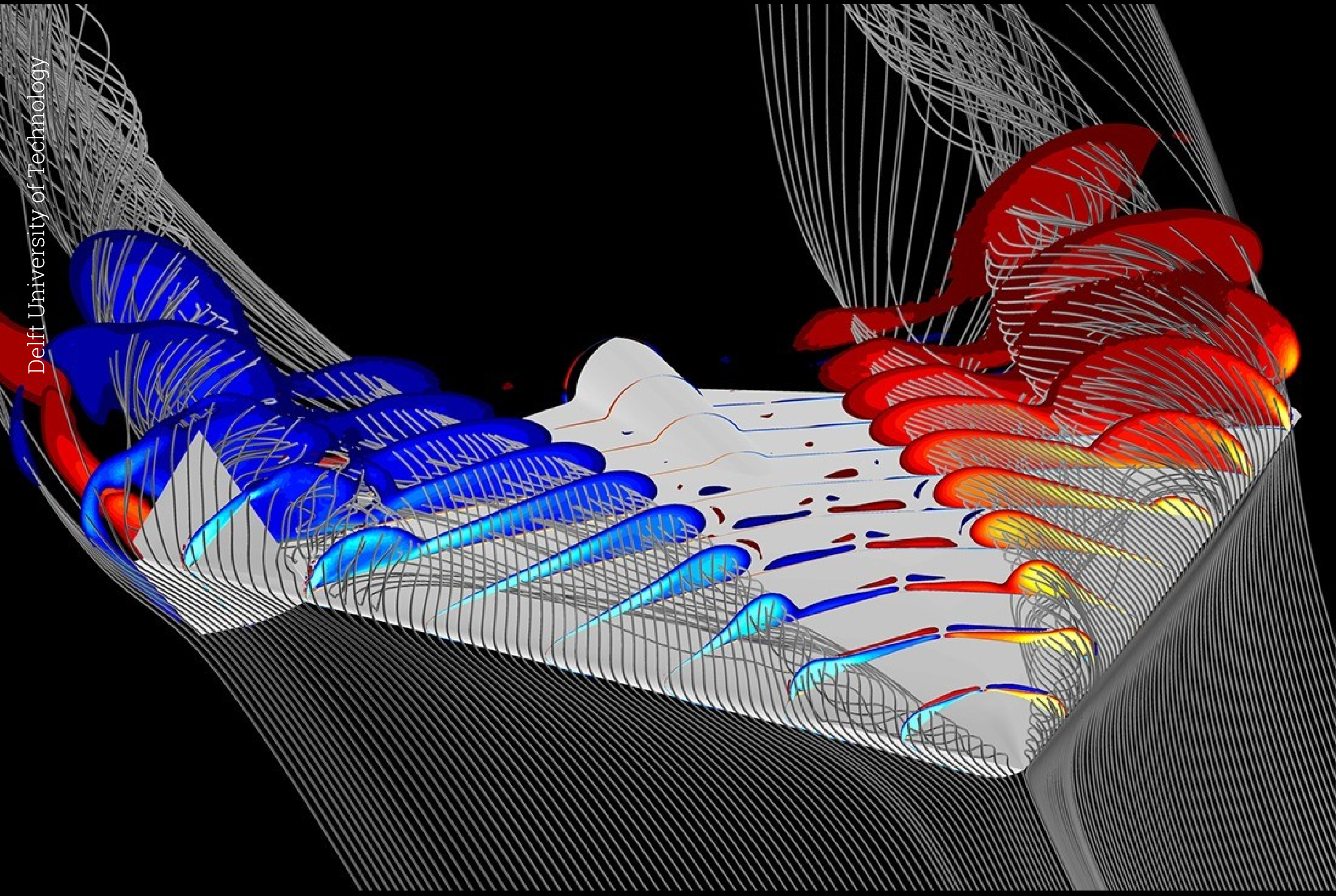
Machine Learning Based Reduced-Order Modeling for the Prediction of Pressure Distribution in the Transonic Flow Regime

MSc Thesis Aerospace Engineering

Anastasios Panagiotopoulos

Delft University of Technology



Machine Learning Based Reduced-Order Modeling for the Prediction of Pressure Distribution in the Transonic Flow Regime

by

Anastasios Panagiotopoulos

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday September 25th, 2024 at 14:00.

Student Number:	5350522	
Supervisors:	Dr. S. J. Hulshoff	Delft University of Technology (TU Delft)
	Dr. ir. M. P. C. van Rooij	Royal Netherlands Aerospace Centre (NLR)
	ir. P. Blom	Royal Netherlands Aerospace Centre (NLR)
Thesis commitee:	Dr.ir. A.H. (Alexander) van Zuijlen	Aerodynamics
	Dr.ir. Coen de Visser	Aircraft system Identification and Control
	Dr. S. J. Hulshoff	Aerodynamics
	Dr. ir. M. P. C. van Rooij,	Royal Netherlands Aerospace Centre (NLR)
	ir. P. Blom,	Royal Netherlands Aerospace Centre (NLR)
Project Duration:	Nov, 2023 - Sep, 2024	
Faculty:	Faculty of Aerospace Engineering, Delft	

Cover Copyright: AER, TU Munich

Preface

I completed my MSc Thesis in collaboration with the Royal Netherlands Aerospace Centre (NLR), in Amsterdam from November, 2023, to September, 2024. As the leading organization in the Netherlands aerospace industry, NLR, and the project's topic perfectly aligned with my academic interests. I want to express my special gratitude to my supervisors, Steven Hulshoff, Michel van Rooij, and Peter Blom. Their guidance and support were invaluable during the MSc Thesis.

Anastasios Panagiotopoulos
Rotterdam, September 2024

Summary

Maneuvering or Stability and Control (S&C) characteristics of an aircraft constitute the most challenging and expensive phase of its design process. The S&C design phase extends into the development process, sometimes leading to unexpected aerodynamic issues. Thus, identifying aerodynamic issues and their consequences, e.g. transonic buffet, in the early stages of development, is crucial. Recent advancements in Computational Fluid Dynamics (CFD) have expanded its utilization. However, despite improvements in CFD, the comprehensive representation of dynamic effects for all possible maneuvers is still unfeasible due to the high computational expense.

Reduced Order Models (ROMs) have been combined with CFD data to predict an aircraft's dynamics in all possible maneuvers. ROMs enable the efficient utilization of high-fidelity CFD data, providing valuable insights into flight dynamic effects [75]. This thesis project took place at the Netherlands Aerospace Center (NLR). The NLR in cooperation with TUDelft, has developed a ROM method for predicting unsteady aerodynamic loads of air vehicles. The current ROM approach combines the Proper Orthogonal Decomposition (POD) of pressure distribution with a Long Short-Term Memory (LSTM) type Neural Network (NN). So far, the POD-LSTM ROM method predicts the pressure distribution well in the incompressible flow regime [10, 12]. However, the increased number of spatial POD modes required to accurately represent the shock discontinuities in a pressure distribution poses challenges to POD-LSTM ROM. This leads to high computational costs, rendering the application of the POD-LSTM ROM in transonic flows impractical. Therefore, this thesis aims to set the foundation for expanding the POD-LSTM ROM for predicting the pressure distribution over sections of the DLR-F22 model in transonic conditions.

This research introduced a novel approach to address the increased number of spatial POD modes needed to approximate shock discontinuities in transonic flows. The enriched Proper Orthogonal Decomposition (ePOD) method introduces an enrichment basis into the standard truncated POD basis. The enrichment basis explicitly accounts for pressure discontinuities caused by shock waves, allowing the standard basis to focus on representing the remaining pressure distribution. The results confirm that the ePOD reduces the DoF required to approximate pressure distribution in transonic flows.

An LSTM neural network was utilized to forecast the time-dependent coefficients and parameters of the enriched reduced-order basis in unseen flow conditions. The results also showed that the ePOD reduced the complexity of the time-variant parameters of the reduced-order basis compared to the standard POD with the same number of degrees of freedom (DoF), facilitating more efficient training of the neural network.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
AE	Auto-Encoder/Decoder
ANN	Artificial Neural Network
AVT	Applied Vehicle Technology
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
CPOD	Clustered Proper Orthogonal Decomposition
DD	Domain Decomposition
DLR	German Aerospace Center
DL	Deep Learning
DNN	Deep Neural Network
DNS	Direct Numerical Simulations
ePOD	Enriched Proper Orthogonal Decomposition
FFT	Fast Fourier Transform
FOM	Full-Order Model
GRU	Gated Recurrent Units
LSTM	Long Short-Term Memory
ML	Machine Learning
MSE	Mean Squared Error
NATO	North Atlantic Treaty Organization
NLR	Royal Netherlands Aerospace Centre
NIROM	Non-Intrusive Reduced Order Model
NN	Neural Network
POD	Proper Orthogonal Decomposition
RANS	Reynolds-Averaged Navier-Stokes
RNN	Recurrent Neural Network
RMS	Root Mean Square
RPF	Relative Peak Factor
ROM	Reduced-Order Model
S&C	Stability and Control
STO	Science and Technology Organization
UCAV	Unmanned Combat Aerial Vehicle

1 Introduction and Objective

1.1. Background and Project Motivation

Stability and Control (S&C) characteristics are fundamental for the design and analysis of aircraft performance [75]. The S&C design phase is often related to unexpected aerodynamic issues, such as buffet in transonic maneuvering conditions, which often are not recognized until very late in the design process, sometimes even during flight tests, when corrections are costly or not possible [13]. Therefore, it is critical to identify those undesirable phenomena as early as possible in the design process of an aircraft [40]. These characteristics are determined by the aerodynamic loads acting on an aircraft, which often are evaluated using Computational Fluid Dynamics (CFD) methods. Although running steady-state Reynolds-averaged Navier-Stokes (RANS) simulations are feasible for the whole flight envelope, they are not applicable to regions where flow separation occurs, nor can they capture transient effects, such as those occurring during maneuvers, which are critical for highly agile aircraft. Therefore, determining aerodynamic performance for full-scaled aircraft configurations requires solving the time-accurate unsteady RANS (uRANS) equations, Detached Eddy Simulations (DES), or Hybrid RANS approaches. However, these methods are computationally expensive and demand substantial energy resources. For instance, performing S&C analysis over an aircraft flight envelope requires aerodynamic data for tens of thousands of different states to encompass all angles of attack, sideslip angles, aircraft speeds, control surface deflections, etc. [24]. Consequently, performing multiple CFD calculations to predict dynamics for all possible states is unfeasible. As a result, predicting precisely the aircraft's aerodynamic performance over a wide range of operating conditions becomes challenging. This often leads to discrepancies between the anticipated and actual aircraft performance. Data-driven Reduced-Order Models (ROMs) can efficiently utilize the expensive high-fidelity CFD data to produce high-quality regression estimates for points between states determined by CFD, thus offering valuable insights into an aircraft's flight envelope. In many applications, such as aeroelastic modeling or fatigue prediction, utilization of ROMs can reduce computational costs while providing highly accurate predictions for aircraft's performance and S&C characteristics.

Over the years, various ROM approaches have been explored for predicting aerodynamic loads in steady and unsteady flow cases. These models can be broadly categorized into physics-based and data-driven ROMs [22]. The latter category is very often applied in unsteady aerodynamics and can be roughly subdivided into the next categories: models using indicial response functions [27, 63, 78, 21], models utilizing Volterra theory [26, 3], single or multi-fidelity surrogate models [42, 25, 41], and last but not least, ROMs with integrated Neural Networks (NNs) and deep learning [31, 60, 80, 20, 19, 55]. In the age of data, ROMs based on Machine learning (ML) are becoming increasingly relevant.

In task group AVT-351 on Enhanced Computational Performance, Stability, and Control Prediction for NATO Military Vehicles, researchers from different universities, research institutes, and industries in Europe and North America work together on developing and evaluating several approaches to constructing surrogate ROMs. The Netherlands Aerospace Centre (NLR), as a member of NATO/STO research task group AVT-351, is investigating the usage of ROMs based on mode decomposition and Machine Learning that can accurately and quickly predict force and moment coefficients as well as the surface pressure distribution of an aircraft. The current approach uses a Long Short-Term Memory

(LSTM) type Neural Network in combination with the Proper Orthogonal Decomposition (POD) of the surface pressure distribution. Specifically, recent investigations have included different regression techniques, clustering, and domain decomposition to improve the performance of the POD-LSTM surrogate model [10, 12]. The combination of modal analysis and Recurrent Neural Networks (RNNs) has shown promising performance in predicting unsteady aerodynamic forces in the incompressible flow regime. Recent results have indicated that, compared with subsonic flow, transonic flow requires more spatial POD modes for accurate flow field approximation [46], rendering applying the current approach in the compressible regime challenging. The previous studies conducted by Bourier [10] and Catalani [12] set the framework for the project direction. The project aims to investigate the current ROM's applicability in the transonic flow regime by enhancing the construction of the reduced-order basis.

1.2. Problem Description

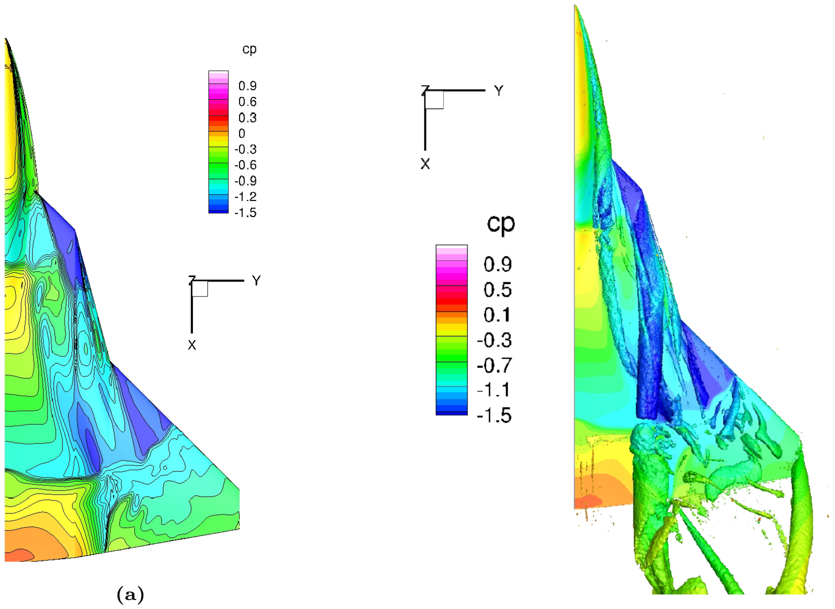
ROMs are employed to predict integral force and moment coefficients, along with their derivatives. Although accuracy requirements for load prediction are relaxed [73], many applications, including aeroelasticity and fatigue prediction, require detailed pressure distribution information [75]. The most critical concerning aeroelastic phenomena often occurs in transonic flow conditions [7]. In the transonic range, various flow phenomena can initiate and produce severe aeroelastic issues such as flutter, limit cycle oscillation, or buffet [33]. Consequently, the accurate development of control laws requires numerous test points, especially in transonic flight conditions featuring complex flow phenomena. For example, approximately 500,000 test points are necessary for a compressed CFD-based development program of a tactical fighter aircraft design [73]. Thus, the problem dimensions for one state practically equal the number of grid points on the surface of the airfoil/wing, which typically ranges from 10^4 to 10^5. When considering all possible states, the problem's dimensions increase dramatically.

Previous studies conducted by Bourier [10] and Catalani [12] at the Royal NLR have focused on developing ROMs for predicting unsteady pressure fields over the Unmanned Combat Air Vehicle (UCAV) configuration at subsonic speed and moving in the vertical plane, i.e. side slip and roll were neglected. In this ROM, the POD was combined with LSTM NNs to predict surface pressure distributions. An POD-LSTM ROM efficiently derives the reduced coordinates by using the truncated basis derived by the POD with the LTSM NN predictions of the time coefficients of the reduced-order basis in unseen flow conditions. This approach reduces the computational burden associated with training and prediction in complex neural network architectures, such as Autoencoders and Decoders. The outcome of the studies previously conducted at NLR [10, 12] has demonstrated the potential of the POD-LSTM method.

The previous POD-LSTM ROM effectively predicted integral loads' coefficients and pressure fields in the incompressible flow regime ($M_\infty \leq 0.2$). However, as the Mach number increases, flow phenomena become more complex, and as a consequence, the projection error of a modal analysis rises. The projection error, defined as the difference between the reduced-order basis and the Full-Order Model (FOM), is the critical indicator of reduced-order basis efficiency. In the case of transonic flows, more spatial POD modes are needed to encapsulate compressibility effects like shock waves and shock-induced separation. Thus, the dimension of the reduced coordinates required for the accurate approximation of the FOM increases.

This research aims to create a surrogate ROM capable of predicting pressure distribution over a specified aircraft configuration in the transonic flow regime. Similarly to previous studies, the modelling is confined to maneuvers in a vertical plane, accommodating variations in angle of attack and pitch rate. The resulting ROM enables fast predictions of aerodynamic forces across a wide range of conditions and can be integrated with structural analysis tools to investigate the aircraft's aeroelastic behavior.

The goal of this thesis is to develop a ROM capable of predicting the complex pressure distributions of the DLR-F22 model in transonic flows. The DLR-F22 model is a generic research wind tunnel model featuring a triple delta wing fighter-type aircraft configuration. The geometry of the DLR-F22 model is presented in detail in the relevant Section 5.3. Furthermore, Figure 1.1 highlights the complexity of flow phenomena over the DLR-F22 model.



(a)

(b)

Figure 1.1: Upper surface pressure distribution (a) and vortex structure visualized with the Q-criterion (b), for Kestrel with the SARC model, and $M = 0.85$, $\alpha = 20°$ [82].

Figure 1.1a illustrates the pressure distribution over the upper surface of DLR-F22 model, while Figure 1.1b visualizes vortex interactions utilizing the Q-criterion. These images reveal the various flow phenomena, including shocks, vortical structures, and vortex interactions. The primary vortical systems identified include the forebody vortex, strake vortex, wing vortex, and wingtip vortex. Furthermore, one can observe interactions between the strake and forebody vortices and among the vortical structures at the leading edge of the main wing. Moreover, the primary shock wave systems can be identified based on Figure 1.1. The first shock wave emerges shortly downstream of the intersection of the forebody and the strake, followed by a second, stronger shock at the middle of the main wing.

The United States Air Force Academy (USAFA) produced the data utilized for ROM development by solving the unsteady, three-dimensional, compressible RANS equations on hybrid unstructured grids [58], for the DLR-F22 model under two motion types: Pseudorandom Binary Sequence (PRBS) signal and Schroeder maneuver. The output data of the simulation was the high-fidelity pressure distribution at six spanwise sections on the upper surface of the DLR-F22 model. Therefore, given the available dataset, the proposed surrogate ROM was developed to predict the pressure distribution at chordwise sections of the upper surface of the DLR-F22 model. Figure 1.2 illustrates the extraction process of the high-fidelity pressure distribution data for Section 1 of the DLR-F22 model under a PRBS signal, as performed by USAFA.

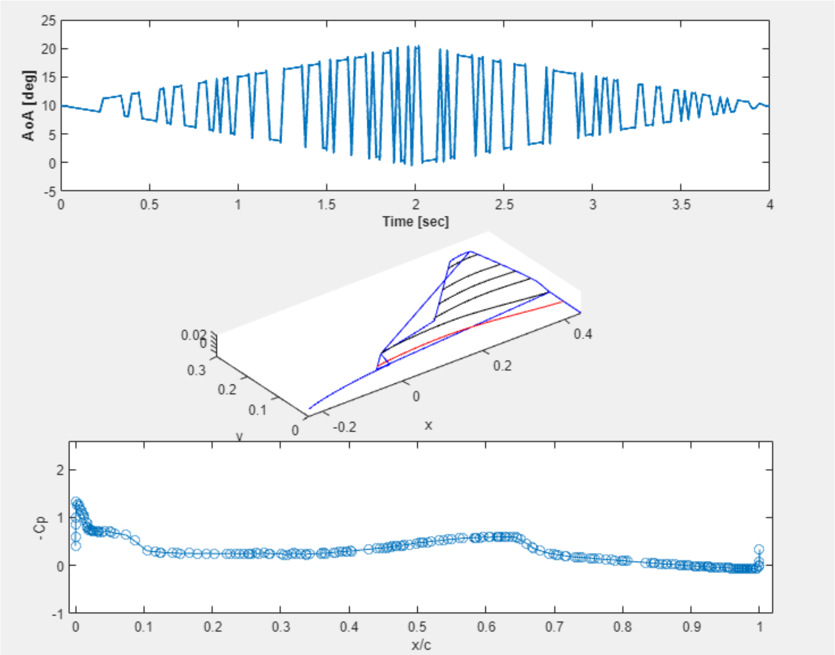


Figure 1.2: Pressure distribution extraction from Section 1 of DLR-F22, PRBS signal [24].

This research considers methods for enhancing the reduced-order basis of the POD-LSTM ROM to address shock discontinuities present in pressure distributions of transonic flows. A new approach was developed to address the increased number of spatial POD modes required to represent pressure discontinuities in transonic flows. This method is referred to as enriched Proper Orthogonal Decomposition (ePOD).

1.3. Research Objective and Questions

1.3.1. Research Objective

The objective of the research can be summarized in the following paragraph:

The research objective is to construct a Machine Learning based Reduced-Order Model (ML-ROM) based on the ePOD method capable of efficiently predicting the pressure distribution at the chordwise sections of the DLR-F22 model, with a specific emphasis on the transonic flow regime. Furthermore, we construct and train a Long Short-Term Memory (LSTM) neural network for predicting the temporal evolution of the enriched reduced-order basis.

1.3.2. Research Questions

To achieve the above objective, the following research questions were addressed:

1. How does the ePOD enriched reduced-order basis approximate the pressure distribution in transonic flows compared to the POD basis with the same number of degrees of freedom?
   (a) Does the ePOD result in a more accurate reduced-order basis than the POD?
   (b) Can the enriched reduced-order basis efficiently approximating the shock discontinuities in transonic flow pressure distributions?
2. How should the enrichment basis be integrated into the reduced-order basis for the DLR-F22 model to accurately represent flow discontinuities?
   (a) Should the enrichment mode be local or global in space?
3. How does the proposed ePOD-LSTM ROM perform when predicting the pressure distribution on a section of the DLR-F22 model compared to an POD-LSTM?

(a) Does the ePOD-LSTM ROM predict discontinuities in the pressure distribution over sections of the DLR-F22 model more accurately than POD-LSTM?
(b) How does the enriched basis of ePOD-LSTM ROM affect the training of LSTM NN compared to POD-LSTM?

1.4. Report Outline

The outline of this report is as follows. In Chapter 1, the problem that the thesis project aims to solve is introduced. Chapter 2 provides an overview of various data-driven Reduced-Order Models (ROMs) developed over the years, with a focus on the compressible flow regime and the current ROM used by NLR. Additionally, this section introduces the Proper Orthogonal Decomposition (POD) method and the Long Short-Term Memory (LSTM) neural network. The test cases used to develop, test, and validate the proposed approach, as well as the architecture of the ePOD-LSTM Reduced-Order Model, are presented in Chapter 3. In Chapter 4 , the proposed enriched Proper Orthogonal Decomposition (ePOD) method is described and evaluated. In Chapter 5, the LSTM network trained to predict the temporal dynamics of the enriched reduced basis in unseen conditions is discussed, along with the limitations and challenges of the regression approach. Furthermore, the results of the ePOD-LSTM ROM are compared with those of the POD-LSTM ROM. Finally, the conclusions of the report are summarized in Chapter 6, along with recommendations for future work.

2

Literature Review

2.1. Reduced-Order Models

Reduce Order Models (ROMs) are employed to reduce the number of degrees of freedom (DoF) in high-dimensional CFD datasets, allowing efficient analysis of a system's dynamics [44]. In other words, ROMs reduce the system's dimensions to a minimum, which still allowing for the accurate representation of flow dynamics. Consequently, this facilitates a rapid approximation of an aircraft's performance and S&C. The ROMs objective involves identifying a latent low-dimensional space that sufficiently encapsulates Full-Order Model's (FOM) dynamics [62]. Even in chaotic turbulent flows, a certain degree of order persists [8], so reduced-order model that approximate the FOM can be derived. This particular part of a ROM is often called the 'offline stage.' Conversely, the 'online stage' of a ROM involves computing flow dynamics for unknown conditions, predicting the system's behavior in unforeseen states. ROMs fall into two primary categories: 'intrusive' or physics-based models and 'non-intrusive' or data-driven models [49]. An 'intrusive' ROM employs the physical representation of the system's dynamics, usually expressing the governing equations using fewer DOF. In contrast, a 'non-intrusive' ROM exclusively uses high-fidelity data (e.g. CFD results) for reproducing the system's dynamics, disregarding the governing equations of the system [2]. Hence, these ROMs are referred to as purely data-driven methods.

2.1.1. Data-Driven ROM's based on Neural Networks

Machine learning (ML) stands out as a promising technology, particularly in the development of reduced-order models (ROMs) in the field of fluid dynamics. Typically, ROMs are utilized to predict integral aerodynamic forces, momentum coefficients, and their derivatives [75]. However, when aeroelasticity becomes relevant or fatigue analysis is required, a detailed pressure distribution is necessary.

The degrees of freedom of the pressure distribution equals the number of grid points on the surface of the airfoil/wing, dramatically increasing the training cost of Neural Networks (NNs). To address the issue of the high dimensionality of the problem, one can use Auto-Encoders/Decoders (AE) NN [29, 35, 77] or apply PODs combined with NNs for approximating the pressure distribution [10, 12, 84].

Auto-Encoder/Decoder

Integrating machine learning into ROMs involves learning and optimizing a reduced coordinate system to represent the system's dynamics. Deep Learning (DL) can be used to reduce the dimensions of the problem, similar to POD. In particular, a Neural Network (NN) Auto-Encoder/Decoder (AE) can be employed for establishing reduced coordinates. The AE features an input and output matching the dimensions of the high-dimensional fluid state, with a bottleneck that compresses it into a low-dimensional latent space. The part of the NN that decreases the coordinates of the high-fidelity data is called the 'Encoder'. Conversely, the part of the NN that maps it back to an approximation high-dimensional state is the 'Decoder' of the AE.

If the encoder and decoder each consist of only one layer, with all nodes employing identity activation functions, the optimal solution for this neural network (NN) will be closely related to POD [4]. However, this simplistic linear AE can be extended to a deep nonlinear AE with multiple layers for encoding and decoding, incorporating nonlinear activation functions. This transformation allows the deep AE to acquire nonlinear manifold coordinates, potentially enhancing compression within the latent layer. An example of this generalized AE is illustrated in Figure 2.1:

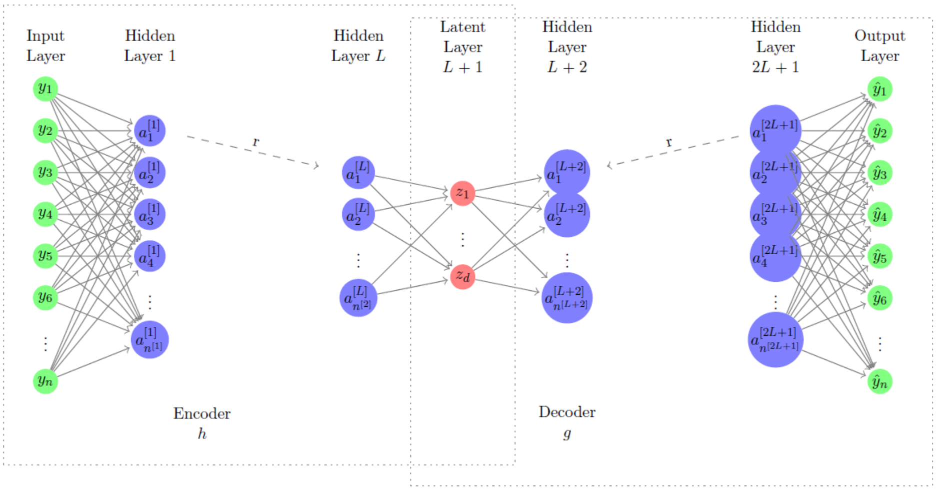


Figure 2.1: A generalized Auto-Encoder/Decoder neural network [35].

Furthermore, Lee and Carlberg [45] recently demonstrated that deep convolutional AE can significantly enhance the performance of classical ROM techniques, such as POD. This concept involves projecting dynamical systems onto nonlinear manifolds through minimum-residual formulations at both time-continuous and time-discrete levels. The first results in manifold Galerkin projection, while the latter corresponds to manifold least-squares Petrov–Galerkin (LSPG) projection. In addition, they suggest a practical method for computing the nonlinear manifold, utilizing a specific convolutional AE for dynamical systems. When the manifold Galerkin and manifold LSPG ROMs use this particular decoder, they are referred to as Deep Galerkin and Deep LSPG ROMs. It is essential to note that the main drawback of this approach is the costly training process. Training a deep convolutional AE is considerably more computationally expensive than POD-based approaches, as described by Lee and Carlberg [45]. Moreover, it introduces significantly more hyperparameters, specifically related to the AE, compared to classical methods.

Zahn [84] utilized a hybrid deep learning ROM to forecast wing buffet pressure distributions on a civil aircraft configuration. This model combines a Convolutional Variational Neural Network AE (CNN-VAR-AE) with a Long Short-Term Memory (LSTM) neural network. The CNN-VAR-AE reduces the DoF of the high-dimensional flow field data, deriving the latent space. The LSTM neural network is employed to predict the temporal evolution of the latent space of pressure distributions. Figure 2.2 shows the schematic representation of this hybrid ROM.

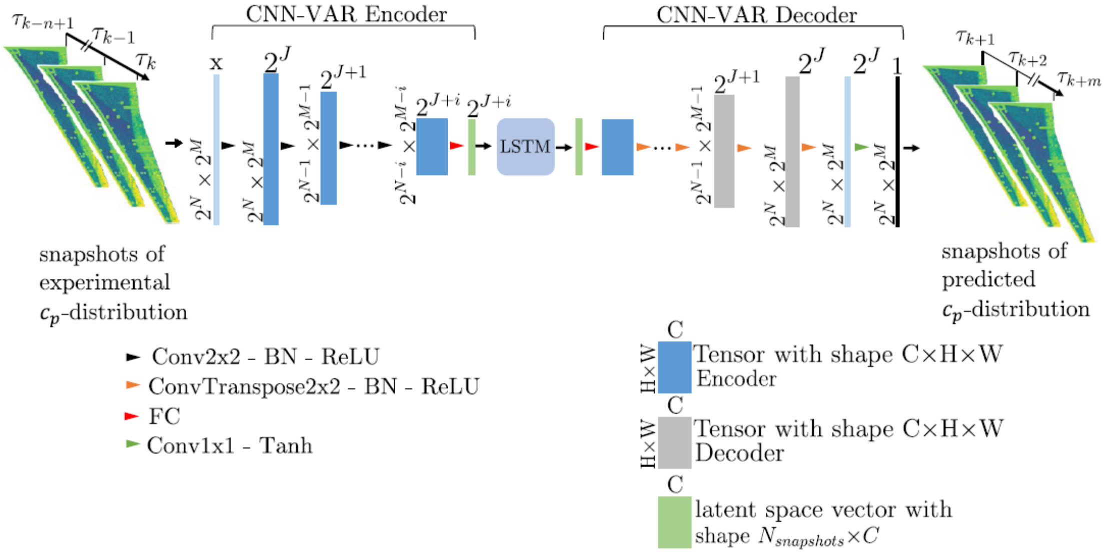


Figure 2.2: Hybrid DL-ROM as proposed by Zahn [84].

The effective performance capabilities of the proposed ROM method in predicting non-linear flow fields were validated by the provided experimental data, as Zahn noted [84]. Their objective was to utilize this ROM method for reconstructing buffet pressure distributions under various flow conditions, including variations in Mach and Reynolds numbers. The resulting ROM utilizes a CNN-VAR-AE for deriving the reduced coordinates. Prediction quality was improved by adjusting the pre-processing routine and incorporating techniques such as Proper Orthogonal Decomposition [84].

POD combined with NNs

A promising approach involves combining Proper Orthogonal Decomposition (POD) with Neural Networks (NNs) to predict surface pressure distribution [10, 12, 20, 55]. When pressure distribution is required, the dimensionality of the problem increases significantly. Training an end-to-end ML-based ROM demands a large amount of high-fidelity training data to accurately forecast pressure distribution. Consequently, training an Auto-Encoder/Decoder becomes computationally inefficient.

Instead of employing an end-to-end ML model, the proposed method splits the encoding and decoding components. Firstly, the POD, or a similar dimensionality reduction technique, calculates a reduced-order basis approximating the FOM. Secondly, a neural network learns the temporal dynamics of the reduced-order basis. Recurrent Neural Networks (RNNs) are frequently employed when dealing with time-dependent problems, as described in the relevant chapter section 2.3.

The architecture of the POD-NN method was well described by Mohan and Gaitonde [55], who developed a ROM using POD combined with a Long Short-Term Memory (LSTM) neural network. The POD-LSTM ROM proposed by Mohan and Gaitonde [55] follows a similar structure to the ROM approach developed in NLR. The primary steps of the POD-LSTM ROM are outlined below:

- Selection of the Training Datasets: The initial step in developing the POD-LSTM method is the selection and organization of data. As outlined in [80], this process involves extracting two-dimensional planes from the three-dimensional high-fidelity data to serve as training datasets. Furthermore, select a subset of these planes as the test dataset. The performance of the POD-LSTM ROM is evaluated by predicting the test dataset's behavior after learning the dynamics from the training datasets.
- Extraction of POD spatial modes and time coefficients: Extract the dominant spatial POD modes and their corresponding time coefficients for the training and test datasets. Detailed information about spatial modes and time coefficients is available in the subsequent paragraphs. The time coefficients of the test dataset will validate the LSTM neural network prediction.
- Training the LSTM neural network: Train the LSTM/BiLSTM neural network to predict the time coefficients for the truncated spatial POD modes from the previous step. During validation,

use a short history of the test dataset time coefficients as input to predict the time coefficients of the next few time instants. Compare the LSTM network's predictions with the actual time coefficients from the test dataset. Repeat this process for all chosen spatial POD modes.

- Reconstruction of predicted flow field: Finally, use spatial POD modes and the predicted temporal coefficients to reconstruct the predicted flow field.

Figure 2.3 illustrates the main stages of the POD-LSTM ROM proposed by Mohan and Gaitonde [55].

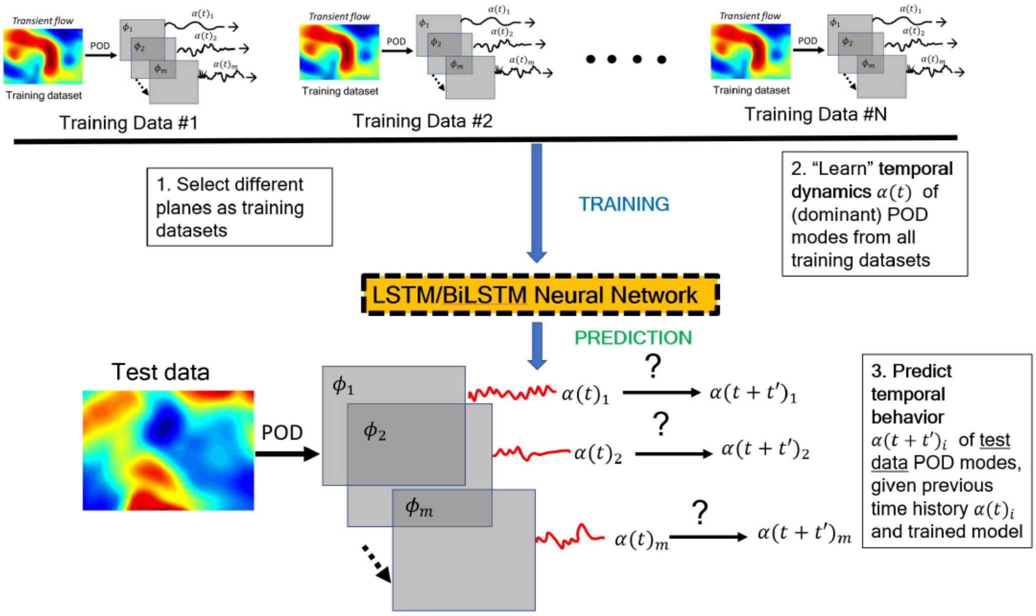

Transient flow
POD
Training dataset
ϕ_1
ϕ_2
ϕ_m
$\alpha(t)_1$
$\alpha(t)_2$
$\alpha(t)_m$
Training Data #1
Training Data #2
Training Data #N
1. Select different planes as training datasets
TRAINING
2. "Learn" temporal dynamics $\alpha(t)$ of (dominant) POD modes from all training datasets
LSTM/BiLSTM Neural Network
PREDICTION
Test data
?
$\alpha(t+t')_1$
$\alpha(t+t')_2$
$\alpha(t+t')_m$
3. Predict temporal behavior $\alpha(t+t')_i$ of test data POD modes, given previous time history $\alpha(t)_i$ and trained model

Figure 2.3: The ML-based ROM, which combines POD with LTSM NN [55].

