

# Automated Process for Robust Airfoil Design-Optimization Incorporating Critical Eigenmode Identification and Production-Tolerance Evaluation

Von der Fakultät für Maschinenbau, Elektro- und Energiesysteme  
der Brandenburgischen Technischen Universität Cottbus-Senftenberg  
zur Erlangung des akademischen Grades eines Doktor-Ingenieurs genehmigte Dissertation

vorgelegt von

**Dipl.-Ing. Ivo Martin**

geboren am 16.07.1983 in Rüdersdorf

Vorsitzender: Prof. Dr.-Ing. Klaus Höschler  
Gutachter: Prof. Dr.-Ing. habil. Dieter Bestle  
Gutachter: Prof. Dr.-Ing. Arnold Kühhorn

Tag der mündlichen Prüfung: 05.10.2018



# Acknowledgment

The research contained by this thesis was conducted as part of the joint research program COOREFLEX-turbo in the frame of AG Turbo. The work was supported by the Bundesministerium für Wirtschaft und Technologie (BMWi) as per resolution of the German Federal Parliament under grant number 03ET7021J. I gratefully acknowledge AG Turbo and General Electric Switzerland, especially Michael Lötzerich, for their support and permission to publish this thesis. The responsibility for its content lies solely with me.

First of all, I would like to give special thanks to my thesis advisor and reviewer, Prof. Dieter Bestle for the opportunity to write this thesis, and his significant support for my research. His numerous inputs and our productive discussions are highly appreciated. My gratitude also goes to Prof. Kühorn who took part in the board of examiners and reviewed the present thesis as well as Prof. Höschler who agreed to be the chairman of the examination board of this thesis.

Furthermore, I am thankful to all members of the Chair of Engineering Mechanics and Vehicle Dynamics for an enjoyable time both during working hours as well as outside of work. Special gratitude goes to Dr. Michael Lockan for always being a source of valuable inspiration and to my project partner Lennard Hartwig whose knowledge and enthusiasm helped to bring the project to fruition. Another friend and colleague who I would like to thank is Dr. Christian Janke, as he brought my attention to this project, as well as Dr. Tommy Grunert who proof read this thesis and whom I could always rely on for exciting discussions during our weekly dinners.

Finally, I am most thankful to my parents Ulrike and Gerald for their unwavering support and acceptance during this time, to my brother Robin for helping out at several occasions, and to my wife Lillian who agreed to my wish to do a PhD and took the increased effort to look after our beautiful daughter Aurelia, who herself is a huge source of motivation to me.



# Abstract

For many countries, gasturbine technology is one of the key technologies for the reduction of climate-damaging pollutant emissions. The profitability of such facilities, however, is highly dependent on the price for the utilized fossil fuel, which is why there is a constant need for increased efficiency. The potential of increasing the efficiency of the individual components is basically limited by factors which will reduce operating life. The goal of this thesis is to develop methods for improved automated structural design optimization, which shall be developed on the basis of compressor airfoils. Special attention is payed to avoid the excitation of failure critical eigenmodes by detecting them automatically. This is achieved by introducing a method based on self-organizing neural networks which enables the projection of eigenmodes of arbitrary airfoil geometries onto standard surfaces, thereby making them comparable. Another neural network is applied to identify eigenmodes which have been defined as critical for operating life. The failure rate of such classifiers is significantly reduced by introducing a newly developed initialization method based on principle components. A structural optimization is set up which shifts the eigenfrequency bands of critical modes in such a way that the risk of resonance with engine orders is minimized. In order to ensure practical relevance of optimization results, the structural optimization is coupled with an aerodynamic optimization in a combined process. Conformity between the loaded hot-geometry utilized by the aerodynamic design assessment and the unloaded cold-geometry utilized by the structural design assessment is ensured by using loaded-to-unloaded geometry transformation. Therefor an innovative method is introduced which, other than the established time-consuming iterative approach, uses negative density for a direct transformation taking only a few seconds, hence, making it applicable to optimization. Additionally, in order for the optimal designs to be robust against manufacturing variations, a method is developed which allows to assess the maximum production tolerance of a design from which onwards possible design variations are likely to violate design constraints. In contrast to the usually applied failure rate, the production tolerance is a valid requirement for suppliers w.r.t. expensive parts produced in low-quantity, and therefore is a more suitable optimization objective.



# Kurzfassung

Für viele Länder ist die Gasturbine eine der Schlüsseltechnologien, um ihren klimaschädlichen Schadstoffausstoß zu reduzieren. Die Profitabilität solcher Anlagen hängt jedoch wesentlich von den Preisen des jeweiligen fossilen Brennstoffs ab, weshalb ein stetiger Entwicklungsdruck zu immer effizienteren Anlagen zu verzeichnen ist. Die mögliche Effizienzsteigerung der einzelnen Komponenten wird jedoch im Wesentlichen durch lebensdauerreduzierende Faktoren beschränkt. Daher ist das Ziel dieser Arbeit, Methoden zur besseren automatisierten strukturdynamischen Bauteiloptimierung am praktischen Beispiel von Verdichterschaufeln zu entwickeln. Insbesondere liegt der Fokus darauf, die Erregung kritischer Eigenschwingungen zu vermeiden. Zu diesem Zweck müssen vorher als kritisch definierte Eigenformen automatisiert erkannt werden. Dies geschieht mit Hilfe einer auf Basis von selbstorganisierenden neuronalen Netzen entwickelten Projektionsmethode, welche es ermöglicht, Schwingungseigenformen beliebiger Schaufelgeometrien auf eine Standardfläche zu projizieren, um eine Vergleichsgrundlage zu erhalten. Mit Hilfe eines weiteren neuronalen Netzwerks wird geprüft, ob eine projizierte Eigenform einer der zuvor als kritisch definierten Eigenformen entspricht. Dieser Klassifizierer wurde durch eine neu entwickelte Initialisierungsmethode basierend auf Principle-Component-Analysis in seiner Fehlerrate entscheidend verbessert. In einen strukturellen Optimierungsprozess eingebettet führen diese Methoden zu einer Minimierung des Risikos von Bauteilresonanz zwischen Eigenschwingungen und Anregungen. Um den praktischen Nutzen der Optimierungsergebnisse sicherzustellen, wird der strukturelle Optimierungsprozess mit einem aerodynamischen gekoppelt. Dabei wird die Konsistenz zwischen den auf der belasteten heißen Geometrie beruhenden aerodynamischen und den ausgehend von der kalten unbelasteten Geometrie berechneten strukturmechanischen Ergebnissen durch eine innovative Methode erreicht, welche eine negative Bauteildichte zur direkten Transformation in nur wenigen Sekunden nutzt und somit im Gegensatz zur bisherigen iterativen Methode für Optimierungsprozesse praktisch anwendbar ist. Damit die optimalen Entwürfe aus dem Optimierungsprozess zusätzlich robust gegen stochastische Produktionsvariationen sind, wurde eine Methode zur Bestimmung der Produktionstoleranz eines Entwurfs entwickelt, ab der mögliche Entwürfe Versagenskriterien verletzen würden. Die Produktionstoleranz ist eine valide Anforderung an Zulieferer für teure und in geringer Stückzahl produzierte Bauteile und somit, als Optimierungskriterium formuliert, aussagekräftiger als die üblicherweise verwendete Ausfallrate.





# Contents

<b>Nomenclature</b>	<b>III</b>
<b>Acronyms</b>	<b>IX</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Compressor Design Process . . . . .	3
1.2 Current State of Robust and Structural Design Optimization . . . . .	5
1.3 Motivation and Outline of Thesis . . . . .	8
<b>2 Multidisciplinary Design-Optimization Process for Axial Compressors</b>	<b>11</b>
2.1 Optimization Process . . . . .	11
2.2 Geometry and Parameter Model . . . . .	14
2.3 Deterministic Optimization Problem and Constraints . . . . .	16
2.4 Robust Design Optimization . . . . .	24
<b>3 Structural Assessment of Compressor Airfoils Using the Finite-Element Method</b>	<b>29</b>
3.1 Fundamental Equations . . . . .	29
3.2 Finite-Element Model . . . . .	32
3.2.1 Topology and Mesh . . . . .	32
3.2.2 Loads and Boundary Conditions . . . . .	33
3.3 Loaded-to-Unloaded Transformation . . . . .	35
3.4 Influence of Fixation Uncertainties onto Eigenfrequency Bands . . . . .	37
<b>4 Projection of Arbitrary Airfoil Geometries onto a Rectangular Map</b>	<b>43</b>
4.1 Mapping of Airfoil Data onto Rectangular Map . . . . .	44
4.2 Training of Neural Network . . . . .	48
<b>5 Normalization and Dimension Reduction of Eigenmode Data</b>	<b>53</b>
5.1 Alignment and Normalization of Eigenmode Data . . . . .	53
5.2 Procedures for Dimension Reduction of Eigenmode Data . . . . .	61
5.3 Assessment of Dimension Reduction Procedures . . . . .	65

<b>6</b>	<b>Classification of Eigenmodes and Assignment of Frequency Bands</b>	<b>73</b>
6.1	Classification of Eigenmode Shapes . . . . .	73
6.1.1	Nearest-Neighbors Algorithm . . . . .	74
6.1.2	Support-Vector Machine . . . . .	75
6.1.3	Feed-Forward Neural Network . . . . .	77
6.2	Assessment of Classification Performance . . . . .	86
6.3	Procedure for Assigning Eigenmode Bands . . . . .	93
<b>7</b>	<b>Structural Optimization Results for a Rotor Blade</b>	<b>97</b>
7.1	Parameter Fitting of an Actual Design . . . . .	97
7.2	Results of Deterministic Optimization . . . . .	99
7.3	Robust Design Results . . . . .	104
<b>8</b>	<b>Conclusions and Outlook</b>	<b>109</b>
	<b>Appendix A Additional Figures</b>	<b>113</b>
	<b>Appendix B Frequency Veering</b>	<b>119</b>
	<b>Appendix C Mathematics of Chosen Methods</b>	<b>123</b>
C.1	Principal Component Analysis . . . . .	123
C.1.1	Linear PCA . . . . .	123
C.1.2	Nonlinear PCA with Kernel-trick . . . . .	125
C.2	Multiple Discriminant Analysis . . . . .	129
C.2.1	Fisher-Discriminant Analysis . . . . .	129
C.2.2	Linear MDA . . . . .	130
C.2.3	Nonlinear MDA . . . . .	132
C.3	Modal-Assurance Criterion . . . . .	135
C.4	Support-Vector Machines . . . . .	137
C.5	Training of a Three-Layer Feed-Forward Neural Network . . . . .	141
	<b>List of Figures</b>	<b>145</b>
	<b>List of Tables</b>	<b>149</b>
	<b>Bibliography</b>	<b>151</b>

# Nomenclature

## Roman Letters

$b$	offset between hyperplane and origin
$c$	section-wise chord length of airfoil, eigenmode class
$C$	constraint penalty
$d$	weight-function, reduced dimensionality
$d_{st}$	distance metric
$D$	unreduced dimensionality
$E$	Young's modulus
$f$	function
$f^A$	engine order frequency [Hz]
$f^E$	eigenfrequency [Hz]
$F$	dimensionality of feature space
$F_C$	centrifugal force
$F_R$	failure rate
$h$	neighborhood function
$h_F$	fillet height
$H$	blade height
$i, j, k, m, n$	indices
$I, J$	number of nodes in a rectangular mesh
$k$	spring rate
$L$	Lagrange function
$L_{FL1}, L_{FL2}$	first and second flutter limit
$L_{FR}$	limit of dynamic stress ratio
$L_S$	limit of distance between sample point and sampling bound
$L_D$	dual Lagrange function
$L_R$	learning rate
$L_R^+, L_R^-$	increase and reduction factor of learning rate
$m$	point mass

---

$n$	shaft speed
$n_F$	counter of failed tries
$N$	number of elements in a general data set
$N_{aero}$	number of aerodynamic constraints
$N_A$	number of samples failing design constraints
$N_B$	maximum number of training epochs
$N_C$	number of clusters (classes)
$N_E$	number of finite elements
$N_F$	maximum number of failed tries
$N^I$	number of circumferential installation of a specific source
$N_M$	number of eigenmodes in reference set
$N_N$	number of nodes
$N_R$	number of compressor rows
$N_{struct}$	number of structural constraints
$N_S, N_T$	sample size
$N^r$	number of misclassifications on row $r$
$p$	design parameter
$p_{SS}, p_{PS}$	static pressure field on the suction and pressure side
$P^{95}$	95% percentile
$P_{\Delta p_t}^{95}$	95% percentile of $\Delta p_t$
$P_{\omega_{OD}}^{95}$	95% percentile of $\omega_{OD}$
$r$	radial compressor coordinate, compressor-row number
$r^{Hub}, r^{Tip}$	radius of the compressor hub and casing
$R$	radius
$R_F$	fillet radius at the blade
$R_{p02}$	yield strength
$s$	silhouette value
$\Delta s_{\perp}$	symmetric production tolerance
$S$	surface of airfoil or body
$St$	Strouhal-number
$t$	section-wise local airfoil thickness, time
$v^{rel}$	relative flow velocity
$V, V_B$	volume, volume of body
$w$	weight or penalty factor
$w^A, w^E$	penalty factor for engine orders and eigenmode shapes
$x, y, z$	Cartesian coordinates
$x_S$	axial shift

**Greek Letters**

$\alpha$	free parameter, Lagrange or KKT multiplier
$\beta$	chamber-line angle
$\beta_L, \beta_T$	leading- and trailing-edge-metal angle
$\gamma$	Lagrange or KKT multiplier
$\varepsilon$	error, deviation
$\zeta$	margin
$\eta$	percentage of correct classification
$\theta$	circumferential angle, phase shift
$\theta_S$	shift in circumferential direction
$\lambda$	eigenvalue
$\nu$	Poisson's ratio
$\xi$	slack variable
$\rho$	density
$\sigma$	variance
$\sigma_v, \sigma_m, \sigma_a$	von Mises stress, its mean and dynamic part
$\phi$	phase shift
$\omega$	oscillation frequency, eigenfrequency, angular velocity
$\Omega$	tolerance hull around airfoil surface

**Vectors and Matrices**

$\mathbf{b}$	vector of inequality constraints
$\mathbf{B}$	matrix of eigenvectors $\boldsymbol{\beta}^i$
$\mathbf{B}^L, \mathbf{B}^{NL}$	linear and nonlinear strain-displacement-transformation matrices
$\mathbf{C}$	stress-strain tensor
$\mathbf{D}$	diagonal matrix resulting from singular value decomposition
$\mathbf{e}_i$	$i^{\text{th}}$ unit vector
$\mathbf{f}, \mathbf{g}, \mathbf{h}$	vector functions
$\mathbf{f}^C, \mathbf{f}^S, \mathbf{f}^V$	concentrated, surface, and volume force vector
$\mathbf{F}$	matrix of high dimensionally transformed samples
$\mathbf{F}^C, \mathbf{F}^S, \mathbf{F}^V$	sum of concentrated, surface, and volume force vectors
$\mathbf{H}$	displacement-interpolation matrix
$\mathbf{I}$	identity matrix
$\mathbf{K}$	kernel matrix
$\mathbf{K}^L, \mathbf{K}^{NL}$	linear and nonlinear stiffness matrices
$\mathbf{L}$	lower triangular matrix
$\mathbf{m}, \mathbf{m}_c$	estimated mean vector

---

<b>M</b>	mass matrix
<b>n</b>	normal vector
<b>p</b>	vector of design variables
<b>p<sup>Δ</sup></b>	sampled design vector
<b>s<sub>c</sub></b>	within-cluster scatter
<b>t<sup>HL</sup>, t<sup>OL</sup></b>	net activation vector of hidden and output layer
<b>u<sup>⊥</sup>, u<sup>→</sup>, u<sup>↑</sup></b>	projected displacement-field vectors
<b>U</b>	combined vector of displacements of all FE-nodes
<b>v</b>	eigenvector
<b>V</b>	matrix of eigenvectors <b>v<sup>i</sup></b>
<b>w<sup>HL</sup>, w<sup>OL</sup></b>	hidden- and output-layer-weight vector
<b>W<sup>HL</sup></b>	hidden-layer-weight matrix
<b>x</b>	general coordinate vector
<b>β</b>	eigenvector of subspace eigenproblem
<b>Λ</b>	diagonal matrix of eigenvalues
<b>Λ<sup>+</sup></b>	diagonal matrix of inverse eigenvalues different from zero
<b>μ</b>	true mean vector
<b>σ</b>	real stress tensor
<b>Σ</b>	covariance matrix
<b>Σ<sub>B</sub>, Σ<sub>C</sub>, Σ<sub>W</sub></b>	between-class, class, and within-class scatter matrix
<b>τ</b>	Cauchy-stress tensor
<b>φ</b>	structural eigenvector or vector based on structural eigenvector data
<b>Subscripts</b>	
<b>0</b>	reference or initial configuration
<b>L</b>	leading edge
<b>R</b>	matrix or vector of reduced size
<b>T</b>	trailing edge
<b>Superscripts</b>	
<b>*</b> , <b>o</b>	case-dependent markers
<b>C</b> , <b>H</b>	cold and hot configuration
<b>L</b>	lower stiffness bound
<b>max</b>	maximum value
<b>nom</b>	nominal value
<b>T</b>	transposed
<b>U</b>	upper stiffness bound

## Symbols and Operators

$\dot{\bullet}, \ddot{\bullet}$	first and second time derivative
$\bar{\bullet}$	average, mean value or complex conjugate of a complex number
$\hat{\bullet}$	transformation of matrix or vector into higher dimension
$\delta_{ij}$	Kronecker delta (= 1 for $i = j$ , = 0 else)
$\tilde{\bullet}$	normalized value
$\Delta$	difference
$\bullet'$	projection of a vector
$\pi$	permutation
$\check{\bullet}, \hat{\bullet}$	lower and upper limits
<i>conv</i>	filters a set of elements for those that form the convex hull
<i>diag</i>	diagonal of matrix
<i>dist</i>	distance between a vector and the closest member in set of vectors
$\overline{dist}$	average distance
<i>exp</i>	exponential function
<i>E</i>	expected value (mean value operator)
$E^S$	cross entropy
<i>MDA, kMDA</i>	MDA and kMDA operator applied to a dataset
<i>PCA, kPCA</i>	PCA and kPCA operator applied to a dataset
<i>Re</i>	real part of a complex number
<b>Sets</b>	
$\mathcal{D}$	set of parameters vectors (designs)
$\mathcal{H}$	set of points on hyperplane
$\mathcal{P}$	admissible design space
$\mathcal{T}$	set of design variations within a specific production tolerance
$\mathcal{V}$	set of vectors
$\mathcal{V}_T, \mathcal{V}_V$	training and validation set





# Acronyms

2D	2-Dimensional
3D	3-Dimensional
blisk	Bladed Integrated Disk
BPNN	Back-Propagating-Neural Network
CAD	Computer-Aided Design
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
CPV	Characteristic Pixel Vector
DoE	Design of Experiments
DoF	Degree(s) of Freedom
FDA	Fisher-Discriminant Analysis
FE, FEA, FEM	Finite Elements, Finite-Element Analysis and Method
FFNN	Feed-Forward-Neural Network
HCF	High-Cycle Fatigue
HL	Hidden Layer
IGV	Inlet Guide Vane
IL	Input Layer
KKT	Karush-Kuhn-Tucker
kMDA	Kernel-based Multiple Discriminant Analysis
kPCA	Kernel-based Principle-Component Analysis
kPDA	kMDA applied to kPCA pojection of data
LMCA	Linear Modal-Assurance Criterion
MCA	Modal-Assurance Criterion
MDA	Multiple Discriminant Analysis
MRF	Mode-Resonance Factor
OL	Output Layer
PCA	Principle Component Analysis
Q3D	Quasi 3-Dimensional

RSM	Response Surface Model
SOM	Self-Organizing Map
TLFRA	Time-Linearized-Forced-Response Analysis

# 1 Introduction

For electric energy supply in modern societies, the gas turbine is one of the key technologies to meet the increasingly restrictive limitations on exhaust emissions. Due to its ability to operate with a large variety of combustibles, it is a technology which enables the continuous usage of fossil fuels and is able to fill the gap between repelling nuclear energy and the favored renewable energies. In particular, coal as the cheapest and most available fossil fuel will continue to be the main energy source for many third world countries which often cover over 60% of their energy demand with coal (World Energy Council (2016)). Since the introduction of pressurized fluidized bed combustion (PFBC; Mudd (1995)) and its permanent enhancement, coal can be utilized in a much cleaner way with gas turbines than in conventional plants with boilers and steam turbines, which makes the gas turbine to be an important future technology. But also countries deploying renewable energies are dependent on gas turbine technology, because small but flexible power plants are required to compensate the erratic availability of sun, water, and wind in order to ensure stable power networks.

Designing gas turbines is a complex task, since objectives and constraints of multiple disciplines must be considered. For large scale stationary gas turbines installed to cover the main network demand, this task is becoming increasingly challenging due to the need to exceedingly reduce fuel consumption, maintenance effort (reduction of running costs and downtime due to maintenance), and environmental pollution. Traditionally, the development of gas turbines is subdivided into the development of task specific components such as compressor, combustor, and turbine, see Fig.1.1. Clearly, the compressor takes most of the length and parts of the gas-turbine, owing to the circumstance that the pressure rise counteracts the airflow which is therefore exposed to boundary-layer separation, i.e., the risk of surge which limits the possible diffusion (pressure rise) per row (de Haller (1953) and Lieblein et al. (1953)). The overall pressure rise of the compressor is a requirement which is defined based on the power demand and the thermodynamic Brayton cycle (Russel D'A (1973)). The limited diffusion per row and the demand for aerodynamically most efficient designs has lead to slender airfoil structures which are conflicting with the need for structurally more durable designs requiring less maintenance. The multidisciplinary design requirements the compressor design has to account for enforce time-consuming iterations between different design tools and also within the discipline-specific design tools. Therefore, engineers are more and more troubled to explore better designs, where automation will become a mandatory necessity.

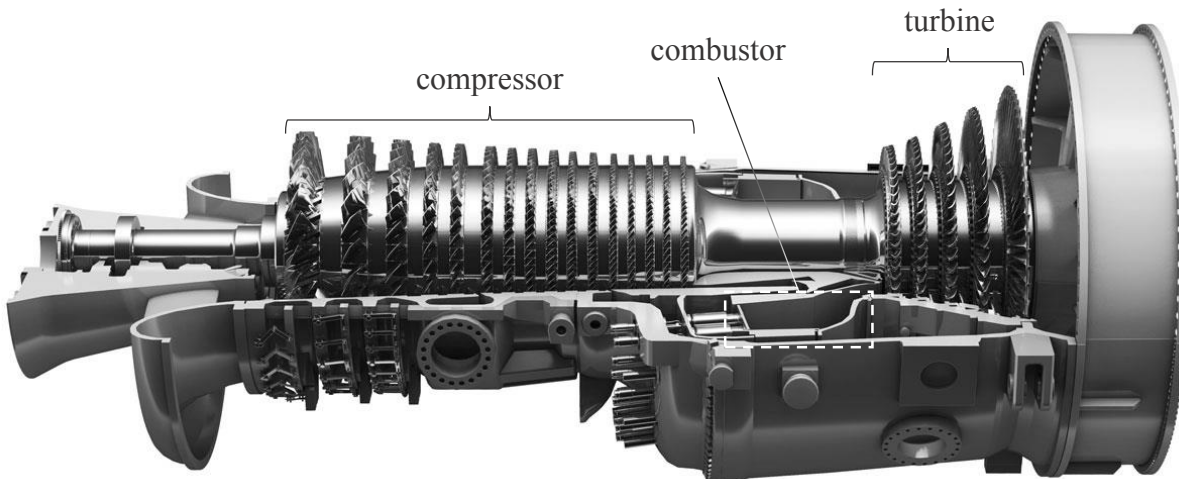


Figure 1.1: Compressor, combustor, and turbine of ALSTOM gas turbine GT13E2 (printed by courtesy of General Electric Switzerland)

The subject of process automation becomes additionally pressing when design methodology changes from a deterministic to a probabilistic approach where uncertainties in running, environmental, and installation conditions as well as in production deviations require statistically relevant numbers of design variations in order to capture the stochastic behavior of design properties in service. As described by Flassig (2011), the goal of the probabilistic methodology is to find robust designs, i.e., designs which have little variance in their performance and little risk of failing design constraints w.r.t. design-parameter uncertainties and varying operating conditions.

