation. In contrast, the simulations of the full-span model have been conducted using 16 nodes with 48 cores each. Therefore, 768 cores are applied for the full-span model simulations. The training, validation and test computations of the hybrid ROM have been conducted on a workstation equipped with an Intel Xeon W-2295 3 GHz processor. Here, a single CPU core has been applied in order to provide the simulation time in terms of wall clock time.

As a first step, the costs for the training procedure of the hybrid ROM are analyzed. Considering the half-span and full-span model, the SAPRBS simulations for the generation of the training data set took approximately 38 and 97 hours on the SuperMUC-NG, respectively. Hence, considering the number of applied cores, the computational costs are summed up to 18240 ($= 38 \cdot 480$) and 74496 ($= 97 \cdot 768$) CPU hours. The training of the CNN-AE with the data representing mode 11 and mode 14 took around 30 hours on the workstation, while the training of the LSTM was summed up to approximately 20 hours for each respective mode. Therefore, the computational time for training the hybrid ROM on a single CPU is defined by 50 CPU hours. Considering the simulation time of each SAPRBS, the computational costs for the overall training are represented by 18290 and 74546 CPU hours for the half- and full-span model, respectively.

Following the efficiency evaluations provided in Section 4.7 and Section 6.3.4, a detailed overview of required CPU hours for the hybrid ROM and the fully-resolved URANS simulation is given. Considering the half-span model applied for symmetric mode simulations, the application of the hybrid ROM includes approximately 18240 CPU hours for the SAPRBS simulation, 50 CPU hours for the overall ROM training, a small number of initial timesteps for the multi-step predictions and about 0.03 CPU hours for the application of the hybrid ROM on a test data set. In contrast, the full URANS simulation comes up with a total of approximately 10000 CPU hours for each frequency-amplitude variation. Here, 10000 CPU hours are assumed as an average computational time, keeping in mind that each frequency is characterized by a different number of resolved timesteps. With a computational time of 10000 CPU hours, approximately 2000 timesteps are computed. Therefore, since a batch of 32 timesteps is provided for ROM initialization, approximately 160 CPU hours are required for the computation of the ROM initialization input.

Focusing on the full-span model, the ROM application is defined by approximately 74500 CPU hours for the SAPRBS simulation, 50 CPU hours for training of the ROM, initial time-steps for ROM initialization and approximately 0.04 CPU hours for the application of the ROM. The fully-resolved URANS simulations of the harmonic motions sum up to approximately an average of 20000 CPU hours. Therefore, 320 CPU hours are necessary for the timestep computations for ROM initialization. In Table 6.9, the number of CPU hours required for the application of the hybrid ROM and the full-resolved URANS computations are summarized. Here, only a single harmonic motion test case is considered.

		Half-span model	Full-span model
ROM	SAPRBS simulation	18240	74500
	ROM training	50	50
	ROM initialization	160	320
	ROM application	0.03	0.04
Fully resolved	URANS simulation	10000	20000

Table 6.9: Comparison of CPU hours of the half- and full-span model, considering a single harmonic motion test case.

Considering a proper aeroelastic analysis, a set of approximately 50 symmetric and asymmetric modes needs to be analyzed using URANS computations. Further, a fair amount of parameter variations for each mode would be needed, covering the frequency range of interest (e.g. $k_{red,Ex} = [0.2{:}0.1{:}0.8]$) and a minimum number of two amplitude variations ($f = [0.5\%, 1\%]$). Therefore, a number of 16 simulations must be conducted for each respective symmetric and asymmetric mode.

Focusing on the half-span model simulations, the number of ROM related CPU hours sum up to 18290 for the training of the hybrid ROM and 2560 ($= 16 \cdot 160$) CPU hours for the computation of the initial test data set solution for each frequency-amplitude variation. Further, the application of the ROM itself is defined by approximately 0.5 ($= 16 \cdot 0.03$) CPU hours. In contrast, the full URANS simulation would take $16 \cdot 10^4$ ($= 16 \cdot 10000$) CPU hours.

For the full-span model simulations, 74546 CPU hours for the training of the hybrid ROM as well as 5120 ($= 16 \cdot 320$) CPU hours for the initial test data set timesteps are required. In addition, approximately 0.65 ($= 16 \cdot 0.04$) CPU hours are set for the application of the hybrid ROM. For the full URANS simulation, in total, a computational time of $32 \cdot 10^4$ ($= 16 \cdot 20000$) CPU hours would be required.

In Figure 6.43, a comparison between ROM- and URANS-required CPU hours for both the half-span and full-span model is provided. Here, the CPU hours for a single mode computation are considered. As shown by the blue line, the increase of CPU hours required for the URANS simulations is very high compared to the CPU hours necessary for the ROM application, even after only a couple of test cases.

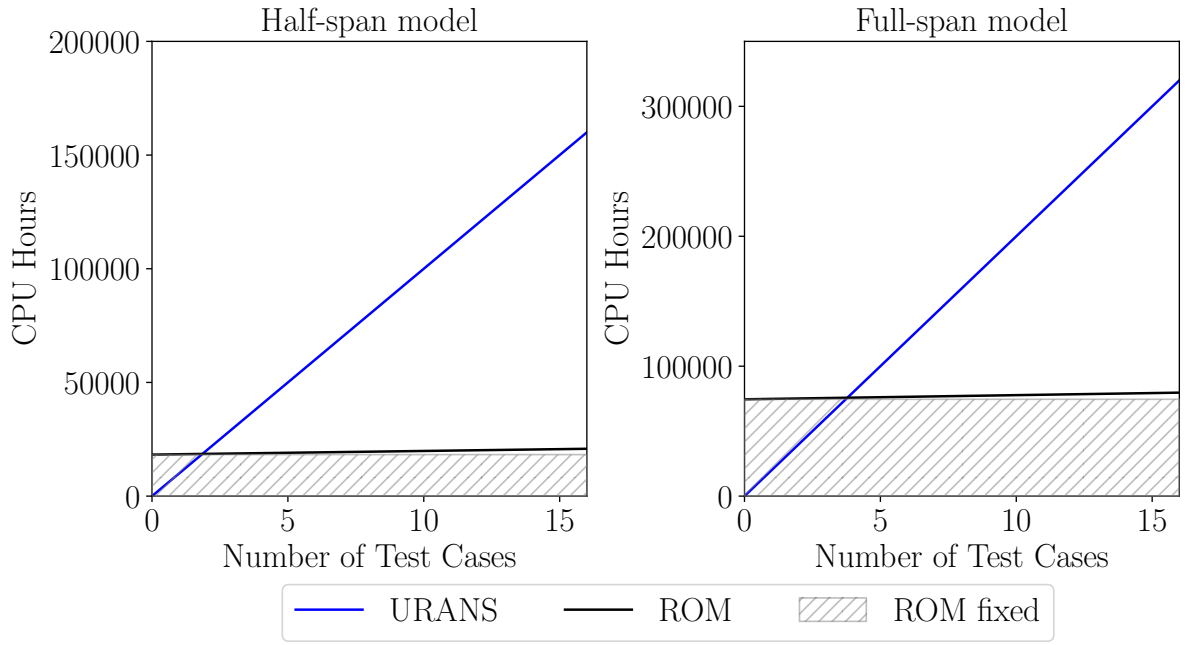


Figure 6.43: Comparison of CPU hours required for the fully-resolved URANS simulation and the hybrid ROM simulations (half-span model (left) and full-span model (right)).