NLR uses a ROM approach with a similar structure to the POD-LSTM ROM proposed by Mohan and Gaitonde [55]. This POD-LSTM method was first introduced by Bourier [10]. Bourier [10] developed a POD-LSTM ROM that employs a reduced-order basis constructed from high-fidelity CFD samples and a LSTM neural network to predict the temporal dynamics of this basis in unseen conditions. The POD-LSTM ROM was employed to predict the surface pressure coefficient distribution of the MULDICON UCAV configuration. The performance of the POD-LSTM model was evaluated in terms of accuracy, computational training (training cost), and prediction time (online cost). The structure of the POD-LSTM ROM proposed by Bourier [10] is illustrated in the Figure 2.4:

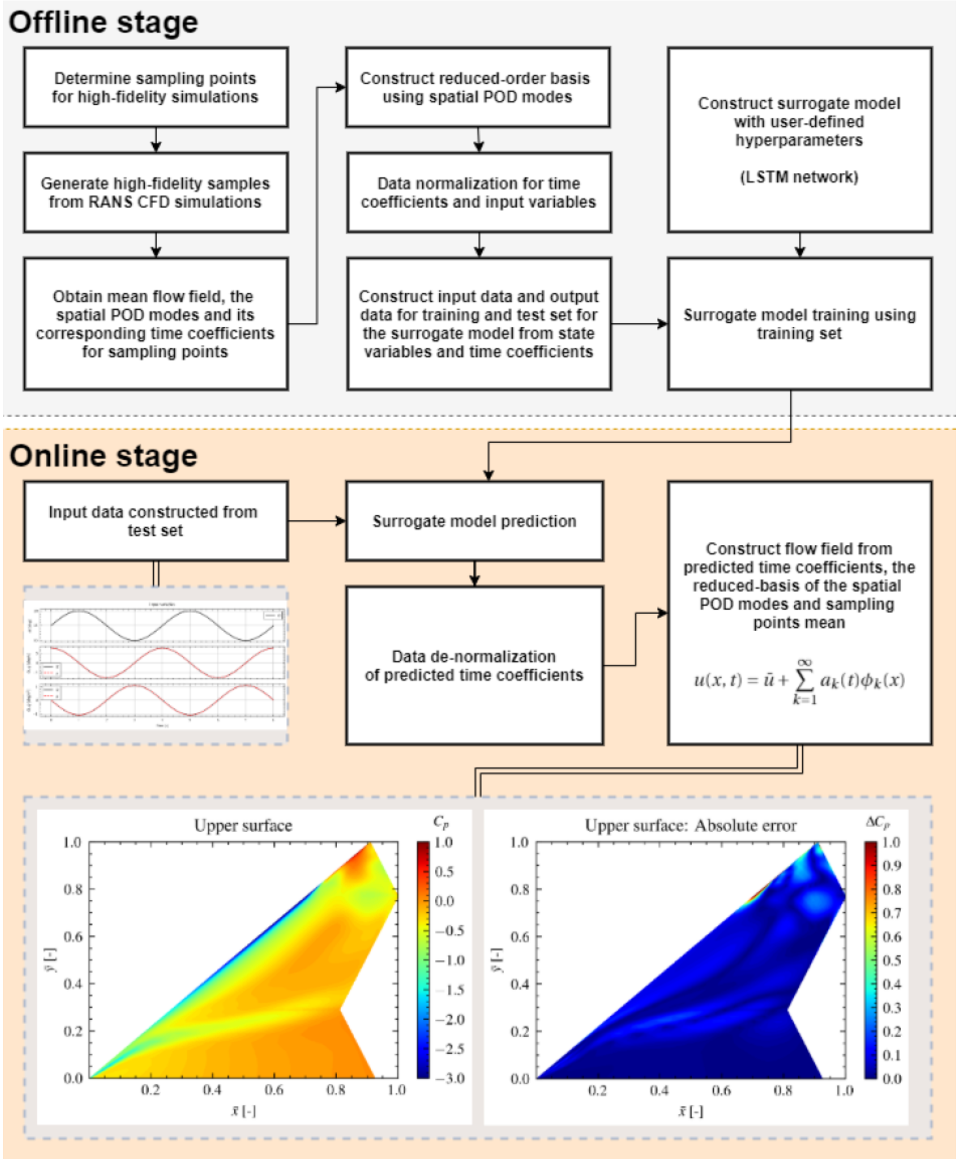

Offline stage
Determine sampling points for high-fidelity simulations
Generate high-fidelity samples from RANS CFD simulations
Obtain mean flow field, the spatial POD modes and its corresponding time coefficients for sampling points
Construct reduced-order basis using spatial POD modes
Data normalization for time coefficients and input variables
Construct input data and output data for training and test set for the surrogate model from state variables and time coefficients
Construct surrogate model with user-defined hyperparameters
(LSTM network)
Surrogate model training using training set
Online stage
Input data constructed from test set
Surrogate model prediction
Data de-normalization of predicted time coefficients
Construct flow field from predicted time coefficients, the reduced-basis of the spatial POD modes and sampling points mean
$u(x,t) = \bar{u} + \sum_{k=1}^{\infty} a_k(t)\phi_k(x)$
Upper surface
C_p
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
$\bar{y}$ [-]
$\bar{x}$ [-]
0.0
0.2
0.4
0.6
0.8
1.0
Upper surface: Absolute error
ΔC_p
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

Figure 2.4: The proposed POD-LSTM ROM by Bourier [10].

In order to assess the performance of the POD-LSTM two primary measurement errors were introduced by Bourier [10]. The first was the time-averaged projection error of the surface pressure coefficient for the training samples, evaluated in terms of Mean Squared Error (MSE) in comparison to the Full-Order Model. The second was the time-averaged neural network error of the surface pressure distribution for the training samples, assessed in terms of MSE in comparison to the actual reduced-order basis, refer to [10] for details. In the POD-LSTM ROM, the projection and neural network errors was evaluated separately. This distinction allows for the easy identification of error sources within the model and facilitates conclusions about the performance of both the offline and online phases. The main conclusions drawn from Bourier [10] regarding the offline and online stages are as follows.

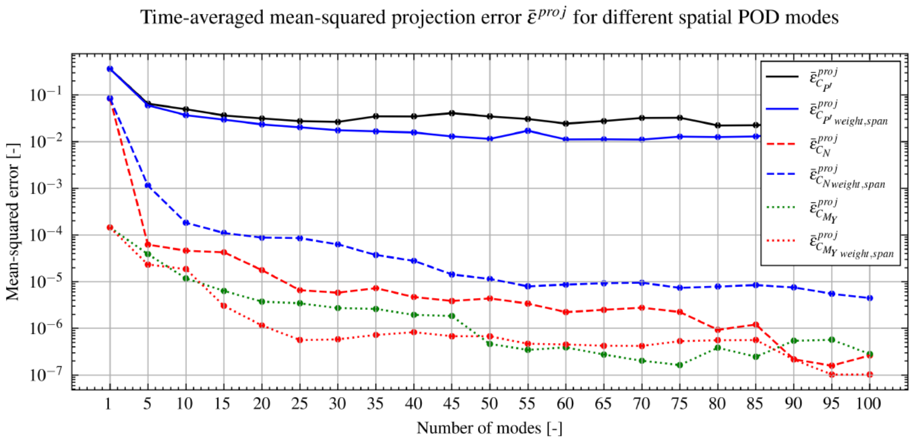


Figure 2.5: Time-averaged projection error, of pressure distribution, normal force, and pitching moment coefficients vs the number of spatial POD modes [10].

The time-averaged projection error of the pressure distribution, normal force, and pitching moment coefficient are represented in Figure 2.5. The POD of the pressure distribution results in the highest projection error. The primary limitation of the reduced-order basis was that the high projection errors only cover a small region near the leading edge, specifically near the wingtip. This observation explains why the MSE value remained almost constant after the first 50 modes, as shown in Figure 2.5. Although these errors could significantly affect the pitching moment coefficient, the impact was minimal due to the limited affected region. Consequently, the restriction on using the reduced-order basis was negligible under the subsonic conditions for the MULDICON UCAV configuration.

Furthermore, the effect of weighted POD was also investigated. The study demonstrated that spatial POD modes with a linearly weighted relationship along the span, particularly emphasizing the tip region, resulted in lower projection errors for both the surface pressure coefficient and the pitching moment coefficient, while maintaining the same number of degrees of freedom as the standard POD. Overall, the projection error of the reduced-order basis was minimal, indicating that the limitations of the POD method were negligible within the subsonic flow regime.

The construction and training of the LSTM neural network have demonstrated that the number of LSTM units in the model is the most significant hyperparameter influencing both the computational cost and the accuracy of the ROM. The LSTM units correspond to the number of LSTM memory cells. For more details, refer to Figure 2.3.2. The relationship between the training cost and the number of units is shown in Figure 2.6.

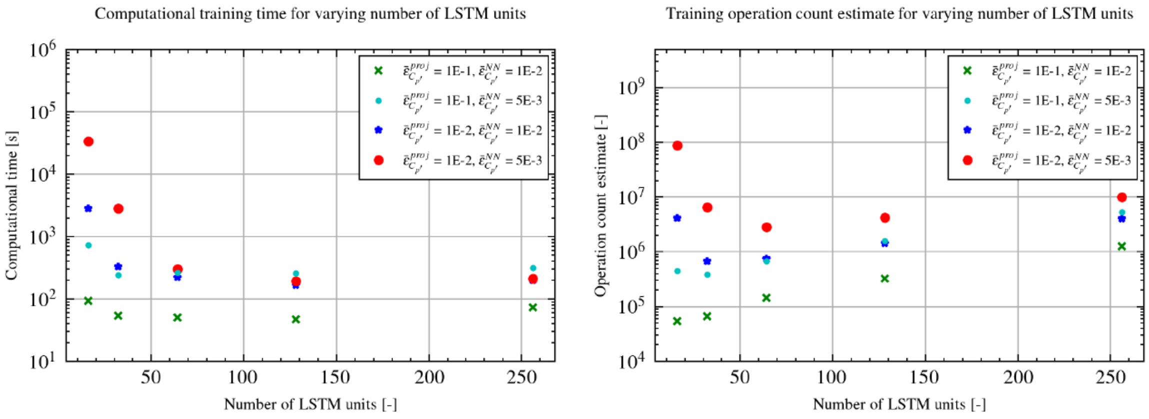


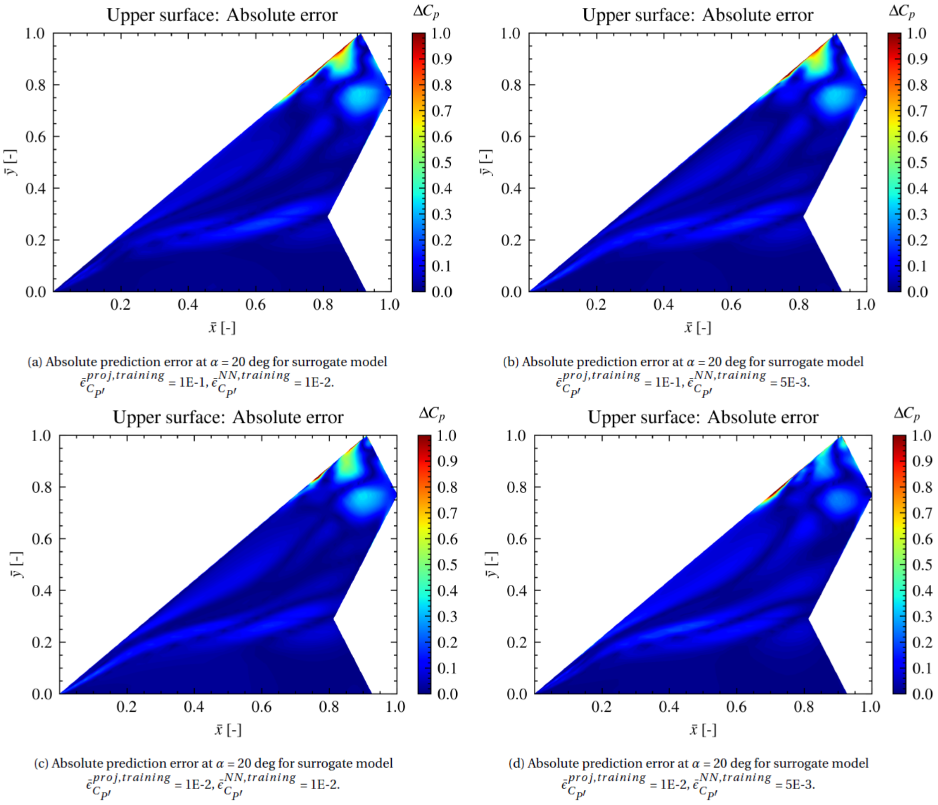
Figure 2.6: Training cost of LSTM neural network versus the number of memory cells [10].

Based on Figure 2.6, it was derived that when the model's parameters are optimized, the training cost was reduced to an average of 4 minutes to achieve sufficient accuracy. The advantage of replacing the encoder component of an Autoencoder (AE) with a POD-based reduced-order technique becomes evident when comparing computational costs. Papp's thesis [60], previously conducted at NLR, developed a ROM using deep Convolutional Neural Networks (CNNs). In this study, the primary deep encoding-decoding CNN predicted pressure distribution over the MULDICON UCAV configuration. The corresponding training cost was on the order of several hours (12h 24m for the primary Neural Network, for details refer to [60]). This discrepancy arises because the POD-based approach does not need to train a complex AE to derive the low-dimensional latent space. Furthermore, in addition to the higher training cost, complex Neural Networks, such as the deep convolutional neural networks used by Papp [60], exhibit unexpected behaviors such as overfitting. In other words, the more parameters a neural network needs to predict, the more sensitive it is to the effects of overfitting.

In [10], the POD-LSTM predictions were compared to the high-fidelity CFD results to evaluate the online stage. Bourier [10] concluded that the normal and axial force coefficients were adequately approximated by the suggested model for all angles of attack. However, the pitching moment coefficient showed inaccurate predictions due to an incorrect surface pressure approximation near the wingtip region. Therefore, the prediction of integrated forces and moments was decoupled from the pressure distribution. Specifically, the signals of forces and moments served as the input/output of the ROM, along with time coefficients, potentially in a separate ROM.

Moreover, it was derived that models with a higher projection error, i.e., reduced-order bases with fewer spatial modes, showed better accuracy in predicting the pitching moment coefficient. Although one might expect that a reduced-order basis with a lower projection error would produce more accurate predictions for the surface pressure coefficient, this was not the case for the steady-state result, as illustrated in Figure 2.7.

In models with lower projection errors (i.e., more spatial modes), the neural network needed to predict a larger number of time coefficients. As a result, the accuracy of the time coefficients for the initial spatial modes decreased compared to models with fewer POD modes. Furthermore, a system with many degrees of freedom required a more complex neural network for forecasting its temporal evolution. Consequently, it required more intense training compared to a model with fewer spatial modes. Additionally, the frequency of the time coefficients increased with the order of the POD modes, making it challenging to predict the range of frequencies when the order of spatial modes is high. Bourier's observations [10] are significant, as they highlighted a decrease in the accuracy of POD-LSTM predictions when more spatial modes were required for an accurate approximation of the pressure distribution. This limitation becomes particularly relevant in the transonic regime, where an increased number of spatial modes is necessary to approximate flow discontinuities effectively.



(a) Absolute prediction error at $\alpha = 20$ deg for surrogate model $\bar{\epsilon}_{C_{p'}}^{proj,training} = 1\text{E-}1$, $\bar{\epsilon}_{C_{p'}}^{NN,training} = 1\text{E-}2$.

(b) Absolute prediction error at $\alpha = 20$ deg for surrogate model $\bar{\epsilon}_{C_{p'}}^{proj,training} = 1\text{E-}1$, $\bar{\epsilon}_{C_{p'}}^{NN,training} = 5\text{E-}3$.

(c) Absolute prediction error at $\alpha = 20$ deg for surrogate model $\bar{\epsilon}_{C_{p'}}^{proj,training} = 1\text{E-}2$, $\bar{\epsilon}_{C_{p'}}^{NN,training} = 1\text{E-}2$.

(d) Absolute prediction error at $\alpha = 20$ deg for surrogate model $\bar{\epsilon}_{C_{p'}}^{proj,training} = 1\text{E-}2$, $\bar{\epsilon}_{C_{p'}}^{NN,training} = 5\text{E-}3$.

Figure 2.7: Absolute prediction error of the POD-LSTM ROM for upper surface pressure under different model settings, for $\alpha = 20°$ [10].

The pitching moment coefficient showed the highest prediction errors in all cases for unsteady simulations. This was attributed to inaccuracies in predicting the surface pressure coefficient near the wing tip region. Therefore, enhancing the reduced-order basis for the surface pressure coefficient requires additional training samples in this specific region.

In summary, the POD-LSTM ROM proposed by Bourier [10] showed promising results for predicting surface pressure coefficients and integral aerodynamic load coefficients. Specifically, the normal and axial force coefficients of the UCAV MULDICON configuration exhibited promise in both steady and unsteady simulations. However, a limitation of this model is its incapability to accurately predict the surface pressure coefficient near the wing tip region, resulting in high errors in the pitching moment coefficient.

Catalani [12], in his thesis, improved the previously developed POD-LSTM ROM. The investigations involved various regression techniques, including Artificial Neural Networks (ANN) and Gaussian Process Regression. The study conducted by Catalani [12] aimed to address issues raised in Bourier's investigations. The research objective was to enhance and extend the performance of the POD-LSTM ROM for predicting integral loads and pressure fields of the UCAV MULDICON aircraft design. Specifically, the focus was on Local Modal Decomposition approaches to improve the prediction of reduced dynamics. The primary limitations of Bourier's method arose from using a global POD basis, where projecting full-order dynamics into lower-dimensional linear subspace resulted in high prediction errors, particularly when considering a limited number of modes. To overcome this limitation, Catalani [12] employed three techniques: Global (scaled) POD, Domain Decomposition POD (DD-POD), and Cluster POD (CPOD). Scaling the pressure field snapshots was introduced to derive a reduced-order basis that more efficiently captured complex flow structures. This method enhanced the ROM's performance in critical regions, such as the wingtip, where large projection errors were previously observed by Papp [60] and Bourier [10]. Domain Decomposition (DD) was utilized to isolate regions containing complex flow phenomena like the wingtip vortex [50]. Cluster POD (CPOD) further refined the process by generating local reduced-order bases in the parameter space, grouping solutions with similar patterns into the same set of modes, and constructing local ROMs for each cluster [34].

The projection error of the pressure distribution was a key metric used to evaluate the offline performance of the ROM. Lower projection errors within a fixed-dimensional latent space indicated more efficient information compression. Figure 2.8 shows the time-averaged Mean Squared Error (MSE) of the projection for random instances of pitching motion.

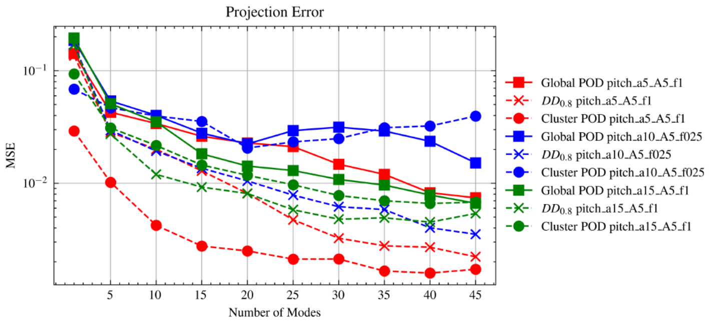


Figure 2.8: Time-averaged projection error as a function of the number of modes for harmonic pitching [12].

As observed in Figure 2.8, CPOD performed better at low-pitch angles, whereas DD-POD achieved the lowest projection error at higher pitch angles. The performance of the ROM was also influenced by the LSTM's ability to predict the reduced-order temporal dynamics. The total error served as an indicator of overall ROM performance. A limited number of spatial POD modes was considered to compare the reconstructed pressure field using the various ROMs. The total Mean Squared Error (MSE) for the three different ROMs under pitch harmonic motion is illustrated in Figure 2.9.

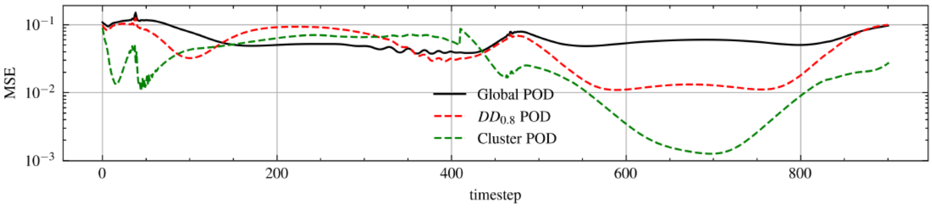


Figure 2.9: Total error as a function of the timestep during harmonic pitching [12].

The following remarks were derived based on Figure 2.9. The CPOD performed better than the DD-POD and the Global POD for most time instances, reducing the total error of the POD-LSTM ROM by one order of magnitude in the range of α between 0° and 5°. The DD-POD error appeared less than the Global POD for almost all timesteps. The offline phase of the ROMs represented the most computationally demanding part of the models' construction. This phase involved determining the reduced-order basis and training the LSTM neural network to predict the low-order temporal dynamics. Although the cost of determining the reduced-order basis was low, the computational cost of training the neural network was relatively high. An overview of the offline and online computational costs for various ROMs is given in Table 2.1.

Table 2.1: Offline and online cost of the various LSTM-based ROMs, using 5 spatial modes [12].

ROM	Offline Cost [s]	Online Cost [s]
Global POD	2111	5
Cluster POD	2344	7
DD POD	4939	6

The Global POD method utilized a single set of input-output pairs, necessitating only one training session for the neural network (NN). In contrast, local POD methods required training the NN on a more localized basis. Specifically, the DD-POD approach required training two neural networks: one for predicting the temporal evolution of the latent space in each domain. Similarly, the CPOD method required four LSTM neural networks.

In summary, the results indicated that Local ROMs enhanced the efficiency of the global latent space representation by producing sets of spatial modes with more localized information content. The proposed Cluster POD-based ROM, in particular, was developed through a systematic procedure that automatically detected parameter space partitions based on an a posteriori error indicator. The proposed ROM showed significant potential in terms of accuracy and computational efficiency. However, it is important to note that these findings were limited to the incompressible flow regime and did not address the increased number of spatial modes needed for approximating flow fields with discontinuities.

2.1.2. Discussion

Over the years, numerous approaches have been developed for constructing ROMs for predicting unsteady aerodynamics in the compressible flow regime. Intrusive methods exhibit several disadvantages and have limited applicability in full-scale problems. For example, ROMs based on the Galerkin Projection method exhibit good predictive ability and are less constrained in application. However, the computation of POD coefficients, which are crucial for predicting the flow field, requires solving a reduced system of equations derived from projecting the Navier–Stokes equations onto a reduced space. Consequently, this approach becomes computationally expensive, particularly when only the surface pressure distribution is needed. Additionally, the Unsteady Residual requires a large set of spatial POD modes, making it computationally costly. Frequency Domain methods, while capable of converging to time-accurate pressure distributions, require an increasing number of Fourier modes, resulting in higher computational time.

Non-intrusive methods such as Indicial Response Functions and Volterra Theory can accurately reproduce the complete pressure distribution around an airfoil or aircraft. However, employing these methods becomes computationally unfeasible due to the resulting dimensions of the problem, which arise from the large number of simulations required when sampling indicial responses and impulse functions for nonlinear ROMs.

Machine Learning is increasingly employed to develop ROMs in fluid dynamics. In the class of non-intrusive ML-ROMs, two main approaches exist. The first approach involves constructing an end-to-end ML-based ROM using an Autoencoder/Decoder neural network. However, the training of the AE is computationally expensive. Training a deep convolutional Autoencoder/Decoder is considerably more computationally expensive than computing the left singular vectors of a snapshot matrix. To deal with that issue, ROMs combining modal analysis with neural networks have been explored. One of the most promising approaches involves the combination of POD with an LSTM neural network. Bourier [10] and Catalani [12] laid the foundation for the current method used at NLR. The POD-LSTM ROM has demonstrated strong performance in predicting aerodynamic loads. However, there is a notable distinction between computing pressure distributions for flight dynamics, where load coefficients are the primary concern, and for applications like fatigue analysis or aeroelasticity modeling, where accurate pressure distribution is required. For instance, in the transonic regime, accurately identifying the precise location of the shock is essential for fatigue analysis.

The current POD-LSTM approach struggles to approximate flow discontinuities caused by shock waves. Catalani [12] employs various local ROMs to enhance the global POD basis. Although his approach shows promise, its applicability in compressible aerodynamics remains questionable. In the transonic flow regime, the projection error increases when using a truncated number of POD modes for pressure distribution. Consequently, more spatial modes are needed to capture the dynamics of transonic flow in the latent space. The results indicate that, unlike subsonic flow, transonic flow features significant high-frequency spatial POD modes characterized by low energy [46]. Therefore, accurately reconstructing shock waves requires a greater number of POD modes.

As a result, the proposed ML-based ROMs need to learn more temporal coefficients $a_k(t)$ of the dominant modes for the training datasets, as illustrated in Figure 2.3. Thus, the training cost of the neural network in the transonic flow regime increases. Furthermore, predicting the range of frequencies for the time coefficients becomes challenging as the order of spatial modes increases, as noted by Bourier [10]. Hence, the accuracy of POD-LSTM ROM predictions and time efficiency decreases because more spatial POD modes are required to approximate complex flow phenomena. The proposed ePOD-LSTM approach seeks to address this issue by introducing an 'enrichment' basis to the standard POD reduced-order basis. The enrichment basis explicitly accounts for the pressure discontinuities caused by a shock wave. As a result, the previously proposed POD-LSTM ROM can be applied without the computational burden of training neural networks or the reduced accuracy for an augmented number of spatial POD modes. Therefore, the objective was to establish an accurate ROM with minimal complexity. Utilizing the POD helps us avoid training costs and potentially unpredictable behaviors associated with a complex AE. Employing an enrichment basis reduces the degrees of freedom of the latent space, thereby decreasing the training data requirements and enhancing online efficiency.

The POD order reduction technique and LSTM neural network remain the fundamental components of the proposed ROM. Hence, their representation is crucial for understanding the structure of the proposed ROM.

2.2. Proper Orthogonal Decomposition

The Proper Orthogonal Decomposition (POD) is a modal decomposition method used for the dimensional reduction of high-fidelity data. By decomposing a large dataset into orthonormal modes, one can extract the minimum number of POD modes that represent all the important characteristics of the given dataset. This technique is also known as Principal Component Analysis and the Karhunen-Loève decomposition in other scientific fields. It was first introduced to the field of fluid dynamics by Lumley [51]. Later, Sirovich [72] introduced the method of snapshots, enabling efficient application in the field of CFD. The POD method produces a set of modes that efficiently represent the significant characteristics and patterns of the flow field, allowing the construction of a ROM [76, 65, 48, 38]. A general description of how the POD is applied on a fluid dynamics follows, based on the notation of [81] and [76]:

Firstly, a set of snapshots of the flow field (e.g. velocity, pressure), including the spatial and temporal values is required. For the purposes of this review, a snapshot of the velocity field $U = (u, v, w)$ is utilized. The position vector is denoted as $X = (x, y, z)$ while the time is t. The POD is typically applied to the fluctuations of the velocity, and thus the temporal mean $\bar{U}$ is subtracted from the snapshots, as follows in Equation (2.1):

$$U'(X,t) = U(X,t) - \bar{U}(X) \tag{2.1}$$

The matrix $U(X,t)$ has dimensions $n \times m$, where n represents the number of spatial discrete points, and m denotes the number of time instances. It's important to note that the matrix can exist in both 2D and 3D configurations. However, for the sake of simplicity, readers are encouraged to consider it as two-dimensional.

The primary objective of Proper Orthogonal Decomposition (POD) is to break down the flow properties into a collection of spatial modes ϕ_k and corresponding time coefficients $a_k(t)$, as illustrated in Equation (2.2).

$$U'(X,t) = \sum_{k=1}^{\infty} a_k(t)\phi_k(X) \tag{2.2}$$

The classical Proper Orthogonal Decomposition (POD) technique looks for the optimal set of basis functions that accurately represent high-fidelity flowfield data. The optimality, in this context, is defined by minimizing the error in the energy norm during the re-interpolation of the data. In other words, looking for the ϕ_k functions that can efficiently reconstruct the high-fidelity data with the least number of modes. This is accomplished by determining the eigenvectors and eigenvalues of the following problem.

$$C\phi_k = \lambda_k \phi_k \tag{2.3}$$

The eigenvalues are arranged in decreasing order as $\lambda_1 \geq \ldots \geq \lambda_n$, ensuring that the spatial modes follow the order of importance in capturing the kinetic energy of the flowfield. The C is the covariance matrix of $U'(X,t)$ and computed by the following equation:

$$C = \frac{U'^T U'}{m-1}, \ \in \mathrm{R}^{n \times n} \tag{2.4}$$

While solving the eigenvalue problem in Equation 2.3 is straightforward, challenges arise when dealing with large datasets. The problem of Equation 2.3 is solved by utilizing the eigenvalue decomposition of the matrix $C \in \mathrm{R}^{n \times n}$, where n represents the number of spatial discrete points or the number of degrees of freedom of the dataset. The eigenvalue decomposition of the matrix C becomes computationally expensive due to large grids usually employed in high-fidelity CFD simulations. Sirovich [71] introduced the *method of snapshots*, which allows the application of the POD to datasets with large spatial dimensions. Mathematically, there is no difference between the temporal variable t and the spatial variable x, and thus Equation 2.2 is symmetric in t and x. Utilizing the same snapshot matrix U, the eigenvalue decomposition is applied to the following temporal correlation matrix:

$$C_s = \frac{U'^T U'}{m-1}, \ \in \mathrm{R}^{m \times m} \tag{2.5}$$

Therefore, a decomposition in deterministic temporal modes with random spatial coefficients instead of spatial modes and time coefficients is performed. Usually, the number of snapshots m is less than the number of spatial discrete points n. Hence, the resulting eigenvalue problem has less computational cost than the direct POD, while it returns the same set of eigenvectors and values. Unfortunately, this is not the case in the pressure distribution over the sections of DLR-F22, where the size of the snapshots m is almost equal to the spatial discrete points n. These methods are closely related to the *Singular Value Decomposition* (SVD) of the snapshot matrix U [81]. In matrix form, the snapshot matrix U can be decomposed directly using the SVD as follows:

$$U = L\Sigma R^T \tag{2.6}$$

where L is an orthogonal matrix with dimension $m \times m$, Σ is an rectangular diagonal matrix with dimensions $m \times n$ and R is an $n \times n$ orthogonal matrix. The non-zero diagonal elements of Σ are typically a set r positive numbers arranged in decreasing order, i.e. $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq 0$. These are the singular values of U. SVD diagonalizes any rectangular matrix, whereas eigenvalue decomposition only works for square matrices. Therefore, both POD approaches are equivalent to the SVD of the matrix $U/\sqrt{m-1}$ the spatial modes of the direct POD are given by its right singular vectors R, the temporal modes of the snapshot POD are given by the left singular vectors L. The eigenvalues of both methods are the squares of the singular values of SVD. For more details regarding the SVD, refer to Taira et al. [76].

The optimum basis to express the high-fidelity data can be easily derived by solving the eigenvalue problem Equation (2.3) utilizing the Equation (2.4) or by performing the SVD of matrix $U/\sqrt{m-1}$. The eigenvectors of Equation (2.3) or the right singular vectors R are the spatial modes, ϕ_k, of the POD and are sorted column-wise in a matrix of size $n \times n$. The spatial POD modes are orthonormal, so the next equation is valid:

$$\iiint_X \phi_{k_1}\phi_{k_2} dX = \begin{cases} 1 & \text{for } \ k_1 = k_2 \\ 0, & \text{for } \ k_1 \neq k_2 \end{cases} \tag{2.7}$$

This property is very important as it is ensures that each temporal coefficient $a_k(t)$ corresponds to the spatial mode ϕ_k. The temporal coefficients can be simply computed as the following inner product:

$$a_k = \langle U', \phi_k \rangle \tag{2.8}$$

The matrix containing temporal coefficients has dimensions of $n \times m$.

The number of modes required to represent fluctuations in the flowfield data can be calculated using the eigenvalues of the problem. Generally, the modes are truncated based on the next equation:

$$\frac{\sum_{k=1}^{r} \lambda_k}{\sum_{k=1}^{n} \lambda_k} \simeq 1 \tag{2.9}$$

Then, the reconstruction of flow properties based on the important spatial modes is as follows:

First, use Equation (2.2), modified for r spatial modes:

$$U' = \sum_{k=1}^{r} a_k \phi_k^T \tag{2.10}$$

By substituting Equation (2.1) into Equation (2.10), we derive the reconstructed flow property:

$$U = \sum_{k=1}^{r} a_k \phi_k^T + \bar{U} \tag{2.11}$$

A comprehensive explanation of why Proper Orthogonal Decomposition (POD) requires additional modes in the case of transonic flow was provided by Li and Zhang [46]. They conducted a POD analysis of the flow field over NACA 0012 for transonic flow ($M = 0.8$). The first three POD modes of the pressure field are illustrated in Figure 2.10, and the surface pressure distribution at a random time instance is shown in Figure 2.11. The spatial POD modes showed smooth changes throughout the flowfield, except where the shock wave occurs. As illustrated in Figure 2.10, a ribbon pattern emerged at the location of the shocks, with increasing order of POD modes. As the order of the spatial mode increased, the amplitude of the modes decreased, and the frequency increased. Despite the small amplitude of these modes, they remained dynamically relevant. Given the necessity of accurately reconstructing the shock, these high-order spatial POD modes become essential. The shock was reconstructed through the linear superposition of the modes, making it challenging to offset the high-order modes at different amplitudes and frequencies. Consequently, an oscillation phenomenon occurs before and after the shock, commonly known as Gibbs' phenomenon, as visible in Figure 2.11.

