DLR-IB-AS-BS-2022-111

Outcomes of the DLR-Airbus Patenschaft “Development of Future Aerodynamic Data Modelling Methodology” 2018–2022

Forschungsbericht

Autor
Dr. Anna Bertram

Deutsches Zentrum
DLR für Luft- und Raumfahrt

Bericht des Instituts für Aerodynamik und Strömungstechnik
Report of the Institute of Aerodynamics and Flow Technology

DLR-IB-AS-BS-2022-111

Outcomes of the DLR-Airbus Patenschaft "Development of Future Aerodynamic Data Modelling Methodology" 2018–2022

Anna Bertram

Herausgeber:
Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Aerodynamik und Strömungstechnik
Lilienthalplatz 7, 38108 Braunschweig

ISSN 1614-7790

Stufe der Zugänglichkeit: 1
Braunschweig, im August 2022

Institutsdirektor:
Prof. Dr.-Ing. habil. Cord-Christian Rossow

Verfasser:
Dr. Anna Bertram

Abteilung: $C^2A^2S^2E$
Abteilungsleiter: Prof. Dr. Stefan Görtz

Der Bericht enthält:
59 Seiten
16 Bilder
3 Tabellen
45 Literaturstellen

Abstract

From 2018 until 2022, Airbus and DLR have been collaborating in the field of aerodynamic data fusion in the framework of a so-called *Patenschaft*, a jointly funded research position dedicated to the topic "Development of Future Aerodynamic Data Modelling Methodology". The goal of this cooperation was the development, improvement and industrialization of methods for the modeling of high-dimensional aerodynamic data for realistic aircraft configurations for all relevant flight conditions – a topic which is of high interest to both partners: Airbus can significantly accelerate the transfer of new approaches into the industrial context. For DLR, the application-oriented investigations and the research has the potential to provide important impulses for the work on the virtual data model of the aircraft pursued in the guiding concept "Virtual Product". The main achievements and outcomes of this Patenschaft are collected in this final report.

Contents

1 Introduction

From 2018 until 2022, Airbus and DLR have been collaborating in the field of aerodynamic data fusion in the framework of a so-called *Patenschaft*, a jointly funded research position dedicated to the topic "Development of Future Aerodynamic Data Modelling Methodology". In particular, the goals of this cooperation were

- the investigation, definition and development of methods for the robust prediction of aerodynamic data of full aircraft configurations across the flight envelope, for extreme flight conditions and off-design cases;
- the development of data modeling solutions for configurations with movables, e.g. high lift devices and spoiler deployment;
- the investigation of methods for the analysis and fusion of data from different sources;
- the application of methods in the industrial context as well as the development of necessary tool chains;
- the development of further alternative methods and the definition of a future integrated process for data creation and delivery.

The topic is of high interest to both partners: Airbus can significantly accelerate the transfer of new approaches into the industrial context. For DLR, the application-oriented investigations and the research has the potential to provide important impulses for the work on the virtual data model of the aircraft pursued in the guiding concept "Virtual Product".

The main achievements and outcomes of this Patenschaft are collected in this final report. Some parts of this work, as well as large sections of the following motivation and literature overview, have been published before in [6].

1.1 Motivation and Literature Overview

During the development of an aircraft, a wealth of aerodynamic data is required for different flight conditions throughout the flight envelope. Quantities like the scalar-valued

lift coefficient or the pressure distribution at the surface are, for instance, significant for structural and geometrical design, performance and loads evaluations and the design of the flight control system. In the last decades, this data has been increasingly acquired through Computational Fluid Dynamics (CFD) simulations [10]. However, due to the complexity of the problem, a direct numerical simulation for industrial aircraft configurations is not even feasible for high-performance computers in the foreseeable future. Simplifying assumptions are made to ease the problem, but CFD simulations remain computationally demanding. In addition, due to convergence issues especially for extreme flight conditions, numerical data cannot be reliably provided throughout the entire flight envelope. CFD data is therefore complemented by data from wind tunnel experiments and flight testing. Because of errors mainly introduced by the physical modeling and the discretization of the problem on the one hand and experimental limitations on the other hand, the data from these different sources will, however, always show some differences to deal with. The heterogeneity in the data becomes even more complex when, instead of scalar-valued quantities, field data has to be considered—as often required by industry: For example, for the structural design of an aircraft, one is interested in reliably predicting the location of aerodynamic shocks as indicated by the surface pressure distribution. While CFD provides this quantity in each cell on the surface of the discretized aircraft, it is typically only available at comparatively few sensor locations in wind tunnel and flight tests which results in a high discrepancy in the dimensions of computational and experimental data.

Data fusion techniques aim at combining the individual strengths of different data sources to provide consistent and reliable data sets. For scalar-valued quantities of interest, a popular data fusion strategy is the use of variable-fidelity surrogate models. A base assumption of these models is that the different data sources have different levels of accuracy. The simplest way to fuse such variable-fidelity data is by employing so-called bridge functions: A surrogate model for the low-fidelity data is used to approximate the high-fidelity data via an additive, multiplicative or hybrid correction, [26]. Another example of data fusion with bridge function is given in [30] where data from two computer codes of different fidelity were combined via a Kriging-based bridge function. Kriging, also known as Gaussian process regression, is based on the assumption that the given data points are realizations of correlated random variables. It can handle highly nonlinear responses and features fast evaluation times which makes it a popular method in various different fields such as design and analysis of computer experiments [39, 40], machine learning [37] and surrogate modeling [21]. Direct extension of Gaussian process regression to the variable-fidelity framework are known as Cokriging [21, p. 177], [31, 28, 27, 8], and Hier-

archical Kriging, [25]. Both approaches have been used in the context of aerodynamic applications for variable-fidelity modeling based on different sources of computational data, [18, 20, 21, 19, 25, 27], and for experimental and numerical data, [32]. A non-hierarchical approach for the fusion of scalar-valued quantities based on Gaussian process regression was recently presented in [17]: A Gaussian process model is built for every data source individually. Afterwards, these models are combined via a weighted sum based on modeling uncertainty and expert knowledge on the model fidelity.

For the surrogate modeling of vector-valued quantities based on data of different levels of fidelity, the variable-fidelity methodology was extended in [5, 7]. The authors considered aerodynamic data from computer simulations of different accuracy on the same computational grid in order to construct a surrogate model for the expensive, high-fidelity computer code. A common orthonormal basis was computed employing the dimensionality reduction technique Proper Orthogonal Decomposition (POD), [36, 4]. The scalar-valued basis coefficients of this POD basis are then interpolated via a variable-fidelity surrogate model like Cokriging or Hierarchical Kriging. Another data fusion approach for vector-valued quantities was recently proposed in [38], where in a Bayesian setting, experimental and numerical data to the same aerodynamic flow conditions and the same spatial grid were combined via a weighted sum. A drawback of these methods is however that low- and high-fidelity data have to be given at the same spatial grid which prevents their application for the fusion of sparse experimental sensor data and high-dimensional numerical data.

The latter can be achieved with the so-called Gappy POD method. Gappy POD combines POD with a least-squares problem to reconstruct not completely known data. The idea is that within the subspace spanned by the POD modes a solution can be found which minimizes the differences to reference data at a few discrete locations in a least-square sense. The method was first developed for the reconstruction of human face images from incomplete data sets [16]. Later, the method was extended to fluid dynamic applications in [13], where it was shown that it can be used to reconstruct missing data in CFD snapshots based on a set of complete CFD snapshots for steady aerodynamic flow around an airfoil. Modifications of the original approach have been proposed in [44] in order to enhance its robustness and effectiveness for the reconstruction of spatio-temporal incomplete flow field data. A comparison of Gappy POD and the statistical interpolation method Kriging for the reconstruction of unsteady aerodynamic flow data was done in [24]. In [45], the method was used for the reconstruction of unsteady flow data and, based on this procedure, a strategy for optimal sensor placement was introduced. To avoid overfitting, especially when dealing with real experimental data, a regularized version of Gappy POD

was introduced in [22]. The authors compared different regularization methods and applied their regularized Gappy POD approach to fuse CFD and experimental surface pressure data from a steady aerodynamic flow around the flap of a transport aircraft. The resulting surface pressure distribution matches the experimental values as close as possible while providing dense information on the whole surface as a CFD analysis would. The approach was further extended such that the fused aerodynamic surface data sums up to the overall integral coefficients as measured by the wind tunnel balance in [34]. Technically this has been achieved by extending the regression to account for an equality constraint. One of the limitations of the aforementioned approach stems from the assumption of linearity within the POD model and the consecutive least-squares problem. Furthermore, the acceptance of such data-driven approaches is often limited by the fact that it is hard to estimate the uncertainty in the predicted data. Recently, an alternative approach to the Gappy POD methods was proposed in [15]. The idea is to directly map the sensor measurements to the full field solution via a shallow neural network. In a case study with CFD data, the shallow neural network approach was able to outperform ordinary and regularized Gappy POD. Up to our knowledge, there is no study available in literature in which the method is tested for real data fusion tasks using experimental and numerical data. A problem might be that for data fusion tasks, experimental training data, i.e. pairs of sensor data and corresponding full field solution, are in general not available.

1.2 Contribution of the Patenschaft

In the framework of this Patenschaft, the Gappy POD approach has been further extended in two different ways: First, a weighted Gappy POD was formulated which works by scaling the entries of the Gappy POD residual individually. By applying suitable weights, the impact of each sensor can be controlled and thus the approach allows to include expert knowledge on the accuracy of the individual sensor measurements. Second, a Bayesian Gappy POD was introduced, [6]. The approach is based on the equivalence of the Gappy POD problem of least-squares with a linear regression task for a specific choice of sample data. By taking the regression perspective, the Gappy POD problem can be solved employing Gaussian process regression. The final data fusion result is then given in terms of a probability distribution, which provides valuable information on the predicted modeling accuracy. Furthermore, information on the measurement uncertainty of the experimental data can be taken into account. The limitations of the ordinary approach are overcome by considering nonlinear covariance functions.

Both extensions, as well as the constrained Gappy POD approach introduced in [34], have been implemented into DLR's Surrogate Modeling for AeRodata Toolbox in Python SMARTy, [3] and are available in the latest release. The methods have been tested and compared by means of the XRF1 test case, an Airbus provided industrial standard multi-disciplinary research test case representing a typical configuration for a long range wide body aircraft. The results given in this report can therefore serve as an indication of the applicability of the data fusion approaches for real industrial problems.

Furthermore, as an alternative to the Gappy POD methods, the shallow neural network approach proposed in [15] was investigated for data fusion tasks. A comparative study of this method, Bayesian Gappy POD and regularized Gappy POD was made to assess the potential of the new method.

1.3 Overview of this work

This work is structured as follows. In Chapter 2, the investigated data fusion problem is described mathematically and the notation used throughout this report is introduced. Chapter 3 summarizes the current state-of-the-art Gappy POD models: ordinary, regularized and constrained Gappy POD. The extensions to these approaches developed within the framework of the Patenschaft—weighted Gappy POD and Bayesian Gappy POD—are presented in Chapter 4. In Section 4.3, Bayesian Gappy POD is investigated and compared to regularized Gappy POD by means of the XRF1 test case. Chapter 5 deals with the shallow neural network approach proposed in [15] as an alternative to the Gappy POD methods. The approach is briefly described in Section 5.1. In Section 5.2, the method is investigated theoretically and a comparative study with Gappy POD is given in Section 5.3. Chapter 6 concludes this report and provides an outlook.

2 Problem Setup

Assume that there is a functional dependency between a set of input parameters $\xi \in \mathcal{D} \subset \mathbb{R}^d$ and the resulting high-dimensional quantity of interest, $y\colon \mathcal{D} \to \mathbb{R}^N$. This functional dependency is not explicitly given but can be assessed for given input parameters $\xi^* \in \mathbb{R}^d$ using two different data sources: A primary source, which provides accurate but incomplete information, i.e. the quantity of interest $y(\xi^*)$ can only be observed at $s \ll N$ pairwise distinct components $\{j_1, \ldots, j_s\} \subset \{1, \ldots, N\}$. In addition, a secondary data source $\tilde{y}\colon \mathcal{D} \to \mathbb{R}^N$ is available which describes the same quantity of interest with a lower accuracy. Based on the incomplete information given by the primary data source, we are aiming to reconstruct the full quantity of interest $y(\xi^*)$ with the aid of sampled data from the secondary data source. This setup is very general and may arise in different fields of application. In the special application we consider in this work, the goal is to reconstruct the pressure coefficient distribution at the surface of an aircraft on a highly resolved computational mesh from sparse experimental pressure sensor data using information given by a set of pre-computed CFD solutions. In the subsequent sections, different methods are introduced addressing this problem. For simplicity, we will in the remainder of this work use the common terms related to our target application, i.e. the primary data is referred to as experimental or sensor data and the secondary data is referred to as simulation data. Nevertheless, all methods and approaches are not restricted to this special setting and may transfer to other applications as well.

3 Data Fusion with Gappy POD – State of the Art

In the following, an overview of the current state-of-the-art in data fusion, the Gappy POD methodology, is provided based on the initial work [12] and it's extensions [22] and [34]. Large parts of this overview have been published in our work [6].

3.1 Ordinary Gappy POD

Let $\{\tilde{y}^1 := \tilde{y}(\xi^1), \ldots, \tilde{y}^n := \tilde{y}(\xi^n)\} \subset \mathbb{R}^N$ be a set of sampled data obtained via CFD simulations for different input parameters $\mathcal{P} = \{\xi^1, \ldots, \xi^n\} \subset \mathcal{D}$ and let the corresponding outputs be stored in the *snapshot matrix* $Y \in \mathbb{R}^{N \times n}$,

$$Y := [\tilde{y}^1, \ldots, \tilde{y}^n]. \tag{3.1}$$

For simplicity of the notation and without loss of generality assume that the snapshots are centered with respect to their mean, that is

$$\sum_{i=1}^{n} \tilde{y}^i = 0. \tag{3.2}$$

Performing a singular value decomposition (SVD) [23, Sec. 2.4] of the snapshot matrix yields

$$Y = U \Sigma V^\top, \tag{3.3}$$

where $U = [u^1, \ldots, u^N] \in \mathbb{R}^{N \times N}$ and $V = [v^1, \ldots, v^n] \in \mathbb{R}^{n \times n}$ are orthonormal matrices, i.e. $U^\top U = U U^\top = I_N$ and $V^\top V = V V^\top = I_n$, and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) \in \mathbb{R}^{N \times n}$ contains the singular values $\sigma_1 \geq \ldots, \geq \sigma_n \geq 0$ in descending order. Suppose the rank of the snapshot matrix Y is $r = \mathrm{rank}(Y)$, then only the first $r \leq n$ singular values are non-zero. The corresponding r left singular vectors, which are the first r columns of the matrix U, constitute an orthonormal basis $\{u^1, \ldots, u^r\}$ of the space spanned by the snapshots

$\tilde{y}^1, \ldots, \tilde{y}^n$, the so-called *POD basis*, [23, Corollary 2.4.6].