The clear need for optimization processes which incorporate multidisciplinary automated design processes is more and more accounted for by industry and science. However, still industry struggles with the broad application of such processes since their complexity and high expectations in the results (design requirements for automated design processes are often defined more challenging, because they don't soften as a result of renegotiation between different departments), make computations expensive and time-consuming. The problem is that acquiring engineering knowledge and combining existing design tools in order to create automated design processes does not necessarily lead to the exploitation of the whole potential of optimization processes, because either the applied automated design processes are too slow, the architecture of the optimization process is poor, or optimization objectives and constraints do not sufficiently consider the specific need of the optimization algorithms being applied. In fact, further research and development has to consider new methods, architectures, and criteria which accommodate the potential of process automation.

As one of the key components in an airfoil-design process, structural design assessment is of special importance, since no part will be produced if it does not fulfill the mandatory criteria. Additionally, structural design assessment is especially challenging because ensuring a design's reliability throughout its lifetime often requires complex modifications of the geometry in order to achieve the desired structural dynamic characteristic. The

goal of this thesis is to introduce methods which allow to automatically improve the structural characteristics of airfoils in a most efficient manner and to ensure structural integrity throughout the lifetime w.r.t. uncertainties which may harm design robustness. As those methods have to prove their value within an optimization process, which can only provide reasonable results by accounting for the conflict between aerodynamic and structural design goals, this thesis will validate the methods using a very effective coupled aerodynamic and structural design-optimization process. Since the aerodynamic part of the design process is a result of another work package within the underlying joint project between ALSTOM and Brandenburg University of Technology Cottbus-Senftenberg, this part will only be explained to such an extent as it concerns the understanding of the structural methods and optimization results.

In order to introduce subject and structure of the thesis, firstly, the compressor-design process of the industrial partner ALSTOM will be explained. Thereafter a literature survey will lead through former contributions to automation of design processes and structural assessments where key drawbacks will be highlighted to motivate the key contributions of this thesis.

## 1.1 Compressor Design Process

Compressor design has to account for various criteria from different disciplines, which is why the final design will result from an iterative interaction between different design tools. However, as the design goal is to most efficiently compress air at a given ratio defined by the performance requirement for the whole gas-turbine (Cumpsty (2004)), the preliminary design phase is purely driven by aerodynamic considerations and constraints only, whereas structural integrity is typically not considered. Based on the requirements for the pressure ratio, efficiency, mass-flow, and surge-margin, heuristic calculations and data from familiar, previously designed compressors are used to calculate the annulus contour of the compressor and average performance along its meanline (central meridian surface). The definition of the meanline aerodynamics involves the calculation of geometric parameters such as stage segmentation, number of blades or vanes per row, and solidity, but also aerodynamic conditions like meridional temperature distribution, loading of the stages (stage-pressure ratios), and velocity triangles. The latter are the boundary conditions or average target conditions for the subsequent definition of the radial flow distribution. Therefore, the annulus contour between casing and hub is radially subdivided into 21 equidistant meridional streamtube surfaces also called S1 surfaces, Wu (1952). Taking into consideration the flow displacement by the airfoils and radially different loading distributions for blades and vanes, the throughflow is calculated for each of those S1 surfaces using 2D solution methods, Lakshminarayana (1996). As the streamtubes are unknown at first, the calculations have to be carried out iteratively until convergence of flow conditions is reached and the streamtubes are found, Rühle and Bestle (2010). The radial distribution of the flow conditions and the streamtubes, as the result of this throughflow process, are then used to calculate suitable airfoil profiles along those streamtubes, i.e., S1 surfaces, which meet the flow requirements at minimal losses and wide working ranges.

This quasi-3D (Q3D) blading process uses an optimization process and 2D computational fluid dynamics (CFD) on the S1 surfaces and the perpendicular S2 surfaces along the axial direction of the compressor. The resulting airfoil profiles on the S1 streamtubes are then radially stacked to three-dimensional airfoil geometries which, however, do not account for the actual three-dimensional flow field. In order to reduce secondary flow losses, additional design principles such as lean and sweep are applied (Bräunling (2015)) and assessed stage-wise using 3D-CFD with mixing planes between each rotor and stator. The results of the stage-wise three-dimensional flow field are compared with the requirements (pressure ratio, efficiency, constraints). Finally, the whole compressor with all stages is calculated with 3D-CFD, completing the aerodynamic design process. However, if at any of the 3D-CFD assessments constraints or requirements are not met, changes have to be applied to the geometry and the design has to be re-assessed, see Fig. 1.2. In the case of failing design criteria repetitively, they may have been chosen to challenging and must be redefined. Then the design process is set back to the appropriate design stage, i.e., either meanline, throughflow, or blading.

As soon as the aerodynamic design process delivers a three-dimensional airfoil geometry, structural design evaluation can be conducted. In a first step, a socket (root) including a mounting device is designed for the airfoil where proper positioning and dimensioning of

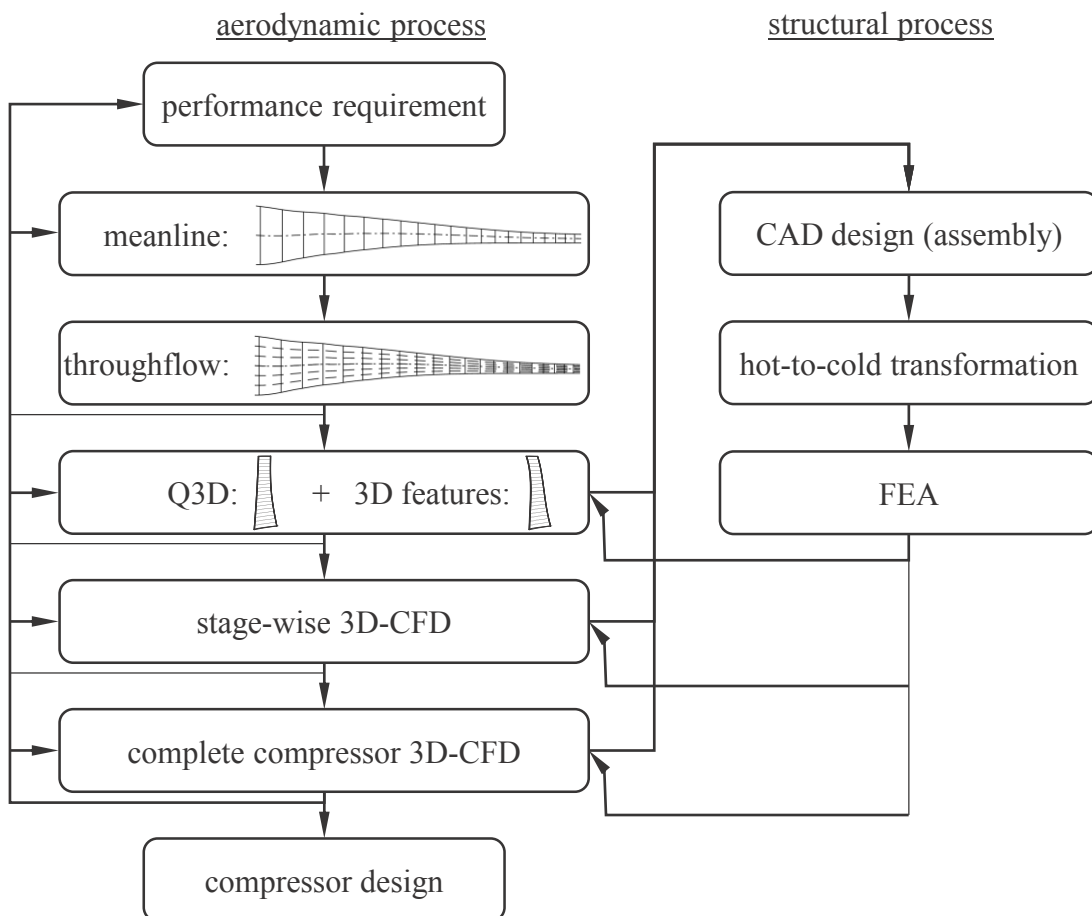


Figure 1.2: Compressor design process

the socket is initially calculated using low-fidelity tools as suggested by Schörner and Bestle (2012). After blade assembly, which represents the loaded geometry at the design-operating conditions of the gas-turbine, has to be transformed to the unloaded geometry which is how the part will be delivered. Only then the loads and temperature conditions will deform the geometry correctly into the loaded state defined by the aerodynamic process, so that the structural assessment will be consistent with the aerodynamic one. Finally, the structural properties are calculated using finite-element analysis (FEA) providing static and dynamic stresses as well as eigenfrequencies and characteristic vibration modes. If at any of the last three aerodynamic design stages structural integrity is not ensured (structural criteria are violated), the design has to be modified and re-assessed (aerodynamic- and structural-wise), see Fig. 1.2. However, if it is repetitively not possible to pass both aerodynamic and structural constraints, or meet performance targets, returning to one of the prior stages of the design process becomes necessary and it may also be necessary to weaken some of the design constraints. In case this is not acceptable, better materials or lower performance requirements have to be considered. Further structural assessments of other compressor parts such as shaft and casing are not required, as those are massive parts in case of stationary gas-turbines from ALSTOM and, therefore, are hardly critical to failure.

## 1.2 Current State of Robust and Structural Design Optimization

The primary goal of structural design optimization of compressor airfoils is to ensure structural integrity of the airfoils and maximize their lifetime (durability). While the short-term integrity can be assured by limiting the maximal stresses, the improvement of lifetime requires reduction or prevention of high-cycle fatigue (HCF), i.e., failure after at least  $10^4$  load cycles within the elastic strain rate of the material. Root causes for such dynamic loads are excitations of structural eigenmodes which may be caused by self-induced cyclic pressure fluctuations (flutter) or by periodic pressure wakes of installations (resonance through engine orders) such as upstream and downstream located rotor and stator airfoils, probes et cetera, see Campbell (1924). Whether or not an excitation actually leads to resonance with an airfoil eigenfrequency depends on the aerodynamic damping in case of flutter, see Grieb (2009), and the agreement between the shape of the excitation source and shape of the eigenmode of the compressor-row assembly in case of engine orders, see Singh et al. (1988). While accurate prediction of damping requires computationally expensive, coupled unsteady flow-structure analysis, the assessment of the source-eigenmode-shape agreement can either be easy or impossible depending on the design of the assembly. In order to keep design optimization computationally cheap, many authors demand that no excitation frequency meets a relevant eigenfrequency of the specific part within the working speed range of the considered machine. The definition of a relevant eigenfrequency and the method of avoiding its excitation, however, greatly vary between different authors and often seem to be rather arbitrary. For example, Seppälä and Hupfer (2014) apply automatic design optimization to a turbine guide vane, in order to prevent any of the first five eigenfrequencies from decreasing w.r.t. a reference design, because they consider the

first five eigenmodes to be most critical in regards to failure due to their higher kinetic energy. However, they focus on resonance only and their criterion does not really prevent eigenfrequency excitation, and certainly it does not assess the risk level of a source to excite a specific eigenmode. In contrast to this, Pugachev et al. (2014) simply avoid definition of an optimization criterion or detection strategy for resonance by forcing the user to define goals for the shifting of eigenfrequencies. Clearly, such a method, which was applied to a compressor blade, is not suited to be used in a general automated design process due to the need for user guidance.

An earlier approach, more suitable to automation was developed by Astrua et al. (2012), where design optimization is used to avoid resonance of the first eight eigenmodes by shifting them away from intersections with engine orders. Further, production limitations on the thickness of leading and trailing edges are considered, and fixed safety margins are applied to the eigenfrequencies to account for prediction uncertainties due to production tolerances and changes in running as well as installation conditions. This was already suggested earlier by Otto and Bestle (2007) demanding that resonance should be avoided for all eigenfrequencies of compressor airfoils within the working speed range of an aero-engine. In contrast to other authors, they also account for the possibility of flutter caused by eddy shading in the Kármán-vortex street and evaluate the fatigue resistance w.r.t. the Haigh diagram. The fixed eigenfrequency uncertainty margins w.r.t. the uncertain environmental and installation conditions used in both papers may be substituted with the prediction of upper and lower eigenfrequency limits by considering possible extrema in temperature and fixation conditions as suggested by Fedorov et al. (2010) for mounted airfoils, or in case of bladed integrated disks (blisks) by calculating the eigenfrequencies of lowest and highest possible nodal diameter, see Strehlau and Kühhorn (2010). Based on these eigenfrequency extrema, Hecker et al. (2011) introduced a penalty strategy which assigns specific penalties to each eigenmode shape and engine-order. The reasoning is that, based on experimental studies and in-service experience, different eigenmode shapes result in different risk of failure due to the mode-shape-specific stress field and due to different sensitivity of eigenmode shapes on flow excitation. The specific penalties for different excitation sources (engine orders) are related to their different strengths. With this penalty strategy, the authors shift the first ten eigenfrequency bands of a compressor blade away from intersections with engine orders and succeed in changing the invalid design into one free of resonance. Besides the amplitude-frequency ratio as described by Otto and Bestle (2007), which however requires the experimental determination of a geometry specific correction factor, the penalty strategy is the only approach so far which accounts for different risk of failure and excitation of eigenmodes without the need of conducting time-consuming coupled aero-elastic calculations. For application it requires the assignment of eigenfrequency bands to specific eigenmode shapes being considered as relevant. Hecker et al. (2011) assign these by hand prior to the optimization, without taking into account that the order of eigenmodes may change w.r.t. the eigenfrequency order when the design is modified during optimization, which may be seen as a drawback.

Alternatively, Blocher and Fernández (2014) accelerate coupled unsteady aero-elastic computations by developing a time-linearized-forced-response-analysis method (TLFRA



method) which is incorporated into a coupled aerodynamic and structural optimization process of a counter rotating fan by Aulich et al. (2013) as well as Fernández and Blocher (2014). Blocher and Aulich (2013) found that similar eigenmode shapes have similar aerodynamic damping characteristics, i.e., a similar risk of flutter or large vibration amplitudes due to forced response, which supports the reasoning for the penalty strategy by Hecker et al. (2011). The advantage of the penalty strategy over TLFRA is its speed and that uncertainties can easily be considered. Further, TLFRA has to be applied to specific operating conditions, hence, in case of uncertainties at least two TLFRA analyses, one at the lower and one at the upper limit of each eigenfrequency band, have to be performed for each intersection with engine orders (maybe even more to resolve the prediction within the eigenfrequency bands). In case of rather low uncertainties, as for the counter rotating fan with bolted clevises fixing the blades onto the disk, this may be feasible, but not for mounted blades with higher uncertainties in the tightness of the fixation, since the type of mounting significantly influences eigenfrequencies. At the downside of the mode-shape-specific penalty strategy one would require a separate flutter criterion and, in order to be applicable to automated optimization without any user interaction, a method which is able to assign correct eigenfrequency bands to relevant mode shapes for different airfoil designs.

As described in the previous section, a structural design process has to be coupled with an aerodynamic process in order to deliver feasible designs. The coupled processes suggested by Astrua et al. (2012) as well as Diener et al. (2016) and Pierret et al. (2007) execute aerodynamic and structural design evaluations subsequently for each design which slows down the optimization, because one discipline has to wait for the other before it can start. This causes idle time and does not take advantage of process parallelization. Astrua et al. (2012) accelerate the optimization by using a response-surface model (RSM) from an initial set of data to find the new designs faster while the calculated performance of each design is used to update the RSM. Other authors such as Buske et al. (2016), Aulich et al. (2013), and Siller and Marcel (2010) apply the concept of a single master process running the optimizer and multiple slave processes, each performing a serial aerodynamic and structural evaluations of a design. Hence, the possibility of parallelization has been used, but still considerable idle time is accumulated because the evaluation tools for each discipline have to wait for one another. Due to this, not only optimization time is used inefficiently but also licenses. In case of several slave processes the number of licenses for involved tools depends on the number of slaves and cannot be picked individually according to the actual need. Another problem of serial slave processes is that including additional evaluation tools from further disciplines may further degrade the performance. Therefore, Lockan et al. (2017) introduced a new strategy which takes advantage of individual RSMs for each evaluation discipline. This enables the usage of an arbitrary number of slave processes for each discipline, since the necessity of consistent design evaluation by different processes is abolished. This method, therefore, takes most advantage of computational resources and the available licenses, e.g., three structural evaluation processes may be combined with seven aerodynamic processes et cetera.

Being able to arbitrarily combine processes for aerodynamic and structural design

evaluation creates a problem of geometric inconsistency, because aerodynamic evaluations have to be performed on the loaded geometries under running conditions, whereas structural evaluations must be performed on the associated unloaded geometry which will match the loaded geometry, after all load and temperature conditions have been applied. The need for loaded-to-unloaded transformation between aerodynamic and structural processes has so far only been recognized by Buske et al. (2016), Aulich et al. (2013), and Siller and Marcel (2010) using an iterative method described by Goerke et al. (2012) which is actually standard in industry but considerably time-consuming and complex to implement. Other authors such as Astrua et al. (2012) and Joly et al. (2014) simply claim that the differences between structural results received by evaluating the unloaded or loaded geometry are insignificant.

Another hardly recognized issue is that each design optimization process should include at least one objective considering design robustness. The reason is simple: generally design optima will be located at the borders of the constrained design regions where design variations due to production tolerances or operating uncertainties will then violate constraints, and a significant fraction of the produced parts will fail in service. Thus, harshly said, optimization results without consideration of design robustness are useless for the final product. Additionally, uncertainties may deteriorate the performance of an "optimal" design to such an extent that other designs which perform poorer w.r.t. the design objectives, but are less sensitive, may be more suitable. Flassig (2011) introduces the difference between deterministic uncertainties, resulting from e.g. uncertain operating conditions, and such resulting from stochastic production variations. Typically authors evaluate probabilistic design robustness either through mean value and variance of objectives such as Vinogradov et al. (2016), or through failure rate w.r.t. constraints which, however, is misleading as it will be shown later. Others like Dow and Wang (2015) assess production tolerances, but only w.r.t. robust design objectives and not constraints. Furthermore, the authors do not evaluate the possible production tolerance directly, but apply standard deviations to every point on the blade surface. This provides a qualitative, but not quantitative measure which, nevertheless, allows to be used as an optimization objective to reduce production costs by maximizing tolerances. A weak point is, that for the final product neither standard deviation nor failure rate are requirements which can be communicated to the producing suppliers, but geometrical tolerances could.

### 1.3 Motivation and Outline of Thesis

The literature survey in the previous section reveals several shortfalls of the current state of structural design optimization processes. It is the intention of this thesis to develop a structural design optimization process which is a major improvement to the current state in research in order to succeed in the goals of making such processes faster and improving their results w.r.t. real-world application in gas-turbines.

The major goal of structural airfoil design is to ensure structural integrity of designs. The lifetime of a design is limited by HCF mainly caused by flutter and forced response. Those phenomena are related to aero-elasticity and are computationally time-consuming to

assess even with the approach of Blocher and Fernández (2014). It is therefore decided to apply the flutter criteria used by Otto and Bestle (2007) and the shape-specific eigenmode penalty strategy proposed by Hecker et al. (2011). For the latter to be applicable within an automated design process, a procedure will be developed which is able to identify relevant eigenmodes and assign proper eigenfrequency bands. The identification of relevant eigenmode shapes requires making data of different designs comparable. Therefore, a method is introduced which is able to project node displacements of arbitrary surface geometries and meshes onto standard rectangular surfaces. These standard surfaces are then used for dimension reduction and classification of eigenmode data. In order to examine what proper eigenfrequency bands are, the behavior of eigenfrequencies and the corresponding mode shapes at varying boundary conditions are analyzed with special consideration of frequency veering. Additionally, a smooth intersection measure between eigenfrequency bands and engine orders is introduced to improve the performance of optimization algorithms regarding mode tuning. The consistency between structural and aerodynamic design evaluation is achieved by an innovative direct loaded-to-unloaded transformation which is faster and easier to implement than the conventional iterative approach described by Goerke et al. (2012).

The quality of the structural design-evaluation process is tested within a coupled aerodynamic and structural optimization process which applies the method introduced by Lockan et al. (2017) with multiple slave processes for different disciplines evaluating non-matching sets of designs, where discipline specific RSMs are utilized by the optimizer to find the global optimum. Thanks to the work of Hartwig and Bestle (2016) and Hartwig and Bestle (2017), the quality of the RSMs and performance of the optimizer were significantly improved. The application of RSMs also allows for computationally justifiable robust-design analysis. In the scope of this thesis, a method is suggested which directly calculates the possible production tolerance within which designs won't violate any constraints. Based on the production tolerance, an optimization objective is defined to facilitate the search for designs with higher production tolerances, hence, lower production costs.

This thesis covers the topic in eight chapters, where the first is the present introduction. The second chapter contains details about the proposed multidisciplinary optimization process, including the parameterization model, a definition of the constraints and objectives, a new mode-shape-specific resonance criterion, and the invented method for analyzing the possible production tolerance. Based on this problem formulation, the third chapter will explain the structural design assessment based on FEA and introduce a new method of loaded-to-unloaded transformation. Additionally, the influence of fixation uncertainties onto eigenfrequencies and mode shapes will be studied. Since the new resonance criterion requires automated identification of relevant mode shapes, the fourth chapter will introduce a method for projecting information of arbitrary airfoil surfaces onto standard rectangular surfaces. Subsequently, the fifth chapter will compare methods of data normalization and dimension reduction in order to filter the relevant information of the projected eigenmode-displacement fields which characterize the different mode shapes of interest. The subject of the sixth chapter is to find a classification algorithm with a low misclassification

rate for mode shapes and to develop an assignment procedure for mode-shape-specific eigenfrequency bands based on the fixation study from the third chapter. The feasibility of the structural design-evaluation process and robust-design assessment is validated in the seventh chapter where the multidisciplinary design optimization process is applied to an example of an industrial test compressor row. In the final chapter, conclusions will be drawn and an outlook on further possible improvements will be given.

# 2 Multidisciplinary Design-Optimization Process for Axial Compressors

The goal of the thesis is to contribute to the enhancement of structural design evaluation of compressor airfoils within multidisciplinary optimization processes and to assess design robustness, i.e., design sensitivity with respect to uncertainties in production, operation, and installation conditions. In order to validate the design evaluation process as well as the feasibility of newly developed objectives and constraints a, corresponding optimization process is needed. As the airfoils have to be aerodynamically most efficient while ensuring structural integrity, this optimization process has to combine both aerodynamic and structural design evaluation. In the following, details about the optimization process will be described, where the focus is on structural design aspects, whereas the aerodynamic process is part of another work-package within the superordinate research project. Also the developed optimization strategies, including the underlying response-surface models, are not part of this thesis and will only be mentioned or referenced, but not explained in detail.

The first part of this chapter explains the general structure of the optimization process and its components. Then, details about the implemented geometry and parameter model of compressor airfoils will be given, followed by a definition of the deterministic optimization problem and relevant design objectives and constraints. Finally, the optimization problem will be redefined in the context of robust design by introducing proper objectives and developing a method for estimating maximal possible production tolerances.