Comparing the CPU hours required for the URANS simulations and the ROM application, a reduction in computational time by one order of magnitude is possible. However, if the CPU hours for the training data set simulations are neglected, computational time can be reduced by up to three orders of magnitude. Further, a reduction in computational costs could be achieved by training a hybrid ROM by means of superposed symmetric and asymmetric modes.

6.5 Summary

In this chapter, a LSTM-based ROM and a CNN-AE/LSTM-based hybrid ROM have been applied for the prediction of motion-induced integral and local wing buffet characteristics. As a test case, the NASA CRM configuration in combination with the FERMAT structural model has been selected.

In the first part of this chapter, the training and application of a LSTM-ROM for the prediction of integral wing buffet characteristics due to wing pitching motions was described. For the training of the LSTM, an optimized set of hyperparameters in terms of number of hidden layers, number of neurons, sequence length as well as initial learning rate has been defined. Equal to the analysis in Chapter 4, for performance assessment the trained LSTM has been used for the computation of lift and pitching moment coefficient trends due to prescribed harmonic pitching excitations, covering different frequencies and amplitudes. A comparison with the reference, high-fidelity fully resolved simulations showed that the LSTM is able to capture the trends of the resulting aerodynamic

coefficients to a sufficient extend. In particular, an overall fit in terms of the fit factor of 69% to 73.5% was achieved.

Further, it was shown that a hybrid CNN-AE/LSTM-based ROM approach is applicable for the prediction of wing buffet pressure distributions due to structural eigenmode-based deformations. However, compared to the analysis in Chapter 5, some conceptual changes have been applied to the architecture of the hybrid ROM. Instead of using a VAR-AE, only a standard AE has been used. In order to create data sets for different symmetric and asymmetric modes, a structural deformation framework based on the FERMAT structural model has been developed and implemented in the DLR-TAU solver. For the training and performance evaluation of the hybrid ROM, one symmetric and one asymmetric wing bending mode shape have been considered. Based on the results of the hybrid ROM application and a comparison with the fully resolved URANS simulations, it was shown that the hybrid ROM is able to capture the deformation-based pressure distributions of the symmetric mode shape. Further, a comparison between the pressure distributions of the asymmetric mode and the numerical solution revealed higher deviations, especially for advancing timesteps of the multi-step prediction mode. However, in general it was shown that the proposed ROM is also able to model asymmetric mode shape pressure distributions at a developed buffet condition. Besides the evaluation of the prediction performance it was shown that the application of the hybrid ROM, especially in the context of an aeroelastic analysis, potentially enables a saving of computational costs by several orders of magnitude.

To mitigate the deviations between the results of the ROM and the reference URANS solution, several strategies could be employed in the future: Firstly, increasing the information content of the training signal by increasing the number of training samples as well as incorporating additional physical parameters through enlarging the amplitude and frequency content of the applied signal. Secondly, taking advantage of advanced CPU or GPU hardware for both the ROM and URANS simulation could further accelerate and improve the training and application process.

7 Conclusion and Outlook

In the following final chapter, a recapitulation of this thesis is presented. The key findings are briefly outlined and summarized with respect to the research objectives as defined in Chapter 1. Further, an outlook on potential future reduced-order model (ROM) applications and their possible performance improvement is given.

As already motivated in the introduction, the focus of the present thesis was put on a fast and accurate computation of both integral and local two- and three-dimensional buffet characteristics using deep learning-based ROM methods. In particular, the focus lies on the computation of the buffet flow features due to varying freestream conditions as well as external excitations based on rigid body motions and eigenmode-based structural deformations. Besides an accurate prediction of the buffet characteristics, a reduction in computational time was pursued. For the training and evaluation of the proposed ROM methods, high-fidelity numerical and experimental data sets have been used. In order to obtain the data sets representing the relevant buffet flow physics, fully resolved Computational Fluid Dynamics (CFD) simulations using an Unsteady Reynolds-Averaged Navier-Stokes (URANS) approach have been performed. The experimental data sets have been obtained during several wind tunnel test campaigns at the European Transonic Windtunnel (ETW) facility.

The ROM approaches proposed in the present work are based on long short-term memory (LSTM) neural networks, convolutional neural networks (CNNs) and (variational) autoencoder (VAR-AE). Further, hybrid ROMs combining these architectures have been employed. The LSTM neural network has been applied for time-series prediction tasks, such as the trends of the unsteady aerodynamic forces and moments. In contrast, the CNN and AE enabled the computation of unsteady buffet pressure distributions and the reduction of the high-dimensional flow field data, respectively. For a comprehensive evaluation of the proposed ROMs, three unsteady buffet test cases have been taken into account.

As a first research task, outlined in Chapter 4, an accurate computation of two-dimensional buffet characteristics was pursued. As a test case, the NACA0012 airfoil has been selected. In particular, the aim was to predict changes in the lift and moment coefficient trends due to prescribed pitching and combined pitching and plunge excitations covering varying frequencies and amplitudes. Further, special focus was on an accurate reproduction of the lock-in effect, where the frequency of the buffet instability and the excitation frequency synchronize. For this task, a LSTM-based ROM was trained and applied. In order to generate an accurate training data set, URANS simulation using the DLR-TAU solver have been computed in order to capture the shock motion and the resulting separation and reattaching of the boundary layer. For performance evaluation, the results obtained by the trained LSTM have been compared to the reference full-order CFD solutions, using both time-domain and frequency-domain representations. Based on the results it was shown that the trained LSTM is able to

capture the trends of the aerodynamic coefficients. In addition, the lock-in effect was accurately modelled. Further, a detailed analysis of the application of the LSTM-ROM for reducing computational time compared to the full-order CFD simulations has been conducted. As shown, the application of a ROM compared to full-order CFD simulations for airfoil buffet analysis enables a computational speed-up of approximately one to two orders of magnitude.

In Chapter 5, an efficient computation of wing buffet characteristics at varying freestream conditions was presented. Therefore, a LSTM-based ROM has been applied for capturing local pressure fluctuations, whereas a hybrid CNN-VAR-AE/LSTM-based ROM has been used for the computation of unsteady wing pressure distributions. For the training and evaluation of both ROMs, experimental data measured during wind tunnel test campaigns in the ETW facility has been employed, with a wind tunnel model representing the Airbus XRF-1 configuration. Prior to the ROM application, a flow analysis based on the experimental pressure data revealed that the measured data include a high amount of noise. Besides the characteristic buffet frequencies, frequencies due to model vibrations have been captured in the pressure measurements. Therefore, applying the experimental data imposed a challenge to the training and application process of the ROMs. For performance evaluation, the results obtained by both ROMs are compared to the data obtained by the experiments. Based on the results it was indicated that both ROMs are able to model wing buffet characteristics due to varying freestream conditions with a high degree of accuracy.

In Chapter 6 the ROM methods introduced in Chapter 4 and Chapter 5 were applied for the prediction of wing buffet characteristics due to rigid body motions and eigenmode-based deformations. For the computation of motion-induced force- and moment coefficient trends, a LSTM-based ROM has been applied, whereas for the prediction of pressure distributions due to the structural deformations a hybrid CNN-AE/LSTM has been used. Here, both symmetric and asymmetric eigenmode-based deformations of the wing have been considered. For the training and validation, full-order CFD solutions have been applied, with the NASA CRM configuration as a test case. For structural modeling, the FERMAT structural model has been used. An examination of the results yielded an overall good agreement between the reference CFD solution and the results modeled by the trained ROMs. Further, computational time was reduced compared to the usage of extensive CFD simulations in the framework of an aeroleastic analysis.

Based on the previous outlined work packages, an evaluation of the research objectives is performed on the basis of the key findings in this thesis:

Research Objective 1: To what degree is it possible to predict unsteady integral and local buffet loads due to freestream parameter variations?