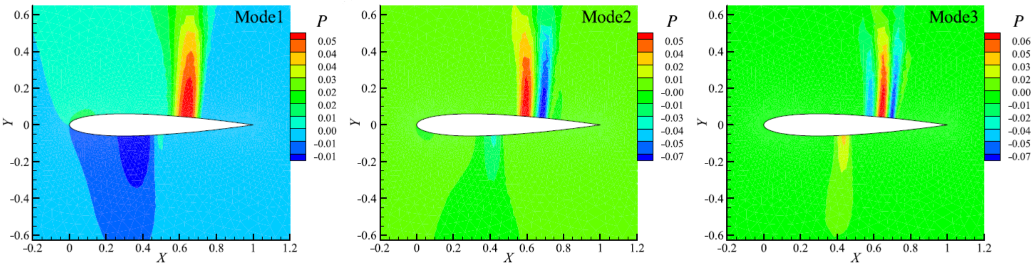


Figure 2.10: POD modes for a transonic flow (M=0.8) over NACA 0012 airfoil [46]

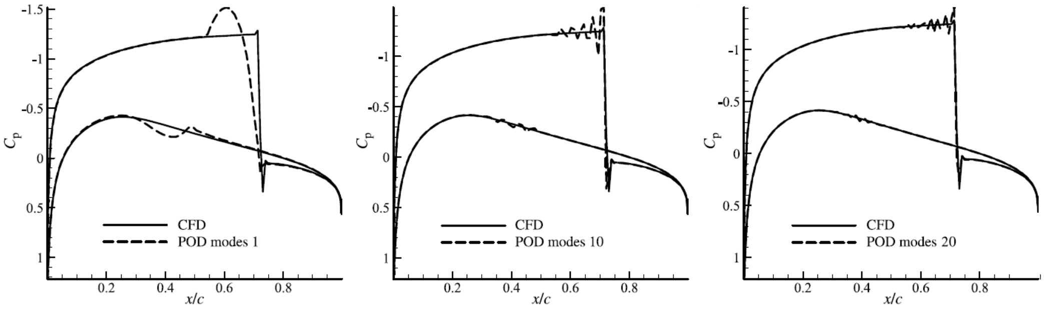


Figure 2.11: Comparison between CFD and POD of surface pressure distribution, and occurrence of Gibbs' phenomenon.[46]

.

The POD forms a reduced eigenvalue problem based on the snapshot matrix, resulting in a lower-dimensional optimal set of flow modes and corresponding eigenvalues that describe the energy content of each flow mode [61]. POD modes optimally represent the data in the L_2 sense, minimizing the L_2 norm between the reconstructed and original data. Specifically, the eigenvalues corresponding to the kinetic energy captured by the spatial POD modes are arranged in decreasing order from the largest to the smallest [76]. Consequently, the spatial POD modes are organized in terms of importance for capturing the kinetic energy of the flow field. In cases involving shock discontinuities, certain low-energy modes significantly influence the dynamics [65]. Therefore, the assumption that low-energy modes are insignificant is no longer valid.

2.3. Machine learning

Machine learning is a subfield within the broad class of artificial intelligence (AI), specifically focused on the development of algorithms capable of learning from data without relying on detailed mathematical or physical models [68]. Numerous groundbreaking advances in machine learning (ML) have arisen from the utilization of deep learning (DL), which relies on neural networks (NNs) featuring multiple hidden layers between input and output. A crucial factor contributing to the remarkable success of DL is its ability to learn hierarchically, as described by Vinuesa and Brunton [79]. Specifically, the initial layers of a deep learning neural network (DL NN) learn simple relationships within the data. As we progress to the deeper layers, these relationships are combined to understand more complex relationships. This hierarchical learning approach proves effective in modeling various physical problems that show complex hierarchical behavior, making DL and ML valuable tools in these fields. Machine learning is experiencing increased popularity within the field of Fluid Dynamics, with research particularly concentrated on three key aspects [79]: accelerating Direct Numerical Simulations (DNS) [6], enhancing turbulence modeling [17], and constructing Reduced Order Models (ROMs). The discussion will focus on the latter category, depicting some of the most commonly utilized NNs to model the low-order dynamics resulting from the modal reduction of a high-dimensional dynamical system, as elaborated in Section 2.1.

2.3.1. Artificial Neural Networks

Artificial Neural Networks (ANNs) constitute a category of machine learning models inspired by the biological Neural Networks present in the human brain. Neural Networks (NNs) stand out as one of the most significant architectures in deep learning [11]. Operating as fundamental nonlinear function approximators, NNs have gained considerable attention in recent years, with numerous studies investigating their efficiency. Hornik's universal approximation theorem [39], declares that any function can be approximated by a sufficiently large and deep network.

An ANN includes numerous individual units, often referred to as artificial neurons or perceptrons [64], connected with coefficients (weights) and create the form of the neural structure. Each neuron receives an input, processes it through an activation function, and produces an output. An illustration of neuron or perceptron is following in Figure 2.12:

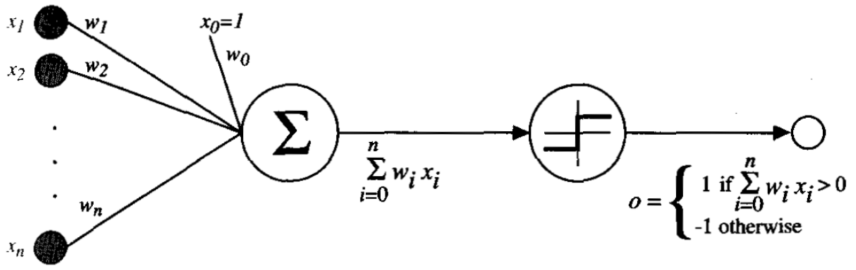

x_1
w_1
x_2
w_2
$x_0=1$
w_0
w_n
x_n
Σ
$\sum_{i=0}^{n} w_i x_i$
$o = \begin{cases} 1 \text{ if } \sum_{i=0}^{n} w_i x_i > 0 \\ -1 \text{ otherwise} \end{cases}$

Figure 2.12: Illustration of a perceptron [54].

Multiple neurons or perceptrons can be organized into various structures. The regression ability of these structures enhances insight into the problem and the nature of the data. One commonly used structure is the feedforward network, which includes several layers of neurons. The weighted output from one layer serves as the input for the next one. NN designs typically feature an input layer for data reception and an output layer for generating predictions. Deep feedforward networks, or Multi-Layer Perceptrons (MLPs), stand out as ideal deep learning models [30]. Nonlinear optimization techniques, such as backpropagation [66], are employed to determine network weights and minimize the error between predictions and labeled training data. Deep neural networks (DNNs) incorporate multiple layers and diverse nonlinear activation functions. When these activation functions are expressed through convolutional kernels, another class of networks is derived, called Convolutional Neural Networks (CNNs). CNNs have shown remarkable performance in image and pattern recognition [59]. In the case of these simple NNs, no feedback loop connects the outputs of the model back to itself. When feedforward NNs are developed to incorporate feedback loops, they are termed Recurrent Neural Networks.

2.3.2. Recurrent Neural Networks

As previously discussed, Recurrent Neural Networks (RNNs) [67] enhance feedforward neural networks by incorporating the output of adjacent time steps, introducing a temporal dimension to the model [47]. The network's edges that connect neighboring time steps are named recurrent edges. RNNs operate on sequences of data (e.g., video frames, time series), and their weights are determined through backpropagation through time [11]. An illustration of an RNN is provided in Figure 2.13:

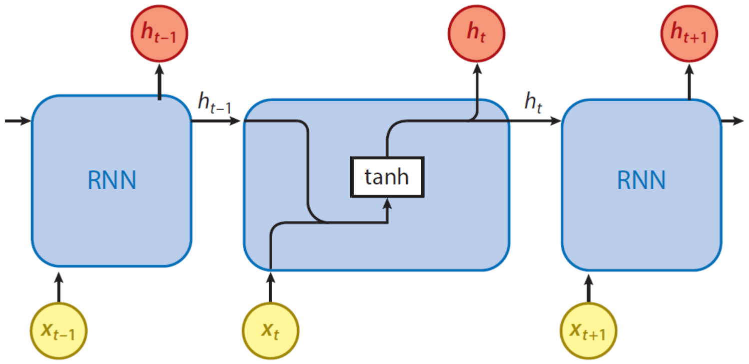


Figure 2.13: Recurrent Neural Network (RNN) for time series predictions. Abbreviations: h_{t-1} is the previous cell's output, h_t is the current cell's output, and x_t is the input vector [11].

On the other hand, the effectiveness of RNNs has been hindered by the exploding gradients during training. The renewed interest in RNNs results from the development of Long Short-Term Memory (LSTM) [36], or Gated Recurrent Units (GRUs) [14] algorithms. These algorithms utilize cell and gating mechanisms to store and forget information about past inputs. Hence, alleviating the issues with gradients and improving the transmission of long-term information, which is a common challenge for classic RNNs. RNNs demonstrate significant potential in predicting unsteady flow fields due to the introduction of the time concept. Consequently, they are of particular interest to fluid mechanics [11].

Long Short-Term Memory

Long Short-Term Memory (LSTM) network was introduced by Hochreiter and Schmidhuber [36] to address the issues of vanishing or exploding derivatives and short transmission of information in standard RNNs. The LSTM architecture replaces the hidden layer of a standard RNN with a memory 'cell.' Each memory cell contains a node with a self-connected recurrent edge of fixed weight, creating paths through time where gradients can flow without vanishing or exploding [47, 30]. A significant improvement is to make the weight on this self-loop depend on the context instead of being fixed, as proposed by Gers et al., [23]. The structure of the LSTM network is illustrated in Figure 2.14:

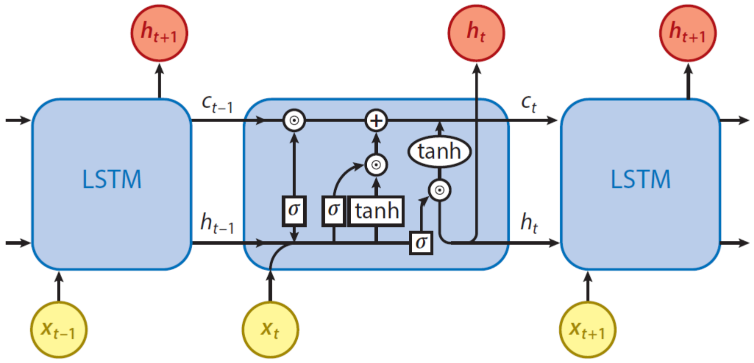


Figure 2.14: The Long-Short Term Memory (LSTM) neural network. Abbreviations: c_{t-1} is previous cell memory, c_t is current cell memory, h_{t-1} is the previous cell's output, h_t is the current cell's output, and x_t is the input vector [11].

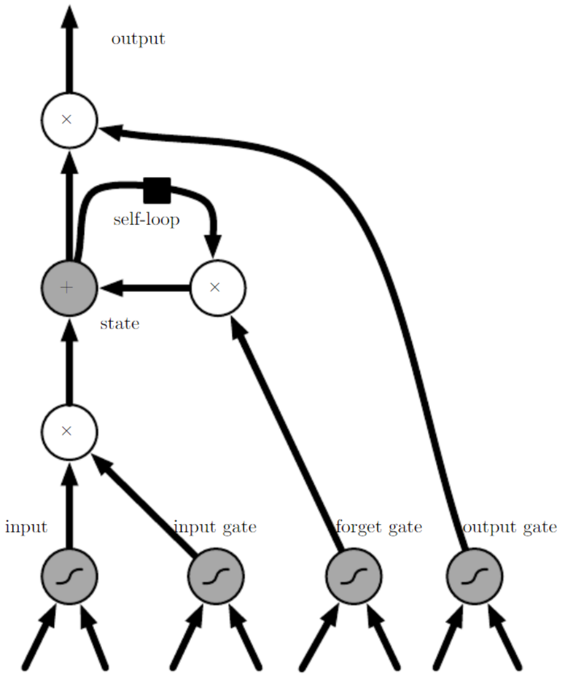


Figure 2.15: The Long-Short Term Memory (LSTM) 'cell', as illustrated in [30].

Based on Figure 2.15, the elements of the LSTM network are explained, following the notation of [47, 30, 11]: The LSTM cell has the same input (x_t) and output (h_t) vectors, as an standard RNN, as one can derive be comparing Figure 2.13 and Figure 2.14. But, it has more parameters and a system of gate units that controls the flow of information. The most important component is the state unit s_t, which has a linear self-loop. The state unit is given by Equation (2.12):

$$s_t = i_t \odot g_t + f_t \odot s_{t-1} \tag{2.12}$$

The i_t is the input node, which uses the activation function on a combination of the current input vector (x_t) and the previous hidden layer h_{t-1}. The input node is given by Equation (2.13)

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i) \tag{2.13}$$

Where σ is the sigmoid function. U_i and W_i are the input weights and the recurrent weights for the input node correspondingly. The b_i denotes the biases for the input node.

The g_t represents the external input gate, acting as a sigmoidal unit. Similar to the input node, it applies the sigmoid activation function in combination with the current input vector (x_t) and the previous hidden layer h_{t-1}. The term 'gate' is due to its property of allowing or blocking information flow. When its value is zero, it cuts off the values from other nodes. Conversely, when its value is 1, all the information passes through the gate. The input gate is given by the Equation (2.14):

$$g_t = \sigma(U_g x_t + W_g h_{t-1} + b_g) \tag{2.14}$$

Where, U_g, W_g, b_g are respectively the input weights, recurrent weights, and biases of the input gate.

To continue with, the self-loop weight is controlled by a forget gate unit f_t, which sets this weight to a value between 0 and 1 via a sigmoid unit. The forget gate is described by Equation (2.15):

$$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f) \tag{2.15}$$

Once again, U_f, W_f, b_f are respectively the input weights, recurrent weights and biases of the forget gate.

The output h_t of the LSTM cell is given by Equation (2.16):

$$h_t = \tanh(s_t) \odot q_t \tag{2.16}$$

The output H_t can controlled (shut off) by the output gate q_t, which also uses a sigmoid unit for gating:

$$q_t = \sigma(U_q x_t + W_q h_{t-1} + b_q) \tag{2.17}$$

Where U_f, W_f, b_f are respectively the input weights, recurrent weights and biases of the output gate.

Moreover, some parameters are not learned during the training process but are instead set by the user. These parameters are commonly referred to as hyperparameters. The key hyperparameters for the LSTM neural network include the number of hidden layers, the number of LSTM units in each layer, the learning rate, the batch size, the dropout rate, the number of epochs, and many others. The main hyperparameters are defined as follows:

- Number of hidden layers: The number of consecutive recurrent layers in the NN architecture.
- Number of LSTM units: The number of LSTM memory cells in each layer.
- Learning rate: This denotes the step size for the gradient descent optimization during backpropagation.
- Batch size: It represents the number of training examples utilized in a single iteration.
- Dropout rate: Is the percentage of nodes omitted during the training of a layer in each iteration.
- Number of epochs: Defines the number of times the entire dataset needs to be processed by the neural network.

As discussed previously, there are more hyperparameters to consider for the LSTM NN. For instance, the number of spatial POD modes and the required accuracy for the surface pressure distribution are hyperparameters for the NN. Each of these hyperparameters affects the NN differently, and selecting the correct values can significantly improve the NN's performance, influencing its training complexity. Catalani [12] and Bourier [10] conducted fine-tuning optimization to find the optimal set of hyperparameters for the LSTM NN, focusing on enhancing training efficiency for predicting the temporal evolution of the latent space. Given that a goal of this research is to reduce the number of degrees of freedom of the reduced-order basis, the hyperparameters that have been previously defined for the LSTM NN used at NLR will be adopted for the ePOD-LSTM approach. In the ePOD-LSTM reduced-order model (ROM), the LSTM network is used to predict the temporal evolution of the enriched reduced-order basis along various sections of the DLR-F22 model. This involves forecasting the time coefficients of the spatial modes as well as the time-dependent parameters of the enrichment modes. Hence, the number of dominant shock discontinuities represented by the enrichment basis was an additional hyperparameter.

3 Test Cases and ROM Architecture

Training data from high-fidelity simulations (computational or experimental) are utilized to enable non-intrusive Reduced-Order Models (NIROMs) to learn the system's dynamics. This constitutes the most computationally expensive phase in constructing a ROM, rendering it unfeasible when a vast amount of data is necessary for model training. In Stability and Control (S&C) applications, an efficient training maneuver is required to cover the regressor space of state variables [40], facilitating the construction of ROMs. McDaniel et al. [53] propose an approach to combine the high-fidelity results and the ROMs, as outlined below:

- Computational training maneuvers that represent relevant flow physics are solved using CFD simulations, resulting in a high-fidelity dataset.
- An ROM is constructed for the aircraft based on a training subset of high-fidelity data obtained from the CFD simulations.
- The ROM is validated by comparing its results with a testing subset of high-fidelity CFD data.
- Predictions for all flight test points are generated using the validated ROM.

Generating training data that effectively represents the desired complex flow phenomena is particularly crucial when neural networks are part of the ROM. This is mainly due to the fact that, in general, the size of a training dataset required to train a neural network is not known a priori, and issues related to overfitting or underfitting may arise [30].

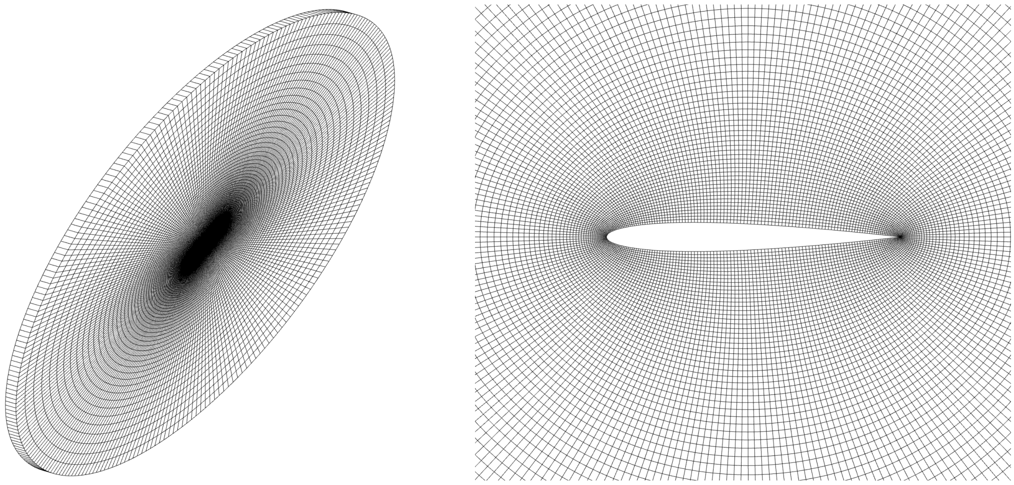
Previous studies conducted at NLR [10, 12] have demonstrated that the training data generated by the Schroeder maneuver [69] adequately captures the relevant flow physics and was utilized to develop the POD-LSTM ROM for subsonic cases. However, this does not guarantee that the same approach will be effective for transonic data, where different physical phenomena emerge. Investigating this issue further, however, was beyond the scope of this study. Additionally, the Schroeder maneuver is one of the two motions employed by the United States Air Force Academy (USAFA) [24] to generate high-fidelity data for the DLR-F22 aircraft configuration, which serves as the demonstrator for the proposed methodology.

Before introducing the DLR-F22 model and the maneuvers used to derive the training dataset, we first outline the NACA 0012 test case, along with the corresponding flight conditions and motions. The inviscid flow around the NACA 0012 airfoil in the transonic regime ($M = 0.85$) was selected as a use case to develop and test the proposed enriched Proper Orthogonal Decomposition (ePOD) method. The motivation for choosing this particular case lies in the fact that a symmetric airfoil in unsteady transonic flows can produce distinct and well-defined shock discontinuities in the pressure distribution, as observed in Figure 3.2.

3.1. NACA 0012

The symmetric four-digit airfoil, NACA 0012 [1], was selected for testing and developing the ePOD method. The unsteady inviscid flow in the transonic regime ($M_\infty = 0.85$, matching the free-stream velocity of the DLR-F22 model) was chosen to generate high-fidelity pressure distributions. Euler equation-based flow simulations around the NACA 0012 were performed using the in-house CFD solver of NLR, ENSOLV [43]. ENSOLV is an advanced CFD code with which three-dimensional steady or time-dependent, incompressible, or compressible flows around complex aerodynamic configurations can be computed. These configurations can be either fixed or moving relative to an inertial reference frame and can be either rigid or flexible, or consist of multiple bodies moving relative to one another.

An O-type grid was employed to solve inviscid flow around the NACA 0012 airfoil. The O-type mesh, consisting of 33 thousand cells, is illustrated in Figure 3.1.



(a) Iso view

(b) Side view

Figure 3.1: The O-grid around the NACA 0012 airfoil for Euler equation-based flow simulations.

The unsteady flow around the NACA 0012 airfoil under harmonic rigid motion was simulated. The parameters used for the time-accurate simulation and harmonic motion are listed in Table 3.1:

Table 3.1: Parameters of the time-accurate unsteady inviscid flow around the NACA 0012 airfoil.

Mach number	0.85
Time integration scheme	2^{nd} order implicit backward
Number of periods	2
Number of time steps per reference period	16
Reduced frequency (of rigid motion and flow)	0.152
Amplitude of rigid motion	10°
Phase lag of rigid rotation	0°

In addition to the parameters of Table 3.1, the rotation center of the rigid motion was set to be identical to the aerodynamic center of NACA 0012, which was located one-quarter of the airfoil chord length back from the leading edge. The pressure distribution on the suction side of the NACA 0012 airfoil at a characteristic time step, where a strong shock discontinuities appeared is visible in the following graph:

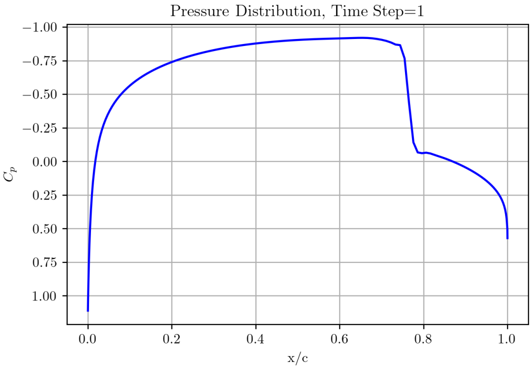


Figure 3.2: Pressure distribution on the upper surface of NACA 0012 Airfoil at the second timestep.

3.2. DLR-F22 Model

The DLR-F22 model configuration was utilized to evaluate the proposed Machine Learning based Reduced-Order Model for predicting more complex pressure distributions in transonic flows. The main design parameters are listed in Table 3.2, and the geometry of the DLR-F22 model is illustrated in Figure 3.3.

Table 3.2: Design parameters of the DLR-F22 model [75].

Model geometry	DLR-F22
Span width	0.5 [m]
Mean aerodynamic chord	0.22716 [m]
Projected wing area	0.112120 [m^2]
Moment reference point, x	0.13929 [m]

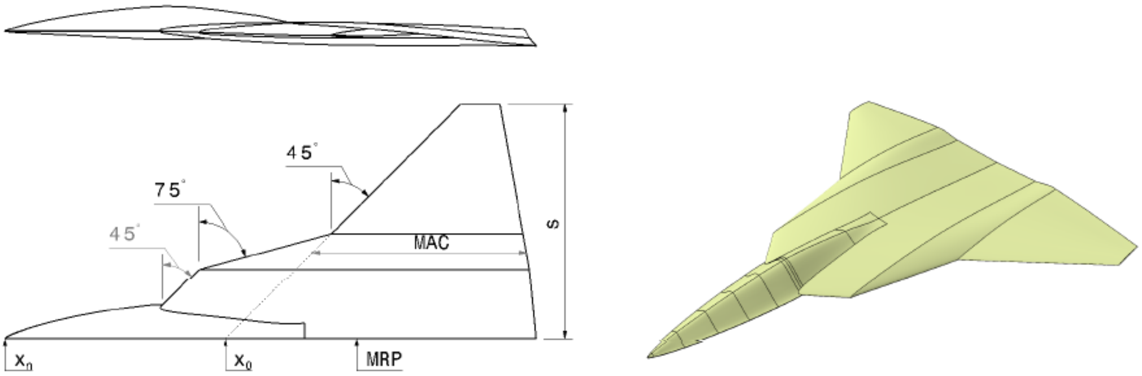


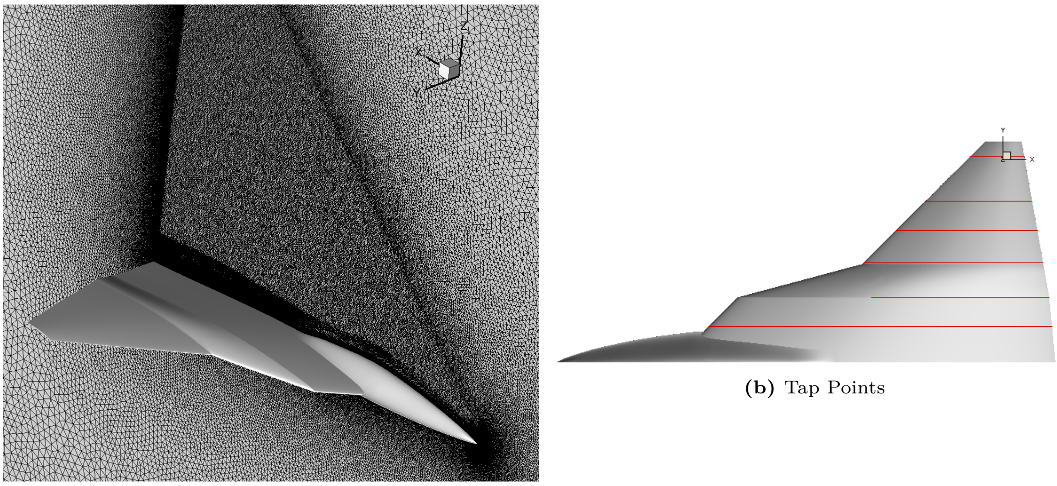
Figure 3.3: DLR F22 model configuration [75].

Figure 3.4a illustrates the hybrid computational grid for the half model, consisting of approximately forty million nodes. Quadrilaterals were used to resolve the boundary layer, while tetrahedra extended from the boundary layer edge to the far-field. To better capture vortices, a refined region was specified above the model, as shown in the figure. All first off-body grid nodes satisfy $y^+ < 1.0$, ensuring sublayer resolution. The far-field boundary was positioned at a distance equal to one hundred times the chord length.

Additionally, tap points were defined at six spanwise sections on the upper surface of the DLR-F22 model, as depicted in Figure 3.8b. The exact spanwise positions of these tap points are as follows:

$$y = \{0.05, 0.09, 0.136, 0.18, 0.22, 0.28\} \quad [\text{m}] \tag{3.1}$$

These points were fixed on the surface of the DLR-F22 model and move with the mesh. The output data include the tap coordinates and pressure coefficient values at every 200th-timestep for the six spanwise sections 1 to 6, where section 6 is located near the wingtip. The simulation runs for 20,000 timesteps over a duration of 4 seconds. For further details on the derivation of the pressure distribution dataset, please refer to [52]. Consequently, the available dataset for constructing the ePOD-LSTM ROM consists of the pressure distribution along these six spanwise sections on the upper surface of the DLR-F22 model, covering two training maneuvers: the PRBS signal and the Schroeder maneuver.



(b) Tap Points

(a) Computational Grid

Figure 3.4: (a) Near field of the computational grid of the DLR-F22 model, and (b) six spanwise sections [24].

All simulations were performed by USAFA using the CREATE$^{\text{TM}}$-AV/Kestrel CFD flow solver KCFD [52]. KCFD solver uses a second-order accurate cell-centered finite-volume discretization and solves the unsteady, three-dimensional, compressible RANS equations on hybrid unstructured grids [58] using the Method of Lines (MOL) to separate temporal and spatial integration schemes from each other. More details regarding the solver are described by McDaniel et al., [52]. Additionally, the Spalart–Allmaras with rotational/curvature correction (SARC) turbulence model introduced by [70] was chosen. For further details, please refer to Ghoreyshi et al., [24].

For the scope of this thesis, we considered motion in the vertical plane, specifically pitch and plunge motions, while neglecting the side slip and roll. Therefore, the unsteady simulations of the DLR-F22 model were focused on in-plane motions without side-slip. A prescribed-body motion was used to vary input parameters (angle of attack α and pitch rate q) under given freestream conditions. The flow conditions for the two motion types of the DLR-F22 model are listed in Table 3.3:

Table 3.3: Flow conditions for the two motions of DLR-F22 model.

Motion Type	α **[deg]**	q **[deg/s]**	$\mathbf{M}_\infty$ **[-]**	**P [Pa]**	**T [K]**	$\mathbf{Re}_\infty$**[-]**
PRBS Signal	$0°-20°$	-1000$-$1000	0.85	49881	266.5	$\approx 3 \times 10^6$
Schroeder Maneuver	$0°-20°$	-1000$-$1000	0.85	49881	266.5	$\approx 3 \times 10^6$

3.2.1. Pseudorandom Binary Sequence (PRBS) Signal

The first input signal includes a pseudorandom binary sequence (PRBS) motion. During that maneuver, the angle of attack and pitch rate vary in a periodic and deterministic manner, displaying characteristics similar to white noise [24]. Typical PRBS signals involve sudden variations between two distinct values (e.g., minimum and maximum angles of attack). However, the signal used in this thesis was modified so that the step changes depend on time.

An example of a PRBS signal is shown inFigure 3.5. The maximum possible period for a maximum

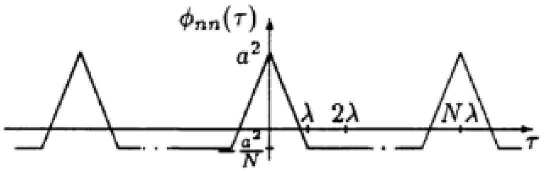


Figure 3.5: PRBS signal characteristics [5].

length sequence, N, is:

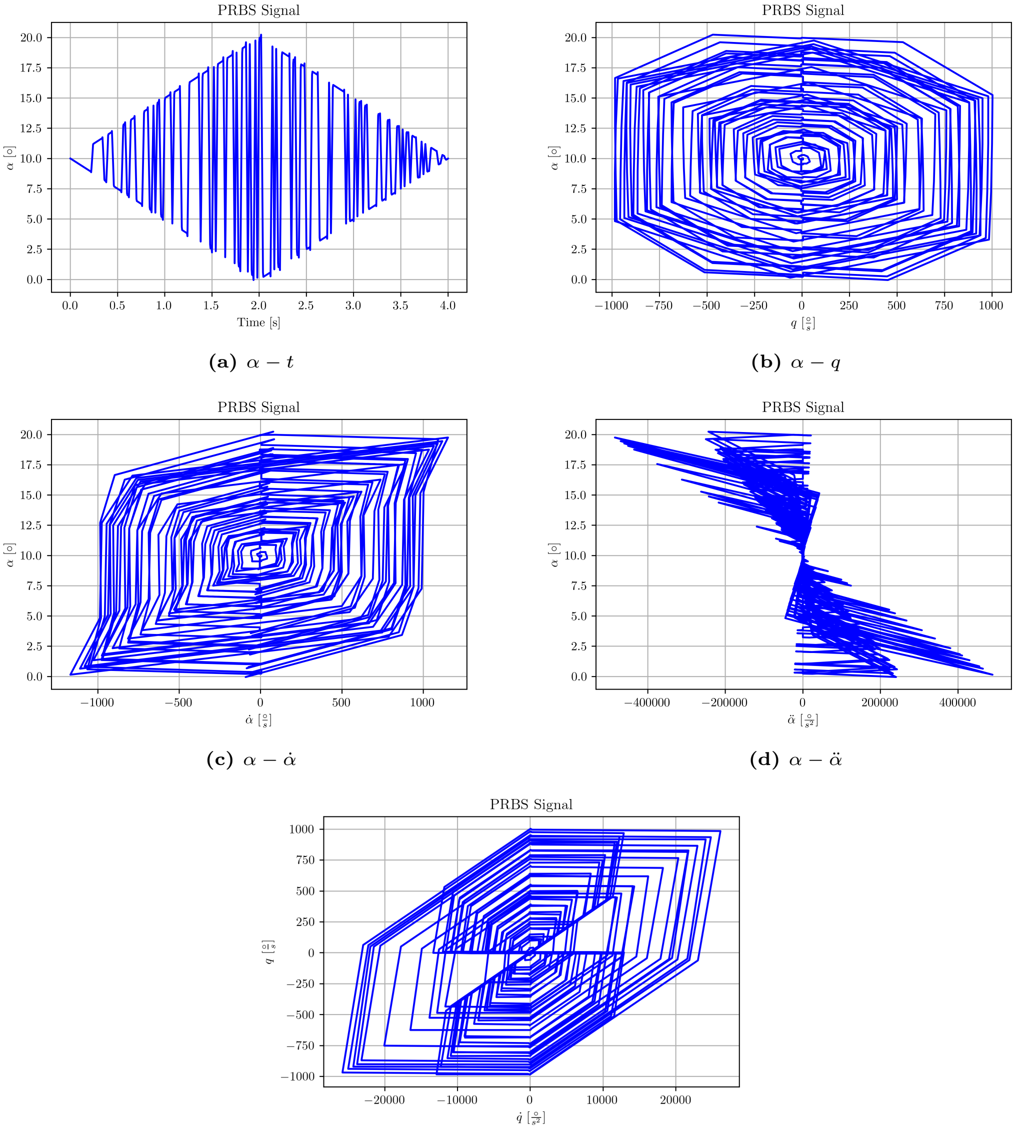
$$N = 2^m - 1 \tag{3.2}$$

where m represents the order of the PRBS signal. In Figure 3.5, λ denotes the shifting time, i.e. the duration of the shortest impulse, and a represents the amplitude of the PRBS signal. These parameters must be carefully selected, as they influence the reduced frequency, and thus the unsteadiness of the flow. The power spectral density of the PRBS signal shows that all frequencies up to $\frac{\omega_{\max}\lambda}{2} = \frac{\pi}{4}$ are excited, as discussed by Bányász [5] . The highest excited radial frequency $\omega_{\max}$ is related to the shorter time constant, $T_{\min}$, by the following equation:

$$\omega_{\max} = \frac{3}{T_{min}} \tag{3.3}$$

therefore $\lambda = \frac{\pi}{6}T_{\min}$. For more details regarding the PRBS refer to [5]. For the scope of this study, the PRBS signal had a duration of 4 seconds with a mean angle of attack of 10°. The amplitude, a, increases linearly from 0 to 10° for the first half of motion and then linearly falls to 0 for the second half of the signal. The number of shifts was set to 40. The PRBS signal was designed for the DLR-F22 model at a constant Mach number: $M_\infty = 0.85$, with pressure data on the upper surface recorded at six spanwise sections every two hundred time steps. In the case under consideration, the input signal in the angle of attack-pitch rate sample space suggests that the derivatives of these parameters might significantly influence the instantaneous pressure distribution. Therefore, it is essential to include these parameters in the input data for training the neural network. Hence, the input parameters of the designed PRBS signal are shown in Figure 3.6.

Figure 3.6a illustrates the variations in the angle of attack over time. The model is configured with a wind vector at $\alpha = 10°$, with the pitch angle initially set to zero [24]. In this way, the pitch rate and the time rate of changes in the angle of attack are the same. Figure 3.6b shows the angle of attack ($\alpha[°]$) versus pitch rate ($q[°/s]$). It is important to note that the maximum frequency of the signal, and consequently the pitch rate, was constrained to achieve a maximum reduced frequency of 0.01, ensuring quasi-steady aerodynamic behavior. Figure 3.6b demonstrates that the PRBS signal provides good coverage of the $\alpha - q$ input space. Figure 3.6d depicts the coverage of the $\alpha - \ddot{\alpha}$ space by the PRBS signal, where $\ddot{\alpha}$ is given in units of $[°/s^2]$. This figure shows a wide range of variations in $\ddot{\alpha}$, though most points are located at the center of the plot. Finally, Figure 3.6c and Figure 3.6e display the variations of α and q with their respective first-time derivatives.



(a) $\alpha - t$

(b) $\alpha - q$

(c) $\alpha - \dot{\alpha}$

(d) $\alpha - \ddot{\alpha}$

(e) $q - \dot{q}$

Figure 3.6: PRBS signal coverage space.

3.2.2. Schroeder Maneuver

As discussed previously, the second motion type for the DLR-F22 model was based on the Schroeder sweep input design method, where reference signals consist of oscillations with varying amplitudes and frequencies. The POD-LSTM ROM, which relies on the Schroeder maneuver for deriving the pressure distribution datasets, demonstrated better accuracy compared to models based on the chirp and spiral maneuvers [40, 10, 60, 12]. Therefore, the Schroeder maneuver was the primary maneuver considered for deriving the training datasets for the neural network.