## 2.1 Optimization Process

The principle structure of the process is outlined in Fig. 2.1 and consists of two subprocesses, the pre-blading and the blading process. Using heuristic design and low-fidelity evaluation tools, the pre-blading process optimizes the annuls contour of the compressor and submits an initial guess of the blade shape to the blading process utilizing high-fidelity as well as low-fidelity evaluation tools to explore optimal airfoil designs. In a first step, the pre-blading process combines a heuristic meanline definition process (Keskin (2007)) with a partly heuristic throughflow definition process (Rühle and Bestle (2010)) to a coupled

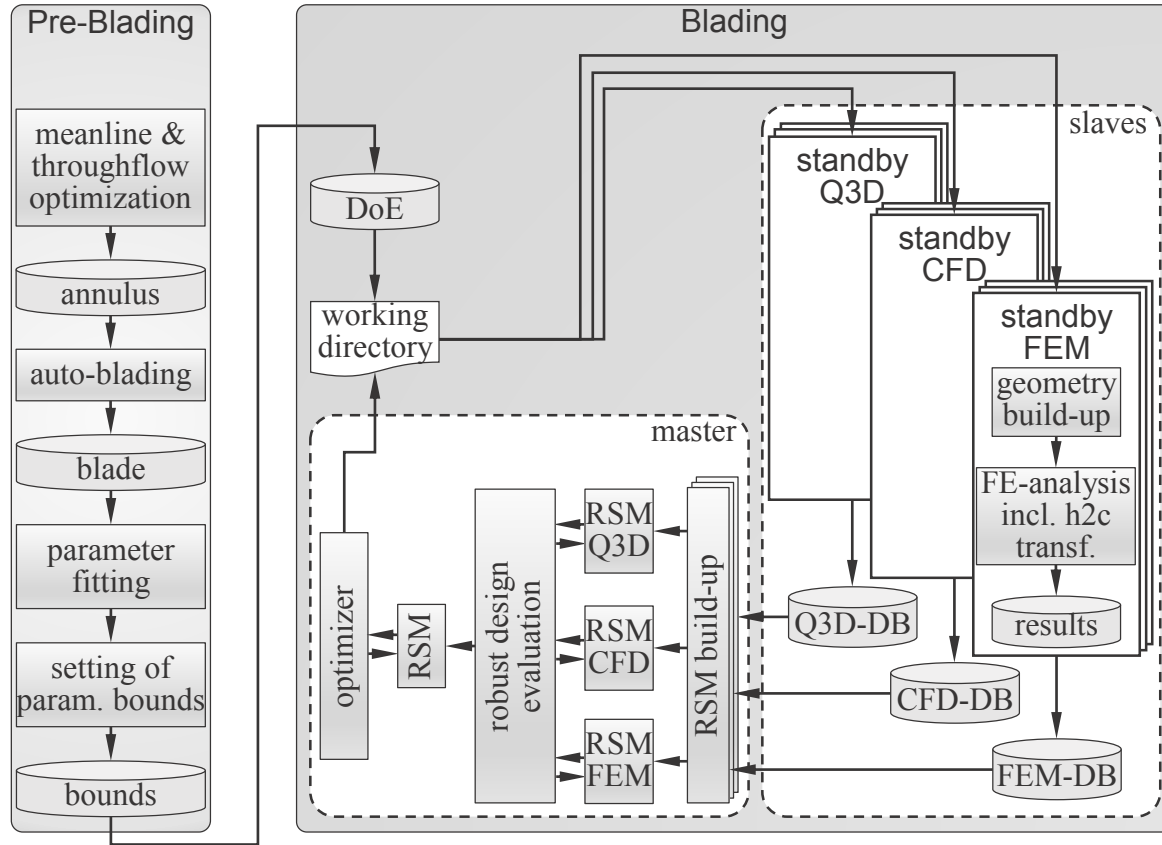


Figure 2.1: Multidisciplinary compressor-design-optimization process with an initial design guess of airfoils calculated by the pre-blading process and optimized by the blading process

optimization process (Pöhlmann and Bestle (2012)), which determines an optimal geometry of the annulus, proper segmentation of the stages, and the number of blades/vanes per stage. Subsequently, the confidential auto-blading process calculates a first guess for the airfoil geometries based on correlations and iterative optimization by performing calculations on 2D meridional and radial planes (Wu (1952), Cumpsty (2004)), in order to meet pressure and temperature requirements defined by the meanline-throughflow optimization. The parameter model used by the blading process (Section 2.2) is then fitted to each airfoil as initial geometry. The fitting allows to use any arbitrary start geometry which must not necessarily be received from the meanline-throughflow and auto-blading processes.

For the subsequent high-fidelity optimization of the airfoil geometries by the blading process, the bounds of the optimization parameters are set based on the fitted airfoil parameter model. In contrast to this, the annulus design is not assessed by any high-fidelity tool because, within the narrow speed-range of stationary gas turbines, the aerodynamic characteristic of the large scale annulus structure is well captured by less costly heuristic and 2D tools. For the same reason, it is assumed that the aerodynamic efficiency of the annulus is not prone to production tolerances and changing operation conditions. Thus, no robust-design assessment is applied to the annulus geometry as well. Due to the large scale and massive rotor design of ALSTOM gas turbines, further structural assessments

w.r.t. shaft and casing are also not needed.

The bounds of the airfoil geometry parameters, i.e., the specified design range, is communicated to the blading process and a design of experiments (DoE) is executed using Latin-hypercube sampling (McKay et al. (1979)). This is the initial foundation for the further optimization, where each sample is written as a separate file into a working directory from where the slave processes may pick designs to be evaluated. These are independently and simultaneously running processes related to low-fidelity quasi-3D CFD analysis, high-fidelity 3D-CFD, or FEA. The results are written into databases, one for each discipline. The possibility of running multiple instances of arbitrary evaluation processes (slaves) simultaneously is superior to the execution of, e.g., performing FEA on multiple CPU's, because in this way also pre- and post-processing profit from parallelization. In the present application three FEM-processes, each using only a single CPU, are able to process more samples from the working directory per hour with fewer licenses required than a single FEM process using 20 CPU's: 1 instance with 1 CPU runs 4 evaluations/h and requires 6 licenses  $\rightarrow$  3 instances with 1 CPU per instance run 12 evaluations/h and require 18 licenses, whereas 1 instance with 20 CPU's needs 25 licenses and performs 6 evaluations/h only. After all samples (designs) in the working directory have been analyzed, the slaves idle in standby until new designs are written into the working directory.

Due to the different speed of different design evaluation processes (here FEM and 3D-CFD take more time than Q3D) and since different disciplines may run different numbers of instances, the databases do not necessarily contain the same set of evaluated designs. Therefore, the master process has to build individual response-surface models (RSMs) for the optimizer, where the multi-objective genetic algorithm NSGA-II by Deb et al. (2002) is applied. The RSMs will introduce inaccuracies in prediction of the design behavior, which is no issue since the optimizer proposals are written again to the working directory and subsequently evaluated by the slave processes, which in turn will update the RSMs properly, making them more accurate in the relevant search regions. This approach, developed by Hartwig and Bestle (2016), is able to find better optimal solutions faster than requiring all slave processes to evaluate exactly the same designs from the working directory at a time, because it enables low-fidelity processes to explore the design space a lot faster than the high-fidelity processes in order to guide search to the most promising design regions. In order to avoid conflicting predictions between the high-fidelity CFD and the low-fidelity Q3D process, they do not share any design objective. However, among the aerodynamic constraints the exit flow angle is shared by both processes and a possible conflict is eased by setting wider bounds for the Q3D than for the CFD. As the violation of the Q3D and CFD predicted exit flow angle is punished by associated penalties, the minimization of their sum in the constraint function (to be introduced later) will cause the optimizer to seek for regions within the design space with least conflict between both predictions.

The RSMs are generated by using the Kriging method (Kriging (1951); Forrester et al. (2008)) in combination with partial-least squares (Wold (1985); Rosipal and Krämer (2006); Bouhlef et al. (2016); Hartwig and Bestle (2017)). Additionally, each discipline in Fig. 2.1 may actually include several RSMs, one for each design objective and constraint

as suggested by Hartwig and Bestle (2016). Based on these RSMs, the robust-design evaluation calculates the final optimization objectives (see Section 2.4), writes them to the databases, and calculates new RSMs w.r.t. the final objectives.

## 2.2 Geometry and Parameter Model

In accordance to Dutta (2011), an airfoil geometry is defined on 21 equidistantly distributed sections between the hub-annulus contour  $r^{Hub}(x)$  and tip-annulus contour  $r^{Tip}(x)$  which are received from the meanline-throughflow process. Thus, w.r.t. the axial coordinate  $x$  of the compressor, the radial coordinate of each section is described by

$$r(x, i) := r^{Hub} + \frac{i-1}{21-1} (r^{Tip} - r^{Hub}) \quad \text{where } i \in [1, 21] \subset \mathbb{N}. \quad (2.1)$$

Alternatively to the index  $i$ , the sections may be identified by an associated average radius

$$\bar{r}(i) := \frac{r_L(i) + r_T(i)}{2} \quad (2.2)$$

where  $r_L$  and  $r_T$  are the radii of a section at leading and trailing edge of an airfoil. For generality of the following geometry parameterization, the average radius of the sections (2.2) is rescaled between zero and one as

$$\tilde{r}(i) := \frac{\bar{r}(i) - \bar{r}(1)}{\bar{r}(21) - \bar{r}(1)} \in [0, 1]. \quad (2.3)$$

The airfoil geometry may then be defined in the  $\tilde{x}$ - $\tilde{r}$ - $\theta$ -plane, where  $\theta$  is the circumferential angle in Fig. 2.2a. Within each section, the airfoil geometry is build-up by perpendicularly

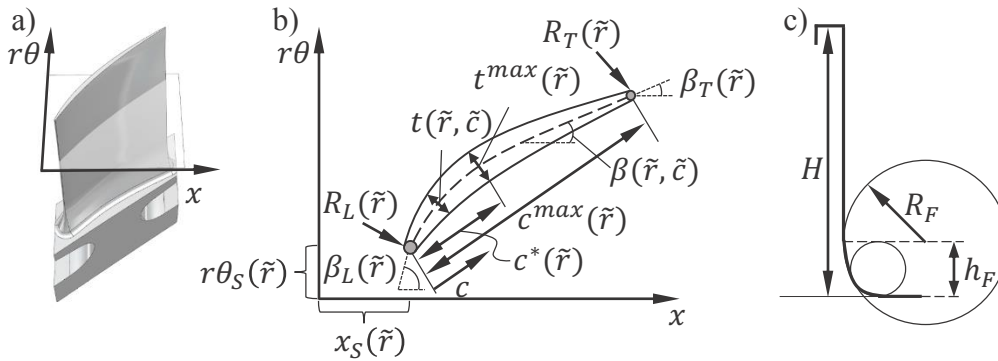


Figure 2.2: Geometry model with a) plane of constant average radius, b) geometry parameters, and c) fillet parameters



superposing the thickness distribution  $t(\tilde{r}, \tilde{c})$  described by its dimensionless distribution

$$\tilde{t}(\tilde{r}, \tilde{c}) := \begin{cases} \frac{t(\tilde{r}, \tilde{c}) - R_L(\tilde{r})}{t^{max}(\tilde{r}) - R_L(\tilde{r})} & \text{for } 0 \leq \tilde{c} \leq c^* \\ \frac{t(\tilde{r}, \tilde{c}) - R_T(\tilde{r})}{t^{max}(\tilde{r}) - R_T(\tilde{r})} & \text{for } c^* \leq \tilde{c} \leq 1 \end{cases} \quad (2.4)$$

with the camber-line defined by angle distribution  $\beta(\tilde{r}, \tilde{c})$  or

$$\tilde{\beta}(\tilde{r}, \tilde{c}) := \frac{\beta(\tilde{r}, \tilde{c}) - \beta_L(\tilde{r})}{\beta_T(\tilde{r}) - \beta_L(\tilde{r})} \quad (2.5)$$

where  $\tilde{c} := c/c^{max}(\tilde{r})$  is the dimensionless chord position associated with the actual chord position  $0 \leq c \leq c^{max}$  of a section. The position of maximal thickness  $t^{max}(\tilde{r})$  is  $c^*$ , and  $R_L(\tilde{r})$  and  $R_T(\tilde{r})$  are the leading- and trailing-edge radius distributions;  $\beta_L(\tilde{r})$  and  $\beta_T(\tilde{r})$  are the inlet and outlet blade-angle distributions, see Fig. 2.2b.

Additionally, lean and sweep may be applied via theta shift  $\theta_S(\tilde{r})$  and axial shift  $\tilde{x}_S(\tilde{r})$ . The fillet of the airfoil is constructed by a superposition of a circle with radius  $R_F$  tangentially attached to the airfoil surface and a second circle with diameter  $h_F \leq R_F$  tangentially attached to the root and the first circle, see Fig. 2.2c. With respect to the blade height  $H := \bar{r}(21) - \bar{r}(1)$ , those parameters may be substituted by the dimensionless parameters

$$\tilde{R}_F := R_F/H \text{ and } \tilde{h}_F := h_F/R_F, \quad (2.6)$$

which may be reasonably bounded between zero and one. Finally, the complete airfoil geometry is generated by piecewise cubic-spline interpolation (Piegl and Tiller (1997)) between the sections.

The clearance between airfoil tip and compressor casing is preselected according to engineering experience for the hot and loaded airfoil geometry, which is the basis for the CFD evaluation. Therefore, the blade is extrapolated beyond the 21<sup>st</sup> section and afterwards trimmed down to  $r(x, 21)$  minus the required tip clearance. The airfoil root is not part of the optimization, but it is sized using heuristic methods before optimization. However, it may also be calculated within an automated optimization as described by Schörner and Bestle (2012). Although, implementation into the present process is no issue, it would, however, increase optimization time significantly.

Considering all 21 sections with the corresponding geometry parameters would lead to an impractically high number of design parameters for design optimization. Instead, the distributions of the geometry parameters  $t^{max}(\tilde{r})$ ,  $\theta_S(\tilde{r})$  and  $x_S(\tilde{r})$  are represented by cubic Bezier-splines (Fig. 2.3a),  $R_L(\tilde{r})$ ,  $R_T(\tilde{r})$ ,  $\beta_L(\tilde{r})$  and  $\beta_T(\tilde{r})$  by quadratic Bezier-splines,  $\tilde{\beta}(\tilde{r}, \tilde{c})$  by a Bezier-spline surface of quadratic order in  $\tilde{r}$ -direction and cubic order in  $\tilde{c}$ -direction (Fig. 2.3b), and  $\tilde{t}(\tilde{r}, \tilde{c})$  by a piecewise quadratic B-spline surface (Fig. 2.3c). Hence, the optimization parameters are now the control points of these spline curves

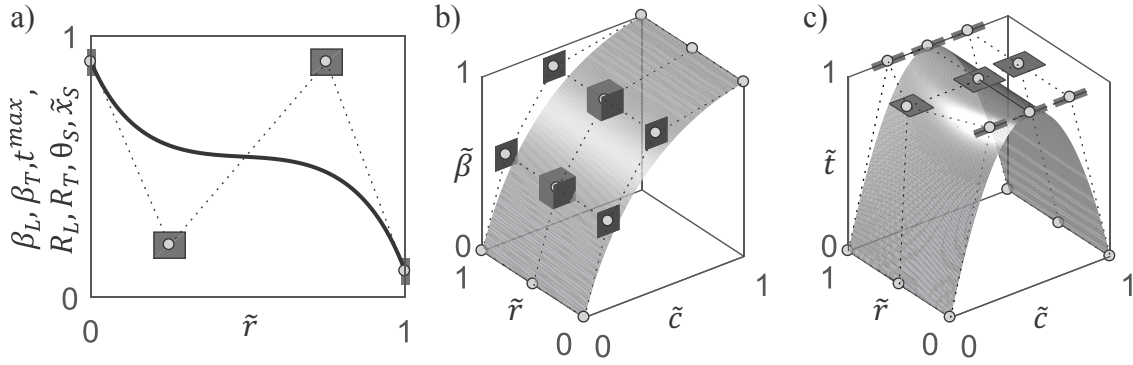


Figure 2.3: B-Spline representation of a) geometry parameters with only radial dependency, b) camber-line angle, and c) thickness distribution

and surfaces (Piegl and Tiller (1997)). However, not all control points are free design parameters; e.g., in order to ensure that the root of the airfoil is not shifted in case the theta and axial shift are being modified, the first control point of  $\theta_S(\tilde{r})$  and  $x_S(\tilde{r})$  is held fixed at position  $[0, 0]$ , respectively. Furthermore, control points of  $\tilde{t}(\tilde{r}, \tilde{c})$  and  $\tilde{\beta}(\tilde{r}, \tilde{c})$  with the same grid position in  $\tilde{r}$ -direction have the same  $\tilde{r}$ -coordinate. This improves optimization results noticeably due to fewer design parameters and has only little influence on the design freedom of the spline surfaces. According to the degrees of freedom (DoF) of the control points in Fig. 2.3 (black line  $\equiv$  one DoF, gray square  $\equiv$  two DoF, and gray cube  $\equiv$  three DoF), the total number of design parameters is 57.

For optimization the design space has to be limited to a reasonable range. The limitation acts on the control points of the splines as visualized in Fig. 2.3 by lines, squares, and cubes about the control points according to the DoF of the corresponding control point. Due to normalization of design variables, each optimization parameter  $p_i$  is bounded between zero and one or

$$0 \leq \check{p}_i \leq p_i \leq \hat{p}_i \leq 1 \quad (2.7)$$

where  $\check{p}_i$  and  $\hat{p}_i$  are user-defined bounds. Unless set to zero and one, a meaningful definition of bounds for each  $p_i$  requires firstly a fitting of the spline curves and surfaces in Fig. 2.3 to reference distributions of geometry parameters given either by auto-blading (for compressor designs developed from scratch, see Fig. 2.1) or an existing reference design. As the shape of splines is nonlinearly dependent on the control-point coordinates, fitting is accomplished by an optimization process minimizing the least-square error between a reference data set and the desired spline representation, see Dierckx (1993).

## 2.3 Deterministic Optimization Problem and Constraints

The design of an airfoil has to account for multiple objectives  $f_i$ , which may be contradictory, and inequality constraints  $b_i \leq 0$ . Equality constraints are avoided here in order to enable

the optimizer to explore the design space efficiently. Considering that every objective  $f_i$  can be formulated in such a way that its reduction will improve the design (Bestle (1994)), the general deterministic optimization problem may be stated as

$$\min_{\mathbf{p} \in \mathcal{P}} \mathbf{f}(\mathbf{p}) \quad \text{s.t. } \mathcal{P} := \left\{ \mathbf{p} \in \mathbb{R}^{57} \mid \mathbf{b}(\mathbf{p}) \leq \mathbf{0}, \check{\mathbf{p}} \leq \mathbf{p} \leq \hat{\mathbf{p}} \right\}, \quad (2.8)$$

where  $\mathbf{p} := [p_1, \dots, p_{57}]$  is the design vector with a permitted parameter range  $[\check{\mathbf{p}}, \hat{\mathbf{p}}]$  about the fitted design, see Section 2.2. The vector  $\mathbf{f}(\mathbf{p})$  summarizes objectives to be minimized simultaneously, and  $\mathbf{b}(\mathbf{p})$  is a vector of inequality constraints. For the optimizer to be able to solve the problem (2.8), the  $N_{struct}$  structural and  $N_{aero}$  aerodynamic constraints are combined by the optimizer to a penalty factor

$$C := \sum_{i=1}^{N_{struct}+N_{aero}} \left( w_i \max(0, b_i)^2 \right) \quad (2.9)$$

which is added to each objective  $f_i$  in Eq. (2.8). The distinction between different importances and scaling among the constraints  $b_i$  can be controlled through weights  $w_i \geq 0$ . Thus, when defining the aerodynamic objectives as the total pressure loss  $\Delta p_t$  at design operation conditions of the gas turbine and off-design losses  $\omega_{OD}$ , the deterministic optimization task to be solved by the optimizer states as

$$\min_{\mathbf{p} \in \mathcal{P}} \begin{bmatrix} \Delta p_t + C \\ \omega_{OD} + C \end{bmatrix} \quad \text{s.t. } \mathcal{P} := \left\{ \mathbf{p} \in \mathbb{R}^{57} \mid \check{\mathbf{p}} \leq \mathbf{p} \leq \hat{\mathbf{p}} \right\}. \quad (2.10)$$

Hence, the optimizer seeks to minimize each objective and the violation of constraints simultaneously.

There are several constraints ensuring structural integrity of designs and their producibility. As most of the criteria except for the mode-resonance factor (MRF) are commonly applied

the applied constraints without greater details. The MRF, however, will be explained in detail because of its central meaning for many methods developed in this thesis.

### Mode-Resonance Factor

In order to prevent high-cycle fatigue (failure after a minimum of  $10^4$  load cycles with low stresses within the elastic strain rate of the material) within the required lifetime of an airfoil, excitation of the airfoil's eigenmodes should be avoided. One reason for excitations may be forced response to flow disturbances (pressure wakes) caused by installations around the circumference, e.g., stators in case of a rotor blade and visa versa. Let  $N_j^I$  be the number of installations on the circumference of the  $j^{\text{th}}$  source and let the engine shaft speed be  $n$ , then the so called engine-order excitation frequency is

$$f_j^A := N_j^I n. \quad (2.11)$$

In the case that the excitation frequency  $f_j^A$  coincides with an eigenfrequency  $f_i^E$  of an airfoil, large forced-response vibration (resonance) is likely to occur. The focus here is on such forced responses rather than self-induced vibration of the airfoil (flutter), which is covered by other criteria within the subsequent section. Values being considered for  $N_j^I$  are the number of vanes (blades) on the next two stators (rotors) upstream and next one downstream of the rotor (stator) of interest as well as the difference between the number of vanes (blades) of the stators (rotors) adjacent to it. Additionally, some lower engine orders are included due to various other installations.

Because the engine orders (2.11) linearly increase with  $n$  while, in comparison, the eigenfrequencies change only little, there are multiple intersections between  $f_j^A$  and  $f_i^E$ , and thus possible occasions of resonance, see Fig. 2.4a (Campbell (1924)). Note, that some eigenfrequencies increase with shaft speed  $n$  whereas others decrease, because eigenfrequencies are differently effected by increased temperatures (softening) and centrifugal loads (stiffening). From Fig. 2.4a it becomes clear that a complete resonance-free design may not be achievable and one has to focus attention onto the relevant operation speed-ranges representing the major part of the gas turbine's life time, Fig. 2.4b. For stationary gas turbines, the typical speed range  $[\tilde{n}, \hat{n}]$  is about 95%-105% of the electrical net frequency.

Even within this operation speed range, complete avoidance of possible resonances may still be a challenging task. Therefore, it is usually considered that in a complete assembly of airfoils on a disk or blisk (Cumpsty (2004)) not every intersection in the Campbell-diagram actually causes resonance to an airfoil, but only if the shape of the excitation source is the same as the nodal shape around the circumference (nodal diameters) of the eigenmode associated with the eigenfrequency of the assembly (Singh et al. (1988), Strehlau and Kühhorn (2010)). However, in case of uncertainties, e.g. w.r.t. temperature conditions or fixation of the airfoils on the disk, discrete eigenfrequencies turn into frequency bands,

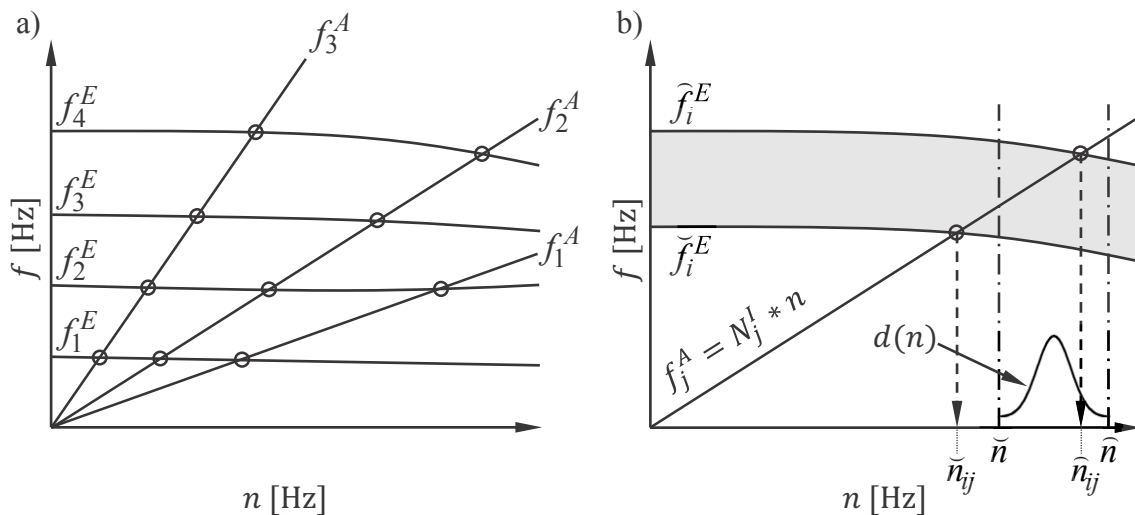


Figure 2.4: Campbell diagram with a) multiple intersections (o) between eigenfrequencies  $f_i^E$  and engine orders  $f_j^A$  and b) an intersection of a single engine-order with an eigenfrequency band due to uncertainties

see Fig. 2.4b, where the prediction of nodal diameters may not be unambiguous. This is especially the case for the compressor evaluated here, where airfoils have hammer-foot roots (see Fig. 2.5a) on most compressor rows and are strung together in grooves turned into a massive rotor for low production costs. The airfoil assembly of such a blade row contains high uncertainties w.r.t. contact conditions with neighboring airfoil roots which makes unambiguous prediction of nodal diameters impossible. The limits of eigenfrequencies, however, may be calculated by assuming either no contact to neighboring roots at surface B (equivalent to zero nodal diameters on a disk or blisk) and full inelastic contact (equivalent to number of diameters being equal to  $\lfloor (\text{number of blades})/2 \rfloor$ ). Dovetail roots as shown in Fig. 2.5b have negligible installation uncertainties at surface C, whereas other uncertainties resulting from varying operation conditions (e.g. temperature) or production accuracies are also present. While the latter will be considered in the final section of this chapter, the first may be considered by calculating the corresponding lower and upper eigenfrequency limits (e.g. calculations with hot-day and cold-day boundary conditions within a finite-element model). The exact combination of temperature, load, and installation conditions used to calculate the eigenfrequency limits are given in Chapter 3, where the FE model is explained.