The data and discussion to answer this research objective is given in Chapter 5, where an accurate computation of unsteady integral and local buffet loads at varying freestream conditions was performed. Both pressure fluctuations measured at different span - and chordwise positions as well as pressure distributions on the entire wing are accurately modeled. Focusing on the locally measured pressure data, the characteristic, broadband buffet frequency content can be captured by a trained LSTM-ROM, although other frequencies related to structural vibrations are present in the data. A comparison between the experimental data and the data predicted by the LSTM-ROM revealed an overall fit between 76% to 80%. Similar findings apply for the computation of pressure distributions at buffet condition. A comparison with the reference experimental data revealed a good prediction capability of the proposed hybrid ROM for nonlinear flow field prediction. Considering both XRF-1 test cases with a different flow topology, the hybrid ROM was able to accurately model the pressure distribution on the clean wing and the wing with the UHBR nacelle installed. Further, a comparison of the POD modes of the original and predicted data showed that the hybrid ROM is able to capture the first six modes of the buffet instability with a high degree of accuracy. Both the low and high frequency content are correctly represented by the trained hybrid ROM.

Research Objective 2: To what degree is it possible to predict unsteady integral and local buffet loads due to rigid-body motions and eigenmode-based deflections?

Based on the results presented in Chapter 2 and Chapter 6, the proposed ROM methods are applicable for predicting motion-induced buffet characteristics. By means of a trained LSTM-ROM it was shown that it is possible to capture both lift and moment coefficient trends due to harmonic pitching and simultaneous pitching and plunge excitations on an airfoil and on an aircraft wing. A comparison of the ROM results and the reference CFD solutions indicated a fit of 90% - 93% and 70% - 73% for the prediction of airfoil and wing buffet aerodynamics, respectively. Further, focusing on airfoil buffet aerodynamics, the trained LSTM-ROM is able to capture the lock-in effect at harmonic excitations with frequencies close to the buffet frequency. Here, the synchronization of both frequencies is reproduced by the trained LSTM-ROM.

Focusing on pressure distribution modeling due to eigenmode-based deformations, the results presented in Chapter 6 revealed an efficient application of the hybrid CNN-AE/LSTM architecture. By extending the hybrid ROM architecture with the information regarding the structural deformations it was shown that it is possible to model the pressure distributions due to both symmetric and asymmetric eigenmode-based deformations. A comparison with the reference URANS solutions yielded that the trained

ROM is able to capture both the spanwise and chordwise c_p-distribution of the symmetric mode shape in terms of magnitude and shock position. In contrast, the application to the asymmetric mode test case yielded higher deviations between the numerical c_p-distribution and the c_p-distribution modelled by the hybrid ROM. Despite the challenge posed by the asymmetric mode deformation, it is important to emphasize that the application of the proposed ROM remains valid, and the overall prediction performance is deemed sufficient.

Research Objective 3: Is it possible to train and apply a ROM by means of experimental data, which is characterized by a high noise content?

In general, using data including a high amount of noise impose a challenge on the training and application of a ROM. Based on the experimental data analysis presented in Chapter 5, it was shown that besides buffet related frequencies, frequencies due to model vibrations are included in the experimental data. Removing nonphysical values is an important preprocessing step, when focusing on data gathered by optical measurements. However, as shown by the ROM results summarized in Chapter 5, an accurate computation of both integral and local wing buffet characteristics using experimental data, is achievable. Although a broadband noise content is present in all considered data spectra, the trained ROMs are able to capture both buffet- and structural-related frequencies with a high degree of accuracy.

Research Objective 4: How much reduction in computational time can be achieved by the application of a trained ROM compared to a full-order CFD or experimental solution?

Based on the results presented in Chapter 4 and Chapter 6, a detailed analysis regarding the efficiency of the ROM approaches relative to the full-order CFD application chain has shown that accelerations of one to two orders of magnitude in computational efficiency can be achieved. However, this factor is highly dependent on the number and complexity of the considered test cases. In general, it was shown that the generation of the reference or training input high-fidelity data set hold the highest share on the overall computational time which is necessary for the training of a ROM. Therefore, if the applied ROM method can amortize the time used for training, which is the case for the investigation of a large number of buffet freestream parameters or in the framework of an aeroelastic analysis, the usage of the proposed ROMs is profitable.

In contrast, a comparison of the computational effort of the ROM application with experimental investigations is more challenging, since it is highly dependent on the applied measurement technique. Considering the application of unsteady pressure sensors, data acquisition and storage time is very small compared to the training time of a ROM. Here, the computational efficiency of a ROM could be referred to the computation of

data if a sensor gets damaged during a test run or the data is insufficient for further analysis. Considering the application of optical measurement techniques such as iPSP, the saving in data acquisition costs compared to the ROM application is assumed to be very beneficial. Further, costs and time due to the intense preparation of the wing coating as well as the acquisition and preparation of the camera equipment could be saved by the application of a trained ROM.

Following the recap of this thesis, an outlook on ROM-based buffet and buffeting analysis is briefly given in the following. Driven by the progress in high-performance computing as well as artificial intelligence algorithms, various new deep learning (DL) methods originated within the last years. Besides the application of CNN-based ROMs, ROMs based on generative adversarial networks (GANs) represent a promising approach for the reconstruction and prediction of nonlinear flow fields. Further, DL methods such as convolutional LSTM (ConvLSTM) neural networks as well as temporal convolutional networks (TCN) are a promising method for the prediction of spatio-temporal flow field data. Further, coupling the proposed aerodynamic ROM approaches with structural dynamics or flight mechanics models enables the simulation of aeroelastic and aeroservoelastic problems, such as transonic buffeting. However, due to the computationally-intense simulations and the need of a sufficient data set covering the overall design space, the training and validation effort would drastically increase.

Besides the application of different or newly developed ROM approaches, the performance of the proposed ROM approaches could be further improved by the following measures: Instead of only using data from one source, data from different sources could be applied for the training of a ROM. It is assumed that a fusion of experimental and numerical data improves the training performance, due to the decreasing amount of noise in the data. However, combining data from different resources could also impose a challenge for a ROM, which has to be accounted for with a tailored preprocessing routine. Another approach for noise reduction in experimental data could be the application of modal analysis techniques such as POD and DMD prior to the training process. Therefore, only buffet relevant modes are considered for training and application. Further, as discussed before, the observed deviations between the ROM predictions and the reference data, which are most likely attributed to factors such as limited training data, data quality issues, and disparities in modeling assumptions between the neural network and the underlying models, underscore the need for a comprehensive solution. These issues could be tackled by advanced CPU or GPU hardware, enabling both optimized neural network implementations and applications as well as increased resources in terms of full-order reference data.

Incorporating the mentioned strategies for future improvements, the ROM methods developed in this thesis can be further adapted and applied to various new aerodynamic and aeroelastic problems.

Bibliography

[1] Y. Afshar, S. Bhatnagar, S. Pan, K. Duraisamy, and S. Kaushik. "Prediction of Aerodynamic Flow Fields Using Convolutional Neural Networks, Computational Mechanics". *Physics of Fluids*, Vol. 64:525–545, 2019.

[2] ANSYS Inc. *ANSYS ICEM CFD User Manual*, 14.5 edition, 2012.

[3] J. L. Ba, J. R. Kiros, and G. E. Hinton. "Layer Normalization". *arXiv preprint*, 2016.

[4] G. Barakos and D. Drikakis. "Numerical Simulation of Transonic Buffet Flows Using Various Turbulence Closures". *International Journal of Heat and Fluid Flow*, Vol. 21(5):620–626, 2000.