A key idea of Gappy POD is to interpret a given vector $t \in \mathbb{R}^s$ of experimental data, where $s < N$ is the number of experimental sensors, as a vector $y \in \mathbb{R}^N$ from which only the components $y_{j_1}, \ldots, y_{j_s}$ with $j_1, \ldots, j_s \in \{1, \ldots, N\}$ are known, that is

$$t = \begin{bmatrix} t_1 \\ \vdots \\ t_s \end{bmatrix} = \begin{bmatrix} y_{j_1} \\ \vdots \\ y_{j_s} \end{bmatrix} = P^\mathsf{T} y \tag{3.4}$$

for a mask matrix $P := [e_{j_1}, \ldots, e_{j_s}] \in \mathbb{R}^{N \times s}$. The components of the vector t are identified with the components of the vector y via a nearest neighbor search of the s sensor coordinates in the highly resolved computational grid.

Assuming that the vector y can be approximated in the POD subspace, POD basis coefficients $\hat{a} = (\hat{a}_1, \ldots, \hat{a}_r)^\mathsf{T} \in \mathbb{R}^r$ can be found such that

$$y \approx \hat{y} = \sum_{j=1}^{r} \hat{a}_j u^j = U_r \hat{a}, \tag{3.5}$$

where $U_r = [u^1, \ldots, u^r] \in \mathbb{R}^{N \times r}$ is the matrix of POD basis vectors, i.e. the first r columns of U. The basis coefficient vector $\hat{a} \in \mathbb{R}^r$ which yields the smallest L_2 error regarding the observed entries of the vector y is defined by the least squares problem

$$\hat{a} = \arg\min_a \left\| P^\mathsf{T} U_r a - t \right\|_2^2. \tag{3.6}$$

Usually, $X = P^\mathsf{T} U_r \in \mathbb{R}^{s \times r}$ has full column rank and therefore Eq. (3.6) has a unique solution given by

$$\hat{a} = (X^\mathsf{T} X)^{-1} X^\mathsf{T} t. \tag{3.7}$$

Substituting this basis coefficient vector $\hat{a}$ into Eq. (3.5) yields the ordinary Gappy POD approximation of the vector y. A schematic overview of the ordinary Gappy POD method is shown in Fig. 3.1.

3.2 Regularized Gappy POD

To avoid overfitting, especially when dealing with data from different sources, it is often necessary to complement the least squares problem with regularization terms on the basis coefficients $\hat{a}$, [22, 34]. Shrinkage methods give preference to smaller basis coefficients $\hat{a}$

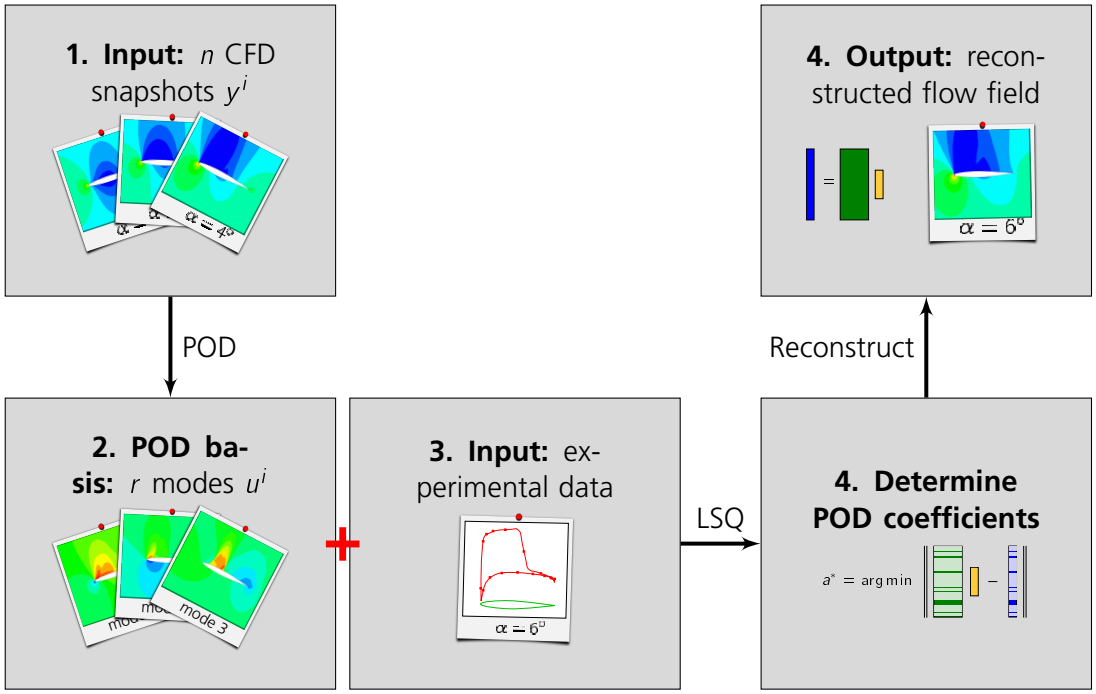


Figure 3.1: Schematic overview of ordinary Gappy POD for fluid flow reconstruction based on sparse sensor data.

by imposing a penalty on their value, cf. [29, Sec. 3]. The influence of less important POD modes, i.e. basis vectors which correspond to small singular values, can thus be restricted by scaling the basis vectors with their corresponding singular values

$$\tilde{U} = [\tilde{u}^1, \ldots, \tilde{u}^r] := [\sigma_1 u^1, \ldots, \sigma_r u^r] \tag{3.8}$$

A popular choice for the regularization of least squares problems is Ridge regression, sometimes also called Tikhonov regularization, [29, Sec. 3.4.1], [43]. It imposes a L_2 penalty on the basis coefficient vector a. The corresponding Gappy POD problem reads,

$$\hat{a}_{\text{rr}} = \arg\min_a \left\| P^\top \tilde{U} a - t \right\|_2^2 + \lambda \|a\|_2^2, \tag{3.9}$$

where $\lambda \in \mathbb{R}_+$ is a parameter which controls the strength of the regularization: the larger the value of λ, the higher the amount of shrinkage towards 0. If $X = P^\top \tilde{U}_r$ has full column rank, the Gappy POD problem has a unique solution,

$$\hat{a}_{\text{rr}} = (X^\top X + \lambda I)^{-1} X^\top t. \tag{3.10}$$

As in the ordinary case, the Gappy POD approximation of the vector y is obtained by evaluating the corresponding linear combination of POD basis modes,

$$\hat{y} = \sum_{j=1}^{r} \hat{a}_j u^j = \tilde{U}_r \hat{a}_{\mathrm{rr}}, \tag{3.11}$$

cf. Eq. (3.5).

3.3 Constrained Gappy POD

Constrained Gappy POD is an additional, straight-forward extension of the ordinary approach and was first introduced in [34, Sec. 2.4]. Suppose, we want the solution a of the Gappy POD problem eq. (3.6) to match a linear constraint of the form

$$\Omega a = c, \tag{3.12}$$

where $\Omega \in \mathbb{R}^{n_c \times r}$, with $n_c \in \mathbb{N}$, is a linear operator and $c \in \mathbb{R}^{n_c}$ is the right hand side. This constraint can be incorporated into the Gappy POD problem eq. (3.6) by adding highly weighted lines to the residual term,

$$\hat{a}_c = \arg\min_a \left\| \begin{bmatrix} P^\mathsf{T} U_r \\ w\Omega \end{bmatrix} a - \begin{bmatrix} t \\ wc \end{bmatrix} \right\|_2^2, \tag{3.13}$$

where the weighting parameter $w \gg 1$ controls the importance of matching the constraint. The regularized Gappy POD problem eq. (3.9) can be extended analogously.

Wind tunnel sensor pressure data often come along with the overall aerodynamic integral coefficients as measured with the wind tunnel main balance. These integral coefficients can be obtained by integrating the surface pressure coefficient and the skin friction coefficient over the surface of the aircraft. In a discrete setting, this integration can be described by a linear operator. Thus, if pressure and skin friction coefficient data is used to compute the POD basis, a linear operator Ω can be found, which maps the POD basis coefficients a to their corresponding integral coefficients.

4 Gappy POD Extensions

In the Patenschaft, several extensions of the ordinary Gappy POD approach have been developed. The two main extensions are described in the subsequent sections: In Section 4.1, sample weights are introduced to the Gappy POD approach. A description of the main result of the Patenschaft, the so-called *Bayesian Gappy POD* method, is given in Section 4.2.

4.1 Weighted Gappy POD

A simple extension of the ordinary Gappy POD approach is the incorporation of weights for the sensor data. Let the weights for the vector $t = [t_1, \ldots, t_s]^\mathsf{T} \in \mathbb{R}^s$ of experimental data be given by $\omega_1, \ldots, \omega_s \in \mathbb{R}_+$. Weighting can be performed by applying the weighting matrix

$$W = \begin{bmatrix} \omega_1 & & 0 \\ & \ddots & \\ 0 & & \omega_s \end{bmatrix} \in \mathbb{R}^{s \times s} \tag{4.1}$$

to the residual term in the ordinary Gappy POD problem Eq. (3.6),

$$\hat{a}_W = \arg\min_a \left\| W P^\mathsf{T} U_r a - W t \right\|_2^2, \tag{4.2}$$

or the regularized Gappy POD problem Eq. (3.9),

$$\hat{a}_{\text{rr, W}} = \arg\min_a \left\| W P^\mathsf{T} \tilde{U} a - W t \right\|_2^2 + \lambda \|a\|_2^2. \tag{4.3}$$

Note that if the same weighting factor $\omega \in \mathbb{R}_+$ is used for all sensors, i.e. $W = \omega I$, the weighted Gappy POD problem Eq. (4.2) has the same solution as the ordinary Gappy POD problem Eq. (3.6),

$$\hat{a}_W = \arg\min_a \left\| W P^\mathsf{T} U_r a - W t \right\|_2^2 = \arg\min_a \omega^2 \left\| P^\mathsf{T} U_r a - t \right\|_2^2 = \hat{a}. \tag{4.4}$$

In case of regularized Gappy POD, using the same weighting factor $\omega \in \mathbb{R}_+$ for all sensors acts like a parameter which controls the impact of the residual term compared to the additional term. The regularized Gappy POD problem with weighting matrix $W = \omega I$ and regularization parameter λ is equivalent to the regularized Gappy POD problem with regularization parameter $\frac{\lambda}{\omega^2}$ without weights.

By giving higher weight to individual sensors, their influence is increased. Contrary, the influence of sensors can be weakened by setting a low weighting factor. The introduction of weights is therefor an easy way to incorporate expert knowledge on the accuracy of the individual sensor responses. For example if it is expected that the response of pressure sensors at the leading edge of an airfoil are less accurate then at other locations, one can account for this by choosing the weights accordingly.

4.2 Bayesian Gappy POD

The Bayesian Gappy POD extension was first introduced in our work [6]. This introduction is in large parts taken from the initial work. Small modifications have been made in order to make the notation in this work consistent.

The Gappy POD problem Eq. (3.6) can be interpreted as a linear regression problem with sample data

$$\{(x_i, t_i) \mid i = 1, \dots, s\}, \tag{4.5}$$

where $x_i \coloneqq (X)_i = (P^\top U_r)_i = (U_r)_{j_i} \in \mathbb{R}^r$ denotes the i-th row of the matrix $X = P^\top U_r$ or, in other words, the j_i-th row of the matrix of POD modes U_r and $t_i = (t)_i$ is the corresponding sensor response. In this setting, we aim at evaluating the regression model f for all rows of the matrix of POD modes $(U_r)_i, i = 1, \dots, N$, to obtain the Gappy POD solution,

$$\hat{y} = f(U_r) = \left(f\left((U_r)_i\right)\right)_{i=1,\dots,N}. \tag{4.6}$$

In case of ordinary and regularized Gappy POD, the standard linear regression model,

$$f(x) = x^\top w, \tag{4.7}$$

with weight vector $w \in \mathbb{R}^r$ is considered. In this work, we take a Bayesian perspective on the linear regression problem Eq. (4.5) which allows us to derive probability distributions for the data fusion result. While in the ordinary Gappy POD formulation, only point estimates are provided as data fusion result, this new perspective allows to provide valuable

information about the estimated accuracy of the prediction.

In order to give a better insight into our method, we will first review some basics of linear regression from a statistical point of view based on [37, Sec. 2]. This leads us to the concept of Gaussian Process Regression which will then be used to solve the regression problem Eq. (4.5) in our Bayesian Gappy POD extension.

Suppose we are given a training data set $\{(x_i, t_i) \mid i = 1, \dots, s\}$ with input variables $x_i \in \mathbb{R}^r$ and corresponding outputs $t_i = t(x_i), i = 1, \dots, s$. Our goal is to model the relationship between inputs and outputs. The linear regression model with Gaussian noise reads,

$$f(x) = \phi(x)^\top w, \qquad t(x) = f(x) + \varepsilon, \tag{4.8}$$

with a fixed set of basis functions $\phi(x) = (\phi_1(x), \dots \phi_m(x))^\top$, corresponding weight vector $w \in \mathbb{R}^m$, and additive independent, identically distributed Gaussian noise ε with zero mean and stationary variance $\sigma^2 > 0$. The likelihood function, i.e. the probability density of the observations given the weights, is given by

$$\begin{aligned} p(t \mid X, w) &= \prod_{i=1}^{s} p(t_i \mid x_i, w) \\ &= \frac{1}{\sqrt{(2\pi\sigma^2)^s}} \exp\left(-\frac{1}{2\sigma^2}\|t - \Phi^\top w\|_2^2\right), \end{aligned} \tag{4.9}$$

where $t = (t(x_1), \dots, t(x_s))^\top = (t_1, \dots, t_s)^\top$ is the vector of observed outputs and

$$\Phi = [\phi(x_1), \dots, \phi(x_s)] = \begin{bmatrix} \phi_1(x_1) & \dots & \phi_1(x_s) \\ \vdots & \ddots & \vdots \\ \phi_m(x_1) & \dots & \phi_m(x_s) \end{bmatrix} \in \mathbb{R}^{m \times s} \tag{4.10}$$

is the matrix of function values of the basis functions $\phi(x)$ evaluated at the sample points $x_1, \dots, x_s$.