As discrete eigenfrequencies  $f_i^E$  turn into eigenfrequency bands  $[\check{f}_i^E, \hat{f}_i^E]$  due to operation uncertainties, intersections with engine orders  $f_j^A$  result in intersection speed-ranges  $[\check{n}_{ij}, \hat{n}_{ij}]$  of possible resonances, see Fig. 2.4b. An intersection measure between the  $j^{\text{th}}$  engine order  $f_j^A$  and  $i^{\text{th}}$  eigenfrequency band  $[\check{f}_i^E, \hat{f}_i^E]$  may then be calculated by first introducing a weight function  $d(n)$ , which is zero outside of the operation speed-range  $[\check{n}, \hat{n}]$ . Integration of  $d(n)$  over the intersection speed-range  $[\check{n}_{ij}, \hat{n}_{ij}]$  and scaling of the

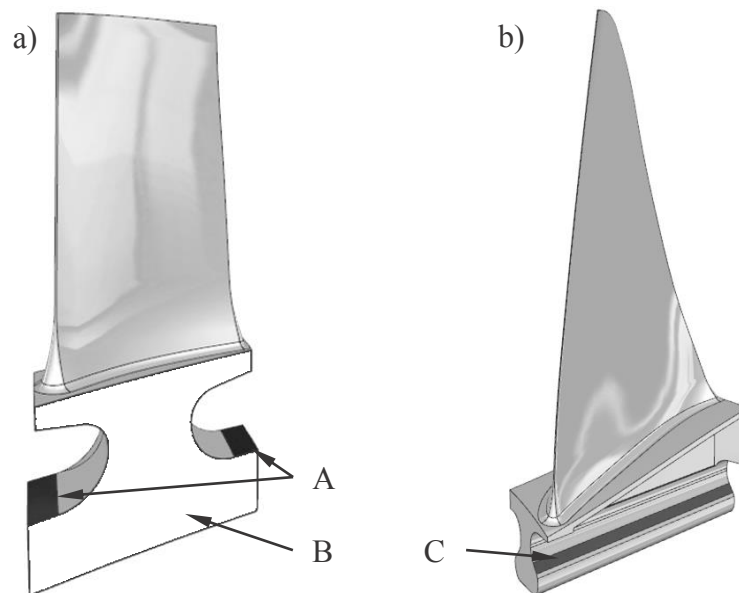


Figure 2.5: Rotor compressor airfoils with a) hammer-foot root (radially fixed at A and circumferentially at B by neighboring airfoils) and b) dovetail root (radially and circumferentially fixed at C)

result between zero and one gives

$$Int_{ij} := \frac{\int_{\check{n}_{ij}}^{\hat{n}_{ij}} d(n) dn}{\int_{\check{n}}^{\hat{n}} d(n) dn} \in [0, 1] . \quad (2.12)$$

This intersection measure is zero in case there is no intersection within  $[\check{n}, \hat{n}]$  and one in case of full intersection. The weight function  $d(n)$  may be used to account for runtime probability within  $[\check{n}, \hat{n}]$  or simply as a parameter that controls how the optimizer changes the eigenmode bands. For example, a cut-off Gaussian distribution

$$d(n) := \begin{cases} \sqrt{\frac{2}{\pi (\hat{n} - \check{n})^2}} \exp\left(-2 \frac{\left(n - \frac{\hat{n} + \check{n}}{2}\right)^2}{(\hat{n} - \check{n})^2}\right) & \text{for } \check{n} \leq n \leq \hat{n} \\ 0 & \text{else} \end{cases} \quad (2.13)$$

may put the optimizer's focus more on mode shifting, while a constant distribution

$$d(n) := \begin{cases} 1 & \text{for } \check{n} \leq n \leq \hat{n} \\ 0 & \text{else} \end{cases} \quad (2.14)$$

may shift the focus to tightening the eigenfrequency bands, i.e., reducing sensitivity to uncertainties.

Because different eigenfrequencies of an airfoil are related to different eigenmode shapes (characteristic displacement fields of airfoil surface), the stress patterns and thus the risk of failure is different for each eigenmode shape. For example, eigenmodes with stress concentrations at the leading or trailing edge are likely to cause crack initiation and failure. Additionally, the various engine orders differ in their excitation strength, and therefore in their contribution to the risk of failure. Hecker et al. (2011) consider both factors by assigning risk specific penalty values  $w_i^E$  and  $w_j^A$  to each eigenmode shape and engine order, respectively, which they successfully utilize in an automated design-optimization process. They also consider the existence of eigenmode bands due to uncertainties (Fedorov et al. (2010)), but do not apply a smooth intersection measure as e.g. Eq. (2.12). Instead they activate the penalties whenever engine-orders intersect with eigenmode bands. In contrast to this, combining Eq. (2.12) with this penalty strategy to the mode-resonance factor

$$MRF_i := \sum_j Int_{ij} (w_j^A + w_i^E) \quad (2.15)$$

introduced by Martin and Bestle (2016) defines a smooth penalty measure which should give better performance in conjunction with most optimization algorithms.

During optimization it is the goal that eigenfrequency bands of relevant eigenmode shapes are shifted away from possible intersections with the engine orders by applying proper design modifications. In the present case, the first eleven essential eigenmode

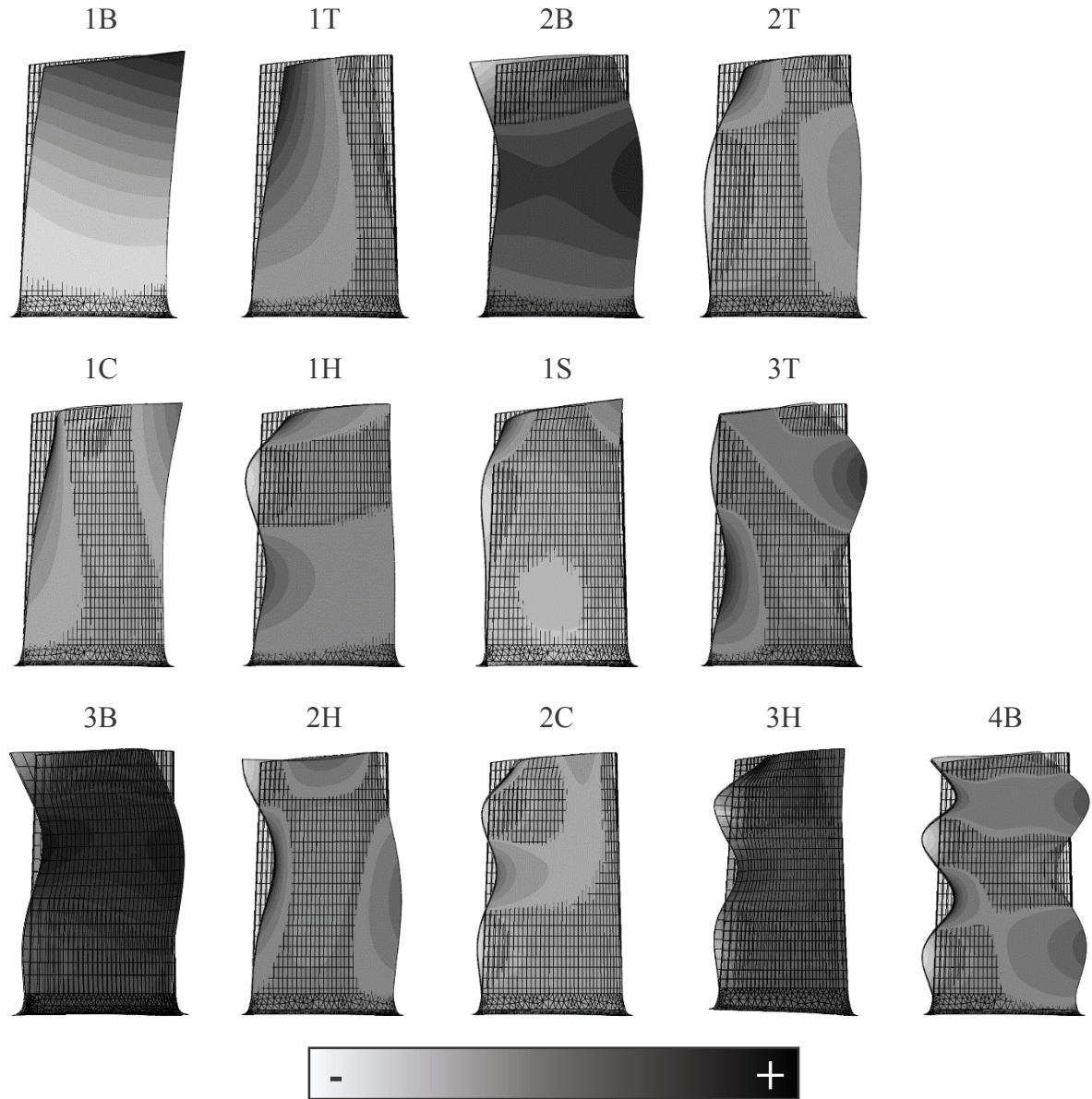


Figure 2.6: Major eigenmodes to be identified, where the naming convention is as follows: a mode counter is followed by mode specifier B for bending, T for torsion, C for chord-wise bending, H for higher-order, S for stiff-wise-bending mode (Note: shown eigenmodes originate not from same geometry).

shapes shown in Fig. 2.6 are of interest, but two more are to be detected in order to reduce misclassification rate as it will be explained in Chapter 6. Each eigenmode in Fig. 2.6 is associated with a specific risk of failure and thus penalty value  $w_i^E$  as shown in Table 2.1. Thus, eleven inequality constraint  $b_{1,\dots,11} := MRF_{1,\dots,11} \leq 0$  have to be considered by the optimizer, but only a fraction is actually going to be active during optimization, because not every eigenmode intersects with engine orders within  $[\check{n}, \hat{n}]$ . The strength of engine orders is weighted via  $w_j^A$  as shown in Table 2.2.

The mode-penalty factor  $w_i^E$  requires identification of corresponding eigenmode shapes

and a pairing of eigenfrequency prediction limits  $\check{f}_i^E$  and  $\hat{f}_i^E$  according to their mode shapes rather than their order. For this purpose a classification method and assignment strategy is required which will be developed in Chapter 6. The cornerstone is the ability to compare displacements fields of different airfoil geometries, which is developed in Chapter 4. Details about the calculation of the eigenfrequency limits  $\hat{f}_i^E$  and lower limit  $\check{f}_i^E$  are given in Chapter 3.

## Flutter

Besides forced response, also self-induced excitation (flutter) has to be considered as critical. One source of flutter is the finite thickness of the trailing edge of airfoils which causes alternating flow-separation with specific frequency and thus pressure fluctuations on the suction and pressure side. The frequency of this eddy shedding in the Kármán-vortex street is described by the Strouhal-number  $St$  (Strouhal (1878)). Defining lower flutter limits  $\check{L}_{FL1,FL2}$  on  $St$  results in two constraints

$$b_{12,13} := \check{L}_{FL1,FL2} - \underbrace{\frac{2\pi c^{max}(\tilde{r} \hat{=} 0.75) \check{f}_{1,2}^E(n \hat{=} 100\%)}{v_L^{rel}(\tilde{r} \hat{=} 0.75)}}_{:=St} \leq 0 \quad (2.16)$$

where  $\check{f}_{1,2}^E(n \hat{=} 100\%)$  are the lower limits of the first and second eigenfrequency at 100% engine speed,  $c^{max}(\tilde{r} = 0.75)$  is the chord length, and  $v_L^{rel}(\tilde{r} = 0.75)$  is the relative speed of attack at 75% blade height. Two additional flutter criteria  $b_{14}$  and  $b_{15}$  are utilized which, however, are confidential.

Table 2.1: Penalty factors assigned to eigenmode types shown in Fig. 2.6

$i$	1	2	3	4	5	6	7	8	9	10	11	12	13	...
mode type	1B	1T	2B	2T	1C	1H	1S	3T	3B	2H	2C	3H	4B	unkn.
$w_i^E$	150	140	130	120	110	100	100	100	100	100	50	0	0	0

Table 2.2: Engine-order penalty factors

$j$	excitation source	$w_j^A$
1	no. of vanes (blades) on next row downstream	50
2	no. of vanes (blades) on next row upstream	100
3	no. of vanes (blades) on third row upstream	100
4	difference of no. of vanes (blades) between next row up- and downstream	25
$\vdots$	lower EO's	100



### Static Stresses

Static stresses  $\sigma_{ij}$ ,  $i, j \in \{1, 2, 3\}$ , where  $i = j$  represents normal stresses and  $i \neq j$  shear stresses, are calculated using FEM, see Chapter 3. Since the stresses  $\sigma_{ij}$  depend on the choice of the coordinate system, an invariant measure is needed to make the three-dimensional stress states of different FE models comparable to the yield strength  $R_{p02}$  of the material (maximum strength before permanent deformation will occur). The von Mises stress (Mises (1913))

$$\sigma_v := \frac{1}{\sqrt{2}} \sqrt{(\sigma_{11} - \sigma_{22})^2 + (\sigma_{11} - \sigma_{33})^2 + (\sigma_{22} - \sigma_{33})^2 + 6(\sigma_{12}^2 + \sigma_{23}^2 + \sigma_{13}^2)} \quad (2.17)$$

is such an invariant measure, where it is assumed that failure is caused by the maximum distortion of the material and not by volumetric changes. These assumptions hold for ductile materials such as metals, which is well suited here for assessing stresses of airfoils.

In order to prevent crack initiation, the maximum von Mises stresses at the leading edge  $\sigma_{vL}^{max}$  and trailing edge  $\sigma_{vT}^{max}$  are not allowed to exceed a certain limit  $\hat{\sigma}_v$ . Further, the overall maximal von Mises stress  $\sigma_v^{max}$  should not exceed a certain percentage  $\alpha_{p02} < 1$  of the yield strength  $R_{p02}(T^{max})$  at maximum possible operation temperature  $T^{max}$  and full speed ( $n \hat{=} 105\%$ ) in order to guarantee structural integrity. In total, this gives three inequality constraints

$$b_{16} := \sigma_{vL}^{max} - \hat{\sigma}_v \leq 0, \quad b_{17} := \sigma_{vT}^{max} - \hat{\sigma}_v \leq 0, \quad b_{18} := \sigma_v^{max} - \alpha_{p02} R_{p02}(T^{max}) \leq 0. \quad (2.18)$$

### Fatigue Resistance

Fatigue resistance is considered by ensuring that the combination of alternating stress  $\sigma_a$  and mean stress  $\sigma_m$  (both are von Mises-stresses), i.e.,  $\sigma_v = \sigma_m + \sigma_a$  remains within the endurance limit of the Haigh-diagram, red line in Fig. 2.7 which is an approximation known from experimental studies. As explained by Nicholas (2006), the Haigh-diagram with mean stress as abscissa and alternating stress as ordinate is constructed from the materials yield strength  $R_{p02}(T^{max})$  and tensile strength  $R_m(T^{max})$  drawn on both axes. Together with the maximum tolerable alternating stress  $\hat{\sigma}_a(T^{max})$  ( $10^7$  load cycles before failure) at zero mean stress, this yields the red endurance limit in Fig. 2.7. The value of  $\hat{\sigma}_a$  is an average result of numerous uniaxial tensile tests, where probes may differ significantly from the actual material condition w.r.t. corrosion, erosion, surface roughness, and notching effects. Thus, the alternating stress  $\sigma_{a,i}$  of the  $i^{\text{th}}$  eigenmode calculated using FEM is required to stay below the possible  $\sigma_a^*(\sigma_{m,i})$ , where  $\sigma_{m,i}$  is the mean stress of the  $i^{\text{th}}$  eigenmode received from FEA. The associated inequality constraints then read as

$$b_{18+i} := \frac{\sigma_{a,i}}{\sigma_a^*(\sigma_{m,i})} - \hat{L}_{FR} \leq 0, \quad (2.19)$$

where  $\hat{L}_{FR} \leq 1$  is a safety limit on the dynamic stress ratio and  $i \in \{1, \dots, 15\}$  refers to the first 15 eigenmodes of an airfoil.

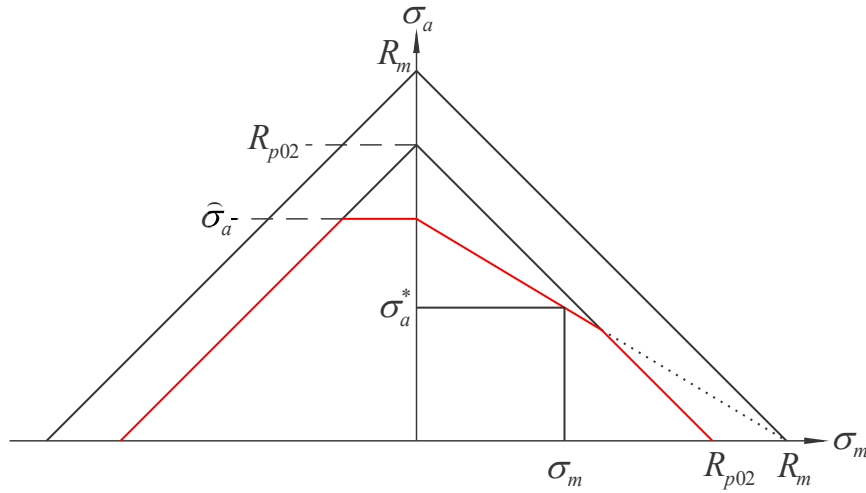


Figure 2.7: Haigh diagram with endurance limit marked by red line

### Geometric Constraints

There are also geometric constraints in order to ensure leading- and trailing-edge radii which can be manufactured, i.e.,  $R_{L/T} \geq \check{R}$ , where  $\check{R}$  is the minimum producible radius, and are less prone to crack initialization due to erosion by demanding e.g.  $R_{L/T} \geq \alpha t^{max}$  with  $0 < \alpha < 1$ . Such constraints are not incorporated directly into the optimization algorithm due to the indirect dependency on  $t^{max}$  and spline-control points. Instead, the spline distributions of  $R_{L/T}(\tilde{r})$  and  $t^{max}(\tilde{r})$  are evaluated for each design and checked whether they violate the geometric constraints or not ("feasibility check" in Fig. 2.1). In case one of the edges is not feasible, the design will not be evaluated by the slave processes and all objectives and constraints are set to high penalty values.

## 2.4 Robust Design Optimization

The constraints  $b_i \leq 0$  in Section 2.3 limit the feasible design space  $\mathcal{P} \subset \mathbb{R}^{57}$ , and without consideration of robustness criteria, optimal designs will most probably be located on the borders of constrained regions where at least one  $b_i = 0$ , see Flassig (2011). Therefore, variations w.r.t. such nominal design parameters, e.g. due to production tolerances, will cause many produced parts to violate these constraints, but also may fail to perform as predicted due to degradation of design objectives. Thus, design robustness must be ensured by requiring that variances of the design objectives w.r.t. production inaccuracies are as small as possible and that constraints are not violated by any produced part within specified production tolerances.

Measuring the design robustness of a nominal design  $\mathbf{p}^{nom}$  w.r.t. constraints is usually done by calculating the failure rate which is the percentage of designs failing at least one constraint  $b_i \leq 0$  in a sample  $\mathcal{D} := \{\mathbf{p}_1^\Delta, \dots, \mathbf{p}_{N_S}^\Delta\}$  of  $N_S$  random points in the neighborhood  $\mathbf{p}^{nom} \pm \Delta\mathbf{p}$  of  $\mathbf{p}^{nom}$ . However, the examples of infeasible design regions A and B (where at least one  $b_i > 0$ ) in Fig. 2.8a show that the failure rate may be a problematic robustness

measure, because in both cases the sample has the same failure rate, but parameter tolerances  $\Delta p_2^A$  and  $\Delta p_2^B$ , within which variations do not fail at all, are different. Therefore, a proper approach for robust design assessment should not analyze failure rate in the design-parameter space in the first place, but analyze if variations to the geometry within a specific production tolerance  $\pm s_\perp$  violate any constraints, see Fig. 2.8b. In case none of these geometry variations violate any constraints,  $\mathbf{p}^{nom}$  may be considered to be robust within the specific tolerance  $\pm s_\perp$ .

It may be difficult to sample and assess geometry variations within a defined production tolerance  $\pm s_\perp$  only, due to the nonlinearity of that region in the parameter space  $\mathcal{T}$  which only encloses all possible geometry variations of the nominal design  $S(\mathbf{p}^{nom})$  within the tolerance band  $\Omega(\mathbf{p}^{nom}, s_\perp)$ , see gray region in Fig. 2.8b&c. In order to identify a subset  $\mathcal{T} \subseteq \mathcal{D}$  of designs  $\mathbf{p}_j^\Delta$ , the corresponding shapes  $S(\mathbf{p}_j^\Delta)$  (e.g. white curve in Fig. 2.8b) are build up as CAD-models and checked whether they are located within the tolerance band  $\Omega(\mathbf{p}^{nom}, s_\perp)$  (gray band in Fig. 2.8b) of the nominal shape  $S(\mathbf{p}^{nom})$  (black curve in Fig. 2.8b). Consequently, constraint violation is checked for these sample points  $\mathbf{p}_j^\Delta \in \mathcal{T}$  only.

In order to find the maximal possible production tolerance  $\pm s_\perp^{max}$ , where design variations  $\mathbf{p}_j^\Delta \in \mathcal{T}$  do not violate any other constraints than the nominal design  $\mathbf{p}^{nom}$ , Martin et al. (2019) introduced following iterative procedure:

1. Specify  $s_\perp := \Delta s_\perp$ , where  $\Delta s_\perp$  may be the minimal feasible tolerance step that can be manufactured, e.g.  $\Delta s_\perp = 0.01\text{mm}$ .
2. Create a random sample  $\mathcal{D}$  of design variations within an adequately large parameter

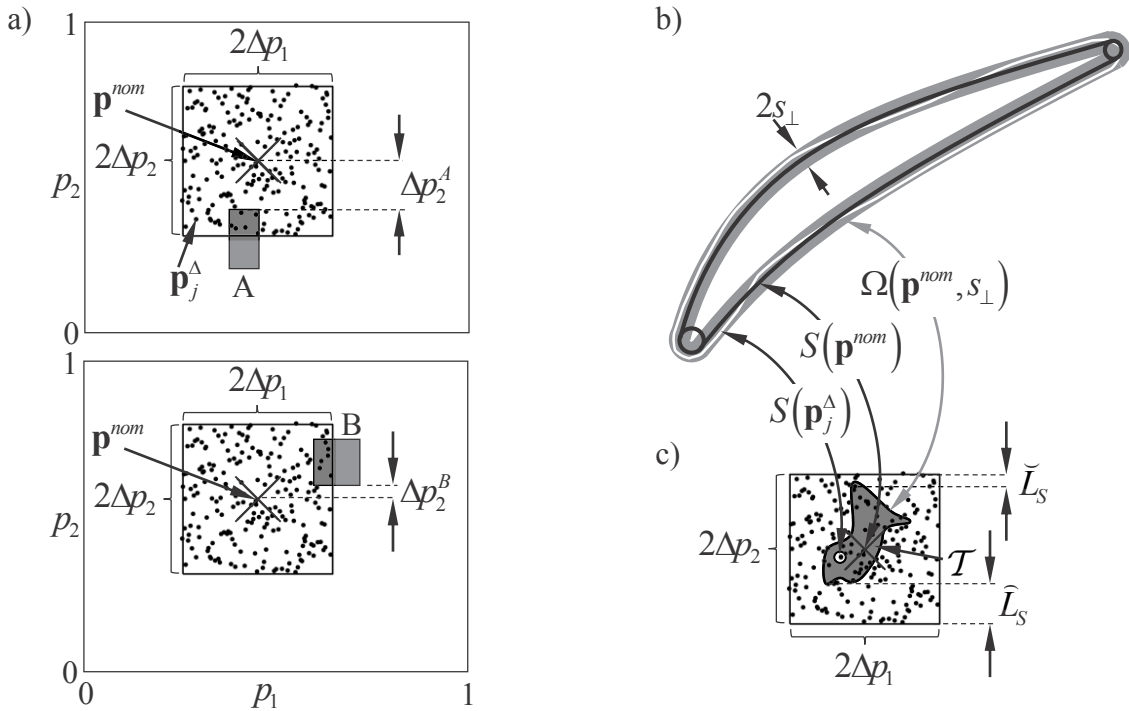


Figure 2.8: Robustness assessment with a) failure rate and b) production tolerance  $\pm s_\perp$

range  $\mathbf{p}_{nom} \pm \Delta\mathbf{p}$ .

