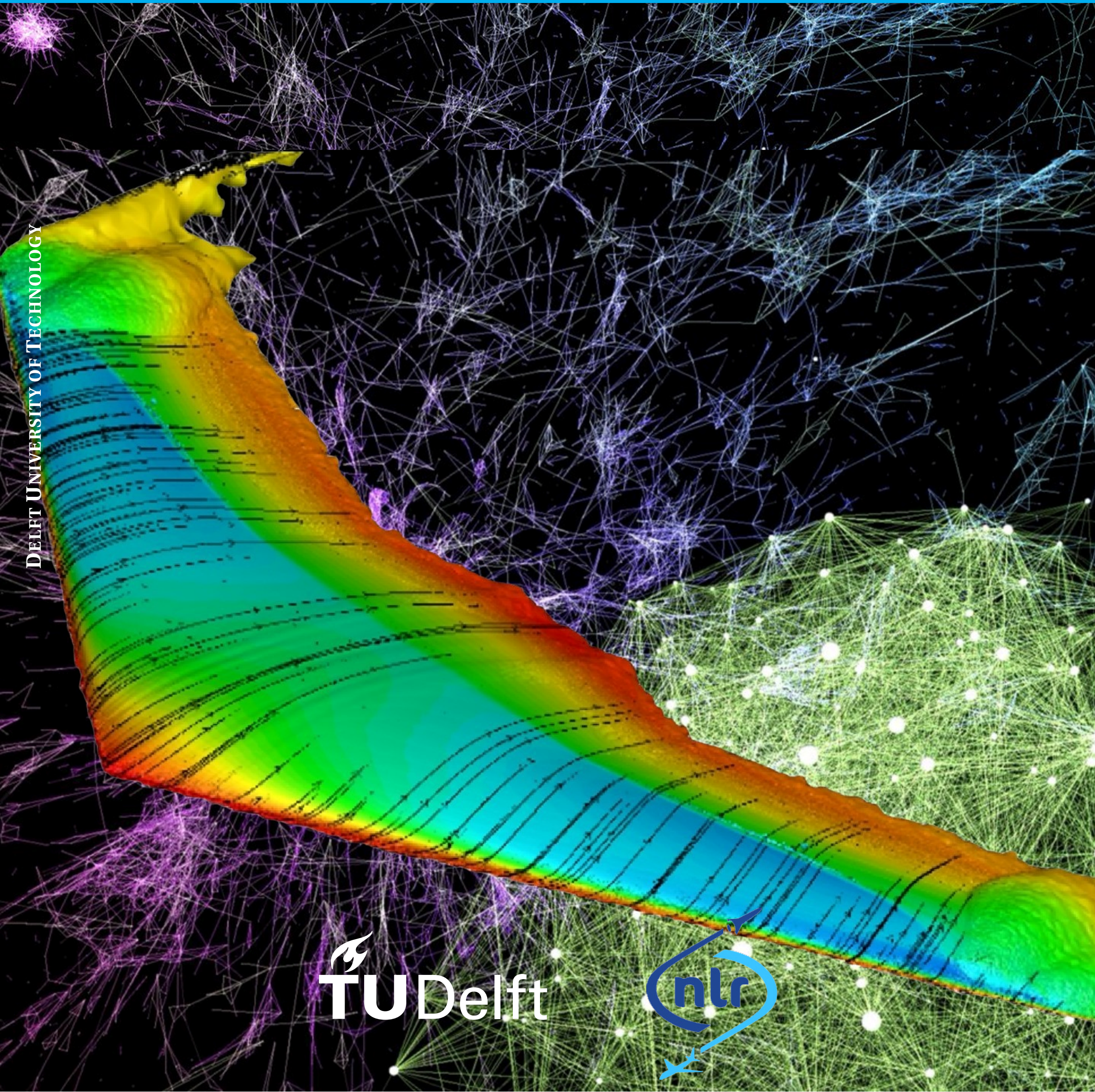
Machine Learning Based Local Reduced Order Modeling for the prediction of unsteady aerodynamic loads

MSc Thesis Aerospace Engineering

Giovanni Catalani



Delft University of Technology

Machine Learning Based Local Reduced Order Modeling for the Prediction of Unsteady Aerodynamic Loads

by

Giovanni Catalani

in partial fulfillment of the requirements for the degree of

Mster of Science
in Aerospace Engineering

at Delft University of Technology
to be defended publicly on Monday, November 21, 2022 at 10:00.

Student Number:	5349788	
Supervisors:	Dr. S. J. Hulshoff	Delft University of Technology (TU Delft)
	Dr. ir. M. P. C. van Rooij	Royal Netherlands Aerospace Centre (NLR)
Thesis commitee:	Dr. R.P. Dwight	Delft University of Technology (TU Delft)
	Dr.ir. E. van Kampen	Delft University of Technology (TU Delft)
	Dr. S. J. Hulshoff	Delft University of Technology (TU Delft)
	Dr. ir. M. P. C. van Rooij	Royal Netherlands Aerospace Centre (NLR)
Project Duration:	Jan, 2022 - Nov, 2022	

Contents

List of Figures vii

List of Tables xiii

Abstract xv

Preface xvii

1 Introduction 1
- 1.1 Background 1
- 1.2 Project Motivation 3
- 1.3 Research Objective & Research Questions 3
- 1.4 Report Outline 4

2 Literature: Reduced Order Models 5
- 2.1 The Proper Orthogonal Decomposition 5
- 2.2 Machine Learning 7
  - 2.2.1 Artificial Neural Networks 8
  - 2.2.2 Recurrent Neural Networks 11
  - 2.2.3 Gaussian Process Regression 14

3 ROM Methodology 17
- 3.1 The MULDICON unmanned combat aerial vehicle 17
- 3.2 Flight Conditions & Flight Dynamics 20
  - 3.2.1 High-Fidelity Solutions of the unsteady aerodynamics 21
  - 3.2.2 Parameter Space Sampling: Training Maneuver 25
  - 3.2.3 Test Maneuvers 28
- 3.3 The scaled global POD 30
- 3.4 Domain Decomposition POD 33
- 3.5 Clustered POD: from global to local 36
  - 3.5.1 Snapshot Clustering 37
  - 3.5.2 Local Basis Computation 42
  - 3.5.3 Surrogate Modelling of cluster local dynamics 43
  - 3.5.4 Assign unseen parameter point to a cluster 45
  - 3.5.5 Gaussian Mixture Models for Overlapping Clusters 47
- 3.6 Latent Dynamics Modelling 49
  - 3.6.1 Machine Learning for Regression 49
  - 3.6.2 The Multi-Layer-Perceptron 51
  - 3.6.3 The Gaussian Process Regression 53
  - 3.6.4 The Long Short Term Memory Network 53
  - 3.6.5 Direct Loads Coefficients Prediction via surrogate modeling 55

3.7 Error Analysis 56

4 Results and Discussion 59

4.1 Scaled POD 59

4.2 Domain Decomposition POD 63

4.3 Cluster POD. 65

4.4 Comparative Study Between Methods 71

4.4.1 Projection Error 71

4.4.2 Total Error 74

4.4.3 ROM Computational Cost 77

4.5 Direct coefficient Prediction and Surrogate Model comparison 79

4.5.1 Integral loads coefficient prediction 79

4.5.2 Accuracy and Efficiency of the Surrogate Models. 82

5 Conclusion 85

5.1 Recommendations for future work 87

A Additional Results 89

A.1 Special Maneuver: Sharp Pitch up-down 89

A.2 Direct Load Predictions 92

A.3 Projection Errors 99

A.4 Total ROM Errors 100

A.4.1 5 Modes Reconstruction 100

A.4.2 10 Modes Reconstruction 107

List of Figures

2.1 Simple neuron operation. [42] 9

2.2 Simple Feedforward Neural Network with one hidden layer [42] 9

2.3 Supervised Training Flowchart of a standard ANN 11

2.4 Unfolded Recurrent Neural Network [42] 12

2.5 Schematic representation of the memory cell of an LSTM network [42] . . 13

3.1 SACCON and MULDICON planforms. [49] 18

3.2 Design mission profile of MULDICON [41] 20

3.3 Reference frames and motion variables description. [51] 21

3.4 Computational Mesh on the MULDICON UCAV surface for the RANS solutions computation by the ENSOLV solver. [49] 22

3.5 Data Structure of the Wing surface pressure coefficient. [9] 24

3.6 High fidelity solutions from ENSOLV of pressure fields in steady conditions ($M = 0.2$) for different angles of attack [49]. 25

3.7 Angle of Attack and Pitch Rate signals over time for training maneuver. . . 27

3.8 Projected regressor space coverage for training maneuver. 28

3.9 Description of pitch and plunge input signal and motions. 29

3.10 $\frac{A_{cell}}{A_{min}}$ Ratio between individual cell areas and minimum cell area along the wing upper surface 30

3.11 First three modes on the wing upper surface resulting from the snapshot scaling. Top: Pressure-POD modes (unscaled). Middle: Force-POD modes. Bottom: Moment-POD modes. 32

3.12 Geometry decomposition of the wing upper surface. The lower surface is continuously attached to the upper surface at the leading edge. 34

3.13 First three modes on the wing upper surface for two different geometry decompositions. First Row: $DD_{0.3}$. Second Row: $DD_{0.8}$ 35

3.14 Second POD coefficients over the training maneuver for the $DD_{0.8}$ decomposition 35

3.15 DD-POD based ROM flowchart. The flowchart does not include the parameter space sampling and the CFD simulation step for the sake of conciseness. 36

3.16 K-means Clustering visualization in the input space ($\alpha - q$) and in Global POD coordinates ($a_1 - a_2$) for two distinct POD decompositions. 40

3.17 First three modes on the wing upper surface for each cluster local basis. Each row corresponds to a cluster. The cluster numbering follows Fig. 3.16. The POD is performed on the Moment scaled snapshots. 43

3.18 Comparison of clusters in the POD coordinates subspace, using GMM with overlap and K-means algorithm. Points in red indicate the overlap regions. $K = 4$. 49

3.19 Pearson correlation coefficient between the instantaneous control inputs and the first five global POD reduced coefficient. 50
3.20 Multi-Layer-Perceptron Architecture. 51
3.21 LSTM many-to-one Architecture. 54

4.1 Absolute projection error $\epsilon_P^{cp}(\boldsymbol{x})$ using three different snapshot scaling methods. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$ at $t = 1s$. $\alpha = 1.71[deg]$,$\dot{\alpha} = 23.4[deg/s]$,$q = 23.4[deg/s]$ 60
4.2 Projection of the integral loads' coefficients using 5 modes and three different scaled bases.Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$. Top: Axial Force Coefficient C_A. Middle:Normal Force Coefficient C_N. Bottom:Pitching Moment Coefficient C_{M_y} 61
4.3 Training and Validation Loss curves for LSTM network trained on the same input. The outputs are the first 5 reduced coefficients of the three bases. . 62
4.4 Absolute projection error $\epsilon_P(\boldsymbol{x})$ using different geometry decompositions. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$ at $t = 0.25s$. Instantaneous parameters: $\alpha = 19.93[deg]$,$\dot{\alpha} = 5.38[deg/s]$,$q = 5.38[deg/s]$ 64
4.5 Time averaged Mean Squared Projection Error vs number of modes over several testing maneuvers, using two different decomposition strategies. . 64
4.6 Absolute total error $\epsilon_P(\boldsymbol{x})$ using different geometry decompositions. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$ at $t = 0.25s$. $\alpha = 19.93[deg]$,$\dot{\alpha} = 5.38[deg/s]$,$q = 5.38[deg/s]$ 65
4.7 DD POD-LSTM pressure field predictions and comparison with CFD ground truth data. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$ at $t = 0.25s$. $\alpha = 19.93[deg]$,$\dot{\alpha} = 5.38[deg/s]$,$q = 5.38[deg/s]$ 66
4.8 Optimal clustering of the input space (α-q) and MAE optimization curves. Cluster features: first 5 global Moment-POD coefficients. In black the validation maneuver: harmonic pitch at $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.25Hz$. 67
4.9 Within Cluster Sum of Squared Errors and Total MSE on validation maneuver vs number of clusters 68
4.10 Time averaged Mean Squared Projection Error vs the number of modes over several testing maneuvers. $k = 4$ clusters. 68
4.11 Instantaneous inputs and Mean Squared Projection error using Global POD, Cluster POD with overlapping GMM and Cluster POD with K-means. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.5Hz$. Number of modes: 10 69
4.12 Cluster POD-LSTM pressure field prediction, comparison with CFD ground truth data, and absolute prediction error. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$ at $t = 0.25s$. Instantaneous parameters: $\alpha = 19.93[deg]$,$\dot{\alpha} = 5.38[deg/s]$,$q = 5.38[deg/s]$ 70
4.13 Time averaged Mean Squared Projection Error vs number of modes over harmonic pitching maneuvers for the Global POD, $DD_{0.8}$-POD and Cluster POD bases. 72

4.14 Time averaged Mean Squared Projection Error vs number of modes over harmonic plunging maneuvers for the Global POD, $DD_{0.8}$-POD and Cluster POD bases. 73
4.15 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 1Hz$. Number of modes: 5. . . . 74
4.16 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 5 . . 75
4.17 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$.Number of modes: 5 . . . 76
4.18 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 5. . 77
4.19 Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 1Hz$ . . . 80
4.20 Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.25Hz$ 80
4.21 Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$ . . 81
4.22 Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.25Hz$ 82

A.1 Time averaged Mean Squared Projection Error vs number of modes over sharp pitch up-down maneuver for the Global POD, $DD_{0.8}$-POD and Cluster POD bases. 90
A.2 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Sharp pitch up-down maneuver.Number of modes: 5 90
A.3 Pressure field prediction for highest error case of pitch up-down maneuver. $\alpha = 21.95[deg]$,$\dot{\alpha} = -0.58[deg/s]$,$q = -20[deg/s]$ 91
A.4 Direct Integral Loads coefficient prediction using the three different surrogate models. Sharp pitch up-down maneuver. 92
A.5 Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$. 92
A.6 Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.5Hz$ . . 93
A.7 Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.5Hz$. 93
A.8 Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.0125Hz$ 94
A.9 Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.25Hz$ 94

A.10 Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.5Hz$ 95
A.11 Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.25Hz$ 95
A.12 Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.5Hz$. 96
A.13 Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.25Hz$ 96
A.14 Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.5Hz$ 97
A.15 Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 1Hz$ . . 97
A.16 Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.0125Hz$ 98
A.17 Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.5Hz$ 98
A.18 Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.25Hz$ 99
A.19 Time averaged Mean Squared Projection Error vs number of modes over harmonic plunging maneuvers for the Global POD, $DD_{0.8}$-POD and Cluster POD bases. 99
A.20 Time averaged Mean Squared Projection Error vs number of modes over harmonic plunging maneuvers for the Global POD, $DD_{0.8}$-POD and Cluster POD bases. 100
A.21 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 5 . . 100
A.22 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.5Hz$.Number of modes: 5 . . . 101
A.23 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.5Hz$.Number of modes: 5 . . 101
A.24 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 1Hz$.Number of modes: 5 . . . 102
A.25 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.5Hz$.Number of modes: 5 . . 102
A.26 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 5 . . 103
A.27 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 5 . . 103

A.28 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.5Hz$.Number of modes: 5 . . . 104
A.29 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 1Hz$.Number of modes: 5 . . . 104
A.30 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.125Hz$.Number of modes: 5 105
A.31 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.25Hz$.Number of modes: 5 . 105
A.32 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.5Hz$.Number of modes: 5 . . 106
A.33 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 5 . . 106
A.34 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 10 . . 107
A.35 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.5Hz$.Number of modes: 10 . . 107
A.36 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 1Hz$.Number of modes: 10 . . . 108
A.37 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 10 . 108
A.38 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.5Hz$.Number of modes: 10 . . 109
A.39 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 1Hz$.Number of modes: 10 . . . 109
A.40 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 10 . 110
A.41 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$.Number of modes: 10 . . . 110
A.42 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 10 . . 111

A.43 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.5 Hz$.Number of modes: 10 . . 111
A.44 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.25 Hz$.Number of modes: 10 . 112
A.45 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 1 Hz$.Number of modes: 10 . . . 112
A.46 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.125 Hz$.Number of modes: 10 113
A.47 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.25 Hz$.Number of modes: 10 113
A.48 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.5 Hz$.Number of modes: 10 . 114
A.49 Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.25 Hz$.Number of modes: 5 . . 114

LIST OF TABLES

3.1 Main Design Parameters of the MULDICON [40] 18
3.2 Reference Geometrical Parameters of the MULDICON configuration [49] . 19
3.3 MULDICON Mission Design Points. [41] 20
3.4 Chosen Flight Conditions.[51] 21
3.5 Multisine input signal for training maneuver. [51] 27
3.6 Motion variables values for pitch test maneuvers [51] 29
3.7 Motion variables values for plunge test maneuvers [51] 30
3.8 Description of the spatial scaling of the snapshots 31
3.9 Decomposition types and corresponding datasets dimensions. 34
3.10 Clustering using 4 cluster: datasets dimensions. 44
3.11 Hyperparamter optimization settings and results. 52
3.12 Hyperparamter and training settings of the final MLP model. 53
3.13 Hyperparameters and training settings of the final LSTM model and the downscaled version. 55

4.1 Projection error on integral loads coefficients using different scaling and for a different number of modes. 62
4.2 Assignment accuracy using global LSTM model for different maneuvers. . 70
4.3 Offline and Online computational time of the considered ROMs equipped with different surrogate models. Number of modes: 5. 78
4.4 Prediction error on the integral loads coefficient of different surrogate models for several maneuvers. 83
4.5 Training time comparison of different surrogate models for the direct prediction of integral loads coefficients. 84

Abstract

Advancements in aircraft performance require increasingly complex design processes and tools. Simulating the unsteady non-linear aerodynamic interaction between a maneuvering aircraft and the surrounding flowfield poses serious challenges. High-Fidelity Computational Fluid Dynamics (CFD) methods, based on the numerical solution of the Navier-Stokes equations, can in general provide accurate solutions but at a computational cost that is often unfeasible for many applications that require real-time evaluation of the aerodynamic responses over a large range of conditions. Reduced Order Models (ROMs) are methods that can alleviate the computational burden of performing High-Fidelity simulations while providing accurate solutions over a wide extent of parametric variations. In the context of unsteady aerodynamics simulations, most traditional Reduced Order Models are limited to the prediction of integral loads and do not scale well for the treatment of high-dimensional systems.

Recently, within the Science and Technology Organization work-group of NATO, a data-driven ROM based on the Proper Orthogonal Decomposition and Neural Networks (POD-LSTM) has been proposed for the prediction of the unsteady pressure fields on the UCAV MULDICON aircraft configuration, showing good potential in terms of computational efficiency and interpretability compared to the previously developed ROM based on end-to-end Convolutional Neural Networks. The POD-LSTM model is limited by the linear modal decomposition method that shows a slow convergence rate, in terms of latent space dimension, to the full-order solution. The high-projection error, localized in specific regions of the parameter space, translates into a consistent inaccuracy in the prediction of integral loads.

In order to overcome these limitations, two main methodologies are proposed to replace the Global POD basis with modal bases localized in i) the computational space or ii) in the parameter space, yielding the development of Local ROMs. The first methodology is based on a Domain Decomposition strategy and aims to reduce the projection error in specific regions of the wing surface. The second methodology, the Cluster-POD, focuses on partitioning the parameter space in relevant flow regimes and constructs Local Reduced Order models on each cluster. The choice of surrogate models, for latent dynamics modeling, is also discussed and Machine Learning methods based on the Long Short Term Memory Neural Network, Multi-layer-Perceptron, and Gaussian Process Regression are tested and compared.

Results show that Local ROMs can improve the efficiency of the global latent representation, by generating sets of modes with more localized information content. The proposed Cluster-POD-based ROM, in particular, can be developed using a systematic procedure that automatically detects the parameter space partitions based on an a-posteriori error indicator. It is also briefly shown, as a proposal for future research, how Gaussian Mixture Models can be used to generate overlapping clusters, that solve some of the limitations of traditional Cluster-based ROMs based on the K-means clustering.

Furthermore, improvements in the prediction of integral loads are demonstrated by including the loads coefficients directly in the Machine Learning surrogate models targets, in order to remove the projection error emerging from the use of a reduced order basis.

Preface

Firstly, I would like to express my gratitude to my supervisors Steven and Michel for constantly guiding me during this project in the past months, providing useful feedback and offering valuable insights in every weekly meeting and discussion we had. I hope that our paths will cross again in the future. Thanks to Sebastien for the inputs and the guidance he offered in the first weeks of the projects. At the same time, I would like to thank NLR for providing me with the opportunity and the support to work on this project.
This thesis marks the end of two unforgettable years as a student at TU Delft, an experience that would not have been possible without the love and the support of my family, and especially the sacrifices of my parents that have allowed me to follow my aspirations since I was a kid.

A special thanks goes to the people that filled the past two years with memories: to my housemates Riccardo, Luka and Cristian, my friends Luis and Toma, to Marco, Pietro, Davide, Francesca, Valentine, Irene, Lukas, Gregoire, Guillaume, the Erasmus group and many other people that contributed to shape the person that I am today. Thanks to Lou for the long study breaks during which I came up with most of the ideas for this thesis. Finally, thanks to my old friends in Italy: Laura, Milani, Fonti, Peppe, Francesco, Gigi, the Math friends and the friends from Norma, for being always present for me.

Giovanni Catalani
Delft, November 2022

1

Introduction

1.1. Background

The performance, control, and stability characteristic are ultimately determined by the aerodynamic loads acting on an aircraft. Modern combat aircraft are designed to perform complex maneuvers over a wide range of operating conditions, thus experiencing highly non-linear interaction with the surrounding flow field. The unsteadiness of this interaction poses serious issues and unwanted consequences such as flutter instability [21] and other aeroelastic phenomena that must be taken into account during the design phase of the aircraft. The correct prediction of the aerodynamic loads and loads distributions is therefore a fundamental step of the design process and the accuracy of these predictions determines the gap between the theoretical and the real performances of the aircraft. Traditional analytical models for the evaluation of aerodynamic loads [64, 60] are limited to simplified configurations and are not capable of correctly modeling the unsteady and non-linear aerodynamics of a full-scale aircraft [28], typically characterized by a large number of degrees of freedom.

Computational Fluid Dynamics (CFD) methods, on the other hand, offer powerful tools to simulate the mathematical model used to approximate the physical reality of fluid flows and can provide high-fidelity solutions that optimally correlate with experimental results and real-life scenarios. Several frameworks are available to numerically solve the Navier-Stokes Equations on a discretized domain: performing Direct Numerical Simulations (DNS) of this system of PDEs allows to solve the entire range of spatial and temporal scales of the flow at a prohibitive computational cost for most applications. The Large Eddy Simulation (LES) and Reynolds Averaged Navier Stokes (RANS) formulation, include physical considerations to fully (RANS) or partially (LES) model the dynamics of the smallest scales of motion, reducing the computations of DNS while providing high-fidelity solutions. However, in the context of multi-disciplinary design, typically an abundant number of simulations over a wide range of operating conditions is needed in order to perform aerodynamic optimization tasks or to devise flow control strategies: in these cases performing high-fidelity CFD simulations becomes extremely

demanding.
Reduced Order Models (ROMs) are a class of methods developed in order to alleviate the computational burden associated with the simulation of complex physical problems characterized by a large number of degrees of freedom while providing relatively accurate solutions. In the context of aerodynamics load predictions most ROM strategies found in the relevant literature are based on Modal Reduction [33, 56, 14], Indicial Response Theory [61, 22], Volterra Theory [55, 4, 24] and surrogate-based frameworks [24].

State-of-the-art surrogate models include Bayesian approaches such as Gaussian Process Regression [54, 16] and Machine Learning methods such as Artificial Neural Networks (ANNs). In particular, thanks to the increasing availability of data, the development of more powerful computational units (such as GPUs and TPUs), and the scalability of modern Machine Learning algorithms, the use of Deep Learning architectures for the construction of ROM are gaining in popularity [10]. Nevertheless, most Deep Learning based ROMs for the analysis of unsteady non-linear aerodynamics are limited to the prediction of a small number of integral aerodynamics scalar quantities such as force and moment coefficients [45, 66] and are mostly based on Recurrent Neural Networks (RNNs). More complex is the problem of generating high-dimensional flowfield predictions such as velocity or pressure field distributions over an airfoil or a wing, as the problem becomes computationally untractable for most standard Artificial Neural Networks. Convolutional Neural Networks (CNNs) have demonstrated unmatched efficiency in processing high-dimensional spatially related data and have found application in the generation of ROMs for the prediction of the steady non-linear aerodynamic [38, 17]. The application of CNNs to high-dimensional and time-dependent aerodynamics is problematic, requiring a large amount of data to train complex models. The end-to-end CNN model proposed by Papp [51] shows good promise for the prediction of pressure fields over a maneuvering aircraft but suffers from several drawbacks related to the difficulty of including time-history effects in the model and the complexity of the model that ultimately influences the training time. Bourier [9] proposes to combine a dimensionality reduction technique based on the Proper Orthogonal Decomposition [31] and to model the low-order dynamics on the POD linear subspace through a Long Short Term Memory Network (LSTM). The resulting POD-LSTM improves the computational efficiency of the CNN method with similar levels of accuracy. Nevertheless, the use of the global POD basis presents limitations and large projection errors originate from the linear subspace projection of the full-order dynamics on the POD manifold. These errors are also responsible for the low level of accuracy in the prediction of certain integral quantities such as the pitching moment coefficient.
In this study, the POD-LSTM model serves as a starting point for the analysis of the dimensionality reduction techniques and the reduced dynamics modeling. The identification of local bases in the computational space through Domain Decomposition or over the parameter space through Clustering techniques is proposed, yielding local ROMs with the goal of overcoming the limitation of the use of a global linear subspace. Alternatives to the surrogate modeling of the reduced dynamics are discussed, with the aim of understanding the accuracy and computational efficiency gained from using a Recurrent Neural Network architecture. Moreover, the integral loads' predictions are directly obtained as an output of the surrogate model, limiting the influence of projection errors

in the computation of these quantities.

1.2. Project Motivation

This project is part of the ongoing research on the development of Reduced Order Models for the multi-disciplinary design and simulation of combat aircraft, conducted by the AVT-351 North Atlantic Treaty Organisation (NATO) STO research task group. Previous studies conducted by David Papp [51] (within the AVT-251 research task group) and Sébastien Bourier [9] at the Royal Netherlands Aerospace Center (NLR) have focused on the development of ROMs for the prediction of the unsteady pressure fields over the Unmanned Combat Air Vehicle (UCAV) configuration. The ultimate goal of the research is the development of an unsteady ROM that can be used for the prediction of pressure and loads distribution over a specified aircraft configuration, which can be generalized to complex six degrees of freedom maneuvers and can include variations in the free stream conditions (Mach and Reynolds numbers). Such a model could allow fast prediction of the aerodynamic forces over a large range of conditions and could be coupled with structural analysis tools in order to investigate the aeroelastic behavior of the aircraft. This work is primarily focused on clarifying some of the aspects of the POD-LSTM method and identifying alternative procedures to generate the ROM based on Local Decomposition methods in order to improve the accuracy and efficiency of the baseline method.

1.3. Research Objective & Research Questions

The primary research objective that is addressed by the M.Sc. thesis work can be stated as follows.

"The objective of this research is to improve and extend the performance Reduced Order Model based on Modal Decomposition and Neural Networks for the prediction of the integral loads and the pressure fields of the UCAV MULDICON aircraft design, with a specific focus on Local Modal Decomposition approaches and alternative surrogate models for the prediction of the reduced dynamics"

In order to achieve this objective, the following main research question is formulated:

"What are the limitations of the POD-LSTM method, and how can Local ROMs increase the performance of the existing ROM in the prediction of the unsteady pressure fields and integral loads of the UCAV MULDICON test case?"

The following sub-questions, which specify the limits and the extension of this investigation, are identified.

1. *Can the Cluster POD local bases improve the reduced representation of the full-order system? How can the snapshot set be optimally clustered?*

2. *Can the Domain Decomposition local bases improve the accuracy of the ROM in*

specific regions of the wing compared to the standard POD? How does the decomposition geometry influence the spatial accuracy of the resulting ROM?

3. *How do the performances of an LSTM Neural Network compare to a simpler Multi-Layer-Perceptron and to a Bayesian approach based on Gaussian Process Regression?*

4. *Can the ROM prediction accuracy of the integral loads coefficient be increased through a modification of the POD basis? Is there a more efficient procedure to predict these quantities?*

1.4. REPORT OUTLINE

In this report, the limits of the Global POD-LSTM model are investigated and new approaches for the construction of local Reduced Order Models for unsteady aerodynamics are proposed and tested. In Chapter 3, the methodology of the ROMs is presented, with a special focus on the MULDICON UCAV test case, the generation of the dataset, the dimensionality reduction techniques, and the Machine Learning surrogate models. In Chapter 4, the main results and outcomes are presented, centering the discussion on the comparison of the various proposed approaches in order to provide satisfactory answers to the research questions. In Chapter 5, the main conclusions are given, by answering the research questions and several recommendations to further develop the methodology are suggested. In the Appendices A, additional material and results are provided for completeness.

2

Literature: Reduced Order Models

Reduced Order Models (ROMs) are a class of methods that allow the simulation of complex physical systems with limited resources, reducing the computational effort of high-fidelity solvers. Fluid dynamics systems can often be described accurately by the Navier-Stokes equations, but the typically high number of degrees of freedom needed to capture the spatial and time evolution of these dynamical systems makes the numerical simulation of these equations unpractical for real-time applications.
In the field of Aerodynamics, several ROMs have been developed in order to model the dynamics of fluid flows, to describe, control, and predict the evolution of the system in time or for different parametric instances. In particular, for the prediction of loads and pressure distribution over a maneuvering aircraft, various models have been described in the relevant literature: these models are based on modal reduction methods, indicial response theory, Volterra theory, and surrogate-based models (including Machine Learning methods). This brief literature review will cover the basic theory of non-intrusive Reduced Order Models based on modal dimensionality reduction and surrogate modeling of the low-order dynamics.

2.1. The Proper Orthogonal Decomposition

The Proper Orthogonal Decomposition (POD) in Fluid dynamics was first introduced by Lumley [7] to study the dynamics of the turbulent wake behind a cylinder, but the technique is widely known in other fields under different denominations (Principal Component Analysis, Kevin-Love decomposition). The objective of POD is to determine an optimal representation of an ensemble of data using a reduced number of modes, namely a linear subspace V_r of the full space V_N, spanned by a reduced number of POD modes. The ensemble of data from which the full order space is obtained can be generated by sampling the high-fidelity solution in the computational domain Ω at different instants in time or at different values of the system parameters. In the latter case, the set of samples (commonly referred to as snapshots) $\mathscr{S} = \{\boldsymbol{u}_{N,m}\}_{m=1}^{M}$ corresponds to M distinct realizations of the dynamical system in the p-dimensional parameter domain Θ: $\vec{\theta}_m \in \mathbb{R}^p$. The goal of POD is to find the optimal set of spatial basis functions $\{\boldsymbol{\phi}_n(\boldsymbol{x})\}_{n=1}^{N}$ such that each element in the original set of snapshots can be expressed as:

$$\boldsymbol{u}_{N,m}(\boldsymbol{x}) = \sum_{n=1}^{N} \boldsymbol{a}_{n,m}\boldsymbol{\phi}_n(\boldsymbol{x}) \tag{2.1}$$

where the coefficients $\{\boldsymbol{a}_{n,m}\}_{n=1}^{N}$ are obtained by projection of the snapshots on the POD orthonormal basis. If the average of the snapshots is not centered around zero, it can be advantageous [58] to use only consider the fluctuating part of the snapshot:

$$\boldsymbol{u}'_{N,m}(\boldsymbol{x}) = \boldsymbol{u}_{N,m}(\boldsymbol{x}) - \bar{\boldsymbol{u}}(\boldsymbol{x}) \tag{2.2}$$

but in the remainder of the discussion, the notation is not changed for clarity. The POD optimality condition is expressed mathematically as a minimization problem in the form:

$$\min_{\{\boldsymbol{\phi}_n(\boldsymbol{x})\}_{n=1}^{r}} \sum_{m=1}^{M} |\boldsymbol{u}_{N,m} - \sum_{l=1}^{r} (\boldsymbol{u}_{N,m} \cdot \boldsymbol{\phi}_l)(\boldsymbol{\phi}_l)|^2 \tag{2.3}$$

where $||\cdot||$ is the norm induced by the $L^2(\Omega)$ inner product $(\cdot)$ [37]. The choice of the inner product determines the notion of optimality, thus several formulations of POD can be derived. However, the inner product on the Hilbert space of square-integrable complex-valued functions in Ω is naturally suited for fluid dynamics application, since it corresponds to a finite kinetic energy flow.

It can be shown by means of variational calculus tools [31] that the solution to the minimization problem of Eq. 2.3 can be found by computing the Singular Value Decomposition of the snapshot matrix $\boldsymbol{X}$:

$$\boldsymbol{X} = \boldsymbol{\Phi\Sigma\Omega} \tag{2.4}$$

where $\boldsymbol{X} = [\boldsymbol{u}_{N,1}, ..., \boldsymbol{u}_{N,M}] \in \mathbb{R}^{N\times M}$, $\boldsymbol{\Sigma} = diag(\sigma_1, ..., \sigma_p, 0, ..., 0) \in \mathbb{R}^{N\times M}$ is a diagonal matrix, and the left singular matrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_N] \in \mathbb{R}^{N\times N}$ contains the POD modes. Ordering the eigenvalues σ_i by magnitude, provides the ranking of the POD modes in terms of energetic content of the original snapshot set contained in each mode, in other words, the mode associated with the largest eigenvalues produces the smallest average projection error of the snapshot set on said mode, the modes associated to the second largest eigenvalue the second smallest average projection error and so on.

The coefficients of the POD expansion 2.1 of the snapshots can be determined using the orthonormality property of the modes:

$$\boldsymbol{a}_{n,m} = \boldsymbol{u}_{N,m} \cdot \boldsymbol{\phi}_n. \tag{2.5}$$

Thanks to the energetic ranking of the POD modes the truncation of the POD expansion in Eq. 2.1 yields the optimal representation of the snapshot (in the POD sense) using a reduced number r of modes:

$$\boldsymbol{u}_{N,m}(\boldsymbol{x}) \approx \sum_{n=1}^{r} \boldsymbol{a}_{n,m}\boldsymbol{\phi}_n(\boldsymbol{x}) \tag{2.6}$$

and therefore the full dynamics of the system can be compressed to the knowledge of a reduced set of coordinates $\{\boldsymbol{a}_{n,m}\}_{n=1,m=1}^{n=r,m=M}$. These coefficients can then be used to train a data-driven surrogate model to approximate the latent dynamics as a function of the parametric inputs:

$$\{\boldsymbol{a}_{n,m}\}_{n=1}^{r} = \mathscr{F}(\vec{\theta}_m) \tag{2.7}$$

The Proper Orthogonal Decomposition is a powerful tool to reduce the order of complex systems. The optimality of the POD modes can be expressed in two ways: firstly in terms of minimization of the mean square error between the original snapshots and their truncated representation, and secondly in terms of the minimum number of modes required to reconstruct the snapshots for a given error.

The orthogonality of POD modes is a useful property for the computation of the reduced coordinates and a prerequisite for projection-based methods such as the Galerkin projection method [57]. Nevertheless, the global POD basis suffers from several shortcomings associated with the linear nature of the method. Firstly, POD modes are arranged only according to their energy contents and do not take into account the dynamic importance of each mode. As a result, truncation leads to ignoring low energy modes that can have a relevant impact on the dynamics of the retained modes [5]. Correlated to this observation, the POD modes rarely resemble physical evolving structures: as the temporal and spatial frequencies are mixed, the physical interpretation of such structures is complicated [50]. Furthermore, when large parametric variations in the input space are considered, the ensemble of snapshots contains many different flow regimes as the Navier-Stokes equation can exhibit bifurcating behavior for certain values of the parameters. For instance, in transonic flows, the solutions are extremely sensitive to the Mach number and angle of attack variations, and bifurcation of the solutions appears at the critical Mach number. Computing a POD basis mixing subsonic and supersonic snapshots would yield a reduced basis that does not capture structures that are peculiar to one of the two regimes, but rather an average of the various flow topologies. Although at subsonic Mach numbers the solution continuously depends on the input parameters, many flow structures are typical of a specific location in the computational domain or of a specific region in the parameter space. These observations motivate the modification of the Global ROM approach to a Local approach, which will be a central theme of this thesis work and will be further discussed in the following Chapters.

2.2. MACHINE LEARNING

Machine Learning (ML) is a generic expression used to refer to a broad class of models and algorithms that involve a certain form of learning. Machine Learning is gaining growing popularity in the field of Fluid Dynamics, and research has been focused especially on three aspects [62]: accelerating CFD high-fidelity simulations, identifying data-driven closure models for turbulence modeling, and constructing ROMs. The focus in the following paragraphs is put on the latter, in particular with respect to some of the most commonly used methods to model the low-order dynamics deriving from the dimensionality reduction of a high-dimensional dynamical system.

2.2.1. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANNs), are the most representative learning models in the Machine Learning field. ANNs have been used in a variety of applications, and have shown an unmatched ability to solve complex learning tasks in a supervised and unsupervised manner. In the field of speech recognition, natural language processing, and computer vision, for example, Deep Neural Networks typically outperform traditional algorithms and represent the state-of-the-art tools to solve machine perception tasks [42]. In particular, ANNs present the inherent capability of learning hierarchical representations of the data [18], while traditional models often rely on hand-engineered features. The success of Machine Learning should also be attributed to the development, in the last decades, of computing power and parallel computing architectures such as GPUs and TPUs. The availability of large datasets to properly train large Neural Networks is also a determining factor, as simple linear models tend to underfit when a large amount of data is employed, leading to under-utilization of the computing resources. The goal of this section is to introduce the key aspects of ANNs, and focus on a particular class of Networks, namely the Recurrent Neural Networks (RNNs). The notation, as well as some of the diagrams, are borrowed from the excellent review paper of Lipton et al. [42].

NEURONS, WEIGHTS AND BIASES

ANNs are biologically inspired computational models: the building blocks of Neural Networks are the neurons, often called nodes, connected between each other through weighted connections (the equivalent of synapses). Each neuron j carries an associated activation function $l_j(\cdot)$ which is a non-linear function of the weighted sum of the neuron inputs. In particular, denoting the weight associated with the connection from the node j' to the node j as $w_{jj'}$ the output value v_j of every neuron j is computed as follows:

$$v_j = l_j\left(\sum_{j'} w_{jj'} \cdot v_{j'}\right) \tag{2.8}$$

and it is schematically described in Fig. 2.1, where the circles represent the neurons, the edges are the connections and the sigmoid activation function is used. Other common choices for the activation function are the hyperbolic tangent $\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ and the rectified Linear Unit (ReLu) $l_j(z) = max(0, z)$. In general, the choice of this function can depend on the type of task performed by the ANN.

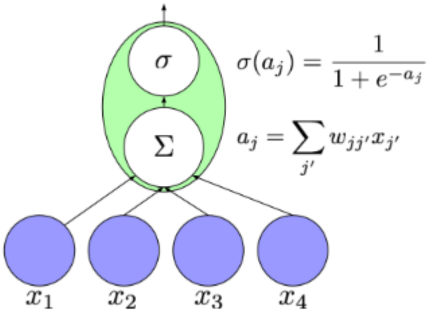

σ
Σ
$\sigma(a_j) = \frac{1}{1+e^{-a_j}}$
$a_j = \sum_{j'} w_{jj'} x_{j'}$
x_1
x_2
x_3
x_4

Figure 2.1: Simple neuron operation. [42]

Feedforward Networks, Loss Function, and Backpropagation

A wide range of architectures and topologies can be derived by choosing different ways to connect the neurons of a network, and therefore the order in which the different computations are performed between the input and the output nodes. The most classical architecture is the Feedforward Network, displayed in Fig. 2.2: between the input and the output layers, a chosen number of hidden (intermediate) layers can be present and the value of each node in a certain layer is computed progressively as a function of the outputs of the prior layer, until the final output.

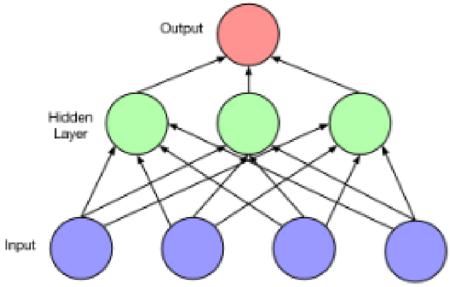


Figure 2.2: Simple Feedforward Neural Network with one hidden layer [42]

For supervised learning tasks, the Network undergoes a training phase during which the weights associated with the connection edges are updated in order to achieve sufficient accuracy on the training data. In particular, the training data is given as a list of inputs $\boldsymbol{x}$ and corresponding target $\boldsymbol{y}$ values. The Network processes the inputs with the above-explained operations and provides the output $\hat{\boldsymbol{y}}$ at the utmost layer. The distance between the target and the network output can be measured with a scalar-valued loss function $\mathscr{L}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ and the objective of the training phase can be expressed as a minimization of the loss function over the training data. The most widely used algorithm that can be used to perform this optimization task is the Backprogation introduced by Rummelhart et al. [15]. This procedure allows the computation of the gradients of the loss function with respect to the weights of the network. In particular, at each output node k the following quantity can be computed:

$$\delta_k = \frac{\partial \mathscr{L}\left(\hat{y}_k, y_k\right)}{\partial \hat{y}_k} \cdot l'_k\left(a_k\right) \tag{2.9}$$

using the chain rule, at each node j in the previous layer the following expression is derived:

$$\delta_j = l'\left(a_j\right) \sum_k \delta_k \cdot w_{kj} \tag{2.10}$$

and performing the same operation for the previous layers the σ_j values are obtained and represent the partial derivative of the loss function with respect to the node's incoming activation. As the values v_j are calculated during the forward pass, the partial derivatives of $\mathscr{L}$ with respect to the weight $w_{jj'}$ is given by:

$$\frac{\partial \mathscr{L}}{\partial w_{jj'}} = \delta_j v_{j'}. \tag{2.11}$$

Once these gradients are known, the weights can be updated using stochastic gradient descent:

$$w \leftarrow w - \eta \nabla_{\boldsymbol{w}} F_i \tag{2.12}$$

where $\nabla_{\boldsymbol{w}} F_i$ is the gradient of the loss function with respect to the weights calculated on the sample (x_i, y_i). Usually, this operation is not performed on single samples, but on mini-batches of samples. The hyper-parameter η represents the learning rate and can be adaptively tuned.

TRAINING A SUPERVISED NEURAL NETWORK ALGORITHM

During the training phase of a supervised Neural Network algorithm the weights are continuously updated in order to minimize the Loss Function over a set of training samples. During each epoch of the training, the computational loop summarized in Fig. 2.3 is performed. In the Batch Gradient Descent framework, all the training samples are processed together, with the training loss being evaluated once per epoch and the weight update being performed once per epoch. When dealing with large training datasets, samples are typically processed in multiple batches [25]: Stochastic Gradient Descent algorithms process each sample singularly in a random order, iteratively updating the weights multiple times during the same epoch. This procedure allows to reduced the memory requirements. Most modern optimizers implement the Mini-Batch Gradient Descent procedure: the batch size is therefore an hyperparameter and indicates the number of processed samples for an evaluation of the loss function and an update of the ANN weights.