Instead of determining a specific weight vector w, we express our beliefs about the probability distribution of the weight vector w before seeing the data in terms of a *prior distribution* $p(w)$. Based on this prior distribution and the likelihood function, we can now derive the *posterior distribution* of w, that is the probability distribution of w after seeing

the data by making use of Bayes' Theorem, cf. [37, Sec. A.1],

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{marginal likelihood}}, \tag{4.11}$$

$$p(w \mid X, t) = \frac{p(t \mid X, w) \cdot p(w)}{p(t \mid X)}, \tag{4.12}$$

where the *marginal likelihood*,

$$p(t \mid X) = \int p(t \mid X, w) p(w) \, \mathrm{d}w, \tag{4.13}$$

is a normalizing constant independent of the weight vector w.

Finally, predictions for the output $f_* := f(x_*)$ for a new input variable $x_* \in \mathbb{R}^r$ can be made by evaluating the *predictive distribution*,

$$p(f_* \mid x_*, X, t) = \int p(f_* \mid x_*, w) p(w \mid X, t) \, \mathrm{d}w. \tag{4.14}$$

One can show that for different choices of prior distributions $p(w)$, the maximum of the predictive distribution equals the prediction of well-known regression techniques such as ordinary linear regression, Ridge regression and LASSO, cf. [35, Tab. 7.1, p. 226]. However, the Bayesian point of view enables to go a step further: As we consider the weight vector as a statistical quantity, the output f is a statistical quantity as well. Thus, instead of providing a regression model which is a point estimate of the probability distribution for the input-output mapping $f(x)$, we can provide the entire distribution of probable regression models.

Except for some special choices of prior distributions $p(w)$, there is no explicit expression for the predictive distribution in Eq. (4.14). One of those special choices, which allows for an analytical expression of the predictive distribution, is the assumption of a Gaussian prior,

$$p(w) \sim \mathcal{N}(0, \Sigma_w). \tag{4.15}$$

Applying some calculus one can show that in this case the posterior distribution of the weight vector w and, consequently, the predictive distribution of f_* are also Gaussian with

$$\mathrm{E}[f_*] = \phi_*^\mathsf{T} \Sigma_w \Phi \left(\Phi^\mathsf{T} \Sigma_w \Phi + \sigma^2 I \right)^{-1} t, \tag{4.16}$$

$$\mathrm{Var}[f_*] = \phi_*^\mathsf{T} \Sigma_w \phi_* - \phi_*^\mathsf{T} \Sigma_w \Phi \left(\Phi^\mathsf{T} \Sigma_w \Phi + \sigma^2 I \right)^{-1} \Phi^\mathsf{T} \Sigma_w \phi_* \tag{4.17}$$

where $\phi_* := \phi(x_*)$, [37, p. 17]. Defining the function

$$k(x, x') = \phi(x)^\top \Sigma_w \phi(x'), \tag{4.18}$$

we can rewrite these expressions in terms of the function k,

$$\mathrm{E}[f_*] = k(x_*) \left(K + \sigma^2 I\right)^{-1} t, \tag{4.19}$$

$$\mathrm{Var}[f_*] = k(x_*, x_*) - k(x_*)^\top \left(K + \sigma^2 I\right)^{-1} k(x_*), \tag{4.20}$$

where $k(x_*) := (k(x_*, x_i))_{i=1,\dots,s} \in \mathbb{R}^s$ and $K := \left(k(x_i, x_j)\right)_{i,j=1,\dots,s} \in \mathbb{R}^{s\times s}$. It is easy to show that $f(x)$ defines a zero-mean Gaussian process whose covariance $\mathrm{Cov}[f(x), f(x')]$ for any two inputs $x, x' \in \mathbb{R}^r$ is given by $k(x, x')$,

$$f(x) \sim \mathcal{GP}(0, k(x, x')). \tag{4.21}$$

Because of this property, the function $k(x, x')$ is also called *covariance function*, [37, p. 12].

In the above setting, a set of basis functions $\phi(x)$ has been chosen and assumptions on the prior distribution of the weight vector w have been made, which implicitly define the covariance of the Gaussian process $f(x)$. The key idea of Gaussian Process Regression, also known as Kriging, is to bypass this concept by defining a positive definite covariance function for $f(x)$ explicitly, [9, p. 160]. In turn, this implicitly defines a (potentially infinite) set of basis functions $\phi(x)$ due to Mercer's theorem, [35, p. 483]. Of course, the choice of the covariance function has a large impact on the predictions. It is typically chosen such that it reflects the property that the outputs $f(x)$ and $f(x')$ of input variables x and x' that are *close* or *similar* to each other are more strongly correlated than the outputs of more distinct input variables, [9, Sec. 6.4.2]. A widely used class of covariance functions is given by

$$k(x, x') = \theta_0 \cdot \exp\left(-\theta_1 \|x - x'\|^2\right) + \theta_2 x^\top x', \tag{4.22}$$

cf. Eq. (6.63) in [9, Sec. 6.4, p. 307], with hyperparameters $\theta_0, \theta_1, \theta_2$. Instead of defining the hyperparameters in advance, they are usually determined from the data by maximizing the log marginal likelihood function,

$$\ln p(t \mid X) = -\frac{1}{2} t^\top (K + \sigma^2 I)^{-1} t - \frac{1}{2} \ln \det(K + \sigma^2 I) - \frac{s}{2} \ln 2\pi. \tag{4.23}$$

For a detailed discussion of covariance functions for GPRs and different methods for hyperparameter estimation, the reader is referred to Sec. 4 and Sec. 5.4 of the textbook [37].

Remark. In some cases, the noise variance σ^2 of the observations can be estimated e.g. from expert knowledge or derived from information about the accuracy of the physical sensors used. Otherwise it can be determined with the hyperparameters of the covariance kernel during the marginal likelihood optimization.

The derivation of Gaussian Process Regression from linear regression with a special choice of prior information motivates our Bayesian regression extension for Gappy POD: We propose to solve the Gappy POD problem Eq. (4.5) by employing GPRs. A kernel function $k(x, x')$ is chosen from the class Eq. (4.22) such that the log marginal likelihood function Eq.(4.23) is maximized. Afterwards, a predictive distribution for all rows of the matrix of POD modes, $(U_r)_i, i = 1, \ldots, N$, is obtained by evaluating Eq. (4.19) and Eq. (4.20).[1]

Note that the final data fusion result will in general not lie in the POD subspace, as is the case for ordinary or regularized Gappy POD. Consequently, no vector of POD basis coefficients can be found.

The new Bayesian Gappy POD extension, as well as the established approach have been implemented in DLR's python-based Surrogate Modeling for AeRodata Toolbox SMARTy, [3].

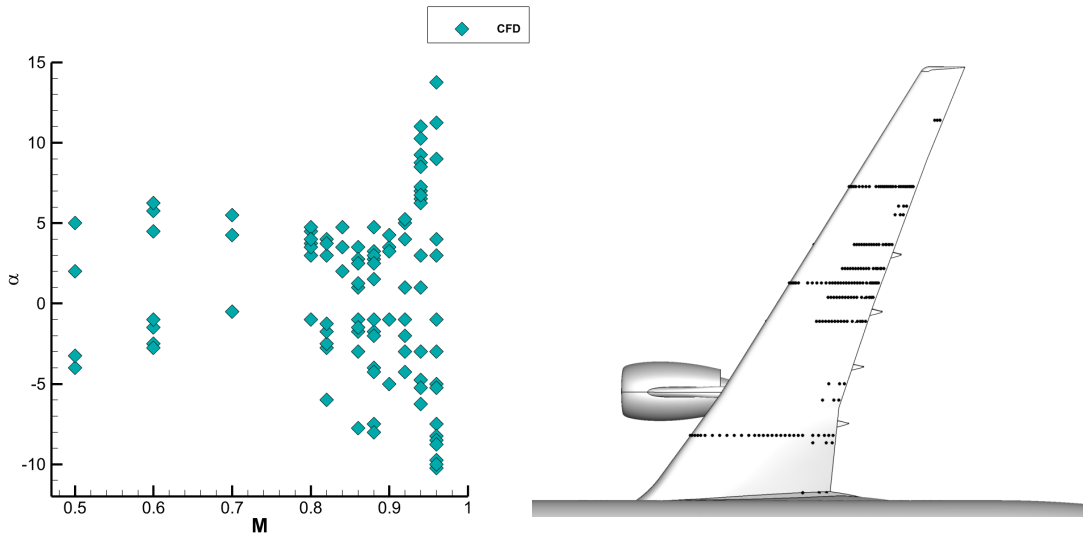
4.3 Application to a Transport Aircraft Test Case

Large parts of this section are taken from our work [6].

4.3.1 Case study description

The test case configuration considered here is an industrial-relevant aircraft configuration known as XRF1. The XRF1 is an Airbus provided industrial standard multi-disciplinary research test case representing a typical configuration for a long range wide body aircraft. It is used by Airbus to engage with external partners on development and demonstration of relevant capabilities and technologies. For the computational data, a highly resolved CAD model of the XRF1 is considered as half configuration. The corresponding surface grid consists of $N = 388,918$ grid points. High-fidelity RANS-CFD simulations were carried out with the DLR flow solver TAU, [41] using the SST turbulence model. Structural deformation are accounted for by coupling the CFD analysis to a computational structural mechanics (CSM) investigation which relies on a finite element model of the wind tunnel aircraft structure. In this way, a total number of $n = 100$ pressure coefficient (c_p) distributions on the surface of the aircraft were computed for different Mach numbers $\mathrm{M} \in [0.5, 0.96]$ and angles of

[1] In this setting each entry of the Gappy POD solution $(\hat{y})_i, i = 1, \ldots, N$ is a random variable.



(a) CFD data base (b) Pressure tap locations

Figure 4.1: Data-points which are sampled during the numerical analysis as well as the location of pressure taps on the upper wing surface

attack $\alpha \in [-11°, 14°]$ and are considered as CFD snapshots during the following investigations. The sampling points are displayed in Figure 4.1a. The Reynolds number was fixed to $\mathrm{Re} = 25 \times 10^6$. Wind tunnel tests for the XRF1 configuration were carried out in the European Transonic Windtunnel (ETW). The surface pressure coefficient was obtained at 314 pressure taps distributed over 26 section cuts along the wing with locations shown in Figure 4.1b for a number of 196 different combinations of Mach number and angle of attack.

The sampled parameter combinations cover a wide range of aerodynamic phenomena including fully attached flow, severe trailing edge separation, strong shock waves on the upper and lower wing surface as well as shock induced separation. Based on the specific configuration at hand and observed aerodynamic characteristics, the Mach number vs. angle of attack diagram has been divided in different regions as shown in Figure 4.2. In particular these are the design range, the linear region and the nonlinear region. Note that the XRF1 is an aircraft concept provided from Airbus for the means of demonstration of relevant capabilities and technologies and displayed regions should be regarded as a generic classification of flow phenomena based on common aerodynamic knowledge. Moving from the design region towards the edge of the envelope it is expected that the prediction accuracy of numerical simulation tools decreases [1]. Hence, discrepancies be-

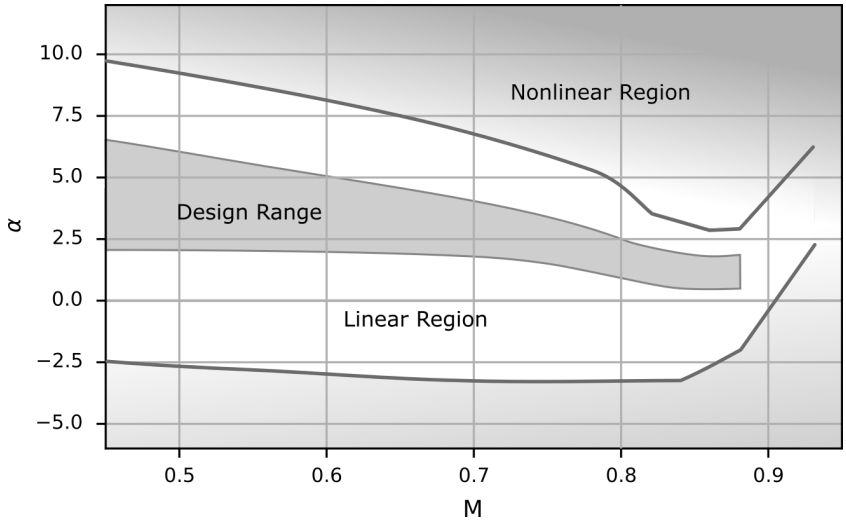


Figure 4.2: XRF1 envelope zones, kindly provided by Airbus.

tween numerical and experimental results are likely to grow. This should directly result in enlarged credible intervals for fused pressure distributions at more challenging conditions.

4.3.2 Results

Different points within the envelope are chosen to provide an insight into the data fusion capabilities and are discussed next. Especially within the nonlinear region, towards the edge of the envelope, information provided by CFD snapshots is sparse and likely also inaccurate due to various simulation challenges including turbulence modeling. For all presented cases, data fusion was employed using regularized Gappy POD with Ridge regression, cf. 3.2, and the new Bayesian Gappy POD approach with Gaussian Process Regression as introduced in Sec. 4.2. In the latter case, the kernel function was chosen from the class Eq. (4.22), where the hyperparameters $\theta_0, \theta_1, \theta_2$ were determined by maximizing the marginal likelihood function Eq. (4.23). Two different studies were carried out:

1. At first, the noise variance σ^2 of the sensor data is assumed to be unknown and estimated along with the kernel hyperparameters in the maximum likelihood optimization.

2. In a second study, the noise variance σ^2 is determined a priori from the manufacturer's specifications on the accuracy of the pressure sensors used in the wind tunnel experiment. Other sources of uncertainty are neglected.