3. Identify all those designs from  $\mathbf{p}_j^\Delta \in \mathcal{D}$  where the corresponding airfoil shapes are located within the tolerance hull, i.e.,  $S(\mathbf{p}_j^\Delta) \in \Omega(\mathbf{p}^{nom}, s_\perp)$ , in order to receive a design subset  $\mathcal{T}$ .
4. Calculate the number of design variations  $\mathbf{p}_j^\Delta \in \mathcal{T}$  which violate any additional constraints compared to  $\mathbf{p}^{nom}$ , i.e.

$$\Delta N_V(\mathbf{p}^{nom}, s_\perp) := \left| \left\{ \mathbf{p}_j^\Delta \in \mathcal{T} \mid \exists i : b_i(\mathbf{p}^{nom}) \leq 0 \wedge b_i(\mathbf{p}_j^\Delta) > 0 \right\} \right|. \quad (2.20)$$

5. If  $\Delta N_V(\mathbf{p}^{nom}, s_\perp) = 0$ , increase tolerance  $s_\perp := s_\perp + \Delta s_\perp$  and continue from step 2; else maximum tolerance associated with  $\mathbf{p}^{nom}$  is obtained as  $s_\perp^{max} = s_\perp - \Delta s_\perp$ .

In order to ensure an adequately large parameter range, hence, minimize risk of incomplete sampling in step 2, it should be verified that the minimum distance of any  $\mathbf{p}_j^\Delta \in \mathcal{T}$  from the border of the sampling neighborhood is at least above an user-defined gap value  $\check{L}_S$ , see Fig. 2.8c; else  $\pm\Delta\mathbf{p}$  has to be adapted. Simultaneously, the sampling range in step 2 should not be set too large in order to prevent insufficient sampling density. Thus, it should be ensured that at each sampling border the closest  $\mathbf{p}_j^\Delta \in \mathcal{T}$  has a distance below an user-defined maximum gap value  $\hat{L}_S$ . Both these limiters are another advantage of this assessment of production tolerance over the commonly used failure rate, since that way sampling bounds  $\pm\Delta\mathbf{p}$  are no arbitrarily chosen values anymore.

In contrast to Dow and Wang (2015) who suggested a method for qualitative production-tolerance assessment to be applied to optimization, the method presented here allows for a more practical quantitative assessment. However, although the calculated possible tolerance  $s_\perp^{max}$  presents a valuable criterion for design robustness, it will not be incorporated directly because of its discrete nature ( $s_\perp^{max} = k\Delta s_\perp$ ,  $k \in \mathbb{N}$ ) which may harm optimization performance. A smooth measure, however, may be received by using the failure rate as follows. Due to the production tolerance assessment it is known that the failure rate of designs within  $\pm s_\perp^{max}$  is zero. Thus, the failure rate within the set of samples  $\mathcal{T}$  for the next but in-robust tolerance step  $s_\perp^{max} + \Delta s_\perp$  is an estimator of how much of this non-robust tolerance step might be possible. Thus, by calculating the failure rate

$$F_R := \frac{\Delta N_V(\mathbf{p}^{nom}, s_\perp^{max} + \Delta s_\perp)}{|\mathcal{T}|}, \quad (2.21)$$

a continuous minimization objective for the production tolerance may be defined as

$$\tilde{s}_\perp^{max} := - (s_\perp^{max} + (1 - F_R) \Delta s_\perp). \quad (2.22)$$

Obviously the suggested approach for evaluating  $s_\perp^{max}$  may be computationally expensive, because it requires evaluation of the constraints  $\mathbf{b}_i(\mathbf{p}_j^\Delta)$  for each  $\mathbf{p}_j^\Delta \in \mathcal{T}$  by building up CAD models to receive  $S(\mathbf{p}_j^\Delta)$  for each  $\mathbf{p}_j^\Delta \in \mathcal{D}$  about each nominal design

$\mathbf{p}^{nom}$ . Thus, for a practical application of this method to the example in Chapter 7 reduction of computational effort is required. Therefore, calculation of the constraints  $\mathbf{b}_i$  is accelerated by utilizing the RSMs of the optimization process (Fig. 2.1), cutting the required time for a robust-design evaluation (including buildup of CAD-models) of one nominal design  $\mathbf{p}^{nom}$  down to a few hours, which makes it applicable to the evaluation of a few selected nominal designs. However, in order to make assessment of production tolerances applicable to design optimization in this thesis, limited computational resources (number of CPUs) require to avoid generation of CAD model  $S(\mathbf{p}_j^\Delta)$  for each sample  $\mathbf{p}_j^\Delta$  to identify subset  $\mathcal{T} \subset \mathcal{D}$ . This is achieved by rather evaluating the spline distributions of  $\mathbf{p}_j^\Delta$  for intersections with  $\mathbf{p}^{nom} \pm s_\perp$  with the drawback that only parameters can be sampled which are directly related to  $s_\perp$  and are decoupled from others, namely  $R_L$ ,  $R_T$ , and  $t^{max}$ . Since splines can be calculated simultaneously for all  $\mathbf{p}_j^\Delta \in \mathcal{D}$  and much faster than CAD models can be build up, the assessment of  $s_\perp^{max}$  becomes applicable to the optimization process (evaluation of  $s_\perp^{max}$  for 1000 nominal design takes about 2-3 hours)

Despite considering design robustness w.r.t. constraints, robustness w.r.t. the design objectives  $\Delta p_t$  and  $\omega_{OD}$  in Eq. (2.10) has to be considered in a probabilistic manner as well. This is typically done by minimizing mean and variance of each objective, because smaller variance is associated with lower sensitivity to design variations, Flassig (2011). However, splitting each objective into its mean and variance doubles the number of design objectives and thus harms optimization performance. Instead, as suggested by Du et al. (2004), the mean and variance are substituted with 95%-percentiles  $P_{\Delta p_t}^{95}$  and  $P_{\omega_{OD}}^{95}$ , which represent the corresponding value where 95% of the observations result in lower values. Finally, the deterministic optimization problem (2.10) is transferred into a probabilistic optimization problem:

$$\min_{\mathbf{p}^{nom} \in \mathcal{P}} \begin{bmatrix} P_{\Delta p_t}^{95} + C \\ P_{\omega_{OD}}^{95} + C \\ \tilde{s}_\perp^{max} + C \end{bmatrix} \quad \text{s.t. } \mathcal{P} := \left\{ \mathbf{p}^{nom} \in \mathbb{R}^{57} \mid \check{\mathbf{p}} \leq \mathbf{p}^{nom} \leq \hat{\mathbf{p}} \right\} . \quad (2.23)$$

As percentile are a statistical measure, suitable sampling size and range  $\Delta \mathbf{p}$  have to be defined. However, first the number of parameters shall be reduced in order to increase sampling density. Therefor, only design parameters related to production tolerances are sampled. This means that the ordinate-parameter values of  $\beta_L$ ,  $\beta_T$ ,  $\tilde{\beta}$ , and  $\tilde{t}$  in Fig. 2.3 are not considered, because their variations result from production tolerances in the first place. For the remaining parameters, sampling ranges are set to  $\Delta p_i := 0.05\text{mm}$ , which is five times the minimum step size  $\Delta s_\perp$  utilized during the assessment of the robust production tolerance  $s_\perp^{max}$ . This sampling range is used for all nominal designs  $\mathbf{p}^{nom}$  in order to ensure comparability. The samples for both objectives and constraints are generated using Latin-hypercube sampling (McKay et al. (1979)) with a size of  $N_S := 1000$ , because the deviation of the calculated objectives compared to these based on 5000 samples is less than 3%, as shown in Fig. 2.9 for the objectives.

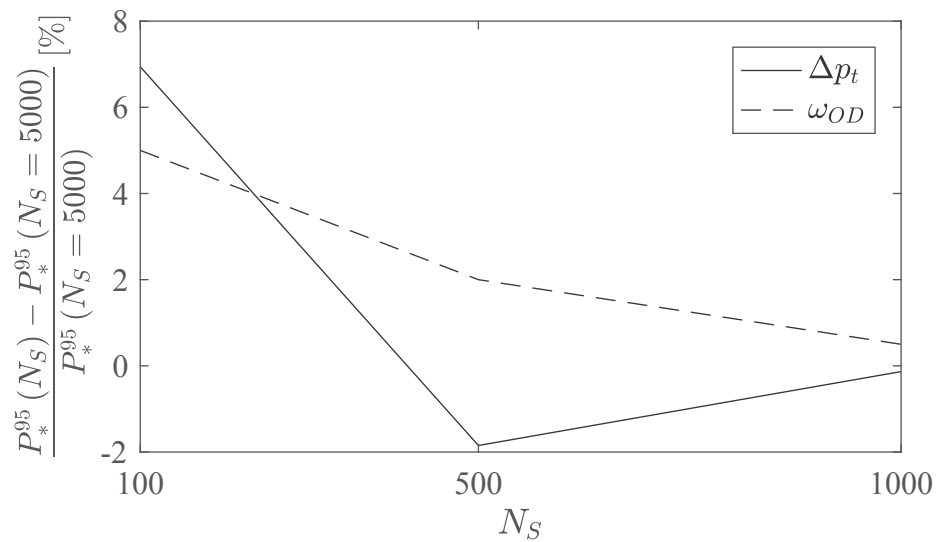


Figure 2.9: Average objective errors (calculated from all nominal designs of a DoE with 3000 samples) for different sample sizes  $N_S$  w.r.t. sampling with  $N_S := 5000$

# 3 Structural Assessment of Compressor Airfoils Using the Finite-Element Method

Finite-element models are used throughout the thesis to provide the necessary information on stresses and eigenmodes. Therefore, firstly the fundamental formalism of finite-element methods (FEM) will be introduced to provide a basic understanding and explanation on the need for nonlinear analysis for the calculation of eigenfrequencies, although no nonlinearities might be present. Afterwards, the mesh types and mesh topology for the present compressor airfoils will be explained, followed by details about load and boundary conditions. Finally, the necessity for loaded-to-unloaded transformation will be verified and an innovative method will be introduced which can be directly incorporated into FE analysis in contrast to the current state of technology.

## 3.1 Fundamental Equations

Within the scope of this thesis, finite-element analyses shall be used to evaluate the stress and strain fields of airfoils by calculating their three-dimensional displacement field  $u_i(x, y, z)$ ,  $i = 1, 2, 3$ . The airfoils are exposed to temperature changes  $T - T_{ref}$  and external volume  $\mathbf{f}^V = [f_1^V, f_2^V, f_3^V]^T$ , surface  $\mathbf{f}^S = [f_1^S, f_2^S, f_3^S]^T$ , and concentrated loads  $\mathbf{f}_j^C = [f_{j1}^C, f_{j2}^C, f_{j3}^C]^T$ . Since complex displacement fields cannot be described by single analytic function, the continuum with volume  $V_B$  and surface  $S_B$  is discretized by  $N_E$  elements with volumes  $V_B^{(k)}$  and surfaces  $S_B^{(k)}$  where  $k \in \{1, \dots, N_E\}$ . The displacements of all  $N_N$  element nodes are combined in a single node-displacement vector  $\mathbf{U} \in \mathbb{R}^{3 \times N_N}$ , and an element-displacement-interpolation matrix  $H_{is}^{(k)}(x, y, z)$  is used to calculate the continuous displacement field of each element ( $k$ ) based on the node displacements  $U_s$ , hence

$$u_i^{(k)}(x, y, z) = H_{is}^{(k)}(x, y, z) U_s, \quad (3.1)$$

considering summation over identical indices (Einstein summation convention) and that  $H_{is}^{(k)}$  equals one at the location of the  $s^{\text{th}}$  node and zero at all other nodes. Generally,  $H_{is}^{(k)}$  depends on the type of element (number of nodes and geometry) and is calculated in

the natural coordinate system of each element before being transformed into the global coordinate system. Here it is assumed that  $H_{is}^{(k)}$  has already been transformed into the global coordinate system. Now the goal is to calculate the node-displacement vector  $\mathbf{U}$ . For an analysis considering nonlinearities, displacements depend on the history of load and boundary conditions. Therefore, nonlinear analysis requires to split the application of loads into incremental calculation steps  $\Delta t$  that can be linearized, which also applies to nonlinear steady-state analyses as performed in this thesis. For this purpose, the following formulations will use superscripts  $t$  and  $t + \Delta t$  to refer to the last known and current unknown configuration, respectively, and subscript  $t$  to refer to an increment between configurations at  $t$  and  $t + \Delta t$ . With this, and assuming that for metal, density can be considered to be constant and damping may be neglected, the equilibrium equations to be solved as described by Bathe (1996) are

$$\begin{aligned}
 & \underbrace{\sum_{k=1}^{N_E} \int_{t+\Delta t V_B^{(k)}} \rho^{(k)} H_{is}^{(k)} H_{ir}^{(k)} dV}_{t+\Delta t M_{sr}} \quad {}^{t+\Delta t} \ddot{U}_s + \\
 & \underbrace{\sum_{k=1}^{N_E} \int_{t V_B^{(k)}} {}^t C_{jimn}^{(k)} {}^t B_{mns}^{L(k)} {}^t B_{jir}^{L(k)} dV}_{{}^t K_{sr}^{L}} \quad {}^t U_s + \underbrace{\sum_{k=1}^{N_E} \int_{t V_B^{(k)}} {}^t \tau_{ji}^{(k)} {}^t B_{jisr}^{NL(k)} dV}_{{}^t K_{sr}^{NL}} \quad {}^t U_s = \\
 & \underbrace{\sum_{k=1}^{N_E} \int_{t+\Delta t V_B^{(k)}} {}^{t+\Delta t} f_i^V {}^{(k)} H_{ik} dV}_{{}^{t+\Delta t} F_r^V} + \underbrace{\sum_{k=1}^{N_E} \int_{t+\Delta t S_B^{(k)}} {}^{t+\Delta t} f_i^S {}^{(k)} H_{ik} dS}_{{}^{t+\Delta t} F_r^S} + \\
 & \underbrace{\sum_{k=1}^{N_E} \sum_j {}^{t+\Delta t} f_{ji}^C {}^{(k)} H_{ik}}_{{}^{t+\Delta t} F_r^C} - \underbrace{\sum_{k=1}^{N_E} \int_{t V_B^{(k)}} {}^t \tau_{ji}^{(k)} {}^t B_{jir}^{L(k)} dV}_{{}^t F_r^I} \quad , \tag{3.2}
 \end{aligned}$$

where

$${}^t B_{mns}^{L(k)} := \frac{1}{2} \left( \frac{\partial H_{ms}^{(k)}}{\partial {}^t x_n} + \frac{\partial H_{ns}^{(k)}}{\partial {}^t x_m} \right) \quad \text{and} \quad {}^t B_{jisr}^{NL(k)} := \frac{1}{2} \left( \frac{\partial H_{ks}^{(k)}}{\partial {}^t x_i} \frac{\partial H_{kr}^{(k)}}{\partial {}^t x_j} + \frac{\partial H_{kr}^{(k)}}{\partial {}^t x_i} \frac{\partial H_{ks}^{(k)}}{\partial {}^t x_j} \right) \tag{3.3}$$

are the linear and nonlinear strain-displacement-interpolation matrices, and  ${}^t C_{jimn}$  is the incremental stress-strain tensor. This tensor  ${}^t C_{jimn}$  is constant for linear elastic materials such as metals, and can be calculated using the Kronecker-delta  $\delta_{ij}$  ( $\delta_{ij} = 1 \forall i = j$  and  $\delta_{ij} = 0 \forall i \neq j$ ), elastic modulus  $E$ , and Poisson's-ratio  $\nu$  as

$${}^t C_{jimn}^{(k)} = \frac{E\nu}{(1+\nu)(1-2\nu)} \delta_{ij} \delta_{mn} + \frac{E}{2(1+\nu)} (\delta_{jn} \delta_{im} + \delta_{jm} \delta_{in}) \quad . \tag{3.4}$$



Equation (3.2) may be written in vector format as

$${}^{t+\Delta t}\mathbf{M} {}^{t+\Delta t}\ddot{\mathbf{U}} + ({}^t\mathbf{K}^L + {}^t\mathbf{K}^{NL}) {}^t\mathbf{U} = ({}^{t+\Delta t}\mathbf{F}^V + {}^{t+\Delta t}\mathbf{F}^S + {}^{t+\Delta t}\mathbf{F}^C) - {}^t\mathbf{F}^I, \quad (3.5)$$

where the mass matrix can be calculated based on the initial state because displacement-based isoparametric elements are being used, i.e.,  ${}^{t+\Delta t}\mathbf{M} = {}^0\mathbf{M} = \mathbf{M}$ , and  ${}^t\mathbf{F}^I$  accounts for initial element stresses, e.g. due to temperature changes where  ${}^t\tau_{ji}^{(k)}$  has to be calculated based on the thermal-strain matrix and  ${}^tC_{jimm}^{(k)}$ . Equation (3.5) may be solved implicitly for each time step using, e.g., Newton-Raphson Scheme (Kreyszig et al. (2011)), while the integrals in Eq. (3.2) may be calculated using Gauss integration (Golub and Welsch (1969)), and  ${}^{t+\Delta t}\ddot{\mathbf{U}}$  may be approximated, e.g., using central differences, i.e.

$${}^{t+\Delta t}\ddot{\mathbf{U}} = ({}^{t-\Delta t}\mathbf{U} - 2{}^t\mathbf{U} + {}^{t+\Delta t}\mathbf{U}) / \Delta t^2. \quad (3.6)$$

As mentioned before, Eq. (3.5) is used for both unsteady and steady analyses. In case of the latter, inertia effects are neglected by removing the first term from Eq. (3.5), and the time steps refer to successive increase of the loads until the final static conditions are established. With the displacement field obtained as the solution to Eq. (3.5), the incremental strain field can be calculated, see Bathe (1996). The incremental stress field  ${}^t\tau_{ji}^{(k)}$  is then calculated based on the incremental strain field and a suitable chosen stress-strain relation, e.g., Hooke's law.

The focal point of this thesis is to improve the structural dynamic behavior of compressor airfoils. Thus, calculation of eigenfrequencies and eigenmode shapes is required. Therefore, the natural response of a system is calculated by using (3.5) and neglecting external forces and initial displacements, i.e.

$$\mathbf{M}\ddot{\mathbf{U}} + ({}^{t_1}\mathbf{K}^L + {}^{t_1}\mathbf{K}^{NL}) \mathbf{U} = \mathbf{0}, \quad (3.7)$$

where  $t_1$  refers to the final state of an antecedent static calculation. The consideration of nonlinear effects via  ${}^{t_1}\mathbf{K}^{NL}$  in an antecedent static calculation allows to account for stiffening effects onto eigenfrequencies through loads (see Eq. (3.2)). This is the main reason for performing a nonlinear structural analysis on airfoils, since other nonlinear effects are generally negligibly small. This fact will be useful for developing a fast loaded-to-unloaded transformation method later on.

The general solution of the linear differential equation (3.7) is of the form

$$\mathbf{U} = \boldsymbol{\phi} \sin(\omega(t - t_0)) \quad (3.8)$$

where  $\boldsymbol{\phi}$  is a displacement-amplitude vector of same dimension as  $\mathbf{U}$ ,  $\omega$  is the oscillation frequency of the sinus function,  $t$  is the time variable, and  $t_0$  is a time shift. Substituted

into Eq. (3.7) yield the generalized eigenvalue problem

$$\left( {}^{t_1}\mathbf{K}^L + {}^{t_1}\mathbf{K}^{NL} \right) \boldsymbol{\phi} = \omega^2 \mathbf{M} \boldsymbol{\phi} \quad (3.9)$$

with eigenvalues  $\omega_i^2$  and eigenvectors  $\boldsymbol{\phi}_i$ , which may be solved using e.g. the Lanczos-algorithm (Lanczos (1950)).

## 3.2 Finite-Element Model

The structural analysis of compressor airfoils requires a FE model which is sufficiently accurate and accounts for all relevant effects. Most of the airfoils of the test compressor are assembled to the shaft and casing via hammer-foot roots (see Fig. 2.5a) that are held in radial direction by circumferential grooves (Schörner and Bestle (2012)). Depending on the structural dynamic behavior, the first rotor stages are equipped with dovetail roots (see Fig. 2.5b) being more expensive to manufacture, which is why the hammer-foot design is preferred wherever possible. The hammer-foot design, however, suffers higher fixation uncertainties and, therefore, also structural uncertainties. Nevertheless, the mesh topologies are only slightly different and the main difference lies in the boundary conditions to be applied. The airfoils are assembled to a massive shaft and due to its high stiffness it does not need to be included into the analysis. Accordingly, circumferential and axial vibration modes of the shaft do not need to be considered within the airfoil design process, because frequencies of relevant eigenmodes of such shafts are well above any natural airfoil frequency of interest.

### 3.2.1 Topology and Mesh

The mesh of the airfoils with dovetail as well as hammer-foot root contain two different element types: fully integrated triangular prisms of quadratic order (15 nodes) for the airfoil and fully integrated tetrahedrals of quadratic order (10 nodes) for the root, see Fig. 3.1. Preferring second-order (quadratic) over first-order (linear) elements has the advantage that through the quadratic element-displacement-interpolation functions in Eq. (3.1) more complex geometric features and deformations can be captured. Therefore, the stress, strain, and displacement field can be calculated more accurately, which makes quadratic elements less prone to shear locking (MacNeal and Faulkner (1994)) under bending loads (main state of deformation for airfoils). Shear locking describes the artificial occurrence of shear stresses due to insufficient representation of the actual state of deformation by the finite elements. The ability to fit geometric features, e.g. curvatures, better than linear elements allows to use less elements, but for a higher computational effort per element. A possibility to reduce the computational effort is to use reduced-integrated elements instead of fully-integrated ones (Zienkiewicz and Hinton (1976)). Quadratic elements using reduced integration do not suffer from shear locking, but have hourglass modes (zero energy deformation modes) which, however, can hardly propagate through the mesh. Since, the focus of this thesis is structural dynamic design optimization, no mesh and element validation studies were performed, and the decision was made to use

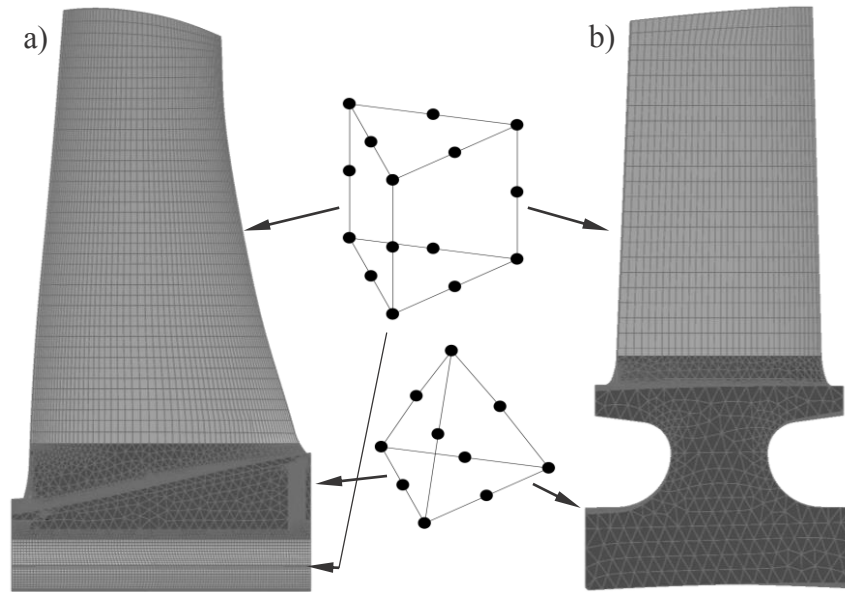


Figure 3.1: Finite-element mesh of a) dovetail root and b) hammer-foot root

fully integrated elements. The mesh density was set according to validations and design experiences of the industrial partner.

### 3.2.2 Loads and Boundary Conditions

The loads experienced by compressor airfoils are mainly centrifugal and pressure loads. Because the latter has only minor influence on stresses and eigenfrequencies (frequency deviation less than 1Hz for first 10 eigenfrequencies of all airfoils of test compressor), pressure loads may be neglected. However, depending on the pressure, the temperature raises downstream of the compressor and becomes a major factor of influence on stresses and eigenfrequencies through decreasing elastic modulus of the material. According to design experience of the industrial partner, the temperature distribution within a compressor row as well as changes due to design modifications are negligible as long as the temperature level is accounted for. Those values can be received with sufficient accuracy from meanline-throughflow analysis, which allows to run decoupled aerodynamic and structural design evaluation (else RSMs would have to be employed for the coupling parameters in order to prevent idling of coupled processes). Nevertheless, different operation conditions have significant influence on temperature levels, and results such as stresses and eigenfrequencies become uncertain. This effect has to be captured by evaluating limits of design values (constraints) based on the extrema of the operation conditions, namely, running on a hot day ( $T^{max}$ ) at electric-network frequency with full power output by the turbine (nominal condition), and running on a cold day at electric network frequency, but without power output by the turbine (idle condition).