Another important aspect of training is related to the formulation of the minimization problem. The loss function, in fact, is minimized across multiple epochs in order to obtain a model that optimally fits the training data. However, the goal of the ANN is to provide good prediction capabilities on unseen data (within the specific ranges) for which the targets are not already available. A common undesired overfitting behaviour can emerge from the difference between loss function definition and generalization goal of the ANN: due to the noise inherently present in the data and/or the relatively small size of the dataset, the training loss decrease over epochs does translates in poorer performance of the Network predictions on unseen test data. This behaviour is a manifestation of the fact that the weights are learning noise and unimportant relations in the data. In order to prevent the model from overfitting, part of the training data can be used as a validation set and not used to optimize the network. The loss function can be computed on the validation data, providing and indication on whether the network is able to generalize on unseen data. The training phase can be stopped when a minimum in the validation loss is reached: practically this can be done by setting a patience hyperparamter which indicates the number of epochs after which the training is stopped if a minimum in the validation loss is not reached.

The opposite pattern of overfitting, namely underfitting, can occurr when the model is not expressive enough to capture the non-linearities of the data. This behaviour typically translates in a non-decreasing training loss after a certain number of epochs. Underfitting can be simply obtained when trying to fit data distributed according to a high-order polynomial law, using a lower-order regression model (for instance a linear model).

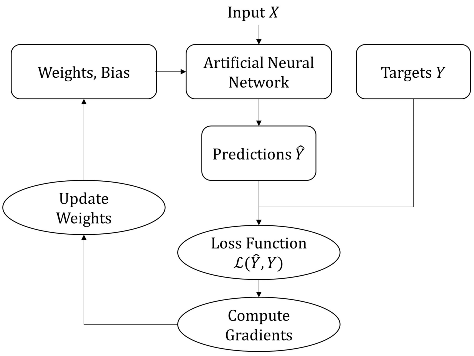


Figure 2.3: Supervised Training Flowchart of a standard ANN

2.2.2. Recurrent Neural Networks

Recurrent Neural Networks (RNN) are a special type of ANN, in which the time history of the input-target samples is taken into account. Unlike simple feedforward NN, recurrent edges that connect adjacent time-steps are present. RNNs are therefore suitable to be employed in tasks involving time-sequential data: the input of the network can be expressed as $\boldsymbol{x}^{(t)}$ and the corresponding output as $\hat{\boldsymbol{y}}^{(t)}$. If time-history effects are included, $\boldsymbol{x}^{(t-1)}$ can influence $\hat{\boldsymbol{y}}^{(t)}$ and the next time-steps. Considering the simple RNN in Fig. 2.4, the forward pass of the network can be expressed through the hidden state $\boldsymbol{h}^{(t)}$:

$$\boldsymbol{h}^{(t)} = \sigma\left(W^{\text{hx}}\boldsymbol{x}^{(t)} + W^{\text{hh}}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_h\right) \tag{2.13}$$

and therefore the output will be:

$$\hat{\boldsymbol{y}}^{(t)} = \sigma\left(W^{\text{yh}}\boldsymbol{h}^{(t)} + \boldsymbol{b}_y\right) \tag{2.14}$$

where the weights are now compactly organized in the matrices W^{hx}, W^{hh} and the biases vectors $\boldsymbol{b}$ are included.

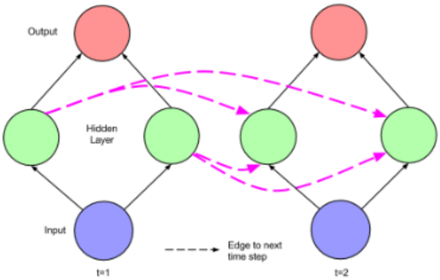


Figure 2.4: Unfolded Recurrent Neural Network [42]

In Fig. 2.4, the same network is unfolded in time to better explain the logic of the recurrence, but it is clear that the weights are shared through time steps. Training an RNN is typically more difficult than a standard Neural Network, especially when long time-range dependencies are considered [6]. This problem is related to the vanishing-exploding gradient phenomenon well described in the work of Hochreiter et al. [30]. As the expression suggests, in fact, the gradients of the error function at time t with respect to the input at a previous time τ tend to grow or decrease exponentially with the time-lag $t-\tau$: this happens because the weight of the recurrent edge remains the same at each time-step. Depending on the value of this weight the gradient will explode (if the weight is greater than one) or vanish (if the weight is smaller than one).

LONG SHORT-TERM MEMORY NETWORKS

To overcome the problems associated with vanishing gradients of standard RNNs, the Long Long Short-Term Memory Networks (LSTM) were introduced by Hochreiter and Schmidhuber [29]. In this type of Network, neurons of the hidden layers are replaced by memory cells: each cell contains a neuron with a self-connected edge of constant unitary weight. In this way, the growth of the gradient across many time steps stays contained, avoiding the above-described issues. The memory cell can be seen as a combination of simple nodes connected between each other in a specific manner, and with the inclusion of special multiplicative nodes. The anatomy of a memory cell is displayed in Fig. 2.5 where all the fundamental elements and connections are represented.

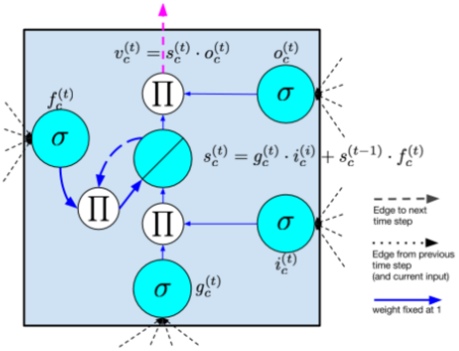

$v_c^{(t)} = s_c^{(t)} \cdot o_c^{(t)}$
$o_c^{(t)}$
$f_c^{(t)}$
σ
$\prod$
$s_c^{(t)} = g_c^{(t)} \cdot i_c^{(i)} + s_c^{(t-1)} \cdot f_c^{(t)}$
$i_c^{(t)}$
$g_c^{(t)}$
Edge to next time step
Edge from previous time step (and current input)
weight fixed at 1

Figure 2.5: Schematic representation of the memory cell of an LSTM network [42]

In particular, the following nomenclature can be introduced:

- Input node: indicated by g_c, applies an activation function to the weighted combination of the input $\boldsymbol{x}^{(t)}$ and the hidden state at the previous time-step $\boldsymbol{h}^{(t-1)}$.

- Input gate: indicated by its value i_c, it is typically a sigmoid activation on the weighted combination of the input $\boldsymbol{x}^{(t)}$ and the hidden state at the previous time-step $\boldsymbol{h}^{(t-1)}$. The name gate refers to the fact that its value is multiplied by the value of another node, and therefore when $i_c = 0$ the gate is closed there is no flow from that node, while when $i_c = 1$ the gate is open and all the flow is passed through.

- Internal state: the node s_c placed at the center of the memory cells contains the internal state, and consists of a linear activation of its inputs. A self-connected recurrent edge of constant unitary weight is used to transfer information about the previous internal state:

$$\boldsymbol{s}^{(t)} = \boldsymbol{g}^{(t)} \odot \boldsymbol{i}^{(t)} + \boldsymbol{s}^{(t-1)} \tag{2.15}$$

  where $\odot$ is the pointwise multiplication (Hadamard product) and it is performed by the multiplicative nodes $\prod$.

- Forget gate: indicated by f_c, it is used to filter the content of the previous time-step, therefore the full equation for the internal state reads:

$$\boldsymbol{s}^{(t)} = \boldsymbol{g}^{(t)} \odot \boldsymbol{i}^{(t)} + \boldsymbol{f}^{(t)} \odot \boldsymbol{s}^{(t-1)} \tag{2.16}$$

- Output gate: indicated by o_c, has a similar working principle to the input gate but multiplies the internal state s_c at the current time-step (or a $tanh$ activated version of the internal state).

The full algorithm of a memory cell computation can therefore be summarized in the following formulas:

$$\begin{aligned}
\boldsymbol{g}^{(t)} &= \phi\left(W^{\mathrm{gx}}\boldsymbol{x}^{(t)} + W^{\mathrm{gh}}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_g\right)\\
\boldsymbol{i}^{(t)} &= \sigma\left(W^{\mathrm{ix}}\boldsymbol{x}^{(t)} + W^{\mathrm{ih}}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_i\right)\\
\boldsymbol{f}^{(t)} &= \sigma\left(W^{\mathrm{fx}}\boldsymbol{x}^{(t)} + W^{\mathrm{fh}}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_f\right)\\
\boldsymbol{o}^{(t)} &= \sigma\left(W^{\mathrm{ox}}\boldsymbol{x}^{(t)} + W^{\mathrm{oh}}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_o\right)\\
\boldsymbol{s}^{(t)} &= \boldsymbol{g}^{(t)} \odot \boldsymbol{i}^{(i)} + \boldsymbol{s}^{(t-1)} \odot \boldsymbol{f}^{(t)}\\
\boldsymbol{h}^{(t)} &= \phi\left(\boldsymbol{s}^{(t)}\right) \odot \boldsymbol{o}^{(t)}
\end{aligned} \tag{2.17}$$

The effect of the gates can be analyzed for both the forward and the backward pass. During the forward pass, the LSTM input gate learns when to let the activation into the internal state and the output gate when to let the value go out. If both gates' value is zero, hence the gates are closed, the activation cannot exit the memory cell and does not influence the intermediate time-steps output. During the backward pass, the gradient is propagated across several time steps without exploding or vanishing thanks to the constant recurrent connection. Compared to standard RNNs, which possess long-term memory in terms of learning the weights and short-term memory in terms of the simple activation function, LSTMs are characterized by an intermediate memory type, through the memory cell composite unit.

2.2.3. Gaussian Process Regression

A common surrogate model that can be found in the ROM literature in combination with dimensionality reduction techniques or as a standalone model to predict scalar quantities, such as aerodynamics loads [23], is the Gaussian Process Regression. In the recent paper of Maulik et al. [46], the Gaussian Process Regression is used to evolve the latent dynamics resulting from the dimensionality reduction performed through POD, a Convolutional Auto-Encoder (CAE) and a Variational Auto-Encoder (VAE). Compared to traditional interpolation and regression methods, GPR formulates the regression task in probabilistic terms: the predictions are interpreted as the most probable outcome (mean) in sampling a probability distribution over the function space. Hence, this formulation allows for direct quantification of the uncertainty in the predictions.

The definition of a Gaussian Process can be introduced as a collection of random variables such that every finite subset of that collection is distributed according to a multivariate normal distribution [54]. The main assumption of GPR is that the output y of a function $\boldsymbol{f}$ is related to its input $\boldsymbol{x}$ through:

$$y = \boldsymbol{f}(\boldsymbol{x}) + \epsilon \tag{2.18}$$

where ϵ is a Gaussian noise distribution with zero mean and variance σ_ϵ:

$$\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2). \tag{2.19}$$

The basic idea of GPR is that any observation y of a non-linear system is inherently affected by noise, which cannot be reduced by increasing the number of observations that are made. At the same time, also the underlying function $\boldsymbol{f}$ is modeled as a random process, namely a Gaussian Process: in this way the uncertainty on its outputs is naturally included in the model. Specifically, it is possible to write:

$$\boldsymbol{f}(\boldsymbol{x}) \sim GP(\boldsymbol{\mu}(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \tag{2.20}$$

where $\boldsymbol{\mu}(\boldsymbol{x})$ is the mean function, whose estimation over a generic $\boldsymbol{x}$ represents the goal of the regression. The function $k(\boldsymbol{x},\boldsymbol{x}')$ is the covariance function, also referred to as kernel function, and measures the correlation between the function $\boldsymbol{f}$ at two different input points. A particular set of functions, namely the Radial Basis Function (RBFs) is a common choice to express the kernel. In fact, under the hypothesis of the function smoothness, the correlation between closer inputs should be increasingly high, reaching the maximum value for coincident inputs, while approaching zero in the limit of infinitely distant inputs. The RBF expression reads:

$$k(\boldsymbol{x},\boldsymbol{x}') = \sigma_0^2 e^{-\frac{||\boldsymbol{x}-\boldsymbol{x}'||}{2l^2}} \tag{2.21}$$

hyper-parameterized by the length-scale l and the variance σ_0 which can be optimized in order to increase or decrease the maximum correlation between points and the rate of decay of the correlation between further points.
Using a Bayesian approach the uncertainty on the function f can be reduced by the knowledge of some observed system outputs measured at different values of the input parameters $\{\boldsymbol{x}_t, \boldsymbol{y}_t\}$. By definition of GP, these points define the prior Multivariate Gaussian distribution:

$$\boldsymbol{y}_t = [\boldsymbol{y}_{1,t}, \boldsymbol{y}_{2,t}, ..., \boldsymbol{y}_{M,t}]^T \sim \mathcal{N}(0, \boldsymbol{\Sigma}_t). \tag{2.22}$$

The predictions $\boldsymbol{y}_p$ for new unseen inputs $\boldsymbol{x}_p$ can be made by drawing samples from the posterior distribution $p(\boldsymbol{f}|\{\boldsymbol{x}_t, \boldsymbol{y}_t\})$. In particular, since the Gaussian Process models both observation and predictions, the global joint multivariate Gaussian distribution can be written as:

$$\begin{pmatrix}\mathbf{y}_t \\ \mathbf{f}_p\end{pmatrix} \sim \mathcal{N}[\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}\mathbf{K}(\boldsymbol{x}_\mathbf{t}, \boldsymbol{x}_\mathbf{t}) + \boldsymbol{\sigma_\epsilon^2 I} & K(\boldsymbol{x}_t, \boldsymbol{x}_p) \\ K(\boldsymbol{x}_p, \boldsymbol{x}_t) & K(\boldsymbol{x}_p, \boldsymbol{x}_p).\end{pmatrix}] \tag{2.23}$$

The conditional posterior distribution $p(\boldsymbol{f}|\{\boldsymbol{x}_t, \boldsymbol{y}_t\})$ can be obtained by using Bayes theorem and can be shown to be a Multivariate normal distribution with mean:

$$K(\boldsymbol{x}_p, \boldsymbol{x}_t)[K(\boldsymbol{x}_t, \boldsymbol{x}_t) + \sigma_\epsilon^2 \boldsymbol{I}]^{-1} \boldsymbol{y}_t \tag{2.24}$$

and covariance:

$$K(\boldsymbol{x}_p, \boldsymbol{x}_p) - K(\boldsymbol{x}_p, \boldsymbol{x}_t)[K(\boldsymbol{x}_t, \boldsymbol{x}_t) + \sigma_\epsilon^2 \boldsymbol{I}]^{-1} K(\boldsymbol{x}_t, \boldsymbol{x}_p). \tag{2.25}$$

The choice of the hyper-parameters is crucial to ensure a good fit of the training data and satisfactory generalizability to unseen data. In order to find the optimal hyper-parameter values it is possible to maximize the marginal likelihood:

$$p(\mathbf{y}|\boldsymbol{x}) = \int p(\mathbf{y}|\boldsymbol{f}, \boldsymbol{x}) p(\boldsymbol{f}|\boldsymbol{x}) d\boldsymbol{f} \tag{2.26}$$

which under the hypothesis of Gaussian prior can be analytically integrated yielding the log marginal likelihood expression:

$$log p(\mathbf{y}|\boldsymbol{x}) = -\frac{1}{2}\mathbf{y}^T (K_{tt} + \sigma_\epsilon^2 \boldsymbol{I})^{-1} \mathbf{y} - \frac{1}{2} log|K_{tt} + \sigma_\epsilon^2 \boldsymbol{I}| - \frac{n}{2} log 2\pi \tag{2.27}$$

which can be maximized using a gradient-based algorithm [54]. It is interesting to clarify the role of the three terms included in Eq. 2.27. The first term contains the observation and therefore measures the data-fit quality of the model. The second term is a complexity penalty term and is only a function of the inputs and the covariance function. The first term only works as a normalization term. Most of the computational complexity lies in inverting the covariance matrix in the first term, while the computation of the derivatives of the log marginal likelihood can be performed efficiently when the inverse is known: the use of gradient-based methods for the optimization of the hyperparameters is therefore advantageous.

3

ROM Methodology

In this chapter, the methodology for the construction of the Reduced Order Models is described in depth. In particular, Section 3.1 provides an extensive description of the aircraft configuration taken into consideration, the CFD methodology adopted to generate the data and the sampling approach of the ROM. In Section 3.3 the systematic procedure to obtain scaled reduced order basis is discussed and justified. Sections 3.4 and 3.5 include the methodology to construct local reduced order basis respectively using Domain Decomposition and Clustering techniques. The modeling of the latent dynamics, obtained from the global and the local dimensionality reduction methods is described in Section 3.6.

3.1. The MULDICON unmanned combat aerial vehicle

The test case considered in this study for the generation of the high-fidelity simulations and the assessment of the proposed ROM methodology is the Unmanned Combat Air Vehicle (UCAV) test configuration of the Science and Technology Organization (STO) research task group AVT-251 [40]. This configuration is denominated Multi-Disciplinary-Configuration (MULDICON) [41] and has been used to assess the performance of high-fidelity numerical tools for the simulation of the non-linear unsteady aerodynamic phenomena characterized by strong vortical flow physics. The MULDICON UCAV has also been used as the test case for the assessment of the performance of the CNN-based ROM of Papp [51] and the POD-LSTM ROM of Bourier [9].

The MULDICON UCAV is a tailless Blended-Wing-Body (BWB) and represents the second design iteration of the predecessor SACCON UCAV (Stability and Control Configuration) [40], developed in order to improve the performance of the SACCON at higher angles of attack and to meet desirable maneuverability requirements of an agile combat aircraft [49]. The main design requirements of the MULDICON are inherited from the SACCON and are displayed in Tab. 3.1.

Design Parameter	Value
Propulsion	Single Turbofan without afternburner
Thrust to weight ratio	0.4
Payload Mass	2 × 1250 kg
Design Range	3000 km without refueling
Cruise Mach number	0.8
Cruise Altitude	11 km
Stability margin	0 – 3% MAC
CG range	5.82 - 6.00 m

Table 3.1: Main Design Parameters of the MULDICON [40]

The planform of the MULDICON is an evolution of the SACCON, with many common characteristics. Both configurations share the same leading edge sweep angle and semi-span length. The main differences are relative to the trailing edge sweep angle, which is reduced in the MULDICON with the aim to improve the control performance of the aircraft. Both platforms are displayed in Fig. 3.1, where the main geometrical differences can be appreciated.

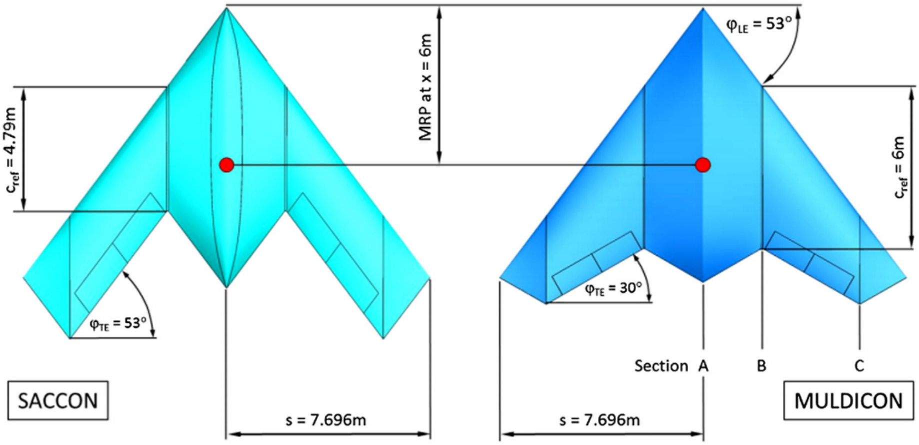


Figure 3.1: SACCON and MULDICON planforms. [49]

The geometrical reference parameters of the MULDICON configuration are also summarized in Tab. 3.2.

The design requirements of the MULDICON are similar to a "Bomber- low-level penetration" mission [41]. The mission consists of several phases: the initial climb at a low Mach number, followed by a cruise flight at high altitude, a descent phase, and a low

Reference parameter	**Symbol**	**Value**
Reference Area $[m^2]$	S_{ref}	77.8
Reference Chord $[m]$	c_{ref}	6.0
Reference Span $[m]$	b_{ref}	15.38
Moment Reference Point $[m]$	$(x_{ref}, y_{ref}, z_{ref})$	$(6.0, 0, 0)$

Table 3.2: Reference Geometrical Parameters of the MULDICON configuration [49]

altitude dash towards the target, which is then followed by a turning phase and a return through the same flight profile. The mission profile is also sketched in Fig. 3.2, in terms of altitude and Mach number trends over the mission distance. The key design points for the several phases of the mission are summarized in Table 3.3.

Parameter	Take-Off	Cruise	Approach
Altitude $[km]$	0	11	0
Mach Number [-]	0.2	0.8	0.4
Max AoA $[deg]$	$20-21$	$12-14$	-
Roll Rate $[\frac{deg}{s}]$	$30/1.1$	$90/1.7$	$90/1.7$
Pitch Rate $[\frac{deg}{s}]$	20	20	20
Yaw Rate Rate $[\frac{deg}{s}]$	$10-15$	$10-15$	$10-15$

Table 3.3: MULDICON Mission Design Points. [41]

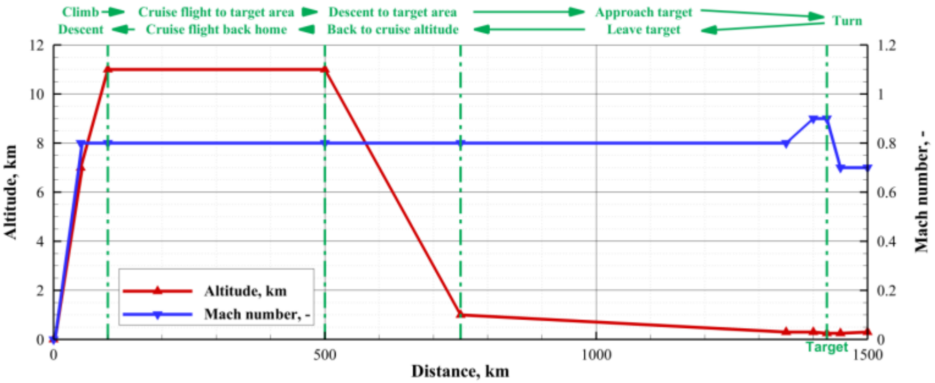


Figure 3.2: Design mission profile of MULDICON [41]

3.2. FLIGHT CONDITIONS & FLIGHT DYNAMICS

The chosen flight conditions for the evaluation of the proposed ROM strategies in this study correspond to the take-off design point in Table 3.3. In particular, the Mach number is kept constant to a low value in the incompressible regime, and a zero constant altitude is considered with standard atmospheric conditions. Large ranges of variation in the angle of attack and the pitch rate are considered. Only maneuvers in the vertical plane are considered, in order to simplify the analysis and the dimension of the parameter space. The specific flight conditions are summarized in Table 3.4.

The flight dynamics of the aircraft, for maneuvers limited to the vertical plane, can be fully described in the earth reference frame through the angle of attack $\alpha(t)$ and the pitch angle $\theta(t)$. In particular, $\alpha(t)$ describes the angle between the velocity vector and the body reference frame, while $\theta(t)$ indicates the angle between the earth reference frame and the body reference frame, as schematically depicted in Fig. 3.3.

Flight Conditions	**Symbol**	**Value**
Mach Number [-]	M_∞	0.2
Velocity $[\frac{m}{s}]$	V_∞	68.06
Pressure $[Pa]$	p_∞	101325
Density $[\frac{kg}{m^3}]$	ρ_∞	1.225
Temperature $[K]$	T_∞	288.15
AoA range $[deg]$	α	$[-2, 20]$
Max Pitch Rate $[\frac{deg}{s}]$	q_{max}	20

Table 3.4: Chosen Flight Conditions.[51]

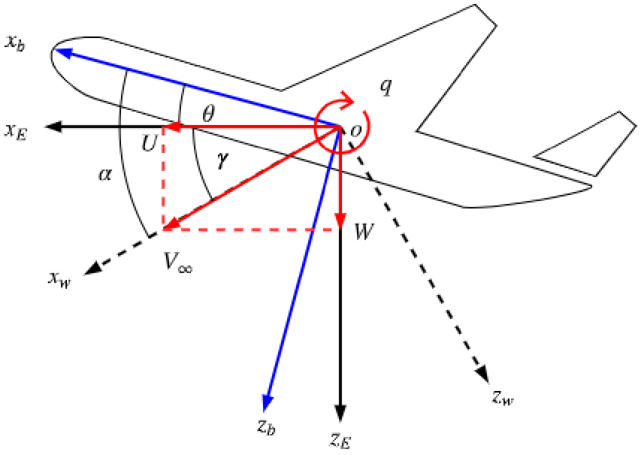


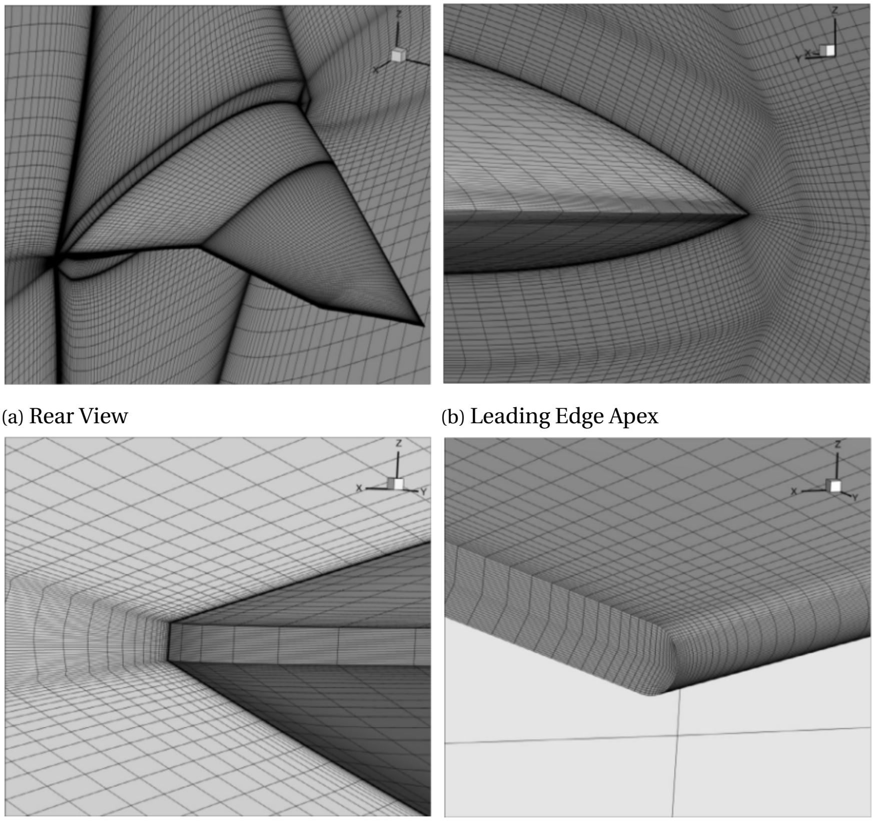
Figure 3.3: Reference frames and motion variables description. [51]

The components of the velocity in the different reference frames can be simply calculated by applying the planar rotation matrix to the constant module velocity vector V_∞. A more detailed description of this procedure can be found in the thesis of Papp [51].

3.2.1. HIGH-FIDELITY SOLUTIONS OF THE UNSTEADY AERODYNAMICS

The high-fidelity simulations for the generation of the dataset used for the construction of the ROM have been carried out during a previous computational campaign at the Netherlands Aerospace Centre (NLR), using the unsteady flow-solver ENSOLV [36]. This CFD solver numerically approximates the solution of the discretized Reynolds Averaged Navier Stokes system of equations, employing a cell-centered finite volume approach

[49] on a structured multi-block grid. The computational mesh is an O-type mesh consisting of 12 million cells in total, with 112 cells along the chord and 128 cells along the span of the wing surface. The visualization of the mesh along the wing is provided in Fig. 3.4. The employed turbulence model is the Menter's Shear Stress Transport (SST) two-equation model [47]. For the integration in time, a dual-stepping scheme with the Runge-Kutta method is used. Due to the considered flight motion, symmetry with respect to the midplane is assumed and the flow is only solved over half of the wingspan.



(a) Rear View

(b) Leading Edge Apex

(c) Trailing Edge on mid-plane

(d) Leading Edge at wingtip

Figure 3.4: Computational Mesh on the MULDICON UCAV surface for the RANS solutions computation by the ENSOLV solver. [49]

The outputs of the CFD solver, at each time-step, used for the construction of the ROM are the pressure distribution, in terms of the non-dimensional coefficient:

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty \rho_\infty V_\infty^2} \tag{3.1}$$

from which the integral force and moment coefficients (only pressure components) are derived by integration of the pressure coefficient over the wing:

$$\begin{aligned} C_F = [C_A, C_N, C_Z] &= \frac{1}{\frac{1}{2}\rho_\infty \rho_\infty V_\infty^2 S_{ref}} \sum_{i=1}^{N} \Delta F_i \\ C_M = [C_{M_x}, C_{M_y}, C_{M_z}] &= \frac{1}{\frac{1}{2}\rho_\infty \rho_\infty V_\infty^2 S_{ref} l_{ref}} \sum_{i=1}^{N} \Delta M_i \end{aligned} \tag{3.2}$$

where the contributions of each cell to the total force distribution is expressed as:

$$\begin{aligned} \Delta F_i &= C_p \frac{1}{2} \rho_\infty V_\infty^2 A_i \vec{n}_i \\ \Delta M_i &= \vec{r}_i \times \Delta F_i. \end{aligned} \tag{3.3}$$

The pressure coefficient, in particular, is given on each vertex of the wing surface and organized in a matrix format of dimensions [113,65] after a halving procedure performed to limit the memory requirements of the models [9]. The organization of the data matrix with respect to the wing geometry is illustrated in Fig. 3.5. Although the flowfield is solved over the entire computational domain, for the construction of the ROM, the number of spatial degrees of freedom is greatly reduced by only considering the surface pressure fields.

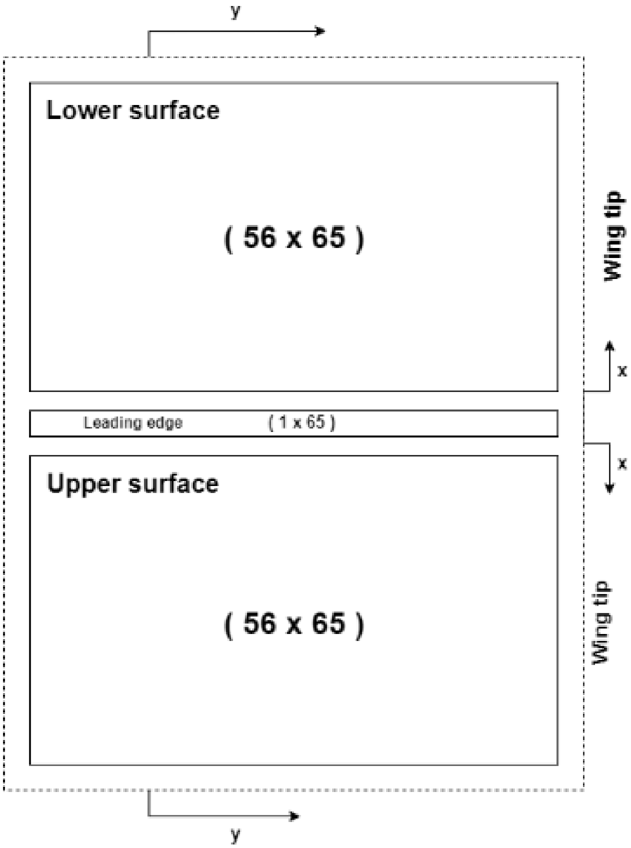
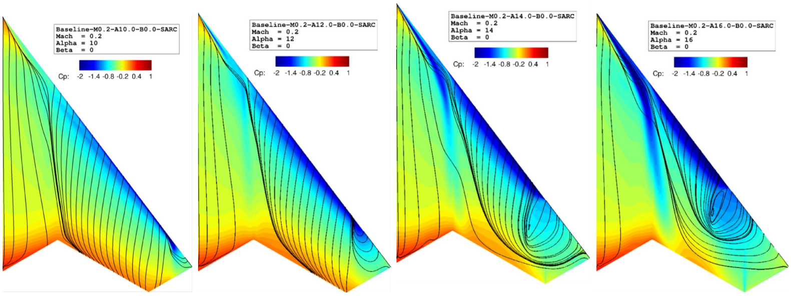


Figure 3.5: Data Structure of the Wing surface pressure coefficient. [9]

In Fig. 3.6 the CFD solutions, for different angles of attack, at steady conditions and subsonic Mach number 0.2, are displayed. These plots show the pressure coefficient distribution and the skin friction lines along the wing surface and are useful to provide an overview of the main vortical structures characterizing the flowfield at higher angles of incidence. For $\alpha = 10[deg]$, two main vortices appear on the wing: the first one originates from the wing apex and follows an outboard trajectory moving downstream, while the second one is the wing-tip vortex which is contained in a small region near the leading edge. For larger values of the incidence, the vortices increase in intensity and interest larger regions of the wing, with the main apex vortex being responsible for the generation of the large low-pressure region located approximately along the line that connects the two main sections of the wing. Furthermore, the outboard movement of the vortex towards the leading edge causes strong spanwise flow: this consideration is important because it might explain the observed fluctuations of the pitching moment coefficient that will be presented in the following Chapters, and that is at the basis of the difficulty of predicting this integral quantity encountered in the work of Bourier [9] and Papp [51].



(a) $\alpha = 10[deg]$ (b) $\alpha = 12[deg]$ (c) $\alpha = 14[deg]$ (d) $\alpha = 16[deg]$

Figure 3.6: High fidelity solutions from ENSOLV of pressure fields in steady conditions ($M = 0.2$) for different angles of attack [49].

3.2.2. Parameter Space Sampling: Training Maneuver

In order to construct a data-driven non-intrusive Reduced Order Model that effectively learns the dynamics of the system over the considered range of variation of the parameters, it is fundamental to sample the parameter space efficiently. The high-fidelity samples should, in fact, provide good coverage of the parameter space and as a consequence, a sampling strategy with good space-filling properties should be employed. Additionally, a wide range of flow phenomena should be included in the CFD dataset, in order to be present on a reduced basis: refining the sampling density in regions of the parameter space characterized by larger variations in the flowfield is often beneficial. On the other hand, the number of samples should be limited, to fully exploit the computational advantages of using a ROM. The generation of high-fidelity snapshots is in fact the most demanding step of the ROM construction.

Design of Experiments (DoE) techniques can be adopted to sample the parameter space efficiently: these strategies can generally be classified between a-priori and adaptive sampling strategies. The POD-greedy procedure [26] is an example of an adaptive strategy, where the sampling points are iteratively chosen as the points where the projection error of the full model on the reduced basis reaches its maximum. In the context of ROM for unsteady aerodynamics, the parameter space is not discretely sampled in a specific number of points, but the sampling is defined by a continuous input maneuver in the space of the motion variables. Adaptive approaches are not typically applicable in a straightforward manner. A-priori sampling strategies can be used to generate a training maneuver, with specific requirements in terms of space-filling properties: the paper of Jirasek et al. [32] compares the performance of unsteady ROMs generated using different training maneuvers.

In this work, the already existing CFD training dataset obtained by Papp [51], following an a-priori sampling technique based on Schroeder-sweeps [48], is used. In particular, an extension of the Schroeder sweep, introduced by Morelli [48], for the design of multiple input signals, is employed. The motion variables $\alpha(t)$ and $\theta(t)$, generically denoted by u are prescribed as a sum of phase-shifted sinusoids:

$$u_i = \sum_{k=1}^{M} A_k cos(2\pi f_k t_i + \phi_k) \quad t_i = i\Delta t \tag{3.4}$$

where M is the total number of frequencies used, A_k, f_k are the amplitude and the frequency of the k^{th} component of the signal, and the angles ϕ_k are the phase shift angles of the k^{th} component. The signal is evaluated at each time step (δt) of the CFD simulation. The frequencies f_k are sequentially defined from the minimum observable frequency $f_1 = \frac{1}{T}$, being T the total duration of the signal, to the chosen maximum frequency. The amplitudes are chosen in order to obtain uniform power distribution for all the signal components:

$$A_k = \frac{A}{\sqrt{N}} \tag{3.5}$$

where A is selected depending on the desired range of values taken by the motion variable. In general, if the oscillation is not centered around zero, Eq. 3.4 can be modified into:

$$u_i = A_0 + \sum_{k=1}^{M} \frac{A}{\sqrt{N}} cos(2\pi f_k t_i + \phi_k) \quad t_i = i\Delta t \tag{3.6}$$

as A_0 defines the new center of oscillation. The phase shifts ϕ_k are calculated in such a way as to minimize the so-called Relative Peak Factor (RPF):

$$RPF(\boldsymbol{u}) = \frac{(\max_i u_i - \min_i u_i)}{2\sqrt{2\sum_i u_i^2}} \tag{3.7}$$

which reaches its minimum at 1 when the multi-signal input contains only one component. Lower RPFs are in fact desirable because the associated signals can extract good information content from the system response with a relatively low amplitude perturbation. Morelli [48] describes the RPF as an efficiency measure of the signal: using low RPF he is able to describe the aircraft dynamics with a linearized model about a reference point. More detail about the optimization procedure for the computation of the phase shifts can be found in the thesis of Papp [51]. The final training signal parameters are summarized in Table 3.5: these values are in accordance with the design parameters for the take-off configuration introduced in the previous sections. The angle of attack range is between $0[deg]$ and $20[deg]$: the nominal value A_0 is therefore set to $12.5[deg]$ with a signal amplitude of $10[deg]$ in order to achieve good coverage of the parameter space. The maximum frequency is selected at $1Hz$, a value that is much larger than the pitch rate requirement of $50deg/s$: this yields a signal built from 25 sinusoids of uniformly spaced frequencies between $0.04Hz$ and $1Hz$.

The angle of attack and pitch rate signals for the training maneuvers are illustrated in Fig. 3.7.

Parameter	**Symbol**	**Value**
Signal Duration [s]	T	25
Minimum Frequency [Hz]	f_{min}	0.04
Maximum Frequency [Hz]	f_{max}	1
Number of Components	M	25
Initial AoA [deg]	α_0	12.5
Initial Pitch Angle [deg]	θ_0	0
AoA Amplitude [deg]	A_α	10
Pitch Angle Amplitude [deg]	A_θ	10

Table 3.5: Multisine input signal for training maneuver. [51]

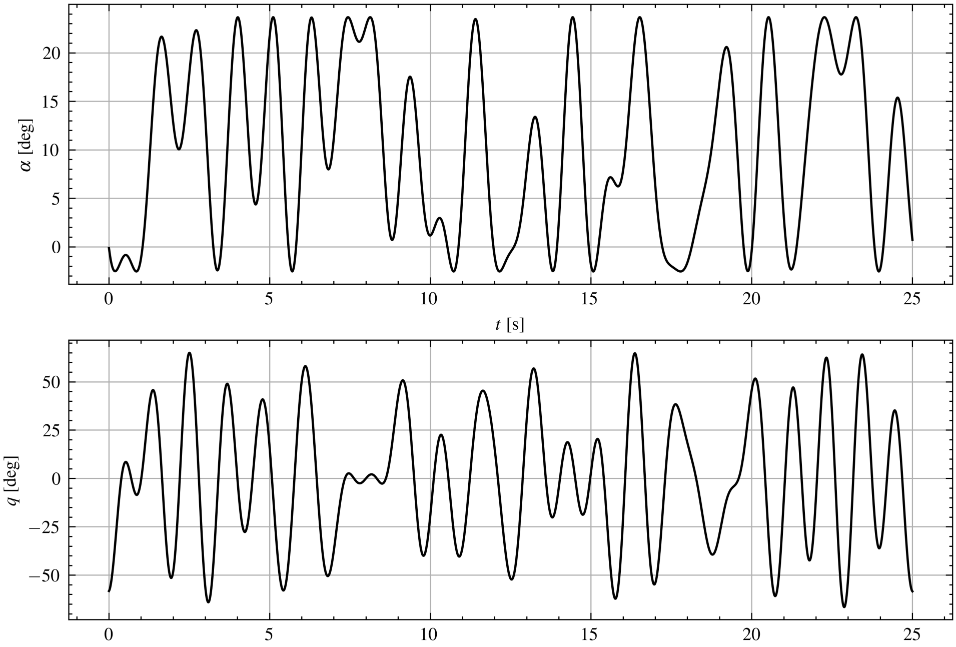
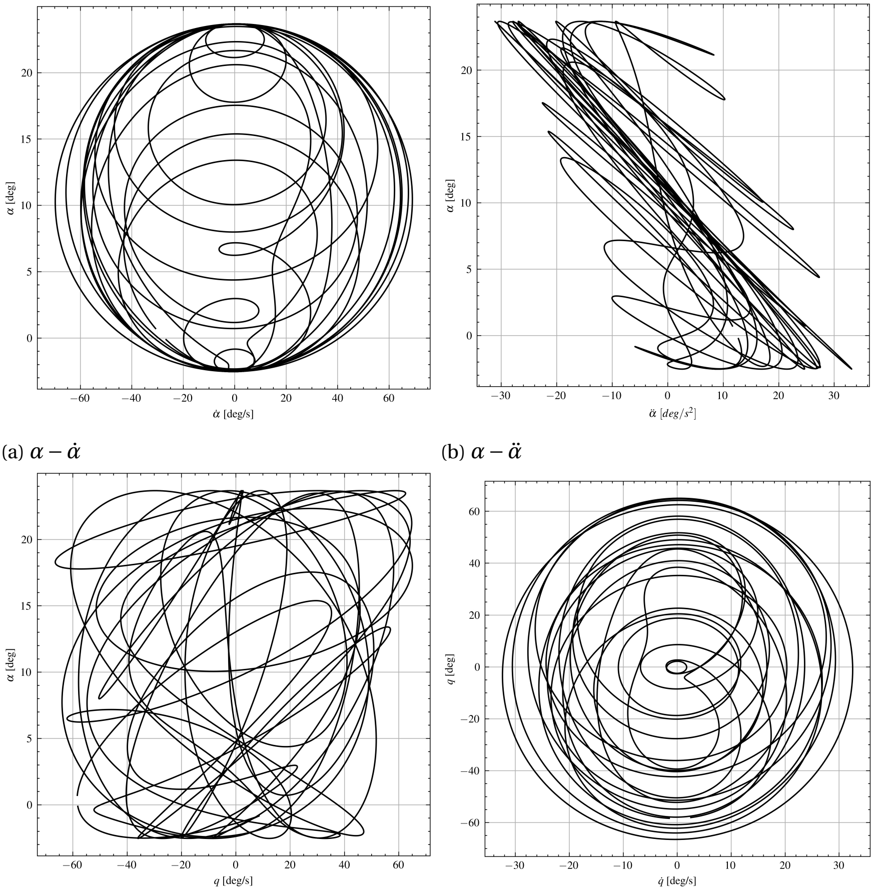


Figure 3.7: Angle of Attack and Pitch Rate signals over time for training maneuver.