[5] Y. Bengio. *"Practical Recommendations for Gradient-Based Training of Deep Architectures"*. Neural Networks: Tricks of the Trade, Springer, 437-478, 2012.

[6] B. Benoit and I. Legrain. "Buffeting Prediction for Transport Aircraft Applications Based on Unsteady Pressure Measurements". In *Proceedings of the 5th AIAA Applied Aerodynamics Conference*, Monterey, CA, USA, 1987.

[7] G. Berkooz, P. Holmes, and J. L. Lumley. "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows". *Annual Review of Fluid Mechanics*, Vol. 25:539–575, 1993.

[8] F. M. Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi, and R. Jenssen. *"Recurrent Neural Networks for Short Term Load Forecasting: An Overview and Comparative Analysis"*. Springer Briefs in Computer Science, Springer, 2017.

[9] C. M. Bishop. "Pattern Recognition and Machine Learning". *Springer-Verlag, 1st Edition, New York, ISBN 978-0-387-31073-2*, 2006.

[10] V. Brunet and S. Deck. "Zonal-Detached Eddy Simulation of Transonic Buffet on a Civil Aircraft Type Configuration". In *Advances in Hybrid RANS-LES Modeling*, Kissimmee, FL, USA, 2015.

[11] C. Gao and W. Zhang and Y. Liu and Z. Ye and Y. Jiang. "Numerical Study on the Correlation of Transonic Single-Degree-of-Freedom Flutter and Buffet". *Science China Physics, Mechanics & Astronomy*, Vol. 58(8):1–12, 2015.

[12] M. Chandon, M. Esposito, O. Levinski, R. Carrese, N. Joseph, S. Koschel, and P. Marzocca. "On the Application of a Long Short-Term Memory Deep Learning Architecture for Aircraft Dynamic Loads Monitoring". In *60rd AIAA Structures, Structural Dynamics and Materials Conference, AIAA-2020-0702, Orlando, FL, USA*, 2020.

[13] S. Chen, C. F. N. Cowan, and P. M. Grant. "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks". *IEEE Transactions on Neural Networks*, Vol. 2(2):302–309, 1991.

[14] A. R. Collar. "The Expanding Domain of Aeroelasticity". *The Journal of the Royal Aeronautical Society*, Vol. 50(428):613–636, 1946.

[15] T. J. Cowan, A. S. Arena, and K. K. Gupta. "Accelerating Computational Fluid Dynamics Based Aeroelastic Predictions Using System Identification". *Journal of Aircraft*, Vol. 38(1):81–87, 2001.

[16] J. D. Crouch, A. Garbaruk, and D. Magidov. "Predicting the Onset of Flow Unsteadiness Based on Global Instability". *Journal of Computational Physics*, Vol. 224(2):924–940, 2007.

[17] J. D. Crouch, A. Garbaruk, D. Magidov, and A. Travin. "Origin of Transonic Buffet on Aerofoils". *Journal of Fluid Mechanics*, Vol. 628:357–369, 2009.

[18] J. D. Crouch, A. Garbaruk, and M. Strelets. "Global Instability Analysis of Unswept-and Swept-Wing Transonic Buffet Onset". In *Fluid Dynamics Conference, AIAA-2018-3229, Atlanta, GA, USA*, 2018.

[19] D. Schwamborn and T. Gerhold and R. Heinrich. "The DLR-TAU-Code, Recent Applications in Research and Industry". *European Conference on Computational Fluid Dynamics ECCOMAS CFD*, 2006.

[20] J. Dandois. "Experimental Study of Transonic Buffet Phenomenon on a 3D Swept Wing". *Physics of Fluids*, Vol. 28(1):016101, 2016.

[21] J. Dandois, P. Molton, A. Lepage, A. Geeraert, V. Brunet, and J. B. Dor et al. "Buffet Characterization and Control for Turbulent Wings". *AerospaceLab 6*, pages 1–17, 2013.

[22] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. "Identifying and Attacking the Saddle Point Problem in High-Dimensional Non-Convex Optimization". *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Vol. 2:2933–2941, 2014.

[23] S. S. Davis and G. N. Malcolm. "Transonic Shock-Wave/Boundary-Layer Interactions on an Oscillating Airfoil". *AIAA Journal*, Vol. 18(11):1306–1312, 1980.

[24] S. T. M. Dawson, N. K. Schiavone, C. W. Rowley, and D. R. Williams. "A Data-Driven Modeling Framework for Predicting Forces and Pressures on a Rapidly Pitching Airfoil". In *54th AIAA Fluid Dynamics Conference, AIAA Aviation Forum, Dallas, TX, USA*, 2015.

[25] S. Deck. "Numerical Simulation of Transonic Buffet Over a Supercritical Airfoil". *AIAA Journal*, Vol. 43(7):1556–1266, 2005.

[26] C. Despere, D. Caruanal, A. Mignosi, O. Reberga, and M. Correge. "Buffet Active Control-Experimental and Numerical Results". In *Proceedings of RTO AVT Symposium on Active Control Technology of Enhanced Performance Operational Capabilities of Military Aircraft*, Germany, 2001.

[27] J. Duchi, E. Hazan, and Y. Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". *Journal of Machine Learning Research*, Vol. 12:2121–2159, 2011.

[28] J. R. Edwards and S. Chandra. "Comparison of Eddy Viscosity-Transport Turbulence Models for Three-Dimensional, Shock Separated Flowfields". *AIAA Journal*, Vol. 34(4):756–763, 1996.

[29] J. W. Edwards and J. L. Thomas. "Computational Methods for Unsteady Transonic Flows". *Unsteady Transonic Aerodynamics*, Vol. 16(6):211–261, 1989.

[30] M. Ehrle, A. Waldmann, T. Lutz, and E. Krämer. "Simulation of Transonic Buffet with an Automated Zone DES Approach". *CEAS Journal*, Vol. 11:1025–1036, 2020.

[31] W. E. Faller and S. J. Schreck. "Unsteady Fluid Mechanics Applications of Neural Networks". *Journal of Aircraft*, Vol. 34(1):48–55, 1997.

[32] W. E. Faller, S. J. Schreck, and M. W. Luttges. "Neural Network Prediction and Control of Three-Dimensional Unsteady Separated Flowfields". *Journal of Aircraft*, Vol. 32(6):1213–1220, 1995.

[33] K. Finke. "Unsteady Shock Wave-Boundary Layer Interaction on Profiles in Transonic Flow". In *AGARD CP-183, Flow Separation*, Goettingen, Germany, 1975.

[34] G. Fouad and E.-W. Bai. "Block-Oriented Nonlinear System Identification". *Springer, London*, 2010.

[35] C. Gao, W. Zhang, and Z. Ye. "A New Viewpoint on the Mechanism of Transonic Single-Degree-of-Freedom Flutter". *Aerospace Science and Technology*, Vol. 52:144–156, 2016.

[36] E. Garnier and S. Deck. "Large-Eddy Simulation of Transonic Buffet Over a Supercritical Airfoil". *Direct and Large-Eddy Simulation VII, ERCOFTAC Series*, Vol. 13:549–554, 2010.

[37] M. Ghoreyshi, A. Jirásek, and R. M. Cummings. "Reduced Order Unsteady Aerodynamic Modeling for Stability and Control Analysis Using Computational Fluid Dynamics". *Progress in Aerospace Science*, Vol. 71:167–217, 2014.

[38] N. F. Giannelis and G. A. Vio. "Investigation of Frequency Lock-In Phenomena on a Supercritical Airfoil in the Presence of Transonic Shock Oscillations". In *Proceedings of the 17th International Forum of Aeroelasticity and Structural Dynamics*, Como, Italy, 2017.