A similar problem arises from the fixation of airfoils. Airfoils with dovetail roots (separate groove for each root) have well determined contact with the groove (no transverse displacements) at surface C in Fig. 2.5b and, therefore, are mechanically decoupled from

each other due to the massive shaft. In contrast to this, the ones with hammer-foot roots (single circumferential groove for all roots of one compressor row) are able to perform coupled circumferential and axial vibrations and the stiffness of an assembly determines the number of nodal diameters (Strehlau and Kühhorn (2010)). Due to the uncertain installation condition of hammer-foot roots, i.e., whether one root is in full contact with its neighbors at surface B in Fig. 2.5a, completely free, or something in between with friction damping effects taking place, no discrete nodal diameters can be calculated and possible forced responses cannot be identified using the SAFE-diagram (Singh et al. (1988)). Therefore, the possible range (uncertainty margin) of each eigenfrequency due to uncertain fixation and temperature condition needs to be estimated, see explanation in Section 2.3. The upper limit is set by the stiffest configuration possible, i.e., fully transversally fixed nodes at surface B in Fig. 2.5a at idle conditions (called tight-idle) and completely free nodes at surface B in Fig. 2.5a at nominal conditions (called loose-nominal). In both cases the hammer-foot roots are fully transversally fixed in all directions at the nodes of surface A in Fig. 2.5a. Table 3.1 summarizes the different calculations required to account for uncertain operation and fixation conditions.

Table 3.1: Description of analyses performed for structural design evaluation of airfoils with hammer-foot root

analysis name	description	comment
loose-nominal	adjacent roots have no contact at hot day condition and full turbine power output at electric network frequency	lower limit of eigenfrequency uncertainty range
loose-idle	adjacent roots have no contact at cold day condition and no turbine power output at electric network frequency	used for flutter criterion
tight-idle	adjacent roots have full contact at cold day condition and no turbine power output at electric network frequency	upper limit of eigenfrequency uncertainty range
static		results are included in the loose-nominal analysis; basis for stress results

Because the load and fixation conditions effect the stiffness of the airfoil-root assembly, those effects have to be captured by the analysis. According to Eq. (3.9) such stiffening effects can be considered by performing nonlinear static analysis (gives the nonlinear stiffness matrix in addition to the linear stiffness and mass matrix) prior to modal analysis (solving the eigenproblem (3.9)). Figure 3.2 shows the error between linear and nonlinear analysis for the first 10 eigenfrequencies of a front (dovetail root), a middle, and a rear stage of the test compressor at loose fixation condition. One may conclude that even at loose fixation, where centrifugal loads are the only driver for airfoil stiffening (no contribution

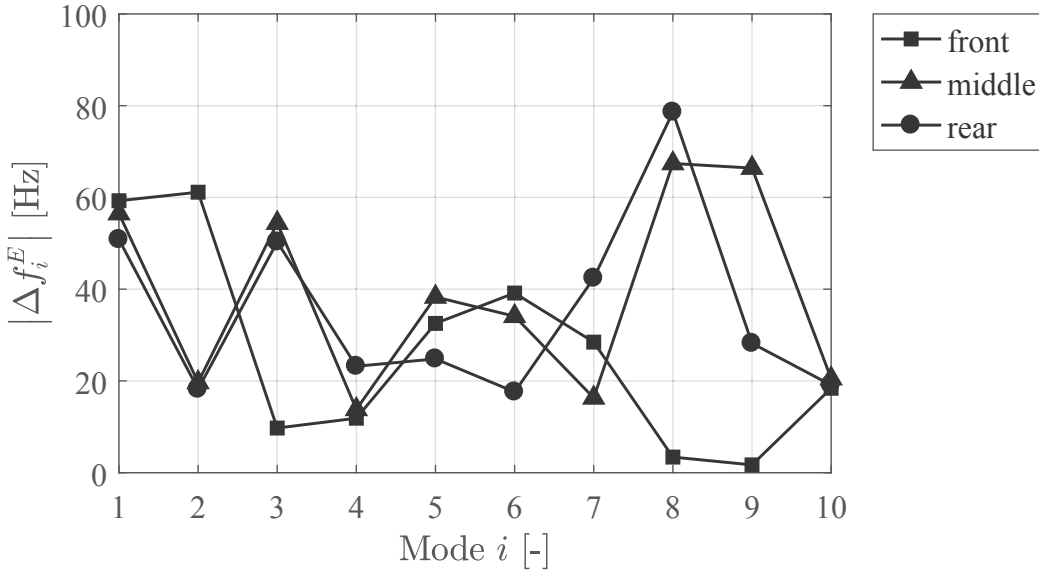


Figure 3.2: Error between linear and nonlinear analysis of first 10 eigenfrequencies for three different stages of the test compressor at loose fixation condition

from fixation stiffness), stiffening effects onto eigenfrequencies cannot be neglected.

### 3.3 Loaded-to-Unloaded Transformation

The aerodynamic optimization of an airfoil is typically based on its loaded hot geometry, i.e., the airfoil geometry during operation conditions. However, structural design evaluation should be applied to the corresponding unloaded cold geometry in order to deform to the intended hot geometry when actual centrifugal and pressure loads as well as temperature conditions are applied. Only when the unloaded cold geometry deforms to the loaded hot geometry, structural and aerodynamic evaluation results will be consistent.

The general method of calculating the corresponding unloaded geometry to a given loaded geometry, as described by Goerke et al. (2012) and shown in Fig. 3.3a, is performed iteratively as follows:

1. Based on the known loaded geometry  $S_0^L$ , assume that the unloaded geometry is identical, i.e.  $S_1^U \equiv S_0^L$ ; set  $i := 1$ .
2. Apply the associated temperature and load conditions to  $S_i^U$  resulting in a corresponding loaded geometry  $S_i^L$ .
3. Subtract the difference  $\Delta_i := S_i^L - S_i^U$  from the given loaded geometry  $S_0^L$  which gives a better estimate  $S_{i+1}^U$  of the unloaded geometry.
4. Set  $i := i + 1$  and continue with steps 2 and 3 until the absolute error  $|S_i^L - S_0^L| \leq \varepsilon$  is below an user-defined limit  $\varepsilon$ ; finally  $S_{i+1}^U$  is the desired estimate of the unloaded geometry corresponding to  $S_0^L$ .

This procedure accounts for nonlinearities as the iterative FE-analyses are nonlinear. However, it is rather time-consuming, because for each step displacements have to be extracted from FE-results and subtracted from the CAD-geometry  $S_0^H$  in order to create the new CAD-geometry (unloaded geometry) and import it into the FE-tool followed by preprocessing (setting boundary and load conditions as well as creating mesh) and FE-analysis.

Design optimization requires a faster, non-iterative method which can preferably be directly integrated into the FE-solver. The most forward approach may be to reverse the load and temperature conditions shown in Fig. 3.3b (pressure  $p_{SS}$  and  $p_{PS}$  on suction and pressure side, centrifugal load  $F_C$ , and temperature change  $\Delta T$ ). This is rather simple for pressure loads ( $-p_{PS}$ ) and ( $-p_{SS}$ ) and temperature conditions  $-\Delta T$ , see Fig. 3.3c, especially in the case where pressure influence is negligible as has been shown by Janke et al. (2016). Reversing centrifugal loads for a finite volume  $dV$ , however, is not that obvious and, as suggested by Martin et al. (2019), can only be achieved by assigning a negative density ( $-\rho$ ) which is applicable in Abaqus. Changing the sign of quantities  $r$  and  $dV$  instead, would change the model geometry, be not applicable, or distort other distributed loads. It should be emphasized that reversing centrifugal loading may cause nonlinear buckling instabilities to slender structures such as airfoils which, however, can be suppressed by performing linear deformation analysis only. The discrepancies in loaded-geometry eigenfrequencies calculated using linear unloaded-geometry estimates are sufficiently small ( $< 3\text{Hz}$ ) for the first 10 eigenfrequencies of the first, middle, and last rotor of an industrial compressor as shown in Fig. 3.4 using black markers.

Without loaded-to-unloaded transformation, i.e., if loads and temperature conditions are directly applied to the loaded geometry, the errors are highly significant (white

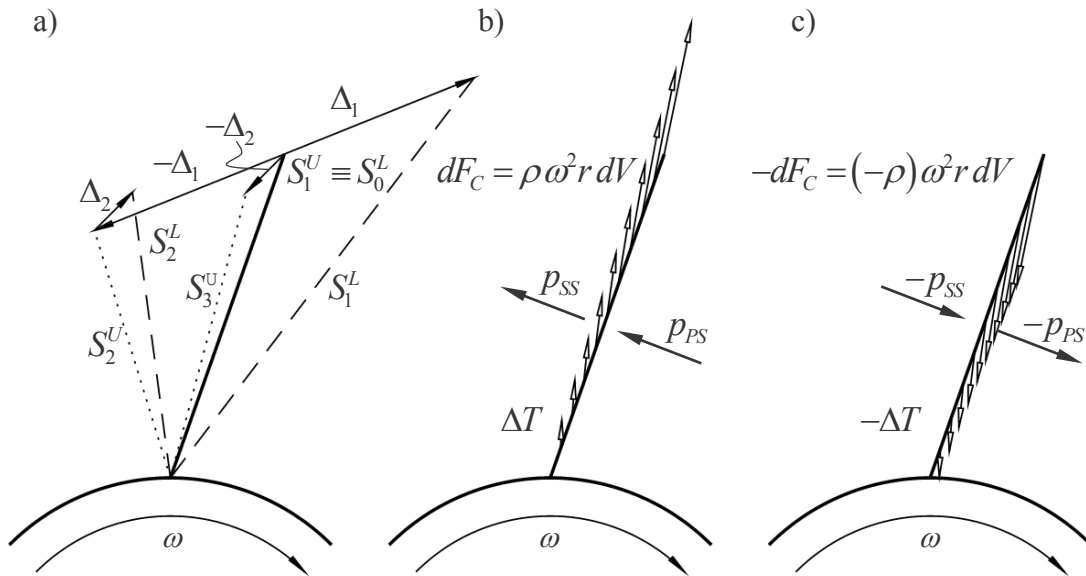


Figure 3.3: Iterative loaded-to-unloaded transformation (a), sketch of airfoil loads (b), and reversed loads (c)

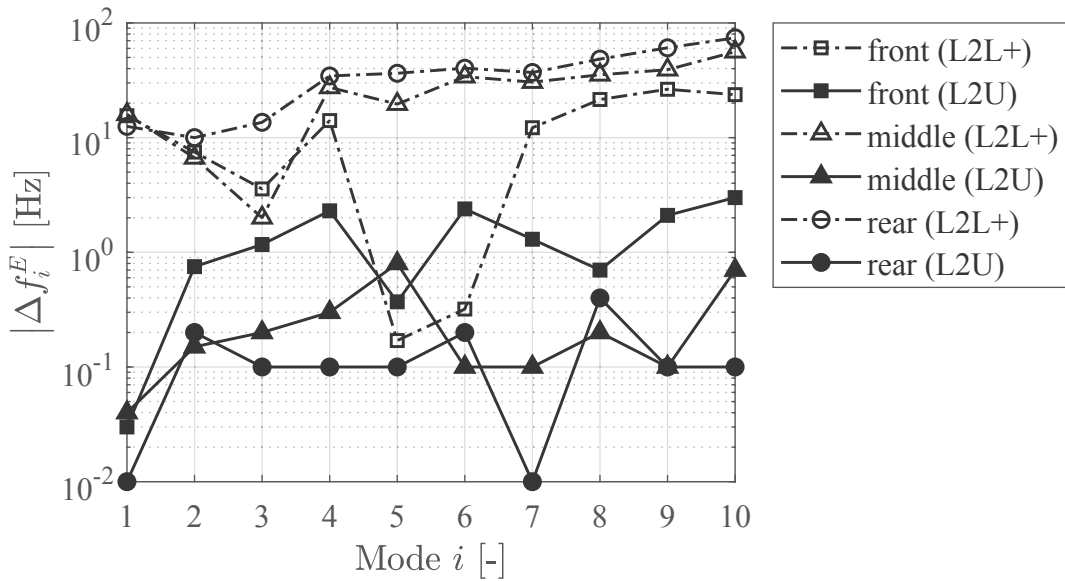


Figure 3.4: Error of first 10 eigenfrequencies computed with linear loaded-to-unloaded transformation (L2U) and without (L2L+) for a) the first, b) middle, and c) last rotor of an industrial gas-turbine

markers for loaded-to-Additionally loaded  $\hat{=}$  L2L+). The error regarding maximal stresses reduces from about 80MPa to less than 2MPa when using linear loaded-to-unloaded transformation. Because the suggested linear loaded-to-unloaded transformation can be directly implemented into FE-analysis (as first calculation step where displacement field has to be set as stress free before continuing with further FE-calculations) and only causes additional computational effort of less than 5 seconds in the present application, loaded-to-unloaded transformation becomes applicable to design optimization.

### 3.4 Influence of Fixation Uncertainties onto Eigenfrequency Bands

As reasoned in Section 2.3, the goal is to assign an eigenfrequency band to each eigenmode shape of interest in Fig. 2.6, which requires to develop a classification method. But, as will be shown, in some cases the classification fails to deliver unambiguous results due to misclassification or extremely deteriorated mode shapes. In these cases, proper decision rules have to be found. Therefore it is necessary to firstly investigate and understand how eigenmode shapes and their eigenfrequency bands change between the limits of prediction, i.e., loose-nominal and tight-idle in Table 3.1. For the dovetail roots of the test compressor, no changes in the order of eigenmode shapes are observed, but significant changes arise for stator and rotor airfoils with hammer-foot roots. In Fig. 3.5 the changes of the first ten eigenmodes shapes and eigenfrequencies of a middle compressor blade (rotor) are shown, where changes of the eigenmode shapes are high for varying fixation conditions (Fig. 3.5a vs. Fig. 3.5b), but small for varying temperature conditions (Fig. 3.5b vs. Fig. 3.5c).

Therefore, and for the sake of simplicity, the following investigation will only consider changing fixation conditions for a rotor airfoil with hammer-foot-root design.

The uncertain installation condition of hammer-foot-root assemblies originates from the uncertain contact condition which results in an uncertain friction condition. A direct analysis of this effect would be computationally too expensive and unlikely to give useful insight into the phenomena. Instead, it is assumed that fixation uncertainty is caused by uncertain stiffness of contact layers (clamping material), which are added to the FE-model according to Fig. 3.6a. The stiffness of the clamping material is changed by varying Young's modulus  $E$  between  $6 \cdot 10^{-6} \cdot E_0$  and the real modulus  $E_0$  of the blade material. Consistent behavior of the different materials under centrifugal loads can be accomplished by adapting the density ratio to the modulus (speed of sound equivalence), i.e.,  $\rho = \rho_0 E / E_0$ . Transversal displacements of all nodes of the outer surfaces of the clamping material, equivalent with surface B in Fig. 2.5a, are set to zero.

The change of eigenfrequencies w.r.t. Young's modulus for the last blade in the test compressor is shown in Fig. 3.6b. Corresponding analyses have also been performed for the blade of stage 5 (Fig. A.1; first stage equipped with hammer-foot roots) and stage 8 (Fig. A.2), but have been moved to Appendix A, since the basic conclusions can already

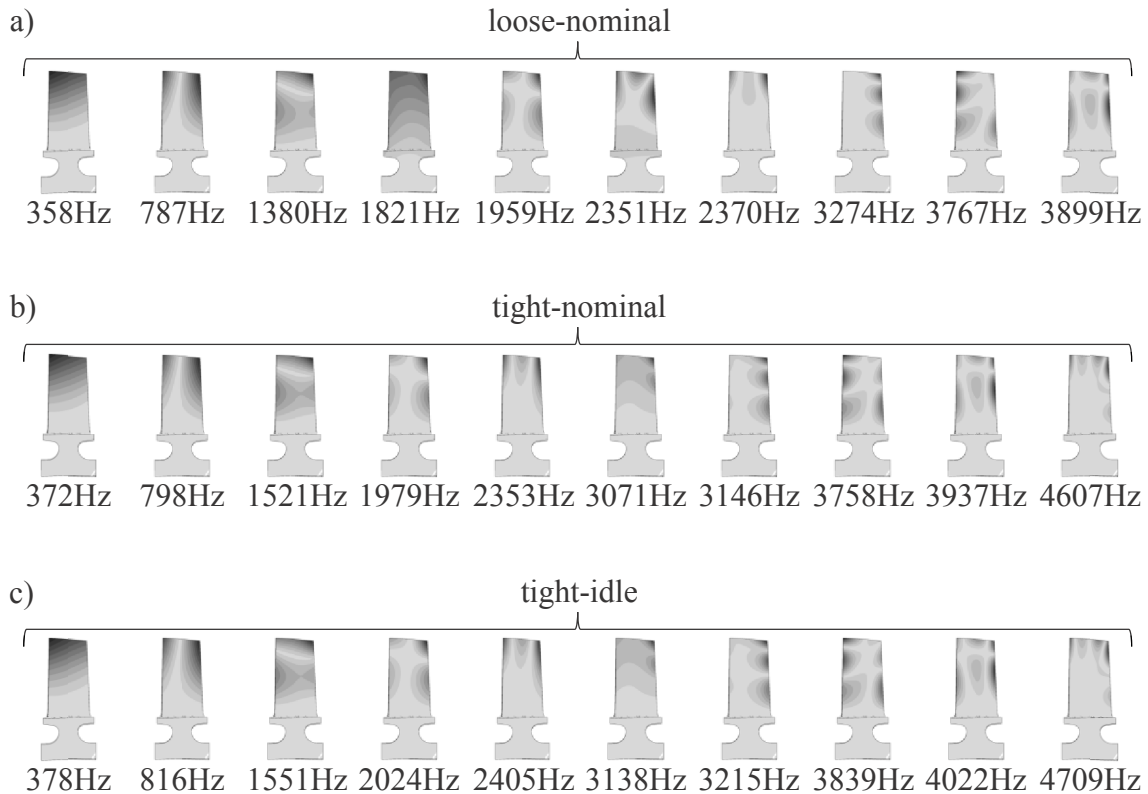


Figure 3.5: First ten eigenmode shapes and eigenfrequencies of a compressor blade (middle stage; dark regions show high absolute displacements) at a) loose fixation and nominal temperature, b) tight fixation and nominal temperature, and c) tight fixation and idle temperature



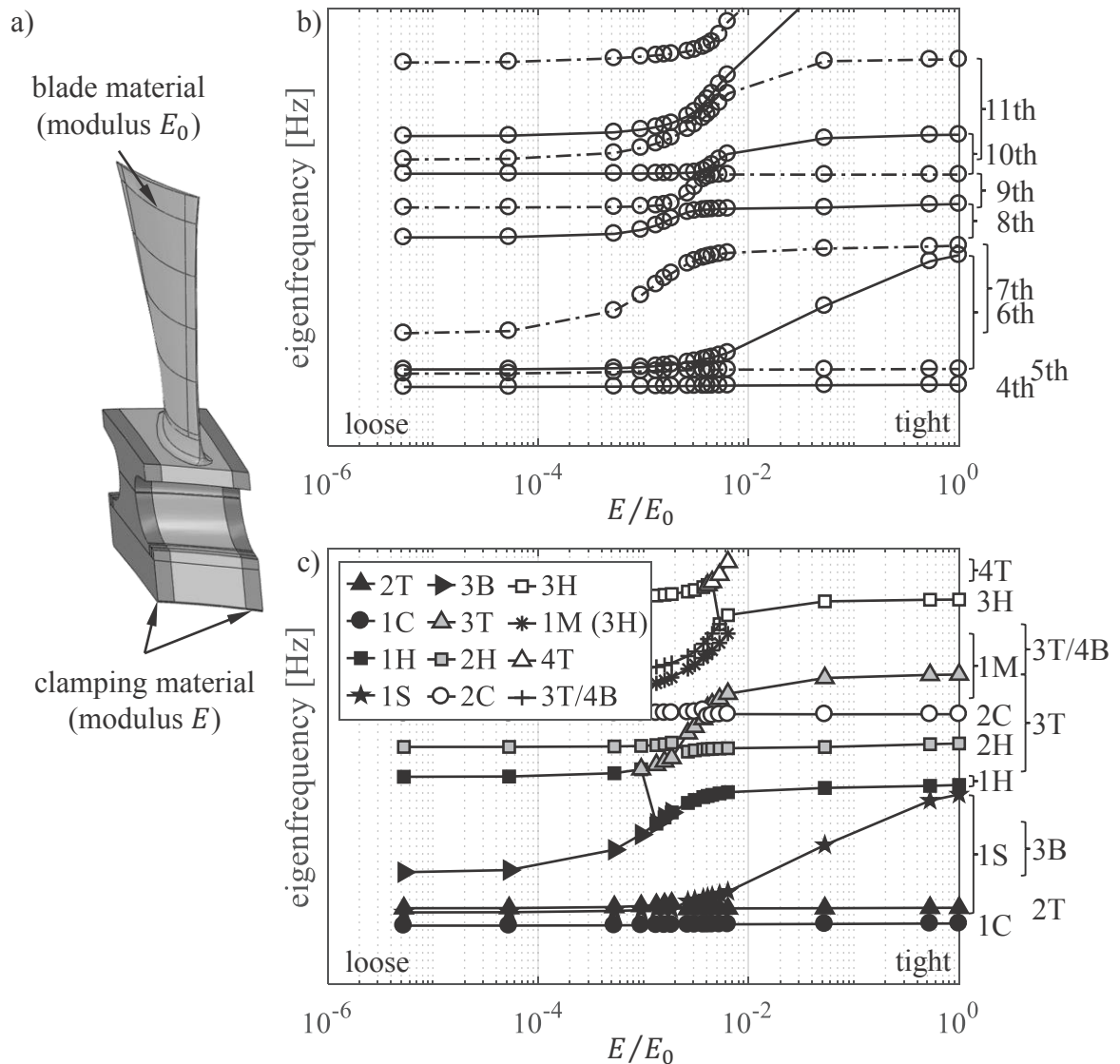


Figure 3.6: Material for fixation study (a) and corresponding eigenfrequencies sorted by b) frequency values and c) eigenmode shapes of a rear rotor blade (stage 15)

be drawn from Fig. 3.6b&c. The focus lies on the region between the fourth and the 13<sup>th</sup> eigenfrequency, because there is hardly any change of the first three eigenfrequencies and corresponding eigenmode shapes. It can be noticed that frequency veering occurs at several regions, meaning that adjacent eigenfrequencies come close before veering away without crossing. The phenomenon is caused by coupling of eigenmodes through the geometry (see Appendix B with an academic example for explanation) which causes fundamental changes in the eigenmode shapes of both veering partners, or as Leissa (1974) stated: "a dragonfly one instant, a butterfly the next, and something indescribable in between". The fact that a complex geometry causes different levels of coupling between eigenmodes, explains the different levels of frequency veering between the eigenmodes in Fig. 3.6b. It can be proven that the physical eigenfrequencies do not cross (see Appendix B), but for eigenfrequencies assigned to specific eigenmode shapes, frequency veering may become

frequency crossing as can be seen in Fig. 3.6c, where the naming convention and relevant mode shapes are given in Fig. 2.6 (additional mode M refers to mixed mode which cannot be categorized). The actual change of the mode shapes during veering is shown in Fig. 3.7. Clearly, classifying the eigenmodes according to their shape is not always unambiguous and it has to be considered that eigenmodes can change their order, vanish, or appear. Thus, a robust classification method complemented by a heuristic eigenfrequency-band assignment strategy has to be developed in order to identify proper mode-shape-specific eigenfrequency bands. Both, classification method and assignment strategy will be explained in Chapter 6. However, the latter relies on conclusions drawn from following observations.