The corresponding plots in the parameter space $(\alpha, \dot{\alpha}, \ddot{\alpha}, q, \dot{q})$ can be obtained by projecting the training maneuver into each of the coordinate planes. The most relevant control variable space projections are presented in Fig. 3.8. It can be seen from Fig. 3.8c, that good coverage of the $\alpha - q$ space is achieved within the design requirements of the mission. This coverage is not homogeneous but s increasingly dense towards the lower and upper bound of the angle of attack range. Nevertheless, from Fig. 3.8a, it can be inferred a poorer coverage of the central region of the $\alpha - \dot{\alpha}$ plane, corresponding to a smaller rate of change of the angle of attack.



(a) $\alpha - \dot{\alpha}$ (b) $\alpha - \ddot{\alpha}$

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

Figure 3.8: Projected regressor space coverage for training maneuver.

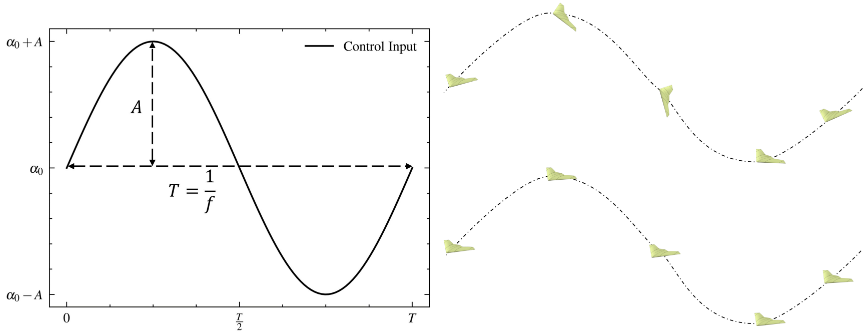
3.2.3. TEST MANEUVERS

In order to evaluate the performance of the developed ROM strategies in the prediction of unseen operating conditions, additional test maneuvers over the parameter space are computed. Two distinct types of motions are considered: pitch and plunge. Pitching motions only interest the variation of the pitch angle, leaving the absolute velocity vector fixed. Plunging motions, instead, keep a fixed angle of attack while changing the normal and axial components of the absolute velocity vector. The illustration of these two types of motion is given in Fig. 3.9b. The resulting pitching and plunging maneuvers are defined as sinusoidal inputs of the angle of attack, generically sketched in Fig. 3.9a: the

Variable	Symbol	Value											
Nominal AoA [deg]	A_0	5			10			10			15		
Amplitude [deg]	A	5			5			10			5		
Frequency [Hz]	f	0.25	0.5	1.0	0.25	0.5	1.0	0.125	0.25	0.5	0.25	0.5	1.0
Period [s]	T	4	2	1	4	2	1	8	4	2	4	2	1
Horizontal Velocity [$\frac{m}{s}$]	U_0	67.8			67.0			67.0			65.7		
Vertical Velocity [$\frac{m}{s}$]	W_0	5.9			11.8			11.8			17.6		

Table 3.6: Motion variables values for pitch test maneuvers [51]

three main parameters to describe these maneuvers are in fact the nominal value of the harmonic oscillation A_0 (center of oscillation), the amplitude of the oscillation A and its frequency f.



(a) Generic test input signal and parameters for the angle of attack.

(b) Pitch (top) and plunge (bottom) motion illustration.

Figure 3.9: Description of pitch and plunge input signal and motions.

For the pitch maneuvers, the considered values are provided in Table 3.6: the changes in the AoA are obtained by directly changing the pitch angle while keeping both horizontal and vertical components of the velocity constant.

The plunge maneuvers parameters are described in Table 3.7. The pitch angle is kept constant and therefore the sinusoidal variation in the angle of attack is obtained by changing the components of the velocity vector (while keeping the Mach number constant).

Variable	Symbol	Value											
Nominal AoA $[deg]$	A_0	5			10			10			15		
Amplitude $[deg]$	A	5			5			10			5		
Frequency $[Hz]$	f	0.25	0.5	1.0	0.25	0.5	1.0	0.125	0.25	0.5	0.25	0.5	1.0
Period $[s]$	T	4	2	1	4	2	1	8	4	2	4	2	1
Horizontal Velocity min $[\frac{m}{s}]$	U_{min}	67.3			65.7			64.0			64.0		
Horizontal Velocity max $[\frac{m}{s}]$	U_{max}	68.1			67.8			68.1			67.0		
Vertical Velocity min $[\frac{m}{s}]$	W_{min}	0.0			5.9			0.0			11.8		
Vertical Velocity max $[\frac{m}{s}]$	W_{max}	11.8			17.6			23.3			23.3		

Table 3.7: Motion variables values for plunge test maneuvers [51]

3.3. THE SCALED GLOBAL POD

The Proper Orthogonal Decomposition procedure directly applied to the pressure fields yields a set of modes that minimizes the total projection error of the high-fidelity samples on the reduced basis, in the standard L^2 optimal sense. The notion of optimality, however, depends on the metrics adopted, which in turn is dependent on the objective of the ROM. In this context, the pressure coefficient fields are defined along the wing surface on cell areas of different sizes, resulting from the discretization of the computational domain: the density of the grid points greatly increases towards the leading edge of the wing, resulting in a large scale separation of cell areas as summarized in Fig. 3.10. Hence, directly the direct POD on the pressure snapshots accounts for the all different pressure values independently from the cell areas on which they are defined, inevitably leading to relatively high projection error in the larger regions of the wing where the cell density is lower. As a consequence, this inaccuracy is amplified in the prediction of the integral force and moment coefficients, as the contribution of larger cell areas has a larger weight for the computation of these scalar quantities.

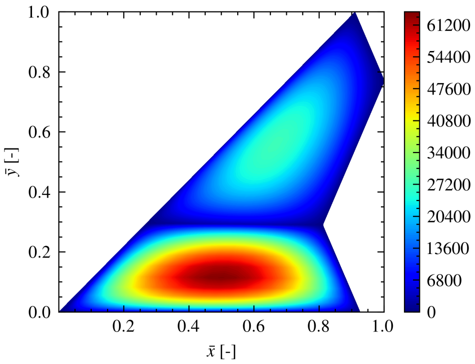


Figure 3.10: $\frac{A_{cell}}{A_{min}}$ Ratio between individual cell areas and minimum cell area along the wing upper surface

In order to account for this aspect and to capture more efficiently the larger coherent structures, an adequate scaling of the snapshots can be introduced. In the work of

Scaling Name	**Scaling Vector $\boldsymbol{w}$**
Pressure POD	Unscaled
Force POD	$\{A(x_i)\}_{i=1}^{N}$
Moment POD	$\{A(x_i) \cdot \lvert\vec{r}_i \times \vec{n}_i\rvert\}_{i=1}^{N}$

Table 3.8: Description of the spatial scaling of the snapshots .

Bourier [9], the scaling is defined in each cell as the sum of the surrounding cell areas. In this work two natural choices for the scaling are employed: the cell area distribution and the cell area distribution multiplied by the modulus of the cross product between the pressure points and the moment reference point. Pre-multiplication with these scaling quantities, in fact, respectively yields the force and moment distributions along the wing (locally oriented along the cell normal vectors). Since the integral coefficient is defined as a linear combination of these fields, an increased accuracy in the computation of these quantities is expected compared to the results of using a Pressure POD. The scaling strategies are summarized in Tab. 3.8.

In order to obtain the reduced basis corresponding to a given scaling, the pressure coefficient snapshots are pre-multiplied by the scaling vector $\boldsymbol{w}$. In matrix form this product is expressed as:

$$\tilde{\boldsymbol{u}} = diag(\boldsymbol{w})\boldsymbol{u} = \boldsymbol{W}\boldsymbol{u}. \tag{3.8}$$

The SVD of the matrix $\tilde{\boldsymbol{X}}$, containing the scaled snapshots $\{\boldsymbol{u}_{\tilde{N},m}\}_{m=1}^{N}$ generates the orthonormal modes $\boldsymbol{\phi}^{w}$ associated with the scaled quantity, i.e. pressure, force, and moment modes as displayed in Fig. 3.11. The scaling-back operation of the modes can be performed through the multiplication by $\boldsymbol{W}^{-1}$: this yields fields with the dimension units of pressure, but which are not orthogonal under the standard inner product, as the multiplication by a non-identity matrix does not preserve orthogonality of the modes. However, these pressure fields are orthogonal under the weighted inner product defined by the square of the scaling matrix:

$$\langle \boldsymbol{W}^{-1}\boldsymbol{\phi}_i^{w}, \boldsymbol{W}^{-1}\boldsymbol{\phi}_j^{w} \rangle_{W^2} = \boldsymbol{W}^{-1}\boldsymbol{\phi}_i^{w} W^2 \boldsymbol{W}^{-1}\boldsymbol{\phi}_j^{w} = \langle \boldsymbol{\phi}_i^{w} \boldsymbol{\phi}_j^{w} \rangle = \delta_{i,j}. \tag{3.9}$$

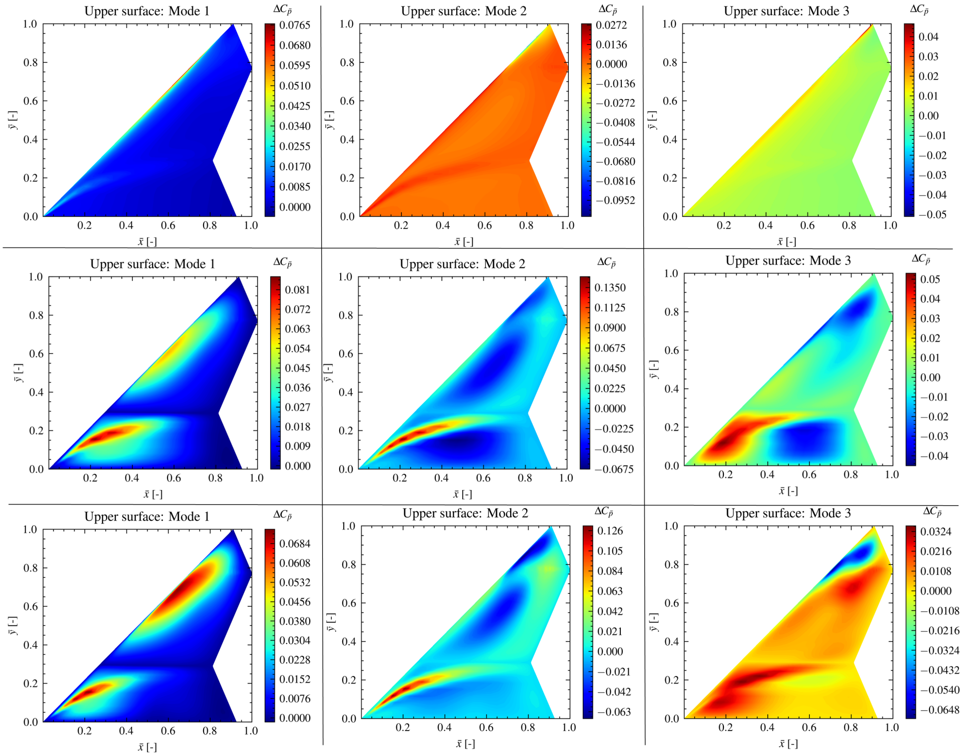


Figure 3.11: First three modes on the wing upper surface resulting from the snapshot scaling. Top: Pressure-POD modes (unscaled). Middle: Force-POD modes. Bottom: Moment-POD modes.

In order to clarify the idea of scaling and relate it to the choice of the optimality measure of the reduced space [63], it is interesting to note that the modes associated with the scaled quantity $\boldsymbol{\phi}^w$ could be obtained by modifying the metric of the POD optimization problem as follows:

$$\min_{\{\boldsymbol{\phi}_n^w(\boldsymbol{x})\}_{n=1}^r} \sum_{m=1}^{M} ||\boldsymbol{u}_{N,m} - \sum_{l=1}^{r} (\langle \boldsymbol{u}_{N,m}\boldsymbol{W}^{-1}\boldsymbol{\phi}_l^w)(\boldsymbol{W}^{-1}\boldsymbol{\phi}_l^w)\rangle_{W^2}||_{W^2}^2 \quad where \quad ||\boldsymbol{W}^{-1}\boldsymbol{\phi}^w||_{W^2} = 1 \tag{3.10}$$

These considerations serve to demonstrate the equivalence of the two procedures: performing the POD on the scaled snapshots, with the scaling defined by the matrix $\boldsymbol{W} = diag(\boldsymbol{w})$, or performing the POD directly on the pressure field but choosing the weighed inner product defined by the matrix $\boldsymbol{W}^2$ for the POD maximization problem. Obviously, it is much simpler to perform the POD on the scaled snapshots, and use the modes of the scaled quantities for the projection and the linear expansion of the reconstructed fields, performing the scaling back operation only on the final predicted fields.
Moreover, it is interesting to observe that the unconstrained optimization problem for a Goal Oriented approach to mode decomposition [11], would yield analogous results. In

fact, if the quantity of interest can be expressed as a linear function of the state variable (i.e. pressure and normal force distribution are obtained from the matrix multiplication of the pressure distribution) through the matrix $\boldsymbol{W}$, the Lagrangian associated to the Goal Oriented problem would read:

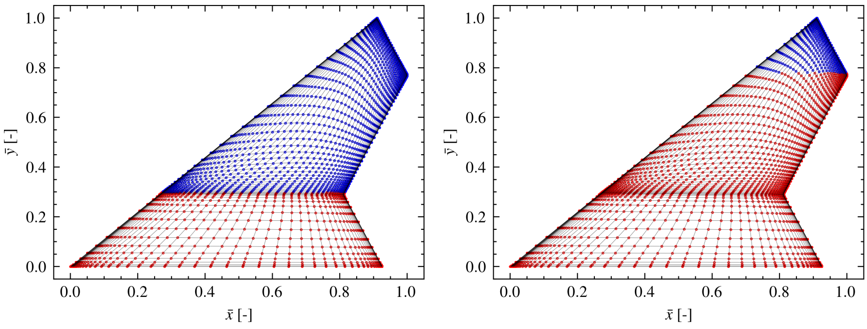
$$\mathscr{L} = \frac{1}{2}\sum_{m=1}^{M} (\boldsymbol{u}_{N,m} - \boldsymbol{\Phi}\boldsymbol{a}_m)^T \boldsymbol{W}^{\boldsymbol{T}} \boldsymbol{W}(\boldsymbol{u}_{N,m} - \boldsymbol{\Phi}\boldsymbol{a}_m)^T + \frac{\beta}{2}\sum_{j=1}^{M}(1-\boldsymbol{\phi}_j^T\boldsymbol{\phi}_j)^2 + \frac{\beta}{2}\sum_{i,j=1}^{M}(\boldsymbol{\phi}_i^T\boldsymbol{\phi}_j)^2. \tag{3.11}$$

This function is analogous to the weighed POD optimization problem if the terms associated with β, which promote the orthogonality of the resulting modes, are set to zero. In the analyzed case, orthogonality of the pressure modes is not essential, as the modes associated with the scaled quantity can be used and the predicted pressure fields can be retrieved in the final phase of the ROM through multiplication by the (inverse of the) scaling matrix.

3.4. Domain Decomposition POD

Domain Decomposition is a technique employed to separate the computational domain into a series of overlapping or non-overlapping sub-domains, and extensively used in CFD solvers to mitigate computational and memory requirements or to speed up calculations. In the context of dimensionality reduction, Domain Decomposition has often been used together with POD (DD-POD) in order to model the dynamics of a moving shock [44, 43]. In the present work, DD-POD is used to generate local non-intrusive ROMs with improved accuracy in critical regions of the domain, also by leveraging the additional flexibility in the choice of the number of modes used for reconstruction of the pressure fields in each domain. The adopted strategy follows the monolithic approach described in the paper of Baiges et al. [3]: the wing is divided into two non-overlapping domains, meaning that each grid point is assigned to only one of the two sub-domains. The border between each region has been chosen to be coincident with the vertices of the geometry of the leading edge and to run along the axial direction. This choice yields the two possible decompositions displayed in Fig. 3.12. In the first one (Fig. 3.12a), the two sub-domains are separated at $y = 0.3$: this division appears to make sense also in light of the shape of the global force and moment distribution modes in Fig. 3.11, as the coherent dominant flow structure appear to be distinct between the two domains due to the grid refinement and small grid areas near that region. In the second one (Fig. 3.12a) the division is placed at $y = 0.8$: this choice is motivated by the necessity to capture accurately the dominant flow structures at the wing-tip of the wing which has been observed in the present work and in the results of Bourier [9] to be responsible for high projection error of the global POD-LSTM model in that region of the computational domain. It should be noted that although only the upper surface of the wing is displayed, each of the two regions extends on the lower surface.

Two SVDs are performed independently on the two snapshot matrices resulting from the decomposition of each pressure coefficient field: the time complexity and memory requirements of these operations depend on the sizes of the snapshots matrices and therefore on the number of snapshots and grid points considered. A summary of these



(a) $DD_{0.3}$

(b) $DD_{0.8}$

Figure 3.12: Geometry decomposition of the wing upper surface. The lower surface is continuously attached to the upper surface at the leading edge.

Decomposition Type	Number of grid points N	Number of Snapshots m
$DD_{0.3}$		
Domain 1	5570	5666
Domain 2	1775	5666
$DD_{0.8}$		
Domain 1	2201	5666
Domain 2	5144	5666

Table 3.9: Decomposition types and corresponding datasets dimensions.

parameters is presented in Table 3.9.

In Fig. 3.13 the first three modes shapes on the upper surface of the wing are displayed for the two types of decomposition. The bottom row highlights how isolating the wing-tip regions leads to modes that capture more accurately the fluctuations in these specific regions. In particular, observing the evolution of the time coefficient associated with the second mode of the wing-tip subdomain in Fig. 3.14, an "on-off" behavior can be detected: the value is close to zero everywhere, except at high angles of attack. Although POD modes rarely resemble true physical evolving phenomena, in this case, the second mode carries the information of wing-tip vorticity generated at high incidences. As a consequence, the projection error deriving from the use of a reduced basis is expected to decrease when compared to the global POD performance.

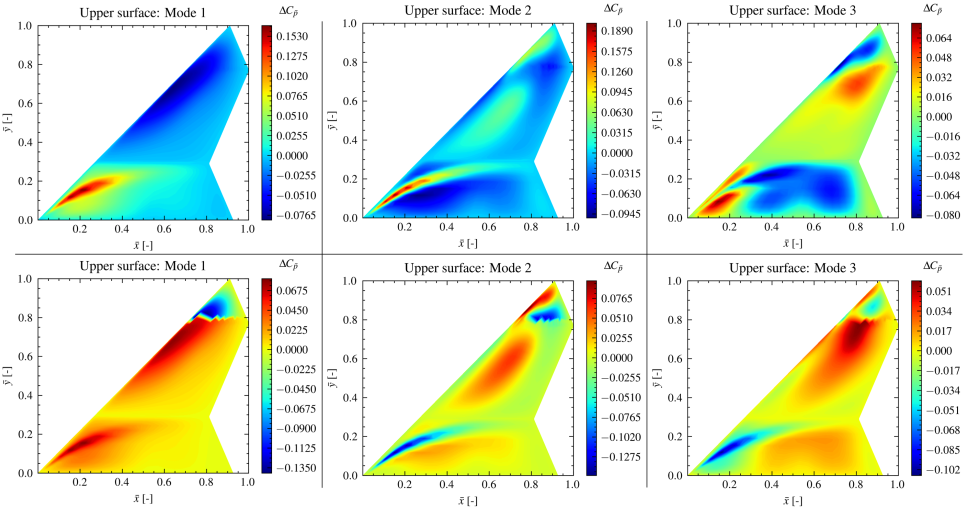


Figure 3.13: First three modes on the wing upper surface for two different geometry decompositions. First Row: $DD_{0.3}$. Second Row: $DD_{0.8}$

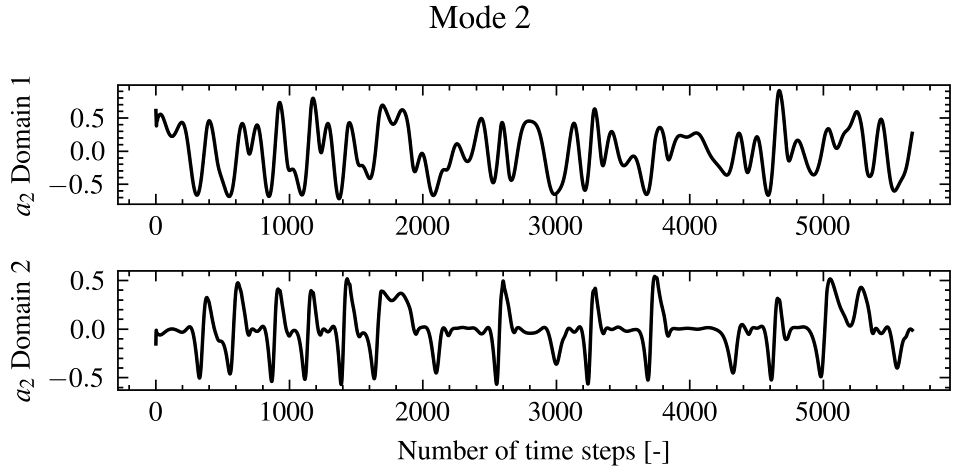


Figure 3.14: Second POD coefficients over the training maneuver for the $DD_{0.8}$ decomposition

The projection of the high-fidelity decomposed snapshots on the two reduced basis yields the time coefficients corresponding to the evolution of the pressure fields in the reduced spaces: two training datasets (one for each domain) consisting of 5666 samples are then used to train the low-order regression model, typically represented by a Long Short Term Memory Network in a similar fashion to the global POD-LSTM model. The offline stage of the ROM is completed when convergence in the training procedure is achieved, as the trained ANNs and the POD basis are used to predict the local pressure fields at unseen operating conditions for a given maneuver. The full domain pressure fields are recovered simply through the composition of the local pressure fields: if the predictions of the ROM are accurate enough near the boundary of the decomposition, no discontinuities appear and an additional interpolation step is not needed. The entire procedure of the DD-ROM is schematically illustrated in Fig. 3.15.

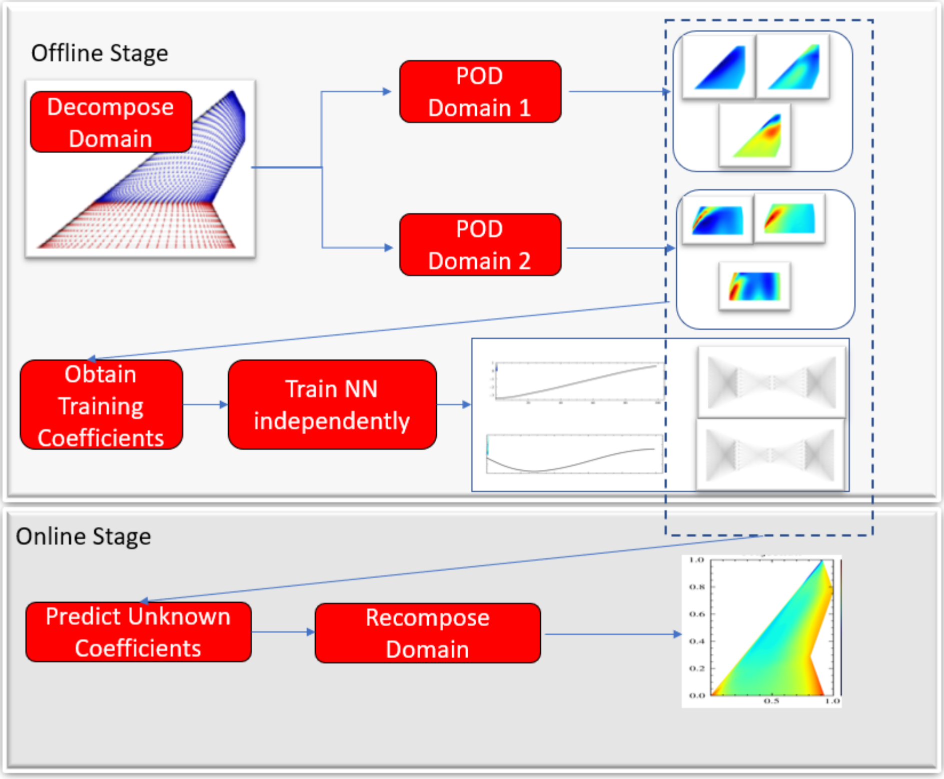


Figure 3.15: DD-POD based ROM flowchart. The flowchart does not include the parameter space sampling and the CFD simulation step for the sake of conciseness.

3.5. CLUSTERED POD: FROM GLOBAL TO LOCAL

The development of a Reduced Order Model through a linear dimensionality reduction technique such as a POD, allows for the identification of a low-dimensional global basis which is assumed to carry the most energetically important features present in the data. However, dynamically important but low energetic structures are often neglected by the global basis, leading to large projection errors and poor performance of the resulting ROM. Local spatial approaches such as the DD-POD described in the previous chapter can improve the low-order approximation of the flow physics in specified regions of the computational domain, through the definition of local basis: these approaches, however, demonstrate their full potential only when applied to problems characterized by a clear spatial separation of the flow physics. Another local approach to dimensionality reduction can instead be motivated by the inefficiency of the global basis to capture the distinct dominant physical features corresponding to large variations of the input variables. The Navier-Stokes equations, present a strong dependency on the parameter's instances and can exhibit bifurcating solutions. The dominant POD modes are therefore in general a mixture of the dominant coherent structures appearing in different subregions of the parameter space, rather than authentic physically significant phenomena. The idea of Clustered-POD (CPOD) is to generate reduced bases that are local in the parameter space, ideally grouping solutions with similar patterns to the same set of modes. The first example of CPOD-based ROM has been used in literature [34] to partition the state space into clusters corresponding to several representative problems, in combination with a probabilistic model based on Markov processes to model the transition dynamics between clusters, however without identification of the local dynamics within a cluster. Another notable example comes from the work of Hess et al. [27]: a Cluster-based ROM is built in order to identify the bifurcations in the solutions of the incompressible Navier-Stokes characterized by variation in the Reynolds number. A similar methodology is described here, although several modifications in the ROM development are proposed. Moreover, the application to unsteady parametric aerodynamics requires adequate adaptation of several aspects of the model. In particular, the procedure to construct the ROM can be divided into four steps, each one described more in detail in a dedicated paragraph:

1. Cluster the training maneuver snapshots, depending on a compressed representation of each snapshot. This step includes the choice of the number of clusters based on an a-posteriori error indicator.

2. Construct a local reduced base for each cluster.

3. Model the dynamics in each cluster with a low-order surrogate model.

4. Assign a new testing parameter to each cluster.

It is important to note that in this description, the phase of the ROM construction related to the generation of data has been left out to avoid redundancy, as this step is in common with the other models presented in this report without further modifications. Nevertheless, it must be emphasized that the entire model only relies on the information present in the training snapshots, and thus the ROM capabilities to capture accurately the dominant features of the solution at testing operating points depend on whether similar features are present in the training snapshots.

3.5.1. Snapshot Clustering

Let $\mathscr{S} = \{\boldsymbol{u}_{N,m}\}_{m=1}^{M}$ be the set of training snapshots corresponding to the sampling set $\Theta_M^{Tr} = \{\vec{\theta}_m\}_{m=1}^{M}$ in the p-dimensional parameter domain Θ: $\vec{\theta}_m \in \mathbb{R}^p$. The clustering procedure aims to divide the snapshot set into k clusters $\{\mathscr{C}_{k'}\}_{k'=1}^{M}$ in such a way that an optimal division is achieved. The definition of optimality, once again, drives the strategy to determine the clustering and to tune the choice of the algorithm hyperparameters.

The algorithm used to perform this operation is the k-means clustering, which is an unsupervised classification algorithm that accepts as inputs the number of clusters and a metric to compute the distance between points in the clusters. In particular, the k-means is an iterative algorithm that is initiated by a random choice of k cluster centroids $c_1, ..., c_k$ among the data-points in $\{\boldsymbol{u}_{N,m}\}_{m=1}^{M}$, and at each step minimizes the following functional:

$$E(c_1, ..., c_k) = \sum_{m=1}^{M} \sum_{k'=1}^{k} I(\boldsymbol{u}_{N,m} \in \mathscr{C}_{k'}) d(\boldsymbol{u}_{N,m}, c_m) \tag{3.12}$$

where $d()$ measures the distance between the centroid and the data point and is typically chosen to be the euclidean distance while I is equal to one of zero depending on whether the data point belongs to the cluster. In fact, each data point is assigned to the cluster for which the distance from the its centroid is minimum. At each iterative step, the position of the cluster centroids is updated until convergence is reached. The sensitivity of the method from the clusters-center initialization is mitigated by restarting the algorithm with random initialization several times. In this context, it is enough to highlight how the results of the clustering depend on the specified number of clusters and the choice of the distance metric, which are critical for the performance of the resulting ROM.

Clustering Features and Distance

The guiding logic behind the method is to cluster the snapshots according to their similarity, in such a way that snapshot belonging to the same cluster exhibit common features. The most straightforward approach would be to simply define $d()$ as the Euclidean distance in the space of the high-fidelity snapshots V_N, thus identifying the clustering features as the entire pressure fields. This choice presents two main issues:

- the dimension N of each snapshot corresponding to the number of cell points in the computational domain, makes the optimization problem associated with the k-means algorithm computationally expensive.

- During the online phase, assigning test points, for which the full prediction of the pressure field is not known a priori (being the ultimate output of the ROM), to a cluster is not directly possible, as the assignment criterion is based on the computation of the above-mentioned distance.

An alternative selection of the clustering features can be made by directly using the input parameters, thus specifying the Euclidean distance in the parameters space Θ. This approach overcomes the limitation of the previous strategy since the parameter space is generally low dimensional (and only a limited set of influential parameters can generally be used) and the input parameters are known a priori during the online phase. Although this choice leads to a significant simplification of the model, it is generally sub-optimal, unless a clear dependence of the full solution on the input parameters is known: this is the case when the solutions exhibit clearly distinct behavior for small differences in the parameter's value (i.e. bifurcations). For example in the case of transonic flow over an airfoil, the appearance of the shock at the critical Mach number divides the parameter space into a subsonic regime $M < M_{crit}$ and a supersonic regime $M > M_{crit}$. In the present case (the incompressible unsteady aerodynamics over the UCAV MULDICON wing), the parameter space is continuously sampled with a training maneuver with a continuous dependence of the solution on the input parameter: it is therefore impossible to optimally establish a priori the desired clustering directly in the parameter space. In addition to that, the challenge of choosing which parameters have the most influence on the solution (quantifying it through weights) must be addressed: using the entire set of parameters is not particularly beneficial as several parameters do not heavily affect the solution. On the other hand, using only one parameter (for example the angle of attack) can lead to neglecting important dependencies.

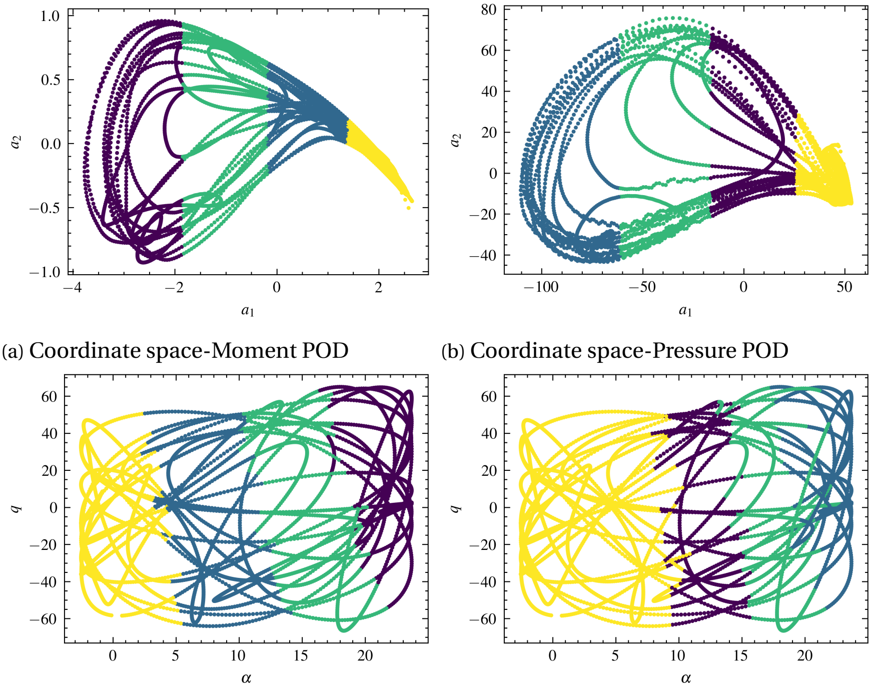
In order to overcome these limitations, a compressed representation of the snapshots can be considered as clustering features: the Global Proper Orthogonal Decomposition is used to determine this representation. In particular, the global POD basis is computed, allowing to express of each training snapshot as:

$$\boldsymbol{u}_{N,m}(\boldsymbol{x}) = \sum_{n=1}^{N} \boldsymbol{a}_{n,m} \boldsymbol{\phi}_n(\boldsymbol{x}) \tag{3.13}$$

where $\{\boldsymbol{\phi}_n(\boldsymbol{x})\}_{n=1}^{N}$ represents the POD basis (and is a basis for the solution space V_N), while the coefficients $\{\boldsymbol{a}_{n,m}\}_{n=1}^{N}$ are obtained by projection of the snapshots on the reduced basis. These reduced coordinates, do not only serve as a compressed representation but are also ranked in terms of energy content and it is, therefore, possible to consider a truncated version of the reduced coordinates as clustering features: $\{\boldsymbol{a}_{n,m}\}_{n=1}^{r}$ where r represent the truncation index. The distance d is thus defined as the Euclidean distance in the subspace $\mathscr{A}^r \in \mathbb{R}^r$ spanned by the first r reduced coordinates. It must be noted that because of the energy ranking of the modes, the coordinates carry decreasingly less information, and therefore the weighted Euclidean distance is employed:

$$d(\boldsymbol{a}_1, \boldsymbol{a}_2) = \sqrt{\sum_{i=1}^{r} w_i (a_{1_i} - a_{2_i})^2} \tag{3.14}$$

where w_i are taken to be equal to the eigenvalue of the diagonal matrix of the Singular Value Decomposition of the snapshot matrix, which is an indication of the "pressure energy" contained in each mode (if the snapshots represent turbulent fluctuations then the eigenvalues indicate the amount of Turbulent Kinetic Energy contained in each mode). Although this method, compared to the snapshot direct clustering, solves the problem associated with the computational cost of running the k-means algorithm, during the online phase of the ROM, the global reduced coordinates of a testing parameter are not known but have to be predicted in order to assign the testing point to a cluster. To circumvent this issue, a surrogate model to map the input parameters to the global reduced coordinates is used: the details of this procedure are described in the following sections. In Fig. 3.16 the results of the k-means clustering algorithm on training snapshots are displayed. In particular, the clustering features are chosen as the first two ($r = 2$) reduced coordinates of the global POD: on the left column, the POD is performed on the pressure distribution while on the right column on the moment distribution (as described in Section 3.3). The plots in the first row show the clustering in the (truncated) reduced coordinate space: the separation between clusters appears almost vertical since the first coordinate (on the x-axis) has the most important in the distance definition. The bottom row shows the same clustering but in the parameter space of the angle of attack and the pitch rate. It is interesting to observe how the clustering based on the pressure POD produces in the parameter domain one large cluster for moderate angles of attack (in the linear regime) and three smaller clusters for higher angles of attack (in the non-linear regime).



(a) Coordinate space-Moment POD (b) Coordinate space-Pressure POD

(c) Input space-Moment POD (d) Input space-Pressure POD

Figure 3.16: K-means Clustering visualization in the input space ($\alpha - q$) and in Global POD coordinates ($a_1 - a_2$) for two distinct POD decompositions.

Optimal Number of Clusters

The number of clusters is a key parameter of the clustering process which needs to be specified in advance. When the training data is amenable to clustering and it is possible to visually partition a compressed representation of the snapshot, the choice of the optimal number of clusters k_{opt} is straightforward. However, in most cases, the parameter sampling (DoE) is designed in such a way to maximize regressor space coverage and uniformity (as it is visible in the bottom plots of Fig. 3.16): as a result, due to the continuous dependency of the solution from the parameter values, also the low-order representation of the data (top plots of Fig. 3.16) presents a rather homogeneous coverage and it is difficult to distinguish clearly predefined clusters. It is therefore important to define a metric to unambiguously determine the optimal number of clusters, which can be generalized to different ROM datasets. The standard choice in literature requires the computation of the k-means variance indicator:

$$Var(k) = \sum_{k'=1}^{k} \sum_{\{m:\boldsymbol{a}_{n,m}\in\mathscr{C}_{k'}\}} |a_{n,m} - c_m|^2 \tag{3.15}$$

where the distance is computed in the space of the reduced coordinates and therefore here c_m indicates the reduced coordinates of the cluster centroid. This quantity monotonously decreases with the number of clusters k and in the limit of the number of clusters being equal to the number of snapshots it reaches its minimum at zero, resulting in the local bases coinciding with each snapshot: clearly not optimal. Moreover, for non-intrusive ROM applications, the surrogate model for the reduced dynamics is dependent on the number of samples in the dataset: after clustering the data is partitioned into smaller local datasets on which a local model is trained (in the case of a Neural Network). Limiting the number of clusters is therefore crucial for the entire ROM development. The criterion associated with the k-means variance is commonly referred to as "Variance Elbow", in fact, k_{opt} is chosen as the smallest integer for which the variation in the derivative of $Var(k)$ is smaller than a specified tolerance, thus visually corresponding to an elbow in the $Var(k)$ vs k plot. The "Variance Elbow" criterion is computationally efficient requiring only a priori evaluation of the clustering distances on the low-order version of the snapshot set. At the same time, the choice of k_{opt} depends on the tolerance adopted for the minimum variation of the slope and can therefore be a source of ambiguity. Furthermore, there is no guarantee that the k_{opt} determined by this criterion is optimal in terms of the performance of the resulting ROM, as this would require a certain form of a-posteriori evaluation of the performance of the local base. Motivated by these observations, a different method for the determination of k_{opt} is proposed, based on an a posteriori indicator of the total projection error produced by the local bases on a validation maneuver. In particular, for each $k = 1,...,k_{max}$ the snapshot set is clustered into k clusters following the procedure described above. Afterwards, a set of of k local POD bases is computed $\{\boldsymbol{\psi}_{n_{k'},k'}\}_{n_{k'}=1}^{N_{k'}}$ (for more details about this step the reader can jump to the next paragraph). A validation maneuver is considered, for which the snapshot set $\{\tilde{\boldsymbol{u}}_{N,m}\}_{m=1}^{\tilde{M}}$ is known, and therefore also the global POD coefficients $\{\tilde{\boldsymbol{a}}_{n,m}\}_{m=1}^{\tilde{M}}$ can be obtained by projection of the snapshot onto the global POD basis. The knowledge of the global coefficients allows for the partition of the validation snapshot set into k disjoint sets, as each snapshot is assigned to a cluster $\mathscr{C}_{k'}$ by the k-means algorithm (in predictive mode). The optimization metric is based on the total Mean Absolute Error of projection of the validation snapshots on the truncated local bases:

$$\begin{aligned} MSE(k) &= \sum_{k'=1}^{k} \sum_{\{m:\tilde{\boldsymbol{a}}_{n,m}\in\mathscr{C}_{k'}\}} |\tilde{\boldsymbol{u}}_{N,m} - \sum_{l=1}^{r} (\tilde{\boldsymbol{u}}_{N,m}\cdot\boldsymbol{\psi}_{l,k'})(\boldsymbol{\psi}_{l,k'})|^2 \\ k_{opt} &= \{k \mid MSE(k) = \min_{k'} MSE(k')\} \end{aligned} \tag{3.16}$$

where r indicates the truncation index and it is typically set to 5. The smallest k for which a minimum is reached by the error metrics is chosen as the optimal number of clusters. It is important to motivate the use of a validation maneuver: simply evaluating the projection error on the training maneuver does not provide such an accurate indication of how the local basis would perform when used in the predictive mode for unseen test data, in fact, all the training snapshots are contained within the local basis (because included into the local snapshot matrices) and the projection error on the snapshot located at the border of the clusters is contained. When a testing snapshot is located close to the border of the clusters, an increase in the projection error is instead observed due to the poor extrapolation capabilities of the local basis when predicting a data point located outside of the cluster range. Additionally, the validation maneuver should be designed in such a way as to cover a wide range of flow conditions and therefore to pass through every cluster: for this reason, in this study, the maneuver with the largest angle of attack and pitch rate amplitudes is selected. Compared to the Variance Elbow method, the described procedure is computationally more demanding as it requires the computation of the set of basis functions corresponding to each cluster. Nevertheless, this process is only carried out once during the online phase, as the optimal number of clusters is a property of the snapshot set.

3.5.2. Local Basis Computation

Following the optimization procedure to determine the optimal number of clusters, the k_{opt} reduced bases can be determined with the Proper Orthogonal Decomposition (this step can be skipped if the bases have already been computed during the optimization process). The local snapshot matrix $\boldsymbol{X}_k$ is assembled in each cluster $\mathscr{C}_k$ $k = 1, ..., k_{opt}$, containing $|\mathscr{C}_k|$ (scaled) snapshots. The local POD minimization problem now reads:

$$\min_{\boldsymbol{\psi}_{1,k},...,\boldsymbol{\psi}_{r_k,k}} \sum_{\tilde{\boldsymbol{u}}_{N,m} \in \mathscr{C}_{\|}} |\tilde{\boldsymbol{u}}_{N,m} - \sum_{l=1}^{r_k} (\tilde{\boldsymbol{u}}_{N,m} \cdot \boldsymbol{\psi}_{l,k'})(\boldsymbol{\psi}_{l,k'})|^2 \tag{3.17}$$

subject to the orthonormality of the modes. The solution is obtained from the SVD of the snapshot matrix:

$$\boldsymbol{X}_k = \boldsymbol{\Psi}_k \boldsymbol{\Sigma}_k \boldsymbol{\Omega}_k \quad k = 1, ..., k_{opt} \tag{3.18}$$

as the first r_k columns of the left singular vector matrix $\boldsymbol{\Psi}_k$ denoted by $\{\boldsymbol{\psi}_{l,k}\}_{l=1}^{r_k}$. The number of modes contained in each cluster basis can be chosen according to the eigenvalue spectrum of the diagonal matrix $\boldsymbol{\Sigma}_k$. In Fig. 3.17 the first three modes shape on the upper surface of the wing are displayed for the four different clusters.