Note that in both cases, we assume the noise variance σ^2 to be stationary meaning that the measurements from all pressure taps are corrupted by the same type of noise.

In addition to the data fusion results, a surrogate model was constructed solely based on the same CFD data set by interpolating the POD basis coefficients—a widespread data-driven surrogate modeling technique introduced by [33]. For the interpolation of the basis coefficients, thin plate spline (TPS) interpolation, [11], was considered. For comparison, the CFD-based surrogate model was evaluated at the flow conditions of interest.

For a quantitative assessment of the agreement with the wind tunnel sensor measurements, the results $\hat{y}$ of every investigated method were evaluated by means of the root mean squared error,

$$\mathrm{RMSE}(\hat{y}) := \sqrt{\frac{1}{s}\|P^{\mathsf{T}}\hat{y} - t\|_2^2}. \tag{4.24}$$

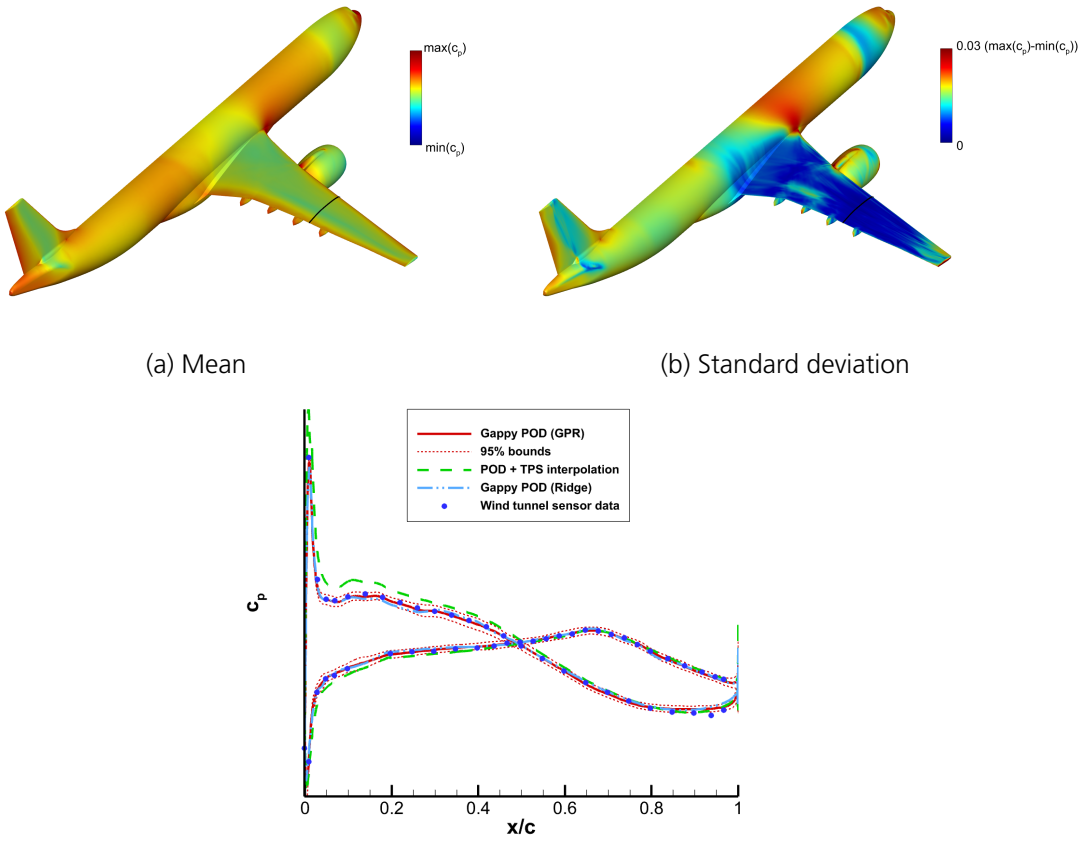
The Bayesian Gappy POD approach introduced in Sec. 4.2, yields a predictive distribution instead of a point estimate that is given by the other methods considered in this study. In every surface grid point, the data fusion result is a Gaussian distributed random variable with a certain mean and variance. In the following, we account for this distribution by always showing the predicted mean together with its standard deviation, i.e. the square root of the variance, or giving 95 % credible intervals.[2] The other investigated methodologies (POD+TPS and Gappy POD based on Ridge regression) only provide point estimates for comparison. Note that parts of the first study were also included in our previous work, [6]. However, the results shown here slightly differ from those presented in [6] by the fact that in this paper, we applied the scaling of the POD basis vectors, as introduced for regularized Gappy POD in eq. (3.8), to both Gappy POD methods which further improved the results.

Results with estimated noise variance

The first analyzed flow condition is $\mathrm{M} = 0.50, \alpha = -1.97°$ which is within the linear region. The mean and the standard deviation of the surface pressure distribution obtained from the Gappy POD approach using GPR is shown in 4.3a and 4.3b, respectively, while a comparison between different methods in a section cut at $\eta = 0.55$ (indicated by a solid black line in the surface plots) is displayed in 4.3c.

A suction peak is observed on the leading edge of the wing slightly shifted towards the lower side of the wing due to the negative angle of attack. The rest of the wing exhibits are smooth pressure distribution as expected at subsonic flow conditions. When comparing the prediction accuracy for the selected section cut, no differences are observed

[2] 95 % credible intervals correspond to the mean plus/minus 1.96 times standard deviation.



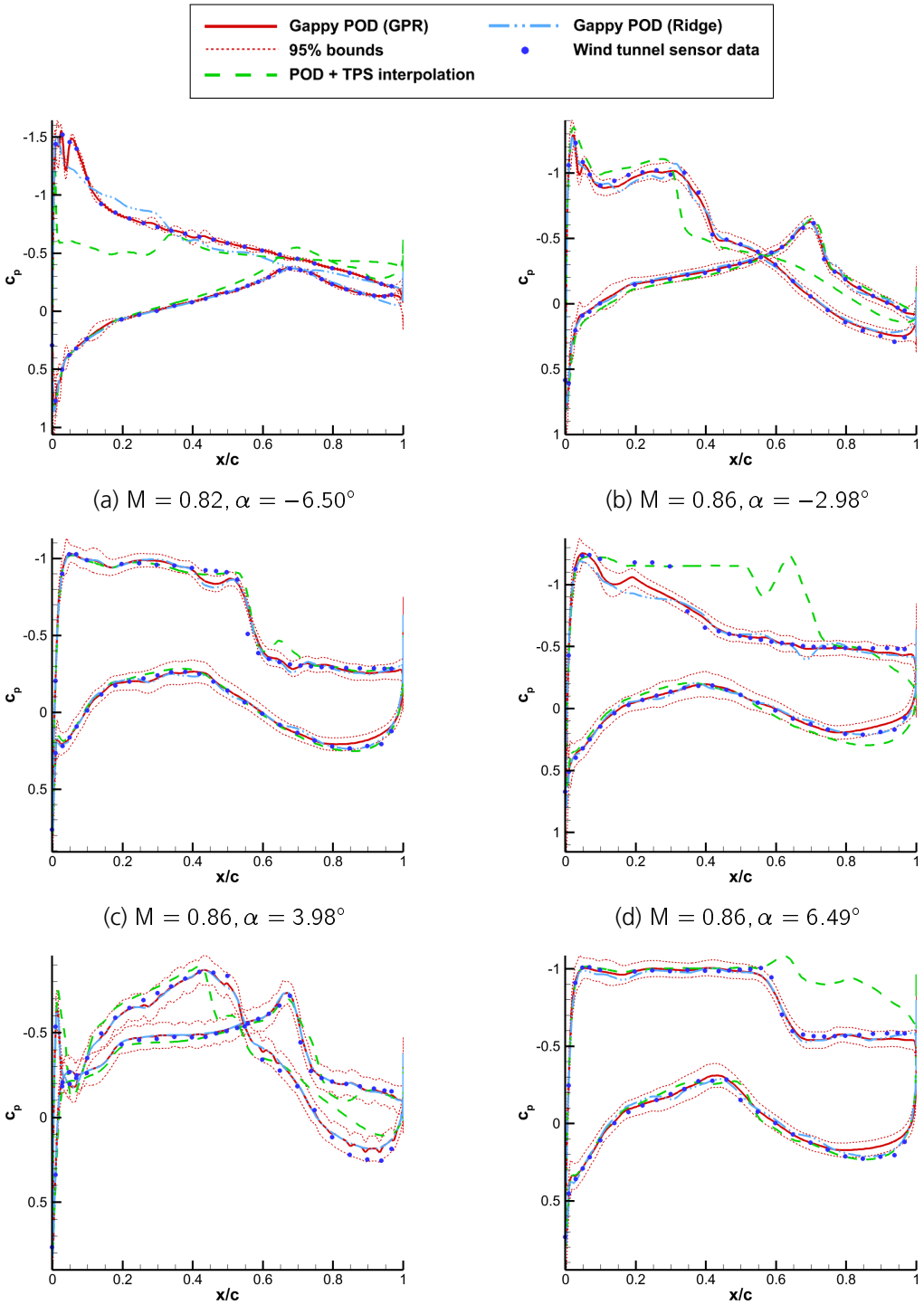
(a) Mean

(b) Standard deviation

(c) Pressure coefficient distribution at selected cut.

Figure 4.3: Pressure coefficient approximation for $M = 0.50, \alpha = -1.97°$ via Gappy POD combined with Ridge regression and GPR in comparison to POD plus TPS interpolation at a wing section cut where wind tunnel sensor data is available. The solid black line indicates the cut location.

between both gappy POD methods (Ridge regression and GPR) whereas the purely CFD-based POD + TPS methodology slightly over-predicts the pressure levels on the lower wing surface. The 3D view of the standard deviation in Figure 4.3b shows that it is the smallest in areas of very uniform flow (e.g. some regions at the fuselage) or in areas where many wind tunnel sensors are located. In areas of non-uniform flow and only a few available measurements (e.g. behind the engine or close to the wing tip) an increase of the standard deviation is observed. This corresponds to a less smooth pressure distribution as displayed in Figure 4.3a. Additional results in the transonic regime are shown in Figure 4.4.



(a) $M = 0.82, \alpha = -6.50°$ (b) $M = 0.86, \alpha = -2.98°$

(c) $M = 0.86, \alpha = 3.98°$ (d) $M = 0.86, \alpha = 6.49°$

(e) $M = 0.92, \alpha = 0.02°$ (f) $M = 0.92, \alpha = 7.01°$

Figure 4.4: Pressure coefficient approximation via Gappy POD combined with Ridge regression and GPR in comparison to POD plus TPS interpolation at the selected wing section cut.

For a quantitative comparison of the results, the root mean squared error Eq. (4.24) for all investigated test cases and methods is given in Table 4.1.

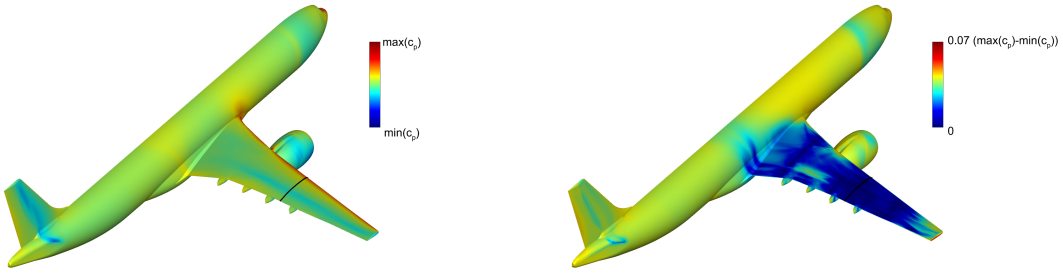
Mach	α [°]	$\mathrm{RMSE}(\hat{y}) \cdot 10^2$		
		POD + TPS	Gappy POD (Ridge)	Gappy POD (GPR)
0.50	−1.97	10.2	3.2	2.1
0.82	−6.50	25.0	7.2	0.1
0.86	−2.98	10.1	5.8	2.5
0.86	3.98	8.7	5.6	5.0
0.86	6.49	26.4	9.0	5.3
0.92	0.02	9.1	5.2	3.3
0.92	7.01	20.0	8.1	7.2

Table 4.1: Root mean squared error of the pressure coefficient approximation for POD + TPS interpolation, Gappy POD with GPR and Gappy POD with Ridge regression.

Figure 4.4a shows the data for the case $\mathrm{M} = 0.82, \alpha = -6.50°$, which is well within the negative nonlinear region. As in the first case a rather smooth pressure distribution is observed, with the Gappy POD with GPR accurately reproducing the wind tunnel data. While the Gappy POD with Ridge regression approach is able to reproduce the overall trend it fails to exactly match the c_p level on the lower surface. The result of the simple POD + TPS interpolation is far off the wind tunnel test result. This can be attributed to the underlying CFD data where the closest data point features a leading edge separation on the lower surface. In comparison to the section cut plot of the first analyzed parameter combination, Figure 4.3c, a completely different behavior of the 95 % credible bounds are observed. While in the first investigated case the predictive variance is dominated by the noise variance, it is strongly driven by the process variance of the GPR model in the second case. Here, the hyperparameter optimization of the covariance function yields a comparatively small noise variance. As a result, the credible intervals are very narrow at the pressure tap locations and get significantly larger apart from the data. While Gappy POD with GPR predicts a smooth mean, the corresponding root mean squared error with respect to the wind tunnel data is one order of magnitude smaller than for all other test cases presented in detail in this study. This shows that the method was able to obtain a very good match with the given data.

These two cases demonstrate that the optimization of the hyperparameters is a challenging task and crucial for resulting variances. The loss function Eq. (4.23) is often multimodal and thus has multiple local maxima, cf. [37, Sec. 5.4]. For example, by looking only at the data, it is often not clear if the fluctuation in the data originates either from the true

relationship of input and output or from noise. More data or information about the uncertainty associated with the data may help to improve the hyperparameter tuning. A detailed discussion is however beyond the scope of this work. Because of this different behavior, additional surface plots of the mean and standard deviation are given in Figure 4.5.



(a) Mean (b) Standard deviation

Figure 4.5: Mean and standard deviation of pressure coefficient distribution for $M = 0.82, \alpha = -6.50^\circ$ as estimated with Bayesian Gappy POD.