[39] N. F. Giannelis, G. A. Vio, and G. Dimitriadis. "Dynamic Interactions of a Supercritical Airfoil in the Presence of Transonic Shock Buffet". In *Proceedings of the 27th International Conference on Noise and Vibration Engineering, Leuven, Belgium*, 2016.

[40] N. F. Giannelis, G. A. Vio, and O. Levinski. "A Review of Recent Developments in the Understanding of Transonic Shock Buffet". *Progress in Aerospace Science*, Vol. 92:39–84, 2017.

[41] E. Goncalves and R. Houdeville. "Turbulence Model and Numerical Scheme Assessment for Buffet Computations". *International Journal of Methods and Fluids*, Vol. 46(11):1127–1152, 2004.

[42] I. Goodfellow, Y. Bengio, and A. Courville. "Deep Learning". *MIT Press, 1st Edition, ISBN 978-0-2620-3561-3*, 2016.

[43] S. Görtz, M. Abu-Zurayk, C. Illic, T. Wunderlich, S. Keye, M. Schulze, T. Klimmek, C. Kaiser, Ö. Süelözgen, T. Kier, A. Schuster, S. Dähne, M. Petsch, D. Kohlgrüber, J. Häßy, R. Mischke, A. Weinert, P. Knechtges, S. Gottfried, J. Hartmann, and B. Fröhler. "Overview of Collaborative Multi-Fidelity Multidisciplinary Design Optimization Activities in the DLR Project VicToria". In *AIAA Aviation Forum 2022*, 2020.

[44] A. Hartmann, M. Klaas, and W. Schroeder. "Time-Resolved Stereo PIV Measurements of Shock-Boundary Layer Interaction on a Supercritical Airfoil". *Experimental Fluids*, 52(3):591–604, 2012.

[45] K. Hasegawa, K. Fukami, T. Muarata, and K. Fukagata. "Machine-Learning-Based Reduced-Order Modeling for Unsteady Flows Around Bluff Bodies of Various Shapes". *Theoretical and Computational Fluid Dynamics*, Vol. 34:367–383, 2020.

[46] S. Haykin. "Neural Networks: A Comprehensive Foundation". *Prentice Hall PTR, 2nd Edition, Upper Saddle River, NJ, USA, ISBN 978-0-1390-8385-3*, 1999.

[47] S. Haykin. "Neural Networks: A Comprehensive Foundation". *Prentice Hall PTR, 2nd Edition, Upper Saddle River, NJ, UAS, ISBN 978-0-1390-8385-3*, 1999.

[48] W. F. Hilton and R. G. Fowler. "Photographs of Shock Wave Movement". *National Physics Laboratory*, Vol. 2692, 1947.

[49] G. E. Hinton and R.S . Zemel. Autoencoders, minimum description length and helmholtz free energy. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems 6, 7th NIPS Conference, Denver, Colorado, USA, 1993*, pages 3–10, 1993.

[50] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". *Neural Computation*, Vol. 9(8):1735–1780, 1997.

[51] S. Illi, C. Fingskes, C. Lutz, and E. Krämer. "Transonic Tail Buffet Simulations for the Common Research Model". In *AIAA 2013-2510*, 2013.

[52] S. Ioffe and C. Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". *Proceedings of the 32nd International Conference on Machine Learning, PLMR*, Vol. 37:448–456, 2015.

[53] M. Iovnovich and D. E. Raveh. "Reynolds-Averaged Navier-Stokes Study of the Shock-Buffet Instability Mechanism". *AIAA Journal*, Vol. 50(4):880–890, 2012.

[54] M. Iovnovich and D. E. Raveh. "Numerical Study of Shock Buffet on Three-Dimensional Wings". *AIAA Journal*, Vol. 53(2):449–463, 2014.

[55] L. Jacquin, P. Molton, S. Deck, B. Maury, and D. Soulevant. "Experimental Study of Shock Oscillations Over a Transonic Supercritical Airfoil". *AIAA Journal*, Vol. 47(9):1985–1994, 2009.

[56] K. Lindhorst and M. C. Haupt and P. Horst. "Efficient Surrogate Modelling of Nonlinear Aerodynamics in Aerostructural Coupling Schemes". *AIAA Journal*, Vol. 52(9):1952–1966, 2014.

[57] D. P. Kingma and J. L. Ba. "Adam: A Method for Stochastic Optimization". *International Conference on Learning Representations*, Vol. 12, 2015.

[58] T. Klimmek. "Parametric Set-Up of a Structural Model for FERMAT Configuration Aeroelastic and Loads Analysis". *Journal of Aeroelasticity and Structural Dynamics*, Vol. 3(2):31–49, 2014.

[59] S. Koike, M. Ueno, K. Nakakita, and A. Hashimoto. "Unsteady Pressure Measurements of Transonic Buffet on the NASA Common Research Model". In *34th AIAA Aplied Aerodynamic Conference, AIAA-2016-4044, Washington D.C., USA*, 2016.

[60] J. Kou and W. Zhang. "Multi-Kernel Neural Networks for Nonlinear Unsteady Aerodynamic Reduced-Order Modeling". *Aerospace Science and Technology*, Vol. 67:309–326, 2017.

[61] J. Kou and W. Zhang. "Data-Driven Modeling for Unsteady Aerodynamics and Aeroelasticity". *Progress in Aerospace Science*, Vol. 125:110725, 2021.

[62] A. Kuzmin. "Bifurcations and Buffet of Transonic Flow Past Flattened Surfaces". *Computational Fluids*, Vol. 38(7):1369–1374, 2009.

[63] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin. "Exploring Strategies for Training Deep Neural Networks". *Journal of Machine Learning Research*, Vol. 10:1–40, 2009.

[64] S. G. Lawson, D. Greenwell, and M. Quinn. "Characterization of Buffet on a Civil Aircraft Wing". In *Proceedings of the 54th AIAA Aerospace Science Meeting*, San Diego, CA, USA, 2016.

[65] Y. LeCun. *Modeles Connexionistes de L'Apprentissage*. Ph.D. thesis, Universite de Paris, 1987.

[66] Y. LeCun. "Generalization and Network Design Strategies". *Technical Report CRG-TR89-4*, Vol. 326, 1989.

[67] B. H. K. Lee. "Oscillatory Shock Motion Caused by Transonic Shock Boundary-Layer Interaction". *AIAA Journal*, Vol. 28(5):942–944, 1990.

[68] B. H. K. Lee. "Self-Sustained Shock Oscillations on Airfoils at Transonic Speeds". *Progress in Aerospace Science*, Vol. 37(2):147–196, 2001.

[69] B. Van Leer. "Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel Godunov's method". *Journal of Computational Physics*, Vol. 32(1):101–136, 1979.

[70] K. Li, J. Kou, and W. Zhang. "Deep Neural Network for Unsteady Aerodynamic and Aeroelastic Modeling Across Multiple Mach Numbers". *Nonlinear Dynamics*, Vol. 96:2157–2177, 2019.

[71] Y. Li, J. Chang, Z. Wang, and C. Kong. "An Efficient Deep Learning Framework to Reconstruct the Flow Field Sequences of the Supersonic Cascade Channel". *Physics of Fluids*, Vol. 33:056106, 2021.

[72] M. S. Liou. "A Sequel to AUSM: AUSM+". *Journal of Computational Physics*, Vol. 129(2):364–382, 1975.