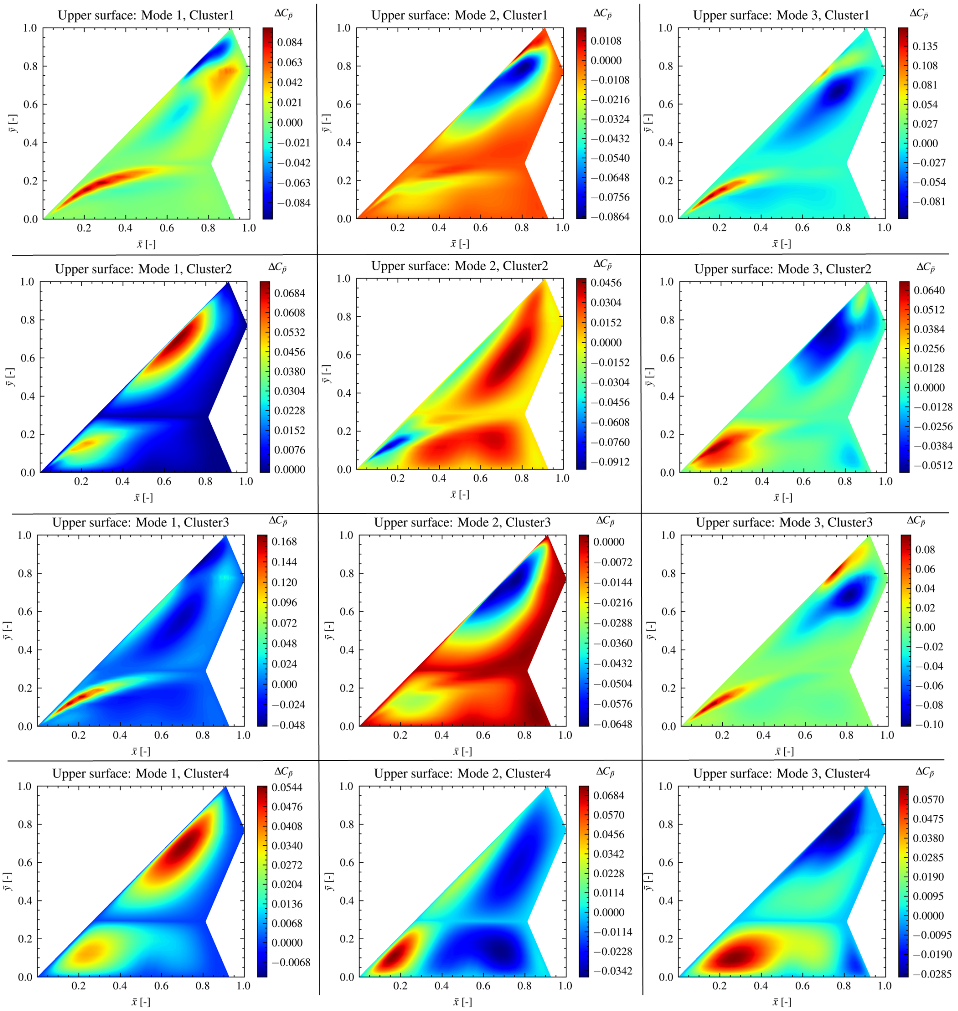


Figure 3.17: First three modes on the wing upper surface for each cluster local basis. Each row corresponds to a cluster. The cluster numbering follows Fig. 3.16. The POD is performed on the Moment scaled snapshots.

3.5.3. SURROGATE MODELLING OF CLUSTER LOCAL DYNAMICS

The identification of several regions in the parameter space, and the computation of a local reduced basis over each cluster, allow for a substantial reduction of the degrees of freedom of the high-dimensional snapshots into a low-order representation of the global features through the POD reduced coordinates. In the Global POD, these coordinates are obtained by projecting the snapshots into the global basis: for the CPOD, the projection is performed cluster by cluster, only considering the snapshot that belongs to a cluster, onto the cluster local basis:

$$\boldsymbol{a}^{k}_{n,m} = \boldsymbol{u}_{N,m} \cdot \boldsymbol{\psi}_{n,k} \quad k = 1, \dots, k_{opt} \tag{3.19}$$

Cluster Number	Number of grid points n	Number of Snapshots m
1	5666	1336
2	5666	1286
3	5666	1159
4	5666	1185

Table 3.10: Clustering using 4 cluster: datasets dimensions.

where the index k denotes the cluster number, and it is used to distinguish local reduced coordinates from global ones. The projection, therefore, gives rise to k_{opt} low-order local dynamics:

$$\boldsymbol{a}^k_{n,m} = f^k(\vec{\theta}^k) \quad k = 1,..., k_{opt} \tag{3.20}$$

where each input parameter vector is mapped to its local reduced representation. Intrusive methods, such as the Galerkin Projection [57], identify the dynamics with the governing equations (projected on the basis), and therefore do not require different definitions of the dynamics depending on the cluster. Nonintrusive methods, however, approximate the true underlying dynamics with a surrogate model, typically based on data-driven methods. In this case, it is more convenient to define the surrogate dynamics separately within each cluster, as the projection onto different bases inevitably produces discontinuous hypersurfaces over the parameter domain. Training a Neural Network on discontinuous data, presents several challenges, as the information of a cluster transition has to be learned by the network during training, and the weights must be able to model different local dynamics. Additionally, ANNs are typically continuous maps from the input to the output space and achieve better performance when approximating a continuous function. In light of these observations, in this work, the local low-order dynamics are approximated by an LSTM Network (although other models such as the Multi-Layer Perceptron and the Gaussian Process Regression are discussed) with the same architecture in each cluster but trained on the k_{opt} different input-output pairs resulting from the projection in Eq. 3.19. A detailed description of the training process and the surrogate models is extensively presented in the following sections. However, it is important to observe that the clustering of the data produces several datasets with smaller sample sizes: this aspect should be taken into account when analyzing the results of the Network prediction, as Deep Learning models tend to overfit on small datasets and the training sample size is a critical factor in the convergence of the training procedure. On the other hand, by clustering the parameter space, each local regression task is defined on a smaller range of the input space: good model accuracy can therefore be achieved with little data. In Tab. 3.10 the local datasets sample sizes are given for the clustering displayed on the left side of Fig. 3.16.

Once the local dynamics are modeled, the ROM is ready to be deployed for predictions of the high-dimensional pressure fields, and the most computationally demanding phase of the ROM construction is concluded. The offline stage complete algorithm is given in Alg. 1 and Alg. 2 where the Global and Local ROM computation steps are divided for clarity.

Algorithm 1 Offline Stage: Global ROM computation

```
Input: Snapshot set $\{\boldsymbol{u}_{N,m}\}_{m=1}^{M}$ at sampling points $\Theta_{M}^{Tr}$
Output: Global modes $\{\boldsymbol{\phi}_n(\boldsymbol{x})\}_{n=1}^{r}$, coefficients $\{\boldsymbol{a}_{n,m}\}_{n=1}^{r}$, eigenvalues $\{\sigma_n\}_{n=1}^{r}$, map $\mathscr{F}(\vec{\theta}) = \boldsymbol{a}$
Compute Global POD on snapshots set $\{\boldsymbol{u}_{N,m}\}_{m=1}^{M}$
Store First $r$ modes $\{\boldsymbol{\phi}_n(\boldsymbol{x})\}_{n=1}^{r}$, coefficients $\{\boldsymbol{a}_{n,m}\}_{n=1}^{r}$, normalized eigenvalues $\{\sigma_n\}_{n=1}^{r}$
Train NN (LSTM) model $\mathscr{F}$ with $\{(\vec{\theta}, \boldsymbol{a}_n)_m\}_{m=1}^{M}$ samples
```

Algorithm 2 Offline Stage: Local ROM computation

```
Input: Snapshots set $\{\boldsymbol{u}_{N,m}\}_{m=1}^{M}$, global coefficients $\{\boldsymbol{a}_{n,m}\}_{n=1}^{r}$, eigenvalues $\{\sigma_n\}_{n=1}^{r}$, Validation maneuver $\{\tilde{\boldsymbol{u}}_{N,m}\}_{m=1}^{\tilde{M}}$
Output: $k_{opt}$, Local basis $\{\boldsymbol{\psi}_{l,1}\}_{l=1}^{r_1}, \dots, \{\boldsymbol{\psi}_{l,k_{opt}}\}_{l=1}^{r_{k_{opt}}}$, Local surrogate models $\mathscr{F}_1, \dots, \mathscr{F}_{k_{best}}$

for $k = 1, \dots, k_{max}$ do
    k-Means Clustering on scaled global coefficients $\{\boldsymbol{a}_{n,m} \cdot \sigma_n\}_{m=1}^{M}$
    Assign Validation snapshot to clusters based on $\{\tilde{\boldsymbol{a}}_{n,m} \cdot \sigma_n\}_{m=1}^{\tilde{M}}$.
    for $k' = 1, \dots, k$ do
        Compute Local POD Basis $\{\boldsymbol{\psi}_{l,k'}\}_{l=1}^{r_k}$ at cluster $k'$
        Compute Cluster projection error $E(k') = \sum_{\{m:\tilde{\boldsymbol{a}}_{n,m} \in \mathscr{C}_{k'}\}} |\tilde{\boldsymbol{u}}_{N,m} - \sum_{l=1}^{r} (\tilde{\boldsymbol{u}}_{N,m} \cdot \boldsymbol{\psi}_{l,k'})(\boldsymbol{\psi}_{l,k'})|$
    end for
    Compute sum of Cluster Projection Errors $MAE(k) = \sum_{k'=1}^{k} E(k')$
    if $MAE_k > MAE_{k-1}$ then
        $k_{opt} = k - 1$
    end if
end for
Train Local surrogate models $\mathscr{F}_1, \dots, \mathscr{F}_{k_{best}}$ for each cluster with $\{(\vec{\theta}^{k'}, \boldsymbol{a}_n^{k'})_m\}_{m=1}^{|C_{k'}|}$ pairs.
```

3.5.4. ASSIGN UNSEEN PARAMETER POINT TO A CLUSTER

During the online phase of the ROM, the model receives as input an unseen parameter point (or a series of parameter points, i.e. a maneuver), here for brevity referred to as a test point. Unlike the training points, for which a reduced representation of the snapshot set is available, test points are only specified in the parameter space, as the prediction of the snapshots is the actual final output of the model. As a result, deciding on which cluster the parameter instance lies, and thus its associated local surrogate model and local basis requires an additional predictive step. In synthesis, the clustering is performed in the global reduced coordinates space while the online assignment of the cluster has to be performed in the parameter space. In order to close this process, the information from the training snapshot set can be leveraged in different ways. In particular, the clustering $\{\mathscr{C}_k\}_{k=1}^{k_{opt}}$ of the training snapshots $\{\boldsymbol{u}_{N,m}\}_{m=1}^{M}$ induces the clustering of the training parameter points $\Theta^{Tr} = \{\vec{\theta}_m\}_{m=1}^{M}$ into smaller sets:

$$\Theta_k^{Tr} = \{\vec{\theta}_m \in \Theta^{Tr} \quad : \quad \boldsymbol{u}_{N,m} \in \mathscr{C}_k\} \quad k = 1, ..., k_{opt}. \tag{3.21}$$

The simplest method would compute the parameters' mean in each cluster:

$$\bar{\vec{\theta}}_k = \frac{\vec{\theta}}{|\mathscr{C}_k|}, \quad \vec{\theta} \in \Theta_k^{Tr} \tag{3.22}$$

and assign the test parameter instance to the cluster that minimizes its distance from the mean. This method, despite its simplicity, can lead to erroneous results: the cluster's parameter mean does not generally correspond to the cluster's centroid (which is instead defined as an average in the reduced coordinates space) and therefore does not provide a good reference for calculating distances. Additionally, the map from the reduced coordinates space to the parameter space does not conserve distances: clusters have non-uniform sizes in the parameter space and this should be taken into account. The midrange criterion, proposed by Hess [27] for a uni-dimensional parametric case, defines instead a cluster radius and a cluster midrange in the parameter space in order to account for the different sizes of the cluster. Extension of this criterion to a multi-dimensional case is nevertheless complex, as different parameters have different influences on the solution.

Algorithm 3 Online Stage

```
Input:Test maneuver points $\boldsymbol{\vec{\theta}_m^{(Test)}}$, Global regression model $\mathscr{F}$, Local surrogate models $\mathscr{F}_1, ..., \mathscr{F}_{k_{best}}$ , Local basis $\{\boldsymbol{\psi}_{l,1}\}_{l=1}^{r_1}, ..., \{\boldsymbol{\psi}_{l,k_{opt}}\}_{l=1}^{r_{k_{opt}}}$
Output: Snapshots predictions $\hat{\boldsymbol{u}}_{N,m}^{(Test)}$

Predict global coefficients test maneuver $\{\boldsymbol{a_{n,m}^{(Test)}}\}_{n=1}^{r} = \mathscr{F}(\boldsymbol{\vec{\theta}_m^{(Test)}})$
Assign Testing points to cluster based on $\mathbf{a_{n,m}^{(Test)}} \cdot \sigma_n$.
Predict local coefficients test maneuver in each cluster $\{\boldsymbol{\hat{a}_{n,m}^{(Test,k)}}\}_{n=1}^{r_k} = \mathscr{F}_k(\boldsymbol{\vec{\theta}_m^{(Test)}})$
Predict full snapshots $\hat{\boldsymbol{u}}_{N,m}^{(Test)} = \sum_{l=1}^{r} \boldsymbol{\hat{a}_{l,m}^{(Test,k)}} \boldsymbol{\psi}_{l,k} \quad \{m : \boldsymbol{a_{n,m}^{(Test)}} \in \mathscr{C}_k\}$
```

The proposed method, instead, aims to approximate the map between the parameter space and the global reduced coordinates, counting on the fact that the pre-trained k-means algorithm can be used to classify the global coordinates to each cluster. Luckily, not much additional work has to be done in order to find the approximate map, as in the global POD-LSTM, the Neural Network is used exactly for this purpose. This explains why in the first phase of the offline stage, a global regression model is trained. Therefore the composition of the k-means and the pre-trained global surrogate model directly solves the classification task. One can observe that a classification model could be trained on the same data, directly predicting a number in the range $1, ..., k_{opt}$, thus avoiding the composition of the k-means and the global regression model. The results would in fact be really similar, as a regression ANN can be turned into an equivalent classifier just by adding a dense layer with softmax activations. However, the k-means practically plays the part of the dense layer and it is computationally more efficient. Additionally, changing the number of clusters would not require modifications in the regression model, as instead would be the case for a classification model with a variable number of classes.

3.5.5. GAUSSIAN MIXTURE MODELS FOR OVERLAPPING CLUSTERS

Gaussian Mixture Models (GMMs) are a class of probabilistic models based on the assumption that a sample of observed data is generated from a mixture of Gaussian distributions. GMMs can be used as clustering techniques, by identifying the distribution that has the highest probability of having generated a given data-point. This technique can also be seen as a generalization of the K-means algorithm allowing the identification of clusters not only with spherical shapes but with elliptic shape, thanks to the incorporation of information about the data covariance. Compared to K-means clustering, GMMs do not perform "hard cluster assignments" (i.e. the only output of the model is whether a data-point belongs or not to a cluster) but also output an estimate of the probabilities of a data-point belonging to each cluster. This aspect is a key factor for the definition of overlapping clusters: overlap regions between clusters can be detected when the maximum cluster probability is lower than a predefined threshold value. Overlapping clusters, allow for the construction of local snapshots matrices with elements in common, and therefore Local Bases that contain common information. This feature enables a good coverage of the local regressor spaces even along the clusters borders, and eliminates the peaks observed in the projection error in proximity of cluster transition regions.

A Gaussian mixture distribution can be expressed as [8]:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{3.23}$$

where the quantity π_k is denoted as mixing coefficient, and K is the total number of components of the GMM model. The K-dimensional random variable $\boldsymbol{z}$ is introduced, such that only one z_k is equal to 1 and the rest to 0: this variable defines the actual state k (or cluster) among the possible K states. The marginal distribution over $\boldsymbol{z}$ is expressed through the mixing coefficients:

$$p(z_k = 1) = \pi_k \tag{3.24}$$

where pi_k are probabilities, and sum to 1. In the case several independent observation (data-points) $\boldsymbol{X} = \{\boldsymbol{x_1}, ..., \boldsymbol{x_M}\}$ are drawn from the distributions, it is possible to express the log-likelihood of the data given the parameters as [8]:

$$\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{m=1}^{M} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x_m}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{3.25}$$

Maximizing this function with respect to the parameters, means maximizing the probability that the observed data points have been generated from the Gaussian Mixture

Models. Differentiating with respect to the parameters $\boldsymbol{u}_k$ and setting the derivatives to zero, yields the following expressions:

$$\begin{aligned}
\boldsymbol{\mu}_k &= \frac{1}{M_k}\sum_{m=1}^{M}\gamma(z_{mk})\boldsymbol{x}_m \\
\boldsymbol{\Sigma}_k &= \frac{1}{M_k}\sum_{m=1}^{M}\gamma(z_{mk})(\boldsymbol{x}_m-\boldsymbol{\mu}_k)(\boldsymbol{x}_m-\boldsymbol{\mu}_k)^T \\
\pi_k &= \frac{M_k}{M}
\end{aligned} \tag{3.26}$$

where $\gamma(z_{mk}) \equiv p(z_{mk}=1|\boldsymbol{x}_m)$ is the responsibility which indicate the probability of a data-point belonging to each cluster, and can be calculated as:

$$\begin{aligned}
\gamma(z_{mk}) &= \frac{\pi_k\mathcal{N}(\boldsymbol{x}_{\Updownarrow}|\boldsymbol{\mu}_{\|},\boldsymbol{\Sigma}_{\|})}{\sum_{i=1}^{K}\pi_i\mathcal{N}(\boldsymbol{x}_{\boldsymbol{m}}|\boldsymbol{\mu}_{\rangle},\boldsymbol{\Sigma}_{\rangle})} \\
M_k &= \sum_{m=1}^{M}\gamma(z_{mk})
\end{aligned} \tag{3.27}$$

Practically, the log-likelihood is maximized by initializing the parameters, and iteratively evaluating the responsiblities and the parameters, until convergence is achieved [8].

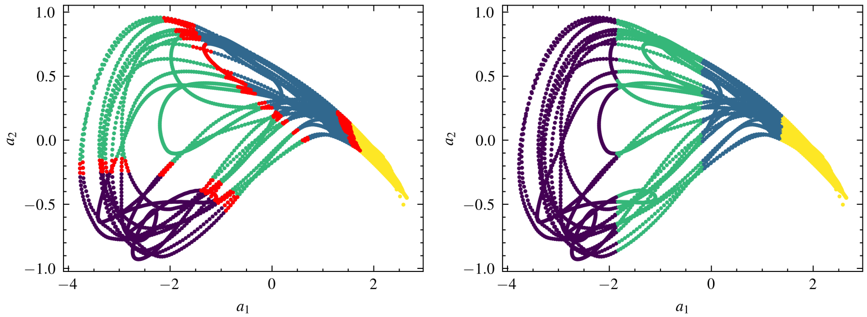
The knowledge of the "responsibilities", i.e. the probability of each data-point belonging to each cluster, can be leveraged to assign data-points to multiple clusters, during the generation of the local snapshot matrices. A threshold probability, p_ϵ can be used:

$$\boldsymbol{x}_m \in \mathscr{C}_k \quad \text{if } \gamma(z_{mk}) > p_\epsilon. \tag{3.28}$$

This criterion can also be used when generating the reduced representations of the training snapshots that are used to train the local regression models. However, in this work, GMM overlapping-clusters are only briefly introduced as a solution to the problem of the high projection error near the cluster transition regions, caused by the poor coverage of the reduced basis in those regions.

In Fig. 3.18, a comparison of the cluster definition in POD reduced coordinate space, between the K-means hard clustering and the GMM clustering with overlap is displayed. The GMM with overlap demonstrates to be a robust dimensionality reduction model, overcoming the main limitation of the Cluster-POD based on disjoint clusters.

Although this clustering strategy presents several advantages compared to the standard K-means clustering, in this thesis it has only been used to demonstrate how the projection error peaks near the cluster transition regions can be reduced by the definition of overlap regions. Therefore, the online results and final predictions of the ROMs are only based on the K-means clustering technique, while a short section (4.3) will be dedicated to the offline results of the GMM-based model.



(a) Gaussian Mixture Model, $p_\epsilon = 0.3$

(b) Disjoint K-means clustering.

Figure 3.18: Comparison of clusters in the POD coordinates subspace, using GMM with overlap and K-means algorithm. Points in red indicate the overlap regions. $K = 4$.

3.6. Latent Dynamics Modelling

This section describes the Machine Learning procedure followed to model the dynamics of the reduced coordinates derived from the dimensionality reduction of the full system. Most of the methods presented are general and applicable to both global and local model decomposition methods. Nevertheless, the specific inputs and parameters for each of the methods are also specified in the following paragraphs. Moreover, some of the most relevant training results of the Machine Learning methods are reported and discussed.

3.6.1. Machine Learning for Regression

The objective of the non-intrusive surrogate model is to learn a continuous map from the parameter space (also referred to as the input features space) to the reduced coordinate space:

$$\mathscr{F}(\vec{\theta}; w) : \vec{\theta}_m \rightarrow \{\boldsymbol{a}_{n,m}\}_{n=1}^{r} \tag{3.29}$$

where the surrogate model $\mathscr{F}$ is a non-linear function parametrized by its weights w. The output space has a user-defined dimension r which corresponds to the truncation index of the sum associated with the modal reconstruction. The input space dimension depends on which features are selected to feed the model: the unsteadiness of the CFD simulations implies that the compressed representation of the snapshots contains time-history effects and therefore, not only the instantaneous input vector $\vec{\theta}_t$ should be included as input features, but also the inputs at previous time-steps. The continuity of the input signal in the $\alpha - q$ sample space implies that the derivatives of these parameters might also have an important influence on the instantaneous pressure fields and should therefore be incorporated in the regressor input features.

The relative importance of each input can be quantified by looking at the correlations between input features and expected outputs. The Pearson product-moment correlation coefficient is used to measure the correlation, being a common choice in the Feature Selection step of Machine Learning pipelines:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{3.30}$$

where x_i, y_i are any of the two input or output features and $\bar{x}, \bar{y}$ the means of these features.

In Fig. 3.19, the heatmap of the Pearson correlation coefficient for the instantaneous inputs and outputs is displayed. The strongest correlation can be found between the angle of attack α and the first reduced coefficient a_1 (associated with the most energetic mode) suggesting that this parameter is the most meaningful in determining the unsteady pressure fields. High correlation to a_1 can be found with the second time derivative of the angle of incidence $\ddot{\alpha}$: this is however due to the high correlation between α and $\ddot{\alpha}$ which results from the sinusoidal shape of the input signal for the training maneuver, and consequently removing the feature $\ddot{\alpha}$ from the inputs should not influence the accuracy of the regression model. The pitch rate q appears to strongly influence the fourth reduced coefficient. Although removing some of the low correlated input features could be beneficial in terms of the model efficiency, in this context all of the considered input features are kept, in order to allow for a simpler comparison of the resulting ROM with the POD-LSTM model of Bourier [9], in which the Neural Network was trained with all the listed inputs. However, when a larger number of degrees of freedom or parameters are considered in the generation of the samples, a Feature Selection study can result in a remarkable gain in computational efficiency and accuracy, by removing the redundant input features from the model.

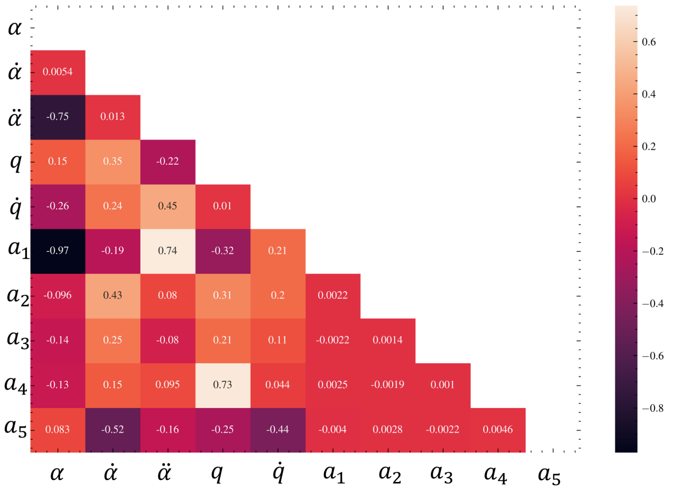


Figure 3.19: Pearson correlation coefficient between the instantaneous control inputs and the first five global POD reduced coefficient.

Data Preparation

The set(s) of reduced coefficients resulting from the Global or Local decomposition methods and the control inputs are fed into the Regression models as training samples. In order to ensure that all components of the multi-input and multi-output vectors have similar ranges of variations, the input and output data should be normalized. The normalization procedure is beneficial for the training convergence because it promotes a loss function that is more spherical with respect to the output features (without normalization, an output feature with a larger scale of variation would have a larger weight in determining the loss minimum). Therefore, all inputs and output features are normalized to lie in the range of values [0,1] using a Min-Max scaler:

$$a_{scaled} = \frac{a - a_{min}}{a_{max} - a_{min}}. \tag{3.31}$$

3.6.2. THE MULTI-LAYER-PERCEPTRON

The Multi-Layer-Perceptron architecture presented in Chapter 2 is a common choice [25] to solve supervised regression tasks due to the computational efficiency and flexibility of the model to learn the non-linear mapping between the inputs and the outputs. Additionally, this model is often used as a baseline for comparison with more complex architecture, such as the LSTM network [59].

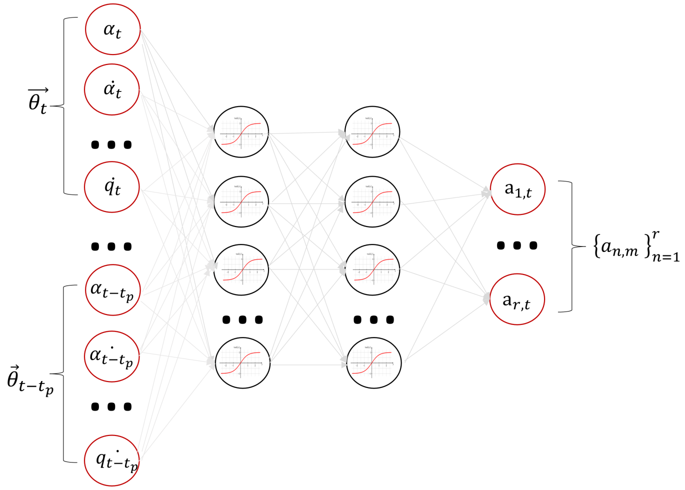


Figure 3.20: Multi-Layer-Perceptron Architecture.

In Fig. 3.20, a schematic representation of the architecture used in this study is given, highlighting the input and output features of the model. In particular, it must be noted how the parameter input vector $\vec{\theta}$ is fed into the input layer at the current time step and at previous time steps until t_p. The number of outputs can be chosen depending on the level of accuracy needed for the reduced representation of the pressure fields and it is generically indicated by r. Note that the MLP architecture is also used in this study for the direct prediction of the loads coefficient as explained in Sec. 3.6.5: in that case, the size of the output layer is 3, corresponding to the Axial Force Coefficient, Normal Force Coefficient, and Pitching Moment Coefficient. The final MLP model in this study is dependent on the choice of the hyperparameters and the loss function. In the following paragraph, these choices are specified and justified.

Hyperparameter	Tested Values	Baseline	Optimal Value
Number of Hidden Layers	[1,2,3,4]	2	3
Number of Hidden Units	[64,128,256,512,1024]	128	512
Batch Size	[8,16,32,64,128]	32	64

Table 3.11: Hyperparamter optimization settings and results.

Loss Function

The loss function used in the training of the MLP model for the prediction of the reduced coordinates is the weighted sum of the Mean Square distance between the projected snapshots and the predicted snapshots and the Mean Square distance between the predicted and the true coordinates. The outputs of the network $\{\hat{\boldsymbol{a}}_{n,m}\}_{n=1}^{r}$ are therefore multiplied by the POD modes and then compared to POD expansion of the true targets $\{\boldsymbol{a}_{n,m}\}_{n=1}^{r}$:

$$\mathscr{L} = || \sum_{n=1}^{r} \boldsymbol{a}_{n,m}\boldsymbol{\phi}_n(\boldsymbol{x}) - \sum_{n=1}^{r} \hat{\boldsymbol{a}}_{n,m}\boldsymbol{\phi}_n(\boldsymbol{x})||^2 + \lambda \sum_{n=1}^{r} ||\hat{\boldsymbol{a}}_{n,m} - \boldsymbol{a}_{n,m}||^2. \tag{3.32}$$

The first term of the sum accounts for the relative importance of the POD coefficients in the full-state representation of the snapshots: the lower-ranked terms play a stronger role in determining the dominant feature of the snapshots and the Network should primarily learn to accurately predict these terms. Nevertheless, a second term weighed by the scalar λ is added, in order to guarantee a certain level of accuracy even on the higher-ranked modes which can have a remarkable impact on the solutions of the testing snapshots. For the direct prediction of the integral loads, the Loss function is simply defined as:

$$\mathscr{L} = \sum_{i=1}^{3} ||\hat{\boldsymbol{C}}_{\boldsymbol{F}\,i,m} - \boldsymbol{C}_{\boldsymbol{F}\,i,m}||^2. \tag{3.33}$$

where:

$$\boldsymbol{C}_{\boldsymbol{F}} = [C_A, C_N, C_{M_y}] \tag{3.34}$$

Hyperparameter tuning and Training Procedure

The hyperparameter choices influence the performance of a Neural Network in terms of minimum loss and training time. An extensive sensitivity study is performed in the work of Bourier [9], in order to understand the influence of each hyperparameter on the performances of the LSTM network in the prediction of the reduced coefficients. In this study, however, the tuning is performed only on the most relevant hyperparameters that define the architecture (Number of Hidden Layers, Number of Hidden Units) and the training phase (Batch size). An optimization script is run, changing each hyperparameter one at a time with respect to the baseline settings (as shown in Tab. 3.11) and performing training for a maximum number of 1000 epochs, which is stopped when the validation loss does not decrease for 200 epochs (patience parameter).

The final model settings are summarized in Tab. 3.12 and are common to all the MLP Networks used in the remainder of the thesis.

Settings	Value
Number of Hidden Layers	3
Number of Hidden Units	[512, 512, 512]
Batch Size	64
Learning Rate	1e-3
Activation	$[tanh, tanh, tanh]$
Optimizer	ADAM
Patience	200
Epochs	2000
t_p	5

Table 3.12: Hyperparamter and training settings of the final MLP model.

The MLP architecture and training modules are implemented within the `Keras` [13] API of the `TensorFlow` library.

3.6.3. THE GAUSSIAN PROCESS REGRESSION

The theoretical background of the Gaussian Process Regression has been introduced in Sec. 2.2.3: the goal of this paragraph is to describe the actual implementation of the method used in this work, with particular respect to the hyperparameter optimization framework.

The library `GPyTorch` [20] is used for the implementation of Gaussian Process Regression: this package is built in `PyTorch` [52] and therefore allows integration of GPR and Neural Network Models. The advantage of using `GPyTorch` routines compared to other popular libraries such as `GPFlow` [65], is the possibility to use GPU acceleration. In fact, the computation of the marginal log-likelihood introduced in Eq. 2.27, is only based on matrix-matrix multiplication, leveraging the BlackBox Matrix-Matrix Multiplication (BBMM) inference, introduced by Gardner et al. [20].

The hyperparameters are optimized using the ADAM algorithm [35] on the loss defined by the negative log marginal likelihood. The training time is remarkably decreased when GPU acceleration is employed, even for a large number of data points. In the specific case of the Global POD surrogate modeling, for $t_p = 5$, the dimension of the input space is 25 and the output space dimension depends on the number of modes used. For the training, 5666 data points are used: most standard implementations would require unfeasible training time.

3.6.4. THE LONG SHORT TERM MEMORY NETWORK

The Long Short Term Memory Network, introduced in Section 2.2.2, is a common choice for unsteady ROM construction, due to its capabilities to process time-dependant data and to naturally model time-history effects. In this section, the main features of the specific LSTM model employed in this work are presented.

The architecture considered for the prediction task (for both the reduced coordinates and the integral coefficient predictions) is the many-to-one-type RNN: the predictions are only performed at the present time-step, based on multiple input vectors given at previous time-steps up to t_p. An illustrative example of this type of architecture is displayed in Fig. 3.21, where the inputs $\vec{\theta}_t$ are highlighted in green and are sequentially processed by the LSTM gates (in blue) across the Network layers (moving upwards). Compared to the Multi-Layer-Perceptron, each sample input vector $[\vec{\theta}_t, ..., \vec{\theta}_{t-t_p}]$ is not flattened along the time-dimension but it is processed as a time-series of dimension $[t_p, p]$ where $p = 5$ is the number of parameters used to describe the input at each time-instant. As a consequence, unlike the MLP, the weights that process the same inputs (or intermediate outputs) are shared across time steps. The architecture in Fig. 3.21 only serves as an example to describe how the information across time steps propagates in the LSTM. The true architecture of the LSTM resembles the MLP in Fig. 3.20, where the standard units in the intermediate layers are replaced by LSTM units, followed by a densely connected layer. This architecture does not differ from the one used in the thesis of Bourier [9].

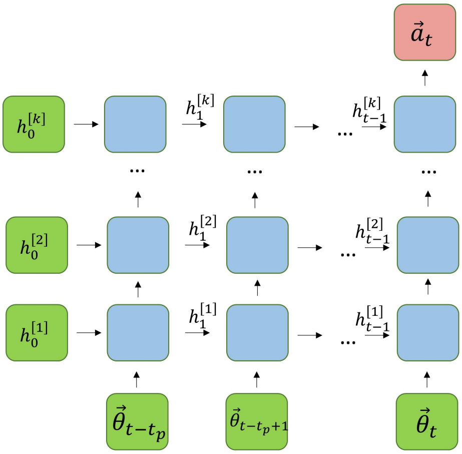


Figure 3.21: LSTM many-to-one Architecture.

The selected loss function is analogous to the MLP loss function of Eq. 3.32 with $\lambda = 0.1$ for the prediction of the reduced coordinates. For the prediction of the integral loads coefficients Eq. 3.33 is used. The hyper-parameter tuning is performed in a similar fashion to the MLP architecture, also exploiting the results of the extensive sensitivity study on the LSTM network parameters present in the work of Bourier [9]. The best accuracy, in terms of minimum validation loss, is achieved for a Network with two LSTM layers, followed by a densely connected layer. Each LSTM layer contains 1024 units, while the densely connected layer contains 90 units. The other hyperparameters are summarized in Table 3.13: after the LSTM layers, a dropout layer with a dropout rate of 0.2 is included to avoid overfitting. This architecture, however, is remarkably larger in terms of number of tunable parameters, with a sensible increase in training time compared to the smaller MLP. In order to allow for a fair comparison of the LSTM and MLP architectures, a smaller LSTM with the same number of hidden layers and hidden units

Settings	**LSTM**	**LSTM (small)**
Number of LSTM Layers	2	2
LSTM Units	[1024, 1024]	[512, 512]
Number of FC Layers	1	1
FC Units	128	128
Batch Size	64	64
Learning Rate	1e-3	1e-3
Activation	$[tanh, tanh, tanh]$	$[tanh, tanh, tanh]$
Dropout Rate	0.2	0.2
Optimizer	ADAM	ADAM
Patience	200	200
Epochs	2000	2000
t_p	5	5

Table 3.13: Hyperparameters and training settings of the final LSTM model and the downscaled version.

as the MLP is considered, where one of the fully connected layers of the MLP is replaced by an LSTM layer. The architectural and training features of this LSTM (referred to as "small" LSTM) are also included and presented in Table 3.13.

3.6.5. DIRECT LOADS COEFFICIENTS PREDICTION VIA SURROGATE MODELING

The predictions of the total integrated force and moment coefficients can be performed by direct integration of the predicted pressure fields: as the pressure fields can be expressed as a linear combination of the first r modes and the integration is a linear operation, the modes can be integrated in advance and the resulting integrated forces can also be expressed as a linear combination of the integrated modes using the same predicted reduced coordinates. Nevertheless, the predicted pressure fields are affected by two main sources of error: the projection error and the network error, which are propagated into the integrated coefficients. It is therefore convenient, in order to bypass the projection error, to directly infer these scalar quantities through the same regression models used to predict the reduced coordinates. The true coefficients could be added to the output of the models by adequately modifying the loss function used to predict the reduced coordinates. However, in the present thesis, distinct models (with the analogous architectures described in the previous paragraphs) are trained using the Loss Function defined in Eq. 3.33 to output the three loads coefficients. It must be noted that this direct procedure offers additional flexibility compared to the integration of the pressure fields, as it can be used as a standalone model for the prediction of integral quantities when the pressure fields are not a desired output of the model. Additionally, it can be generalized to cases where not only the pressure component of the forces is taken into account for the computation of integral forces.

3.7. ERROR ANALYSIS

The dimensionality reduction based on a linear subspace approximation allows for a clear separation between the offline and the online stage of the ROM. The immediate consequence is that access to the latent representation of each snapshot is possible through simple projection on the reduced space, and reconstruction of the truncated version of the snapshots is performed by simple linear expansion. As a result, it is possible to distinguish between two forms of errors:

- the projection error $\epsilon_P^{c_p}$ on the pressure coefficient distribution, resulting from the use of a reduced basis and independent of the regression model performance, which can be calculated as:

$$\epsilon_P^{c_p}(\boldsymbol{x}) = |\boldsymbol{u}_{N,m} - \sum_{l=1}^{r} \boldsymbol{a}_{l,m}\boldsymbol{\phi}_l| \tag{3.35}$$

- the network error $\epsilon_{NN}^{c_p}$ on the pressure coefficient distribution, which has a source in the regressor network and it is essentially a measure of the distance between the true reduced coordinates and the predicted reduced coordinates:

$$\epsilon_{NN}^{c_p}(\boldsymbol{x}) = |\sum_{l=1}^{r} \boldsymbol{a}_{l,m}\boldsymbol{\phi}_l - \sum_{l=1}^{r} \hat{\boldsymbol{a}}_{l,m}\boldsymbol{\phi}_l| = |\sum_{l=1}^{r}(\boldsymbol{a}_{l,m} - \hat{\boldsymbol{a}}_{l,m})\boldsymbol{\phi}_l|. \tag{3.36}$$

The total error $\epsilon_T^{c_p}$ of the ROM on the pressure coefficient distribution can be evaluated as the difference between the ground truth and the predicted pressure fields and it therefore inherently contains both projection and projection error:

$$\epsilon_T^{c_p}(\boldsymbol{x}) = |\boldsymbol{u}_{N,m} - \sum_{l=1}^{r} \hat{\boldsymbol{a}}_{l,m}\boldsymbol{\phi}_l|. \tag{3.37}$$

It might be tempting to express the total error as the sum of the projection and the network error but this is in general not true due to the presence of the absolute values. The described metrics are useful to evaluate the error as a function of space and the specific snapshot within a maneuver. In order to evaluate the performance of the models over multiple snapshots that form a maneuver, a scalar measure of the errors based on the Mean Square error metric is also introduced :

$$\begin{aligned}
\mathscr{E}_P^{c_p} &= \frac{1}{M}\sum_{\boldsymbol{x}} \epsilon_P^{c_p}(\boldsymbol{x})^2 \\
\mathscr{E}_{NN}^{c_p} &= \frac{1}{M}\sum_{\boldsymbol{x}} \epsilon_{NN}^{c_p}(\boldsymbol{x})^2 \\
\mathscr{E}_T^{c_p} &= \frac{1}{M}\sum_{\boldsymbol{x}} \epsilon_T^{c_p}(\boldsymbol{x})^2.
\end{aligned} \tag{3.38}$$

Similarly, the instantaneous errors on the integral loads' coefficients can be introduced:

$$\begin{aligned}
\epsilon_P^{c_f} &= |\mathscr{I}(\boldsymbol{u}_{N,m}) - \mathscr{I}(\sum_{l=1}^{r} \boldsymbol{a}_{l,m}\boldsymbol{\phi}_l)| \\
\epsilon_{NN}^{c_f} &= |\mathscr{I}(\sum_{l=1}^{r} \boldsymbol{a}_{l,m}\boldsymbol{\phi}_l) - \mathscr{I}(\sum_{l=1}^{r} \hat{\boldsymbol{a}}_{l,m}\boldsymbol{\phi}_l)| \\
\epsilon_T^{c_f} &= |\mathscr{I}(\boldsymbol{u}_{N,m}) - \mathscr{I}(\sum_{l=1}^{r} \hat{\boldsymbol{a}}_{l,m}\boldsymbol{\phi}_l)|
\end{aligned} \tag{3.39}$$

where $\mathscr{I}$ indicates the integral operator that maps the pressure field coefficients to the specific integral loads' coefficients. The global MSE over a maneuver is then calculated as:

$$\begin{aligned}
\mathscr{E}_P^{c_f} &= \frac{1}{M}\sum_{\boldsymbol{x}} \epsilon_P^{c_f}(\boldsymbol{x})^2 \\
\mathscr{E}_{NN}^{c_f} &= \frac{1}{M}\sum_{\boldsymbol{x}} \epsilon_{NN}^{c_f}(\boldsymbol{x})^2 \\
\mathscr{E}_T^{c_f} &= \frac{1}{M}\sum_{\boldsymbol{x}} \epsilon_T^{c_f}(\boldsymbol{x})^2.
\end{aligned} \tag{3.40}$$

When the integral loads are computed directly through the surrogate models, it is obvious that:

$$\epsilon_P^{c_f} = 0 \longrightarrow \epsilon_{NN}^{c_f} = \epsilon_T^{c_f} \tag{3.41}$$

4

Results and Discussion

This chapter includes the most relevant results of the application of the three ROMs described in the Methodology chapter: Global POD-ANN, Domain Decomposition POD-ANN, and Cluster POD-ANN. The offline and online stage performances are first described separately for each method and then included in a comparative study of the ROMs, which is at the heart of the thesis work. The effect of the snapshot scaling on the decomposition is also given, in terms of projection error on the integral loads' coefficients. The direct integral load coefficient predictions, performed through the three surrogate models employed (Multi-Layer-Perceptron, Long Short Term Memory Architecture, Gaussian Process Regression), is described in the following sections and serves as a comparison of the regression methods.