As indicated by the section cut plot in Figure 4.4a, the surface plot of the standard deviation in Figure 4.5b shows the characteristic increase of the standard deviation in between the pressure taps leading to greenish streaks especially on the wing. On section cuts where many pressure taps are located, these streaks are interrupted by dark blue lines indicating that the standard deviation in this regions is estimated to be small. In addition, a larger increase of the standard deviation in areas of non-uniform flow and stronger recompression (e.g. on the vertical tail plane and forward fuselage) is observed. This can be interpreted that the uncertainty in the model prediction in these regions is estimated to be high. This is partly in line with our expectations on the corresponding uncertainty in the data fusion result as from an aerodynamic point of view it is indeed difficult to estimate accruing flow phenomena in these regions.

Another case in the negative nonlinear region is presented in 4.4b. Note that, due to the negative angle of attack, a shock wave is present on the lower side of the wing. The CFD-based POD + TPS surrogate model shows a good match of the upper surface pressure distribution, while significant differences to the wind tunnel sensor data can be observed for the lower wing surface. Especially the shock position is too close to the leading edge. Both Gappy POD approaches match the c_p levels of the wind tunnel sensor data including a good representation of both (upper and lower surface) shocks in terms of strength and location. The result obtained with Gappy POD with Ridge regression shows some unphysical oscillations especially for the pressure distribution on the lower surface.

In comparison, a more smooth trend and a better match with the wind tunnel data can be observed for Gappy POD with GPR. This impression is also in line with the overall root mean squared error being less than half as large as for Gappy POD with Ridge regression. Figure 4.4c shows a case at a positive angle of attack of $\alpha = 3.98°$ and $\mathrm{M} = 0.86$. Both Gappy POD approaches accurately reproduce the characteristics of the pressure distribution as measured in the wind tunnel test, including a supersonic plateau and a strong shock on the upper surface. The c_p plateau downstream of the shock, indicating a shock induced separation, is matched as well. The POD + TPS interpolation however fails to accurately reproduce the shock. Further increasing the angle of attack to $\alpha = 6.49°$ (4.4d) and hence extrapolating beyond the range of available CFD data, both applied Gappy POD approaches fail to match the shock. In contrast to the POD + TPS extrapolation, the c_p level after the shock is however matched. The $95\,\%$ credible interval is increased, showing the increased uncertainty of this data fusion result. Although there is a large discrepancy in the prediction of the plateau in front of the shock, the comparatively small root mean squared error for this test case indicates that the wind tunnel measurements in other regions of the wing are better approximated by the Gappy POD methods. The final two data points demonstrate the capability of the presented data fusion method towards the edge of the flight envelope, close to the dive Mach number. In 4.4e the data for $\mathrm{M} = 0.92, \alpha = 0.02°$ is presented. A number of different aerodynamic features (leading edge suction peak, shocks on upper and lower surface) are accurately matched. However, the resulting pressure distribution is not smooth everywhere and the uncertainty bounds are increased. With a smaller root mean squared error as for the last two test cases, both Gappy POD method however show a remarkable good match with the wind tunnel data. In Figure 4.4f both Gappy POD methods accurately predict the strength and position of the shock. While Gappy POD with GPR shows a slightly smoother trend as the state-of-the-art approach, the flow behavior at the trailing edge is better captured by Gappy POD with Ridge regression.

The discrepancy between the purely CFD-based POD + TPS interpolation result and the Gappy POD methods are underlined by a large difference in the root mean squared error for all investigated cases. Compared to the established Gappy POD with Ridge regression, the results obtained by Gappy POD with GPR show a smaller root mean squared error in all cases. In addition to the cases presented above, the root mean squared error was evaluated for all available 196 data sets from the wind tunnel experiment. In Figure 4.6, the average root mean squared error for the predictions of the investigated methods is displayed divided by the different regions of the envelope.

As can be seen from the bar plot, the Gappy POD approaches outperform the simple

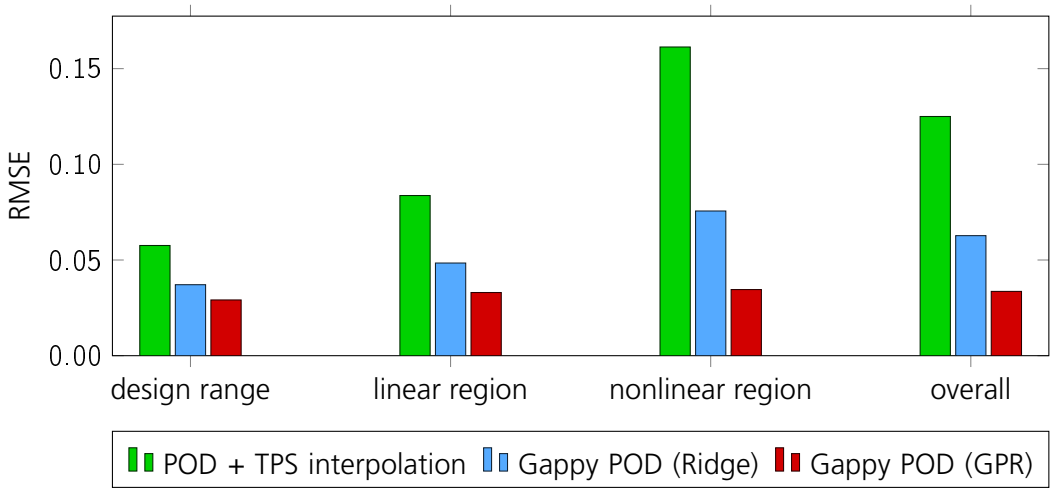


Figure 4.6: Average root mean squared error with respect to the 314 wind tunnel pressure probes for Gappy POD with Ridge regression, Gappy POD with GPR and POD with TPS interpolation for 196 wind tunnel data sets in the different domains of the envelope.

POD + TPS interpolation in all regions of the envelope. While the differences between the two Gappy POD approaches in the design range and linear region appear to be small, Gappy POD with GPR yields a significant improvement in the nonlinear region where the average root mean squared error is less than half as large as for Gappy POD with Ridge regression. Averaged over all 196 wind tunnel test cases, the root mean squared error of Gappy POD with GPR gives a root mean squared error which is about 47 % smaller than the root mean squared error of the established Gappy POD with Ridge regression.

Results for known noise variance

In the second part of this case study, we are aiming at investigating the effect of providing information on the measurement uncertainty. Two sources of measurement error have been considered for this test case: The error due to the precision of the sensors, mounted at the wind tunnel model, and the error due to leakage at the pressure inlet holes and tubes. For simplicity, both errors are assumed to be Gaussian distributed and additive. Furthermore, it is assumed that the wind tunnel measurements were not distorted by any other source of error, i.e. errors due to the positioning of the sensors, the general experimental setup, the model discretization and other possible sources, are neglected in this study. We decided on this simplification since the purpose of this work lies more in an quantitative investigation of the effect of using this kind of information and not in a most

exact description of all existing uncertainties.

The same 7 test cases as in the previous section are investigated in detail in the following. For all of these cases, Table 4.2 shows the assumed noise variance due to sensor precision and leakage effects and the one estimated from the hyperparameter optimization.

M	α [°]	noise variance $\sigma^2 \cdot 10^3$	
		assumed	estimated
0.50	-1.97	6.82	0.66
0.82	-6.50	4.68	0.02
0.86	-2.98	4.54	1.35
0.86	3.98	4.54	3.44
0.86	6.49	4.54	4.33
0.92	0.02	4.37	1.91
0.92	7.01	4.37	6.32

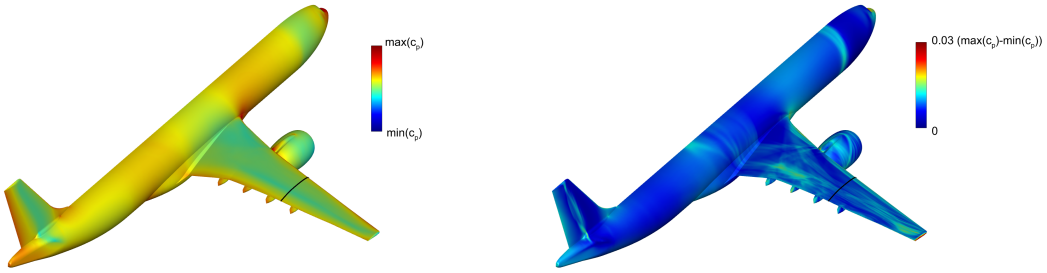
Table 4.2: Estimated and assumed values of noise variance.

As in the previous section, the first analyzed flow condition from the linear region is $\mathrm{M} = 0.50, \alpha = -1.97°$. Figure 4.7a and 4.7b show the mean and standard deviation of the estimated surface pressure coefficient distribution obtained when the noise variance is provided to the Gappy POD with GPR model. A direct comparison of the results of the Bayesian Gappy POD approach with known and estimated noise variance for the selected section cut is given in Figure 4.7c.

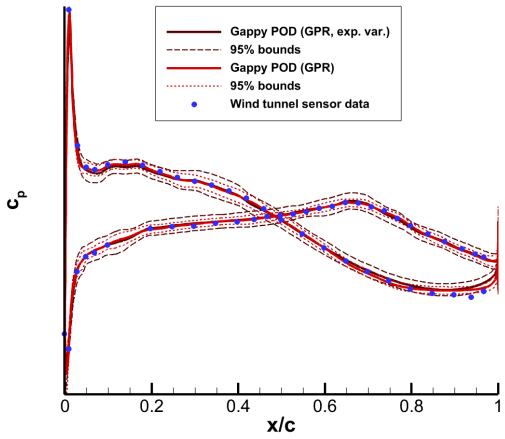
While the standard deviation decreases in large regions of the aircraft's surface in Figure 4.7b in comparison to 4.3b, it increases slightly at the wing. This is also observed in the section cut plot in Figure 4.7c: The mean of the result of both methods is only visibly distinguishable in few small regions of the airfoil while the 95 % credible bounds for Gappy POD with estimated noise variance are in most regions smaller than the bounds for Gappy POD with assumed noise variance.

Results for the same six test cases from the transonic flow regime are shown in Figure 4.8.

For a Mach number of $\mathrm{M} = 0.82$ and an angle of attack of $\alpha = -6.5°$, the estimated noise variance is two orders of magnitude smaller than the assumed one, cf. Tab. 4.2. As a result, Gappy POD with assumed noise variance gives a significantly different prediction than Gappy POD with estimated noise variance: While both methods predict a very similar mean and give a good approximation of the observed wind tunnel data in Figure 4.8a, the credible bounds of Gappy POD with assumed noise variance do not show the remarkable narrowing at the sample locations. For all other investigated cases, the estimated noise



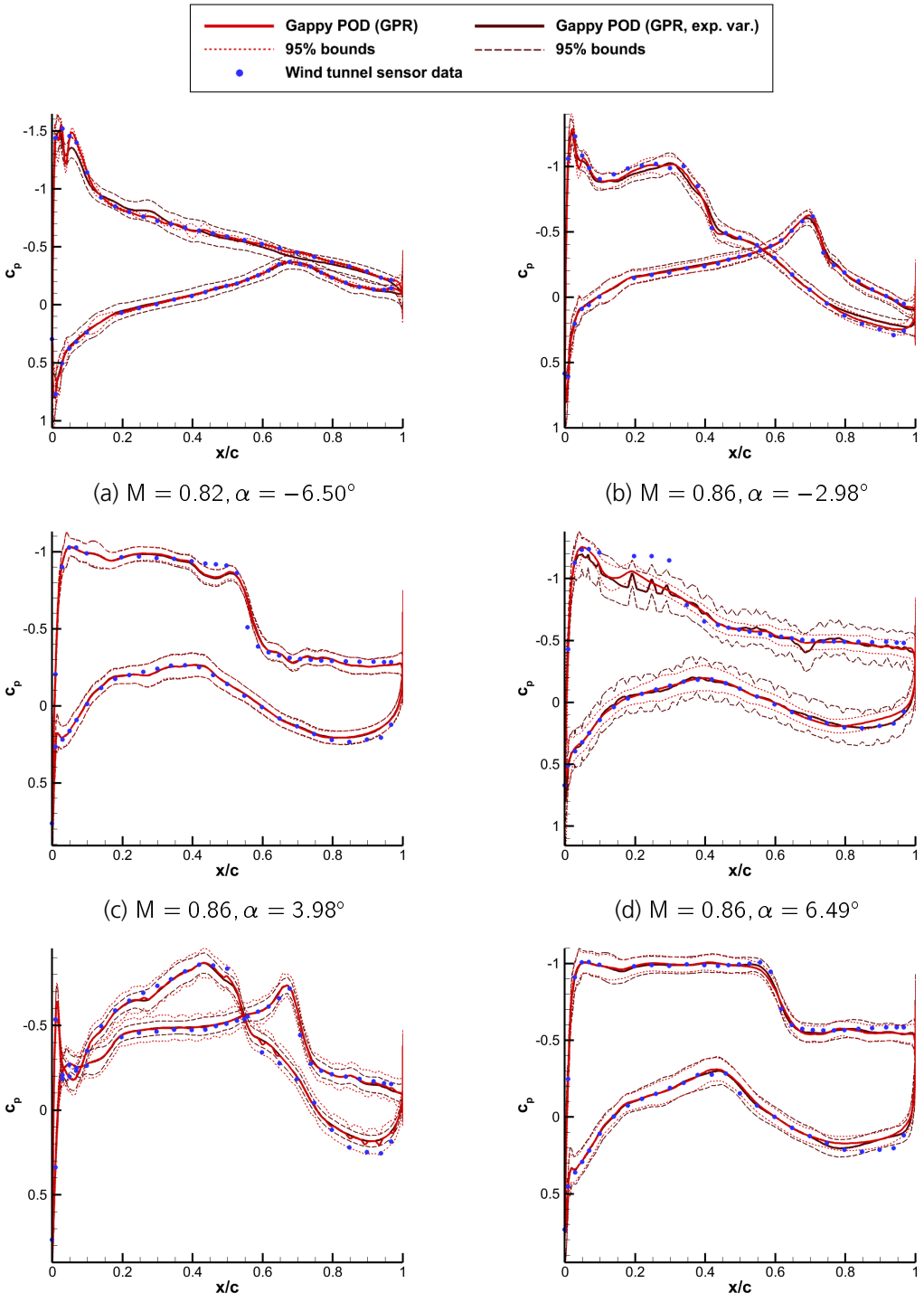
(a) Mean (b) Standard deviation



(c) Pressure coefficient distribution at selected cut.