In the following section, the Schroeder maneuver is represented. Specifically, an extension of the Schroeder sweep, introduced by Morelli [57], is employed to design multiple orthogonal input signals with optimized Relative Peak Factors (RPFs). The motion variables, namely the angle of attack and pitch angle, which describe the aircraft's motion, are represented as a phase-shifted set of sinusoids, as shown in Equation 3.4:

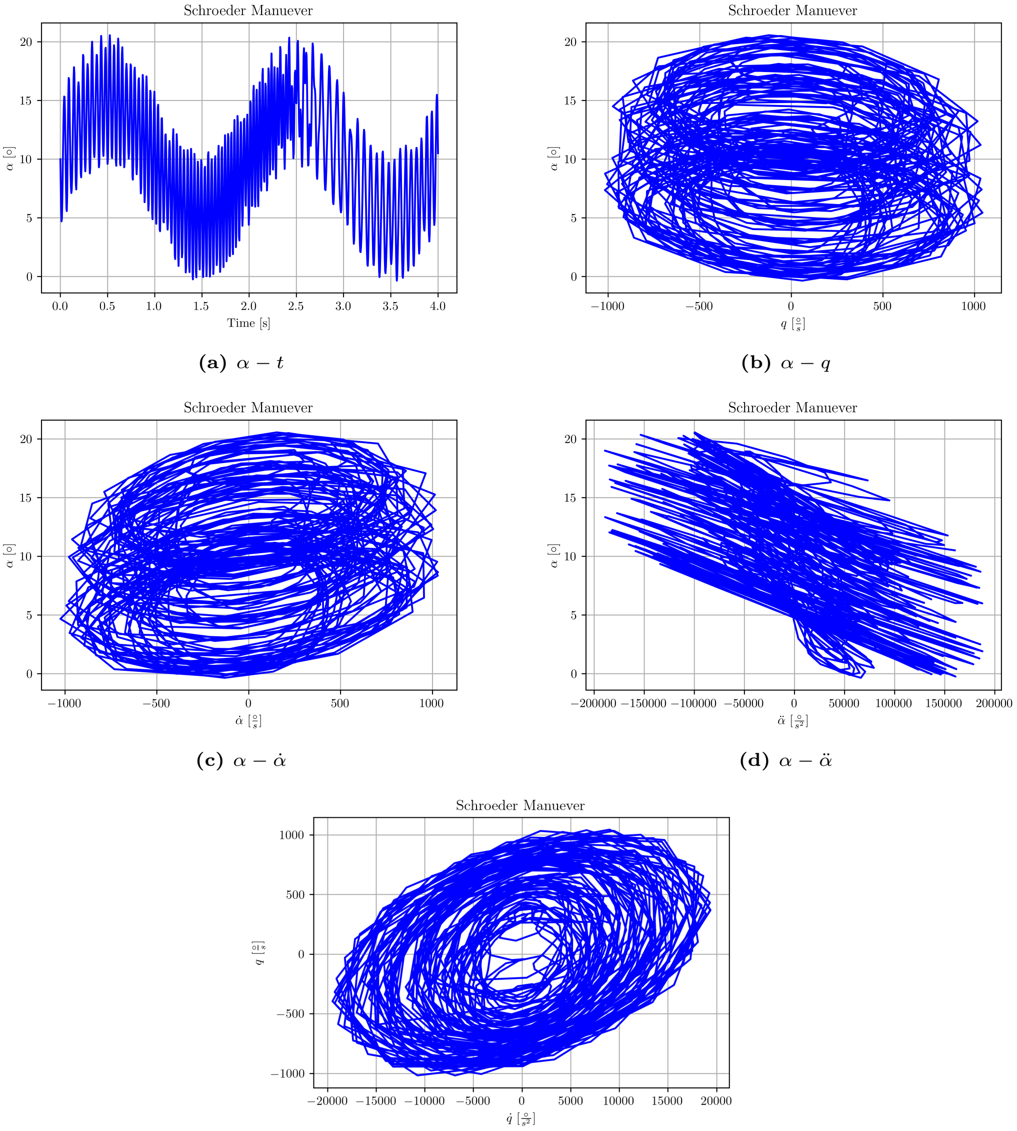
$$u_j = \sum_{k=1}^{N} A_k \cos\left(\frac{2\pi k t}{T} + \phi_k\right) \tag{3.4}$$

where u_j represents the motion variable (α or q), A_k is the amplitude of the maneuver, T is the time length of the maneuver, N is the total number of available frequencies, and ϕ_k denotes the phase angles. By utilizing the parameters N, A_k, and T, one can directly control the coverage of the regressor space through the Schroeder maneuver. Furthermore, the phase shift angles ϕ_k are selected to minimize the Regressor Performance Factor (RPF), which is defined by Equation 3.5:

$$RPF(u_j) = \frac{\max(u_j) - \min(u_j)}{2\sqrt{2}\text{RMS}(u_j)} \tag{3.5}$$

where $\text{RMS}(u_j)$ is the root mean square of the vector u_j. The Relative Peak Factor (RPF) serves as a metric for assessing the efficiency of an input in parameter estimation. Ideally, a low RPF is preferred, as it indicates that the input signals can extract valuable information from the system response with a comparatively low amplitude perturbation. The primary advantage of this technique is that the only a priori information required for its application is the expected frequency range of the dynamics modes. This technique results in multiple sinusoidal inputs that are orthogonal in both time and frequency domains while minimizing the Relative Peak Factor. These characteristics make the extension of the Schroeder sweep proposed by Morelli [57] a good training maneuver.

Similarly to the PRBS signal, the Schroeder maneuver had a duration of 4 seconds with a mean angle of attack of 10°, while the amplitude of the motion, A_k, was varied from 0 to 10°. The Schroeder maneuver was designed for the DLR-F22 model at a constant Mach number: $M_\infty = 0.85$, with pressure data on the upper surface recorded at six sections every two hundred time steps, resulting to 1000 pressure distribution snapshots. Once again, the model was configured with a wind vector at $\alpha = 10°$ angle of attack,and the pitch angle initially set to zero. The input parameters for the Schroeder maneuver are illustrated in Figure 3.7. Figure 3.7b shows the angle of attack ($\alpha[°]$) versus pitch rate ($q[°/s]$), demonstrating that the Schroeder maneuver provides good coverage of the $\alpha - q$ input space. Figure 3.7d depicts the coverage of the $\alpha - \ddot{\alpha}$ space. Finally, Figure 3.7c and Figure 3.7e illustrate the variations of α and q along with their first derivatives.



(a) $\alpha - t$

(b) $\alpha - q$

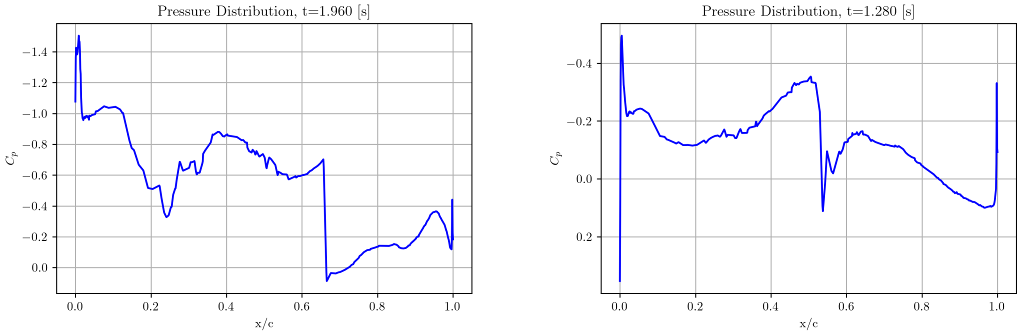
(c) $\alpha - \dot{\alpha}$

(d) $\alpha - \ddot{\alpha}$

(e) $q - \dot{q}$

Figure 3.7: Schroeder maneuver coverage space.

Thus, the two types of motion used to derive the available dataset for constructing the ePOD-LSTM ROM were the PRBS signal and the Schroeder maneuver. Pressure distributions on the upper side of the DLR-F22 model at time steps where strong shock discontinuities appeared, are illustrated in Figure 3.8 to highlight the typical distributions that the proposed ROM should predict.



(a) PRBS Signal, t=1.96s, α=19.75°

(b) Schroeder Maneuver, t=1.28 s, α=5.99°

Figure 3.8: Typical pressure distribution on the upper surface of the DLR-F22 Model, Section 1.

The main challenge in predicting such pressure distributions using a reduced-order basis is related to accurately capturing the strong shock waves. As discussed, the proposed ePOD-LSTM ROM introduces an enrichment basis particularly designed to approximate these pressure discontinuities, e.g. the shock waves appear in Figure 3.8. The remainder of the pressure distribution, excluding the shocks, is approximated using a standard basis. Furthermore, Figure 3.8 confirms that the previously presented NACA 0012 test case serves as a strong foundation for the initial development of the ePOD.

3.3. The ePOD-LSTM ROM

The overall structure of the enriched Proper Orthogonal Decomposition Long Short-Term Memory (ePOD-LSTM) Reduced-Order Model (ROM) is outlined in this paragraph.

Developing a ePOD-LSTM ROM consists of first deriving the enriched reduced-order basis from the Full-Order Model (FOM) and then constructing and training the LSTM neural network. As detailed in Chapter 4, the reduced-order basis contains the standard spatial modes ϕ_k and their associated time coefficients $a_k(t)$, along with the enrichment modes ϕ_e and its time-dependent parameters $p(t)$.

After obtaining the enriched reduced-order basis, the time coefficients $a_k(t)$ and enrichment parameters $p(t)$ of the training, validation, and testing of neural network was extracted. In addition to these values, the input signal included the angle of attack α and pitch rate q. The sample space $\alpha - q$ suggested that the derivatives of these parameters might significantly influenced the instantaneous pressure distribution. Consequently, the time-varying input vector consisted of the angle of attack α, along with its first and second-time derivatives $\dot{a}$ and $\ddot{a}$, respectively, as well as the pitch rate q, and its first-time derivative $\dot{q}$. The next step was the preparation of the time coefficients $a_k(t)$ and parameters $p(t)$ and the input vector for all the datasets. The dataset was divided into training, validation, and testing subsets for the LSTM neural network. During the training phase, the LSTM network learned the patterns of the training data subset by adjusting its weights and biases to minimize the L2 norm between the reconstructed pressure distribution and FOM. The pressure distribution was reconstructed based on the enriched reduced-order basis and the forecast for its time-dependent parameters. Chapter 5 provides details regarding the training phase of the LSTM neural network.

In the verification stage, of ePOD-LSTM ROM, the trained LSTM neural network was employed to predict the temporal evolution of the enriched reduced-order basis for new, unseen datasets. The model forecasts the normalized time coefficients $a_k^{NN}(t)$, corresponding to the standard spatial modes ϕ_k, and parameters $p^{NN}(t)$, associated with the enrichment modes ϕ_e, of the enriched reduced-order basis. The pressure distribution was reconstructed utilizing the enriched reduced-order basis (refer to Equation 4.2) and the predicted de-normalized coefficients and parameters.

A visual representation of the ePOD-LSTM workflow is provided in 3.9.

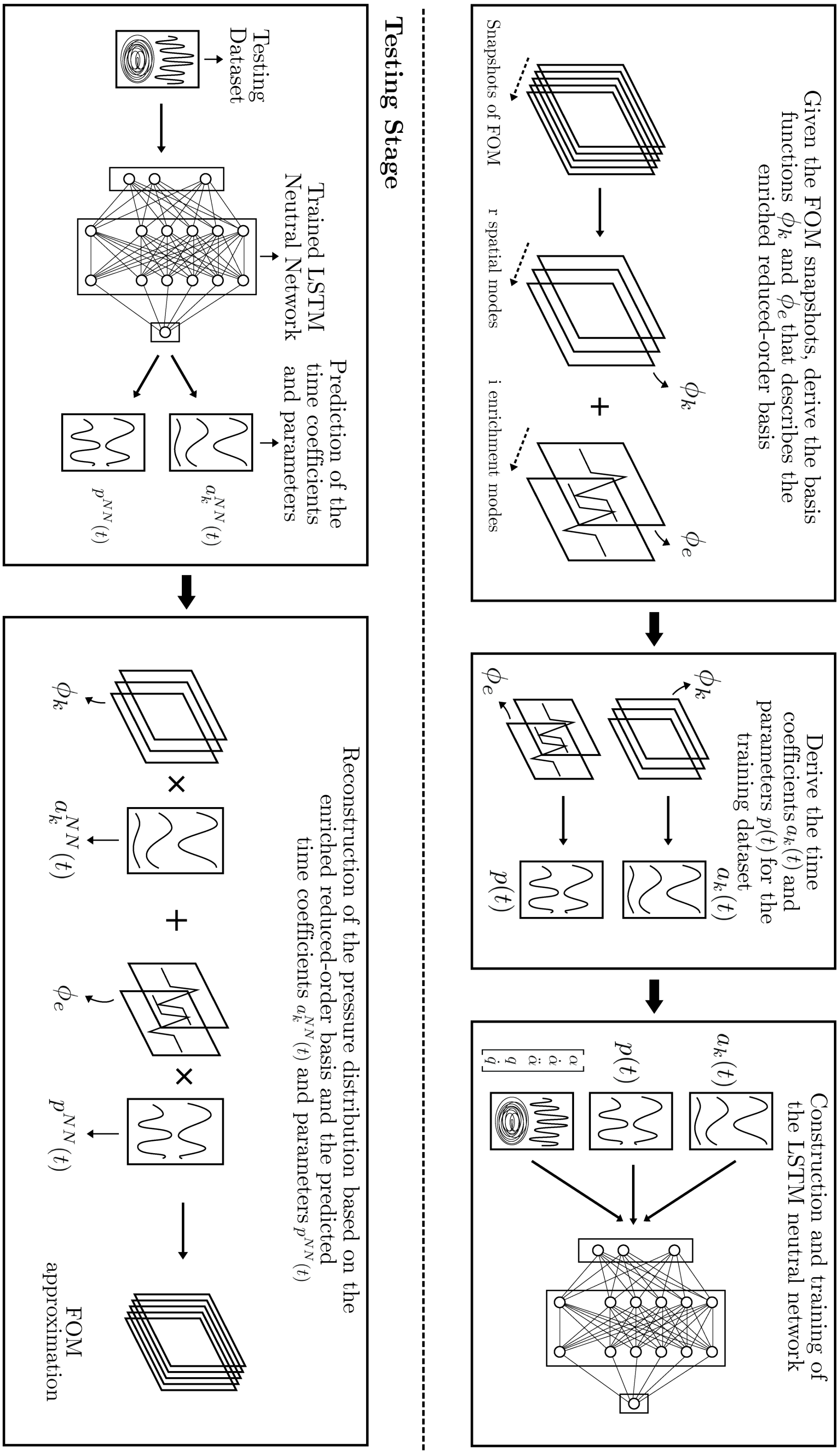

Model Construction and Training
Given the FOM snapshots, derive the basis functions ϕ_k and ϕ_e that describes the enriched reduced-order basis
Snapshots of FOM
r spatial modes
i enrichment modes
ϕ_k
+
ϕ_e
Derive the time coefficients $a_k(t)$ and parameters $p(t)$ for the training dataset
ϕ_k
ϕ_e
$a_k(t)$
$p(t)$
Construction and training of the LSTM neutral network
$a_k(t)$
$p(t)$
α $\dot{\alpha}$ $\ddot{\alpha}$ q $\dot{q}$
Testing Stage
Testing Dataset
Trained LSTM Neutral Network
Prediction of the time coefficients and parameters
$a_k^{NN}(t)$
$p^{NN}(t)$
Reconstruction of the pressure distribution based on the enriched reduced-order basis and the predicted time coefficients $a_k^{NN}(t)$ and parameters $p^{NN}(t)$
ϕ_k
×
$a_k^{NN}(t)$
+
ϕ_e
×
$p^{NN}(t)$
FOM approximation

Figure 3.9: Schematic representation of the ePOD-LSTM Reduced-Order Model.

4 Enriched Proper Orthogonal Decomposition

The enriched Proper Orthogonal Decomposition (ePOD) was applied to the fluctuation of the pressure distribution, similar to POD as discussed in Section 2.2. The time-averaged pressure distribution, denoted as $\bar{C_p}$, was defined separately to enable the model to focus on predicting fluctuations. In other words, the pressure distribution snapshots were decomposed into pressure fluctuations C_p^* and the temporal mean $\bar{C_p}$ as follows:

$$C_p(x,t) = C_p^*(x,t) + \bar{C_p}(x) \tag{4.1}$$

The proposed enriched Proper Orthogonal Decomposition (ePOD) method introduced an enrichment basis specifically designed to explicitly account for shock discontinuities within the latent space. This approach effectively isolates the shock regions, allowing the standard truncated basis to focus on representing the remaining pressure distribution. To implement ePOD, we integrated the enrichment basis into the reduced-order basis as follows:

$$C_{p_k} = \underbrace{\sum_{k=1}^{r} a_k(t)\phi_k(x)^T}_{\text{Standard Basis}} + \overbrace{\sum_{s=1}^{i} \phi_{e_s}(x,p(t))}^{\text{Enrichement Basis}} + \bar{C_p} \tag{4.2}$$

Where r is the number of truncated spatial modes, $\phi_k(x)$ are the spatial modes with corresponding time coefficients $a_k(t)$, i is the number of shock discontinuities, and $\phi_e(x,p(t))$ are the enrichment modes, with time-variant parameters $p(t)$. Once the enrichment basis was defined, the remaining pressure distribution was interpreted using a standard truncated basis, similar to POD. The first step in defining the enrichment basis involved identifying the locations of pressure discontinuities. Therefore, a list of these pressure discontinuities was constructed, referred to as the 'map of discontinuities.'

4.1. Map of Discontinuities

The map of discontinuities identified the spatial and temporal locations of pressure discontinuities, which served as the foundation for defining the enrichment domain. Essentially, it highlighted the regions within the dataset where the enrichment basis was active. For this project, two methods for determining the locations of discontinuities were considered: physics-based sensors and error-based sensors. The following sections will provide a detailed description of these approaches. It is important to note that before constructing this map, the data was pre-processed, facilitating the application of the shock sensors. Specifically, linear interpolation was applied to achieve an evenly spaced pressure distribution along the x-direction at each time step, and the coordinates were normalized along the x-axis. Furthermore, the raw data was pre-processed using a low-pass filter to smooth out high-frequency noise in the dataset, preventing it from affecting the shock sensor's efficiency. The Whittaker smoother, proposed by Eilers [18], was employed with a first-order smoothing penalty of $\lambda = 10^2$. The smoothing was particularly relevant in cases where significant small-scale pressure fluctuations occur, which can impact sensor performance. The smoothed pressure fluctuation $\hat{C}_p^*$ is visible in Figure 4.1, for DLR-F22 model.

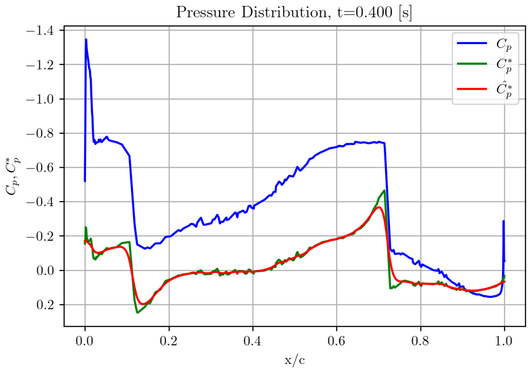


Figure 4.1: Impact of whittaker smoother on pressure fluctuations.

4.1.1. Physics-Based Sensors

Physics-based sensors were employed to determine the shock discontinuities locations. For every time instant of the Full-Order Model snapshots, the gradient of pressure fluctuations was evaluated locally in space. For this purpose, the gradient of the pressure fluctuations was computed using a backward-difference approximation, as follows:

$$\frac{\partial C_p^*(x_i, t)}{\partial x} = \frac{C_p^*(x_i, t) - C_p^*(x_{i-1}, t)}{x_i - x_{i-1}} \tag{4.3}$$

Once the approximation of the gradient of the pressure fluctuations was derived, it was then assessed based on a user-defined criterion. Specifically, the gradient was evaluated and flagged if it exceeded a specified threshold, as indicated in Equation 4.4:

$$\frac{\partial C_p^*(x_i, t)}{\partial x} >= 1 \tag{4.4}$$

The threshold in Equation 4.4 was a default value that can be adjusted according to the specific dataset. In this thesis, a default threshold value of 1 was used. A schematic representation of the physics-based sensor's operation is shown in Figure 4.2.

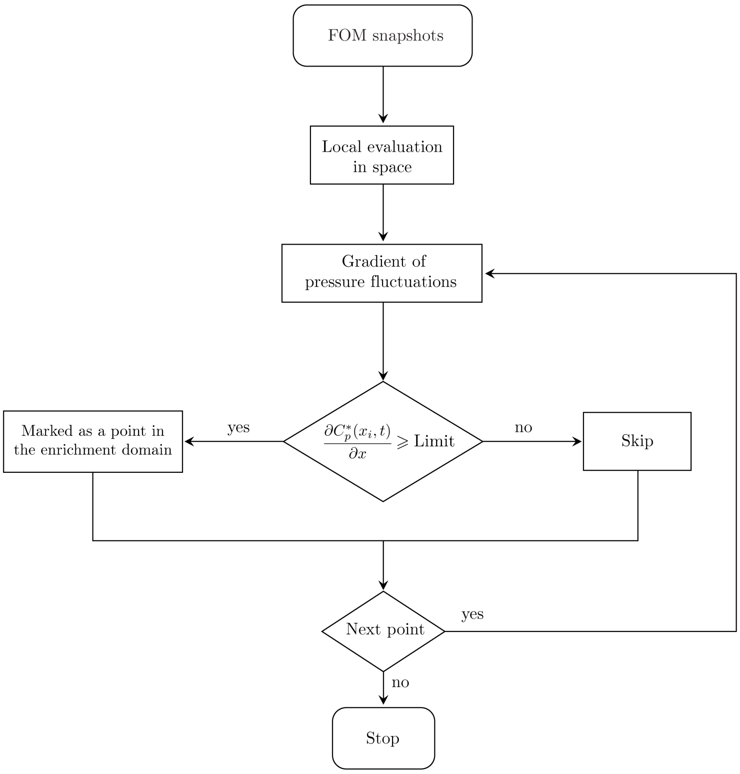

FOM snapshots
Local evaluation in space
Gradient of pressure fluctuations
$\frac{\partial C_p^*(x_i,t)}{\partial x} \geqslant$ Limit
yes
no
Marked as a point in the enrichment domain
Skip
Next point
yes
no
Stop

Figure 4.2: Flowchart of the physics-based sensor.

Once the sensor identified all flagged locations, the points were clustered based on their distances. Each cluster represented a pressure discontinuity at a specific time instance. The distance limit for cluster separation was determined by the pressure distribution's resolution. In the analyzed cases, points were assigned to different clusters if they were more than 10% of the chord length apart. For each cluster or shock, three characteristic points were defined as illustrated in Figure 4.3. The primary point was the shock center x_s, which was identified as the point of maximum pressure gradient for each cluster:

$$x_s = \arg\max_x\left(\frac{\partial C_p^*(x_i,t)}{\partial x}\right) \tag{4.5}$$

The second point was the maximum pressure location, while the third one was the shock foot, which was defined as the point with the largest second derivative of pressure fluctuations, as discussed in [44]. These locations were defined by the following equations:

$$x_{C_{p,\max}^*} = \arg\max_x(C_p^*(x_i,t)) \tag{4.6}$$

$$x_{\text{foot}} = \arg\max_x\left(\frac{\partial^2 C_p^*(x_i,t)}{\partial x^2}\right) \tag{4.7}$$

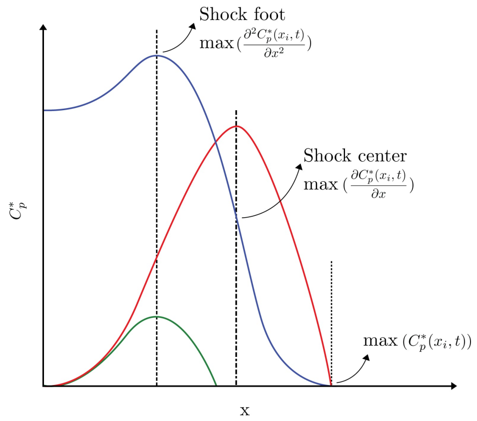

Shock foot
$\max(\frac{\partial^2 C_p^*(x_i,t)}{\partial x^2})$
Shock center
$\max(\frac{\partial C_p^*(x_i,t)}{\partial x})$
$\max(C_p^*(x_i,t))$
C_p^*
x

Figure 4.3: Application of the detection method, representative of a typical pressure distribution across a shock wave.

These three points provided essential information regarding the shock strength and location, which was valuable for defining the enrichment basis.

4.1.2. Error-Based Sensors

Error-based sensors were also tested as an alternative to physics-informed ones. These sensors were based on measuring the squared distance between the reduced-order basis and the true pressure distribution. The key observation was that this distance, derived using the standard Proper Orthogonal Decomposition (POD), tended to be larger at locations with pressure discontinuities. As noted earlier, higher frequencies were necessary to accurately represent discontinuities in the shock region, resulting in significant discrepancies between the truncated basis and the Full-Order Model (FOM). This observation led to the following procedure for deriving the enrichment domain:

First, the standard POD was performed, and the pressure distribution was reconstructed using the truncated basis with a specified number of spatial POD modes. Ten truncated spatial modes used in the cases under consideration. Next, the projection error of POD was calculated. This error was defined as the difference between the FOM and its projection on the reduced-order basis, quantified using the Mean-Squared Error (MSE). The projection error for the standard POD is given by Equation 4.8.

$$\epsilon_{\text{p,POD}} = \frac{1}{N_x}\sum_{i=1}^{N_x} ||C_p^*(x,t) - \sum_{k=1}^{r} a_k(t)\phi_k(x)^T||^2 \tag{4.8}$$

where N_x is the number of chordwise locations, C_p^* is the high-fidelity pressure coefficient fluctuations and r is the number of truncated spatial modes. The error was evaluated globally in space and locally in time. If the integrated projection error at every time instant exceeds a specified limit, the current time instant was flagged for enrichment. This limit was empirical and depends on the use case and desired accuracy of the reduced coordinates. For the tested cases, a desired accuracy of maximum projection error $1 \cdot 10^{-2}$ was selected, thus this limit was defined correspondingly. For flagged time instants, the squared distance between the reduced-order basis and FOM was evaluated locally in space, by the following equation:

$$d(x,t) = |C_p^*(x,t) - \sum_{k=1}^{r} a_k(t)\phi_k(x)^T|^2 \tag{4.9}$$

Furthermore, the maximum value of the distance in the current time instant was identified, and the local distance in space and time was evaluated based on the following optimality criterion:

$$d(x,t) > \frac{1}{3}\max(d(x,t)) \tag{4.10}$$

When the value of the local error exceeds one-third of the maximum value, it was marked as a point within the enrichment domain. This criterion is similar to the optimality criterion of hp adaptive methods as defined by Demkowicz et al. [15]. The operation of the error-based sensor is represented in the following flowchart.

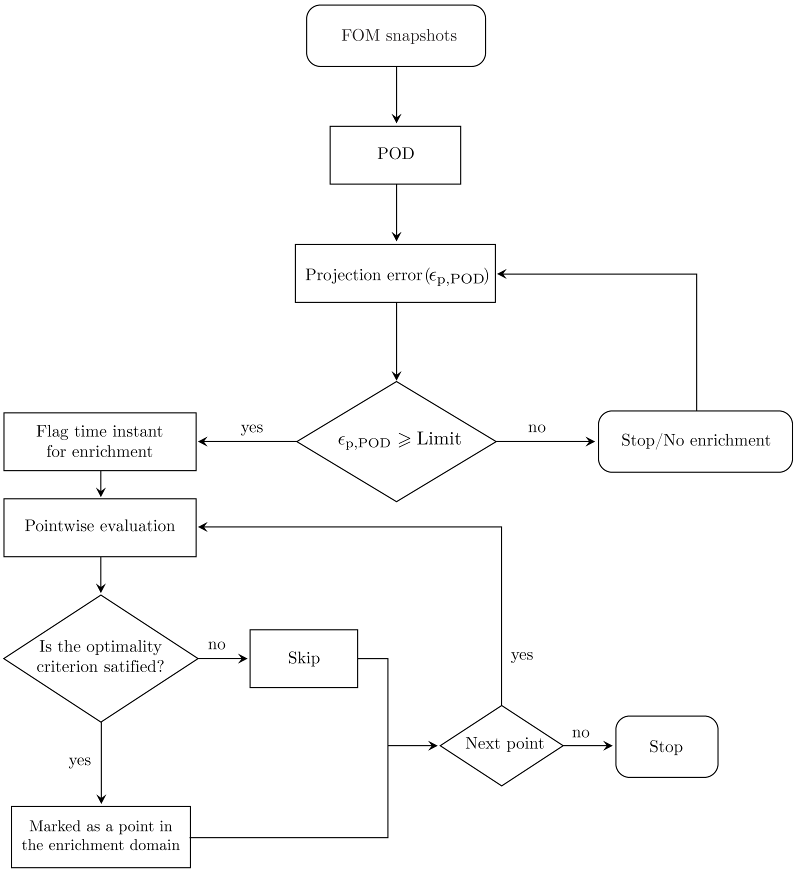
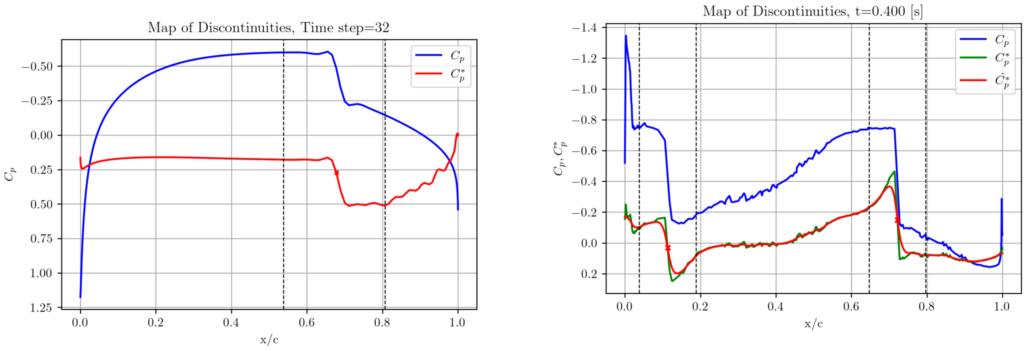

FOM snapshots
POD
Projection error ($\epsilon_{\mathrm{p,POD}}$)
Flag time instant for enrichment
yes
$\epsilon_{\mathrm{p,POD}} \geqslant$ Limit
no
Stop/No enrichment
Pointwise evaluation
Is the optimality criterion satified?
no
Skip
yes
yes
Next point
no
Stop
Marked as a point in the enrichment domain

Figure 4.4: Flowchart of the error-based sensor.

Similar to the physics-based sensors, the flagged locations were clustered based on point distances, with each cluster representing a pressure discontinuity at the current time instance. The same distance threshold between clusters, as used for the physics-based sensors, was applied here. After the clusters were derived, the procedure follows the same steps outlined in Section 4.1.1. The performance of the two sensors was similar for the test datasets considered in this report, therefore the physics based sensor was selected.

4.2. Enrichment Domain

After the sensor detected the centers of shock discontinuities, the next step was to define the enrichment domain. The enrichment domain refers to the localized region in space and time where the enrichment basis was active. To establish this domain, a user-specified value must be provided, which determines the size of the region surrounding the shock center, x_s, that will be included within the enrichment domain. Typically, this parameter ranges from 10% to 20% of the airfoil or wing chord. For the NACA 0012 airfoil and DLR-F22 model, a value of 20% of the chord was selected. Figure 4.5, illustrates examples of the enrichment domain (represented by black dashed lines).



(a) NACA 0012

(b) Section 1 of the DLR-F22 Model, Schroeder maneuver

Figure 4.5: Illustration of the enrichment domain definition.

Although using the shock foot x_{foot} and maximum pressure $x_{C^*_{p,\max}}$ locations for determining the enrichment domain might seem beneficial, it introduced additional parameters into the enrichment basis, thereby increasing the number of time-variant parameters. Hence, the neural network would need to predict more parameters, thus increasing the training cost. Therefore, the shock center was selected as the only time-variable parameter required to define the enrichment domain.

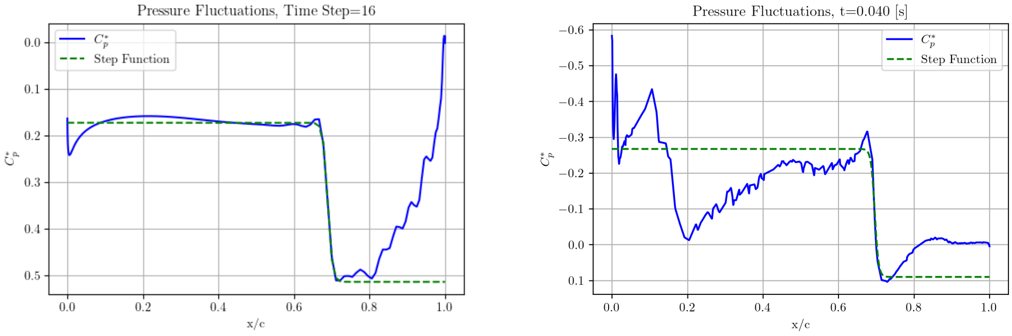
4.3. Enrichment Function

4.3.1. Step Function

Our initial tests employed a sharp enrichment function to represent the shock, as illustrated in Figure 4.6. The enrichment function was fitted to the Full-Order Model (FOM) using non-linear least squares, similar to Section 4.4. This function was based on modification of the Soboleva hyperbolic tangent [74] and was defined as follows:

$$\phi_e(x, p(t)) = \frac{w_1 e^{a(x-x_s)} + w_2 e^{b(x-x_s)}}{w_3 e^{a(x-x_s)} - w_4 e^{b(x-x_s)}} \tag{4.11}$$

Where, the parameters $p(t)$ of this enrichment function are the constants $\{a(t), b(t), w_1(t), w_2(t), w_3(t), w_4(t)\}$ and $x_s(t)$ are the locations of the shock center.



(a) NACA 0012

(b) Section 1 of the DLR-F22 Model, Schroeder maneuver

Figure 4.6: Step function in relation to the pressure fluctuations.

The global nature of the step function and the constraints on its parameters presented several challenges for the enrichment basis. These challenges will be discussed in detail in the consequence paragraphs.

4.3.2. Local Sawtooth Function

To improve the regression of specific discontinuities, the enrichment modes were localized within enrichment domains. In practice, the enrichment modes were set to zero outside their respective domains.

Additionally, zero boundary conditions were applied at the boundaries of the enrichment domains to ensure a smooth transition between the enrichment domain and the surrounding pressure distribution. The local sawtooth enrichment function, representing the enrichment mode, was either of first or higher order.

Piecewise Linear Sawtooth Enrichment

A piecewise linear function was one of the two local sawtooth enrichment functions proposed as enrichment modes. The first-order sawtooth enrichment function was defined as follows:

$$\phi_e(x, p(t)) = \begin{cases} \frac{ax}{x_1}, & \text{for } x < x_1 \\ \frac{(b-a)x}{x_2-x_1} + \frac{ax_2-bx_1}{x_2-x_1}, & \text{for } x_1 < x < x_2 \\ \frac{-bx}{1-x_2} + \frac{b}{1-x_2}, & \text{for } x > x_2 \end{cases} \tag{4.12}$$

Where, $p(t)$ are the time-dependent parameters of this enrichment function. The parameters $p(t)$ of the enrichment basis ware the amplitudes and locations of two interior control points: $\{a(t), b(t), x_1(t), x_2(t)\}$. The piecewise linear sawtooth enrichment was fitted to the pressure fluctuations in the enrichment domain, as described in a subsequent section. The piecewise linear sawtooth enrichment for Section 1 of DLR-F22 model under the Schroeder maneuver is illustrated in Figure 4.7:

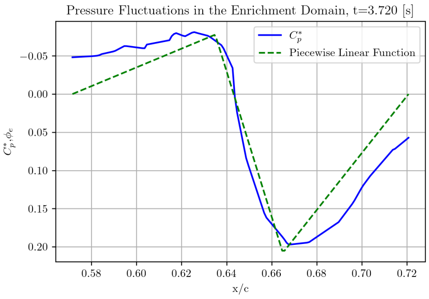


Figure 4.7: Piecewise linear enrichment function, Section 1 of the DLR-F22 Model, Schroeder maneuver.