4.1. Scaled POD

The idea of scaling the pressure field snapshots was introduced in Chapter 3, with the aim of determining a reduced basis with increased efficiency in capturing spatially extended dominant structures and improving the performance of the ROM in key areas of the spatial domain, such as the wingtip region, where large projection errors have been observed in the thesis of Papp [51] and Bourier [9]. As previously observed, the snapshot scaling is equivalent to an alternative definition of the norm that is minimized by the POD procedure. Therefore, by introducing a scaling based on the cell areas, the physical interpretation of the POD subspace is not merely based on the maximization of the "pressure energy" (where the term energy comes from the analogy with POD in the context of turbulent flow analysis, where the maximization of the turbulent kinetic energy is the goal of the POD) contained in the retained modes, but can be seen as the maximization of "force energy" and equivalently the "moment energy" when the distance scaling is considered as well. The load coefficients are calculated from the force and moment distributions on the wing and it is expected that additional accuracy in the prediction of these quantities derives from the use of a scaled POD basis when the same number of modes are used. Looking at the problem from another perspective, the improvement in the projection error of the pressure fields in certain regions (for instance near the wing-tip for the Moment-POD), leads to the consequent improvement of integral quantities that have a stronger dependence on those regions.

In Fig. 4.1 a typical projection error of the pressure distribution along the wing $\epsilon_P^{cp}(\boldsymbol{x})$ resulting from the use of the different scaling procedures is displayed. The $\mathscr{L}^2$-norm of the error would provide a minimum value for the Pressure-POD, as the corresponding basis is tailored for this goal. Nevertheless, larger areas of the domain are characterized by a more significant projection error for the pressure-POD compared to the Force and Moment POD. The former two bases do not present remarkable differences, as it was already clear from the shape of the dominant modes in Fig. 3.11: this is because the multiplication by the cell areas dominates over the multiplication factor that accounts for the distance from the pitching moment pole. One could further modify the scaling by introducing a weighting factor that accounts for the distance from the wingtip, in order to increase the accuracy of the ROM in that region. The choice of the weighting function is arbitrary and depends on the goal functional: Bourier [9] proposes a linear scaling function dependent on the distance from the wing-tip. However, if the goal functional is set as the pitching moment coefficient, the best way of scaling is represented by the Moment-POD, since the optimization problem is defined as the minimization of the projection error on the moment distribution.

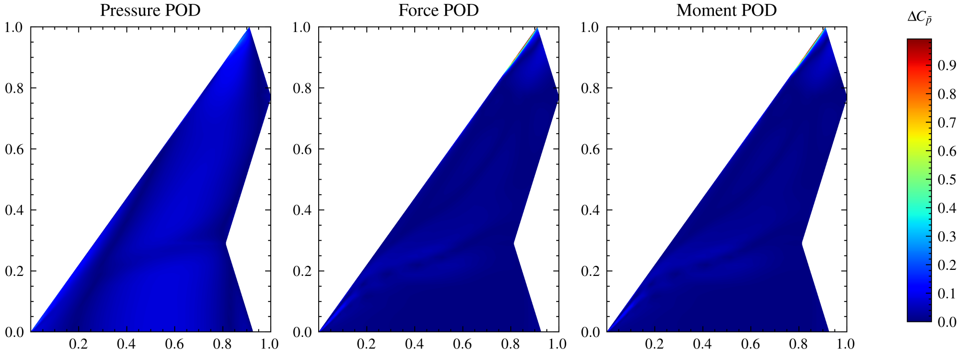


Figure 4.1: Absolute projection error $\epsilon_P^{cp}(\boldsymbol{x})$ using three different snapshot scaling methods. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$ at $t = 1s$. $\alpha = 1.71[deg]$,$\dot{\alpha} = 23.4[deg/s]$,$q = 23.4[deg/s]$

In Fig. 4.2, the projections of axial force coefficient, normal force coefficient, and pitching moment coefficient on the three scaled bases are respectively displayed, and compared to the ground truth data, for the same testing maneuver of Fig. 4.1. In particular, 5 modes are used to perform the projection: this would be the result of integrating the projected pressure fields on a POD subspace spanned by the dominant 5 modes. These plots clearly display the advantages of using a load-scaled basis instead of applying the POD decomposition directly on the pressure fields: the pressure-POD presents large inaccuracies in representing the normal force coefficient C_N, although it manages to capture its general trend. Relative to the pitching moment coefficient C_{M_y}, instead, the inaccuracy largely exceeds the acceptable level of error of a Reduced Order Model, and the number of modes should be greatly augmented to achieve a comparable level of accuracy to the scaled modes.

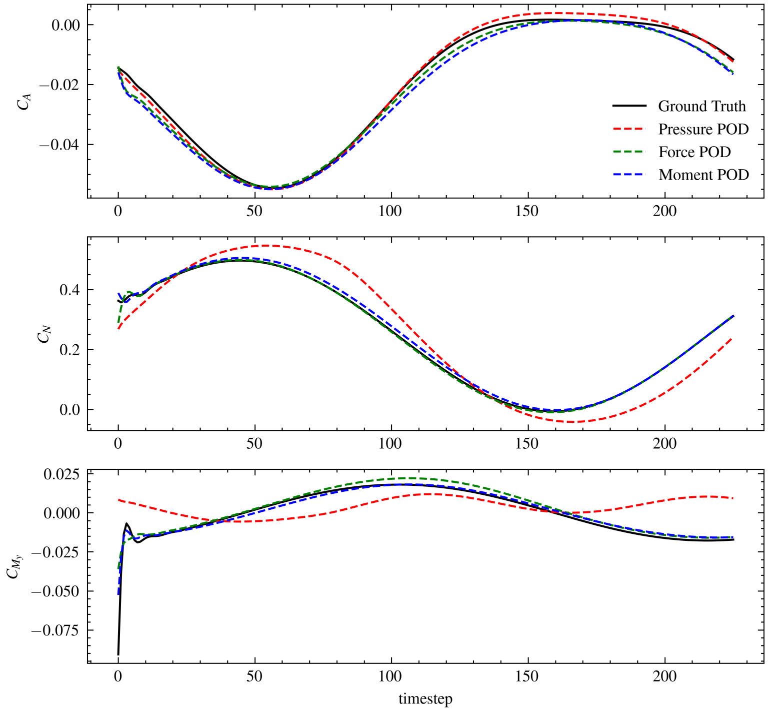


Figure 4.2: Projection of the integral loads' coefficients using 5 modes and three different scaled bases.Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$. Top: Axial Force Coefficient C_A. Middle:Normal Force Coefficient C_N. Bottom:Pitching Moment Coefficient C_{M_y}

In Tab. 4.1 the Mean Absolute Error, for the same maneuver of Fig. 4.2, is presented for a varying number of modes. The error on C_A is contained within acceptable ranges for all methods using a minimal number of modes, although the Pressure-POD shows the best performance in the prediction of this quantity. Significant differences can be detected in the projection error on C_N, with the Force and Moment POD being $O(10^1)$ more accurate than the Pressure-POD for all the considered truncation indices: using 40 Pressure-POD modes provides a similar error of using only 5 scaled POD modes. Similar differences can be observed in the predictions of the pitching moment coefficient, as the Moment-POD displays the best performance with superior accuracy to the pressure-POD even when using a factor of eight fewer modes.

Interestingly, the training results also show differences between the three decompositions for the same Neural Network architecture and hyper-parameter settings. The distribution of the reduced coefficients resulting from the three bases are in fact strongly

Scaling	**N. Modes**	$\mathcal{E}_P^{C_A}$	$\mathcal{E}_P^{C_N}$	$\mathcal{E}_P^{C_{My}}$
Pressure	5	0.0011870	0.0526189	0.0116343
Force	5	0.0016809	0.0028256	0.0026799
Moment	5	0.0023635	0.0078735	0.0016667
Pressure	10	0.0004553	0.0342688	0.0070497
Force	10	0.0007188	0.0022346	0.0013448
Moment	10	0.0009675	0.0023856	0.0007925
Pressure	20	0.0002623	0.0197256	0.0026204
Force	20	0.0002710	0.0003266	0.0003126
Moment	20	0.0003870	0.0013067	0.0003850
Pressure	40	0.0001192	0.0020636	0.0022470
Force	40	0.0001273	0.0002304	0.0002203
Moment	40	0.0001684	0.0003084	0.0001510

Table 4.1: Projection error on integral loads coefficients using different scaling and for a different number of modes.

related to their corresponding POD subspace: the performances of the Neural Network are affected by the distribution of its outputs, tending to overfit on the training data for output signal characterized by a larger amount of noise. This overfitting behavior is visible in Fig. 4.3, where the training-validation loss convergence curve is shown for the three scaling procedures, using three analogous LSTM Networks with a patience of 200 epochs (the training stops when the validation loss does not improve for 200 epochs): the pressure POD validation loss starts to grow after 300 epochs, indicating that the NN is learning non-generalizable patterns in the training data. The minimum validation losses for the Moment and Force POD are around one order of magnitude smaller than the Pressure-POD validation loss, demonstrating the advantages of scaling the snapshots even for the Network prediction error.

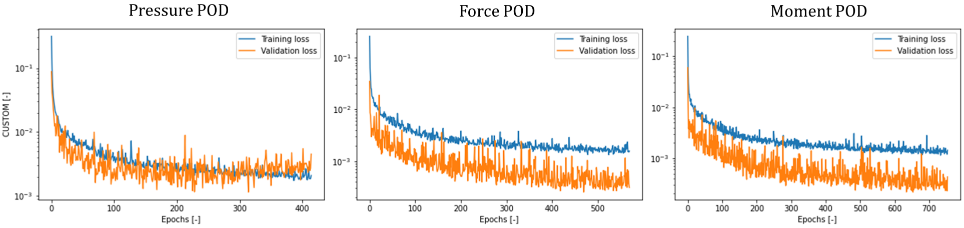


Figure 4.3: Training and Validation Loss curves for LSTM network trained on the same input. The outputs are the first 5 reduced coefficients of the three bases.

Considering the results and the analysis conducted in this Section, scaling the modes by the cell areas appears to consistently improve the performance of the resulting ROMs, both in terms of the integral loads' projection errors and in the Neural Network training convergence. For this reason, in the remainder of the Chapter, unless differently specified, the Moment-POD is used to generate the global and the local basis during the offline stage.

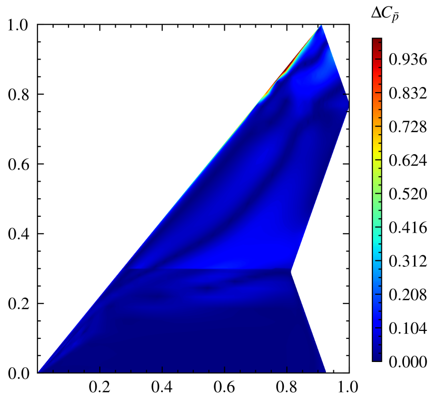
4.2. Domain Decomposition POD

This section includes the result of the Domain Decomposition-POD (DD-POD) based ROM, following the methodology described in Sec. 3.4. In the first paragraph, the offline stage results are presented: the spatial projection error arising from the dimensionality reduction is discussed, focusing on a challenging testing condition, highlighting the advantages of the spatially localized basis and the differences between the two decomposition strategies employed ($DD_{0.8}$ and $DD_{0.3}$). Global measures of the projection error are also presented for several testing maneuvers and for a varying number of modes. The analysis of the second source of error, namely the network error, ends the discussion of the offline stage of the ROM. The second part of the Section deals with the online stage results, in particular the full pressure distribution predictions and the total ROM error.

Offline Stage

During the offline stage of the ROM, two sets of local bases are computed, one for each decomposition type. Each set consists of two local bases: the truncation of these basis allows for the evaluation of the projection error $\epsilon_P(\boldsymbol{x})$ using Eq. 3.35. In Fig. 4.4, this quantity is plotted using a low number of modes, for an operating point corresponding to a high angle of incidence, for which the Global-POD basis shows a large projection error. The same snapshot is also considered in the work of Papp [51], as the worst performances of the CNN-based model in terms of MSE are observed. The largest deviations are concentrated at the leading edge portion in the proximity of the wingtip region of the wing, as the truncation of the basis leads to neglecting the higher order modes that account for the vorticity generated at the wing-tip due to the large pressure difference between the upper and lower surface. However, it is important to note that the $DD_{0.8}$ decomposition limits the magnitude and the extension of the higher-error spots in the wing-tip region, compared to the $DD_{0.3}$ case. This behavior is mainly due to the capabilities of a spatially local basis to better capture the dynamically most relevant structures: the formulation of the DD-POD can in fact be seen as a combination of two distinct L^2 minimization problems over two different domains. It results that, in order to be included in the lower-ranked modes, the local coherent structures in each sub-domain are not "competing" in terms of energetic content with the coherent structures in other sub-domains, thus producing modes with stronger local dynamical significance. On the other hand, by splitting the decompositions, the spatial correlations between sub-domain are also neglected: the dominant modes are in fact computed as the eigenmodes associated with the largest eigenvalues of the correlation matrix $\boldsymbol{C} = \boldsymbol{X}^T\boldsymbol{X}$.

The global performances of the local DD-POD basis over several testing maneuvers, expressed in terms of time-averaged Mean Square Error, are displayed in Fig. 4.5 as a function of the number of modes retained in the reduced representation. For all the considered maneuvers, the error decreases by around one order of magnitude going from a reduced representation with 5 modes (in each domain) to 40 modes. Not surprisingly, the pitch maneuvers centered at a higher angle of attack display a slower convergence



(a) $DD_{0.3}$

(b) $DD_{0.8}$

Figure 4.4: Absolute projection error $\epsilon_P(\boldsymbol{x})$ using different geometry decompositions. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$ at $t = 0.25s$. Instantaneous parameters: $\alpha = 19.93[deg]$,$\dot{\alpha} = 5.38[deg/s]$,$q = 5.38[deg/s]$

rate, due to the difficulty of the reduced basis to capture the vortical structures appearing at those incidences. The $DD_{0.8}$ decomposition performs consistently better than the $DD_{0.3}$, for all considered number of modes settings: this feature clearly demonstrates how the decomposition geometry can affect the resulting basis quality, and in this specific case, how the isolation of the wing-tip region leads to improve performance of the ROM.

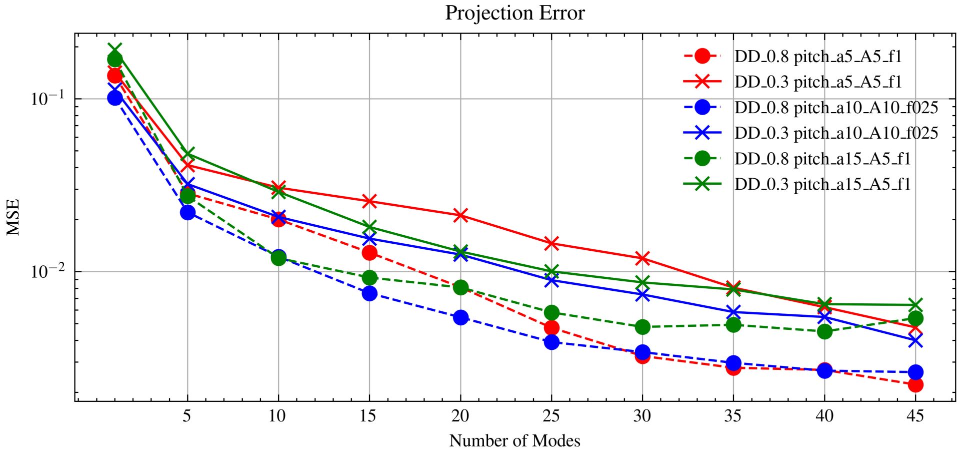
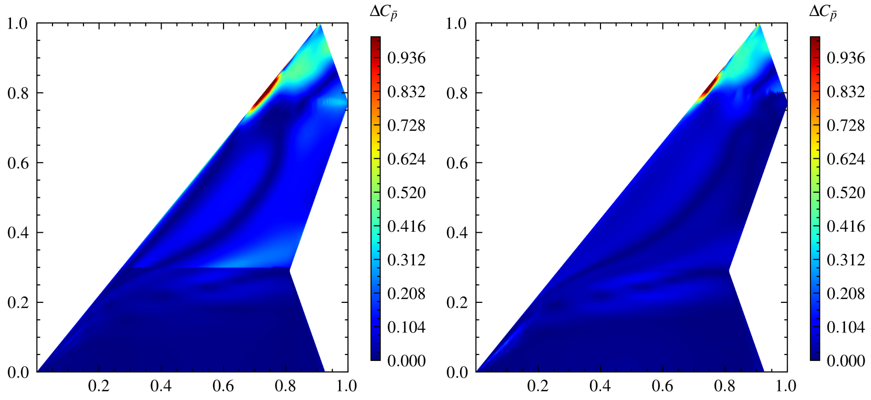


Figure 4.5: Time averaged Mean Squared Projection Error vs number of modes over several testing maneuvers, using two different decomposition strategies.

Online Stage

During the online stage of the method, the reduced coordinates are predicted by the local surrogate models on each sub-domain, and the final pressure field prediction is obtained through the geometric composition of the local POD sum expansion with the local bases. The ROM pressure fields for the same operating condition of Fig. 4.4 is now shown in Fig. 4.7, as the $DD_{0.3}$ (Fig. 4.7a) and the $DD_{0.8}$ (Fig. 4.7b) predictions are compared to the ground truth CFD fields (Fig. 4.7c). This operating condition is of particular interest, due to the complex flowfield pattern, highlighting the largest inaccuracies of the considered ROMs and therefore allowing for a better comparison of the models. In particular, the suction region at the leading edge sections towards the wingtip is underpredicted by the DD-ROMs leading to a peak in the total error over the mentioned area as visible in Fig. 4.6.



(a) $DD_{0.3}$

(b) $DD_{0.8}$

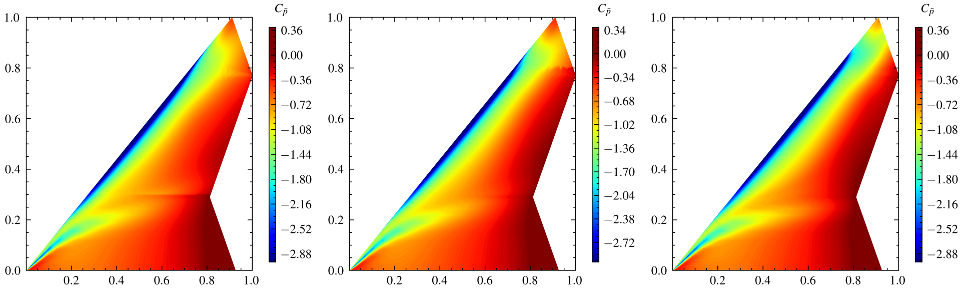
Figure 4.6: Absolute total error $\epsilon_P(\boldsymbol{x})$ using different geometry decompositions. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$ at $t = 0.25s$. $\alpha = 19.93[deg]$,$\dot{\alpha} = 5.38[deg/s]$,$q = 5.38[deg/s]$

4.3. Cluster POD

In this Section, the results concerning the Cluster POD (CPOD) methodology described in Sec. 3.5 are presented and discussed. The CPOD offline stage, in particular, is characterized by several steps: the discussion is here focused on the result of the optimization procedure that justifies the number of clusters employed in the ROM. Additionally, global measures of the projection error are given as a function of the basis truncation index. Some relevant examples of pressure fields predictions are then presented, highlighting the strength and deficiencies of the model.

Offline Stage

During the offline stage of the CPOD, the optimal number of clusters is chosen according to an a-posteriori indicator of the projection error evaluated on a validation maneuver,



(a) $DD_{0.3}$

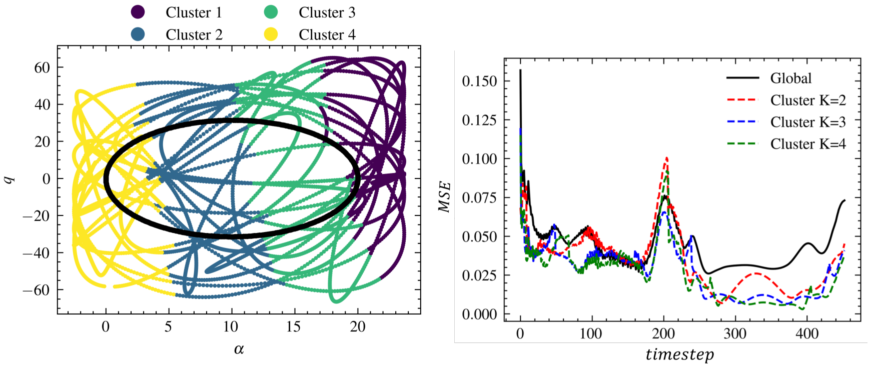
(b) $DD_{0.8}$

(c) Ground Truth

Figure 4.7: DD POD-LSTM pressure field predictions and comparison with CFD ground truth data. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$ at $t = 0.25s$. $\alpha = 19.93[deg]$,$\dot{\alpha} = 5.38[deg/s]$,$q = 5.38[deg/s]$

which is selected to be the Pitch Maneuver at nominal incidence $A_0 = 10[deg]$, amplitude $A = 10[deg]$ and frequency $f = 0.5[Hz]$, because of the good coverage of the parameter space in terms of α and q, which represent the most correlated parameters in determining the dominant feature of the pressure fields. This is also shown in Fig. 4.8a, where the validation maneuver coverage of the α-q space is represented by the black curve. As the number of clusters is varied in the optimization loop, the Mean Square Projection Error for each validation snapshot is computed, while the assignment of a validation point through a cluster is performed in the global reduced coordinate space as explained in Sec. 3.5. In Fig. 4.8b, the MSE over the validation maneuver is shown for a changing number of clusters (k=1,...,4): it is obvious that for the vast majority of timesteps the error decreases with k due to the more localized content of the local bases, yielding a considerable improvement to the global basis performance. Despite the general trend, some isolated peaks in the MSE indicator can be detected in correspondence with the snapshots located near the border of two neighboring clusters: this can be explained by the poor coverage of the local snapshots matrices in said regions of the parameter space.

In order to assess the optimal value for the number of clusters, the summation of the MSE values over the entire validation maneuver is performed, using Eq. 3.16, providing a scalar quantity for each value k. The results of this analysis are displayed in Fig. 4.9b: the minimum total MSE is reached at $k = 4$. The largest improvement appears between one cluster (i.e the global basis) and three clusters, with a reduction of approximately 40% in the total MSE. Although, several other local minima are present at $k = 6$ and $k = 8$ the curve shows an increasing trend: by adding clusters, in fact, the number of border snapshots increases as well, with a detrimental effect on the global error measure. Furthermore, the optimal number of clusters should be chosen as the smallest integer for which a local minimum is detected, in order to address the drawback of training the surrogate model with datasets that are not large enough. Depending on the original database and the snapshot sampling, in fact, a local minimum in the total MSE could be found for a value of the number of clusters that is too large relative to the cardinality of the cluster snapshot set: in such case, a penalization term that accounts for the magnitude of k could be included in the optimization metric.

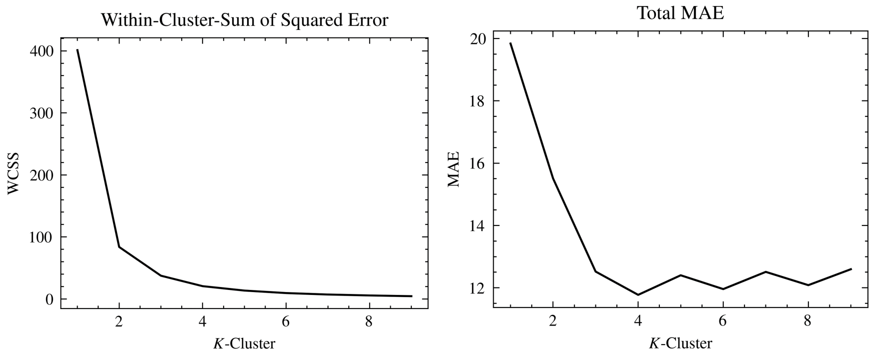


(a) Clusters in the α-q space using $k = 4$. (b) MSE over validation maneuver

Figure 4.8: Optimal clustering of the input space (α-q) and MAE optimization curves. Cluster features: first 5 global Moment-POD coefficients. In black the validation maneuver: harmonic pitch at $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.25Hz$.

The alternative criterion for the choice of k_{opt} is the commonly used "Variance Elbow" method, and requires the computation of the Within Cluster Sum of Squared Variance as in Eq. 3.15: this quantity is calculated for the analyzed case taken and displayed in Fig. 4.9a. The monotonicity of the WCSS does not allow for a simple definition of k_{opt}, which is instead often heuristically identified as the integer for which the curve shows an elbow-like shape. This would correspond to $k = 2$ in the considered plot. This choice, however, would not yield the largest decrease in the total MSE, which instead occurs for $k = 3$. The discrepancy between the results of the two criteria indicates that in many cases, the a-posteriori error indicator can yield a more informed choice of k_{opt} and result in better performance of the resulting ROM. Nevertheless, the additional computational cost and the ambiguity in the choice of the validation maneuver can represent a limiting factor in the adoption of the described criterion, making the Variance Elbow method a more direct and accessible procedure.

The overall performance of the Cluster POD, measured in terms of Mean Absolute projection error and as a function of the number of modes, is displayed for several maneuvers in . 4.10. The maneuvers oscillating around moderate angles of attack ($A_0 = 5[deg]$) are characterized by lower projection errors, even when a small number of modes is considered: for $r = 5$ modes, the error is approximately 10^{-2}, decreasing up to 10^{-3} as the number of modes is increased to 40. The efficient data compression of the local Cluster bases for this range of α is a result of the parameter space clustering. For $\alpha < 12[deg]$, in fact, the testing point are entirely contained inside Cluster 4 and Cluster 2 in Fig. 4.8a: the flowfield topology over this parameter range does not exhibit strong non-linearities, and the contained variations within a cluster can be captured efficiently by a local low-dimensional subspace. For the maneuvers spanning larger portions of the parameter space, the projection error is remarkably higher, showing also a slower convergence rate: the flowfield around $\alpha = 12[deg]$ is characterized by large fluctuations of the dominant coherent structures and large sensitivity with respect to the angle of



(a) WCSS, as in Eq. 3.15.

(b) Total MSE as in Eq. 3.16

Figure 4.9: Within Cluster Sum of Squared Errors and Total MSE on validation maneuver vs number of clusters

attack. This is also visible in Fig. 3.16, when looking at the clustering based on the Pressure POD global features: the largest variations of the global coefficient a_2 are associated with the cluster around $\alpha = 12[deg]$ (colored in purple). Nevertheless, the differences in performances of the local basis on each cluster can be alleviated by choosing different numbers of truncation indices of each local basis: as a consequence, the resulting ROM can have homogeneous accuracy over the entire parameter range. This is not possible using a global basis, as the number of modes is also a global parameter.

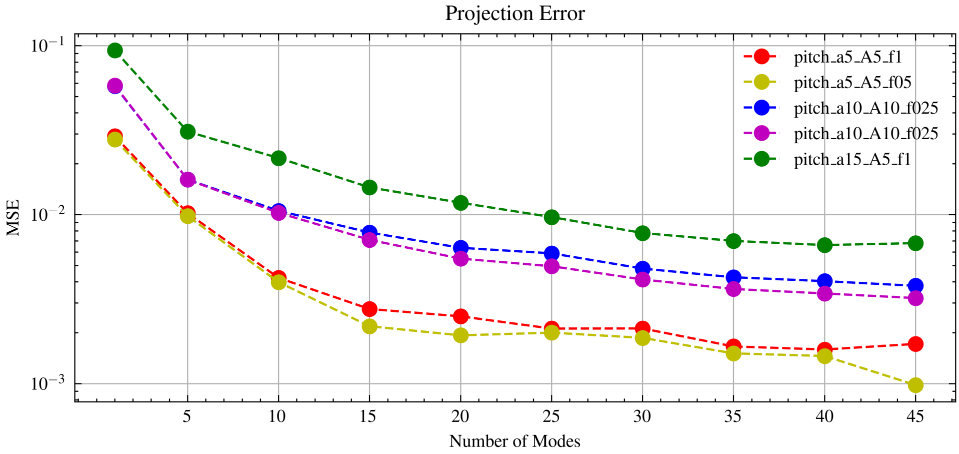


Figure 4.10: Time averaged Mean Squared Projection Error vs the number of modes over several testing maneuvers. $k = 4$ clusters.

Overlapping Clusters with Gaussian Mixture Models

This section is relative to the alternative methodology for the generation of clusters, presented in Sec. 5. In particular, the comparison of the projection error between the K-means and the GMM with overlap clustering is presented. The projection error plots versus time, for the Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.5Hz$ (validation maneuver), comparing the projection error emerging from the use of the truncated Global POD basis, the Local Bases obtained from the K-means clustering and from the GMM clustering with overlap are provided in Fig. 4.11. The GMM and K-means basis produce similar results with the fundamental difference that in proximity of the cluster transition regions $timestep = 65, 205, 265, 405$, the GMM error does not present the characteristic peaks of the K-means methodology.

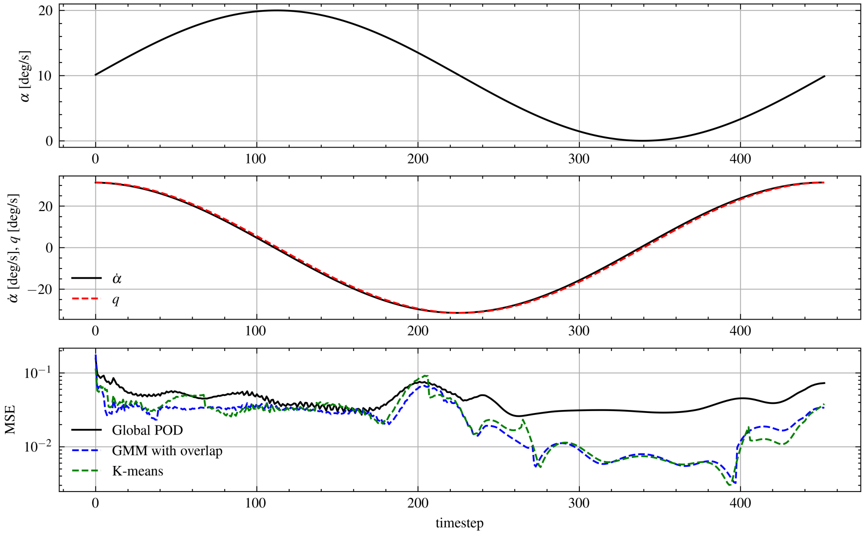


Figure 4.11: Instantaneous inputs and Mean Squared Projection error using Global POD, Cluster POD with overlapping GMM and Cluster POD with K-means. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.5Hz$. Number of modes: 10

Online Stage

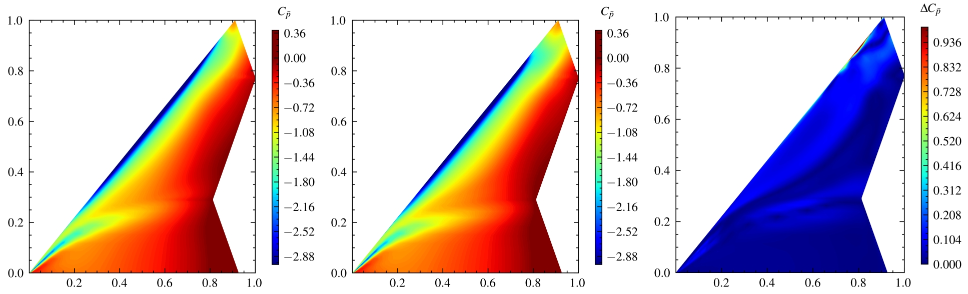
The first step of the online stage of the CPOD-ROM is represented by the assignment of each testing parameter to its corresponding cluster. As the clustering is performed in the (weighed) global POD subspace, the first two global reduced coordinates are predicted through a global surrogate model (LSTM in this case) in an analogous way to the Global POD ROM. Since the prediction of the global coordinates is affected by the intrinsic error of the regressor model, a misplacement of the testing point is expected in the POD subspace with the possibility of the k-means algorithm assigning the point to a cluster different from the exact one (when the true global coordinates are given to the k-means). The wrong cluster assignment can sometimes lead to large projection and network errors for the interested testing point, especially when the distance of the testing point from the true cluster centroid is much smaller than the distance from the assigned one. It is therefore important for the assignment accuracy to be as high as possible, in order to limit those cases. In Tab. 4.2 the assignment accuracy is summarized for some relevant test maneuvers, showing how in most cases, the testing conditions are assigned to the

Maneuver	Accuracy
Pitch $A_0 = 5[deg], A = 5[deg], f = 0.5Hz$	99.7 %
Pitch $A_0 = 10[deg], A = 10[deg], f = 0.5Hz$	98.9 %
Pitch $A_0 = 10[deg], A = 5[deg], f = 0.25Hz$	98.1 %
Pitch $A_0 = 5[deg], A = 5[deg], f = 1Hz$	98.1 %
Pitch $A_0 = 15[deg], A = 5[deg], f = 1Hz$	96.0 %
Pitch $A_0 = 10[deg], A = 10[deg], f = 0.0125Hz$	95.7 %
Pitch $A_0 = 10[deg], A = 10[deg], f = 0.25Hz$	94.5 %

Table 4.2: Assignment accuracy using global LSTM model for different maneuvers.

correct cluster and thus proving the reliability of the clustering predictive algorithm.

Once each testing point is assigned to a cluster, the local basis and surrogate model are used to predict the pressure fields. In Fig. 4.12, the prediction of the CPOD ROM (Fig. 4.12a) are compared with the CFD data (Fig. 4.12b), highlighting the absolute prediction error in Fig. 4.12c. The chosen testing condition is the same as Fig. 4.7, corresponding to a high angle of attack case, where the Global POD displays large errors near the wingtip. The CPOD is instead able to capture with improved accuracy the pressure fluctuations in this region, using a limited number of modes, as it is clear from the comparison of the total error with the plots in Fig. 4.6.



(a) Cluster POD-LSTM

(b) Ground Truth

(c) Total Absolute Error $\epsilon_T(\boldsymbol{x})$

Figure 4.12: Cluster POD-LSTM pressure field prediction, comparison with CFD ground truth data, and absolute prediction error. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$ at $t = 0.25s$. Instantaneous parameters: $\alpha = 19.93[deg], \dot{\alpha} = 5.38[deg/s], q = 5.38[deg/s]$

4.4. Comparative Study Between Methods

In this Section, a comparative analysis of the performance of the Global (scaled) POD, the DD-POD, and the Cluster POD is presented. The decomposition methods are all coupled with the LSTM Network Architecture described in Chapter 3 for the prediction of the global or local low-order dynamics. Although the ROMs are flexible in terms of the surrogate model that can be used for the reduced coefficients, and therefore the considered dimensionality reduction techniques could be coupled with the Multi-Layer-Perceptron and the Gaussian Process Regressor, the goal of this section is to compare the decomposition methods and to highlight the relative advantages and drawbacks of local methods compared to the global POD. Furthermore, the use of the LSTM facilitates the comparison of the obtained results with the POD-LSTM model of Bourier [9].

4.4.1. Projection Error

The projection error on the surface pressure coefficient is a key measure of the offline performance of the ROM. In particular, the efficiency of a decomposition method is a function of the projection error, relative to the dimension of the latent space: the smaller the error for a fixed number of latent coordinates, the more efficient and parsimonious the information compression. In Fig. 4.10, projection error, in terms of time-averaged MSE over a testing maneuver, is presented as a function of the number of modes, for several pitch oscillations centered around three different values of α_0. For all the considered methods, the maneuvers oscillating around $\alpha_0 = 10[deg]$ are characterized by the largest projection errors: this is a result of the input signal design: the sampling is not uniform over the regressor space, but most training samples are concentrated near the boundaries of the $\alpha - q$ plane, leaving the mid-regions with a scarcer coverage. It is interesting to note that for some maneuvers, the projection error curves do not show a monotonous behavior with respect to the number of modes: this is counter-intuitive, as adding dimensions (i.e. adding terms in the POD expansion) should yield a better approximation of the true solution. The explanation for this phenomenon is hidden in the construction of the basis and the choice of the metrics to evaluate the projection error: the employed bases (both local and global) are obtained from the snapshot scaling, as described in Sec. 3.3, and therefore from a modification of the standard L^2 metric (MSE in the discrete setting) with an alternative weighted metric (defined by the weighting matrix that generates the scaling). Consequently, the monotonous convergence of the POD expansion to the true solution is only valid under the alternative metric. In other terms, the Moment-scaled basis approximates better the moment distribution as the number of modes is increased but not necessarily the pressure distribution. Nevertheless, for $r < 20$ modes, the projection error decays rapidly for all the considered test maneuvers.

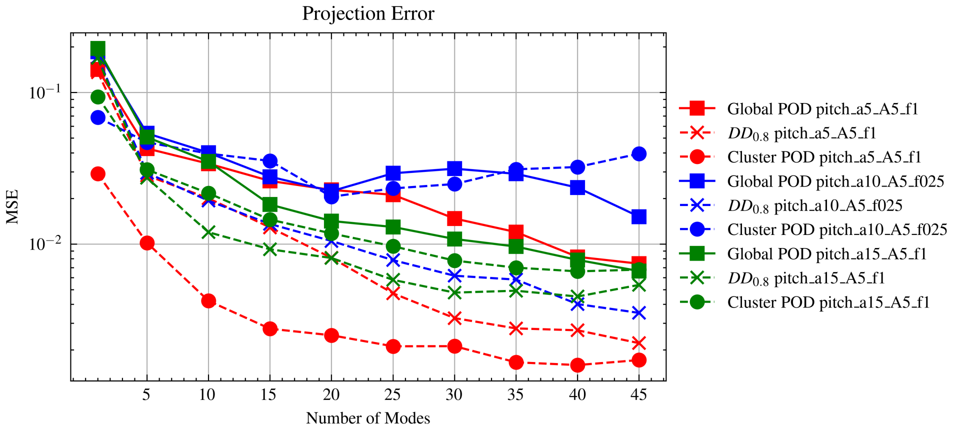


Figure 4.13: Time averaged Mean Squared Projection Error vs number of modes over harmonic pitching maneuvers for the Global POD, $DD_{0.8}$-POD and Cluster POD bases.

For a limited number of modes ($r \leq 5$), the Cluster POD shows the best performance, especially at lower angles of attack ($A_0 = 5[deg]$) where the projection error is three to five times smaller than the DD-POD and the Global POD. For the same maneuver, the Global POD requires 40 modes to achieve a similar error value to the CPOD method with 5 modes. At larger angles of attack $A_0 = 15[deg]$, the CPOD performs consistently better than the Global POD, with the difference being more pronounced for $r \leq 10$ modes. When a larger number of modes is considered, the Domain Decomposition strategy yields the best approximations, displaying the most rapid rate of convergence. This behavior can be explained by looking at the spatial distribution of errors on the wing surface. For most cases, in fact, a small number of modes is enough to accurately capture most pressure fluctuations, except for the wingtip region where the projection error is the highest: adding localized modes in this region yields the largest decrease in projection error, compared to improving the approximation in the rest of the wing where the error is already low.

Similar considerations are valid for the projection error on the plunge maneuvers, displayed in Fig. 4.14, with the main difference that the largest error is obtained for the maneuvers around $\alpha_0 = 15[deg]$.

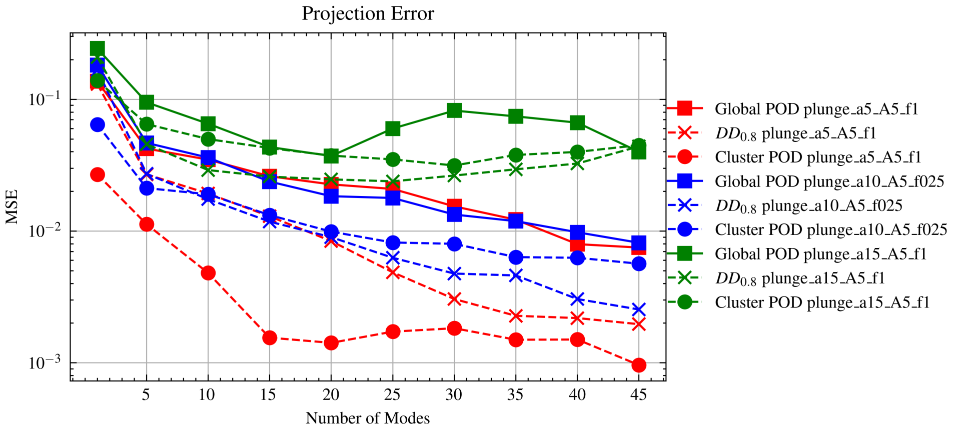


Figure 4.14: Time averaged Mean Squared Projection Error vs number of modes over harmonic plunging maneuvers for the Global POD, $DD_{0.8}$-POD and Cluster POD bases.

4.4.2. TOTAL ERROR

The overall performance of the ROMs is not only dependent on the projection of the reduced bases but is also a function of the performance of the surrogate models for the reduced dynamics. In particular, the use of distinct sets of modes implies that the projection of the full-order dynamics yields datasets with different statistical distributions, on which the same data-driven surrogate models can have different performances in learning the model parameters. In the following the total error plots over time for several maneuvers are presented: the focus here is only to compare the different ROMs and thus a small number of modes (5 modes) for the pressure field reconstruction is considered, in order to better appreciate the models' relative differences. In the Appendix A.4.2, additional plots are provided using a larger number of modes which clearly show a sensible decrease in the total error thanks to the superior size of the corresponding latent spaces.

In Fig. 4.15 the Total Mean Squared error over a maneuver centered at a low angle of attack is plotted, and accompanied by the corresponding values of the inputs over time. The Cluster POD performs better than the DD-POD and the Global POD for all time instants, reducing the total error of the Global POD-LSTM model by one order of magnitude in the range of α between $0[deg]$ and $5[deg]$. The DD-POD error is also smaller than the Global-POD for nearly almost timesteps.

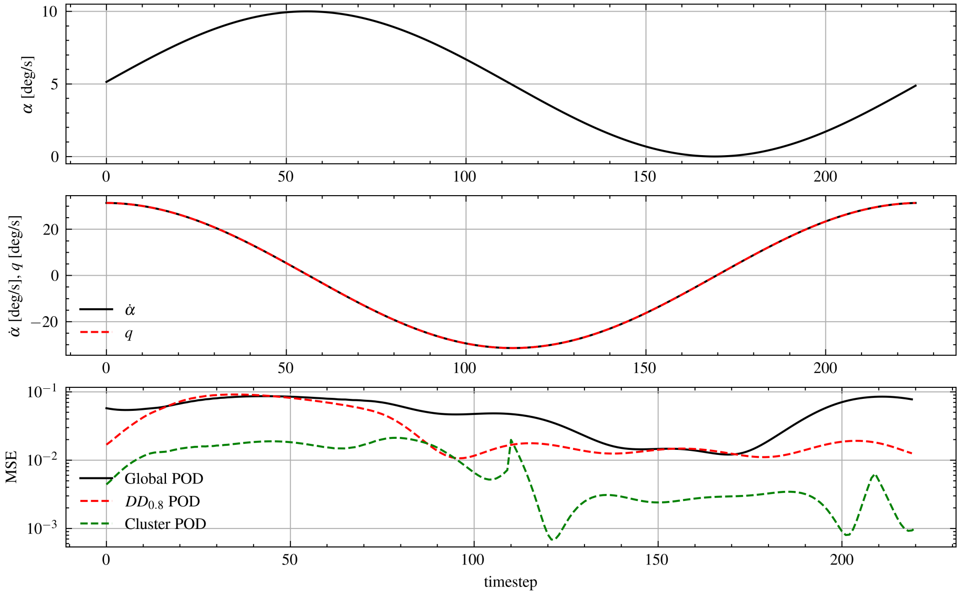


Figure 4.15: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 1Hz$. Number of modes: 5.