[73] T. Liu, J.P. Sullivan, K. Asai, C. Klein, and Y. Egami. *"Pressure and Temperature Sensitive Paints"*. Second Edition (Experimental Fluid Mechanics), Chapter 9, Springer, Berlin, Germany, 2021.

[74] L. Ljung. *"System Identification: Theory for the User"*. Prentice Hall PTR, 2nd Edition, Upper Saddle River, NJ, USA, ISBN 978-0-1365-6695-3, 1999.

[75] J. L. Lumley. "The Structure of Inhomogeneous Turbulence". In *Proceedings of the International Colloquium on the Fine Scale Structure of Atmosphere and its Influence on Radio Wave Propagation*, 1967.

[76] T. Lutz and P. P. Gansel. "Going for Experimental and Numerical Unsteady Wake Analyses Combined with Wall Interference Assessment by Using the NASA CRM-Model in ETW". In *51st AIAA Aerospace Sciences Meeting, AIAA-2013-871, Grapevine, TX, USA*, 2013.

[77] M. Tatar and M. H. Sabour. "Reduced-Order Modeling of Dynamic Stall Using Neuro-Fuzzy Inference System and Orthogonal Functions". *Physics of Fluids*, Vol. 32:2157–2177, 2020.

[78] D. G. Mabey, B. L. Welsh, and B. E. Cripps. "Periodic Flows on a Rigid 14% Thick Biconvex Wing at Transonic Speeds". *RAE-TR-81059, British Royal Aircraft Establishment*, 1981.

[79] A. Mann, G. Thompson, and P. White. "Civil Aircraft Wind Tunnel Feature Rich Testing at the Edge of the Envelope". In *Proceedings of the 54th 3AF International Conference on Applied Aerodynamics*, 2019.

[80] A. Mannarino and E. H. Dowell. "Reduced-Order Models for Computational-Fluid-Dynamics-Based Nonlinear Aeroelastic Problems". *AIAA Journal*, 53(9):2671–2685, 2015.

[81] A. Mannarino and P. Mantegazza. "Nonlinear Aeroelastic Reduced Order Modeling by Recurrent Neural Networks". *Journal of Fluids and Structures*, 48:103–121, 2014.

[82] F. D. Marques and J. Anderson. "Identification and Prediction of Unsteady Transonic Aerodynamic Loads by Multi-Layer Functionals". *Journal of Fluids and Structures*, Vol. 15:83–106, 2001.

[83] L. Masini, S. Timme, A. Ciarella, and A. Peace. "Influence of Vane Vortex Generators on Transonic Wing Buffet: Further Analysis of the BUCOLIC Experimental Dataset". In *Proceedings of the 52nd 3AF International Conference on Applied Aerodynamics*, Lyon, France, 2017.

[84] J. B. McDevitt, L. L. Levy Jr., and G. S. Deiwert. "Transonic Flow About a Thick Circular-Arc Airfoil". *AIAA Journal*, Vol. 14(5):606–613, 1976.

[85] J. B. McDevitt and A. F. Okuno. "Static and Dynamic Pressure Measurements on a NACA0012 Airfoil in the Ames High Reynolds Number Facility". *NASA TP-2485, National Aeronautics and Space Administration*, 1985.

[86] F. R. Menter. "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications". *AIAA Journal*, Vol. 32(8):1598–1605, 1994.

[87] T. Nakamura, K. Fukami, K. Hasegawa, Y. Nabae, and K. Fukagata. "Convolutional Neural Network and Long Short-Term Memory Based Reduced Order Surrogate for Minimal Turbulent Channel Flow". *Physics of Fluids*, Vol. 33:025116, 2021.

[88] O. Nelles. "Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models". *Springer-Verlag, 1st Edition, Berlin, Heidelberg, ISBN 978-3-662-04323-3*, 2001.

[89] Y. A. Ng. "Feature Selection, L_1 vs. L_2 Regularization, and Rotational Invariance". In *Proceedings of the Twenty-First International Conference on Machine Learning*, New York, NY, USA, 2004.

[90] J. Nitzsche. "A Numerical Study on Aerodynamic Resonance in Transonic Separated Flow". In *Proceedings of the International Forum on Aeroelasticity and Structural Dynamics*, Seattle, WA, USA, 2009.

[91] O. San and R. Maulik and M. Ahmed. "An Artificial Neural Network Framework For Reduced Order Modeling of Transient Flows". *Communications in Nonlinear Science and Numerical Simulation*, Vol. 77:271–287, 2019.

[92] Y. Ohmichi, T. Ishida, and T. Hashimoto. "Modal Decomposition Analysis of Three-Dimensional Transonic Buffet Phenomenon on a Swept Wing". *AIAA Journal*, Vol. 56(10):3938–3950, 2018.

[93] E. Paladini, J. Dandois, D. Sipp, and J.C. Robinet. "Analysis and Comparison of Transonic Buffet Phenomenon over Several Three-Dimensional Wings". *AIAA Journal*, Vol. 57(1):1–18, 2018.

[94] S.O. Park, K.H. Jun, S.M. Baek, M.H. Cho, K.J. Yee, and D.H. Lee. "Reduced-Order Model with Artificial Neural Network for Aerostructural Design Optimization". *Journal of Aircraft*, Vol. 50(4):1106–1116, 2013.

[95] R. Pascanu, T. Mikolov, and Y. Bengio. "On the Difficulty of Training Recurrent Neural Networks". *IEEE Transactions on Neural Networks*, Vol. 5(2):157–166, 2013.

[96] B. T. Polyak. "Some Methods of Speeding Up the Convergence of Iteration Methods". *USSR Computational Mathematics and Mathematical Physics*, Vol. 4(5):1–17, 1964.

[97] J. Quan, W. Zhang, C. Gao, and Z. Ye. "Characteristic Analysis of Lock-In For an Elastically Suspended Airfoil in Transonic Buffet Flow". *Chinese Journal of Aeronautics*, Vol. 29(1):129–143, 2016.

[98] S. Raghunathan, R. D. Mitchell, and M. A. Gillan. "Transonic Shock Oscillations on NACA0012 Airfoil". *Shock Waves*, Vol. 8:191–202, 1998.

[99] D. E. Raveh. "Reduced-Order Models for Nonlinear Unsteady Aerodynamics". *AIAA Journal*, Vol. 39(8):1417–1429, 2001.

[100] D. E. Raveh. "Identification of Computational-Fluid-Dynamics Based Unsteady Aerodynamic Models for Aeroelastic Analysis". *Journal of Aircraft*, Vol. 41(1):620–632, 2004.

[101] D. E. Raveh. "Numerical Study of an Oscillating Airfoil in Transonic Buffeting Flows". *AIAA Journal*, Vol. 47(3):505–515, 2009.

[102] D. E. Raveh and E. H. Dowell. "Frequency Lock-In Phenomenon for Oscillating Airfoils in Buffeting Flows". *Journal of Fluids and Structures*, Vol. 27(1):89–104, 2011.

[103] H. Robbins and S. Monro. "A Stochastic Approximation Method". *Annals of Mathematical Statistics*, Vol. 22:400–407, 1951.

[104] P. L. Roe. "Approximate Riemann Solvers, Parameters, Vectors and Difference Schemes". *Journal of Computational Physics*, Vol. 43(2):357–372, 1981.

[105] F. Roos. "The Buffeting Pressure Field of a High-Aspect-Ratio Swept Wing". In *Proceedings of the 18th Fluid Dynamics and Plasmadynamics and Lasers Conference*, Cincinatti, OH, USA, 1985.

[106] V. Rosov and C. Breitsamter. "Data-Driven Prediction of Unsteady Pressure Distributions Based on Deep Learning". *Journal of Fluids and Structures*, Vol. 104:103316, 2021.