Figure 4.7: Pressure coefficient approximation for $\mathrm{M} = 0.50, \alpha = -1.97°$ via Gappy POD combined with Ridge regression and GPR in comparison to POD plus TPS interpolation at the marked section cut.

variance is in the same order of magnitude as the assumed one, which results in similar predictions for $(\mathrm{M}, \alpha) = (0.86, -2.98°)$ in Fig. 4.8b, $(\mathrm{M}, \alpha) = (0.86, 3.98°)$ in Fig. 4.8c, $(\mathrm{M}, \alpha) = (0.92, 0.02°)$ in Fig. 4.8e and $(\mathrm{M}, \alpha) = (0.92, 7.01°)$ in Fig. 4.8f. Although the estimated and assumed variance for $\mathrm{M} = 0.86$ and $\alpha = 6.49°$ are very close, the corresponding predictions in Fig. 4.8d show significant differences, indicating that a different local minimum was found during the hyperparameter optimization.



(a) $M = 0.82, \alpha = -6.50°$

(b) $M = 0.86, \alpha = -2.98°$

(c) $M = 0.86, \alpha = 3.98°$

(d) $M = 0.86, \alpha = 6.49°$

(e) $M = 0.92, \alpha = 0.02°$

(f) $M = 0.92, \alpha = 7.01°$

Figure 4.8: Pressure coefficient approximation via Bayesian Gappy POD with estimated and given noise variance at the selected wing section cut.

5 Shallow neural networks for fluid flow reconstruction

In [15], an approach for fluid flow reconstruction was presented using a shallow neural network. Since we investigated this approach in the scope of this Patenschaft, it is shortly revised in the following. For more details, the reader is referred to the original work.

5.1 Setup

5.1.1 Architecture

A two hidden layer, fully-connected MLP is used to describe the mapping between the sensor data $t \in \mathbb{R}^s$ and the corresponding full surface solution $\hat{y} \in \mathbb{R}^N$. The numbers of neurons of the two hidden layers, n_1 and n_2, are chosen such that it increases from layer to layer, $s \leq n_1 \leq n_2 \leq N$. The last hidden layer is connected to the output layer via a linear activation function,

$$\hat{y} = \Phi z^2 + b, \tag{5.1}$$

with weights $\Phi \in \mathbb{R}^{N \times n_2}$, bias $b \in \mathbb{R}^N$ and where $z^2 \in \mathbb{R}^{n_2}$ denotes the output of hidden layer 2. In this setting, the columns of the weighting matrix Φ and the bias b define an affine subspace in which the network output is contained:

$$\mathcal{V} = \{y \mid y = \Phi z + b, z \in \mathbb{R}^{n_2}\} \subset \mathbb{R}^N. \tag{5.2}$$

Figure 5.1 illustrates the architecture of the network.

5.1.2 Training issue

In contrast to Gappy POD methods, pairs of input and output data are needed for model training, i.e. the full surface solution must be available for the sensor data used for training.

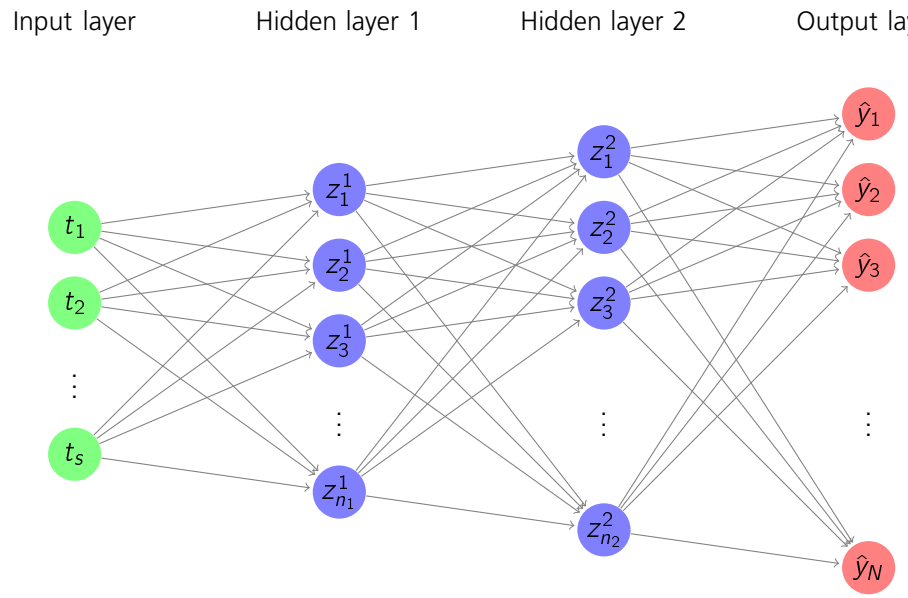


Figure 5.1: Architecture of the shallow neural network

Note that this is in general not the case when dealing with wind tunnel sensor data. To overcome this problem, the following training configurations are possible:

1. **Sensor and CFD data:** A set of sensor measurements is used as input data for training. As corresponding outputs, the corresponding full surface solution is computed for the same flow conditions using CFD simulation.

2. **CFD data only:** The training of the model is restricted to CFD data only. While the full solution serves as output, the information at the nearest neighbors of the sensor locations in the CFD mesh is used for input.

3. **Sensor and PSP data:** Pairs of sensor measurements and their corresponding pressure sensitive paint (PSP) result at a discrete mesh are used for training. CFD data is not considered.

Each of the three configurations has certain advantages and drawbacks. Option 1 seems to be the most intuitive way to overcome the training issue. However, by taking the CFD data as output for sensor data input, it is assumed that the CFD data is the absolute truth for the observed wind tunnel data. The neural network therefore learns to find the CFD result based on wind tunnel measurements which is in general undesirable.

Only relying on CFD data for training is a more consistent setting and is very similar to the Gappy POD methods. On the other hand, option 2 has the clear drawback that, the

final model was not trained on the data which it will be later used on.

Using sensor data together with PSP data for training, option 3, has the advantage that both kinds of data are obtained from the same wind tunnel experiment. Thus, the data should show a good consistency. The final model output can be viewed as surrogate for the PSP data based on sensor inputs. Note that this is completely different from what Gappy POD methods are aiming for.

Because of the similarities with the Gappy POD methods in terms of the setup, we decided on investigating option 2 in the Patenschaft.

5.2 Theoretical investigation of the method

As stated above and mentioned in [15], the output of the neural network lies in an affine subspace $\mathcal{V}$ defined by the weights and the bias of the linear activation function connecting the last hidden and the output layer, cf. eq. (5.2). Without loss of generality, assume the training output vectors $\tilde{y}^i$, $i = 1, \dots, n$ to be centered, i.e.

$$\sum_{i=1}^{n} \tilde{y}^i = 0. \tag{5.3}$$

Because of this assumption, the last activation function can be simplified by setting the bias to zero, $b = 0$, which turns the affine subspace into a linear subspace. The Eckart-Young Theorem, [23, Theorem 2.4.8], gives an optimality condition for the solution space given the training data: The linear subspace of dimension $n_2 < N$ which best approximates the given training data with respect to the 2-norm, is spanned by the first n_2 left singular vectors of the training matrix $Y = [\tilde{y}^1, \dots, \tilde{y}^n] \in \mathbb{R}^{N \times n}$. The following lemma is a direct consequence:

Lemma 5.1. *Let $\mathcal{Y} = \{y^i \mid i = 1, \dots, n\} \subset \mathbb{R}^N$ be a set of centered training snapshots. Let a neural network with the above architecture be trained on the training snapshot set $\mathcal{Y}$ and let the corresponding solution space of dimension n_2 be $\mathcal{V}$. Furthermore, let $\mathcal{U} = \operatorname{span}\{u^1, \dots, u^{n_2}\}$ be the n_2-dimensional POD subspace of the space $\operatorname{span}(\mathcal{Y})$ spanned by the snapshots. Let the orthonormal projections onto $\mathcal{V}$ and $\mathcal{U}$ be given by $P_{\mathcal{V}}$ and $P_{\mathcal{U}}$, respectively. Then,*

$$\|Y - P_{\mathcal{U}} Y\|_2 \leq \|Y - P_{\mathcal{V}} Y\|_2, \tag{5.4}$$

where $Y = [y^1, \dots, y^n] \in \mathbb{R}^{N \times n}$ is the snapshot matrix.

Proof. Due to the Eckart-Young Theorem, [23, Theorem 2.4.8], the linear subspace of dimension n_2 with the smallest projection error for Y with respect to the 2-norm is given by the POD subspace. □

With other words: The linear subspace of fixed dimension $n_2 < n$ which captures most information and thus best approximates the training snapshots $\mathcal{Y}$ is given by the POD subspace $\mathcal{U}$. Although we can not expect the solution space of the neural network to better approximate the training snapshots, it is very relevant how the approximation in the solution space is obtained: Let $t \in \mathbb{R}^s$ be a vector of observations with corresponding "true" output $y \in \mathbb{R}^N$, that is

$$t = \begin{bmatrix} t_1 \\ \vdots \\ t_s \end{bmatrix} = \begin{bmatrix} y_{j_1} \\ \vdots \\ y_{j_s} \end{bmatrix} = P^\mathsf{T} y \tag{5.5}$$

for a mask matrix $P := [e_{j_1}, \dots, e_{j_s}] \in \mathbb{R}^{N \times s}$. Ordinary Gappy POD finds a solution in the POD subspace by computing a least-squares fit of t in the space $P^\mathsf{T}\mathcal{U}$ spanned by the columns of $P^\mathsf{T}U = [P^\mathsf{T}u^1, \dots, P^\mathsf{T}u^{n_2}]$. This least-squares solution is obviously highly dependent on the rows $j_1, \dots, j_s$ of the POD matrix U selected by the mask matrix P. Thus, the final approximation of the output y is in general not the optimal solution in the solution space. The idea of introducing the neural network approach in [15] was to overcome this problem and directly invoke the inputs when creating the solution space. The major difference to our data fusion setup is that a proper training data set with wind tunnel sensor measurements as inputs and corresponding full high-dimensional output is not given.

5.3 Application to an airfoil test case

The test case used to assess the shallow neural network for fluid flow reconstruction is the RAE2822 airfoil, [14], shown in Fig. 5.2. CFD and wind tunnel data were provided by

Figure 5.2: The RAE2822 airfoil

Airbus within the RWC.01 data base. The Airbus RWC.01 data base gathers aerodynamic experimental data acquired in 2016 using the pilot facility of the European Transonic Wind Tunnel (pETW) for a series of 2D airfoil sections. In particular, the reference RAE2822 section geometry equipped with thicker trailing edge was tested to cross-check results with legacy data and extends the range of operating conditions. For different combinations of Reynolds number, Mach number and angle of attack, the pressure coefficient was measured with 36 pressure taps located at the surface of the airfoil. The computational grid consists of 531 surface grid points. A total number of 11466 RANS-CFD simulations for different combinations of the three parameters are available – all of them were generated unsing the DLR flow solver TAU, [41] with the SST turbulence model. The design of experiment is displayed in Figure 5.3.

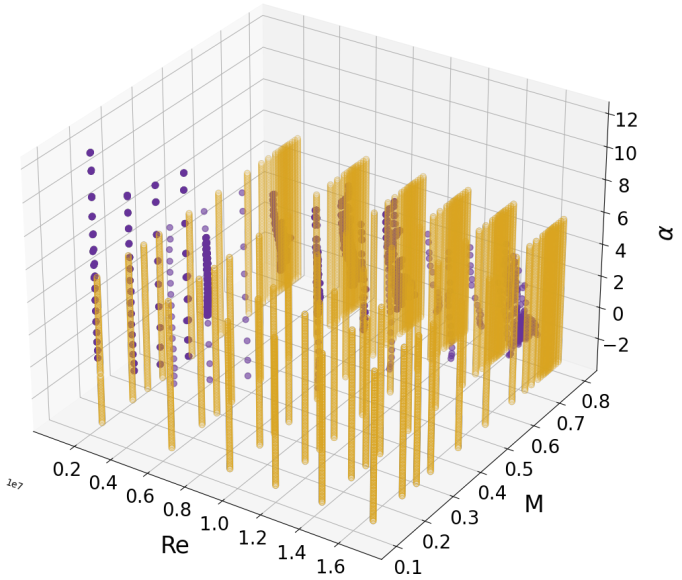


Figure 5.3: Parameter combinations for which wind tunnel data (purple) and CFD data (orange) is available.

A fraction of 70 % of the CFD data base was used for the training of the shallow neural network, 20 % for validation and 10 % for testing. While the full CFD surface solution serves as output, the information at the grid points which are closest to the 36 wind tunnel sensor locations is used as input.

The neural network was set up using the PyTorch, [42]. The size of the second hidden layer was chosen to be $n_2 = 268$, as this corresponds to the dimension of a POD subspace with a relative information content of 99.99 %. For determining the number of neurons for the first hidden layer, the learning rate and the activation function, a study was carried

out using the hyperparameter optimization framework Optuna, [2], with 500 samples generated with the TPESampler. A list of the optimized and fixed hyperparameters is given in Table 5.1.