In Fig. 4.16, the pitch oscillation around $\alpha_0 = 10[deg]$ is considered: this maneuver crosses three clusters (Cluster 2,3,4 in 4.8a) and for a large number of timesteps (100-

400) the testing conditions are placed close to the border clusters. Due to the reduced coverage of the local bases in these areas, the total error of the CPOD ROM is larger than the one of the Global POD. For the rest of the snapshots, the CPOD and DD-POD solutions show remarkable improvements and again the largest differences are observable in the angles of attack smaller than $10[deg]$.

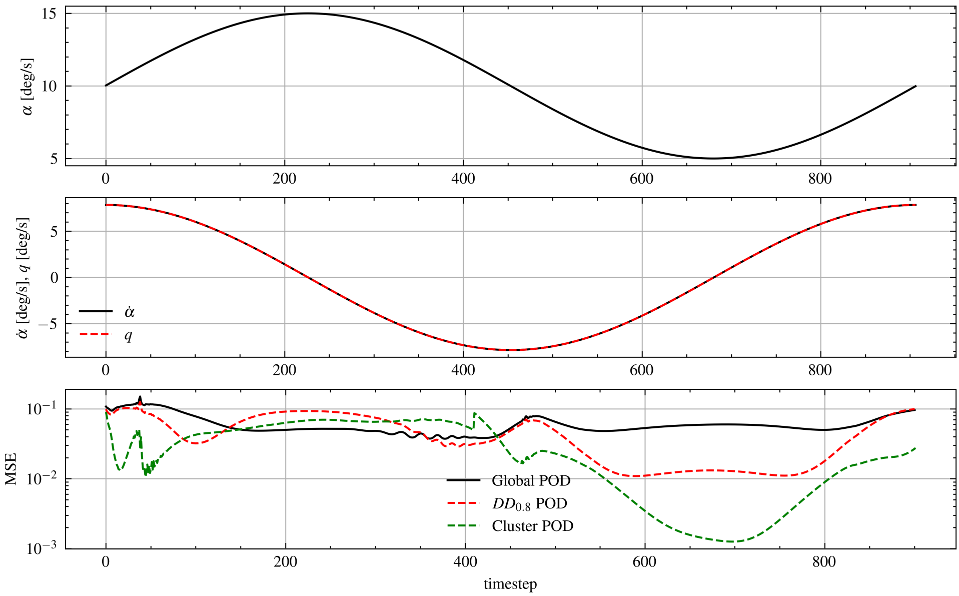


Figure 4.16: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 5

The pitch maneuvers around $\alpha_0 = 15[deg]$ are interesting cases to be discussed since the Global POD ROM shows the largest deficiencies at high angles of attack. The CPOD total has good performance for $\alpha \in [17[deg], 20[deg]]$, corresponding to the local model of Cluster 1. In particular, the largest error of the Global POD (timestep 46) is reduced by a factor of 5 using the CPOD ROM. Nevertheless, the CPOD curve presents several local maxima (peaks), corresponding to cluster transition areas. The considerations made for the pitch maneuvers are also valid in general for the plunge maneuvers, as the main parametric influence in driving the ROMs error is represented by the angle of attack. In Fig. 4.18 the total error plots can almost be overlapped to the plots of Fig. 4.16.

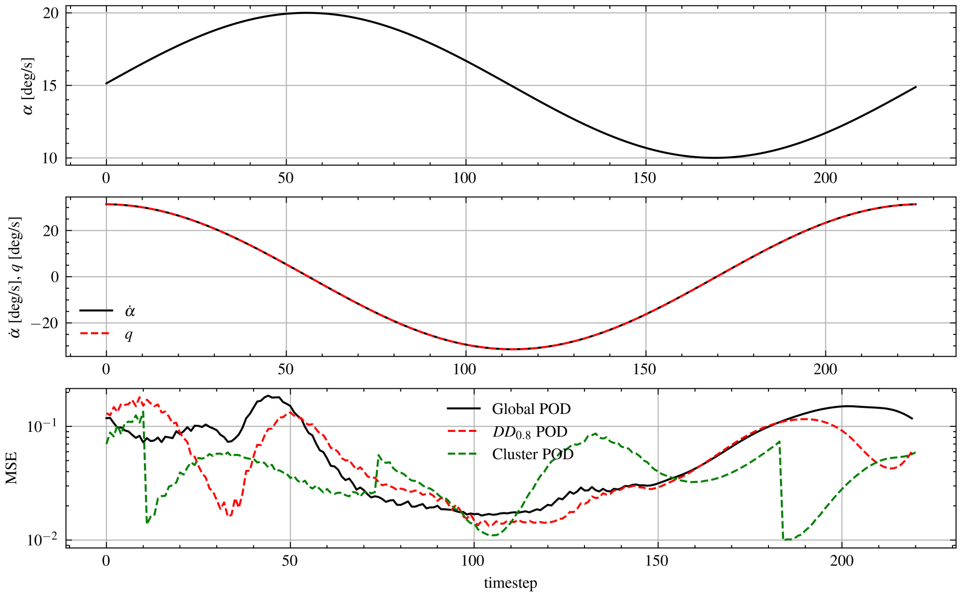


Figure 4.17: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$.Number of modes: 5

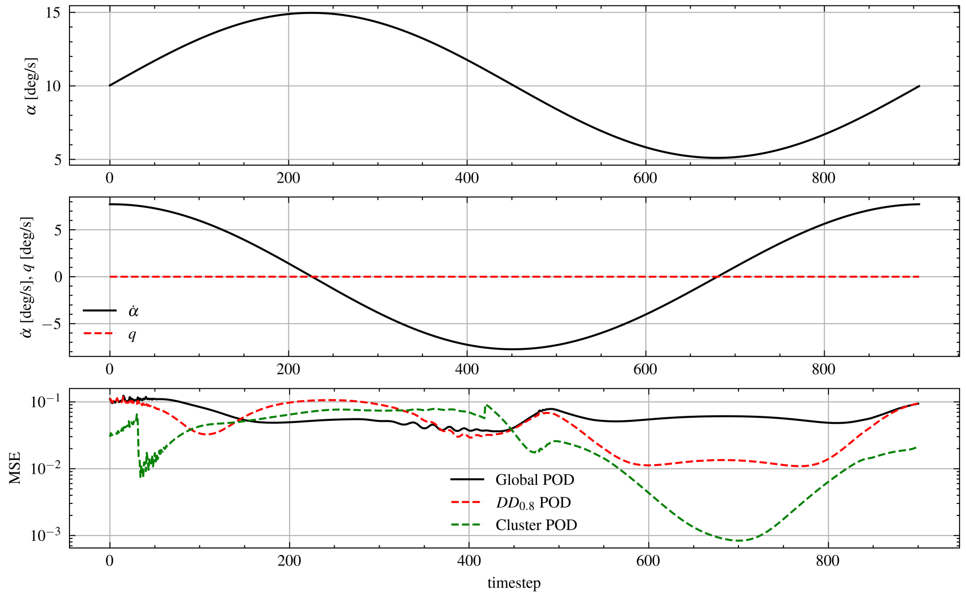


Figure 4.18: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 5.

4.4.3. ROM Computational Cost

The computational cost of the developed ROMs can be quantified by looking at the total elapsed time during the computations. The offline phase of the ROMs represents the most computationally demanding part of the models' construction: it is constituted by the determination of the Reduced Basis (through one or more SVDs on the snapshot matrices), and the training of the single or multiple regressor models for the low-order dynamics. In this work, the SVDs are performed using the `Python` routine `numpy.linalg.eigh` on the correlation matrix and are run on a personal laptop CPU AMD Ryzen™ 5 3500U (2.1 GHz). The trainings of the regressor models are run on a single GPU NVIDIA GeForce RTX 3080.

The Global SVD takes approximately 40 [sec] on the global snapshot matrix of dimensions [7335, 5666], while the two Domain Decomposition Local SVDs take 43 [sec], and four SVDs on the local Cluster Snapshot matrices approximately 135 [sec]. For all methods, the cost of determining the reduced bases is extremely low and could be further cut down by parallelizing the procedure for the local bases or by employing online SVD algorithms [39]. It must be noted that independently from the number of modes used for the POD expansion, the cost of a standard SVDs does not change. For the Cluster POD, the optimization loop for the selection of the optimal number of clusters requires the determination of several sets of local bases and the computation of the a-posteriori error indicator: this translates into a sensible increase in the computational time.

The training time of the regressor models strongly depends on the type of Architecture employed and the choice of the Decomposition Method. The Global POD results in a

ROM	Model	Offline [s]	Online [s]
Global POD	LSTM	2111	5
	MLP	259	4
	GPR	312	5
Cluster POD	LSTM (×4)	2344	7
	MLP (×4)	744	6
	GPR (×4)	821	6
DD POD	LSTM (×2)	4939	6
	MLP (×2)	560	6
	GPR (×2)	723	5

Table 4.3: Offline and Online computational time of the considered ROMs equipped with different surrogate models. Number of modes: 5.

unique set of input-output pairs and thus only one training of the regressor model, while the Local POD methods require the training of multiple regressor models for each local basis (2 for DD-POD and 4 for Cluster-POD). For the Cluster-POD, however, a remarkable decrease in the total training time is observed, as the sizes of the local reduced coordinate datasets are reduced by the partition of the snapshot set, yielding smaller training time per epoch (fewer batches) and at the same time a smaller number of epochs due to the Networks overfitting smaller datasets.

The LSTM Network results in the largest training times, due to the backpropagation of gradients in time, typical of Recurrent Neural Networks. The Multi-Layer-Perceptron and the Gaussian Process Regression demonstrate superior efficiency, with remarkably lower training times. The GPR on the other hand typically requires larger memory storage: ANNs once trained, in fact, only need to store weight and biases in order to perform predictions, while the entire training dataset is needed for GPR to make inferences at new operating conditions.

The computational time of the Online phase of the ROM is typically negligible, and depending on the number of points at which the predictions are performed can differ by a few seconds. In Table 4.3, a summary of the online and offline computational time for the three different ROMs is given, depending on the type of regressor model employed. In this specific case, 5 reduced coordinates are predicted and used for the online reconstruction of the pressure fields.

4.5. Direct coefficient Prediction and Surrogate Model comparison

The integral load coefficients (Normal Force Coefficient C_N, Axial Force Coefficient C_A, and Pitching Moment Coefficient C_{M_y}) can be included as a direct prediction of the surrogate model. Since the reduced coordinates and the integral loads are scalar outputs defined over the same input space, the same models used to learn the map from the parameter space to the reduced order space can be trained to output the coefficients. As explained in Sec. 3.6.5, this procedure is computationally more efficient and does not introduce additional sources of error other than the network error, compared to the integration of the predicted pressure fields which are instead affected by the projection error resulting from the dimensionality reduction.

The predictions are performed using the three different data-driven methods considered: the Multi-Layer-Perceptron, the Long Short Term Memory Network, and the Gaussian Process Regression. Furthermore, two different variants of the LSTM Network are considered. The first one is the analogous architecture used for the prediction of the reduced coordinates and used for the generation of the results included in the previous sections, whose hyperparameters are also summarized in Sec. 3.6.4. The second LSTM (small) is a downscaled version, with a lower number of units, which resembles the MLP architecture, thus allowing for a fair comparison of the two ANNs.

4.5.1. Integral loads coefficient prediction

The predictions of the integral loads coefficient are displayed in Fig. 4.19 for a maneuver at low angles of attack and high oscillation frequency: all the investigated methods accurately capture the axial and normal force coefficient variation over time. Some discrepancies can be detected in the prediction of the pitching moment coefficient: the LSTM and the MLP Networks achieve higher accuracy compared to the GPR which overestimates the maximum value of the coefficient. Considering a maneuver centered at a larger angle of attack, in Fig. 4.20, the variation of the pitching moment coefficient presents more pronounced non-linearities, deviating from the sinusoidal trend of the force coefficients in the range of α between $10[deg]$ and $20[deg]$. As a result, the Neural Network based methods show greater accuracy than the GPR, thanks to the superior capabilities of modeling non-linear dynamics. The LSTM, in particular, is able to better capture (with half of the average MSE) the general trend compared to the MLP: this can be due to the pre-defined time sequential structure of how the Network process the input data.

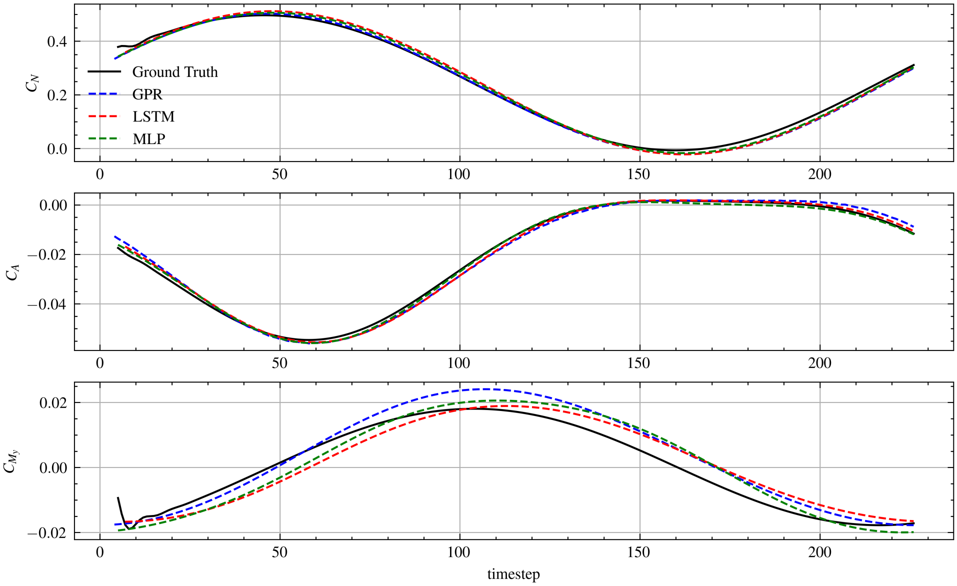


Figure 4.19: Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 1Hz$

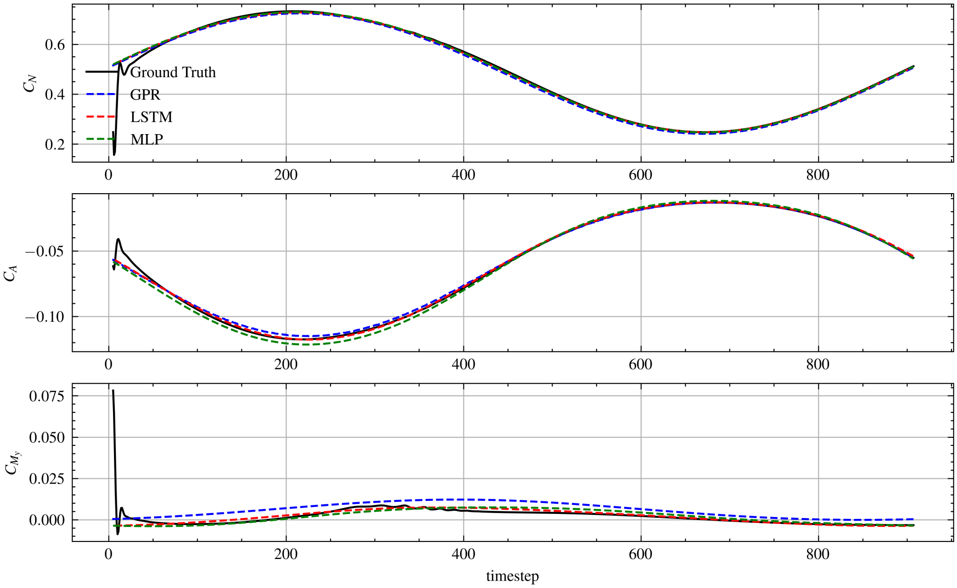


Figure 4.20: Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.25Hz$

Analyzing the results for large values of α in Fig. 4.21, it is clear how the performances of the three methods do not differ when the output has a sinusoidal shape with the same frequency of the input maneuver and therefore can be inferred using a simple linear regression (as in the normal force coefficient C_N). However, the flowfield non-linearities appearing at larger angles of attack influence more strongly the value of the pitching moment coefficient: the use of an ANN model is therefore beneficial to predict those non-linearities.

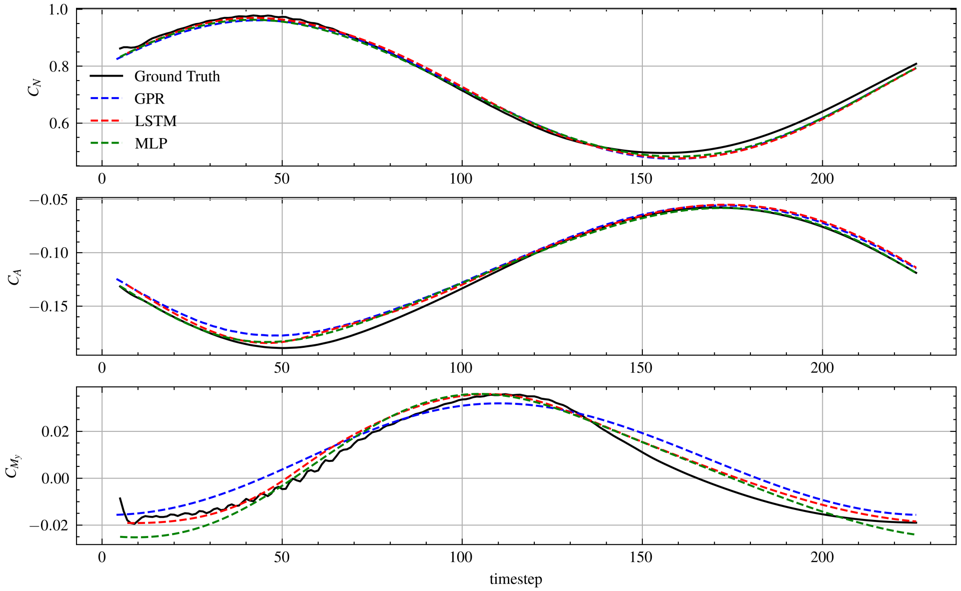


Figure 4.21: Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$

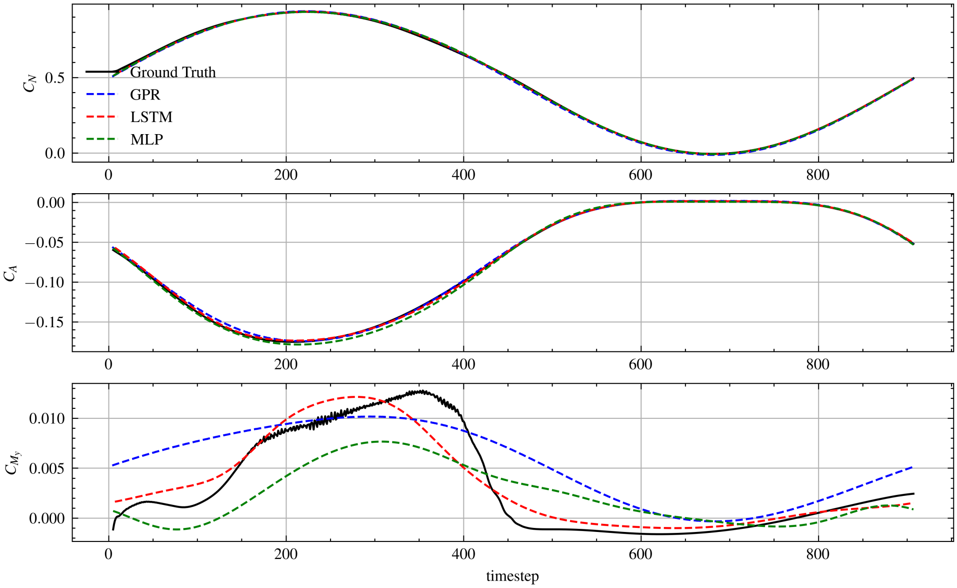


Figure 4.22: Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.25Hz$

4.5.2. Accuracy and Efficiency of the Surrogate Models

The accuracy and efficiency of the various surrogate models on the integral loads' prediction regression task. In Table 4.4 the Mean Square Error in the prediction of the axial force coefficient, normal force coefficient, and pitching moment coefficient are compared for the different surrogate models over several testing maneuvers. Again, the smallest errors are found for the maneuvers oscillating at lower angles of attack, with a weak dependence on the oscillation frequencies. The normal force coefficient is predicted accurately by all regressors, without considerable discrepancies between models. The largest differences in the prediction errors can be appreciated for the pitching moment coefficient: the NN approaches present higher accuracy compared to the GPR, with the larger LSTM network displaying slightly higher performance in general.

The training time for the integral loads coefficient prediction task is given in Table 4.5. Similarly to the reduced coordinates predictions, the trainings are run on a single GPU. As expected, a larger training time is observed for the LSTM networks compared to the MLP and the GPR. In particular the small LSTM architecture, despite having a similar size in terms of layers and units per layer compared to the MLP, takes on average double the time per epoch.

Maneuver	Model	MSE C_A	MSE C_N	MSE C_{M_y}
Pitch $A_0 = 10[deg], A = 5[deg], f = 0.25Hz$	LSTM	3.24e-06	3.30e-04	4.69e-06
	LSTM (small)	3.91e-06	3.43e-04	6.19e-06
	GPR	5.51e-06	5.91e-04	4.03e-05
	MLP	1.00e-05	5.60e-04	1.92e-05
Pitch $A_0 = 10[deg], A = 5[deg], f = 1Hz$	LSTM	1.12e-05	1.36e-03	3.11e-05
	LSTM (small)	1.44e-05	1.31e-03	2.51e-05
	GPR	1.49e-05	1.99e-03	1.23e-04
	MLP	2.04e-05	1.95e-03	9.95e-05
Pitch $A_0 = 15[deg], A = 5[deg], f = 0.5Hz$	LSTM	4.33e-06	1.10e-04	1.29e-05
	LSTM (small)	4.26e-06	1.04e-04	1.15e-05
	GPR	1.12e-05	1.42e-04	2.78e-05
	MLP	3.22e-06	7.45e-05	8.31e-06
Pitch $A_0 = 15[deg], A = 5[deg], f = 1Hz$	LSTM	2.46e-05	2.41e-04	1.26e-05
	LSTM (small)	9.95e-06	2.43e-04	1.68e-05
	GPR	4.20e-05	2.59e-04	3.40e-05
	MLP	1.66e-05	1.92e-04	1.75e-05
Pitch $A_0 = 5[deg], A = 5[deg], f = 0.25Hz$	LSTM	1.53e-07	2.82e-06	2.10e-07
	LSTM (small)	7.06e-08	1.78e-05	3.03e-07
	GPR	1.60e-07	3.09e-05	1.12e-05
	MLP	5.19e-07	2.88e-06	1.61e-06
Pitch $A_0 = 5[deg], A = 5[deg], f = 1Hz$	LSTM	1.64e-06	2.26e-04	1.66e-05
	LSTM (small)	1.24e-06	1.43e-04	8.66e-06
	GPR	3.09e-06	1.58e-04	1.79e-05
	MLP	9.20e-07	1.41e-04	1.54e-05
Plunge $A_0 = 10[deg], A = 10[deg], f = 0.25Hz$	LSTM	3.09e-06	1.96e-05	1.99e-06
	LSTM (small)	8.33e-06	5.13e-05	3.36e-06
	GPR	3.45e-06	6.97e-05	1.16e-05
	MLP	7.97e-06	2.97e-05	9.26e-06
Plunge $A_0 = 10[deg], A = 5[deg], f = 0.25Hz$	LSTM	8.58e-06	1.05e-04	3.11e-06
	LSTM (small)	7.27e-07	4.35e-05	3.15e-06
	GPR	3.47e-06	8.16e-05	2.30e-05
	MLP	3.42e-06	2.42e-05	2.41e-06
Plunge $A_0 = 10[deg], A = 5[deg], f = 1Hz$	LSTM	6.31e-06	2.31e-04	3.64e-06
	LSTM (small)	4.15e-06	3.09e-04	8.03e-06
	GPR	7.37e-06	3.38e-04	3.07e-05
	MLP	3.94e-06	2.79e-04	1.58e-05
Plunge $A_0 = 15[deg], A = 5[deg], f = 0.25Hz$	LSTM	6.78e-06	4.81e-05	5.64e-06
	LSTM (small)	8.79e-06	5.83e-05	8.62e-06
	GPR	1.20e-05	9.71e-05	2.28e-05
	MLP	8.72e-06	4.38e-05	6.10e-06

Table 4.4: Prediction error on the integral loads coefficient of different surrogate models for several maneuvers.

Model	**Epochs Trained**	**Training Time [s]**	**Training Time per epoch [s]**
LSTM	1472	1937	1.3
LSTM (small)	993	537	0.54
GPR	70	16	0.22
MLP	544	126	0.23

Table 4.5: Training time comparison of different surrogate models for the direct prediction of integral loads coefficients.

5 Conclusion

The conclusion to this work can be presented by answering the research questions formulated in the introductory chapter of this report.

"What are the limitations of the POD-LSTM method, and how can Local ROMs increase the performance of the existing ROM in the prediction of the unsteady pressure fields and integral loads of the UCAV MULDICON test case?"
The POD-LSTM method is a Reduced Order Model based on linear dimensionality reduction and recurrent Latent Dynamics modeling that improves the computational efficiency of the previously developed NN-based ROM [51], allowing for the inclusion of time-history effects and greater interpretability of the ROM. The main limitations of this method are a result of the use of the global POD basis: the projection of the full order dynamics into a lower dimensional linear subspace leads to high prediction errors driven by the projection error when a limited number of modes are considered. Local ROMs can be defined on the computational domain through Domain Decomposition, reducing the projection error in critical regions of the wing surface, compared to the Global POD, for the same fixed number of modes. Clustering the snapshots according to global features can lead to the construction of Local ROMs in the parameter space with improved physical meaningfulness and projection efficiency. The procedure of scaling the snapshot according to the force or moment distribution is here demonstrated to be optimal in order to obtain better predictions of the integral loads' coefficients. Nevertheless, directly predicting the force and moment coefficients from the input parameters through a surrogate model, thus bypassing the issue of integrating pressure fields affected by projection error, proves to be the most accurate method.

Can the Cluster POD local bases improve the reduced representation of the full-order system? How can the snapshot set be optimally clustered?
The Cluster POD bases in most cases display a remarkably smaller projection error for the harmonic oscillating test maneuvers compared to the Global POD basis. The lim-

itations of the local bases are related to the approximations of the snapshots located near the cluster borders, due to the limited coverage of the local bases in those regions. Nevertheless, a procedure based on Gaussian Mixture Models to generate overlapping clusters, described in Section , allows to overcome this limitation. The best clustering features are represented by the first global POD coefficients representation of the snapshots. The number of clusters can be chosen through an optimization procedure based on an a-posteriori indicator of the projection error on a validation set of snapshots.

Can the Domain Decomposition local bases improve the accuracy of the ROM in specific regions of the wing compared to the standard POD? How does the decomposition geometry influence the spatial accuracy of the resulting ROM?

The Domain Decomposition provides sets of modes that are more representative of the local flow features on the wing surface. The POD L^2 minimization problem is solved separately on each domain and, depending on the geometry decomposition, more weight can be given to a domain in the reduced representation. For the MULDICON UCAV test case, the isolation of the wing-tip regions is useful to reduce the high-projection error of the Global POD basis for the same number of modes.

How do the performances of an LSTM Neural Network compare to a simpler Multi-Layer-Perceptron and to a Bayesian approach based on Gaussian Process Regression?

The LSTM model appears to perform better in terms of accuracy compared to the Multi-layer-Perceptron and the Gaussian Process Regression for the direct prediction of the integral loads. The advantage of using a Neural Network is more clear when reproducing non-linear mappings between the input and the output features, such as in the pitching moment coefficient prediction. Compared to the MLP, the Recurrent nature of the LSTM appears to be more suitable to process time-sequential data. Nevertheless, both the MLP and the GPR approaches demonstrate smaller training time compared to the LSTM Network.

Can the ROM prediction accuracy of the integral loads coefficient be increased through a modification of the POD basis? Is there a more efficient procedure to predict these quantities?

Pre-scaling the pressure field snapshots in order to obtain Force and Moment distribution along the wing, as described in Section 3.3, is shown to be the optimal (linear) way to generate a POD basis that minimizes the error on the integral loads' projection. In order to avoid the projection error, the same surrogate models used for the reduced coordinates predictions can be trained to predict the integral loads' coefficients.

5.1. RECOMMENDATIONS FOR FUTURE WORK

The presented models address some of the limitations of the global POD-LSTM, mainly related to the dimensionality reduction method, by constructing local reduced-order bases. Future research can focus on improving several aspects of dimensionality reduction:

- **Cluster POD**:
  The methodology described in Section 5 based on Gaussian Mixture Models overlapping clusters can be used to reduced the limitation of the K-means cluster POD at the cluster interfaces and should be tested also for online prediction. Adaptive resampling of the parameter space, as proposed by Dupuis et al. [16], could further improve the performance of the Local Bases by increasing the snapshot density in said regions.

- **Domain Decomposition POD**:
  The choice of the geometrical decomposition could be optimized to minimize the projection error on the local bases. This model should be tested in the case of flow discontinuities, for instance, in the transonic flow regime.

- **Scaled POD**:
  The procedure of scaling the snapshots can be tailored to a specific goal functional, following a Goal Oriented approach [11, 12]. The flexibility in decoupling the governing equations from the constraint, makes the use of the Neural Network (or Gaussian Process Regression) for the adjoint problem possible, allowing to exploit the beneficial properties of ANNs in terms of differentiability.

Concerning the surrogate models for the reduced dynamics, of great interest could be investigating the following:

- **Uncertainty Quantification of the ROM outputs**:
  The Gaussian Process Regression directly allows for the evaluation of the uncertainty, based on the variance of the posterior distribution. Standard Artificial Neural Networks, however, do not directly provide this quantity. Modern approaches, based on Bayesian Neural Networks [2] or Montecarlo Based methods [19] could be leveraged to quantify the surrogate models' uncertainty.

- **Autoregressive methods**:
  Inserting the predictions at previous time steps in the inputs for the predictions of future time steps could yield an increase in accuracy in the reduced coefficient prediction. This autoregressive approach is often used in ROMs for fluid dynamics [46], due to its similarity to time integration schemes.

Moreover, future work should focus on testing the models on more complex dynamics, by increasing the number of Degrees of Freedom for the aircraft maneuver or by including variations in the Mach number. This step requires the generation of additional CFD data and therefore the identification of adequate sampling strategies: greedy approaches to sampling could be considered [53, 1], in order to adaptively sample the parameter space in the regions of higher projection error.

A

Additional Results

A.1. Special Maneuver: Sharp Pitch up-down

The developed ROMs are further tested on a special type of maneuver that differs from the oscillating pitching and plunging maneuver considered in the report: the sharp pitch up-down motion. The motion variables variation over time is described in Fig. A.2: the aircraft is initially at an incidence of $15[deg]$ and zero angular and pitch rate (the vertical component of the velocity is zero and the aircraft is moving horizontally at constant velocity). A step variation of $20\frac{[deg]}{s}$ in the pitch rate is then introduced, which produces a steep constant variation in the angle of attack from $15[deg]$ to $22[deg]$, during which, in order to keep the total velocity vector magnitude constant, the horizontal component decreases while the vertical component increases (in absolute value). After reaching $22[deg]$, the pitch rate is inverted to $-20\frac{[deg]}{s}$, and the angle of attack is symmetrically linearly decreased up to $15[deg]$.

The time-average projection error for a varying number of modes is displayed in Fig. A.1 as a result of using the Global POD and the Local methods. For a limited number of modes, the Cluster POD and the DD-POD show better compression capabilities compared to the global basis. This is true also for larger number of modes for the DD-POD, wh̀ich demonstrates to better capture the vortical structures appearing at the wing-tip at higher incidences.

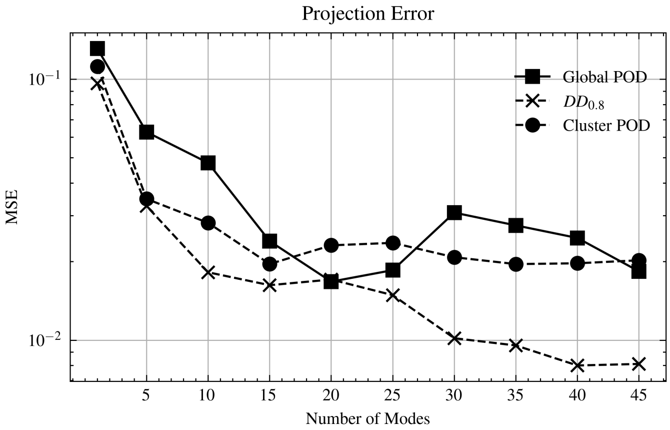


Figure A.1: Time averaged Mean Squared Projection Error vs number of modes over sharp pitch up-down maneuver for the Global POD, $DD_{0.8}$-POD and Cluster POD bases.

The total error for a 5 modes reconstruction of the pressure fields, in terms of mean square error is displayed in Fig. A.2. Most of the differences between the ROMs are visible at higher angles of attack in the range $[20,22][deg]$ where the Cluster POD presents the lowest error, thanks to the localized set of modes which efficiently captures most of the dominant features in said region of the input space. For intermediate values of the incidence, the Global POD yields better performance, thanks to the better coverage of the regressor space.

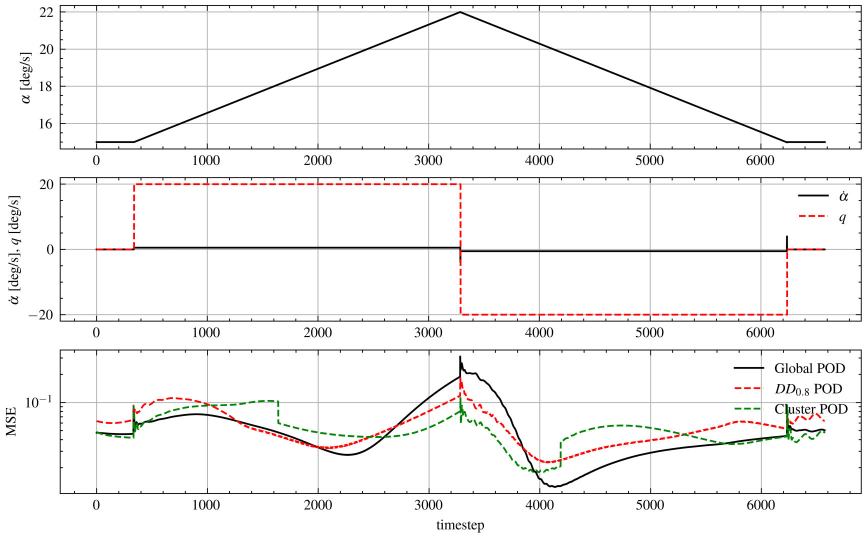
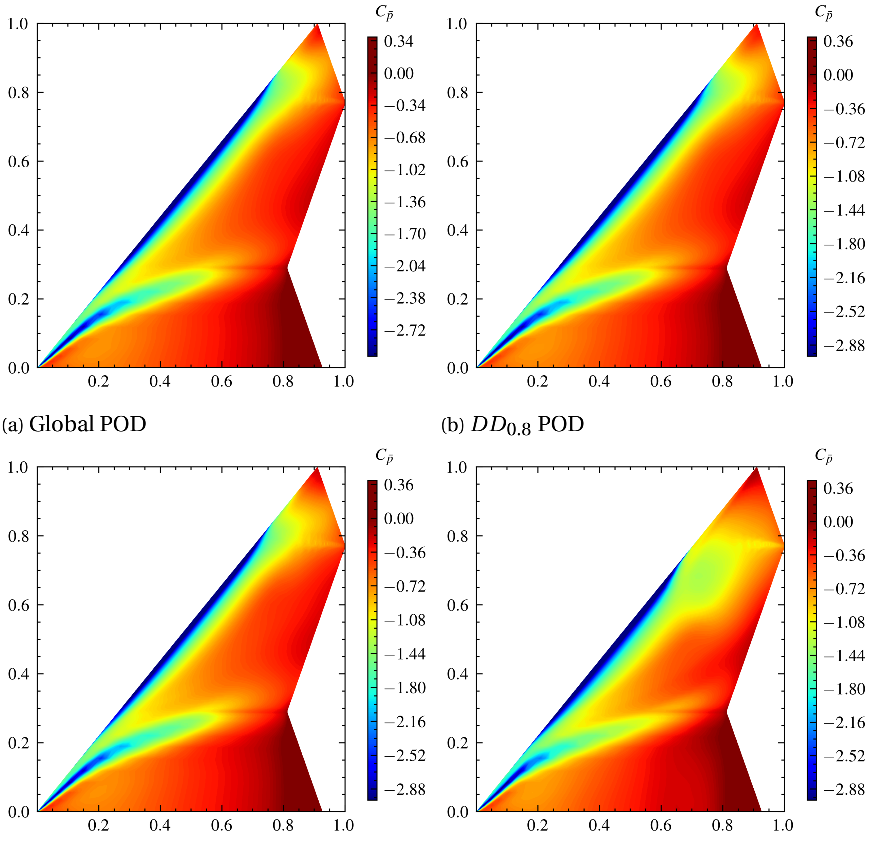


Figure A.2: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Sharp pitch up-down maneuver.Number of modes: 5

The pressure fields predictions, using 5 modes, for the different ROMs are displayed for the case with the highest total error. Most of the error is concentrated towards the outboard segment of the wing where the vortical structures create a large region of low pressure, highly underestimated by the ROMs. The low-pressure trace of the main apex vortex is predicted slightly better by the Cluster-POD.



(a) Global POD

(b) $DD_{0.8}$ POD

(c) Cluster POD

(d) Ground Truth

Figure A.3: Pressure field prediction for highest error case of pitch up-down maneuver. $\alpha = 21.95[deg], \dot{\alpha} = -0.58[deg/s], q = -20[deg/s]$

The direct prediction of the integral loads coefficients via the surrogate models, provides good accordance with the CFD data. Most of the errors are concentrated near the pitching rate inversion zone.

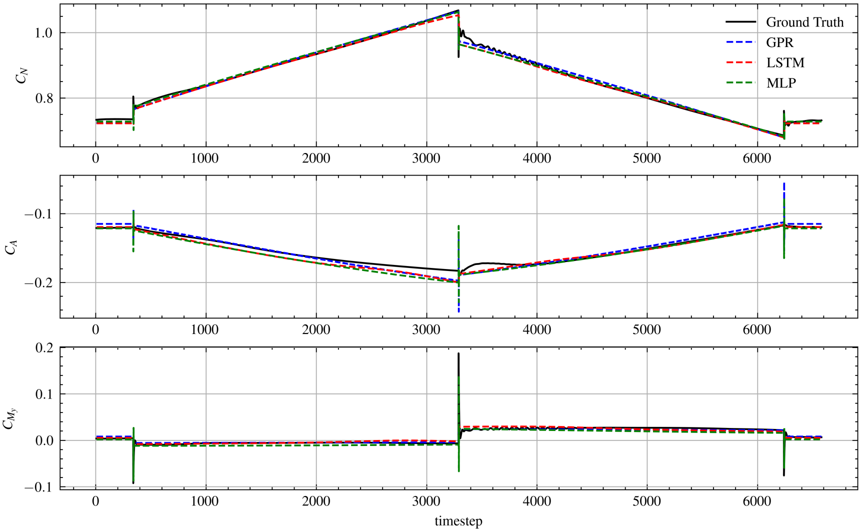


Figure A.4: Direct Integral Loads coefficient prediction using the three different surrogate models. Sharp pitch up-down maneuver.