In Fig. 3.6c frequency bands (marked by brackets) have been assigned to specific mode shapes in Fig. 3.7 by hand. It can be seen that 1S reacts most dynamically to an increase of the root stiffness and, therefore, has a higher uncertainty margin than e.g. modes 1H and 3H. Additionally, an increase of uncertainty margin (frequency change) w.r.t. to the estimated limits at loose and tight fixation can be observed within all veering regions. To some extent this discrepancy between frequency change within the veering region and predicted limits at loose and tight fixation correlates with the coupling strength between eigenmodes (see Appendix B). This is particularly true for the 1H mode which couples with several other modes such as torsion, bending, and chord-wise bending modes. Since attempts failed to predict the height of the frequency change based on the frequency limits at loose and tight fixation, due to high prediction errors on the coupling strength, the addition to uncertainty by frequency change will be neglected further on, considering that the stiffness range of occurrence is very narrow and detailed analyses such as this one are impractical during design optimization. For the same reason of impracticality, one cannot track eigenmode shapes that disappear between the upper and lower limit of fixation, e.g. mode 4T in Fig. 3.6c. In case there are eigenmode shapes such as 3T that can only be identified at one bound but vanish at the other, a definition of the corresponding eigenfrequency band depends on the eigenmode shape where, based on Fig. 3.6 (or also Fig. A.2 and A.1 as well), one may reason that:

- a. If 1H mode is identified at the lower bound (here loose fixation) but not at the upper one (here tight fixation), and 1S mode is existent at the upper bound, frequencies of 1H and 1S may be paired to give frequency band of 1H mode. Else, if no 1S mode is identified at upper bound, assume upper limit of 1H-frequency to be 1Hz higher than the one of the lower limit, because usually 1H-bands are very tight, but zero bandwidth would cause no detection by the *MRF*-value.
- b. In case no 3B and/or 1S mode can be identified at the upper bound, but only at the lower one, the closest higher 1H-frequency (loose or tight) may be picked as the corresponding bound, because both veer with 1H for all investigated blade geometries of the considered reference compressor.
- c. If no corresponding match can be found for 1S/2T at the lower bound, the reasonable choice for the corresponding frequency limit at the lower bound is the 2T/1S mode.
- d. Any other mode, that could not be assigned with a counterpart at the opposite frequency bound, should be paired with the nearest frequency of the opposite limit.

---

Finally it should be mentioned that inserts of relatively soft material ( $E/E_0 \geq 10^{-2}$ ) between neighboring airfoils would prevent changes in the order of the eigenmode shapes and reduce uncertainties significantly. This would also avoid the need for the heuristic assignment strategy while mode-shape classification would still be required for the calculation of the *MRF*-value. Such an insert device, however, is not part of the test compressor design.

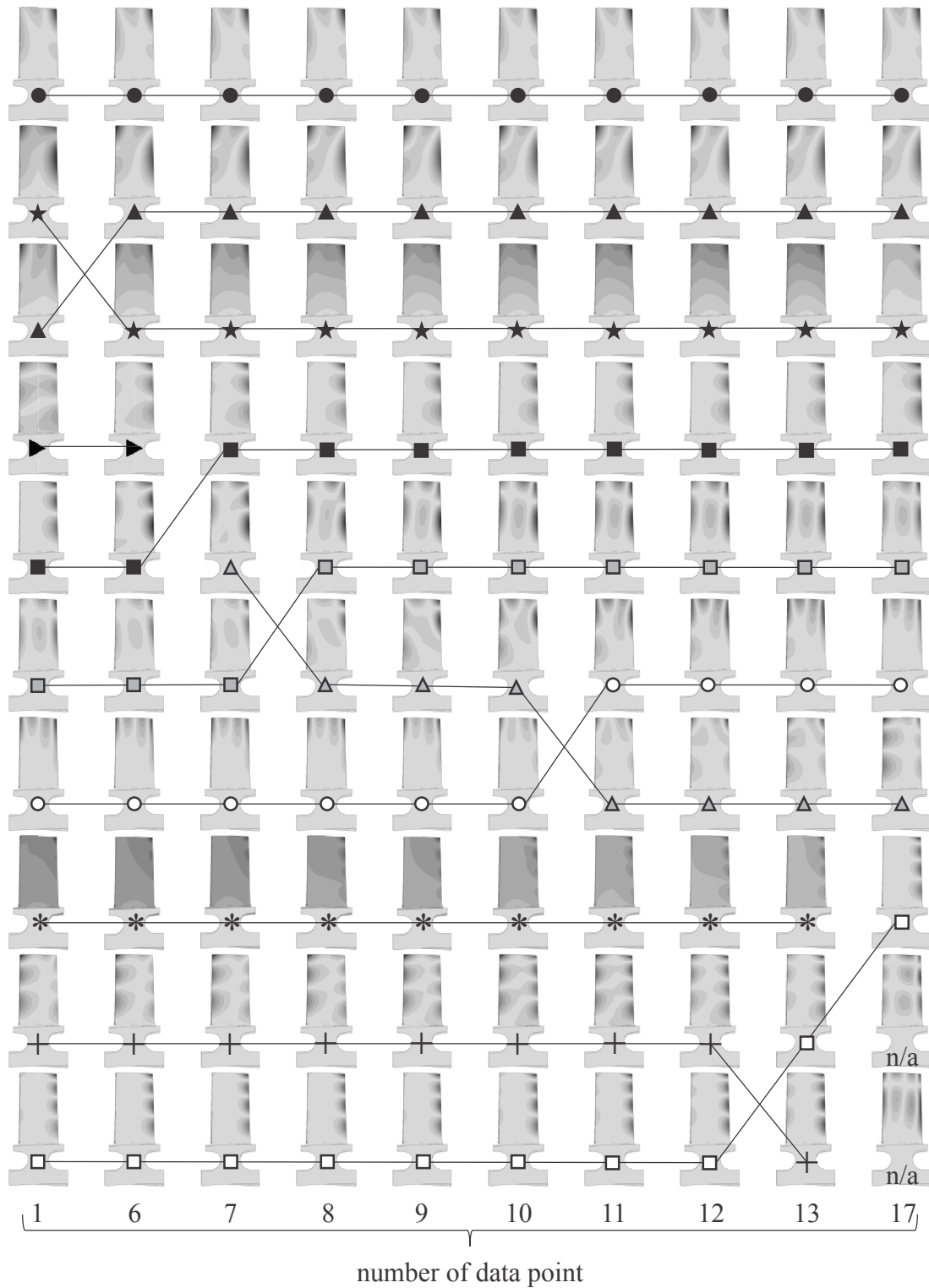


Figure 3.7: Change of the absolute displacement field of the 4th till 13th eigenmode of a rear compressor blade due to varying fixation stiffness (the markers and data point numbers refer to Fig. 3.6)c

# 4 Projection of Arbitrary Airfoil Geometries onto a Rectangular Map

The optimization process is supposed to perform design optimization of airfoils w.r.t. the failure risk of specific eigenmodes which are characterized by the displacement field of the airfoil surface. Therefore, the process must be able to compare displacement fields (eigenmodes) of surfaces from different geometries (airfoil designs), which may be achieved by projecting displacement fields onto a standard rectangular surface. For structured meshes such a projection may be received by using the mesh indices as natural coordinates. For unstructured meshes, however, this is not possible and the current approach of tracking eigenmodes, used by commercial software such as Abaqus, is to use meshes that do not change the indexing of the FE-nodes during design optimization. However, this turns out to be too limiting. Therefore, the goal of this chapter is to introduce a method being able to project any kind of surface mesh, whether structured, unstructured, or mixed, with different resolution and indexing onto a standard rectangular surface, see Fig. 4.1. By this, displacement fields become consistent and can be compared in order to track eigenmodes. The method is independent from a specific FE-software and only requires the node coordinates of the mesh with corresponding displacements or any other variable that is supposed to be mapped.

The method will use a self-organizing network (SOM) to be introduced first. Thereafter,

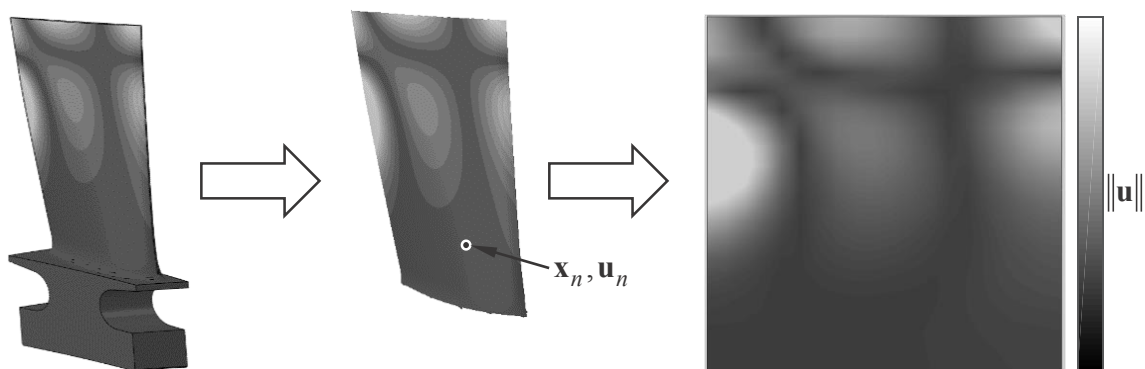


Figure 4.1: Blade surface extraction and projection onto standard square

the methodology of training the network will be explained.

## 4.1 Mapping of Airfoil Data onto Rectangular Map

As mentioned above, the goal is to develop a method which allows for mesh independent comparison of airfoil displacement fields of arbitrary airfoil geometries. The idea is to find a regular mesh approximating an airfoils surface, purely based on the  $N_N$  surface nodes  $\mathbf{x}_n \in \mathbb{R}^3$ ,  $n = 1, \dots, N_N$ , of the FE-mesh (Fig. 4.2) in absences of any knowledge about the spline representation of the underlying design model (CAD model). The indices  $(i, j)$  of such a regular mesh may then be interpreted as natural coordinates of the surface and may be used to project the associated surface coordinates  $\mathbf{x}_{ij} \in \mathbb{R}^3$  onto an invariant standard two-dimensional rectangle or square, see Fig. 4.2. Since the displacements  $\mathbf{u}_n$  of the FE analysis and the corresponding  $\mathbf{u}_{ij} \in \mathbb{R}^3$  on the regular mesh are subject to the same projection  $\mathbf{x}_n \rightarrow \mathbf{x}_{ij} \rightarrow (i, j)$ , an uniform basis for eigenmode identification is given by the squared standard surface.

The problem of finding a regular mesh on a surface from surface points as the only representatives may be accomplished by using a self-organizing map (SOM). SOMs were first invented by Kohonen (1981) as a neural network which maintains the topographic order of its input space and are also referred to as Kohonen-maps. These networks, trained through self-supervised learning, are commonly used for reducing and mining high-dimensional nonlinear data. In order to receive a regular mesh as a representation of an airfoil surface, a rectangular network is formed by neurons  $(i, j)$ ,  $i = 1, \dots, I$ ,  $j = 1, \dots, J$ , where each neuron carries a coordinate information  $\mathbf{x}_{ij}$  as a weight vector, see Fig. 4.3. The network is able to learn the representation of an airfoil surface by being iteratively fed with training vectors  $\mathbf{x}_n \in \mathcal{V}_T$  from a training set  $\mathcal{V}_T = \{\mathbf{x}_1, \dots, \mathbf{x}_{N_N}\}$ . In the present application, these training vectors are the FE-node positions  $\mathbf{x}_n$  on the suction side of an airfoil as shown in Fig. 4.2a. For the network to converge, the members of the training set  $\mathcal{V}_T$  have to be presented to the network multiple times, which requires to reference the

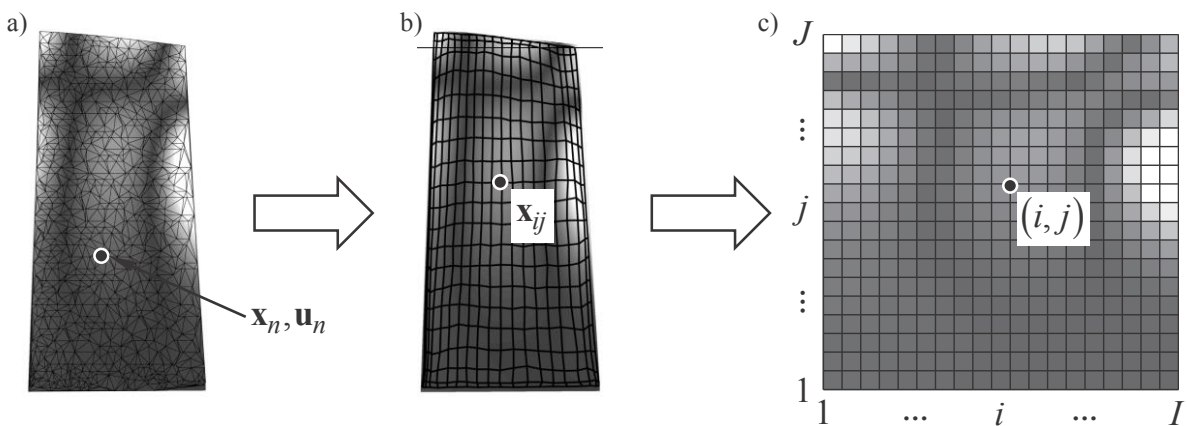


Figure 4.2: Unstructured FE-mesh (a) transferred to structured mesh (b) and projected onto square (c)

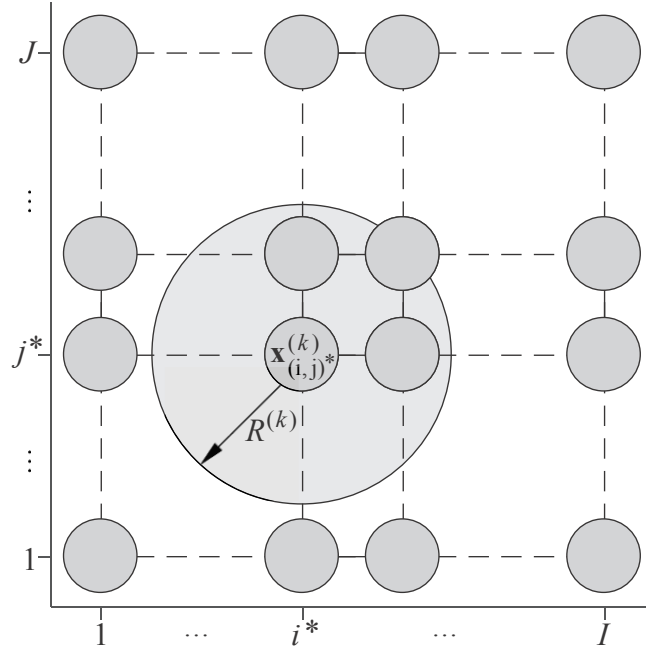


Figure 4.3: Neural network with best matching neuron  $\mathbf{x}_{(i,j)^*}^{(k)}$  and neighborhood radius  $R^{(k)}$

training vectors within  $\mathcal{V}_T$  via an index set  $\mathcal{I} = \{i_1, i_2, \dots\}$  where  $|\mathcal{I}| \gg N_N$ .

After initialization of the neuron weights (detailed information will be provided later), where each neuron  $(i, j)$  is associated with an weight vector  $\mathbf{x}_{ij}^{(k)}$ , the network learns to represent the data in  $\mathcal{V}_T$  by being presented a selected training vector  $\mathbf{x}_{i_k} \in \mathcal{V}_T$  at each iteration step  $k$  (learning step) of overall  $|\mathcal{I}|$  training iterations. During each step  $k$ , the neuron  $(i, j)^*$  with the weight vector  $\mathbf{x}_{(i,j)^*}^{(k)}$  closest to the training vector  $\mathbf{x}_{i_k}$  is identified:

$$(i, j)^* = \arg \min_{(i,j)} \left\| \mathbf{x}_{i_k} - \mathbf{x}_{ij}^{(k)} \right\| . \quad (4.1)$$

After having found the best matching neuron  $(i, j)^*$ , all neurons within a certain distance or neighborhood  $R^{(k)}$  are identified (see Fig. 4.3) and their weight vectors  $\mathbf{x}_{ij}^{(k)}$  are adapted to become more similar to the presented training vector  $\mathbf{x}_{i_k}$  by updating them according to

$$\mathbf{x}_{ij}^{(k+1)} = \begin{cases} \mathbf{x}_{ij}^{(k)} - L_R^{(k)} h_{ij}^{(k)} (\mathbf{x}_{i_k} - \mathbf{x}_{ij}^{(k)}) & \text{if } \|(i, j) - (i, j)^*\| \leq R^{(k)} \\ \mathbf{x}_{ij}^{(k)} & \text{else} \end{cases} \quad (4.2)$$

where  $L_R^{(k)} h_{ij}^{(k)}$  is the adaption rate of the network controlled by the learning rate  $L_R^{(k)}$  and neighborhood function  $h_{ij}^{(k)}$ .

The neighborhood radius

$$R^{(k)} := R^{(0)} \exp\left(-\ln\left(R^{(0)}\right) \frac{k}{|\mathcal{I}|}\right) \quad (4.3)$$

decreases exponentially with the number of iterations  $k$  starting from an initial radius  $R^{(0)}$ , which changes the adaption of the network (4.2) from a flexible global ( $\max_k (R^{(k)}) = R^{(0)}$ ) to a stiff local behavior ( $\min_k (R^{(k)}) = R^{(|\mathcal{I}|)} = 1$ ) which effects only direct neighbors. Accordingly, the learning rate

$$L_R^{(k)} := L_R^{(0)} \frac{R^{(k)}}{R^{(0)}} \quad (4.4)$$

decreases from  $L_R^{(0)}$  to  $L_R^{(|\mathcal{I}|)} = L_R^{(0)}/R^{(0)}$ , and therefore is the driver for the convergence of the network.

The exact definition (shape) of the neighborhood radius (4.3) and learning rate (4.4) is not crucial to the quality of the mapping as long those functions are monotonically decreasing, never zero, and rather constant during the first few thousand iterations, such that the correct topographic order can be established before the adaption becomes too locally focused (Kohonen (2014)). In contrast to this, the setting of the initial learning rate  $L_R^{(0)}$  and neighborhood radius  $R^{(0)}$  has major influence on the performance of the network. For example, if the value for  $R^{(0)}$  is too small, the correct topographic order might not be established, since the coverage of the neurons being modified is too small. If the values are too high on the other hand, a reasonably chosen initial order of the neuron weights (other than random) might get lost. In the present application, values between  $0.2 \max\{I, J\}$  and  $0.7 \max\{I, J\}$  performed well, and it was decided to set  $R^{(0)} = 0.5 \max\{I, J\}$ , because in this case the desired final learning rate  $L_R^{(|\mathcal{I}|)} \approx 0.005$  is met (see details below).

The initial learning rate should always be  $L_R^{(0)} < 1$  in order to prevent  $\mathbf{x}_{(i,j)^*}^{(k)}$  from just snapping onto the input vector  $\mathbf{x}_{i_k}$  which causes kinks and distortions of the network by prohibiting the correct topographic ordering of the network. In the present application, using definitions (4.3) and (4.4), it was found that for  $L_R^{(0)} \approx 0.1$  and  $L_R^{(|\mathcal{I}|)} \approx 0.005$  the mapping quality in regards to mesh orthogonality and equidistant covering of the training data in  $\mathcal{V}_T$  is best.

Originally SOMs were developed to simulate biological networks, where the lateral biological interaction causes weight vectors  $\mathbf{x}_{i_j}^{(k)}$  of neurons closer to the best matching unit  $(i, j)^*$  to be effected stronger by  $\mathbf{x}_{(i,j)^*}^{(k)}$  than those further away. Therefore, Kohonen (1990) introduced a bell-curve like neighborhood function

$$h_{ij}^{(k)} := \exp\left(-\frac{1}{2\sigma^2} \left(\frac{\|(i, j) - (i, j)^*\|}{R^{(k)}}\right)^2\right), \quad (4.5)$$

which decreases from one at the center  $(i, j)^*$  to  $\exp(-1/(2\sigma^2))$  at the edge of  $R^{(k)}$  (actually Kohonen did not use  $R^{(k)}$ , but a general spacing and time/iteration dependent function



$\sigma(t)$ ; but he claims that the shape of  $\sigma(t)$  does not matter and only requires monotonic decrease). The variance  $\sigma^2$  is a shape parameter controlling the width of the neighborhood function (4.5). For higher values this parameter causes a less progressive decrease of  $h_{ij}^{(k)}$ , and thus of the adaption rate  $L_R^{(k)} h_{ij}^{(k)}$  within the region of  $R^{(k)}$ . In the present application of surface fitting, it was found that a higher value  $\sigma^2 \geq 1$ , which gives the bell-curve a more parabolic shape within  $R^{(k)}$ , has hardly any effect on the final mapping error  $\varepsilon_M$  defined as maximum absolute value of the perpendicular distance between reference surface and mapped surface, see Fig 4.4. Although lower mapping errors require  $0 < \sigma^2 < 1$ , such values are found to harm establishing an orthogonal topographic mesh order, and thus the equidistant covering of the training data in  $\mathcal{V}_T$ . In the present application, where the SOM is used for mapping only, the neighborhood function may be set as constant, e.g.  $h_{ij}^{(k)} := 1$ , with negligible effect on the mapping error.

The size of the network is set to be  $40 \times 40$  (note, all following graphics contain maps of size  $20 \times 20$  for better visualization) which is about half of the resolution of the FE-mesh, but sufficient to resolve the eigenmode characteristics. Increasing the resolution would also increase the required time for training. Kohonen (1990) suggests that the number of training iterations  $|\mathcal{I}|$  should be about 500 times the network size, i.e., here  $|\mathcal{I}| = 40 * 40 * 500 = 800000$ . However, in the present case  $|\mathcal{I}| = 100000$  has proven to be sufficient. The final maximum fitting error  $\varepsilon_M$  is then about 0.12mm and is located in a small region at the transition between fillet and airfoil.

Before training the network, each coordinate of the training vectors  $\mathbf{x}_n \in \mathcal{V}_T$  should be scaled w.r.t. the corresponding minimum and maximum values of the whole set  $\mathcal{V}_T$ . This prevents the training from being dominated by the dimension with the biggest range due to the Euclidean distance used in Eq. (4.1). Then, the weight vectors representing node

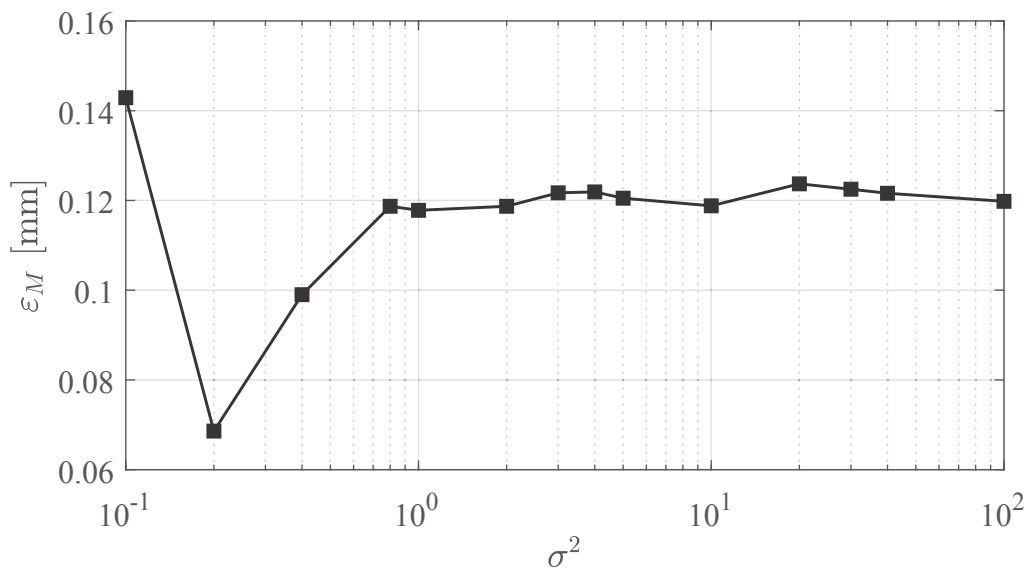


Figure 4.4: Surface error of the mapping depending on shape parameter  $\sigma$  of the neighborhood function

coordinates  $\mathbf{x}_{ij}^{(0)}$  of the network have to be initialized. Due to the geometric symmetry of network and airfoil surface (both have four corners), the SOM is able to find the correct topographic order independently from the initial state, and it performs well even for random initialization, see Fig. 4.5. However, it might happen that the orientation of the mapping (Fig. 4.2c) is rotated by multiples of  $90^\circ$  according to the geometric symmetry, if random initialization is being used. However, if the network is initialized close to its final state, the rotation can be avoided and additionally convergence speed improved. In the present case, a plane square with the normal vector  $\mathbf{n} = [1, -1, 0]^T$  is a good initial fit of the airfoil surface due to the normalization of the coordinates as mentioned above, see Fig. 4.6. With the initial weight vectors  $\mathbf{x}_{ij}^{(0)}$  set to a regular mesh on this plane, the mapping will not be rotated as long as the initial neighborhood radius  $R^{(0)}$  is not set too large.