Parameter	Value
Size of input layer s	36
Size of first hidden layer n_1	185
Size of second hidden layer n_2	268
Size of output layer N	531
Learning rate	0.00248
Activation function	ReLU

Table 5.1: Hyperparameters of the shallow neural network and their corresponding values

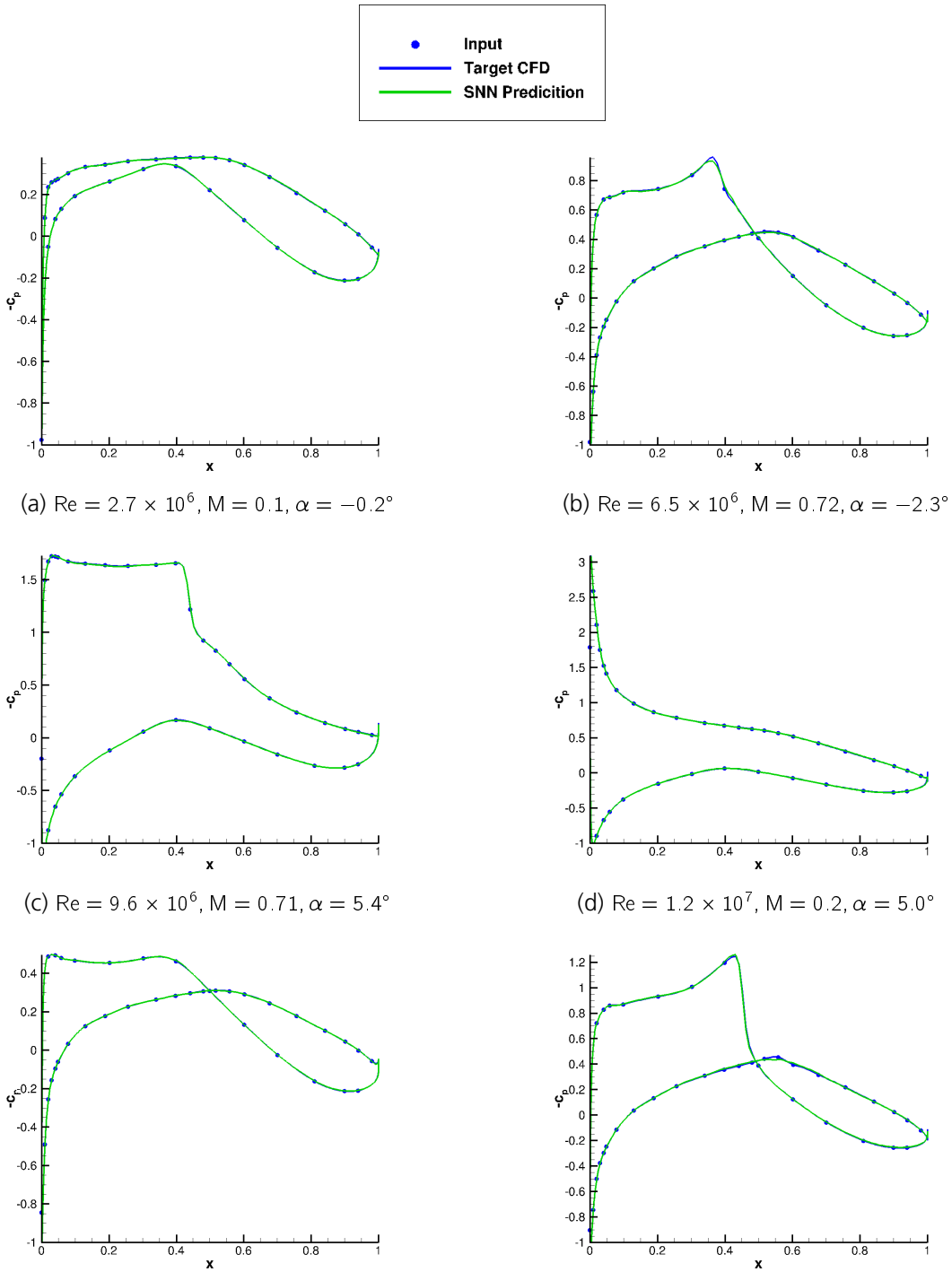
The model was trained for 5000 epochs with batch sizes of 64 training samples and 12 validation samples using the mean squared error to the full surface solution. For comparison with Gappy POD models, a POD subspace of dimension $n_2 = 268$ was computed based on the training and validation data set.

5.3.1 Prediction of CFD data

In a first step, the shallow neural network model, as well as Gappy POD using Ridge regression and Bayesian Gappy POD with GPR have been assessed on the testing data set, i.e. on CFD data which was not included in the training and validation of the models. For all three models, the CFD information at the closest grid points of the 36 wind tunnel sensor locations has served as input. Figure 5.4 shows the results of the neural network for six selected flow conditions in comparison to the target CFD solution. For all investigated data points, an almost perfect match with the target CFD solution is obtained.

For a quantitative comparison of the shallow neural network result with the two Gappy POD methods, the averaged mean squared error on the testing data set with respect to the target CFD solution is visualized in Figure 5.5.

As one can see from the bar plot, the average mean squared error of the shallow neural network prediction is one order of magnitude smaller than for the two Gappy POD methods. Overall, the shallow neural network generalizes very good on the testing data set and clearly outperforms the Gappy POD methods in reconstructing CFD data.



(a) $\mathrm{Re} = 2.7 \times 10^6$, $\mathrm{M} = 0.1$, $\alpha = -0.2°$

(b) $\mathrm{Re} = 6.5 \times 10^6$, $\mathrm{M} = 0.72$, $\alpha = -2.3°$

(c) $\mathrm{Re} = 9.6 \times 10^6$, $\mathrm{M} = 0.71$, $\alpha = 5.4°$

(d) $\mathrm{Re} = 1.2 \times 10^7$, $\mathrm{M} = 0.2$, $\alpha = 5.0°$

(e) $\mathrm{Re} = 1.44 \times 10^7$, $\mathrm{M} = 0.3$, $\alpha = -2.2°$

(f) $\mathrm{Re} = 1.7 \times 10^7$, $\mathrm{M} = 0.745$, $\alpha = -2.9°$

Figure 5.4: Results

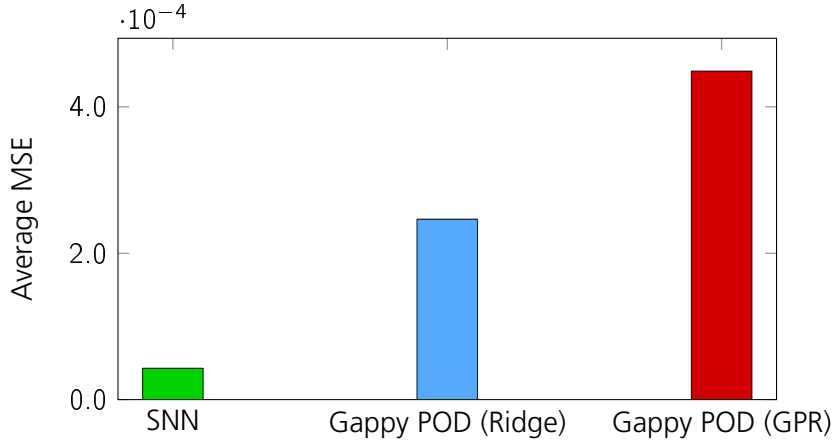


Figure 5.5: Average mean squared error with respect to the target CFD solution for the Shallow Neural Network, Gappy POD with Ridge regression and Gappy POD with GPR evaluated at the testing data set.

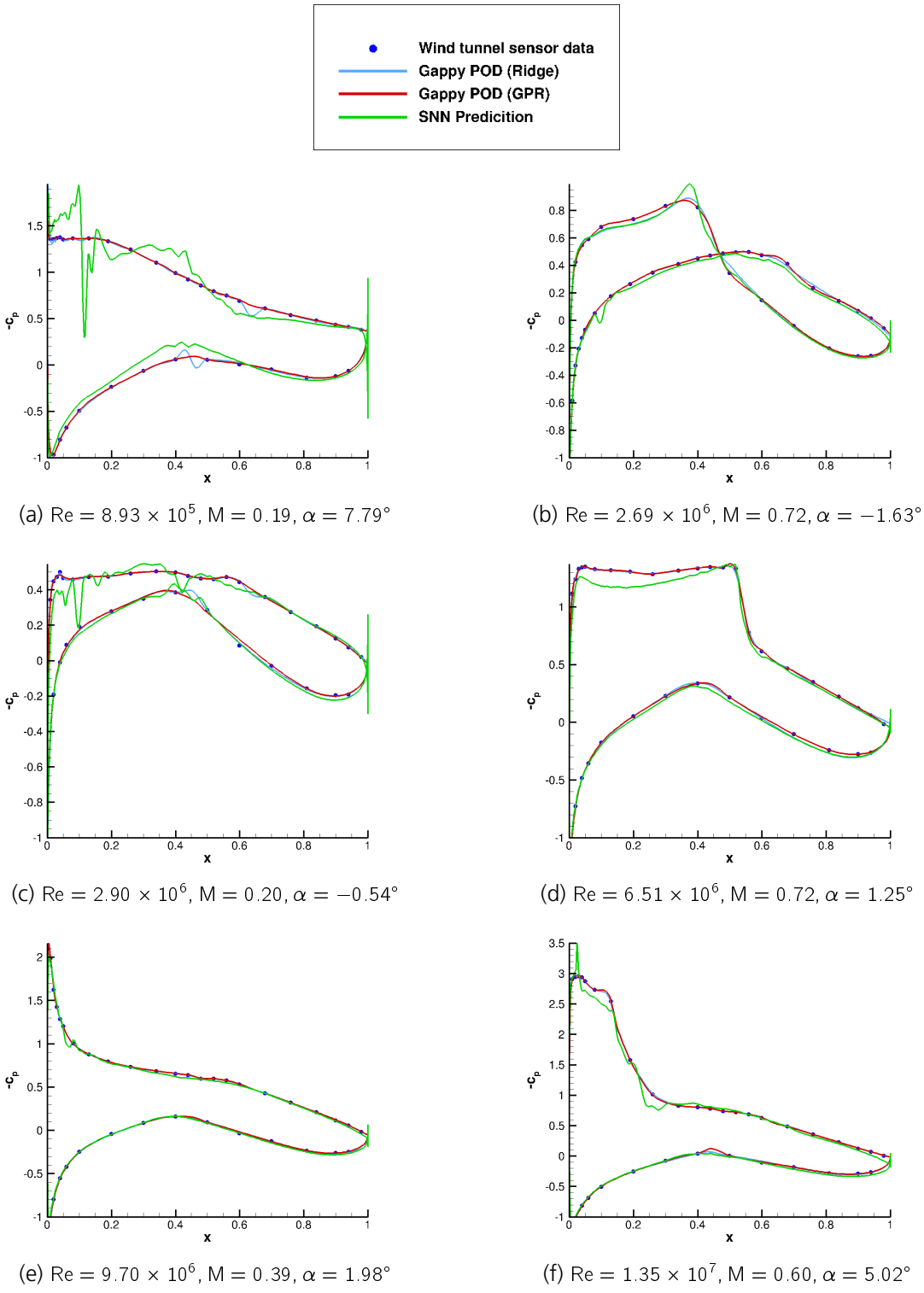
5.3.2 Prediction of wind tunnel data

Next, we go a step further and use the same neural network for the flow reconstruction based on wind tunnel sensor data. Note, that due to the training issue discussed in Section 5.1.2, the experimental data was not part of the training data set and thus, the neural network has never seen this kind of data before. The results for six selected wind tunnel samples ranging from low to high Reynolds numbers are displayed in Figure 5.6.

For the first investigated case of $\mathrm{Re} = 8.93 \times 10^5$, $\mathrm{M} = 0.19$ and $\alpha = 7.79°$ in Figure 5.6a, the shallow neural network prediction shows an unphysical trend with strong fluctuations at the upper surface of the airfoil. On the lower side, the neural network predicts the correct trend but lacks in accurately describing the wind tunnel sensor data. In comparison, a good match with the wind tunnel data is obtained with Gappy POD combined with Ridge regression. Small wiggels occur in between some sensors, especially at the leading edge. Bayesian Gappy POD shows a very smooth trend and accurately matches all wind tunnel sensors.

The shallow neural network shows improved performance for the second investigated test case of $\mathrm{Re} = 2.69 \times 10^6, \mathrm{M} = 0.72, \alpha = -1.63°$ in Figure 5.6b. It gives the right trend with minor discrepancies to the wind tunnel measurements. A smoother trend with a better fit through the wind tunnel data is obtained with both Gappy POD methods.

In Figure 5.6c, for $\mathrm{Re} = 2.90 \times 10^6, \mathrm{M} = 0.20, \alpha = -0.54°$, the neural network prediction again shows strong unphysical fluctuations at the front part of the wing and was not able to match the wind tunnel data accurately. The two Gappy POD methods accurately



(a) Re = 8.93×10^5, M = 0.19, $\alpha = 7.79°$

(b) Re = 2.69×10^6, M = 0.72, $\alpha = -1.63°$

(c) Re = 2.90×10^6, M = 0.20, $\alpha = -0.54°$

(d) Re = 6.51×10^6, M = 0.72, $\alpha = 1.25°$

(e) Re = 9.70×10^6, M = 0.39, $\alpha = 1.98°$

(f) Re = 1.35×10^7, M = 0.60, $\alpha = 5.02°$

Figure 5.6: Results

describe the measurement data whilst having a smooth physical trend.

For $\mathrm{Re} = 6.51 \times 10^6, \mathrm{M} = 0.72, \alpha = 1.25°$ in Fig. 5.6d, all three methods give the correct trend of the flow. The strength and the location of the occurring shock wave is accurately captured by the methods while the shallow neural network predicts the sharpest pressure drop. However, it fails to predict the pressure plateau in front of the shock as indicated by the measurement data, which is well described by both Gappy POD methods.

All three methods accurately match the wind tunnel sensor data for $\mathrm{Re} = 9.70 \times 10^6, \mathrm{M} = 0.39, \alpha = 1.98°$ in Figure 5.6e with the neural network giving small discrepancies at the upper surface of the airfoil.

For the last investigated case in Figure 5.6f of $\mathrm{Re} = 1.35 \times 10^7, \mathrm{M} = 0.60, \alpha = 5.02°$, the shallow neural network gives the right overall trend with unphysical wiggles and larger discrepancies to the wind tunnel sensor data at the front part of the profile. In comparison, a smooth trend and a good agreement with the wind tunnel data is obtained with both Gappy POD methods.

These qualitative observations are also in line with the quantitative comparison of the three methods: The mean squared error of the three different methods with respect to the 36 sensor measurements averaged for all available 7719 wind tunnel data sets is displayed in Figure 5.7.