A.2. Direct Load Predictions

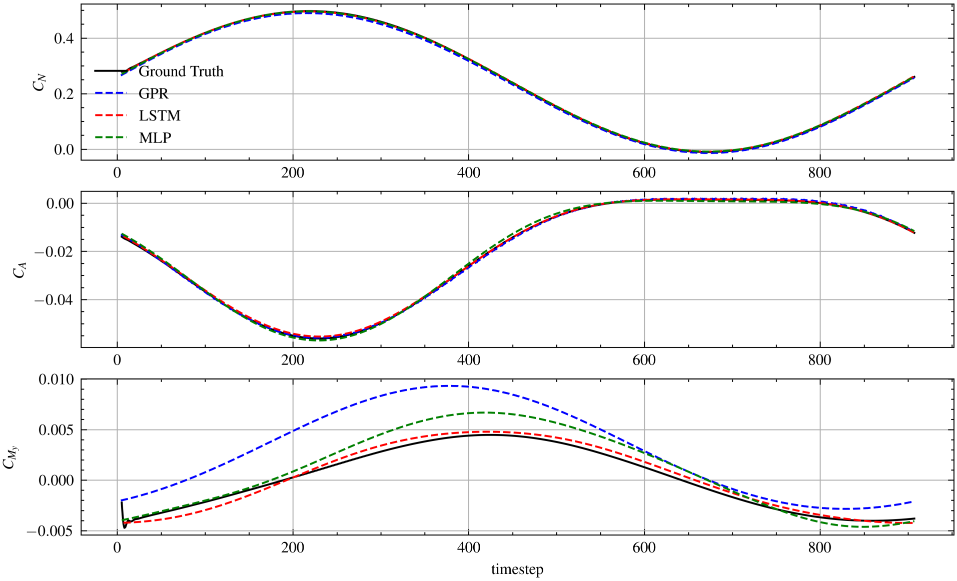


Figure A.5: Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$

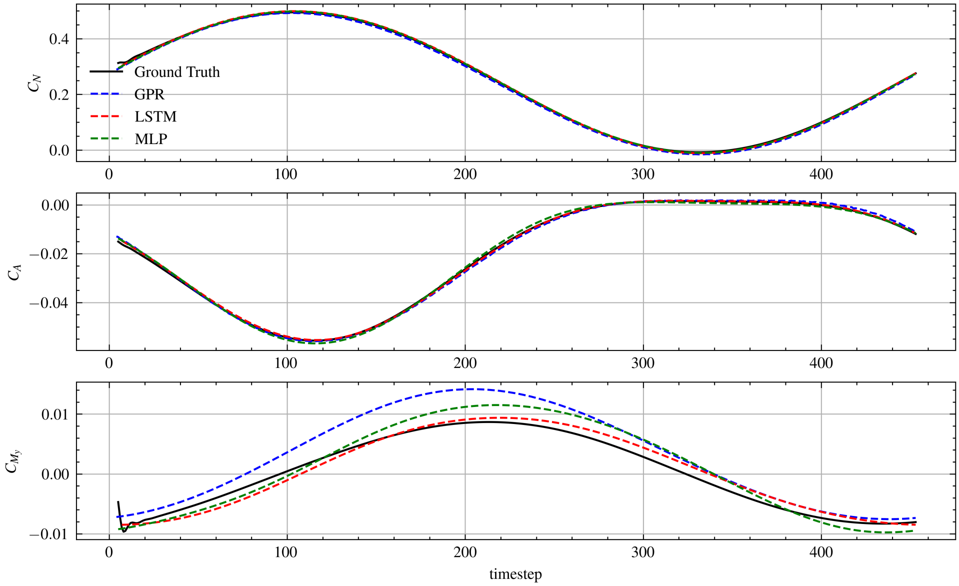


Figure A.6: Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.5Hz$

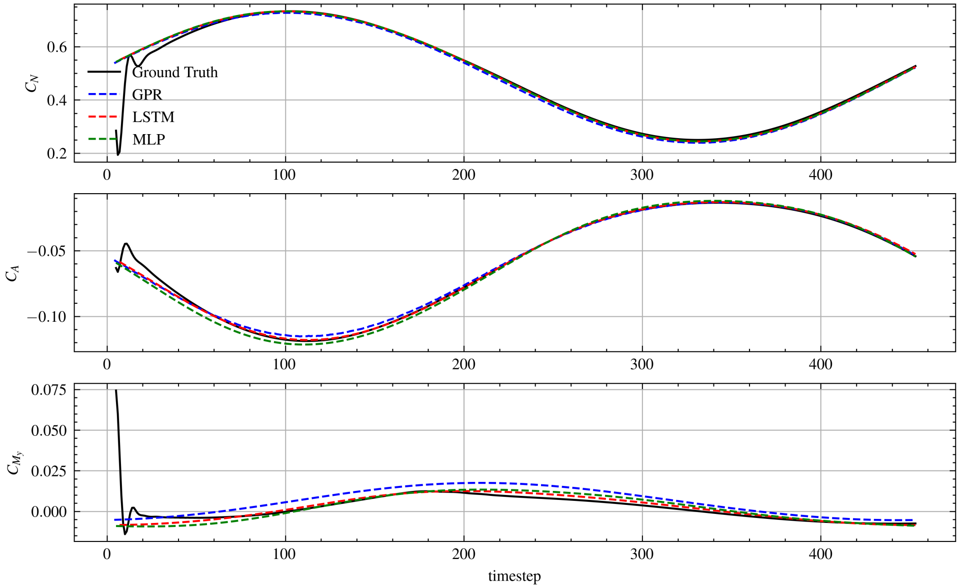


Figure A.7: Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.5Hz$

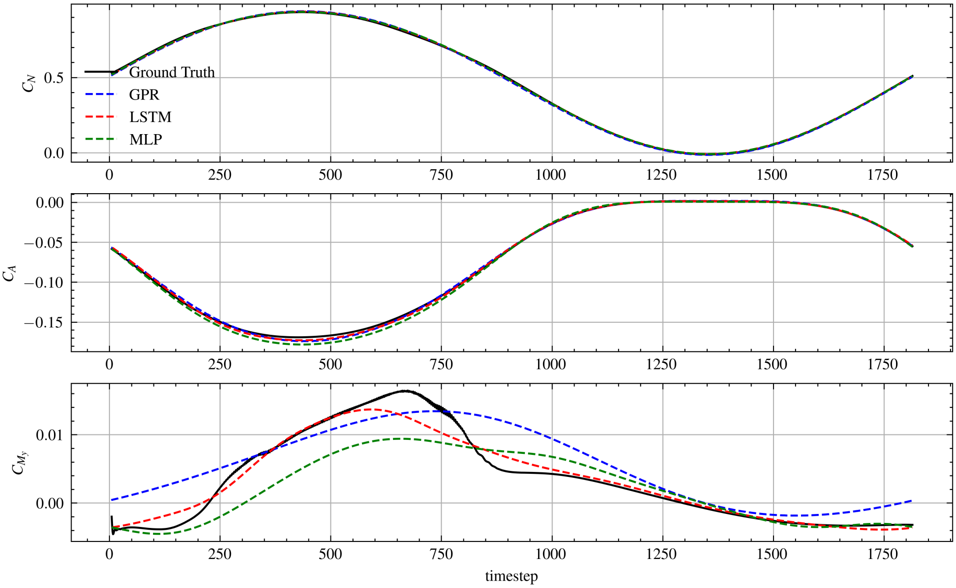


Figure A.8: Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.0125Hz$

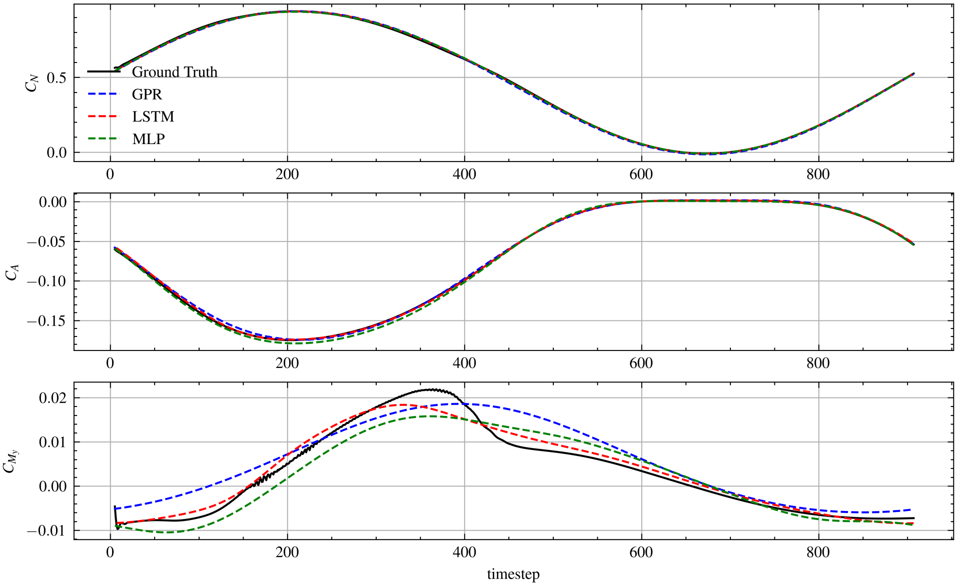


Figure A.9: Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.25Hz$

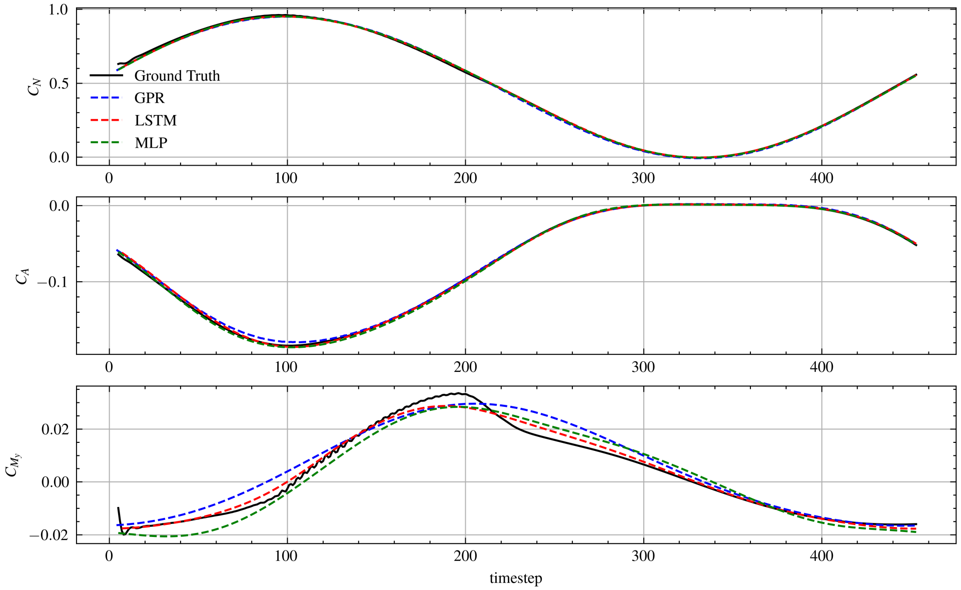


Figure A.10: Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.5Hz$

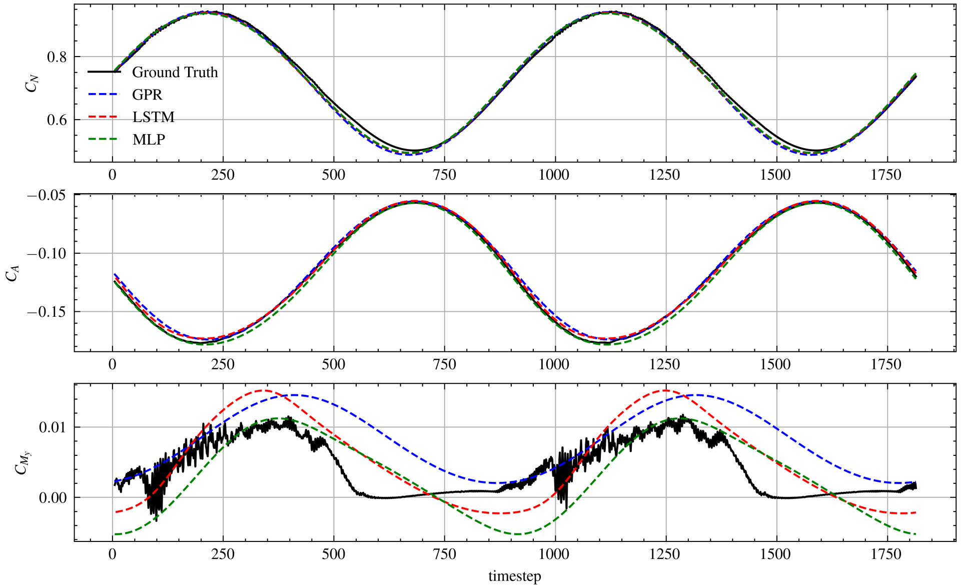


Figure A.11: Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.25Hz$

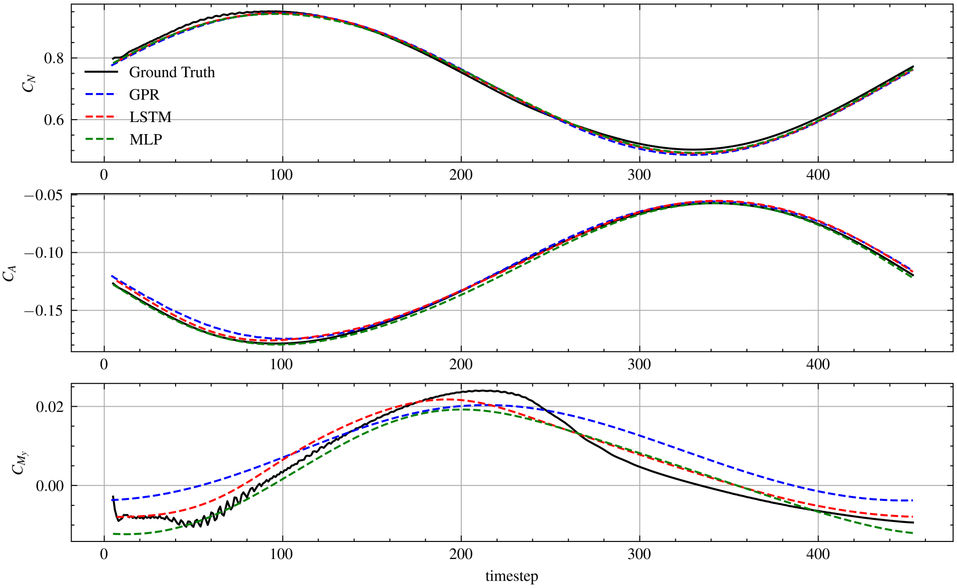


Figure A.12: Integral Loads coefficient prediction using the three different surrogate models. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.5Hz$

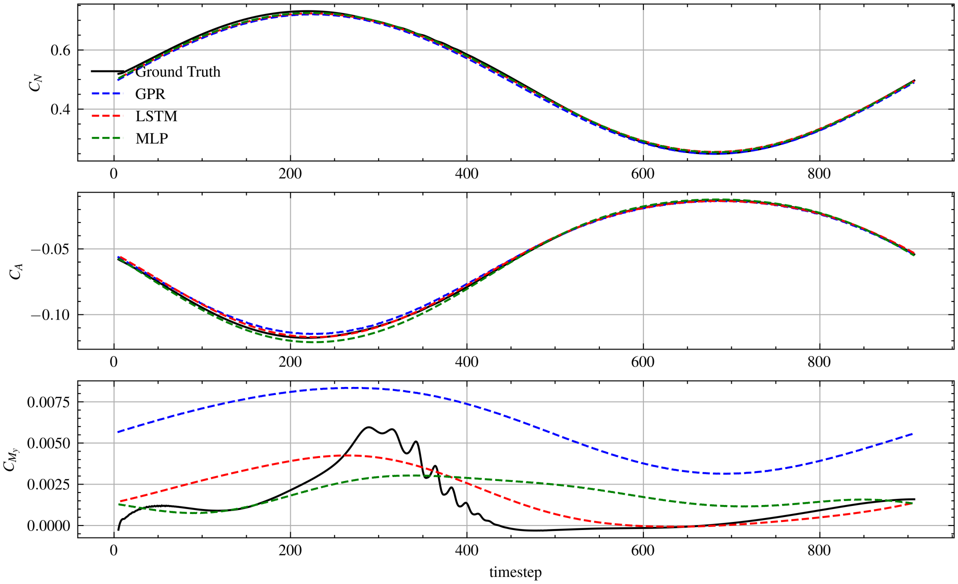


Figure A.13: Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.25Hz$

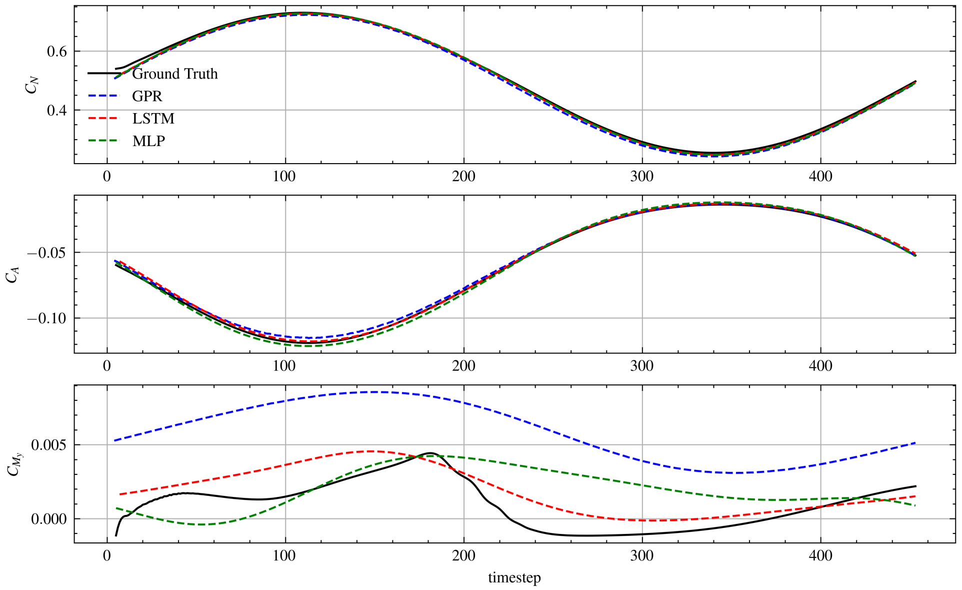


Figure A.14: Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.5Hz$

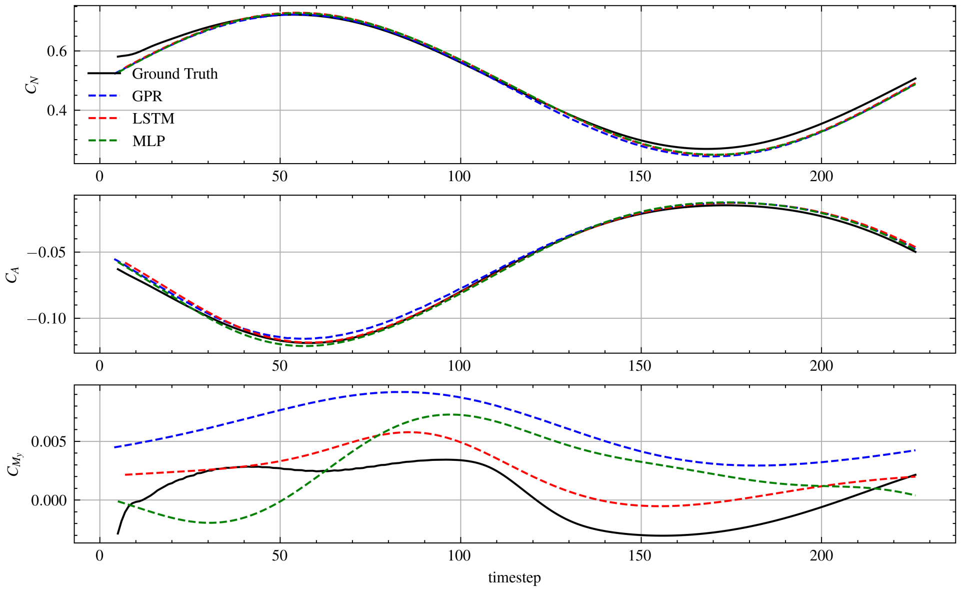


Figure A.15: Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 1Hz$

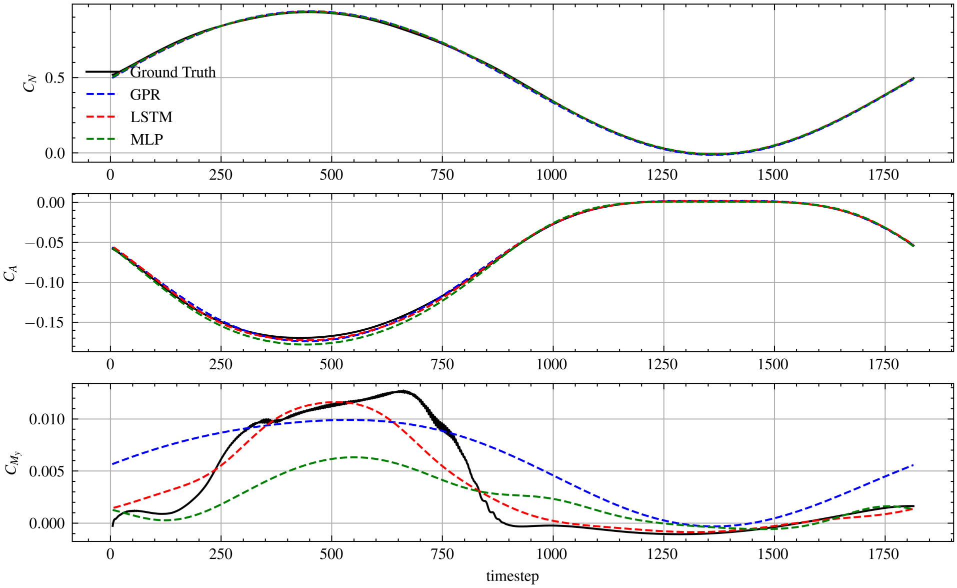


Figure A.16: Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.0125Hz$

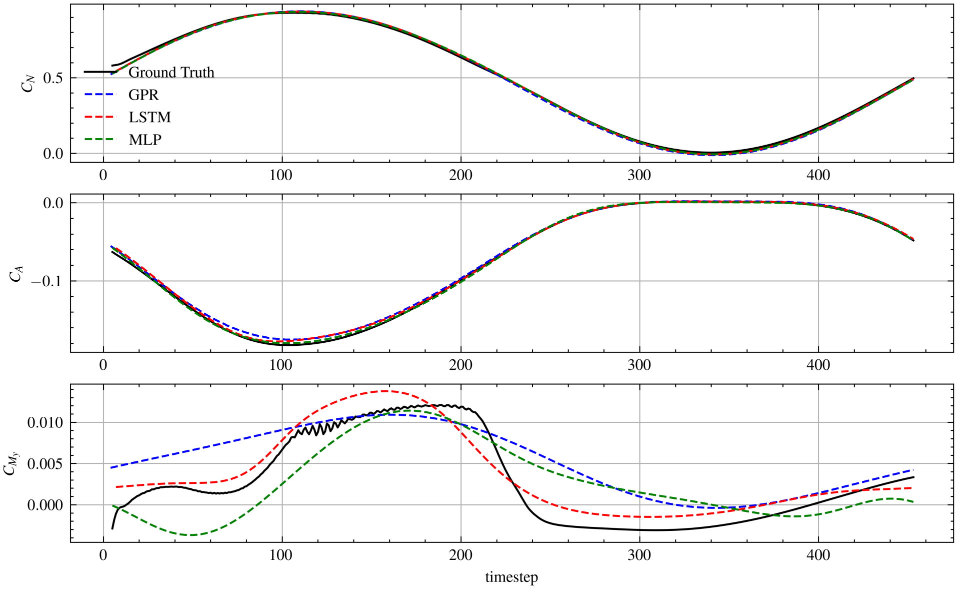


Figure A.17: Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.5Hz$

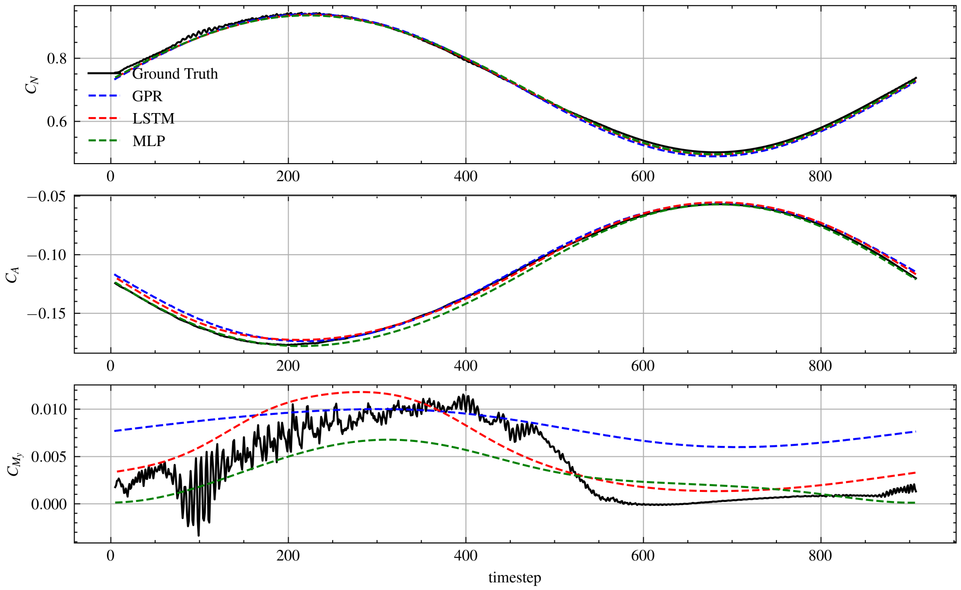


Figure A.18: Integral Loads coefficient prediction using the three different surrogate models. Plunge harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.25Hz$

A.3. PROJECTION ERRORS

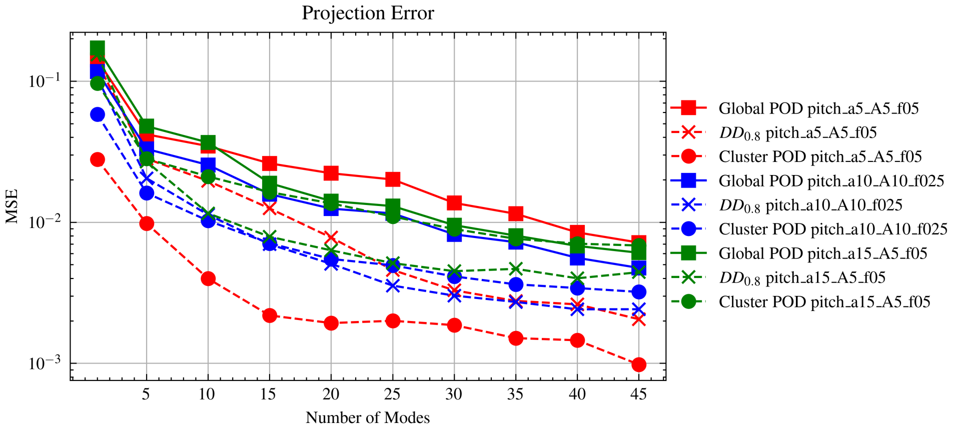


Figure A.19: Time averaged Mean Squared Projection Error vs number of modes over harmonic plunging maneuvers for the Global POD, $DD_{0.8}$-POD and Cluster POD bases.

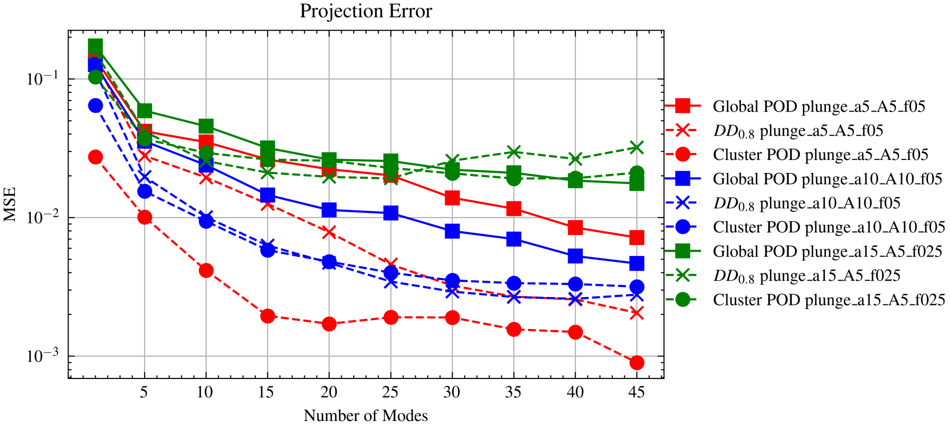


Figure A.20: Time averaged Mean Squared Projection Error vs number of modes over harmonic plunging maneuvers for the Global POD, $DD_{0.8}$-POD and Cluster POD bases.

A.4. Total ROM Errors

A.4.1. 5 Modes Reconstruction

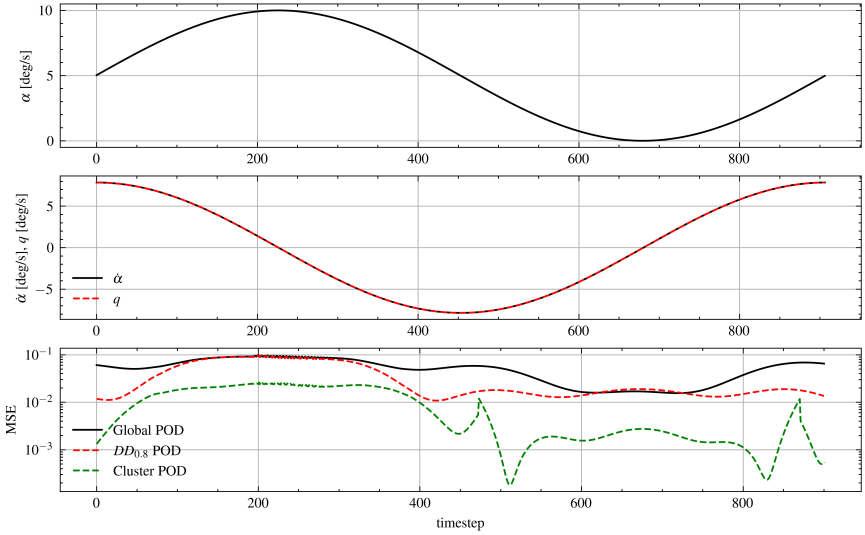


Figure A.21: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 5

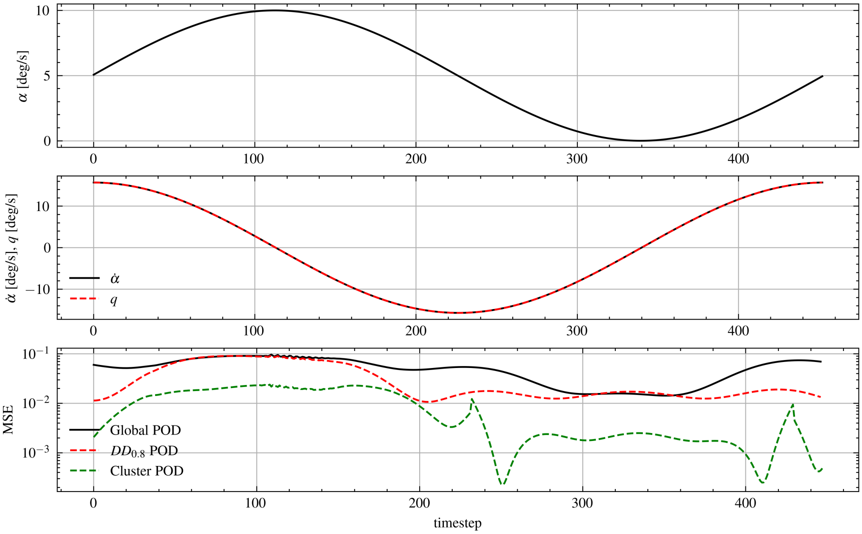


Figure A.22: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.5Hz$.Number of modes: 5

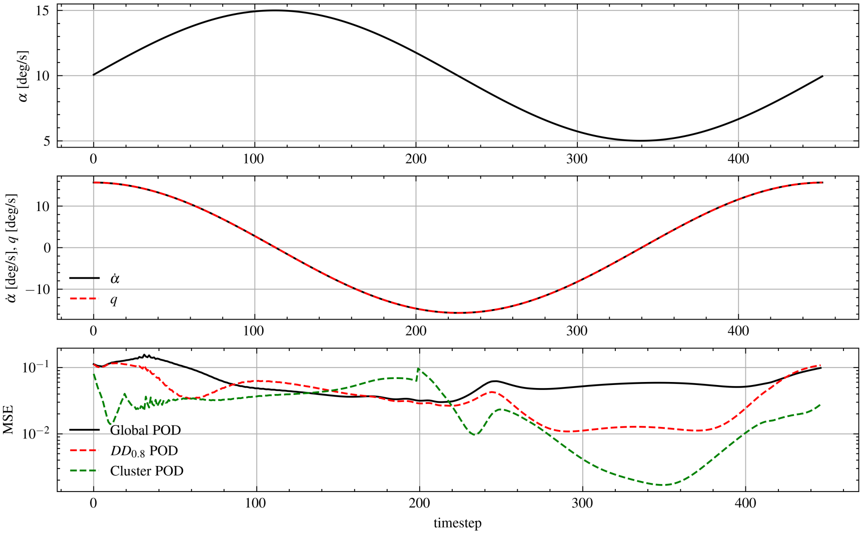


Figure A.23: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.5Hz$.Number of modes: 5

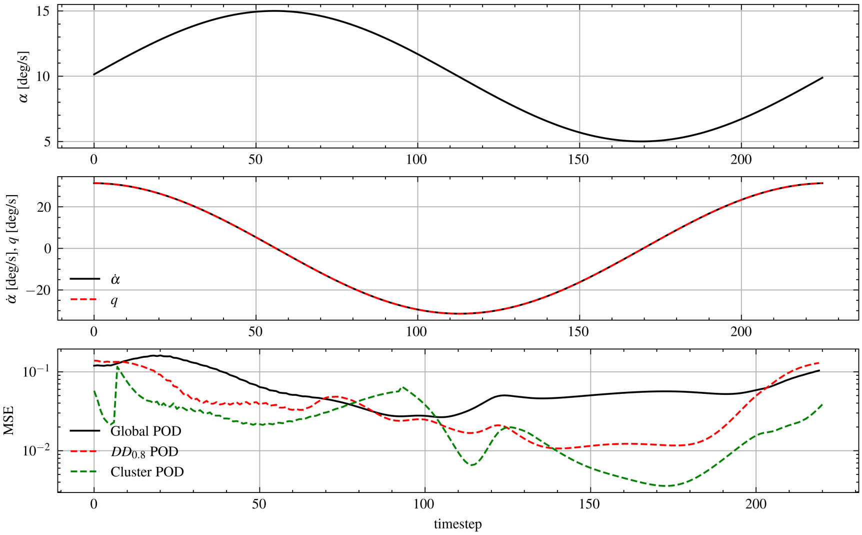


Figure A.24: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 1Hz$.Number of modes: 5

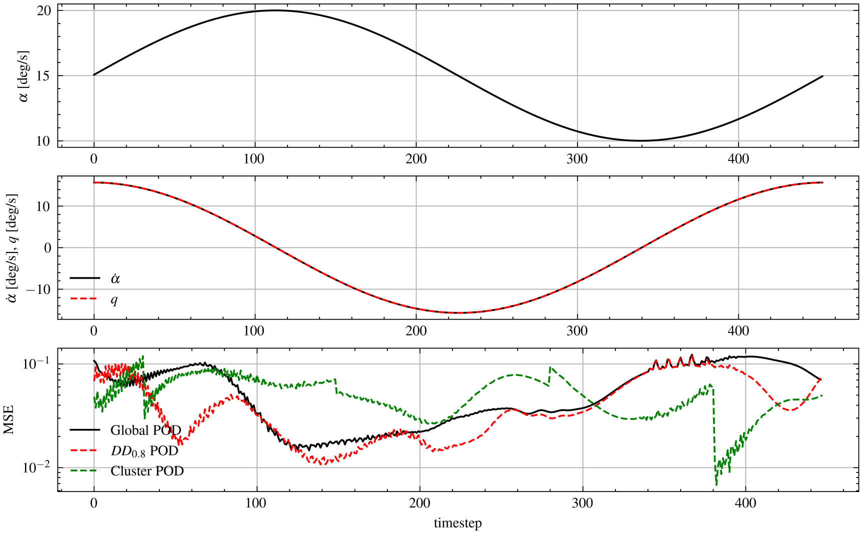


Figure A.25: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.5Hz$.Number of modes: 5

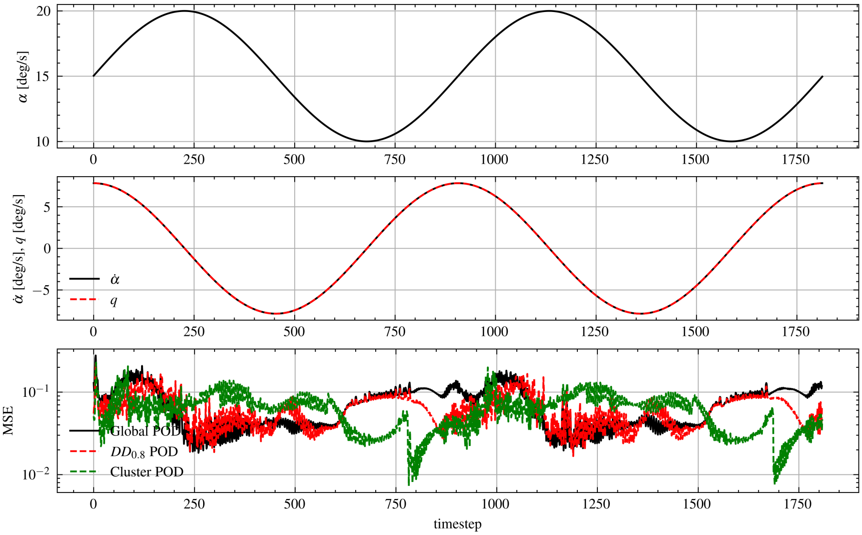


Figure A.26: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 5

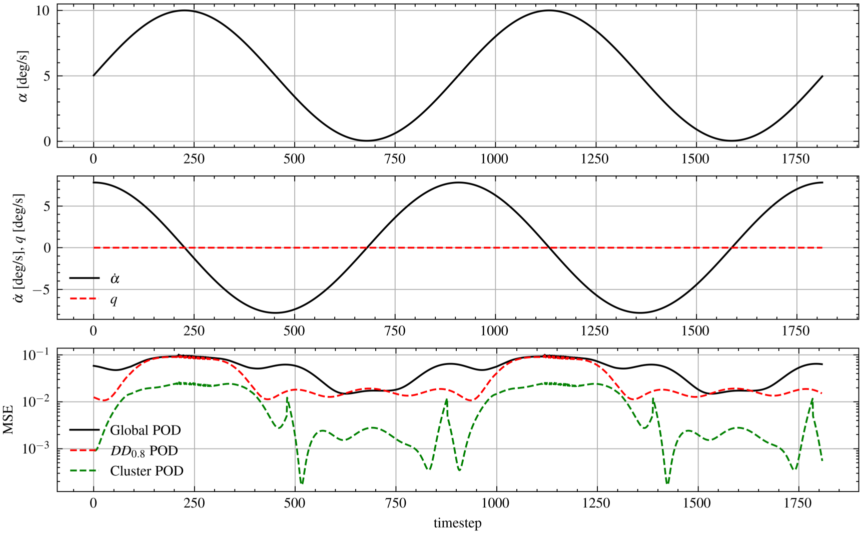


Figure A.27: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 5

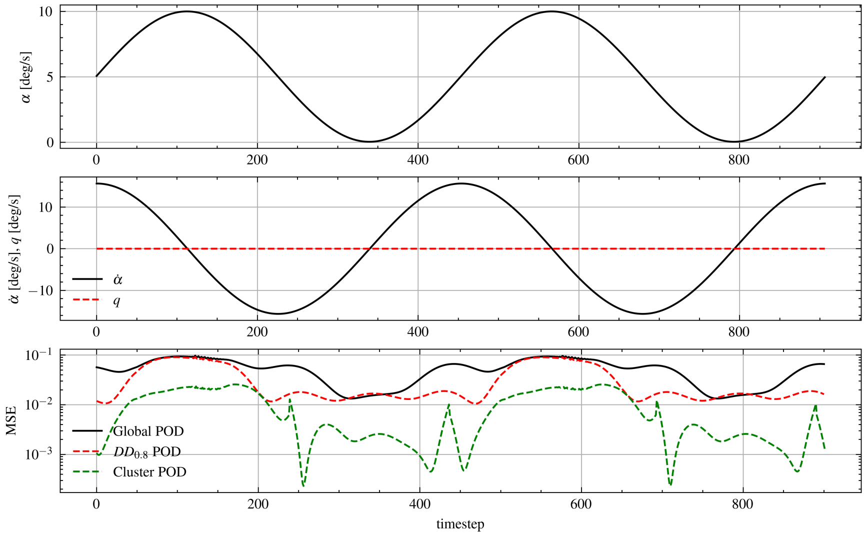


Figure A.28: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.5Hz$.Number of modes: 5

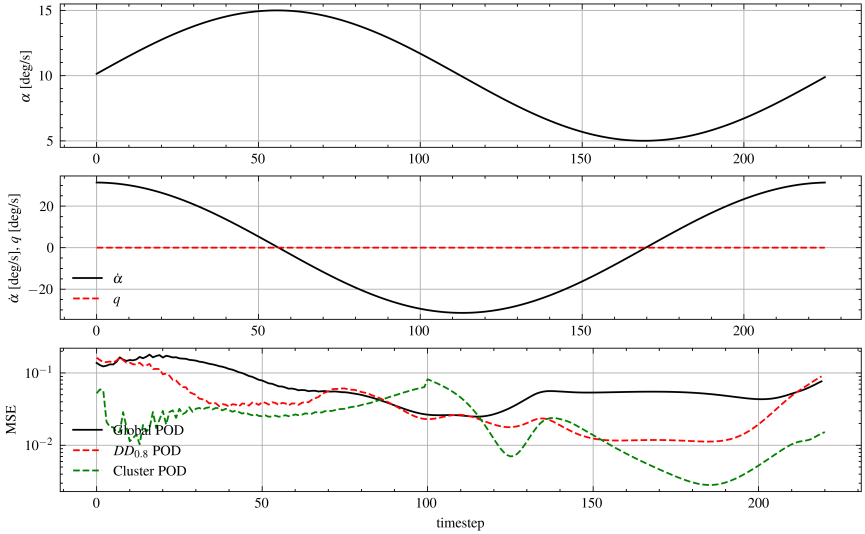


Figure A.29: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 1Hz$.Number of modes: 5

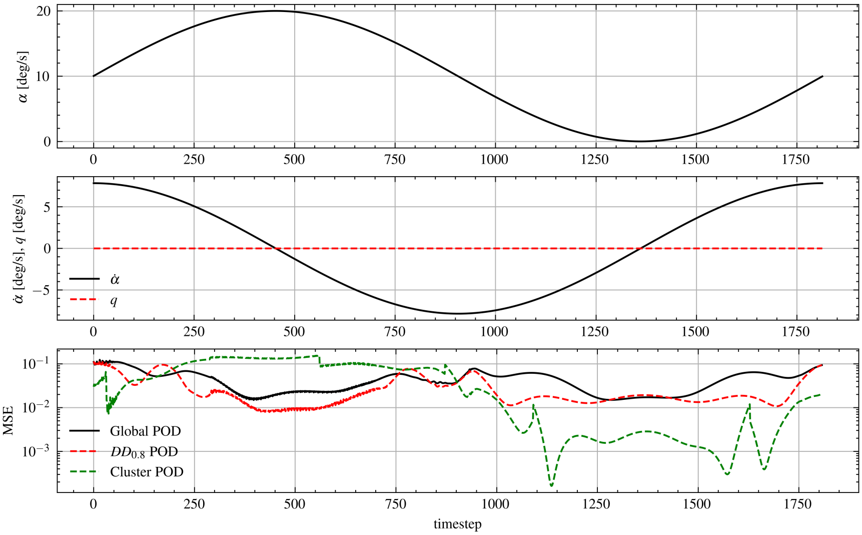


Figure A.30: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.125Hz$.Number of modes: 5

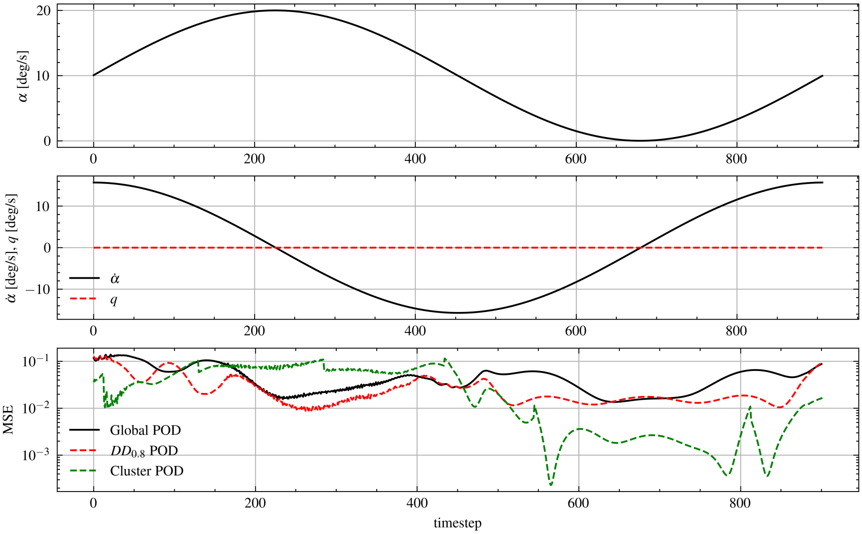


Figure A.31: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.25Hz$.Number of modes: 5