[107] O. Rouzaud, S. Plot, and V. Couaillier. "Numerical Simulation of Buffeting Over Airfoil Using Dual Time Stepping Method". In *Proceedings of the European Conference on Computational Fluid Dynamics ECCOMAS CFD*, Barcelona, Spain, 2000. Springer-Verlag.

[108] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning Representations by Back-Propagating Errors". *Nature*, Vol. 323:533–536, 1986.

[109] C. L. Rumsey, M. D. Sanetrik, R. T. Biedron, N. D. Melson, and E. B. Parlette. "Efficiency and Accuracy of Time-Accurate Turbulent Navier-Stokes Computations". *Computational Fluids*, Vol. 25(2):217–236, 1996.

[110] D. M. Russel. "Error Measures for Comparing Transient Data: Part 1: Development of Comprehensive Error Measure". *Proceedings of the 68th Shock and Vibration Symposium, Hunt Valley, MD*, Vol. 1:175–184, 1997.

[111] S. Walton and O. Hassan and K. Morgan. "Reduced Order Modelling For Unsteady Fluid Flow Using Proper Orthogonal Decomposition and Radial Basis Functions". *Applied Mathematical Modelling*, Vol. 37(20-21):8930–8945, 2013.

[112] F. Sator, C. Mettot, and D. Sipp. "Stability, Receptivity and Sensitivity Analysis of Buffeting Transonic Flow Over a Profile". *AIAA Journal*, Vol. 53(7):1980–1993, 2014.

[113] F. Sator and S. Timme. "Reynolds-Averaged Navier-Stokes Simulations of Shock Buffet on Half Wing-Body Configuration". In *Proceedings of the 53rd AIAA Aerospace Sciences Meeting*, Kissimmee, FL, USA, 2015.

[114] F. Sator and S. Timme. "Delayed Detached-Eddy Simulation of Shock-Buffet on Half Wing-Body Configuration". *AIAA Journal*, Vol. 55(4):1230–1240, 2016.

[115] A. M. Schaefer and H.-G. Zimmermann. "Recurrent Neural Networsk are Universal Approximators". *International Journal of Neural Systems*, Vol. 17(4):253–263, 2007.

[116] P. J. Schmid. "Dynamic Mode Decomposition of Numerical and Experimental Data". *Journal of Fluid Mechanics*, Vol. 656:5–28, 2010.

[117] V. Sekar, Q. Jiang, C. Shu, and B. C. Khoo. "Fast Flow Field Prediction over Airfoils Using Deep Learning Approach". *Physics of Fluids*, Vol. 31:057103, 2019.

[118] H. Shimodaira. "Improving Predictive Inference Under Covariate Shift by Weighting the Log-Likelihood Function". *Journal of Statistical Planning and Inference*, Vol. 90(2):227–244, 2000.

[119] A. Soda and N. Verdon. "Investigation of Influence of Different Modelling Parameters on Calculation of Transonic Buffet Phenomena". In *New Results in Numerical and Experimental Fluid Mechanics V*, pages 487–495. Springer-Verlag, 2006.

[120] P. R. Spalart and S. R. Allmaras. "A One-Equation Turbulence Model for Aerodynamic Flows". *Recherche Aerospatiale*, Vol. 1:5–21, 1994.

[121] P.R. Spalart. "Strategies for Turbulence Modeling and Simulations". *Journal of Heat Fluid Flow*, Vol. 21(3):252–263, 2000.

[122] S. Spinner and R. Rudnik. "Design of a UHBR Through Flow Nacelle for High Speed Stall Wind Tunnel Investigations". In *Deutscher Luft- und Raumfahrtkongress, DGLR*, Bremen, Germany, 2021. Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V.

[123] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". *Journal of Machine Learning Research*, Vol. 15:1929–1958, 2014.

[124] P. C. Steimle, D. C. Karhoff, and W. Schroeder. "Unsteady Transonic Flow Over a Transport-Type Swept Wing". *AIAA Journal*, Vol. 50(2):399–415, 2012.

[125] S. Suresh, S. N. Omkar, V. Mani, and T. N. Guru Prakash. "Lift Coefficient Prediction at High Angles of Attack Using Recurrent Neural Networks". *Aerospace Science and Technology*, Vol. 7(8):595–602, 2003.

[126] T. Tieleman and G. Hinton. "Lecture 6.5.rmsprop: Divide the Gradient by Running Average of its Recent Magnitude". *Coursera: Neural Networks for Machine Learning*, Vol. 4:26–31, 2012.

[127] S. Timme. "Global Instability of Wing Shock-Buffet Onset". *Journal of Fluid Mechanics*, Vol. 885, 2020.

[128] S. Timme and R. Thormann. "Towards Three-Dimensional Global Instability Analysis of Transonic Shock Buffet". In *Proceedings of the AIAA Atmospheric Flight Mechanics Conference*, Washington D.C, USA, 2016.

[129] V. Togiti, B. Eisfeld, and O. Brodersen. "Turbulence Model Study For the Flow Around the NASA Common Research Model". *Journal of Aircraft*, Vol. 54(4):1331–1343, 2014.

[130] J. C. Vassberg, M. Dehann, M. Rivers, and R. Wahls. "Development of a Common Research Model for Applied CFD Validation Studies". In *26^{th} AIAA Applied Aerodynamics Conference and Exhibit, Honolulu, HI, USA, AIAA Paper*, pages 1–22, 2008.

[131] J. C. Vassberg, E.N. Tinoco, M. Mani, B. Rider, T. Zickuhr, D.W. Levy, O.P. Brodersen, B. Eisfeld, S. Crippa, R.A. Wahls, J.H. Morrison, D.J. Mavriplis, and M. Murayama. "Summary of the Fourth AIAA Computational Fluid Dynamics Drag Prediction Workshop". *Journal of Aircraft*, Vol. 51(4):1070–1089, 2014.

[132] A. Waldmann, C. Ehrle, J. Kleinert, and T. Lutz. "Mach and Reynolds Number Effects on Transonic Buffet on the XRF-1 Transport Aircraft Wing at Flight Reynolds Number". *Experiments in Fluids*, Vol. 64(102), 2022.

[133] A. Weiner and R. Seeman. "FlowTorch - a Python Library for Analysis and Reduced-Order Modeling of Fluid Flows". *Journal of Open Source Software*, Vol. 6(68):3860, 2021.

[134] P. J. Werbos. "Backpropagation Through Time: What it does and how to do it". *Proceedings of the IEEE*, Vol. 78(10):1550–1560, 1990.

[135] D. R. Wilson and T. R. Martinez. "The General Inefficiency of Batch Training for Gradient Descent Learning". *Neural Networks*, Vol. 16(10):1429–1451, 2003.

[136] M. Winter. *Nonlinear Aerodynamic Reduced-Order Modeling Using Neuro-Fuzzy Approaches*. Dissertation, Chair of Aerodynamics and Fluid Mechanics, Department of Mechanical Engineering, Technical University of Munich, Garching, Germany, 2021.

[137] M. Winter and C. Breitsamter. "Reduced-Order Modeling of Unsteady Aerodynamic Loads Using Radial Basis Function Neural Networks". In *63rd Deutscher Luft- und Raumfahrtkongress, DGLR*, Augsburg, Germany, 2014. Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V.

[138] M. Winter and C. Breitsamter. "Efficient Unsteady Aerodynamic Loads Prediction Based on Nonlinear System Identification and Proper Orthogonal Decomposition". *Journal of Fluids and Structures*, Vol. 67:1–21, 2016.