B-Spline Sawtooth Enrichment

In many applications, particularly when viscosity is considered, pressure discontinuities are not as steep as in inviscid flows. Therefore, a higher-order continuous enrichment mode may be more appropriate. To address this, a cubic B-spline sawtooth enrichment was introduced into the method to serve as the enrichment mode in such cases. To start with, the spline function was defined based on the following B-spline basis:

$$S(x) = \sum_{i=0}^{n-1} c_i B_{i,k}(x) \tag{4.13}$$

Where $B_{j,k,t}$ are B-spline basis functions of degree $k = 3$ and knots t. The B-Spline basis elements were defined via the Cox-de Boor recursion formula [9], as follows:

$$\begin{aligned} B_{i,0}(x) &= \begin{cases} 1 & \text{if } t_i \le x < t_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \\ B_{i,k}(x) &= \frac{x - t_i}{t_{i+k} - t_i} B_{i,k-1}(x) + \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x) \end{aligned} \tag{4.14}$$

In this section, the x and y coordinates of the control points were utilized in order to define the coefficients and basis of B-spline. In practice, we used a cubic B-spline interpolation with specified boundary conditions, the knots (t) and coefficients (c) were determined by the data points, the degree, and the boundary conditions. The control points were four, and giving by the following equation:

$$P = [(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)] \tag{4.15}$$

Zero boundary conditions were imposed on the enrichment domain boundaries, thus the first and last control points were not real unknowns to the problem and equal $(x_0, y_0) = (0, 0)$ and $(x_3, y_3) = (1, 0)$. Thus, the control points in the problem were defined as follows:

$$P = [(0,0), (x_1, y_1), (x_2, y_2), (1,0)] \tag{4.16}$$

The values in the knot vector define where and how the B-spline basis functions were applied. Dirichlet and Neumann boundary conditions were imposed, meaning the y-values of the data points and the first derivative at the boundaries of the enrichment domains were set to zero. To impose these boundary conditions, we selected the first k knots equal to x_0 and the last k knots equal to x_3. In other words, the knot vector had repeated knots at the ends to enforce the boundary conditions. Thus, the knot vector was defined as follows :

$$t = (\underbrace{0, 0, 0}_{\text{3 knots}}, \overbrace{0, x_1, x_2, 1}^{\text{4 internal knots}}, \underbrace{1, 1, 1}_{\text{3 knots}}) \tag{4.17}$$

In Equation 4.17, the first three knots were 0 (equal to x_0), indicating boundary conditions at the start, and the last three knots were 1 (equal to x_3), indicating boundary conditions at the end. The internal knots were non-uniform and were chosen based on the distribution of the control points.

Cubic B-spline basis functions $B_{i,3}(x)$ were defined recursively using the Cox-de Boor recursion formula Equation 4.14. B-spline basis functions were constructed using the knot vector Equation 4.17 and were a piecewise cubic polynomial defined over four consecutive knot intervals, Equation 4.14.

The convention was that for t knots there were $t - k - 1$ coefficients. The coefficients were the values that the B-spline uses to interpolate the curve. These coefficients were closely related to the y-values of the control points. However, when boundary conditions were applied, the actual coefficients used by the spline might differ slightly to meet these conditions. The coefficients were computed to ensure that the spline passes through the given data points while satisfying the boundary conditions. For each data point (x_j, y_j), the next equation is true:

$$y_j = \sum_{i=0}^{n-1} c_i B_{i,k}(x_j) \tag{4.18}$$

This results in a linear system $Ac = yA$, where A is the design matrix of basis function values evaluated at the data points x_j, c is the vector of coefficients c_i, and y is the vector containing y_j values. The system was solved for c using linear algebra methods. The solution was implemented using the `scipy.interpolate.make_interp_spline` [1], which performs lower–upper (LU) decomposition and uses the factored matrices to solve for c.

Once the spline was defined, it was used to shape the enrichment function within the enrichment domain. The number of points evaluated was equal to the number of pressure fluctuation points within this domain. The values for the control points $(x_1, y_1), (x_2, y_2)$ were obtained through non-linear least squares fitting of the spline to the pressure fluctuations in the enrichment domain, as described in a subsequent section. The B-spline local sawtooth function for Section 1 of DLR-F22 model under the Schroeder maneuver is illustrated in Figure 4.8:

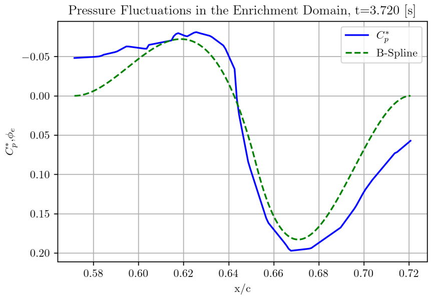


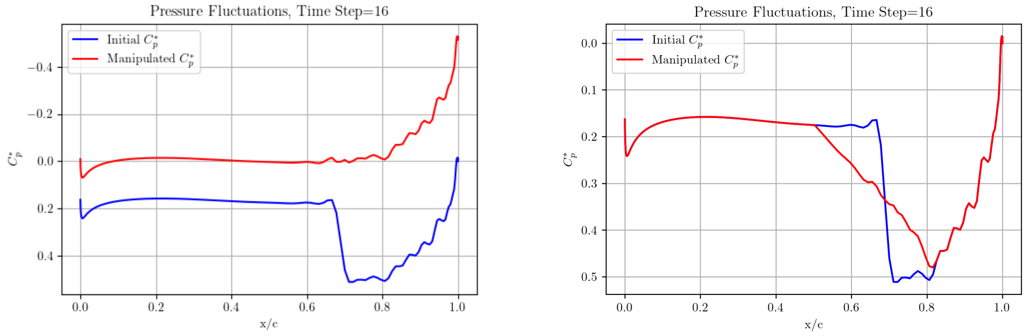
Figure 4.8: B-Spline enrichment function, Section 1 of the DLR-F22 Model, Schroeder maneuver.

[1] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.make_interp_spline.html`.

4.3.3. Step vs Local Sawtooth Function

The local sawtooth function was preferred over the step function for the following reasons: First of all, the step function was global in space, while the idea behind the sawtooth enrichment was only to smooth out the pressure discontinuities within the enrichment domain without affecting the pressure distribution outside. This distinction is visible in Figure 4.9 and Figure 4.10. The step function's global impact makes the pressure distribution highly sensitive to enrichment parameters. Even minor errors in shock parameter prediction can significantly affect the overall pressure distribution, leading to substantial errors.

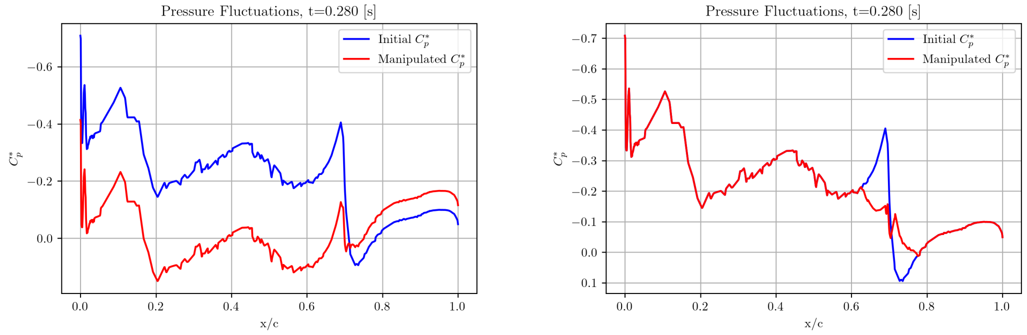
In simulations that account for viscosity, the pressure discontinuities appear less sharp. Consequently, the step function must have high-order smoothness properties to approximate the pressure step in these cases. Therefore, step function resulted in a much wider range of possible enrichment parameter values compared to the sawtooth enrichment function. Local sawtooth function parameters were confined to smaller ranges, such as control point locations between 0 and 1 and amplitudes within the minimum and maximum pressure range (typically between -2 and 2). In contrast, step function parameters can reach values of several hundred, as demonstrated in [28]. Thus, the extended parameter bounds of the step function hinder the fitting to the shock, leading to increased computational time.



(a) Step Function

(b) Local sawtooth Function

Figure 4.9: Effect of the two different enrichment bases on pressure fluctuations, for NACA 0012.



(a) Step Function

(b) Local sawtooth Function

Figure 4.10: Effect of the two different enrichment bases on pressure fluctuations in Section 1 of the DLR-F22 Model, Schroeder maneuver.

4.4. Fitting Enrichment Function on Sections

The primary motivation for selecting the local sawtooth function over the step functions was to control the dataset locally within the enrichment domain. Therefore, the enrichment mode must be zero outside the enrichment domain and at the domain boundaries. The following process was employed to satisfy this requirement. First, the target function $t(x, t)$ was defined as the straight line connecting the two pressure points at the boundaries of the enrichment domain, see Figure 4.12. Linear interpolation between these two points was applied to derive the target function as follows:

$$t(x_i, t) = C_p^*(x_L, t) + \frac{C_p^*(x_R, t) - C_p^*(x_L, t)}{x_R - x_L}(x_i - x_L) \quad \text{for } i = 1, 2, ..., m \tag{4.19}$$

where x_i is the chordwise location of pressure value $C_p^*(x_i, t)$, m is the number of data points in the enrichment domain, and x_L and x_R denote the left and right boundaries of the enrichment domain, respectively.

The target dataset was subtracted from the Full-Order Model (FOM) within the enrichment domain, resulting in the test data $T(x, t)$. The test data are given by Equation 4.20.

$$T(x_i, t) = C_p^*(x_i, t) - t(x_i, t), \quad \text{for } i = 1, 2, ..., m \tag{4.20}$$

As a result, a test dataset within the enrichment domain that satisfies the requirement of zero boundary conditions has been obtained. Consequently, the test function exhibited a distribution similar to the pressure fluctuations within the enrichment domain, but it reached zero values at the boundaries of this domain, as shown in Figure 4.12. The enrichment function was then fitted to the test data, to derive the optimum parameters of the enrichment basis at the current timestep. This approach ensures that the enrichment basis smooths out the shock while leaving the dataset outside the enrichment domain unaffected.

Non-linear least squares were employed to determine the optimal parameters that minimize the discrepancy between the enrichment model and the test dataset. Specifically, non-linear least squares were used to fit the enrichment model, $\phi_e(x, p(t))$, to the test data, $T(x, t)$, within the enrichment domain. In non-linear least squares fitting, the objective is to minimize the sum of the squares of the residuals, $r_i(p)$, which are the differences between the test data point $T(x_i, t)$ and the enrichment predictions, similar to the description of Heath [32].

$$r_i(p) = T(x_i, t) - \phi_e(x_i, p) \tag{4.21}$$

where $p(t)$ is the enrichment parameters vector that minimizes the following sum:

$$\min_p S(p) = \sum_{i=1}^{n} r_i^2(p) = \sum_{i=1}^{n} \left(T(x_i, t) - \phi_e(x_i, p)\right)^2, \quad \text{where } L \leq p \leq U \tag{4.22}$$

$$\min_p S(p) = \frac{1}{2} r^T(p) r(p), \quad \text{where } L \leq p \leq U \tag{4.23}$$

where $r(p)$ is a vector containing the n residual functions [37] and L, U are the vectors containing the lower and upper bounds of the enrichment parameters $p(t)$ correspondingly.
The gradient vector and Hessian matrix of S are given by:

$$\nabla S(p) = J^T(p) r(p) \tag{4.24}$$

and

$$H_S(p) = J^T(p) J(p) + \sum_{i=1}^{n} r_i(p) H_{r_i}(p) \tag{4.25}$$

where $J(p)$ is the Jacobian matrix of $r(p)$, and $H_{r_i}(p)$ denotes the Hessian matrix of the component function $r_i(p)$.

The parameters for the enrichment mode, as described in Equation 4.12 or Equation 4.13, were constrained by the problem's limitations. The normalization of chordwise coordinates confines the chordwise location of the control points to a range between 0 and 1. Additionally, the amplitude of the control points (i.e., the y-values) was constrained by the fluctuations in the pressure coefficient, which typically range from -2 to 2. Consequently, these limitation confines the optimal parameters to a narrow range. To solve the non-linear least squares problem expressed by Equation 4.23, the Trust Region Reflective (TRF) algorithm was selected. TRF works by iteratively improving the parameter estimates p within a region around the current estimate, called the "trust region." [83]. At each iteration, the algorithm defines a region around the current estimate p_k where it "trusts" the model's linear approximation to be accurate. The size of this region is controlled by the trust region radius Δ_k [83]. This was illustrated nicely by Hofer et al. [37], as follows:

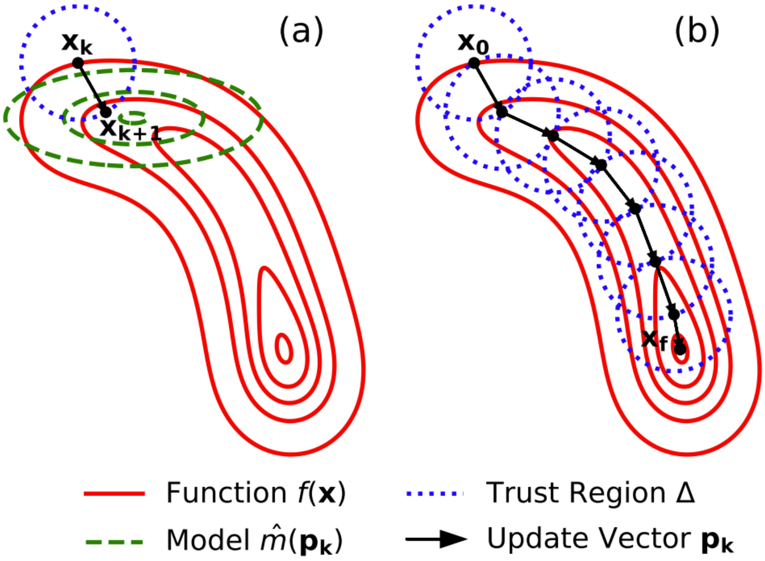


Figure 4.11: Visualization of the trust region method algorithm [37].

For non-linear functions numerical methods must be used to solve the minimization problem of Equation 4.23 [37]. These methods generally involve iterative steps in which the parameter vector p is updated with a parameter update vector p_k. The non-linear function $S(p + p_k)$ was approximated by a quadratic model within the trust region:

$$m_k(p_k) = S(p) + \nabla S(p)^T p_k + \frac{1}{2} p_k^T B p_k \tag{4.26}$$

where $m_k(p_k) \approx S(p+p_k)$, is the step from the current parameter estimate, and B is an approximation of the Hessian matrix Equation 4.25, given by:

$$B = J^T(p)J(p) \tag{4.27}$$

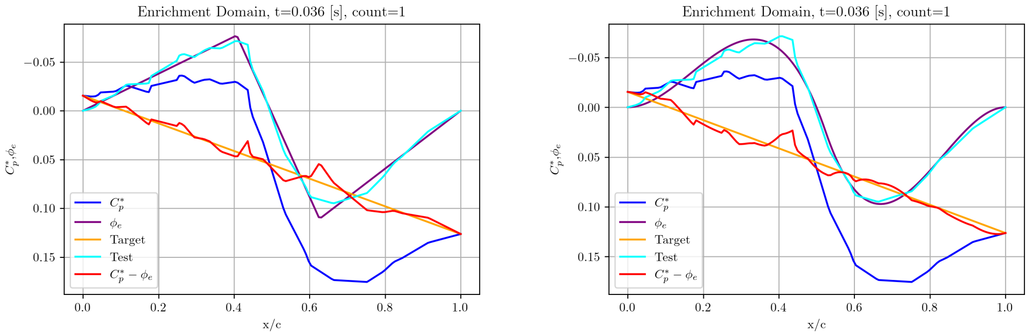
The step p_k was generated by finding an approximate solution of the subproblem:

$$\min_{p_k} m_k(p_k), \quad \text{subject to} \quad \|Dp_k\|_2 \leq \Delta_k, \quad \text{where} \;\; L \leq p + p_k \leq U \tag{4.28}$$

where D is a positive diagonal matrix. This subproblem was solved by an exact method very similar to the one described by More in [56].

The solution Equation 4.23 was implemented utilizing the optimization `scipy.optimize.curve_fit` [2]. TRF algorithm effectively solves the non-linear least squares problem by iteratively refining the enrichment parameter predictions within a trust region. The algorithm handles enrichment parameter bounds through reflection. This characteristic makes TRF particularly suited for non-linear problems with constraints. Furthermore, the TRF algorithm is robust because it is less sensitive to the initial parameter guess and more efficient, as it can often converge quickly due to the small radius of the problem. The fitted enrichment modes to the test data $T(x, t)$ within the enrichment domain, are illustrated in Figure 4.12 for DLR-F22 model:

[2] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html`.



(a) Piecewise linear sawtooth enrichment

(b) B-spline sawtooth enrichment

Figure 4.12: Enrichment function fitting for Section 1 of the DLR-F22 model, Schroeder maneuver.

The computational cost of the optimization process was minimal compared to the overall ePOD-LSTM training cost, as shown in Table 4.1. For the DLR-F22 model, a piecewise linear sawtooth enrichment function was selected, significantly reducing the computational cost relative to the B-Spline sawtooth enrichment function, while producing comparable results, as demonstrated in Figure 4.12.

Table 4.1: Computational cost of the fitting procedure in Section 1.

Test Case DLR-F22	**Sawtooth Enrichment Function**	**Computational cost [s]**
Schroeder Maneuver	Piecewise Linear	26
Schroeder Maneuver	B-Spline	142
PRBS Signal	Piecewise Linear	29
PRBS Signal	B-spline	156

4.5. Evaluation of ePOD

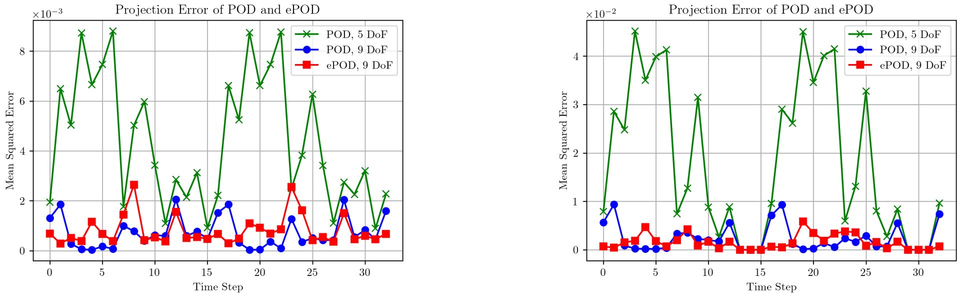
This paragraph evaluates the performance of the enriched Proper Orthogonal Decomposition (ePOD) compared to standard POD. The projection error served as the key performance indicator of the ePOD. The projection error, expressed in terms of Mean-Squared Error (MSE), was the error between the projection on the enriched reduced-order basis and the FOM. The projection for the ePOD is given by Equation 4.29

$$\epsilon_{\mathrm{p}} = \frac{1}{N_x}\sum_{i=1}^{N_x} ||C_p^*(x,t) - \sum_{k=1}^{r} a_k(t)\phi_k(x)^T - \sum_{s=1}^{i}\phi_{e_s}(x,p(t))||^2 \qquad (4.29)$$

As outlined in Equation Equation 4.29, the projection error was evaluated globally in space and locally in time. The proposed ePOD was developed to improve the representation of shock discontinuities in the pressure distribution. Therefore, an alternative assessment of the ePOD method's effectiveness, compared to standard POD, was the local projection error within the enrichment domain. The latter is of more interest in applications where the quality of the shock representation is important.

4.5.1. NACA 0012

In this section of the report, the results of the proposed methodology for the inviscid flow around the NACA 0012 airfoil are presented. The results of the ePOD were compared with those of the standard POD using the same number of spatial modes and degrees of freedom (DoF). The selection of the number of DoF was based on achieving a maximum time-averaged projection error of $1 \cdot 10^{-3}$. Specifically, the performance of ePOD was evaluated, using 5 spatial modes and 1 enrichment mode (4 DoF per enrichment mode), was compared to that of standard POD. Thus, in the latter case, both methods have nine degrees of freedom.



(a) Projection error

(b) Projection error in the enrichment domain

Figure 4.13: Projection error for NACA 0012.

The time-averaged projection error of ePOD was $8.09 \cdot 10^{-4}$, 11.37% higher than that of standard POD with the same DoF. However, in the enrichment domain, ePOD demonstrated a significant advantage, with an time-averaged local projection error of $1.57 \cdot 10^{-3}$. This represents a 32.32% reduction compared to standard POD with the same DoF, which resulted in an time-averaged projection error of $2.32 \cdot 10^{-3}$. Moreover, ePOD's projection error was substantially lower by 81.65% than standard POD with five spatial modes. Within the enrichment domain, the local error for ePOD was 91.37% lower.

Although the standard POD method, using the same number of DoF, exhibited a comparable order of projection error to the ePOD, it consistently failed to approximate shock discontinuities. In some instances, the global projection error was slightly lower due to the following reasons. First, the increased number of spatial modes employed by the standard POD method, more effectively approximated small pressure fluctuations outside the enrichment domain. Additionally, in certain time instances, the shock discontinuity observed at the trailing edge, see Figure 4.14, posed challenges for the proposed method. Figure 4.14 illustrates the pressure distribution at the timestep where the largest discrepancy in projection error between the ePOD and POD methods was observed. Consequently, while the standard POD with an increased number of truncated modes achieved a similar order of global projection error in certain cases, it remained incapable of approximating shock discontinuities in the majority of time instances, as illustrated in Figure 4.15.

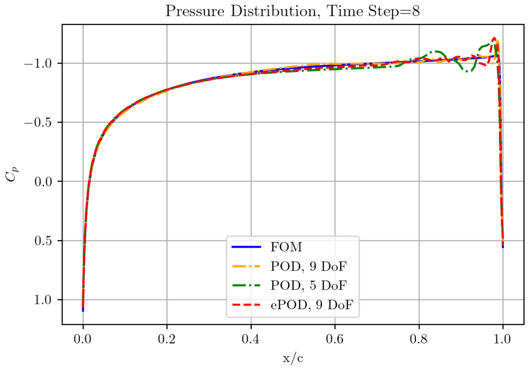
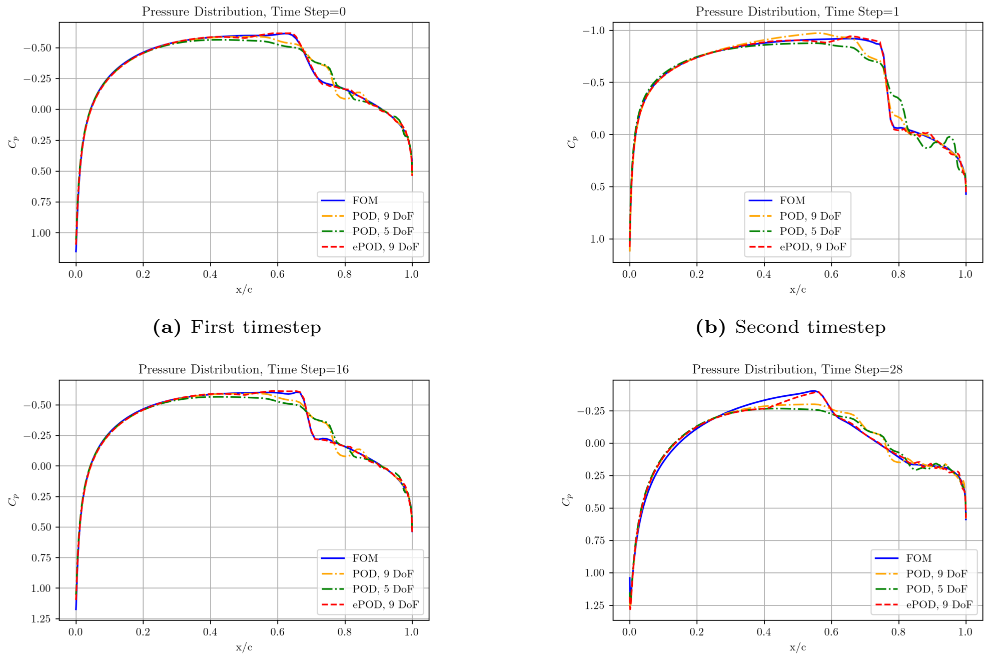


Figure 4.14: Comparison of full-order and reduced-order pressure coefficients for NACA 0012 airfoil at the eighth timestep, using POD and ePOD methods.

The reconstructed pressure distribution was plotted alongside the Full-Order Model (FOM) results and the standard POD, as shown in the following Figure 4.15:



(a) First timestep

(b) Second timestep

(c) Seventeenth timestep

(d) Twenty-ninth timestep

Figure 4.15: Comparison of full-order and reduced-order pressure coefficients for NACA 0012, utilizing POD and ePOD methods.

The following tables summarize the time-averaged projection errors for the NACA 0012 airfoil:

Table 4.2: Summary of time-averaged projection error for NACA 0012.

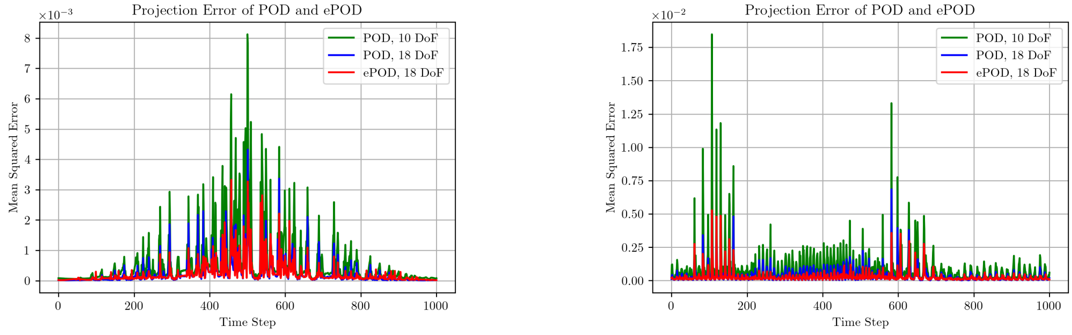
Test Case	Order Reduction Technique	Degrees of Freedom	Projection Error	Projection Error (Enrichment Domain)
NACA 0012	POD	5	$4.41 \cdot 10^{-3}$	$1.82 \cdot 10^{-2}$
NACA 0012	POD	9	$7.17 \cdot 10^{-4}$	$2.32 \cdot 10^{-3}$
NACA 0012	ePOD	9	$8.09 \cdot 10^{-4}$	$1.57 \cdot 10^{-3}$

The ePOD time-averaged projection error was significantly lower within the enrichment domain. This observation shows the effectiveness of the proposed approach in accurately approximating shock discontinuities. While both methods exhibit similar global error orders, ePOD outperforms standard POD in regions where shock waves appear.

4.5.2. DLR-F22 Model

As previously discussed, the initial phase of the proposed ROM involved constructing an enriched reduced-order basis using the enriched Proper Orthogonal Decomposition (ePOD) order reduction technique. To demonstrate the benefits of this approach over the standard method in modeling transonic flows in real-life problems, we evaluated the projection error for the sections of the DLR-F22 model. The dimensions for the enriched and standard reduced-order basis was selected such that the time-average projection error remained below $1 \cdot 10^{-3}$. Specifically, we compared the projection error of the ePOD to the corresponding error of the standard POD method, using an equal number of spatial modes and DoF. In detail, the ePOD case utilized ten spatial modes and two enrichment modes (8 DoF). For the standard method, ten and eighteen spatial modes were utilized. Designed to address pressure discontinuities within the enrichment domain, the proposed approach performed remarkably well in this region.

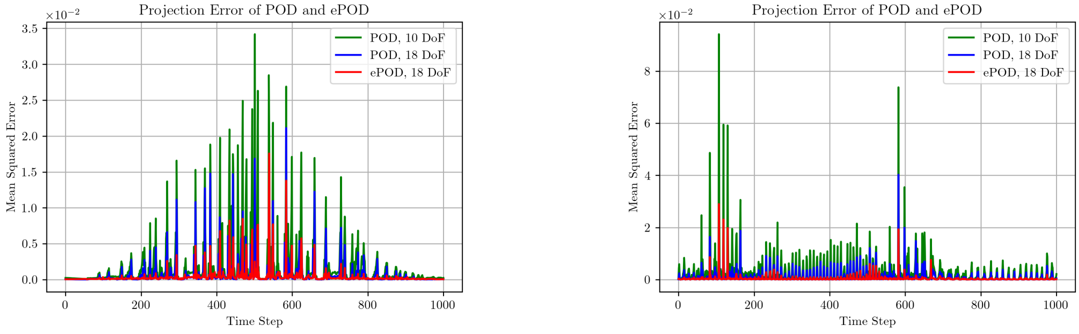
To illustrate its performance, we also assessed the local projection error evaluated within the enrichment domain.



(a) PRBS signal

(b) Schroeder maneuver

Figure 4.16: Projection error in Section 1 of the DLR-F22 model.



(a) PRBS signal

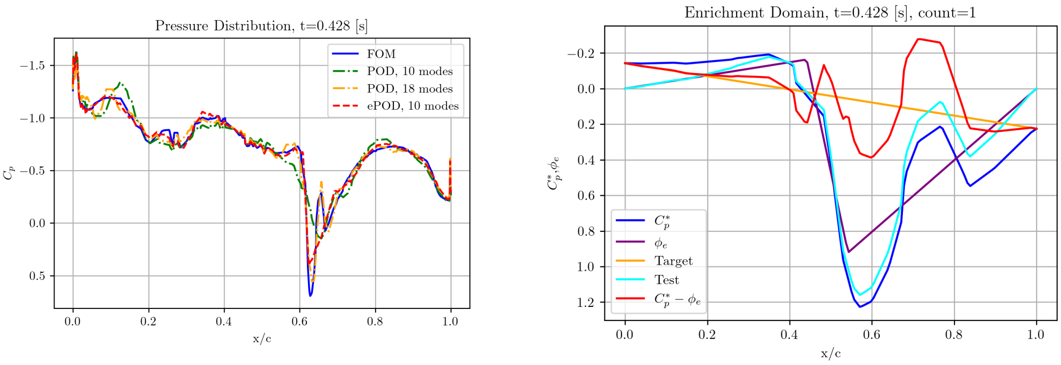
(b) Schroeder maneuver

Figure 4.17: Projection error in the enrichment domain, Section 1 of the DLR-F22 model.

The projection error for the ePOD method was consistently lower than that of the standard POD with the same DoF for the PRBS signal. For Section 1, the time-averaged projection error of the ePOD was $2.25 \cdot 10^{-4}$, whereas for the standard POD with the same number of DoF was 7% higher, with a time-averaged value of $2.42 \cdot 10^{-4}$. The time-averaged local projection error in the enrichment domain for the ePOD was $4.28 \cdot 10^{-4}$, and significantly higher by 40.22% for standard POD with the same DoF.

Furthermore, in the case of the Schroeder maneuver in Section 1, the global projection error of the ePOD was generally lower than that of the standard POD with the same number of DoF. On average, for ePOD it was $2.78 \cdot 10^{-4}$ and increased by 17.75% for POD reaching a time-averaged value of $3.38 \cdot 10^{-4}$. The time-averaged projection error in the enrichment of standard POD was significantly higher by 58.64% relative to the ePOD with the same DoF.

The ePOD outperforms the standard method with the same degrees of freedom, except for three specific instances in time. Further investigation into these time steps revealed that the fitting error was significant in these cases, caused by high oscillations following the shock discontinuity, which resulted in underfitting of the previously described fitting methodology, as observed in Figure 4.18:



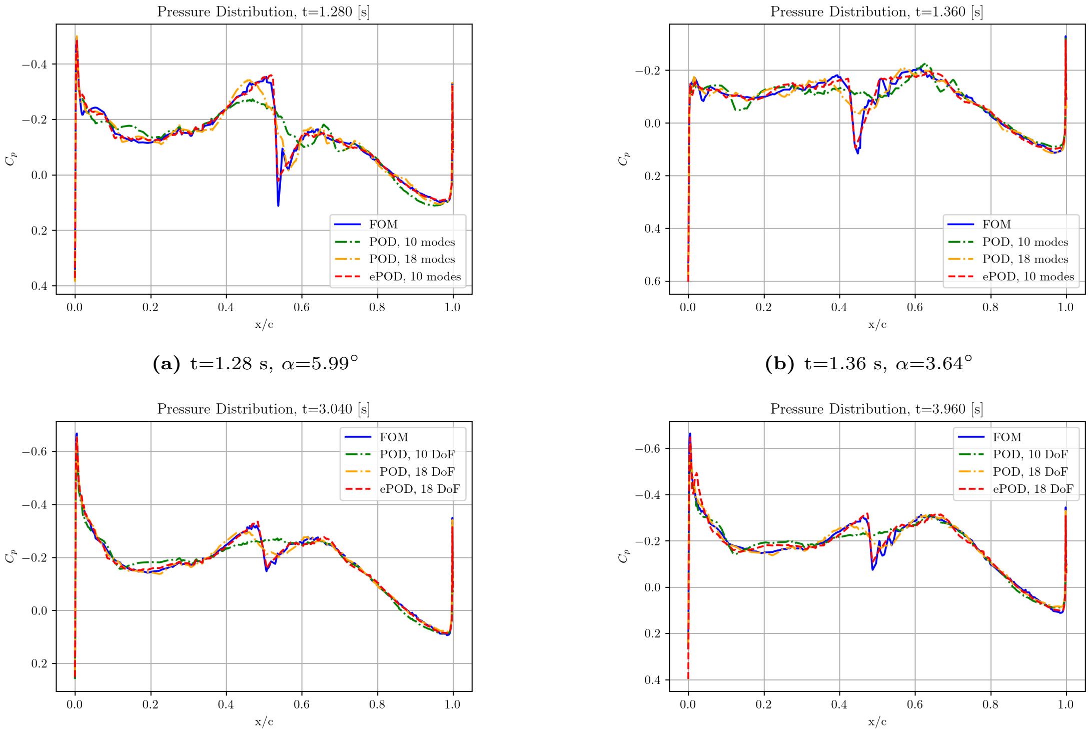
(a) Pressure coefficient reconstruction

(b) Enrichment domain of the main shock

Figure 4.18: Comparison between the ePOD and POD methods, for the Schroeder maneuver in Section 1 at t=0.428 [s].

As it is observed in Figure 4.18, the enrichment mode (purple line) was not fitted correctly to the test function due to the oscillations in the pressure fluctuations (blue line) between 0.65-0.85 of the enrichment domain.

Furthermore, the projection error was computed by averaging the distance between the reduced-order basis and the FOM at each time step. Consequently, minor pressure fluctuations within the dataset contribute to this error. The standard POD had an increased number of spatial modes, better capturing these small pressure fluctuations in regions of the dataset outside the enrichment domain, where no strong shock occurs. Therefore, the projection error order might be comparable in some cases, but the standard POD completely ignored the shock discontinuities in most scenarios. This phenomenon can be observed by comparing the global and local projection error, as well as the pressure distributions for selected time instances in Figure A.5 and Figure 4.19. It becomes evident that the proposed method excels in the shock discontinuity region, outperforming the standard method. Therefore, enabling more efficient use of the ROM for predicting pressure distributions in the transonic flow regime.