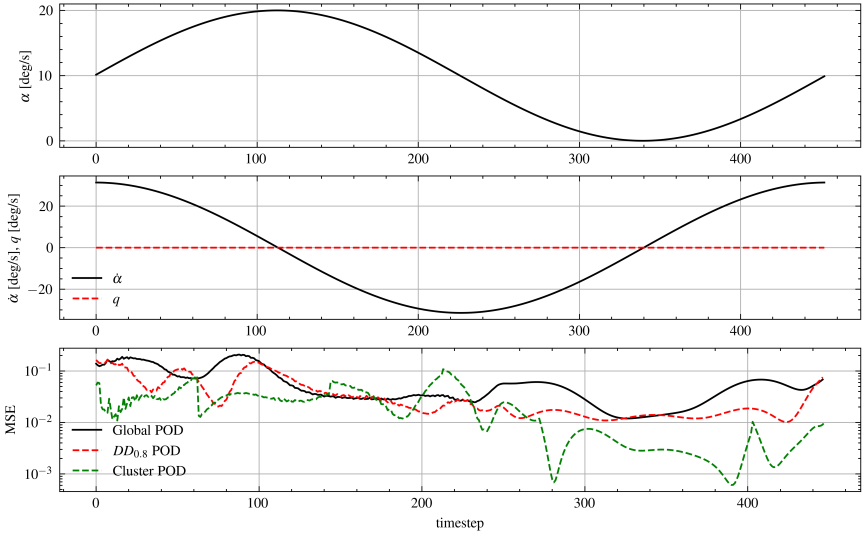


Figure A.32: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.5Hz$.Number of modes: 5

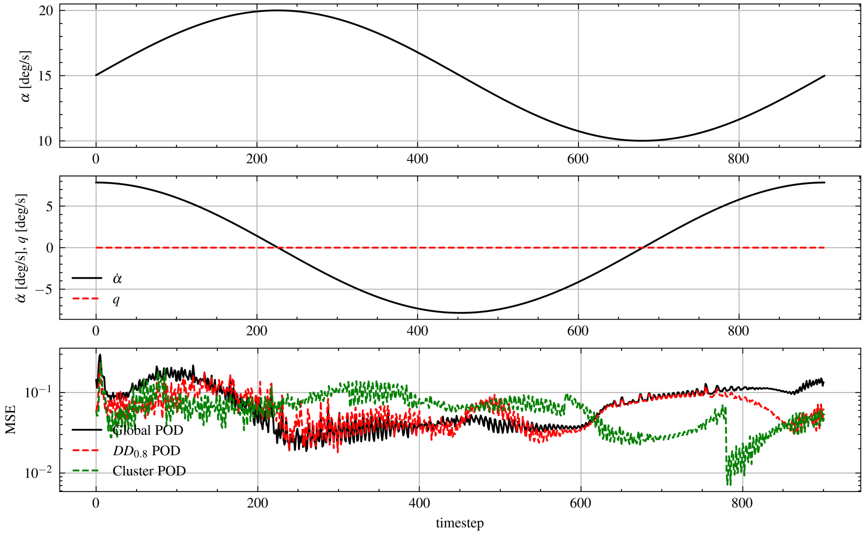


Figure A.33: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 5

A.4.2. 10 MODES RECONSTRUCTION

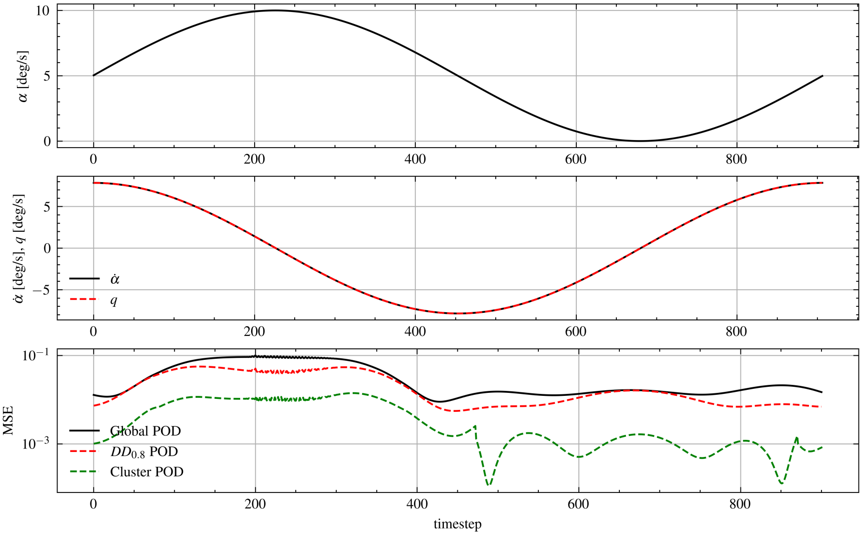


Figure A.34: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 10

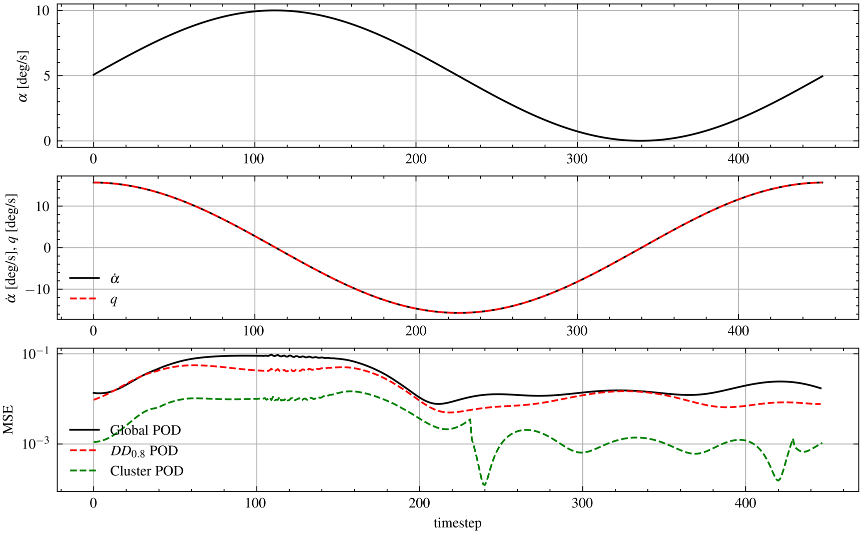


Figure A.35: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.5Hz$.Number of modes: 10

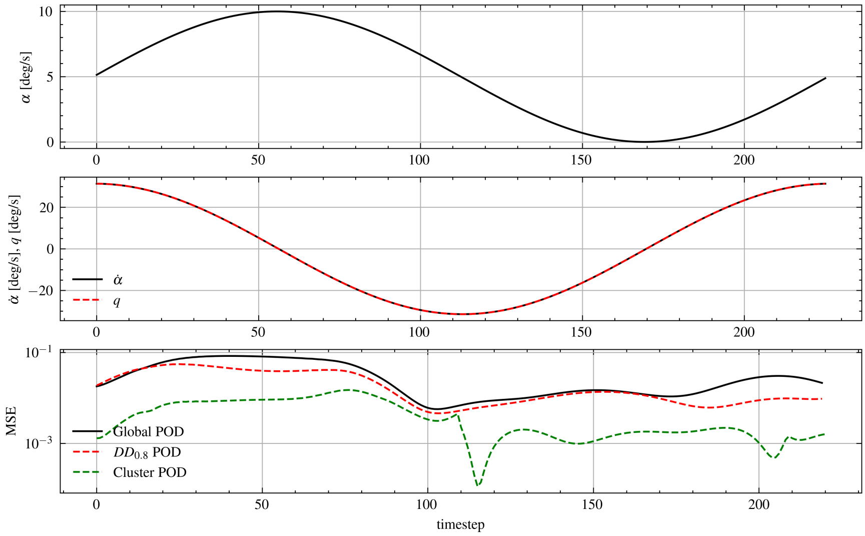


Figure A.36: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 1Hz$.Number of modes: 10

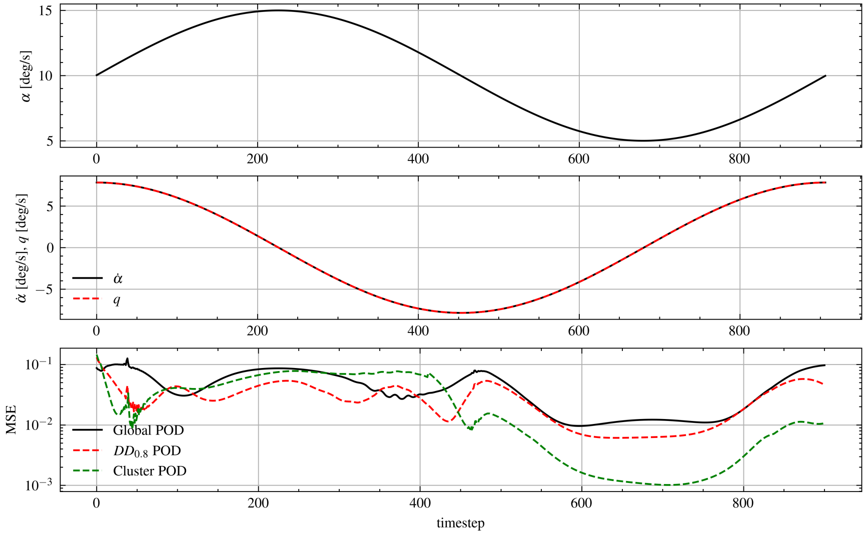


Figure A.37: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 10

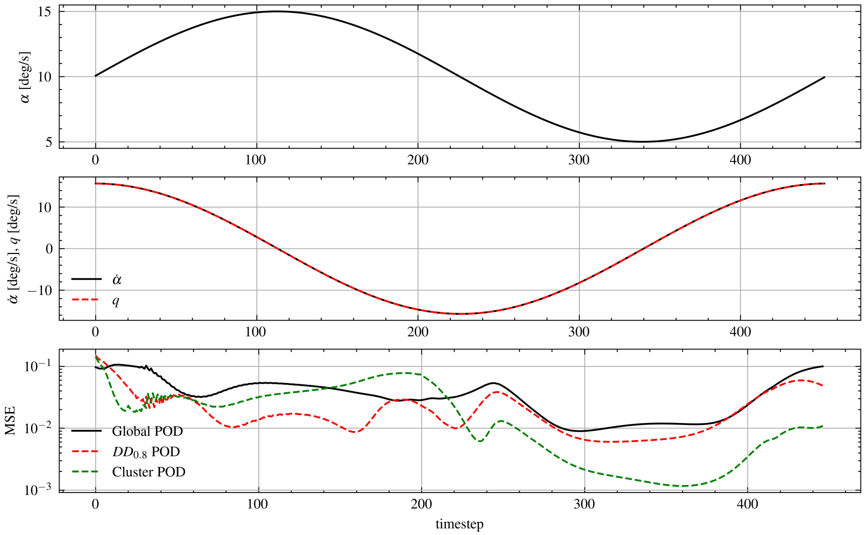


Figure A.38: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.5Hz$.Number of modes: 10

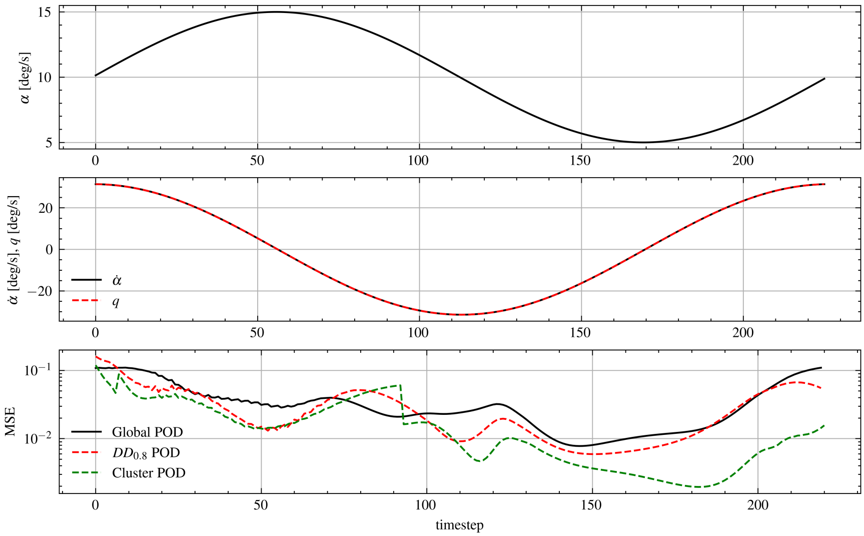


Figure A.39: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 1Hz$.Number of modes: 10

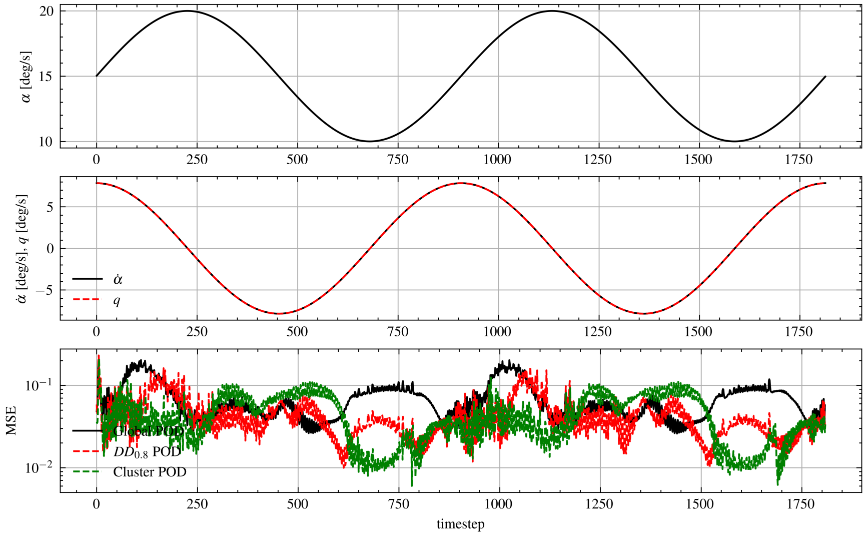


Figure A.40: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 10

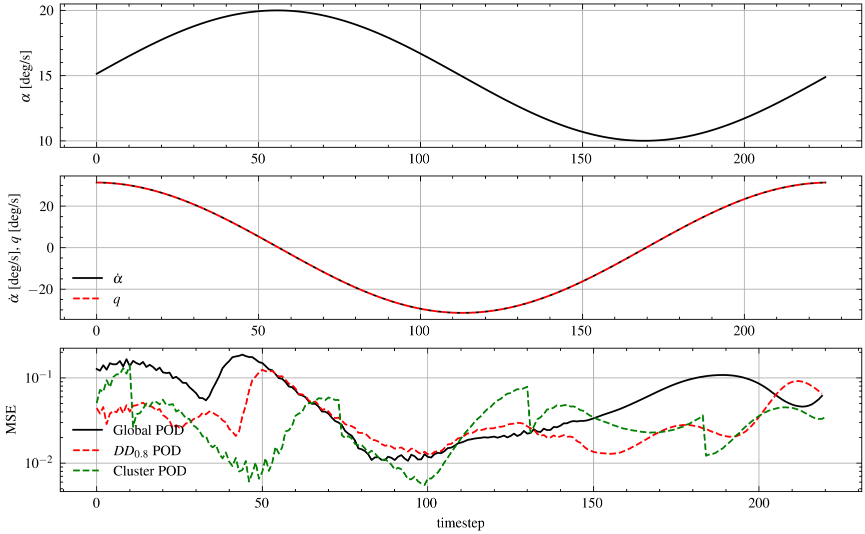


Figure A.41: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Pitch harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 1Hz$.Number of modes: 10

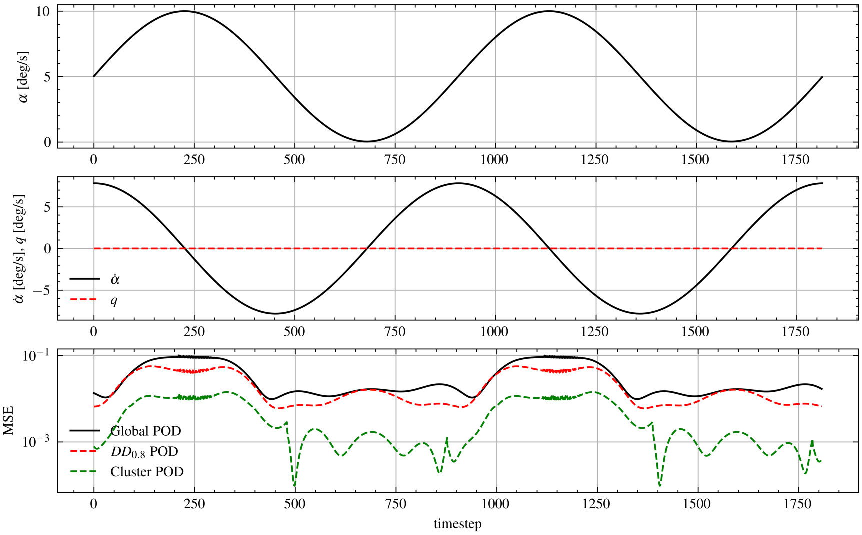


Figure A.42: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 10

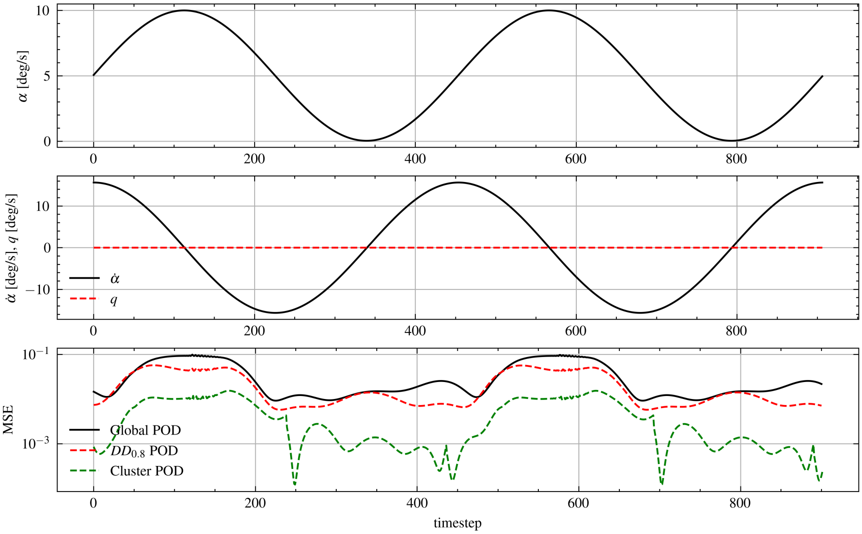


Figure A.43: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 5[deg]$, $A = 5[deg]$, $f = 0.5Hz$.Number of modes: 10

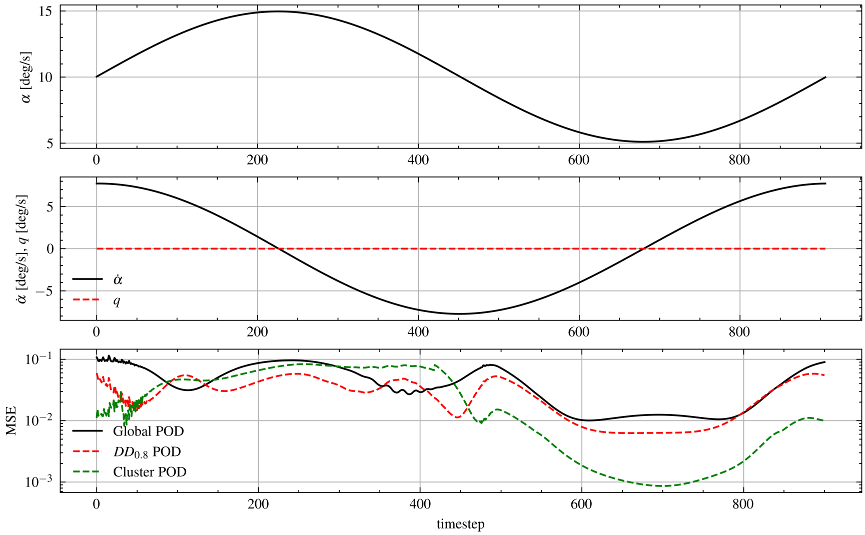


Figure A.44: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 10

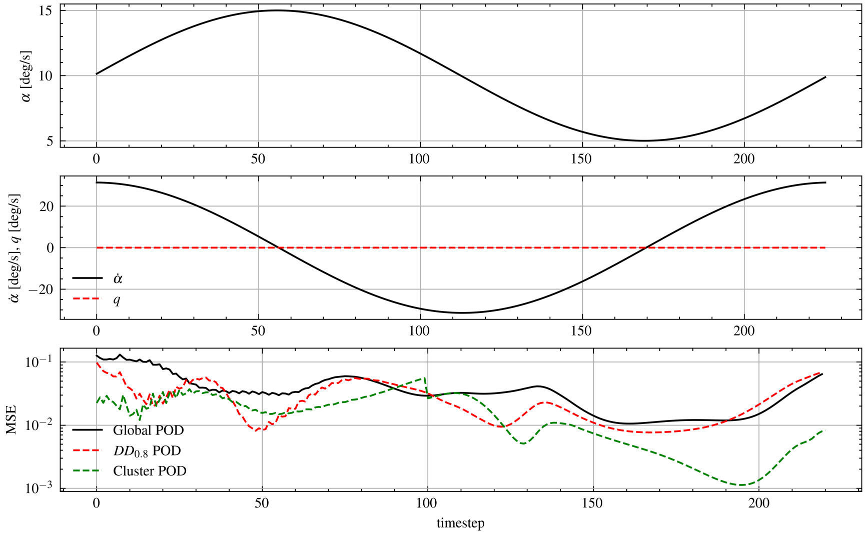


Figure A.45: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 5[deg]$, $f = 1Hz$.Number of modes: 10

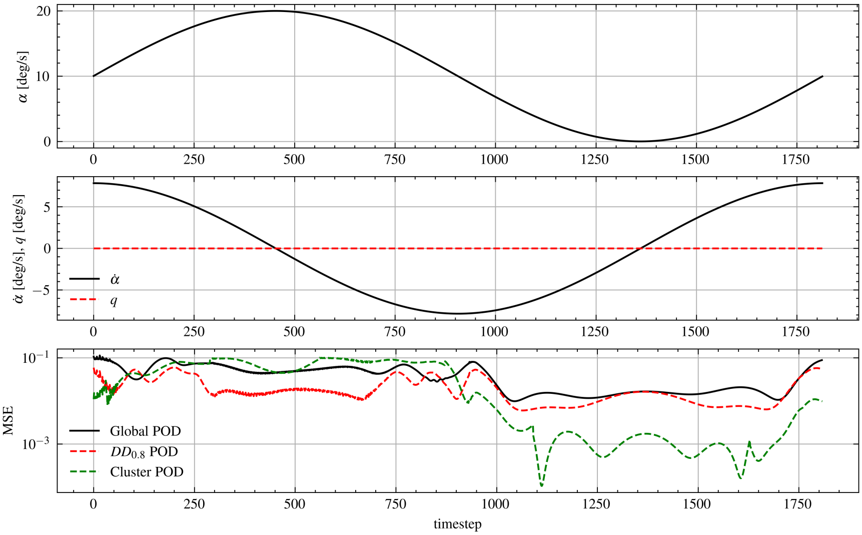


Figure A.46: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.125Hz$.Number of modes: 10

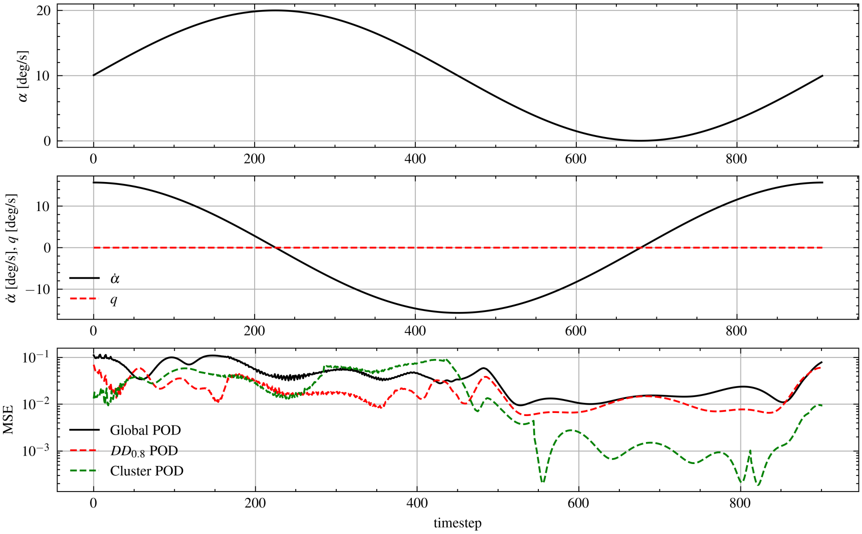


Figure A.47: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.25Hz$.Number of modes: 10

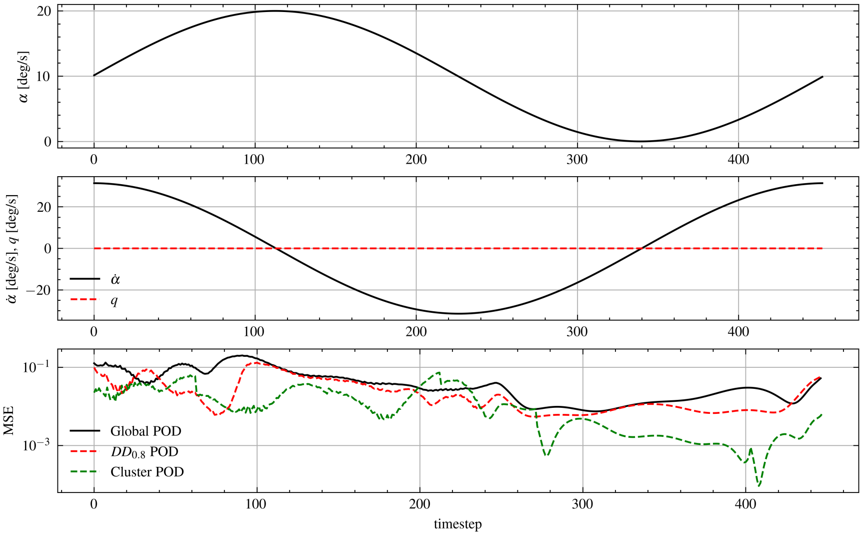


Figure A.48: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 10[deg]$, $A = 10[deg]$, $f = 0.5Hz$.Number of modes: 10

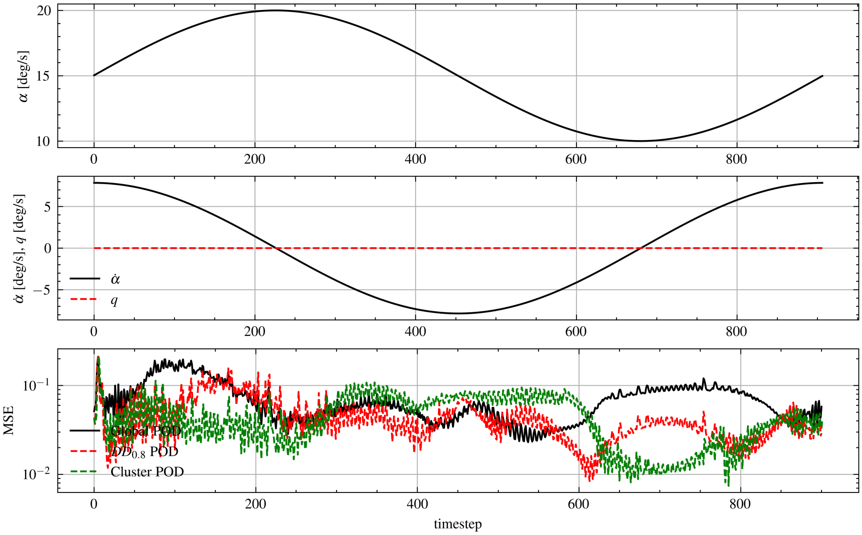


Figure A.49: Instantaneous inputs and Mean Squared Total error using Global POD-LSTM, $DD_{0.8}$-POD-LSTM, Cluster POD-LSTM ROMs. Plunge harmonic maneuver with $A_0 = 15[deg]$, $A = 5[deg]$, $f = 0.25Hz$.Number of modes: 5

Bibliography

[1] Alfio Quarteroni and Gianluigi Rozza. *Reduced Order Methods for Modeling and Computational Reduction*. Ed. by Alfio Quarteroni and Gianluigi Rozza. Springer International Publishing, 2014. DOI: 10.1007/978-3-319-02090-7.

[2] Christophe Andrieu et al. “An introduction to MCMC for machine learning”. In: *Machine Learning* 50.1-2 (Jan. 2003), pp. 5–43. ISSN: 08856125. DOI: 10.1023/A:1020281327116.

[3] Joan Baiges, Ramon Codina, and Sergio Idelsohn. “A domain decomposition strategy for reduced order models. Application to the incompressible Navier-Stokes equations”. In: *Computer Methods in Applied Mechanics and Engineering* 267 (Dec. 2013), pp. 23–42. ISSN: 00457825. DOI: 10.1016/j.cma.2013.08.001.

[4] Maciej Balajewicz, Fred Nitzsche, and Daniel Feszty. “Application of multi-input volterra theory to nonlinear multi-degree-of-freedom aerodynamic systems”. In: *AIAA Journal* 48.1 (Jan. 2010), pp. 56–62. ISSN: 00011452. DOI: 10.2514/1.38964.

[5] Maciej J. Balajewicz, Earl H. Dowell, and Bernd R. Noack. “Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier-Stokes equation”. In: *Journal of Fluid Mechanics* 729 (Aug. 2013), pp. 285–308. ISSN: 00221120. DOI: 10.1017/jfm.2013.278.

[6] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. “Learning Long-Term Dependencies with Gradient Descent is Difficult”. In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166. ISSN: 19410093. DOI: 10.1109/72.279181.

[7] Gal Berkooz, Philip Holmes, and John L. Lumley. “The proper orthogonal decomposition in the analysis of turbulent flows”. In: *Annual Review of Fluid Mechanics* 25.1 (1993), pp. 539–575. ISSN: 00664189. DOI: 10.1146/annurev.fl.25.010193.002543.

[8] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[9] S J L B Bourier. *Development of a CFD data-driven surrogate model using the neural network approach for prediction of aircraft performance characteristics*. Tech. rep. TU Delft, 2021. URL: http://repository.tudelft.nl/..

[10] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. “Machine Learning for Fluid Mechanics”. In: *Annu. Rev. Fluid Mech. 2020* 52 (2019), pp. 477–508. DOI: 10.1146/annurev-fluid-010719. URL: https://doi.org/10.1146/annurev-fluid-010719-.

[11] T Bui-Thanh et al. “Goal-Oriented, Model-Constrained Optimization for Reduction of Large-Scale Systems”. In: *Journal of Computational Physics* Preprint (2006).

[12] Lei Cheng. "A semi-continuous formulation for goal-oriented reduced-order models". English. PhD thesis. Delft University of Technology, 2017. DOI: 10.4233/uuid:861b26cf-499d-4038-af2e-a29641f84ada.

[13] Francois Chollet et al. *Keras*. 2015. URL: https://github.com/fchollet/keras.

[14] A Da Ronch et al. *Model Reduction for Linear and Nonlinear Gust Loads Analysis*. Tech. rep.

[15] Geoffrey E. Hinton Ronald J. Williams David E. Rumelhart. "Learning representations by back-propagating errors". In: *Nature* 323 (1986), pp. 533–536.

[16] Romain Dupuis, Jean-Christophe Jouhaud, and Pierre Sagaut. "Surrogate Modeling of Aerodynamic Simulations for Multiple Operating Conditions Using Machine Learning". In: *Arxiv* (2019).

[17] Cihat Duru, Hande Alemdar, and Ozgur Ugras Baran. "A deep learning approach for the transonic flow field predictions around airfoils". In: *Computers and Fluids* 236 (Mar. 2022). ISSN: 00457930. DOI: 10.1016/J.COMPFLUID.2022.105312.

[18] Clement Farabet et al. "Learning hierarchical features for scene labeling". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1915–1929. ISSN: 01628828. DOI: 10.1109/TPAMI.2012.231.

[19] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *33rd International Conference on Machine Learning, ICML 2016* 3 (June 2015), pp. 1651–1660. DOI: 10.48550/arxiv.1506.02142. URL: https://arxiv.org/abs/1506.02142v6.

[20] Jacob R Gardner et al. "GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration". In: *Advances in Neural Information Processing Systems*. 2018.

[21] I E Garrick and Wilmer H Reed. "Historical Development of Aircraft Flutter". In: *Journal of Aircraft* 18.11 (1981), pp. 897–912. DOI: 10.2514/3.57579. URL: https://doi.org/10.2514/3.57579.

[22] Mehdi Ghoreyshi and Russell M. Cummings. "Unsteady aerodynamics modeling for aircraft maneuvers: A new approach using time-dependent surrogate modeling". In: *Aerospace Science and Technology* 39 (2014), pp. 222–242. ISSN: 12709638. DOI: 10.1016/j.ast.2014.09.009.

[23] Mehdi Ghoreyshi et al. "Collaborative evaluation of CFD-to-ROM dynamic modeling". In: *54th AIAA Aerospace Sciences Meeting*. Vol. 0. American Institute of Aeronautics and Astronautics Inc, AIAA, 2016. ISBN: 9781624103933. DOI: 10.2514/6.2016-1077.

[24] Mehdi Ghoreyshi et al. "Transonic aerodynamic load modeling of X-31 aircraft pitching motions". In: *AIAA Journal* 51.10 (Oct. 2013), pp. 2447–2464. ISSN: 00011452. DOI: 10.2514/1.J052309.

[25] Ian J Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[26] Bernard Haasdonk and Mario Ohlberger. "Reduced basis method for finite volume approximations of parametrized linear evolution equations". In: *Mathematical Modelling and Numerical Analysis* 42.2 (Mar. 2008), pp. 277–302. ISSN: 0764583X. DOI: 10.1051/m2an:2008001.

[27] Martin Hess et al. "A localized reduced-order modeling approach for PDEs with bifurcating solutions". In: *Computer Methods in Applied Mechanics and Engineering* 351 (July 2019), pp. 379–403. ISSN: 00457825. DOI: 10.1016/j.cma.2019.03.050.

[28] Hiller Brett R. "An Usteady Aerodynamics Reduced-Order Modeling Memthod For Maneuvering, Flexible Flight Vehicles". PhD thesis. Georgia Institute of Technology, 2019.

[29] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 08997667. DOI: 10.1162/neco.1997.9.8.1735.

[30] Sepp Hochreiter et al. "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies". In: *A field guide to dynamical recurrent neural networks. IEEE Press* (2001).

[31] Philip Holmes, John L Lumley, and Gal Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge Monographs on Mechanics. Cambridge University Press, 1996. DOI: 10.1017/CBO9780511622700.

[32] Adam Jirásek et al. "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*. American Institute of Aeronautics and Astronautics Inc., 2010. ISBN: 9781600867392. DOI: 10.2514/6.2010-515.

[33] Jer-Nan Juang and Richard S Pappa. "An eigensystem realization algorithm for modal parameter identification and model reduction". In: *Journal of Guidance, Control, and Dynamics* 8.5 (1985), pp. 620–627. DOI: 10.2514/3.20031. URL: https://doi.org/10.2514/3.20031.

[34] Eurika Kaiser et al. "Cluster-based reduced-order modelling of a mixing layer". In: *Journal of Fluid Mechanics* 754 (Sept. 2014), pp. 365–414. ISSN: 0022-1120. DOI: 10.1017/JFM.2014.355.

[35] Diederik P. Kingma and Jimmy Lei Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (Dec. 2014). DOI: 10.48550/arxiv.1412.6980. URL: https://arxiv.org/abs/1412.6980v9.

[36] J C Kok and B B Prananta. "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).

[37] L. Cordier and M. Bergmann. *Proper Orthogonal Decomposition: an overview*. von Karman Institute for Fluid Dynamics, 2003.

[38] Kookjin Lee and Kevin T. Carlberg. “Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders”. In: *Journal of Computational Physics* 404 (Mar. 2020). ISSN: 10902716. DOI: 10.1016/j.jcp.2019.108973.

[39] Xiaodong Li, Steven Hulshoff, and Stefan Hickel. “An enhanced algorithm for online Proper Orthogonal Decomposition and its parallelization for unsteady simulations”. In: *Computers & Mathematics with Applications* 126 (Nov. 2022), pp. 43–59. ISSN: 0898-1221. DOI: 10.1016/J.CAMWA.2022.09.007.

[40] Carsten M Liersch, Russell M Cummings, and Andreas Schütte. *NATO STO/AVT-251: A Joint Exercise in Collaborative Combat Aircraft Design*. Tech. rep.

[41] Carsten M. Liersch et al. “Multi-disciplinary design and performance assessment of effective, agile NATO air vehicles”. In: *Aerospace Science and Technology* 99 (Apr. 2020), p. 105764. ISSN: 1270-9638. DOI: 10.1016/J.AST.2020.105764.

[42] Zachary C. Lipton, John Berkowitz, and Charles Elkan. “A Critical Review of Recurrent Neural Networks for Sequence Learning”. In: *Arxiv* (May 2015). URL: http://arxiv.org/abs/1506.00019.

[43] David J. Lucia, Paul I. King, and Philip S. Beran. “Domain decomposition for reduced-order modeling of a flow with moving shocks”. In: *AIAA Journal* 40.11 (2002), pp. 2360–2362. ISSN: 00011452. DOI: 10.2514/2.1576.

[44] David J. Lucia et al. “Reduced order modeling for a one-dimensional nozzle flow with moving shocks”. In: *15th AIAA Computational Fluid Dynamics Conference*. 2001. DOI: 10.2514/6.2001-2602.

[45] Yongxi Lyu et al. “Unsteady aerodynamic modeling of biaxial coupled oscillation based on improved ELM”. In: *Aerospace Science and Technology* 60 (2017), pp. 58–67. ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2016.10.029.

[46] Romit Maulik et al. “Latent-space time evolution of non-intrusive reduced-order models using Gaussian process emulation”. In: *Physica D: Nonlinear Phenomena* 416 (Feb. 2021). ISSN: 01672789. DOI: 10.1016/j.physd.2020.132797.

[47] F R Menter. “Two-equation eddy-viscosity turbulence models for engineering applications”. In: *AIAA Journal* 32.8 (1994), pp. 1598–1605. DOI: 10.2514/3.12149. URL: https://doi.org/10.2514/3.12149.

[48] Eugene A Morelli. “Paper REG-360 Multiple Input Design for Real-Time Parameter Estimation in the Frequency Domain”. In: *13fh WAC Conference on System Identification*. 2003.

[49] Raj Nangia et al. “Aerodynamic design assessment and comparisons of the MULDICON UCAV concept”. In: *Aerospace Science and Technology* 93 (Oct. 2019), p. 105321. ISSN: 1270-9638. DOI: 10.1016/J.AST.2019.105321.

[50] Bernd R. Noack et al. “Recursive dynamic mode decomposition of transient and post-transient wake flows”. In: *Journal of Fluid Mechanics* 809 (Dec. 2016), pp. 843–872. ISSN: 14697645. DOI: 10.1017/jfm.2016.678.

[51] Dávid Papp. *Prediction of unsteady nonlinear aerodynamic loads using deep convolutional neural networks Investigating the dynamic response of agile combat aircraft*. Tech. rep. TU Delft, 2018. URL: http://repository.tudelft.nl/..

[52] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[53] A. Paul-Dubois-Taine and D. Amsallem. "An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models". In: *International Journal for Numerical Methods in Engineering* 102.5 (May 2015), pp. 1262–1292. ISSN: 10970207. DOI: 10.1002/nme.4759.

[54] Carl Edward. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006, p. 248. ISBN: 026218253X.

[55] D. E. Raveh. "Reduced-order models for nonlinear unsteady aerodynamics". In: *AIAA Journal* 39.8 (2001), pp. 1417–1429. ISSN: 00011452. DOI: 10.2514/2.1473.

[56] A. da Ronch et al. "Nonlinear model reduction for flexible aircraft control design". In: *AIAA Atmospheric Flight Mechanics Conference 2012*. 2012. ISBN: 9781624101847. DOI: 10.2514/6.2012-4404.

[57] Clarence W. Rowley, Tim Colonius, and Richard M. Murray. "Model reduction for compressible flows using POD and Galerkin projection". In: *Physica D: Nonlinear Phenomena* 189.1-2 (Feb. 2004), pp. 115–129. ISSN: 01672789. DOI: 10.1016/j.physd.2003.03.001.

[58] Clarence W. Rowley and Scott T.M. Dawson. *Model Reduction for Flow Analysis and Control*. Jan. 2017. DOI: 10.1146/annurev-fluid-010816-060042.

[59] P. A. Srinivasan et al. "Predictions of turbulent shear flows using deep neural networks". In: *Physical Review Fluids* 4.5 (May 2019). ISSN: 2469990X. DOI: 10.48550/arxiv.1905.03634. URL: https://arxiv.org/abs/1905.03634v1.

[60] Theodorsen T. *General Theory Of Aerodynamic Instability and the Mechanism of Flutter*. Tech. rep. Langley Memorial Aeronautical Cneter, 1949.

[61] Murray Tobak, Gary T Chapman, and Lewis B Schiff. *Mathematical Modeling of the Aerodynamic Characteristics In Flight Dynamics*. Tech. rep. NASA, 1984.

[62] Ricardo Vinuesa and Steven L. Brunton. "The Potential of Machine Learning to Enhance Computational Fluid Dynamics". In: *Arxiv* (Oct. 2021). URL: http://arxiv.org/abs/2110.02085.

[63] Stefan Volkwein. "Model Reduction Using Proper Orthogonal Decomposition". In: *Karl-Franzens-Universitat Graz Institut fur Mathematik und Wissenschaftliches Rechnen* (Mar. 2008).

[64] Herbert Wagner. "Über die Entstehung des dynamischen Auftriebes von Tragflügeln". In: *Mathematica* (1925).

[65] Mark van der Wilk et al. “A Framework for Interdomain and Multioutput Gaussian Processes”. In: *arXiv:2003.01115* (2020). URL: https://arxiv.org/abs/2003.01115.

[66] R. Zahn et al. “Airfoil buffet aerodynamics at plunge and pitch excitation based on long short-term memory neural network prediction”. In: *CEAS Aeronautical Journal 2021, Vol. , Pages: 1–11* 13.1 (Oct. 2021), pp. 1–11. ISSN: 18695590. DOI: 10.1007/S13272-021-00550-6. URL: https://doi.org/10.1007/s13272-021-00550-6.