[139] M. Winter and C. Breitsamter. "Application of Unsteady Aerodynamic Reduced-Order Modeling Techniques to a Complex Configuration". In *17th International Forum on Aeroelasticity and Structural Dynamics (IFASD)*, Como, Italy, 2017.

[140] M. Winter and C. Breitsamter. "Coupling of Recurrent and Static Neural Network Approaches for Improved Multi-step Ahead Time Series Prediction". In Dillmann et al., editor, *New Results in Numerical and Experimental Fluid Mechanics XI*, pages 433–442, Cham, 2018. Springer-Verlag.

[141] M. Winter and C. Breitsamter. "Nonlinear Identification via Connected Neural Networks for Unsteady Aerodynamic Analysis". *Aerospace Science and Technology*, Vol. 77:802–818, 2018.

[142] M. Winter and C. Breitsamter. "Reduced-Order Modeling of Transonic Buffet Aerodynamics". In Dillmann et al., editor, *New Results in Numerical and Experimental Fluid Mechanics XII*, pages 511–520, Cham, 2020. Springer-Verlag.

[143] K. Won, H. M. Tsai, M. Sadeghi, and F. Liu. "Non-Linear Impulse Methods for Aeroelastic Simulations". In *23rd AIAA Applied Aerodynamics Conference and Exhibit, Toronto, Canada*, 2005.

[144] M. Wu, Z. Han, H. Nie, W. Song, C.L. Clainche, and E. Ferrer. "A Transition Prediction Method For Flows Over Airfoils Based on High-Order Dynamic Mode Decomposition". *Chinese Journal of Aeronautics*, Vol. 32(11):2408–2421, 2019.

[145] Q. Xiao, H. M. Tsai, and F. Liu. "Numerical Study of Transonic Buffet on a Supercritical Airfoil". *AIAA Journal*, Vol. 44(3):620–628, 2006.

[146] Y. Sugioka and S. Koike and K. Nakakita and D. Numata and T. Nonomura and K. Asia. “Experimental Analysis of Transonic Buffet on a 3D Swept Wing Using Fast-Response Pressure-Sensitive Paint“. *Experiments of Fluids*, Vol. 59(6):108, 2018.

[147] W. Yao and M.-S. Liou. “Reduced-Order Modeling for Limit-Cycle Oscillations Using Recurrent Artificial Neural Network“. In *12th AIAA Aviation Technology Integration and Operations Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Indianapolis, IN, USA*, 2012.

[148] D. Yorita, C. Klein, U. Henne, V. Ondrus, U. Beifuss, A.-K. Hensch, P. Guntermann R. Longo, and J. Quest. “Successful Application of Cryogenic Pressure Sensitive Paint Technique at ETW“. In *55th AIAA Aerospace Sciences Meeting, AIAA SciTech, Kissimee, FL*, 2018.

[149] R. Zahn and C. Breitsamter. “Airfoil Buffet Aerodynamics at Plunge and Pitch Excitation Based on Long Short-Term Memory Neural Network Prediction“. *CEAS Aeronautical Journal*, Vol. 13:45–55, 2022.

[150] R. Zahn and C. Breitsamter. “Prediction of Transonic Wing Buffet Pressure Based on Deep Learning“. *CEAS Aeronautical Journal*, Vol. 14:155–169, 2023.

[151] R. Zahn, T. Linke, and C. Breitsamter. “Neural Network Modeling of Transonic Buffet on the NASA Common Research Model“. *New Results in Numerical and Experimental Fluid Mechanics XIII*, Vol. 115:697–706, 2021.

[152] R. Zahn, A. Weiner, and C. Breitsamter. “Prediction of Wing Buffet Pressure Loads Using a Convolutional and Recurrent Neural Network Framework“. *CEAS Aeronautical Journal*, 2023.

[153] R. Zahn, M. Winter, M. Zieher, and C. Breitsamter. “Application of a Long Short-Term Memory Neural Network for Modeling Transonic Buffet Aerodynamics“. *Aerospace Science and Technology*, Vol. 113, 2021.

[154] W. Zhang, C. Gao, Y. Liu, Z. Ye, and Y. Jiang. “The Interaction Between Flutter and Buffet in Transonic Flow“. *Nonlinear Dynamics*, Vol. 82(4):1851–1865, 2015.

[155] W. Zhang, B. Wang, Z. Ye, and J. Quan. “Efficient Method for Limit Cycle Flutter Analysis by Nonlinear Aerodynamic Reduced-Order Models“. *AIAA Journal*, Vol. 50(5):1019–1028, 2012.

List of Publications

Peer-Reviewed Publications

- **R. Zahn** and C. Breitsamter, "Neuro-Fuzzy Network-Based Reduced-Order Modeling of Transonic Aileron Buzz", *Aerospace*, Vol. 67, pp. 13-30, 2020.

- **R. Zahn**, M. Winter, M. Zieher, and C. Breitsamter, "Application of a Long Short-Term Memory Neural Network for Modeling Transonic Buffet Aerodynamics", *Aerospace Science and Technology*, Vol. 113, 2021.

- **R. Zahn**, T. Linke, and C. Breitsamter, "Neural Network Modeling of Transonic Buffet on the NASA Common Research Model", *New Results in Numerical and Experimental Fluid Mechanics XIII*, Springer Verlag, Vol. 151, pp. 697-706, 2021.

- **R. Zahn** and C. Breitsamter, "Airfoil Buffet Aerodynamics at Plunge and Pitch Excitation Based on a Long Short-Term Memory Neural Network Prediction", *CEAS Aeronautical Journal*, Vol. 13, pp. 45-55, 2022.

- **R. Zahn** and C. Breitsamter, "Prediction of Transonic Wing Buffet Pressure Based on Deep Learning", *CEAS Aeronautical Journal*, Vol. 14, pp. 155-169, 2023.

- **R. Zahn**, A. Weiner, and C. Breitsamter, "Prediction of Wing Buffet Pressure Loads Using a Convolutional and Recurrent Neural Network Framework", *CEAS Aeronautical Journal*, 2023.

Conference Contributions

- **R. Zahn** and C. Breitsamter, "High-Speed Buffet Aerodynamics Modeling Based On a Long Short-Term Memory Neural Network", *69th Deutscher Luft- und Raumfahrtkongress* (DLRK), Paper 0027, Aachen, Germany, 2020.

- **R. Zahn** and C. Breitsamter, "Prediction of Transonic Wing Buffet Pressure Based on Deep Learning Approaches", *70th Deutscher Luft- und Raumfahrtkongress* (DLRK), Paper 0033, Bremen, Germany, 2021.

- **R. Zahn** and C. Breitsamter, "Application of a Deep Learning Approach for Modeling Transonic Buffet and Aileron Buzz", *8th International Conference on Vibration Engineering (ICVE)*, Shanghai, China, 2021.

- **R. Zahn** and C. Breitsamter, "Convolutional Neural Network-Based Prediction of 2D-Buffet Pressure Distributions", *20th STAB-Workshop*, Göttingen, Germany, 2021.

- **R. Zahn**, A. Weiner and C. Breitsamter, "Prediction of Wing Buffet Pressure Loads Based on a Hybrid Deep Learning Model", *33rd Congress of the International Council of the Aeronautical Science (ICAS)*, Paper ICAS2022-0043, Stockholm, Sweden, 2022 (Paper and presentation awarded with **McCarthy Student Award**).

- **R. Zahn**, M. Zieher and C. Breitsamter, "Transoinc Wing Buffet Load Prediction at Structural Vibration Conditions", *72th Deutscher Luft- und Raumfahrtkongress* (DLRK), Stuttgart, Germany, 2023.