(a) t=1.28 s, α=5.99°

(b) t=1.36 s, α=3.64°

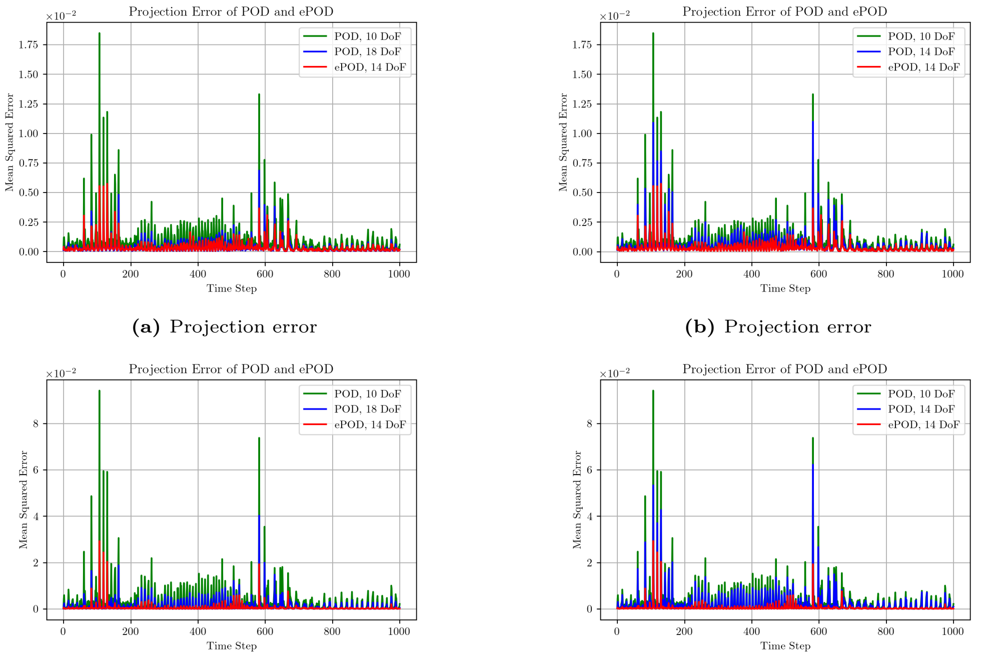
(c) t=3.04 s, α=4.41°

(d) t=3.96 s, α=4.39°

Figure 4.19: Reconstruction of the pressure distribution in Section 1 using ePOD and POD methods, Schroeder maneuver.

Single Enrichment Mode

The ePOD method incorporates the number of shock discontinuities as an input parameter, derived from observations in the dataset. Each shock was represented by an enrichment mode, making the number of shocks a tunable parameter for model optimization. The value of this parameter was determined by analyzing the dataset and identifying the dominant shock systems. A closer examination revealed, in the case of the Schroeder maneuver, one dominant shock system. To investigate the impact of the secondary enrichment mode on the enriched latent space, an enriched reduced-order basis was constructed using only one enrichment mode. The dominant shock was determined by identifying the shock discontinuity with the highest gradient.



(a) Projection error

(b) Projection error

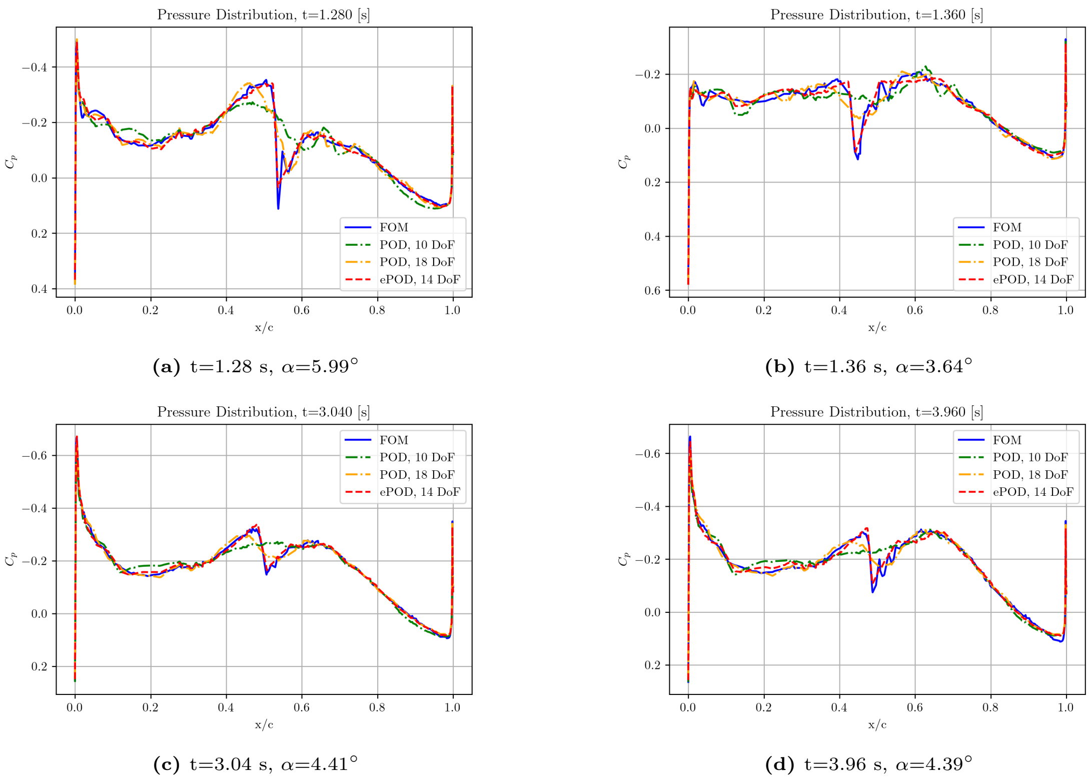
(c) Projection error in the enrichment domain

(d) Projection error in the enrichment domain

Figure 4.20: Comparison of the projection error between ePOD with 10 spatial and 1 enrichment mode and POD with 14 and 18 truncated modes.

Introducing a secondary enrichment mode increased the dimensionality of the enriched reduced-order basis without significantly reducing the projection error. For instance, in the case of the PRBS signal, the ePOD method with one enrichment mode resulted in a 13.79% higher projection error compared to the ePOD method with two enrichment modes. However, the single enrichment mode basis significantly reduced the number of degrees of freedom by almost 30% (four fewer enrichment parameters). This observation was important as it decreased the dimensions of the enriched reduced-order basis, and thus reduced the number of parameters the neural network predicted. Consequently, the computational cost of the ePOD-LSTM ROM was reduced, as discussed in Chapter 5, while maintaining satisfactory accuracy.

In the single shock configuration, for PRBS signal in Section 1, ePOD's time-averaged global projection error was 26.27% lower than POD with the same DoF. Within the enrichment domain, this percentage increased to 59.91%, underscoring the method's performance in shock discontinuity region. Also, for Schroeder maneuver, ePOD exhibited a 35.27% lower time-averaged global projection error and 72.82% less time-averaged local error than POD with an equal number of DoF. Furthermore, the ePOD with a single shock experienced a 57.27% lower time-averaged projection error than the standard POD with almost 30% more DoF. This result confirmed that ePOD outperforms standard POD, resulting in significantly lower projection errors in regions where shock waves appear even with less DoF. Thus, ePOD effectively integrating shock discontinuities into the reduced-order basis. These conclusions were supported by the reconstructed pressure distribution for the ePOD method with one shock, as shown in Figure 4.21.



(a) t=1.28 s, α=5.99°

(b) t=1.36 s, α=3.64°

(c) t=3.04 s, α=4.41°

(d) t=3.96 s, α=4.39°

Figure 4.21: Reconstruction of the pressure distribution in Section 1 using the ePOD method with one enrichment mode, Schroeder maneuver.

Discussion

The following tables summarize the time-averaged projection errors for all the cases of DLR-F22 model examined.

Table 4.3: Summary of Time-Averaged Projection Error for Section 1 of DLR-F22 model.

Test Case DLR-F22	**Section**	**Order Reduction Technique**	**Degrees of Freedom**	**Projection Error**	**Projection Error (Enrichment Domain)**
PRBS Signal	1	POD	14	$3.54 \cdot 10^{-4}$	$1.13 \cdot 10^{-3}$
PRBS Signal	1	POD	18	$2.42 \cdot 10^{-4}$	$7.16 \cdot 10^{-4}$
PRBS Signal	1	ePOD	14	$2.61 \cdot 10^{-4}$	$4.53 \cdot 10^{-4}$
PRBS Signal	1	ePOD	18	$2.25 \cdot 10^{-4}$	$4.28 \cdot 10^{-4}$
Schroeder Maneuver	1	POD	14	$5.16 \cdot 10^{-4}$	$1.74 \cdot 10^{-3}$
Schroeder Maneuver	1	POD	18	$3.38 \cdot 10^{-4}$	$1.03 \cdot 10^{-3}$
Schroeder Maneuver	1	ePOD	14	$3.34 \cdot 10^{-4}$	$4.73 \cdot 10^{-4}$
Schroeder Maneuver	1	ePOD	18	$2.78 \cdot 10^{-4}$	$4.26 \cdot 10^{-4}$

Table 4.3 presents the time-averaged projection error for all the test cases in Section 1 of the DLR-F22 model. The proposed ePOD approach consistently outperformed the standard POD with the same or even a reduced number of DoF. Table 4.4 represents the results for the time-average projection error

in Section 2. Results appeared to be similar to Section 1. The ePOD outperformed the POD, even with the same or fewer degrees of freedom in the enrichment domain. In this section, less strong pressure discontinuities were observed. As a result, the global time-averaged projection error for the standard POD was slightly lower than that ePOD with 18 DoF. This observation underscores the significance of the number of enrichment modes. Consequently, careful selection of parameters for the ePOD is crucial, as multiple enrichment modes can increase the method's complexity without corresponding improvements in the accuracy of the reduced-order basis.

Table 4.4: Summary of time-averaged projection error for Section 2 of DLR-F22 model.

Test Case DLR-F22	**Section**	**Order Reduction Technique**	**Degrees of Freedom**	**Projection Error**	**Projection Error (Enrichment Domain)**
PRBS Signal	2	POD	14	$4.38 \cdot 10^{-4}$	$1.43 \cdot 10^{-3}$
PRBS Signal	2	POD	18	$2.64 \cdot 10^{-4}$	$7.73 \cdot 10^{-4}$
PRBS Signal	2	ePOD	14	$3.83 \cdot 10^{-4}$	$5.74 \cdot 10^{-4}$
PRBS Signal	2	ePOD	18	$3.33 \cdot 10^{-4}$	$4.94 \cdot 10^{-4}$
Schroeder Maneuver	2	POD	14	$5.84 \cdot 10^{-4}$	$1.85 \cdot 10^{-3}$
Schroeder Maneuver	2	POD	18	$3.79 \cdot 10^{-4}$	$1.31 \cdot 10^{-3}$
Schroeder Maneuver	2	ePOD	14	$4.82 \cdot 10^{-4}$	$6.25 \cdot 10^{-4}$
Schroeder Maneuver	2	ePOD	18	$4.14 \cdot 10^{-4}$	$5.69 \cdot 10^{-4}$

In conclusion, ePOD offers a significant advantage over standard POD for constructing the reduced-order basis in transonic flows. The ePOD method effectively captured shock discontinuities and reduced the DoF, enabling more efficient representations of pressure distribution in transonic flows.

5 Machine Learning

5.1. Long Short-Term Memory Neural Network

Recurrent Neural Networks (RNNs) [67] improved feed-forward neural networks by incorporating the output of adjacent time steps, introducing a temporal dimension to the model [47]. The connections between neighboring time steps, known as recurrent edges, enable RNNs to process sequential data (e.g., time series, temporal coefficients). The network's weights are updated using backpropagation through time [11]. The Long Short-Term Memory (LSTM) network was developed to address challenges faced by standard RNNs, such as vanishing or exploding gradients and the limited ability to transmit information over long sequences. Previous studies conducted at NLR by Catalani [12] and Bourier [10] have demonstrated that LSTM neural networks outperform other ANN and regression models in terms of accuracy when predicting time coefficients of spatial POD modes. Bourier [10] conducted a sensitivity analysis to determine the optimal hyperparameters for the baseline model. The outcomes of this analysis, combined with the findings from Catalani's study [12], were used to define the optimal hyperparameters for LSTM network. The LSTM network trained to predict the temporal evolution of the enriched reduced-order basis across different sections of the DLR-F22 model. For further details on the derivation of these hyperparameter values, please refer to the respective thesis reports. The key hyperparameters for the LSTM network are presented in Table 5.1.

The main difference between the previous neural networks parameters and the current ones was found in the loss function. The custom loss function was a modification of the function proposed by Catalani in [12], were the enrichment basis was integrated. Specifically, the loss function used in training the LSTM neural network was the Mean-Square Error (MSE) between the projected and predicted pressure distributions. The predicted time coefficients $\left\{a_k^{NN}\right\}_{n=1}^{r}$ were multiplied by the spatial POD modes, and the predicted parameters $\left\{p^{NN}\right\}_{s=1}^{i}$ were introduced to the enrichment modes of Equation 4.12. Consequently, the predicted pressure distribution was reconstructed according to Equation 4.2. It was

Table 5.1: LSTM neural network hyperparameters.

Hyperparameter	**Value**
Number of LSTM layers	2
Number of LSTM units	64
Number of dense layers	1
Number of dense units	128
Number of time-steps	10
Batch size	32
Drop-out rate	0.2
Model optimization method	ADAM
Model loss function	Custom

then compared to the ePOD expansion of the true targets $\{a_k\}_{n=1}^{r}$ and $\{p\}_{s=1}^{i}$, as shown in Equation 5.1

$$\mathscr{L} = \frac{1}{N_x}\sum_{i=1}^{N_x} || \sum_{k=1}^{r} a_k(t)\phi_k(x)^T + \sum_{s=1}^{i} \phi_{e_s}(x, p(t)) - \sum_{k=1}^{r} a_k^{NN}(t)\phi_k(x)^T - \sum_{s=1}^{i} \phi_{e_s}(x, p^{NN}(t))||^2 \tag{5.1}$$

$$\begin{aligned}\mathscr{L} = \frac{1}{N_x}\sum_{i=1}^{N_x} || \sum_{k=1}^{r} a_k(t)\phi_k(x)^T + \sum_{s=1}^{i} \phi_{e_s}(x, p(t)) - \sum_{k=1}^{r} a_k^{NN}(t)\phi_k(x)^T - \sum_{s=1}^{i} \phi_{e_s}(x, p^{NN}(t))||^2 + \\ +\beta|| \sum_{k=1}^{r} \left[a_k(t) - a_k^{NN}(t)\right] + p(t) - p^{NN}(t)||^2\end{aligned} \tag{5.2}$$

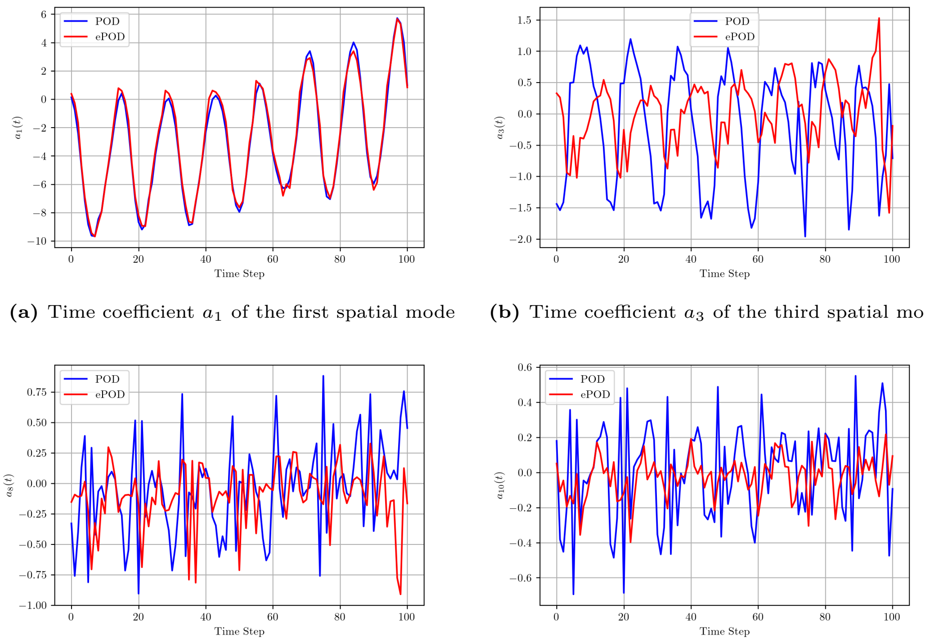
A weighted version of the loss function was employed, where β was a weight to incorporate a parentage of the predicted time coefficients in the loss.

Considering the dataset structure and the limited overlap between the available maneuvers, we implemented the following training strategy: The Schroeder maneuver, selected for training, was partitioned into three subsets: training, validation, and testing. Specifically, the first 700 points were used for training, the next 250 points for validation, and the final 50 points for testing. Once the enriched reduced-order basis was obtained, values of $a_k(t)$ and $p(t)$ were organized based in the three subsets. These must be matched to suitable input vector values. For the case under consideration, the time-variant input vector was defined to include the angle of attack α, the first and second time derivatives of α, the pitch rate q, and the first time derivative of q. This input vector and the $a_k(t)$ and $p(t)$ values form the corresponding subset were used for the training of the LSTM network.

5.2. Model Construction and Training

5.2.1. Time-Dependent Coefficients

The neural network predicted the time coefficients of the spatial modes and the enrichment parameters. Insight into these time-dependent values was crucial to understanding the complexity of the standard POD compared to the proposed ePOD methodology. Consequently, the time coefficients of the truncated spatial modes of POD and ePOD and the enrichment parameters were plotted as a function of time.

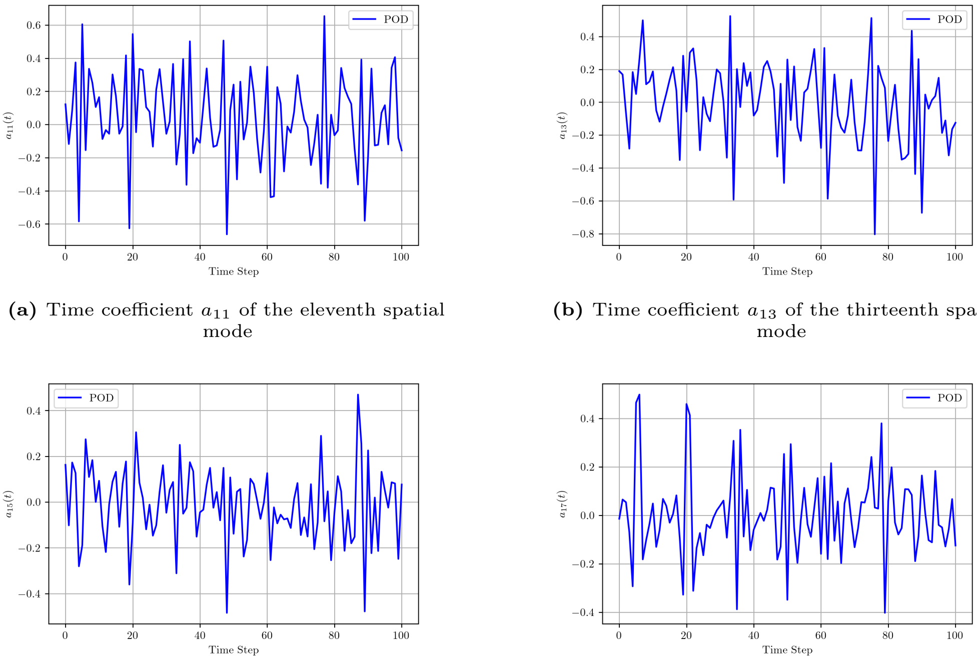


(a) Time coefficient a_1 of the first spatial mode **(b)** Time coefficient a_3 of the third spatial mode

(c) Time coefficient a_8 of the eighth spatial mode **(d)** Time coefficient a_{10} of the tenth spatial mode

Figure 5.1: Comparison of the time coefficients for the spatial modes between POD and ePOD methods over the final 100 timesteps of Schroeder maneuver in Section 1.

The time coefficients of the initial modes exhibited similar evolution over time. However, as the mode order increases, the deviation between the time coefficients of the POD and ePOD methods also increases. In other words, the complexity of time coefficients related to the POD increases with the order of the mode. To support this argument, Figure 5.2 illustrates the time coefficients of additional high-order spatial modes from the truncated POD basis with 18 spatial modes.



(a) Time coefficient a_{11} of the eleventh spatial mode

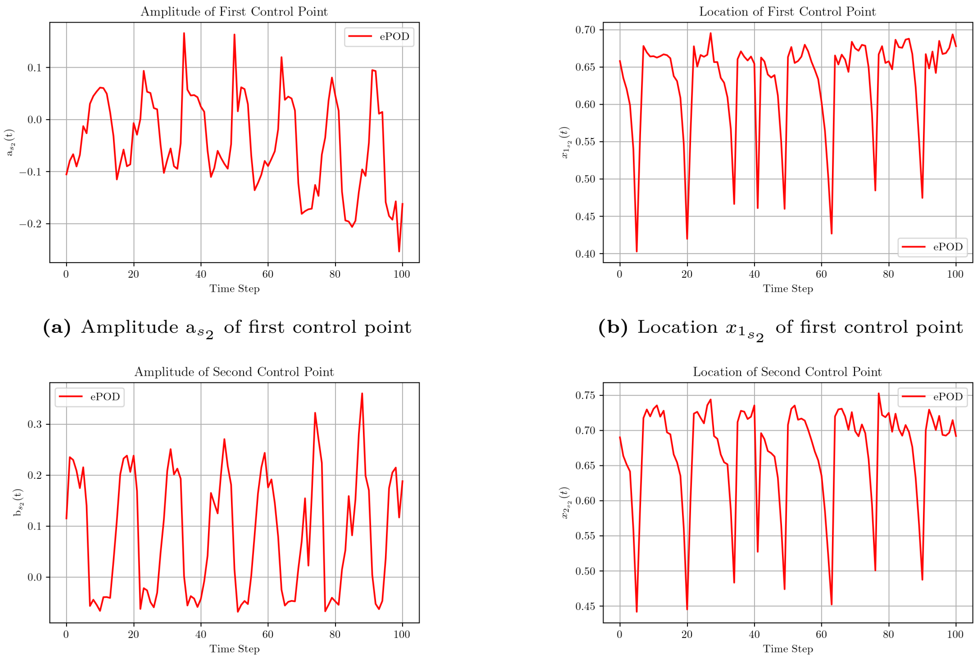
(b) Time coefficient a_{13} of the thirteenth spatial mode

(c) Time coefficient a_{15} of the fifteenth spatial mode

(d) Time coefficient a_{17} of the seventeenth spatial mode

Figure 5.2: Time coefficients of four additional spatial modes of standard POD, over the last 100 timesteps of Schroeder maneuver in Section 1.

The time coefficients of Figure 5.2 for standard POD were compared to the enrichment parameters for the dominant shock discontinuity in the ePOD approach, which are represented in Figure 5.3.



(a) Amplitude a_{s_2} of first control point

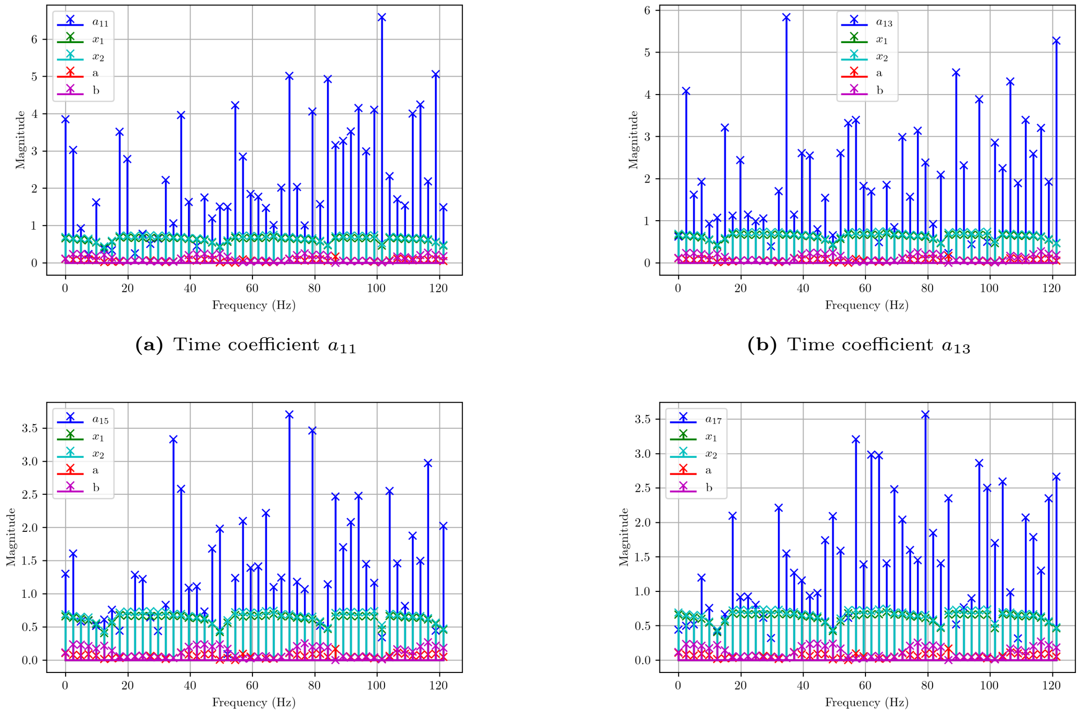
(b) Location $x1_{s_2}$ of first control point

(c) Amplitude b_{s_2} of second control point

(d) Location $x2_{s_2}$ of second control point

Figure 5.3: Enrichment parameters of the primary shock discontinuity, over the last 100 timesteps of Schroeder maneuver in Section 1.

To further analyze the relation between these parameters, the Fast Fourier Transform (FFT) was performed. Representing time-related values in the frequency domain revealed important trends in their complexity. There was no correspondence between the order of the POD modes and the parameters of the enrichment modes. Hence, the time coefficients of the additional high-order spatial modes were compared to all the enrichment parameters of the dominant shock discontinuity in Section 1 of the DLR-F22 model under the Schroeder maneuver.



(a) Time coefficient a_{11}

(b) Time coefficient a_{13}

(c) Time coefficient a_{15}

(d) Time coefficient a_{17}

Figure 5.4: Comparison of the frequency content of time coefficients of high-order spatial POD modes and the corresponding enrichment parameters, for the last 100 timesteps of Schroeder maneuver in Section 1.

The FFT analysis of the time-dependent parameters revealed that, high-order spatial modes primarily contained frequencies with large magnitudes, which were irregularly spread across the frequency range. Conversely, the enrichment parameters were more evenly distributed across frequencies and characterized by lower magnitudes. Furthermore, the enrichment parameters were nearly periodic over time, as shown in Figure 5.3. Based on these two observations, we can conclude that the enrichment parameters were less complex, and likely easier to regress. These observations, were verified during the training stage of the neural network.

5.2.2. Training Stage

Figure 5.5 presents the weighted training and validation loss for the Schroeder maneuver in Section 1. The weighted version of the loss function was used, where 95% was the contribution of the prediction error between the predicted and actual reduced-order basis, and 5% from the mean square distance between the predicted and actual time coefficients. The ePOD-LSTM ROM with 18 DoF fitted the training data with sufficient accuracy, as evidenced by the reduction in training loss as the number of epochs increases. The training loss reduced to approximately $9 \cdot 10^{-4}$ after 1000 epochs and continued to decreasing, eventually reaching a minimum of about $4 \cdot 10^{-4}$. Moreover, the neural network performed with sufficient accuracy on the validation data. The validation loss reached a plateau of approximately $4 \cdot 10^{-3}$ after the first 500 epochs.

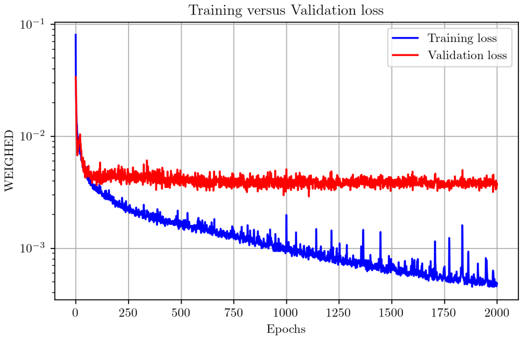
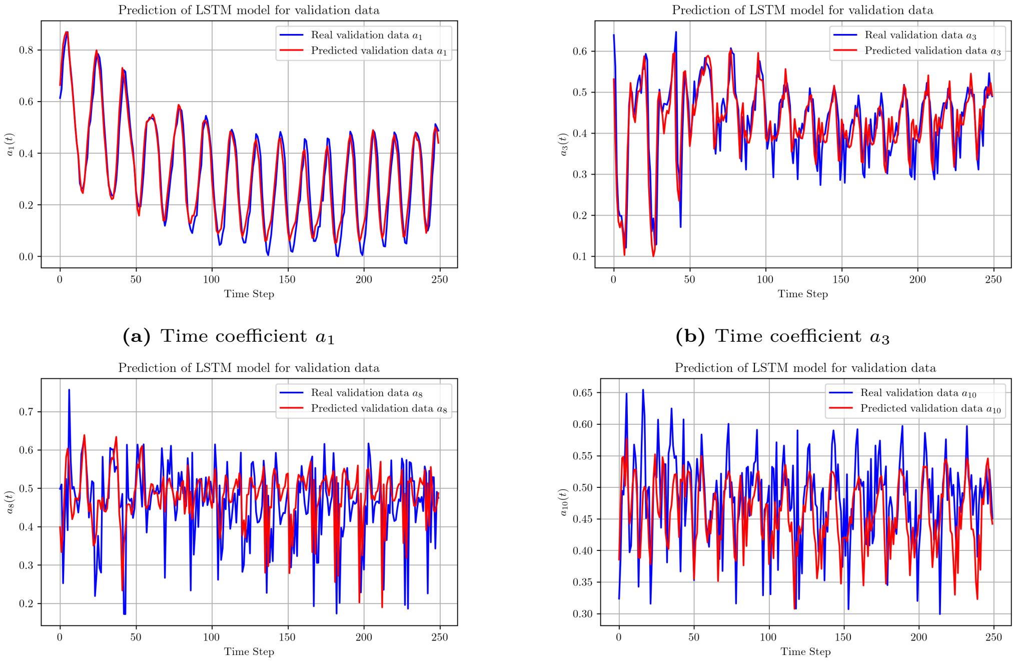


Figure 5.5: Training and validation loss.

The normalized time coefficients and enrichment parameters are shown in Figure 5.7. The training and validation datasets were normalized based on features minimum and maximum values. Normalization enhanced the training efficiency, stability, and performance of the neural network. The time coefficients of the first and last spatial modes and the parameters of the dominant shock discontinuity, predicted by the network were plotted against the actual validation data. By comparing the first and last time coefficients, it was verified that as the order of spatial modes increases, the complexity of its time coefficients increases. Consequently, it becomes more challenging for the neural network to fit these highly oscillating coefficients.



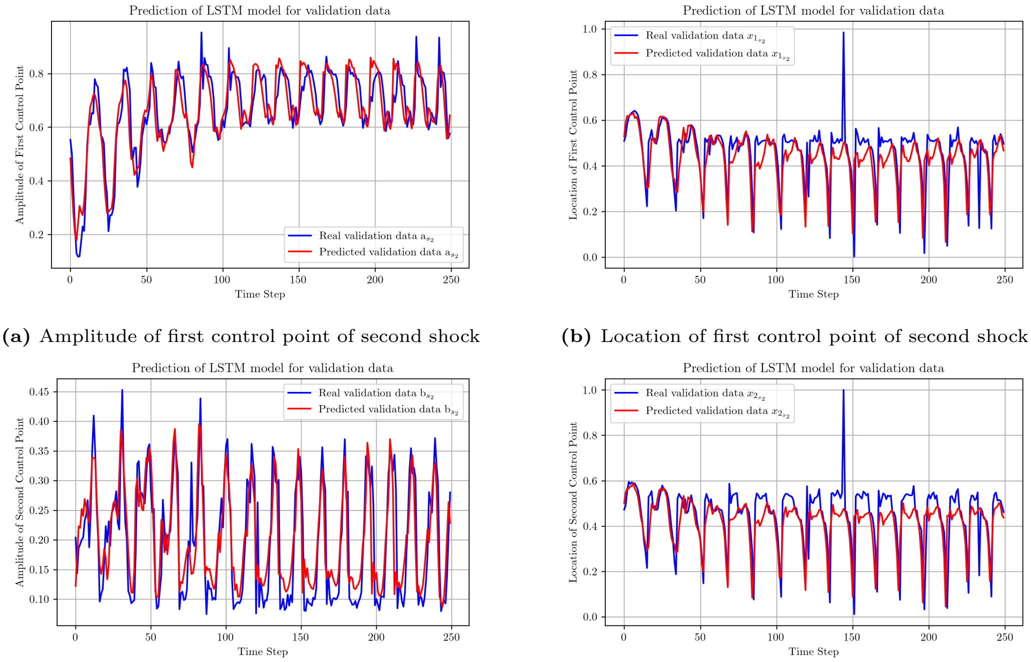
(a) Time coefficient a_1

(b) Time coefficient a_3

(c) Time coefficient a_8

(d) Time coefficient a_{10}

Figure 5.6: Validation performance of LSTM neural network in Section 1.



(a) Amplitude of first control point of second shock

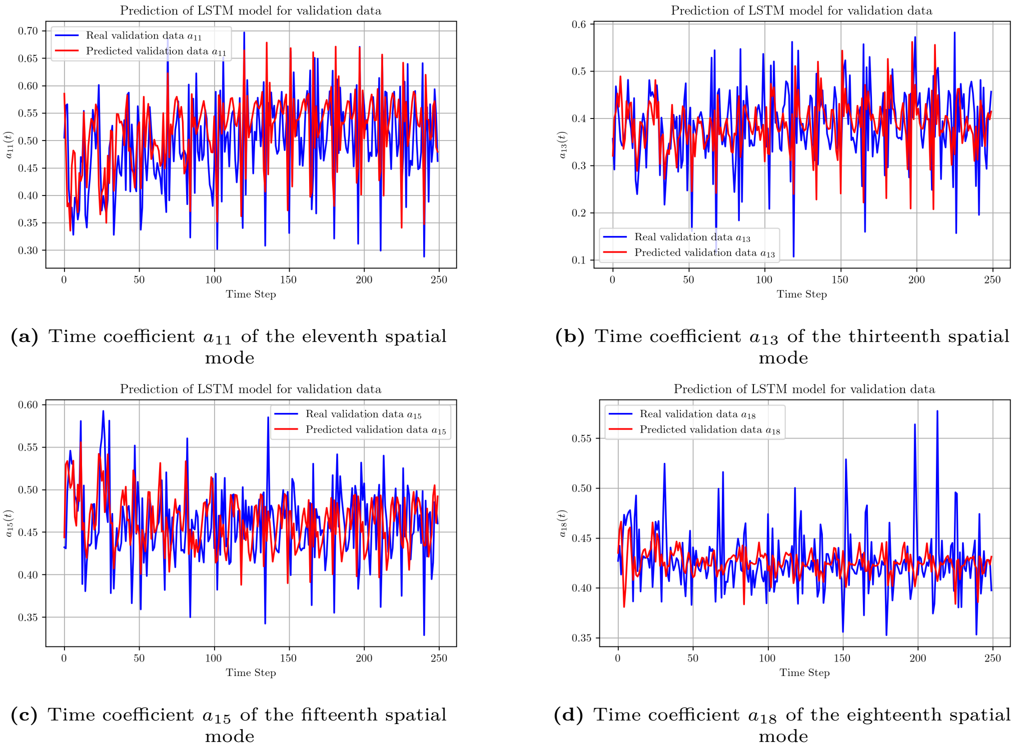
(b) Location of first control point of second shock

(c) Amplitude of second control point of second shock

(d) Location of second control point of second shock

Figure 5.7: Validation performance of LSTM neural network in Section 1.