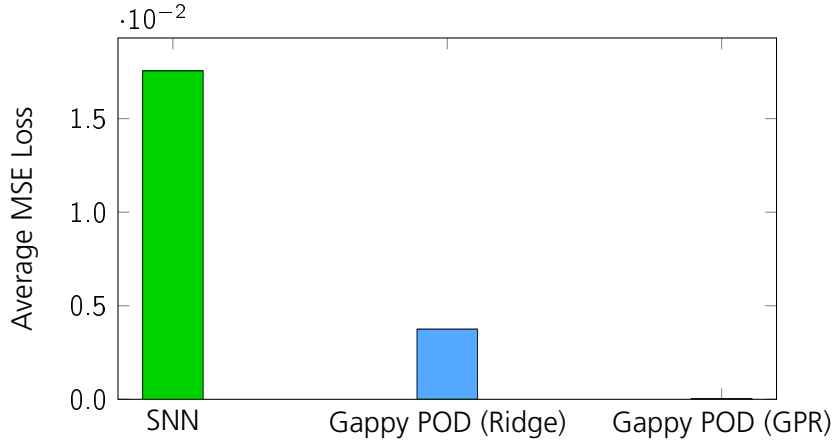


Figure 5.7: Average mean squared error of the three methods with respect to the 36 wind tunnel sensors for the Shallow Neural Network, Gappy POD with Ridge regression and Gappy POD with GPR evaluated for all available wind tunnel data sets.

As indicated by the bar plot, the average mean squared error of the shallow neural network prediction is 1.76×10^{-2} and thus one order of magnitude larger than the one for Gappy POD with Ridge regression of 3.74×10^{-3}. By far the best agreement with the

wind tunnel measurements is obtained with Bayesian Gappy POD with GPR which gives an average mean squared error of 3.01×10^{-5}, which is again two orders of magnitude lower than the one of Gappy POD with Ridge regression.

In summary, for the test case at hand, the shallow neural network trained on CFD data is not able to compete with the two Gappy POD methods for the reconstruction of flow fields based on wind tunnel data.

6 Conclusion and Outlook

In this report, the most important results of the DLR-Airbus Patenschaft on "Development of Future Aerodynamic Data Modelling Methodology" from 2018 until 2022 have been presented. After a short introduction to ordinary, regularized and constrained Gappy POD, new extensions have been derived: Weighted Gappy POD allows to define individual weighting factors for the sensor measurements and is therefore a way to incorporate expert knowledge in data fusion. Bayesian Gappy POD is a natural generalization of Gappy POD as it includes ordinary and regularized Gappy POD as special cases. Moreover, more advanced regression techniques can be considered for the solution of the Gappy POD problem, which makes Bayesian Gappy POD more flexible than the ordinary approach and the problem of linearity can be overcome. Using Bayesian regression techniques, the data fusion result is given in terms of a probability distribution. Mean and standard deviation as well as samples drawn from the probability distribution provide valuable information on the data fusion result. The method was assessed by means of the industrial-relevant XRF1 test case. In comparison to regularized Gappy POD, a better agreement with the wind tunnel measurements can be obtained which is underlined by a root mean squared error which is 47 % smaller than for the state-of-the-art approach. The results serve as an indication for the applicability of the method in the industrial design process. All discussed variants of Gappy POD were made available in SMARTy's latest release.

In the current approach, we take a Bayesian perspective on the regression problem of Gappy POD which allows to include information on the variance of the sensor measurements. Future work is needed towards a fully probabilistic approach which also enables to take information on the uncertainty associated with the CFD data into account. Furthermore, the incorporation of constraints for Bayesian Gappy POD might be a valuable extension – since constrained Gappy POD uses the linearity of the ordinary Gappy POD approach, it cannot be directly transferred to Bayesian Gappy POD. Intelligent sensor placement and techniques for the detection of outliers can help further improving the results.

As an alternative to the Gappy POD methods, a shallow neural network approach proposed in [15] was investigated analytically and evaluated on a 2D airfoil test case. The shallow neural network was able to outperform regularized and Bayesian Gappy POD

when trained and evaluated on CFD data only. However, it showed weak performance in the fusion of numerical and experimental data. A central problem of this approach can be seen in the training issue: While in data fusion, we are aiming to assess the shallow neural network on wind tunnel sensor data, we in general do not have proper wind tunnel training data. The neural network has perfectly learned how to map sparse CFD-based pseudo sensor data onto the corresponding full solution, but has never seen wind tunnel data before it enters the offline phase. Further research could investigate whether results improve if pressure sensitive paint data is included in the training data set. Another idea would be to use the shallow neural network as dimensionality reduction technique in a "Gappy SNN" approach: After the training of the network based on CFD data, its solution space takes the role of the POD subspace in the Gappy POD approach. This has the advantage that the solution space of the neural network is not only learned from the CFD data, as is the case for POD. Because (pseudo) sensor information serve as input to the network, the solution space is created such that the observability of the full solution from the sensor measurements is taken into account which may be beneficial. An investigation of this approach has to be also postponed to future work.

Having a look at the goals initially set for this Patenschaft, almost everything has been achieved: tools for the fusion of numerical and experimental data have been developed and their applicability for industrial-grade problems has been demonstrated. The methods have been shown to be able to provide reliable data across the whole flight envelope, for extreme flight conditions and off-design cases. Modifications and alternative approaches have been analyzed. Mainly due to time constraints and prioritization of both partners, tools for the aerodynamic modeling of aircraft configurations with movables were not discussed within this Patenschaft and therefore need to be investigated in the future.

Bibliography

[1] A. Abbas-Bayoumi and K. Becker. An industrial view on numerical simulation for aircraft aerodynamic design. *Journal of Mathematics in Industry*, 1(10), 2011. doi:10.1186/2190-5983-1-10.

[2] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. In A. Teredesai, editor, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM Digital Library, pages 2623–2631, New York, USA, 2019. Association for Computing Machinery. doi:10.1145/3292500.3330701.

[3] P. Bekemeyer, A. Bertram, D. A. Hines Chaves, M. Dias Ribeiro, A. Garbo, A. Kiener, C. Sabater Campomanes, M. Stradtner, S. Wassing, M. Widhalm, S. Görtz, F. Jäckel, R. Hoppe, and N. Hoffmann. Data-Driven Aerodynamic Modeling Using the DLR SMARTy Toolbox. In *AIAA AVIATION 2022 FORUM (accepted for publication)*, Reston, USA, 2022. American Institute of Aeronautics and Astronautics.

[4] G. Berkooz, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25(1):539–575, 1993. doi:10.1146/annurev.fl.25.010193.002543.

[5] A. Bertram. *Data-driven variable-fidelity reduced order modeling for efficient vehicle shape optimization*. PhD Thesis, TU Braunschweig, Braunschweig, Germany, 2018. doi:10.24355/dbbs.084-201811231243-0.

[6] A. Bertram, P. Bekemeyer, and M. Held. Fusing Distributed Aerodynamic Data Using Bayesian Gappy Proper Orthogonal Decomposition. In *AIAA AVIATION 2021 FORUM*, Reston, USA, 2021. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2021-2602.

[7] A. Bertram, C. Othmer, and R. Zimmermann. Towards Real-time Vehicle Aerodynamic Design via Multi-fidelity Data-driven Reduced Order Modeling. In *2018*

AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Reston, USA, 2018. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2018-0916.

[8] A. Bertram and R. Zimmermann. Theoretical investigations of the new Cokriging method for variable-fidelity surrogate modeling. *Advances in Computational Mathematics*, 44(6):1693–1716, 2018. doi:10.1007/s10444-017-9585-1.

[9] C. M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, USA, corrected at 8th printing 2009 edition, 2009.

[10] J. Blazek. *Computational fluid dynamics: Principles and applications*. Butterworth Heinemann, Amsterdam, Netherlands, third edition edition, 2015.

[11] M. D. Buhmann. *Radial Basis Functions*. Cambridge University Press, Cambridge, UK, 2003. doi:10.1017/CBO9780511543241.

[12] T. Bui-Thanh, M. Damodaran, and K. Willcox. Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics. In *21st AIAA Applied Aerodynamics Conference*, page 519, Reston, USA, 2003. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2003-4213.

[13] T. Bui-Thanh, M. Damodaran, and K. E. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42(8):1505–1516, 2004. doi:10.2514/1.2159.

[14] P. Cook, M. Firmin, and M. McDonald. *Aerofoil RAE 2822: pressure distributions, and boundary layer and wake measurements*. RAE, 1977.

[15] N. B. Erichson, L. Mathelin, Z. Yao, S. L. Brunton, M. W. Mahoney, and J. N. Kutz. Shallow neural networks for fluid flow reconstruction with limited sensors. *Proceedings. Mathematical, physical, and engineering sciences*, 476(2238):20200097, 2020. doi:10.1098/rspa.2020.0097.

[16] R. Everson and L. Sirovich. Karhunen-Loève procedure for gappy data. *Journal of the Optical Society of America A*, 12(8):1657, 1995. doi:10.1364/JOSAA.12.001657.

[17] A. Feldstein, D. Lazzara, N. Princen, and K. Willcox. Multifidelity Data Fusion: Application to Blended-Wing-Body Multidisciplinary Analysis Under Uncertainty. *AIAA Journal*, 58(2):889–906, 2020. doi:10.2514/1.J058388.

[18] A. I. J. Forrester, N. W. Bressloff, and A. J. Keane. Optimization using surrogate models and partially converged computational fluid dynamics simulations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 462(2071):2177–2204, 2006. doi:10.1098/rspa.2006.1679.

[19] A. I. J. Forrester and A. J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1-3):50–79, 2009. doi:10.1016/j.paerosci.2008.11.001.

[20] A. I. J. Forrester, A. Sóbester, and A. J. Keane. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088):3251–3269, 2007. doi:10.1098/rspa.2007.1900.

[21] A. I. J. Forrester, A. Sóbester, and A. J. Keane. *Engineering design via surrogate modelling: A practical guide*. Wiley, Hoboken, USA, 2008.

[22] T. Franz and M. Held. Data fusion of cfd solutions and experimental aerodynamic data. In DLR and ONERA, editors, *ODAS 2017*, 2017. URL: https://elib.dlr.de/114707/.

[23] G. H. Golub and C. F. van Loan. *Matrix computations*. Johns Hopkins studies in mathematical sciences. Johns Hopkins Univ. Press, Baltimore, USA, 4. edition, 2013.

[24] H. Gunes, S. Sirisup, and G. E. Karniadakis. Gappy data: To Krig or not to Krig? *Journal of Computational Physics*, 212(1):358–382, 2006. doi:10.1016/j.jcp.2005.06.023.

[25] Z.-H. Han and S. Görtz. Hierarchical Kriging Model for Variable-Fidelity Surrogate Modeling. *AIAA Journal*, 50(9):1885–1896, 2012. doi:10.2514/1.J051354.

[26] Z.-H. Han, S. Görtz, and R. Zimmermann. Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerospace Science and Technology*, 25(1):177–189, 2013. doi:10.1016/j.ast.2012.01.006.

[27] Z.-H. Han, Zimmermann, and S. Görtz. Alternative Cokriging Method for Variable-Fidelity Surrogate Modeling. *AIAA Journal*, 50(5):1205–1210, 2012. doi:10.2514/1.J051243.

[28] Z.-H. Han, R. Zimmermann, and S. Görtz. A New Cokriging Method for Variable-Fidelity Surrogate Modeling of Aerodynamic Data. In *48th AIAA Aerospace Sciences*

Meetings, Reston, USA, 2010. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2010-1225.

[29] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. Springer series in statistics. Springer, New York, USA, second edition, corrected at 12th printing 2017 edition, 2017.

[30] A. J. Keane. Wing Optimization Using Design of Experiment, Response Surface, and Data Fusion Methods. *Journal of Aircraft*, 40(4):741–750, 2003. doi:10.2514/2.3153.

[31] M. C. Kennedy and A. O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000. doi:10.1093/biomet/87.1.1.

[32] Y. Kuya, K. Takeda, X. Zhang, and A. I. J. Forrester. Multifidelity Surrogate Modeling of Experimental and Computational Aerodynamic Data Sets. *AIAA Journal*, 49(2):289–298, 2011. doi:10.2514/1.J050384.

[33] H. V. Ly and H. T. Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and Computer Modelling*, 33(1-3):223–236, 2001. doi:10.1016/S0895-7177(00)00240-5.

[34] M. Mifsud, A. Vendl, L.-U. Hansen, and S. Görtz. Fusing wind-tunnel measurements and cfd data using constrained gappy proper orthogonal decomposition. *Aerospace Science and Technology*, 86:312–326, 2019. doi:10.1016/j.ast.2018.12.036.

[35] K. P. Murphy. *Machine learning: A probabilistic perspective*. Adaptive computation and machine learning series. The MIT Press, Cambridge, Massachusetts and London, England, 2012.

[36] R. Pinnau. Model reduction via proper orthogonal decomposition. In H.-G. Bock, F. de Hoog, A. Friedman, A. Gupta, H. Neunzert, W. R. Pulleyblank, T. Rusten, F. Santosa, A.-K. Tornberg, L. L. Bonilla, R. Mattheij, O. Scherzer, W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, editors, *Model Order Reduction: Theory, Research Aspects and Applications*, volume 13 of *Mathematics in Industry*, pages 95–109. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi:10.1007/978-3-540-78841-6_5.

[37] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 3. print edition, 2008.

[38] S. A. Renganathan, K. Harada, and D. N. Mavris. Aerodynamic Data Fusion Toward the Digital Twin Paradigm. *AIAA Journal*, 58(9):3902–3918, 2020. doi:10.2514/1.J059203.

[39] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and Analysis of Computer Experiments. *Statistical Science*, 4(4), 1989. doi:10.1214/ss/1177012413.

[40] T. J. Santner, B. Williams, and W. Notz. *Design and analysis of computer experiments*. Springer series in statistics. Springer, New York, 2003. URL: http://www.loc.gov/catdir/enhancements/fy0817/2003045444-d.html.

[41] D. Schwamborn, T. Gerhold, and R. Heinrich. The DLR TAU-Code: Recent Applications in Research and Industry. In P. Wesseling, editor, *ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics*, [s. l.], 2006. Delft University of Technology.

[42] The PyTorch Community. Pytorch. https://pytorch.org/, Dezember 2021.

[43] A. N. Tikhonov and V. J. Arsenin. *Solutions of ill-posed problems*. Scripta series in mathematics. Winston, Washington, D.C., 1977.

[44] D. Venturi and G. E. Karniadakis. Gappy data and reconstruction procedures for flow past a cylinder. *Journal of Fluid Mechanics*, 519:315–336, 2004. doi:10.1017/S0022112004001338.

[45] K. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & Fluids*, 35(2):208–226, 2006. doi:10.1016/j.compfluid.2004.11.006.

DLR-IB-AS-BS-2022-111

Outcomes of the DLR-Airbus Patenschaft "Development of Future Aerodynamic Data Modelling Methodology" 2018–2022

Anna Bertram