To demonstrate the advantages of the proposed approach over the standard method, a corresponding ROM was employed using the standard reduced-order basis with eighteen spatial modes. The POD-LSTM ROM was constructed using the same LSTM neural network, with identical hyperparameters and the same datasets for training, validation, and testing. The time coefficients for the higher-order spatial modes, as predicted by the neural network during training and validation, are represented in Figure 5.8.



(a) Time coefficient a_{11} of the eleventh spatial mode

(b) Time coefficient a_{13} of the thirteenth spatial mode

(c) Time coefficient a_{15} of the fifteenth spatial mode

(d) Time coefficient a_{18} of the eighteenth spatial mode

Figure 5.8: Time coefficients of four high-order spatial modes of standard POD.

Figure 5.8 verifies that as the order of the spatial modes increases, modeling their temporal evolution becomes more challenging. In other words, the higher the mode order, the more difficult for the neural network to fit its time coefficient. For example, the neural network failed to accurately fit the validation dataset for the time coefficient associated with the eighteenth spatial mode. Incorporating more high-order spatial modes in the reduced-order basis increases the model's complexity. Conversely, the enrichment parameters were less complex and, as expected, better suited for regression. As a result, the proposed ePOD order reduction technique mitigates the complexity of time-dependent features of the reduced-order basis, thereby improving the efficiency of the regression process.

Training cost

The POD-LSTM ROMs with the same number of degrees of freedom as the ePOD-LSTM with single and double shock configurations were derived. The training costs for the four ROMs constructed for Section 1 utilizing the same hyperparameters are presented in the following table.

Table 5.2: Total training cost of the LSTM, using various reduced-order bases in Section 1.

Reduced-Order Model	Degrees of Freedom	Training cost [s]
POD-LSTM	14	1028
POD-LSTM	18	1069
ePOD-LSTM	14	914
ePOD-LSTM	18	1003

The reduced complexity of enrichment parameters compared to the time coefficients of extra spatial modes resulted in lower training costs. As shown in Table 5.2, the ePOD-LSTM model with fourteen degrees of freedom or one shock achieved the minimum training cost. This was 114 seconds faster than the POD-LSTM ROM with the same number of DoF and 155 seconds faster than the POD-LSTM ROM with four additional DoF. Furthermore, it was 89 seconds faster than ePOD-LSTM ROM with two enrichment modes. Interestingly, the ePOD-LSTM ROM with eighteen degrees of freedom required less training time than the POD-LSTM ROM with four fewer parameters, attributed to the lower complexity of its parameters. As expected, the POD-LSTM ROM with eighteen truncated spatial modes resulted in the highest computational cost.

5.3. Testing Stage

In this paragraph, the results from the testing stage of the Reduced-Order Models (ROMs) that was constructed are presented and analyzed. The predicted pressure distribution for the LSTM neural network was compared with both the actual reduced-order basis and the full-order pressure distributions. Three different types of errors were assessed for the testing dataset. The first error evaluated was the projection error defined by Equation 4.29 between the enriched reduced-order basis and the FOM. The second was the neural network time coefficients error, defined as the instantaneous error between the actual and the predicted enriched reduced-order basis. The neural network time coefficients error ϵ_{NN} is given by Equation 5.3.

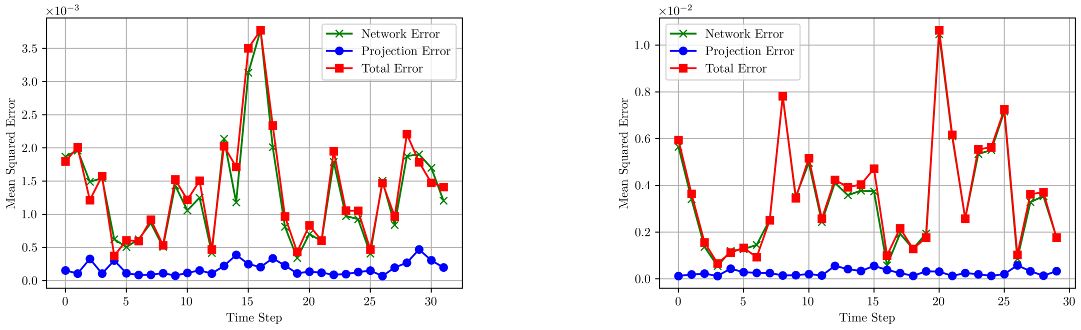
$$\epsilon_{\text{NN}} = \frac{1}{N_x}\sum_{i=1}^{N_x} \left|\left| \sum_{k=1}^{r} a_k(t)\phi_k(x)^T + \sum_{s=1}^{i} \phi_{e_s}(x, p(t)) - \sum_{k=1}^{r} a_k^{NN}(t)\phi_k(x)^T - \sum_{s=1}^{i} \phi_{e_s}(x, p^{NN}(t)) \right|\right|^2 \tag{5.3}$$

Specifically, the predicted time coefficients a_k^{NN} for the testing dataset were multiplied by the spatial modes ϕ_k, and the predicted parameters p^{NN} were introduced to the enrichment mode ϕ_e of Equation 4.12. Thus, the predicted reduced-order basis was derived.

The total error of the ROM, or the distance between the FOM and the predicted enriched reduced-order basis by the neural network, was expressed in terms of Mean Squared Error (MSE) for the testing dataset.

$$\epsilon_{\text{T}} = \frac{1}{N_x}\sum_{i=1}^{N_x} \left|\left| C_p^* - \sum_{k=1}^{r} a_k^{NN}(t)\phi_k(x)^T - \sum_{s=1}^{i} \phi_{e_s}(x, p^{NN}(t)) \right|\right|^2 \tag{5.4}$$

These three different metrics of the performance of the neural network for the ePOD-LSTM ROM with 18 DoF are illustrated in Figure 5.9.



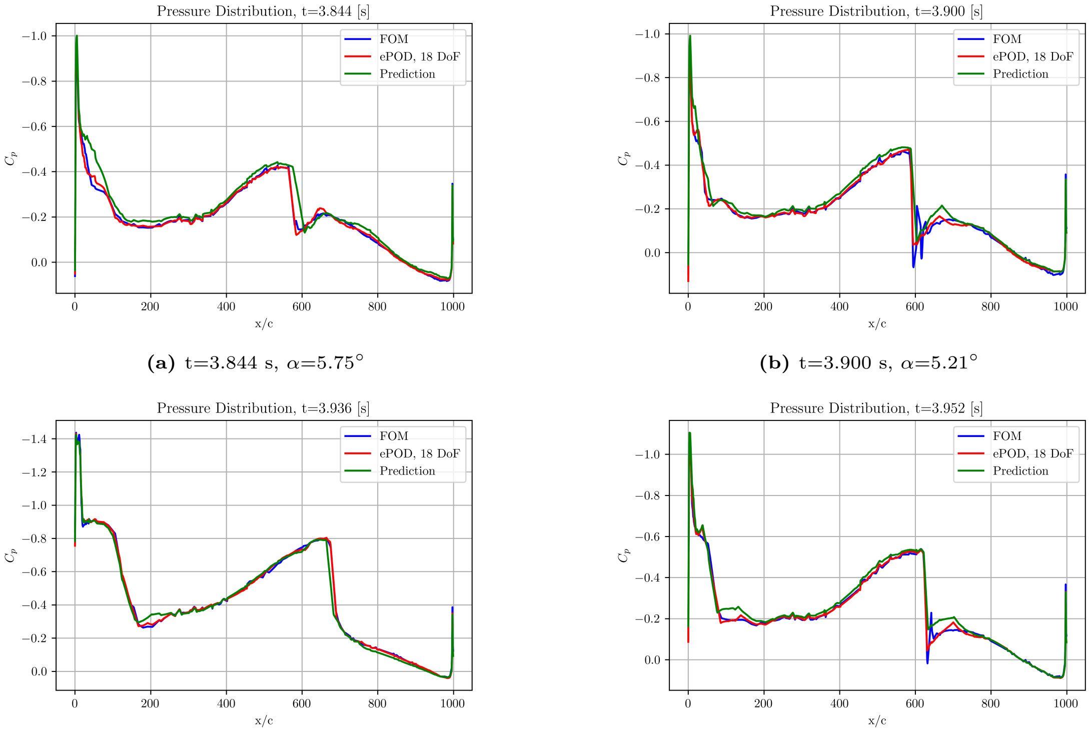
(a) Section 1

(b) Section 2

Figure 5.9: Mean squared error analysis of the predicted pressure distribution.

The projection error represents the highest level of accuracy that the neural network can achieve in this ROM architecture. An optimally constructed and perfectly trained neural network would ideally approach this level of accuracy. For the test dataset, the time-averaged projection error was $1.81 \cdot 10^{-4}$ in Section 1 and $2.04 \cdot 10^{-4}$ in Section 2. Regarding the total error, the time-averaged values were $1.39 \cdot 10^{-3}$ for Section 1 and $2.69 \cdot 10^{-3}$ for Section 2. The time-averaged network time coefficient error was $1.31 \cdot 10^{-3}$ in Section 1 and $2.59 \cdot 10^{-3}$ in Section 2. The main contribution to the total error was the neural network time coefficient error, which was nearly identical to the total error. Therefore, if an optimal neural network were available, the total error would likely be on the order of 10^{-4}. This finding was significant because it highlighted the advantage of decomposing the different sources of errors in the ePOD-LSTM ROM. By evaluating each component individually and assessing its error contribution, this approach offers an advantage compared to ROMs, e.g. Autoencoder/Decoder, where the different sources of error are aggregated into a total error.

Figure 5.10 displays the predicted and projected enriched reduced-order basis, as well as the full-order pressure distribution, for Section 1 at selected time steps. These plots demonstrate that, despite using a limited training dataset, the ePOD-LSTM model accurately predicts the pressure distribution for the test dataset, even in challenging discontinuity locations.



(a) t=3.844 s, α=5.75°

(b) t=3.900 s, α=5.21°

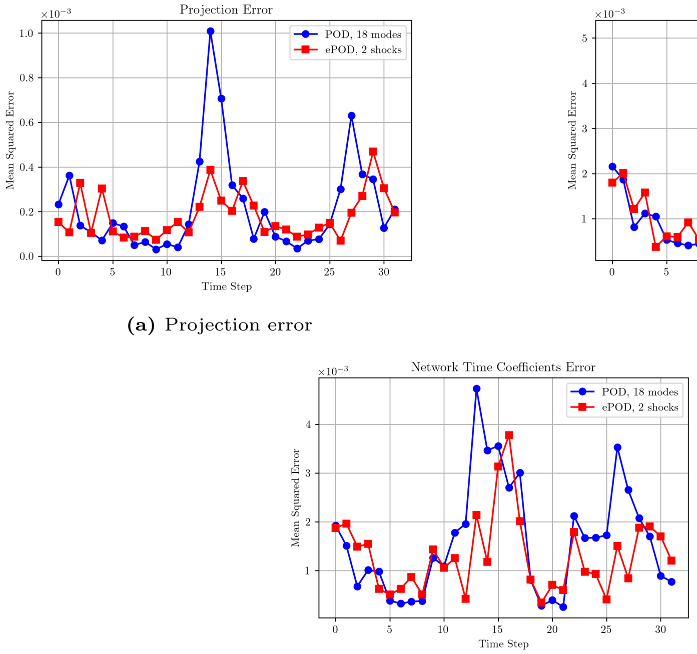
(c) t=3.936 s, α=13.16°

(d) t=3.952 s, α=6.83°

Figure 5.10: Predicted pressure distribution for Section 1 using the ePOD-LSTM ROM.

5.3.1. Comparison of ePOD-LSTM and POD-LTSM ROMs

The projection, total, and network time coefficient errors of the POD-LSTM and ePOD-LSTM models with the same DoF were compared. On average, the ePOD-LSTM exhibited lower projection errors than the standard POD-LSTM ROM. For time instances without strong pressure discontinuities, the error levels were comparable. The time-averaged projection error for the standard POD was 17.72% higher than the ePOD-LSTM with 18 DoF. The total error of the ePOD-LSTM ROM with 18 DoF was lower than that of the standard POD-LSTM for most time steps. The time-averaged total error was $1.39 \cdot 10^{-3}$, a 24.04% reduction compared to the POD-LSTM using the same number of DoF. In the POD-LSTM model, regions with high projection errors significantly contributed to the overall error. In contrast, the network time coefficient error was the primary source of error in the ePOD-LSTM ROM. The time-averaged network time coefficient error for the ePOD-LSTM was 18.63% less than the POD-LSTM with an equal number of DoF. This observation supports the hypothesis that predicting the time coefficients of high-order spatial modes was more challenging than fitting the enrichment parameters. Consequently, the enriched reduced-order basis offers the extra advantage of reducing the neural network time coefficient error, compared to a standard basis with the same number of DoF.



(a) Projection error

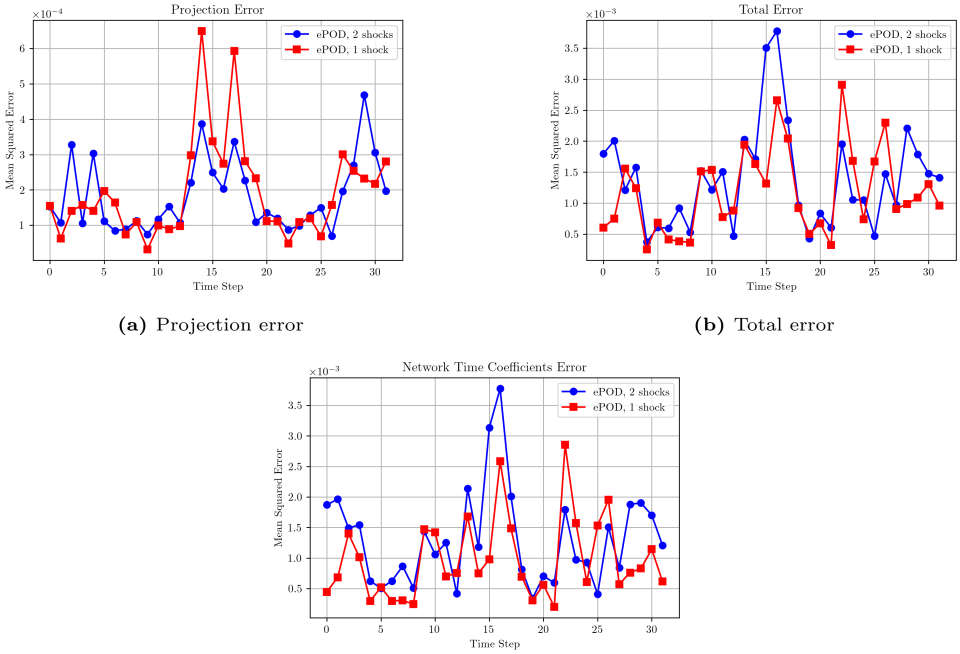
(b) Total error

(c) Neural network time coefficient error

Figure 5.11: Comparison of ROMs using a standard reduced-order basis versus an enriched reduced-order basis with 18 DoF.

5.3.2. Influence of Enrichment Mode on ePOD-LSTM ROM Performance

An ePOD-LSTM model was constructed using a single enrichment mode in the reduced-order basis. The hyperparameters, training, validation, and testing datasets were identical to those used for the model with two enrichment modes. The projection error for the test data had a similar magnitude for both ePOD-LSTM ROMs, consistently remaining lower than $7 \cdot 10^{-4}$. The temporal mean value of the projection error was $1.93 \cdot 10^{-4}$ for the ePOD-LSTM ROM with 14 DoF. Therefore, the time-averaged projection error was increased only by 6% after the removal of the secondary enrichment mode. Hence, the second shock discontinuity had a minor contribution to the projection error for the testing data while significantly increasing the number of DoF by almost 30% more. The decreased number of parameters that needed to be predicted by the neural network reduced the network's time coefficient error. As a result, the total error was lower for the ePOD-LSTM ROM with a single enrichment mode. Hence, the ePOD-LSTM ROM with one enrichment mode performed better in predicting the testing dataset.



(a) Projection error

(b) Total error

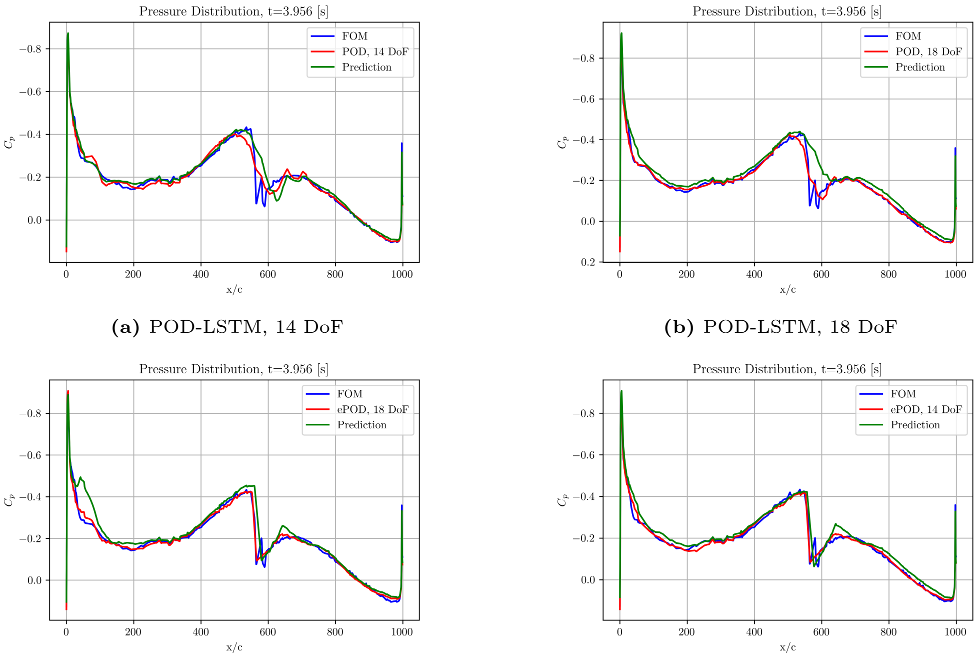
(c) Neural network time coefficient error

Figure 5.12: Performance comparison of ePOD-LSTM ROMs: single vs. double enrichment mode configurations

In conclusion, an augmented number of parameters increased the dimensions and the complexity of the reduced-order basis, hindering the neural network training, and validation, and eventually led to higher neural network error. Hence, one should always consider the trade-off between the accuracy that additional degrees of freedom add to the reduced-order basis representation and the complexity they add, for the neural network or other regression methods.

5.3.3. Discussion

Figure 5.13 represents a typical time instance of the predicted pressure distribution. The POD-LSTM models failed to capture the shock due to the high projection error in this region and the complexity of the time coefficients associated with the high-order modes. In the case of ePOD-LSTM ROM with 18 DoF, a high total error was observed in the first part of the pressure distribution. Attempting to fit the parameters of the secondary shock, while its contribution was minor, increased the complexity of the network and resulted in higher total error. On the other hand, the reduced number of enrichment basis parameters in the case of one shock facilitates the fitting of the neural network, leading to the best prediction results.



(a) POD-LSTM, 14 DoF

(b) POD-LSTM, 18 DoF

(c) ePOD-LSTM, 18 DoF

(d) ePOD-LSTM, 14 DoF

Figure 5.13: Predicted pressure distribution for Section 1, obtained using POD-LSTM and ePOD-LSTM ROMs.

The following tables summarize the time-averaged errors for all the Reduced-Order Models (ROMs) that were constructed, trained, and tested during the thesis project.

Table 5.3: Summary of time-averaged errors for various ROMs.

ROM	**Section**	**Degrees of Freedom**	**Projection Error**	**Network Time Coefficients Error**	**Total Error**
POD-LSTM	1	14	$2.41 \cdot 10^-4$	$1.01 \cdot 10^-3$	$1.25 \cdot 10^-3$
POD-LSTM	1	18	$2.20 \cdot 10^-4$	$1.61 \cdot 10^-3$	$1.83 \cdot 10^-3$
POD-LSTM	2	18	$2.99 \cdot 10^-4$	$3.70 \cdot 10^-3$	$4.00 \cdot 10^-3$
ePOD-LSTM	1	14	$1.93 \cdot 10^-4$	$9.75 \cdot 10^-4$	$1.17 \cdot 10^-3$
ePOD-LSTM	1	18	$1.81 \cdot 10^-4$	$1.31 \cdot 10^-3$	$1.39 \cdot 10^-3$
ePOD-LSTM	2	18	$2.04 \cdot 10^-4$	$2.59 \cdot 10^-3$	$2.69 \cdot 10^-3$

The enriched reduced-order basis enhances the ability of the LSTM neural network to predict the reduced-order pressure distribution in the transonic flow regime. Therefore, by substituting the high-order spatial modes with the enrichment basis, we efficiently approximated the shock discontinuities into the latent space while reducing the complexity of the features the neural network must fit.

Network time coefficient error was the primary contributor to the total error for ePOD-LSTM ROMs. The ePOD-LSTM ROMs with double and single enrichment modes in Section 1 exhibited 18.63% and 39.44% lower time-averaged network time coefficient errors, respectively, than POD-LSTM with 18 DoF. Comparing ROMs with one and two enrichment modes revealed that additional DoF increased model complexity, offsetting the gains in accuracy provided by a more detailed reduced-order basis.

Overall, the ePOD-LSTM ROM with single shock outperformed the other ROMs, reducing total error by 15.83% compared to the double shock configuration. Furthermore, it reduced the total error by 6.5% and 36.06% compared to POD-LSTM with 14 and 18 DoF correspondingly. Therefore, carefully balancing the accuracy of the enriched reduced-order basis with its complexity is crucial.

Furthermore, the POD-LSTM ROM in Section 1 with 14 DoF resulted in a slightly lower total error than the ePOD-LSTM ROM with 18 DoF. The increased dimensions of the enriched reduced-order basis resulted in higher network time coefficient error counterbalancing its increased accuracy. Therefore, as the number of parameters increased, the regression performance of the LSTM network declined. This observation highlights the need for an enriched reduced-order basis for approximating the pressure distribution in transonic flows with the minimum number of DoF.

6 Conclusion and Recommendations

6.1. Conclusion

This research project introduced a novel approach to address the increased number of spatial POD modes needed for accurate approximation of pressure distribution in transonic flows. The proposed enriched Proper Orthogonal Decomposition (ePOD) method introduces an enrichment basis into the standard formulation of the POD, which effectively approximates shock discontinuities, significantly reducing the degrees of freedom (DoF) and complexity of the reduced-order basis for pressure distribution. The ePOD method decreases the projection error, defined as the discrepancy between the reconstructed pressure distribution based on the enriched reduced-order basis and the Full-Order Model (FOM), compared to the standard POD with the same or even a reduced number of DoF. Furthermore, the ePOD method reduces the complexity of the time-variant parameters of the reduced-order basis. Thus, the computational cost of the ePOD-LSTM ROM was less in contrast to the POD-LSTM ROM with the same DoF.

Introducing enrichment basis with local linear sawtooth enrichment modes into the latent space for the NACA 0012 airfoil and the sections of the DLR-F22 model efficiently reduces the number of DoF required to accurately approximate pressure distribution in transonic flows compared to the standard POD. The results demonstrate that ePOD outperforms the standard POD method, leading to significantly lower projection error, particularly in regions with pressure discontinuities caused by shock waves. The time-averaged projection error in the enrichment domain was reduced by up to 73% compared to standard POD, for the same number of DoF. Furthermore, the time-averaged projection error of ePOD was reduced by over 50% in the enrichment domain, even when compared to standard POD with nearly one-third more DoF. The comparison between ePOD with single and double shock configurations for the DLR-F22 model suggests that, in some cases, was better to model only strong discontinuities. The additional enrichment mode increased the dimensionality of the enriched reduced-order basis without significantly improving its accuracy. Therefore, the trade-off between the improved accuracy and complexity or dimensionality of the reduced-order basis should be considered. The ePOD approach also reduces the complexity of the time-dependent parameters of the reduced-order basis that need to be predicted by the neural network, by reducing the high-frequency content of the time signals relative to their POD counterparts.

An Long Short-Term Memory (LSTM) neural network was constructed and trained to predict time-dependent coefficients and parameters of the enriched reduced-order basis in unseen flow conditions. Due to the reduced complexity of the ePOD's time-variant parameters, the training cost of the ePOD-LSTM ROM was reduced by up to 13% relative to the POD-LSTM model with the same DoF, using identical hyperparameters.

Finally, the performance of the ePOD-LSTM ROM was compared to the POD-LSTM ROM. Additionally, ePOD-LSTM ROMs for single and double shock configurations were compared to each other. The ePOD-LSTM ROM outperformed the POD-LSTM ROM in predicting pressure distribution for unseen conditions. The results also demonstrated that increasing the number of parameters predicted by the neural network can negatively impact the model's accuracy. Specifically, predicting the time coefficients of extra spatial modes was generally more challenging than enrichment parameters, resulting in higher neural network time coefficient errors. The time-averaged total error of the ePOD-LSTM ROM was reduced by almost 25% compared to the POD-LSTM ROM with the same DoF. Moreover, the ePOD-LSTM ROM with one shock wave resulted in a total error with a time-averaged, which was almost 16% less than the double shock configuration, and 36% less than the POD-LSTM ROM with almost one-third more DoF. The neural network time coefficient error followed similar trends for the ROMs under consideration.

High-fidelity CFD solvers result in a high computational cost for encompassing all different states of an aircraft. The POD-LSTM ROM developed by the Netherlands Aerospace Centre (NLR) does well reducing the cost of pressure distribution predictions in subsonic cases. However, in the transonic regime, flow complexity increases due to shock phenomena, hindering the application of the POD-LSTM ROM. This thesis shows that ePOD-LSTM efficiently decreases the dimensions of enriched reduced-order basis and the complexity of its time-dependent parameters. Therefore, ePOD-LSTM ROM can substantially reduce computational cost and complexity in predicting pressure distribution for airfoil/wing applications in transonic flows.

6.2. Recommendations For Future Work

The developed ePOD-LSTM ROM is however limited to predicting the pressure distribution over airfoil or wing sections. The novel approach of the study in combination with the limited data availability did not allow for an extension of the ePOD method to three dimensions within the time frame of this thesis. However, expanding the current two-dimensional ePOD method to three dimensions will be crucial for developing a surrogate model capable of predicting surface pressure over the entire wing. Introducing enrichment bases in multiple sections of the wing may lead to an increased number of degrees of freedom for the ePOD approach, potentially diminishing its efficiency compared to the standard POD for approximating surface pressure. To extend the ePOD method over the entire wing surface several recommendations can be considered.

The first recommendation concerns reducing the number of two-dimensional enrichment bases required to represent the effects of three-dimensional flow discontinuities on the wing surface. This can be achieved by identifying an optimal set of sections across the wing surface that accurately approximate complex three-dimensional flow phenomena The approach may include determining optimal chordwise coordinates and orientations of these sections. By interpolating their enrichment modes, it is possible to derive the minimum required number of enrichment bases.

The second recommendation suggests extending the ePOD approach across the wing surface by identifying three-dimensional enrichment modes that explicitly account for pressure discontinuities across the wing. Although this approach can be effective for linear flow discontinuities, deriving basis functions capable of describing complex non-linear discontinuities in the wing's surface pressure distribution remains challenging.

The final recommendation involves clustering the dataset based on the presence of pressure discontinuities caused by shock waves. Specifically, it suggests decomposing the data into clusters according to the expanding enrichment domains over the wing surface. Each cluster would be assigned a reduced-order basis, similar to the cluster POD method [34].

In parallel with the expansion of the enrichment basis to 3D, also more efficient neural network architectures could be considered in order to improve the prediction performance. In the current model, a RNN LSTM neural network has been demonstrated to be the most efficient [12]. However, the RNN's architecture imposes a limitation where only the last time steps are considered for predicting the time-dependent coefficients. In transonic flows, time steps containing shocks may be more significant than those without. Therefore, assigning greater importance to time instances related to shocks could potentially enhance the surrogate model's performance. A more advanced neural network architecture, such as transformer models [16], incorporates attention mechanisms, which allow the model to weigh the importance of different parts of the input sequence. This could be particularly beneficial in transonic flows, where specific time steps may have a more significant impact on the overall prediction.

References

[1] Abbott, Ira H, Von Doenhoff, Albert E, and Stivers Jr, Louis. *Summary of airfoil data*. Tech. rep. 1945.

[2] Ahmed, Shady E, San, Omer, Bistrian, Diana A, and Navon, Ionel M. “Sampling and resolution characteristics in reduced order models of shallow water equations: Intrusive vs nonintrusive”. In: *International Journal for Numerical Methods in Fluids* 92.8 (2020), pp. 992–1036.

[3] Balajewicz, Maciej, Nitzsche, Fred, and Feszty, Daniel. “Application of multi-input Volterra theory to nonlinear multi-degree-of-freedom aerodynamic systems”. In: *AIAA journal* 48.1 (2010), pp. 56–62.

[4] Baldi, Pierre and Hornik, Kurt. “Neural networks and principal component analysis: Learning from examples without local minima”. In: *Neural networks* 2.1 (1989), pp. 53–58.

[5] Bányász, C. and Automatic Control, International Federation of. *Adaptive Systems in Control and Signal Processing 1995: A Postscript Volume from the Fifth IFAC Symposium, Budapest, Hungary, 14-16 June 1995*. IFAC Series. International Federation of Automatioc Control, 1995. ISBN: 9780080423753. URL: `https://books.google.com/books?id=kNVSAAAAMAAJ`.

[6] Bar-Sinai, Yohai, Hoyer, Stephan, Hickey, Jason, and Brenner, Michael P. “Learning data-driven discretizations for partial differential equations”. In: *Proceedings of the National Academy of Sciences* 116.31 (2019), pp. 15344–15349.

[7] Bartels, Robert E and Sayma, AI. “Computational aeroelastic modelling of airframes and turbomachinery: progress and challenges”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365.1859 (2007), pp. 2469–2499.

[8] Berkooz, Gal, Holmes, Philip, and Lumley, John L. “The proper orthogonal decomposition in the analysis of turbulent flows”. In: *Annual review of fluid mechanics* 25.1 (1993), pp. 539–575.

[9] Boor, C de. “Subroutine package for calculating with B-splines”. In: *Los Alamos Scientific Laboratory Report LA 4728-MS* (1971).

[10] Bourier, Sébastien. “Development of a CFD data-driven surrogate model using the neural network approach for prediction of aircraft performance characteristics”. In: (2021).

[11] Brunton, Steven L, Noack, Bernd R, and Koumoutsakos, Petros. “Machine learning for fluid mechanics”. In: *Annual review of fluid mechanics* 52 (2020), pp. 477–508.

[12] Catalani, Giovanni. “Machine Learning Based local Reduced Order Modeling for the prediction of Unsteady Aerodynamic Loads.” In: (2022).

[13] Chambers, Joseph R and Hall, Robert M. “Historical review of uncommanded lateral-directional motions at transonic conditions”. In: *Journal of aircraft* 41.3 (2004), pp. 436–447.

[14] Cho, Kyunghyun, Van Merriënboer, Bart, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. In: *arXiv preprint arXiv:1406.1078* (2014).

[15] Demkowicz, Leszek, Rachowicz, Waldemar, and Devloo, Ph. “A fully automatic hp-adaptivity”. In: *Journal of Scientific Computing* 17 (2002), pp. 117–142.

[16] Dufter, Philipp, Schmitt, Martin, and Schütze, Hinrich. “Position information in transformers: An overview”. In: *Computational Linguistics* 48.3 (2022), pp. 733–763.

[17] Duraisamy, Karthik, Iaccarino, Gianluca, and Xiao, Heng. “Turbulence modeling in the age of data”. In: *Annual review of fluid mechanics* 51 (2019), pp. 357–377.

[18] Eilers, Paul HC. “A perfect smoother”. In: *Analytical chemistry* 75.14 (2003), pp. 3631–3636.

[19] Fresca, Stefania, Dede', Luca, and Manzoni, Andrea. "A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs". In: *Journal of Scientific Computing* 87 (2021), pp. 1–36.

[20] Fresca, Stefania and Manzoni, Andrea. "POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition". In: *Computer Methods in Applied Mechanics and Engineering* 388 (2022), p. 114181.

[21] Frink, Neal T, Hiller, Brett R, Murphy, Patrick C, Cunningham, Kevin, and Shah, Gautam H. "Investigation of reduced-order modeling for aircraft stability and control prediction". In: *AIAA Scitech 2019 Forum.* 2019, p. 0980.

[22] Garbo, Andrea and Bekemeyer, Philipp. "Unsteady physics-based reduced order modeling for large-scale compressible aerodynamic applications". In: *Computers & Fluids* 239 (2022), p. 105385.

[23] Gers, Felix A, Schmidhuber, Jürgen, and Cummins, Fred. "Learning to forget: Continual prediction with LSTM". In: *Neural computation* 12.10 (2000), pp. 2451–2471.

[24] Ghoreyshi, Mehdi, Aref, Pooneh, Stradtner, Mario, Rooij, Michel van, Panagiotopoulos, Anastasios, Blom, Peter Hans Leonard, and Hulshoff, Steven. "Evaluation of Reduced Order Aerodynamic Models for Transonic Flow Over a Multiple-Swept Wing Configuration". In: *AIAA AVIATION FORUM AND ASCEND 2024.* 2024, p. 4158.

[25] Ghoreyshi, Mehdi, Badcock, KJ, Da Ronch, Andrea, Marques, Simao, Swift, A, and Ames, Nigel. "Framework for establishing limits of tabular aerodynamic models for flight dynamics analysis". In: *Journal of Aircraft* 48.1 (2011), pp. 42–55.

[26] Ghoreyshi, Mehdi, Cummings, Russell M, Ronch, Andrea Da, and Badcock, Kenneth J. "Transonic aerodynamic load modeling of X-31 aircraft pitching motions". In: *AIAA journal* 51.10 (2013), pp. 2447–2464.

[27] Ghoreyshi, Mehdi, Jirásek, Adam, and Cummings, Russell M. "Reduced order unsteady aerodynamic modeling for stability and control analysis using computational fluid dynamics". In: *Progress in Aerospace Sciences* 71 (2014), pp. 167–217.

[28] Golev, Angel, Iliev, Anton, and Kyurkchiev, Nikolay. "A note on the Soboleva'modified hyperbolic tangent activation function". In: *International Journal of Innovative Science Engineering and Technology* 4.6 (2017).

[29] Gonzalez, Francisco J and Balajewicz, Maciej. "Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems". In: *arXiv preprint arXiv:1808.01346* (2018).

[30] Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron. *Deep Learning.* `http://www.deeplearningbook.org`. MIT Press, 2016.

[31] Halder, Rahul, Damodaran, Murali, and Khoo, BC. "Deep learning based reduced order model for airfoil-gust and aeroelastic interaction". In: *AIAA Journal* 58.10 (2020), pp. 4304–4321.

[32] Heath, Michael T. *Scientific computing: an introductory survey, revised second edition.* SIAM, 2018.

[33] Heeg, Jennifer. "Overview of the aeroelastic prediction workshop". In: *51st AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition.* 2013, p. 783.

[34] Hess, Martin, Alla, Alessandro, Quaini, Annalisa, Rozza, Gianluigi, and Gunzburger, Max. "A localized reduced-order modeling approach for PDEs with bifurcating solutions". In: *Computer Methods in Applied Mechanics and Engineering* 351 (2019), pp. 379–403.

[35] Hines Chaves, Derrick Armando and Bekemeyer, Philipp. "Data-driven reduced order modeling for aerodynamic flow predictions". In: (2022).

[36] Hochreiter, Sepp and Schmidhuber, Jürgen. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[37] Hofer, Lucas R., Krstajić, Milan, and Smith, Robert P. *JAXFit: Trust Region Method for Nonlinear Least-Squares Curve Fitting on the GPU.* 2022. arXiv: `2208.12187 [cs.LG]`. URL: `https://arxiv.org/abs/2208.12187`.

[38] Holmes, Philip. *Turbulence, coherent structures, dynamical systems and symmetry.* Cambridge university press, 2012.

[39] Hornik, Kurt, Stinchcombe, Maxwell, and White, Halbert. “Multilayer feedforward networks are universal approximators”. In: *Neural networks* 2.5 (1989), pp. 359–366.

[40] Jirásek, Adam, Jeans, Tiger, Martenson, Matthew, Cummings, Russell, and Bergeron, Keith. “Improved methodologies for maneuver design of aircraft stability and control simulations”. In: *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition.* 2010, p. 515.

[41] Jirásek, Adam, Jeans, Tiger L, Martenson, Matthew, Cummings, Russell M, and Bergeron, Keith. “Improved methodologies for the design of maneuver for stability and control simulations”. In: *Aerospace Science and Technology* 25.1 (2013), pp. 203–223.

[42] Kaya, Halil, Tiftikçi, Hakan, Kutluay, Ümit, and Sakarya, Evren. “Generation of surrogate-based aerodynamic model of an UCAV configuration using an adaptive co-Kriging method”. In: *Aerospace Science and Technology* 95 (2019), p. 105511.

[43] Kok, JC and Prananta, BB. “User guide of ENSOLV version 7.60. A flow solver for aerodynamic, aeroelastic, and aeroacoustic applications using 3D multi-block structured grids”. In: *NLR, July* (2016).

[44] Kovacs, Léo, Passaggia, Pierre-Yves, Mazellier, Nicolas, and Lago, Viviana. “Detection method for shock-waves in viscous flows”. In: *Experiments in Fluids* 63.1 (2022), p. 11.

[45] Lee, Kookjin and Carlberg, Kevin T. “Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders”. In: *Journal of Computational Physics* 404 (2020), p. 108973.

[46] Li, Jing and Zhang, Weiwei. “The performance of proper orthogonal decomposition in discontinuous flows”. In: *Theoretical and Applied Mechanics Letters* 6.5 (2016), pp. 236–243.

[47] Lipton, Zachary C, Berkowitz, John, and Elkan, Charles. “A critical review of recurrent neural networks for sequence learning”. In: *arXiv preprint arXiv:1506.00019* (2015).

[48] Lucia, David J and Beran, Philip S. “Projection methods for reduced order models of compressible flows”. In: *Journal of Computational Physics* 188.1 (2003), pp. 252–280.

[49] Lucia, David J, Beran, Philip S, and Silva, Walter A. “Reduced-order modeling: new approaches for computational physics”. In: *Progress in aerospace sciences* 40.1-2 (2004), pp. 51–117.

[50] Lucia, David J, King, Paul I, and Beran, Philip S. “Domain decomposition for reduced-order modeling of a flow with moving shocks”. In: *AIAA journal* 40.11 (2002), pp. 2360–2362.

[51] Lumley, John Leask. “The structure of inhomogeneous turbulent flows”. In: *Atmospheric turbulence and radio wave propagation* (1967), pp. 166–178.

[52] McDaniel, David R, Nichols, Robert H, Eymann, Timothy A, Starr, Robert E, and Morton, Scott A. “Accuracy and Performance Improvements to Kestrel’s Near-Body Flow Solver”. In: *54th AIAA Aerospace Sciences Meeting.* 2016, p. 1051.

[53] McDaniel, DR, Cummings, RM, Bergeron, Keith, Morton, SM, and Dean, JP. “Comparisons of CFD solutions of static and maneuvering fighter aircraft with flight test data”. In: *3rd International Symposium on Integrating CFD and Experiments in Aerodynamics.* 2007, pp. 20–21.

[54] Mitchell, Tom M. *Machine learning.* 1997.

[55] Mohan, Arvind T and Gaitonde, Datta V. “A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks”. In: *arXiv preprint arXiv:1804.09269* (2018).

[56] Moré, Jorge J. “The Levenberg-Marquardt algorithm: implementation and theory”. In: *Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977.* Springer. 2006, pp. 105–116.

[57] Morelli, Eugene A. “Multiple input design for real-time parameter estimation in the frequency domain”. In: *IFAC Proceedings Volumes* 36.16 (2003), pp. 639–644.

[58] Morton, Scott A, Tillman, Brett, McDaniel, David R, Sears, David R, and Tuckey, Todd R. "Kestrel–A Fixed Wing Virtual Aircraft Product of the CREATE Program". In: *2009 DoD High Performance Computing Modernization Program Users Group Conference*. IEEE. 2009, pp. 148–152.

[59] O'Shea, Keiron and Nash, Ryan. "An introduction to convolutional neural networks". In: *arXiv preprint arXiv:1511.08458* (2015).

[60] Papp, David. "Prediction of unsteady nonlinear aerodynamic loads using deep convolutional neural networks: Investigating the dynamic response of agile combat aircraft". In: (2018).

[61] Poplingher, Lior, Raveh, Daniella E, and Dowell, Earl H. "Modal analysis of transonic shock buffet on 2D airfoil". In: *AIAA Journal* 57.7 (2019), pp. 2851–2866.

[62] Rabczuk, T. and Bathe, K.J. *Machine Learning in Modeling and Simulation: Methods and Applications*. Computational Methods in Engineering & the Sciences. Springer International Publishing, 2023. ISBN: 9783031366444. URL: `https://books.google.gr/books?id=wO3aEAAAQBAJ`.

[63] Rooij, Michel PC van, Frink, Neal T, Hiller, Brett R, Ghoreyshi, Mehdi, and Voskuijl, Mark. "Generation of a reduced-order model of an unmanned combat air vehicle using indicial response functions". In: *Aerospace Science and Technology* 95 (2019), p. 105510.

[64] Rosenblatt, Frank. *The Perceptron, a Perceiving and Recognizing Automaton*. Project Para. Cornell Aeronautical Laboratory, 1957.

[65] Rowley, Clarence W and Dawson, Scott TM. "Model reduction for flow analysis and control". In: *Annual Review of Fluid Mechanics* 49 (2017), pp. 387–417.

[66] Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

[67] Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

[68] Samuel, Arthur L. "Some studies in machine learning using the game of checkers". In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229.

[69] Schroeder, M. "Synthesis of low-peak-factor signals and binary sequences with low autocorrelation (Corresp.)" In: *IEEE Transactions on Information Theory* 16.1 (1970), pp. 85–89. DOI: `10.1109/TIT.1970.1054411`.

[70] Shur, Michael L, Strelets, Michael K, Travin, Andrey K, and Spalart, Philippe R. "Turbulence modeling in rotating and curved channels: assessing the Spalart-Shur correction". In: *AIAA journal* 38.5 (2000), pp. 784–792.

[71] Sirovich, L. "Turbulence and the dynamics of coherent structures, Parts I, II and III". In: *Quart. Appl. Math.* (1987), pp. 561–590.

[72] Sirovich, Lawrence. "Turbulence and the dynamics of coherent structures. I. Coherent structures". In: *Quarterly of applied mathematics* 45.3 (1987), pp. 561–571.

[73] Smith, Brian R and McWaters, Marcus A. "Aerodynamic Database Requirements for the Detailed Design of Tactical Aircraft: Implications for the Expanded Application of CFD". In: *Use of Computational Fluid Dynamics for Design and Analysis: Bridging the Gap Between Industry and Developers* (2022).

[74] Soboleva, EV and Beskorovainyi, VV. "The utility function in problems of structural optimization of distributed objects". In: *Kharkiv University of Aair Force* 121 (2008).

[75] Stradtner, Mario, Drazen, David, and Rooij, Michel van. "Introduction to AVT-351: Enhanced Computational Performance and Stability & Control Prediction for NATO Military Vehicles". In: *AIAA SCITECH 2023 Forum*. 2023, p. 0820.

[76] Taira, Kunihiko, Brunton, Steven L, Dawson, Scott TM, Rowley, Clarence W, Colonius, Tim, McKeon, Beverley J, Schmidt, Oliver T, Gordeyev, Stanislav, Theofilis, Vassilios, and Ukeiley, Lawrence S. "Modal analysis of fluid flows: An overview". In: *Aiaa Journal* 55.12 (2017), pp. 4013–4041.

[77] Tekaslan, Huseyin E, Demiroglu, Yusuf, and Nikbay, Melike. “Surrogate Unsteady Aerodynamic Modeling with Autoencoders and LSTM Networks”. In: *AIAA SciTech 2022 Forum.* 2022, p. 0508.

[78] Tobak, Murray and Chapman, Gary T. “Nonlinear problems in flight dynamics involving aerodynamic bifurcations”. In: *AGARD Symp. on Unsteady Aerodyn. Fundamentals and Appl. to Aircraft Dyn.* A-85179. 1985.

[79] Vinuesa, Ricardo and Brunton, Steven L. “Enhancing computational fluid dynamics with machine learning”. In: *Nature Computational Science* 2.6 (2022), pp. 358–366.

[80] Wang, Zheng, Xiao, Dunhui, Fang, Fangxin, Govindan, Rajesh, Pain, Christopher C, and Guo, Yike. “Model identification of reduced order fluid dynamics systems using deep learning”. In: *International Journal for Numerical Methods in Fluids* 86.4 (2018), pp. 255–268.

[81] Weiss, Julien. “A tutorial on the proper orthogonal decomposition”. In: *AIAA aviation 2019 forum.* 2019, p. 3333.

[82] Widhalm, Markus, Stradtner, Mario, Schütte, Andreas, Ghoreyshi, Mehdi, Jirasek, Adam, and Seidel, Jurgen. “Comparison of Reduced Order Models for Evaluating Stability Derivatives for the DLR-F22 ONERA model”. In: *AIAA AVIATION 2023 Forum.* 2023, p. 4199.

[83] Wright, Stephen J. *Numerical optimization.* 2006.

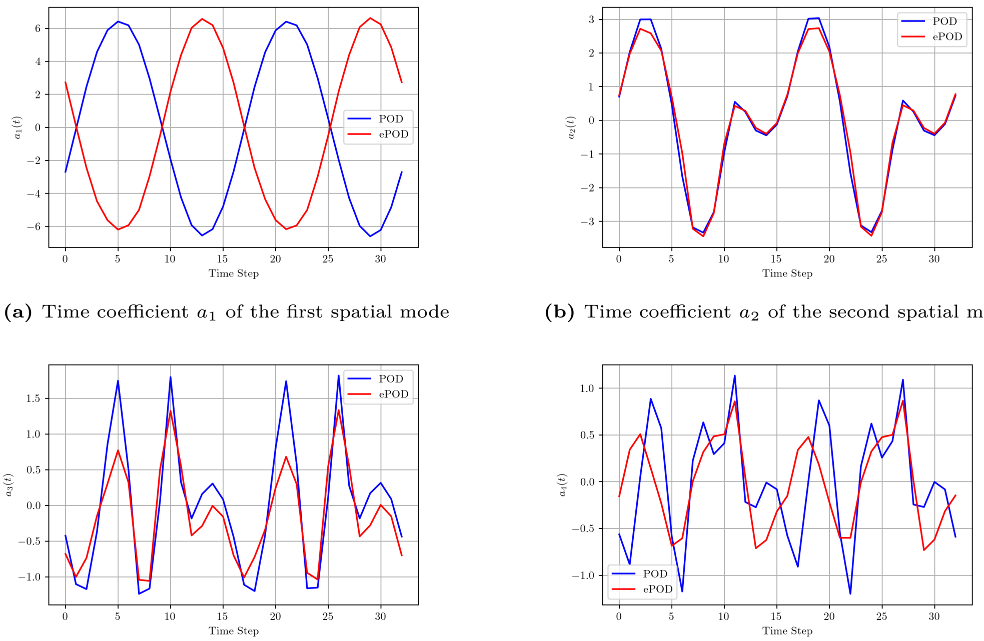
[84] Zahn, R, Weiner, A, and Breitsamter, C. “Prediction of wing buffet pressure loads using a convolutional and recurrent neural network framework”. In: *CEAS Aeronautical Journal* (2023), pp. 1–17.

A

Additional Results

A.1. Time-Dependent parameters of NACA 0012

To gain a deep understanding of the parameters that the neural network needs to predict, the time coefficients of the truncated spatial modes are plotted as a function of time in Figure A.1.



(a) Time coefficient a_1 of the first spatial mode

(b) Time coefficient a_2 of the second spatial mode

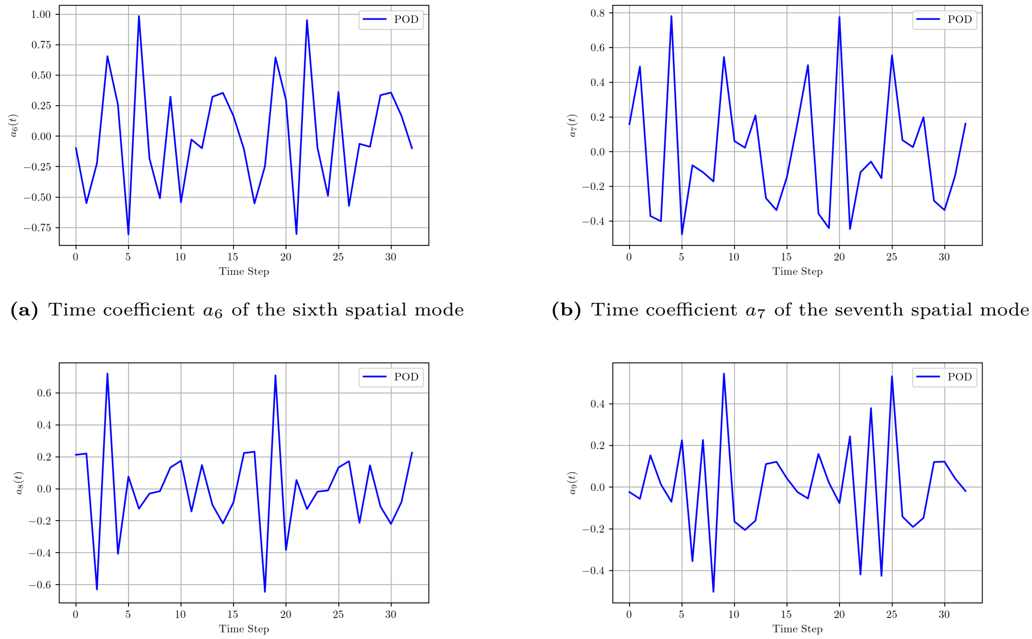
(c) Time coefficient a_3 of the third spatial mode

(d) Time coefficient a_4 of the fourth spatial mode

Figure A.1: Comparison of time coefficients for the first four spatial modes between POD and ePOD methods.

The time coefficients for the initial spatial modes were similar for both the POD and ePOD methods. However, as the mode order increases, the absolute deviation between the time coefficients of the POD and ePOD methods also increases.

Figure A.2 shows the last four modes of the truncated POD basis with 9 spatial modes. These time coefficients for the four additional POD modes were compared to the enrichment parameters of the ePOD approach, which are represented in Figure A.3.



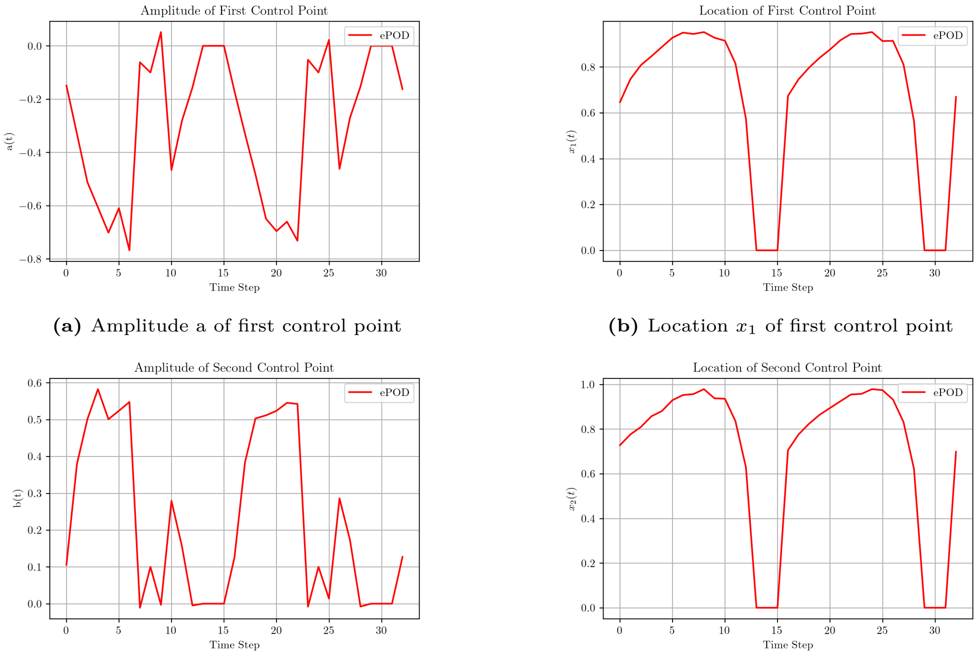
(a) Time coefficient a_6 of the sixth spatial mode

(b) Time coefficient a_7 of the seventh spatial mode

(c) Time coefficient a_8 of the eighth spatial mode

(d) Time coefficient a_9 of the ninth spatial mode

Figure A.2: Time coefficients of the last four spatial modes of standard POD method.



(a) Amplitude a of first control point

(b) Location x_1 of first control point

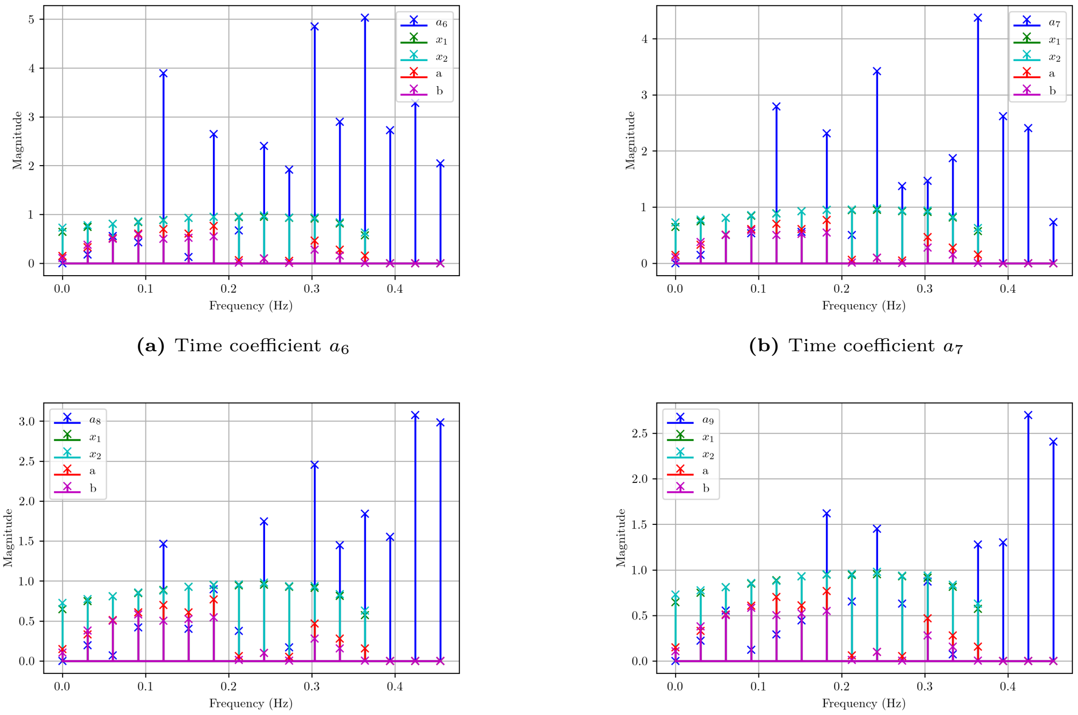
(c) Amplitude b of second control point

(d) Location x_2 of second control point

Figure A.3: Time-dependent enrichment parameters of the ePOD method for the NACA 0012.

To facilitate a comparison of these parameters, the Fast Fourier Transform (FFT) was applied to represent these time-related values in the frequency domain. The time coefficients of the four extra spatial modes of the standard POD was compared with all the enrichment parameters, as follows in

Figure A.4:



(a) Time coefficient a_6

(b) Time coefficient a_7

(c) Time coefficient a_8

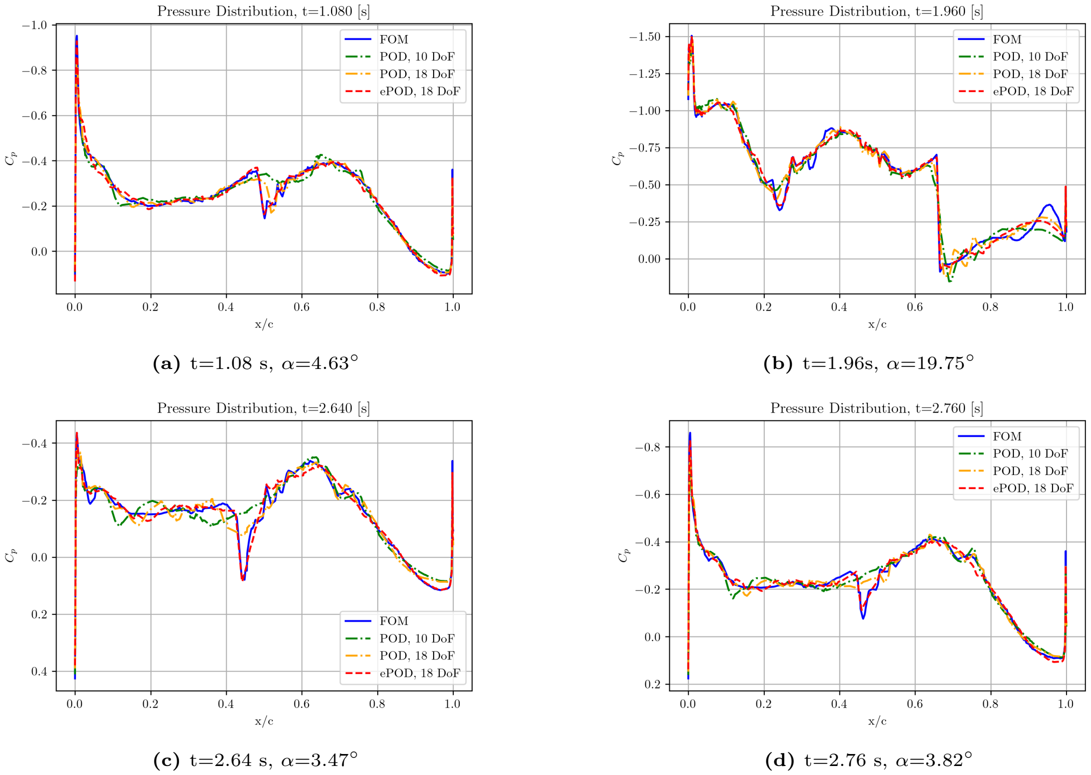
(d) Time coefficient a_9

Figure A.4: Comparison of the frequency content of time coefficients for the four additional POD modes and the corresponding enrichment parameters.

The time coefficients of the extra high-order spatial modes in the standard POD primarily contained high frequencies with large amplitudes. In contrast, the enrichment parameters were more evenly distributed across lower frequencies and exhibited, on average, lower amplitudes. Consequently, the frequency content of the time coefficients for the four additional spatial POD modes was characterized by higher frequencies and larger amplitudes relative to the enrichment parameters, as shown in Figure A.4. Additionally, the enrichment parameters had nearly periodic behavior over time, as observed in Figure A.3. Therefore, the enrichment parameters appeared less complex time signals than the time coefficients of the high-order spatial POD modes, making them more suitable for regression.

A.2. DLR-F22 Model

A.2.1. Reconstructed Pressure Distribution in Section 1, PRBS Signal



(a) t=1.08 s, α=4.63°

(b) t=1.96s, α=19.75°

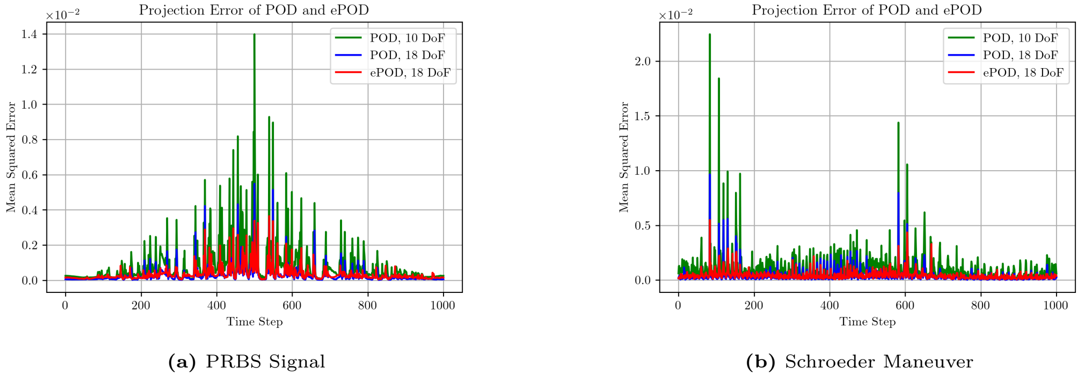
(c) t=2.64 s, α=3.47°

(d) t=2.76 s, α=3.82°

Figure A.5: Reconstruction of the pressure distribution in Section 1 using ePOD and POD methods, PRBS signal.

A.2.2. Evaluation of ePOD in Section 2

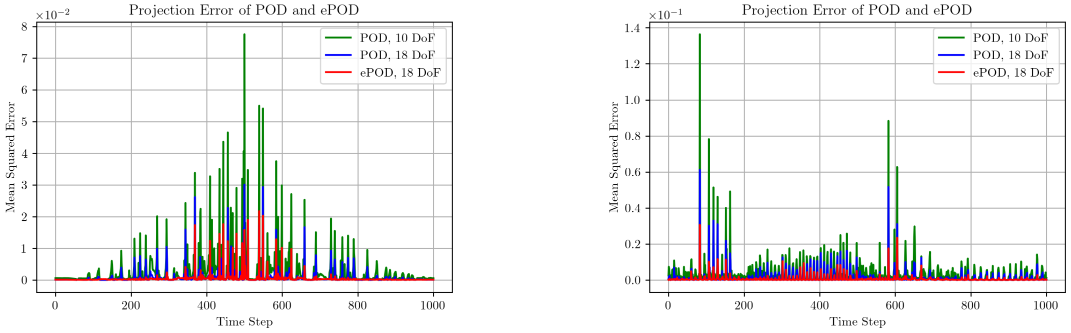
The projection error for the Section 2 of the DLR-F22 Model is represented in the following graphs:



(a) PRBS Signal

(b) Schroeder Maneuver

Figure A.6: Projection error in Section 2 of the DLR-F22 model.

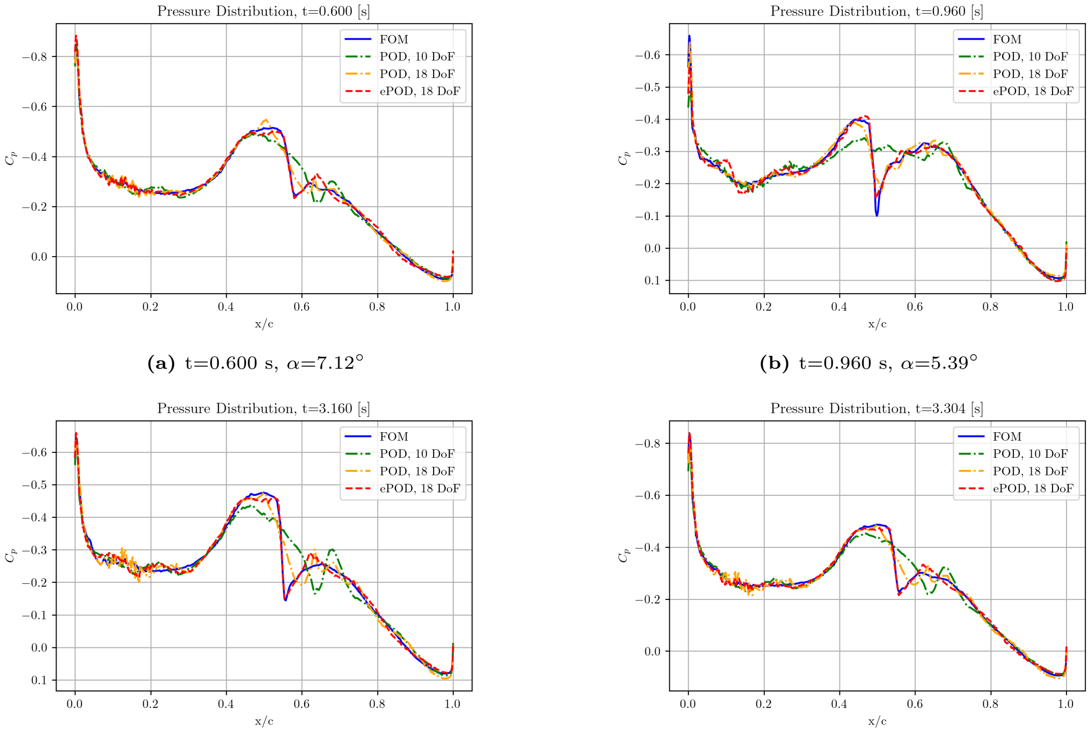


(a) PRBS Signal

(b) Schroeder Maneuver

Figure A.7: Projection error in the enrichment domain, Section 2 of the DLR-F22 model.

The ePOD outperforms the standard method with the same number of degrees of freedom. The projection error remains consistently lower than that of the standard POD with an equivalent number of DoF across all time steps in the PRBS signal and Schroeder maneuver. For further details regarding the time-average projection error, refer to Table 4.4. The reconstructed pressure distributions over various timesteps for the Section 2 under the PRBS signal and Schroeder maneuver are illustrated in Figure A.8 and Figure A.9.



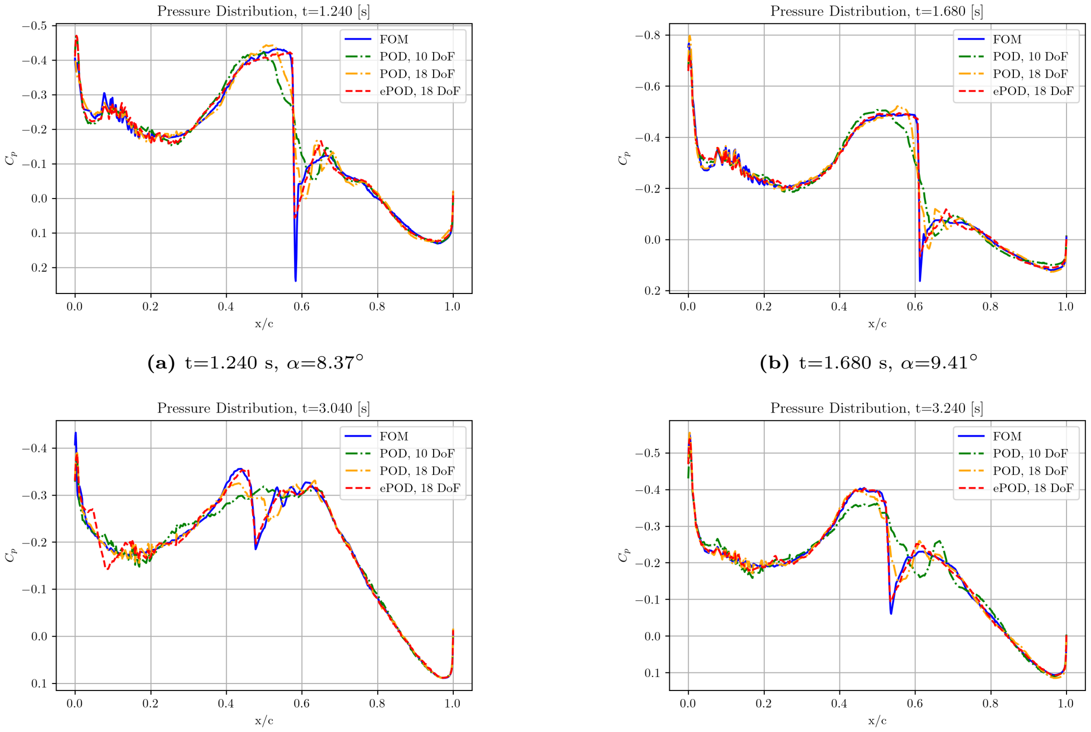
(a) t=0.600 s, α=7.12°

(b) t=0.960 s, α=5.39°

(c) t=3.160 s, α=5.96°

(d) t=3.304s, α=6.52°

Figure A.8: Reconstruction of the pressure distribution in Section 2 using ePOD and POD methods, PRBS signal.



(a) t=1.240 s, α=8.37°

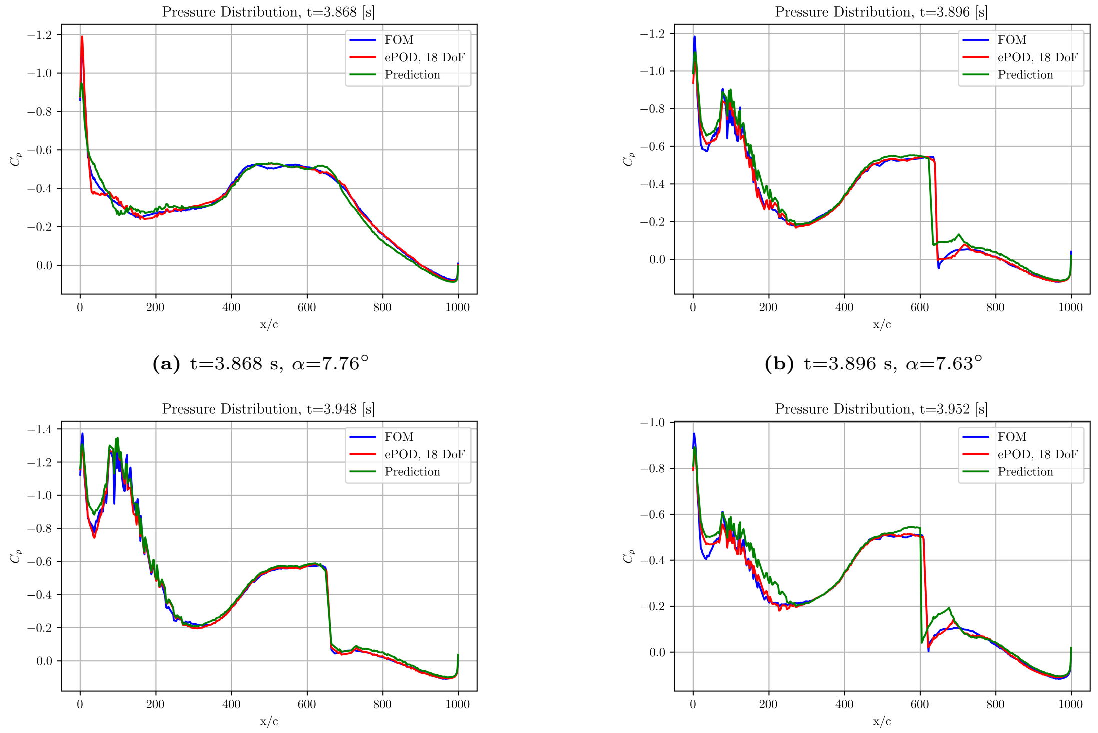
(b) t=1.680 s, α=9.41°

(c) t=3.040 s, α=4.41°

(d) t=3.240 s, α=4.85°

Figure A.9: Reconstruction of the pressure distribution in Section 2 using ePOD and POD methods, Schroeder maneuver.

Figure A.10 shows the predicted and projected enriched reduced-order basis, as well as the full-order pressure distribution, for Section 2 at various time steps. These plots demonstrated that the ePOD-LSTM model accurately predicts the pressure distribution for the test dataset, even in challenging discontinuity locations.



(a) t=3.868 s, α=7.76°

(b) t=3.896 s, α=7.63°

(c) t=3.948 s, α=9.05°

(d) t=3.952 s, α=6.83°

Figure A.10: Predicted pressure distribution for Section 2 using the ePOD-LSTM ROM.