After the training, the final weight vectors of the neurons  $\mathbf{x}_{ij}^{(|I|)}$  represent the airfoil geometry by a regular mesh (Fig. 4.2). The airfoil displacements  $\mathbf{u}_{ij}$  corresponding to the regular mesh can then be interpolated, e.g., with natural-neighbor interpolation (Sibson (1981)) from the original node coordinates  $\mathbf{x}_n$  and the corresponding displacements  $\mathbf{u}_n$  of the FE-mesh. Alternatively, the displacement information may be included into the weight vectors of the SOM, i.e.,  $[\mathbf{x}_{ij}^{(k)T}, \mathbf{u}_{ij}^{(k)T}]^T$ , and modified together by the adaption scheme (4.2), which yields almost the same results as interpolation and does not harm convergence speed of the network. However, for the sake of simplicity the interpolation was preferred for the explanations here.

## 4.2 Training of Neural Network

As mentioned before, training of the network requires a set  $\mathcal{V}_T$  of training vectors (samples). In the present case, the training vectors are the nodal coordinates  $\mathbf{x}_n$  of the FE-mesh. Because the network learns iteratively and is supposed to converge to a static state, each sample  $\mathbf{x}_n \in \mathcal{V}_T$  has to be trained repetitively. Therefore, an index set  $\mathcal{I} = \{i_1, i_2, \dots, i_{N_T}\}$

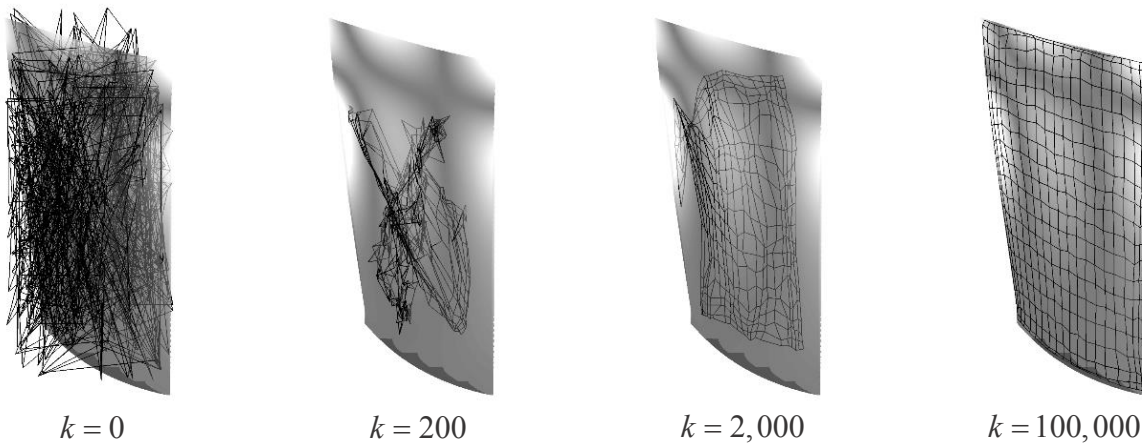


Figure 4.5: Adaption of mesh to airfoil surface during the training of the SOM starting from random initialization of the neurons

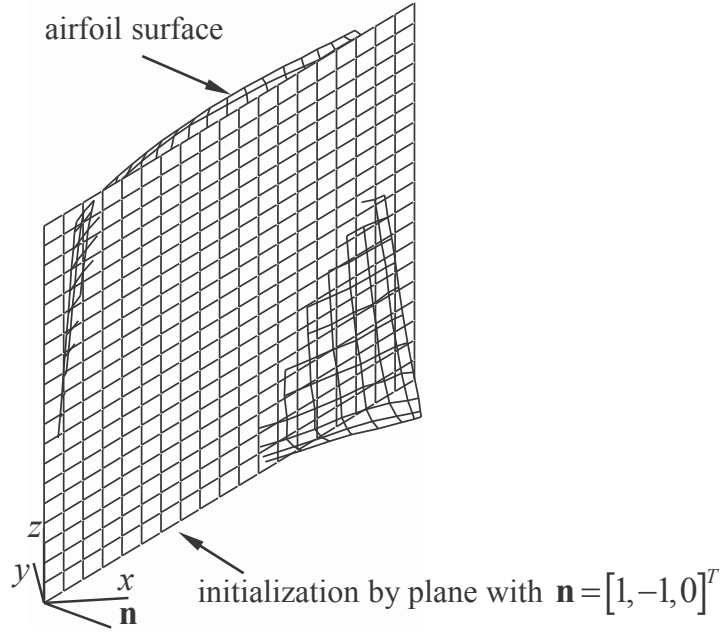


Figure 4.6: Final airfoil surface mesh compared to the initial mesh before training (both normalized)

( $N_T = |\mathcal{I}|$ ) is introduced which contains the order of the samples  $\mathbf{x}_{i_k}$  drawn from  $\mathcal{V}_T$  during the training. It will be shown that the definition of  $\mathcal{I}$ , i.e., the order of training, strongly effects the quality of agreement between SOM results and original airfoil surface.

According to Eq. (4.3), the neighborhood radius  $R^{(k)}$  continuously decreases with each training iteration  $k$ , which is why the global order of the node coordinates  $\mathbf{x}_{ij}$  has to be established before  $R^{(k)}$  becomes so small that training has only local effects. Furthermore, the training is a stochastic process where final node coordinates  $\mathbf{x}_{ij}^{(N_T)}$  are a weighted sum of adaptations w.r.t. training vectors  $\mathbf{x}_{i_k} \in \mathcal{V}_T$  affecting their neighborhood. The weights are associated with the adaption rate  $L_R^{(k)} h_{ij}^{(k)}$  in Eq. (4.2) depending on the iteration step and distance to the best matching unit  $(i, j)^*$ . Thus, both the decreasing  $R^{(k)}$  and stochastic nature of the training process make it necessary that training vectors should be drawn randomly and equally often from the set  $\mathcal{V}_T$ . However, later it will be shown that selective cumulation of specific samples may improve the fitting of the neural network to the airfoil surface. Both can be accomplished by using the permutation operator

$$\pi : \{1, \dots, N_N\} \rightarrow \{1, \dots, N_N\} \quad (4.6)$$

to create a random permutation sequence of the numbers  $1, \dots, N_N$ , which yields a first index set

$$\mathcal{I}_1 := \{\pi(j), j = 1, \dots, N_N\} . \quad (4.7)$$

In order to ensure convergence of the network, each training vector in  $\mathcal{V}_T$  has to be picked multiple times. This is achieved through concatenation of the index set  $\mathcal{I}_1$  with itself

until the cardinality of the resulting set  $\mathcal{I}$  either equals or exceeds the desired number of iterations  $N_T$ :

$$\mathcal{I} := \mathcal{I}_1 \cup \mathcal{I}_1 \cup \dots = \bigcup_{|\mathcal{I}| \geq N_T} \mathcal{I}_1 . \quad (4.8)$$

The mesh after training with set (4.8) is shown in Fig. 4.7a. It can be seen that the agreement between the SOM result and the airfoil surface indicated by black dots representing the FE-nodes agree well except at the edges. The reason for this deviation is that after the training the coordinate vector  $\mathbf{x}_{ij}$  of each neuron is the weighted sum of adaptations due training vectors  $\mathbf{x}_n$  in its neighborhood. Neurons close to the edges of the airfoil, however, have more neighboring training nodes towards the center of the airfoil than in the outer direction (neurons on the edges even have no nodes in outer direction). Those neurons are pulled towards the center of the map away from the edges.

This phenomenon was also investigated by Kohonen (2014) in regards to demodulation of quantized signals, and he suggested that this issue may be solved by combining k-means clustering with SOMs. A different strategy is to force the map focus on the edges by training it with nodes at the edges more often. These nodes may be selected by sorting the node vectors in  $\mathcal{V}_T$  w.r.t. their coordinates and find the indices of a specified fraction  $\alpha < 1$  of members from both ends of each coordinate range. For instance, FE-nodes closest to the leading edge can be identified by ordering nodes in  $x$ -direction:

$$\check{\mathcal{I}}_\alpha^x := \left\{ i_p \in \{1, \dots, N_N\} \mid x_{i_p} \leq x_{i_{p+1}} \wedge 1 \leq i_p \leq \lfloor \alpha N_N \rfloor \right\} . \quad (4.9)$$

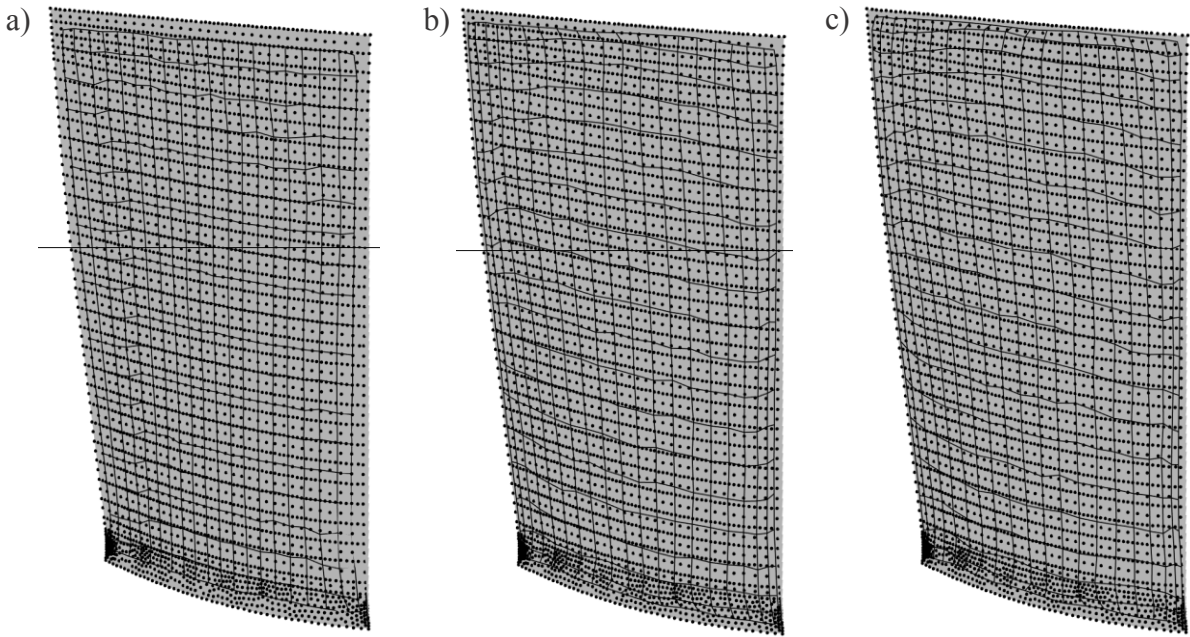


Figure 4.7: FE-mesh nodes ( $\bullet$ ) and the fitted regular mesh ( $-$ ) after  $N_T = 100,000$  iterations with index sets a) (4.8), b) (4.14), and c) (4.19)

FE-nodes close to the trailing edge are given by

$$\hat{\mathcal{I}}_\alpha^x := \left\{ i_p \in \{1, \dots, N_N\} \mid x_{i_p} \geq x_{i_{p+1}} \wedge 1 \leq i_p \leq \lfloor \alpha N \rfloor \right\}. \quad (4.10)$$

Analogous ordering in  $y$ -direction identifies FE-nodes closest to leading and trailing edge as well, while nodes closest to tip and root are received by ordering in  $z$ -direction. It seems redundant to order twice for the leading and trailing edge, but in case of twisted airfoils this is strongly recommended. Combining the sorted sets as

$$\mathcal{I}_\alpha := \check{\mathcal{I}}_\alpha^x \cup \hat{\mathcal{I}}_\alpha^x \cup \check{\mathcal{I}}_\alpha^y \cup \hat{\mathcal{I}}_\alpha^y \cup \check{\mathcal{I}}_\alpha^z \cup \hat{\mathcal{I}}_\alpha^z \quad (4.11)$$

and applying random permutation to randomize this set yields the border set

$$\mathcal{I}_2 := \{ \pi(j), j \in \mathcal{I}_\alpha \} . \quad (4.12)$$

This set  $\mathcal{I}_2$  is then repeatedly concatenated with  $\mathcal{I}_1$  until the cardinality of the final index set  $\mathcal{I}$  either equals or exceeds the desired number of iterations  $N_T$ :

1. Set  $\mathcal{I} = \emptyset$  as empty set.
2. Generate  $\mathcal{I}_1$  according to Eq. (4.7).
3. Set the boundary factor to

$$\alpha = \hat{\alpha} - \frac{|\mathcal{I}|}{N_T} (\hat{\alpha} - \check{\alpha}) , \quad (4.13)$$

where the limit parameters are chosen as  $\hat{\alpha} = 0.3$  and  $\check{\alpha} = 0.05$ .

4. Generate  $\mathcal{I}_2$  according to Eq. (4.12).
5. Append the actual index set  $\mathcal{I}$  by  $\mathcal{I}_1$  and  $\mathcal{I}_2$ , i.e.,

$$\mathcal{I} := \mathcal{I} \cup \mathcal{I}_1 \cup \mathcal{I}_2 . \quad (4.14)$$

6. If  $|\mathcal{I}| \leq N_T$  proceed with step 3, else stop.

Instead of Eq. (4.13), more refined formulas could be used to estimate  $\alpha$  such that (4.14) finally yields  $|\mathcal{I}| \approx N_T$ , but the proposed linear decay has proven to be sufficient, and at the end there is no strong need to ensure  $|\mathcal{I}| \approx N_T$ . Fitting based on set (4.14) improves the approximation at the edges as can be seen in Fig. 4.7b. It is worth to mention that training the edges separately using distinct sets for the edges (as utilized in Eq. (4.14)) gives better results than training with a random permutation of set (4.14), because in this way  $\mathcal{I}_2$  can better counteract the contraction phenomenon.

Further improvement of the fitting at the edges can be achieved by including a third subset into Eq. (4.14) which contains those nodes  $\mathbf{x}_n = [x_n, y_n, z_n]^T$  belonging to the

convex hull in the  $x$ - $z$ -plane of Fig. 4.6, i.e.,

$$\mathcal{I}_3^x = \left\{ i_p \in \{1, \dots, N\} \mid [x_{i_p}, z_{i_p}] \in \text{conv}(\{(x_n, z_n), n \in \{1, \dots, N_N\}\}) \right\}, \quad (4.15)$$

where e.g.

$$\text{conv}(\mathcal{V}_T) := \left\{ \sum_{n=1}^{|\mathcal{V}_T|} w_n \mathbf{x}_n \mid w_n \geq 0 \wedge \sum_{n=1}^{|\mathcal{V}_T|} w_n = 1 \right\}. \quad (4.16)$$

Accordingly the convex hull in the  $y$ - $z$ -plane is given by

$$\mathcal{I}_3^y = \left\{ i_p \in \{1, \dots, N_N\} \mid [y_{i_p}, z_{i_p}] \in \text{conv}(\{(y_n, z_n), n \in \{1, \dots, N_N\}\}) \right\}. \quad (4.17)$$

These sets are then combined and permuted randomly as

$$\mathcal{I}_3 := \{ \pi(j) : j \in \mathcal{I}_3^x \cup \mathcal{I}_3^y \}, \quad (4.18)$$

in order to generate the final index set  $\mathcal{I}$  in accordance with the procedure above, but with Eq. (4.14) substituted by

$$\mathcal{I} := \mathcal{I} \cup \mathcal{I}_1 \cup \mathcal{I}_2 \cup \mathcal{I}_3. \quad (4.19)$$

As a result, Fig. 4.7c show an improved approximation of the airfoil edges.

Apparently, as a trade-off to the better fitting on the edges, the sets (4.14) and (4.19) cause the map to become a bit more distorted (orthogonality decreases). This is considered as an incidental defect, because the classification procedure in Chapter 6 has to be robust against distortions in the mapped displacement fields anyway. For example, it has to operate on eigenmodes from rather different airfoil geometries where displacement fields won't coincide, but only share some characteristics. The fitting of edges, however, is crucial due to the fact that characteristic local extrema in the displacement field are most likely occurring at the edges. Thus, their detection strongly effects the quality of classification.

# 5 Normalization and Dimension Reduction of Eigenmode Data

As one goal of this thesis is the correct automated assignment of eigenfrequency bands w.r.t. fundamental eigenmode shapes, the correct assignment will greatly depend on the correct classification of the eigenmode shapes at the lower and upper stiffness bounds (see Section 3.2). Classification aims to establish borders between members belonging to different categories, where all classification algorithms benefit from proper data pre-processing using data normalization and dimension reduction which improves separation between clusters of members belonging to different categories and makes data less sparse. Within this thesis the members are structural eigenmode shapes represented by vectors of nodal displacements as a result of FE modal analysis.

In the following, first the effect of different methods of data alignment and normalization onto the data structure will be analyzed. For assessment, not just the well established silhouette value will be evaluated, but also two newly defined methods which compensate shortfalls of the silhouette value. Then, a detailed review on different methods of dimension reduction is given, followed by an analysis of their effect on the data structure.

## 5.1 Alignment and Normalization of Eigenmode Data

Eigenmodes of airfoils are characteristic shapes of the displacement field which the structure experiences from harmonic excitation. In the present case, the nodal displacement field refers to FE-nodal information on the suction-side surface of airfoils. The eigenmodes are projected onto standard rectangular surfaces with nodes  $(ij)$ ,  $i = 1 \dots I$ ,  $j = 1 \dots J$ , according to Chapter 4 and Fig. 5.1, in order to make them comparable across different airfoil geometries. Hence, the surface-displacement field is described in the natural coordinate system of the airfoil surface by vectors  $\mathbf{u}_{ij} = [u_{ij}^\perp, u_{ij}^\rightarrow, u_{ij}^\uparrow]^T$  with roughly perpendicular ( $u_{ij}^\perp$ ), tangentially axial ( $u_{ij}^\rightarrow$ ), and tangentially radial ( $u_{ij}^\uparrow$ ) displacement components. A representative eigenvector is assembled from the whole projected displacement field as

$$\mathbf{u} := [\mathbf{u}^{\perp T}, \mathbf{u}^{\rightarrow T}, \mathbf{u}^{\uparrow T}]^T = [u_{11}^\perp, u_{12}^\perp, \dots, u_{IJ}^\perp, u_{11}^\rightarrow, u_{12}^\rightarrow, \dots, u_{IJ}^\rightarrow, u_{11}^\uparrow, u_{12}^\uparrow, \dots, u_{IJ}^\uparrow]^T \in \mathbb{R}^D, \quad (5.1)$$

where the dimension  $D = 3 \times I \times J$  may become rather large, e.g.,  $D = 4800$  for  $I = J = 40$ . This can harm the classification performance, which is the motivation for

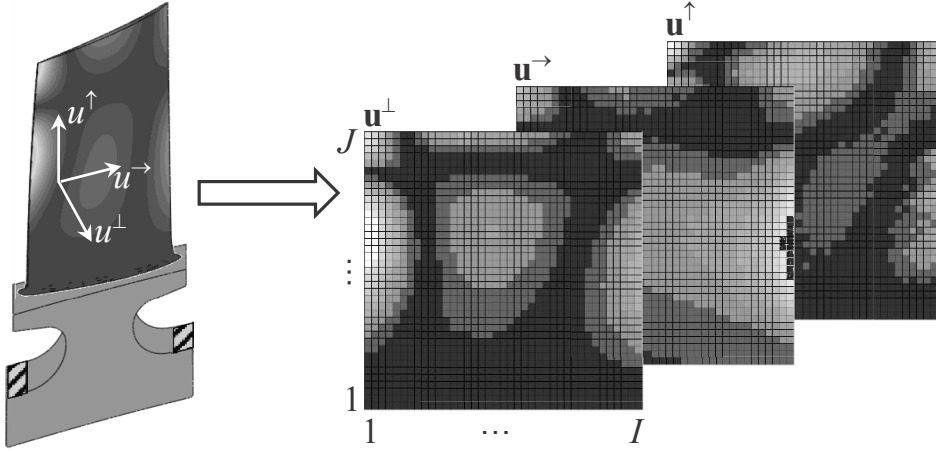


Figure 5.1: Extracted projection of the characteristic displacement fields of a compressor blade

investigating different dimension reduction procedures in Section 5.2. But first, aligning and normalization may help to improve data structure as an initial step.

As shown by Eq. (B.8) in Appendix B, eigenvectors are not unique. Hence, orienting them equally may enhance comparability. Therefore, eigenvectors are redirected such that the perpendicular displacement  $u_{1J}^\perp$  of the upper left corner, as one of the most distinguishing points in airfoil vibration, is equally oriented, i.e., vector (5.1) may be redefined as

$$\mathbf{u} := \begin{cases} -\mathbf{u} & \text{if } u_{1J}^\perp < 0 \\ \mathbf{u} & \text{else .} \end{cases} \quad (5.2)$$

The assessment in Appendix B additionally suggests that eigenvectors may shift the sign of their amplitude ratio during frequency veering, whereas they maintain their characteristics. For example, a stiff-wise bending mode with the main deflection in axial direction ( $\mathbf{u}^\rightarrow$ ) may be identified independently from deflections  $\mathbf{u}^\perp$  and  $\mathbf{u}^\uparrow$ . Thus, as an alternative to Eq. (5.2) redirection of eigenvectors  $\mathbf{u}$  might be performed direction-wise as

$$*\mathbf{u} := \left[ \underbrace{\text{sgn}(u_{1J}^\perp) \mathbf{u}^{\perp T}}_{*\mathbf{u}^{\perp T}}, \underbrace{\text{sgn}(u_{1J}^\rightarrow) \mathbf{u}^{\rightarrow T}}_{*\mathbf{u}^{\rightarrow T}}, \underbrace{\text{sgn}(u_{1J}^\uparrow) \mathbf{u}^{\uparrow T}}_{*\mathbf{u}^{\uparrow T}} \right]^T \quad (5.3)$$

where

$$\text{sgn}(x) := \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{else .} \end{cases} \quad (5.4)$$

As eigenvectors are not unique, normalization with the Euclidean norm  $\|\mathbf{u}\|$  resulting



in

$$\tilde{\mathbf{u}} = [\tilde{\mathbf{u}}^{\perp T}, \tilde{\mathbf{u}}^{\rightarrow T}, \tilde{\mathbf{u}}^{\uparrow T}]^T := [\mathbf{u}^{\perp T}, \mathbf{u}^{\rightarrow T}, \mathbf{u}^{\uparrow T}]^T / \|\mathbf{u}\|, \quad (5.5)$$

$$*\tilde{\mathbf{u}} = [*\tilde{\mathbf{u}}^{\perp T}, *\tilde{\mathbf{u}}^{\rightarrow T}, *\tilde{\mathbf{u}}^{\uparrow T}]^T := [*\mathbf{u}^{\perp T}, *\mathbf{u}^{\rightarrow T}, *\mathbf{u}^{\uparrow T}]^T / \|*\mathbf{u}\| \quad (5.6)$$

might be beneficial, because this projection onto an unit hypersphere makes differences only apparent through the orientation of the eigenvectors but their amplitudes ( $\alpha_i$  in Eq. (B.8)) do not affect the characteristic of eigenvector  $\boldsymbol{\phi}_i$ ). Following this argument, other norms such as Frobenius, Manhattan, or maximum norm seem little promising and actually turned out to perform worse than the Euclidean norm.

In the following, formulas and procedures will be applied to the different formulations (5.1)-(5.3), (5.5) and (5.6) of the characteristic vectors or their sub-vectors. In order to keep the subsequent formulas universal, a general vector

$$\boldsymbol{\phi} \in \{\mathbf{u}, *\mathbf{u}, \tilde{\mathbf{u}}, *\tilde{\mathbf{u}}, \tilde{\mathbf{u}}^{\perp}, *\tilde{\mathbf{u}}^{\perp}, \dots\} \quad (5.7)$$

is introduced instead. Additionally, in order to assess and compare the various strategies, a reference set of samples with known classification is needed. Throughout this thesis, the reference set is based on an industrial compressor of a stationary gas-turbine with  $N_R = 29$  rows of airfoils (IGV plus 14 stages). For each row the first ten eigenmodes have been computed at the lower as well as the upper stiffness limit, resulting in overall  $29 * 10 * 2 = 580$  eigenmodes which have been classified by a human expert. Some modes, however, had to be omitted because they could not be assigned unambiguously to one of the specific classes in Fig. 2.6. Finally, a set of  $N_M = 548$  classified vectors  $\boldsymbol{\phi}_k$  (Eq. (5.7)) with  $k = 1, \dots, N_M$  is received. This set is specified as  $\mathcal{V} := \{\boldsymbol{\phi}_k, k = 1, \dots, N_M\}$  and each vector  $\boldsymbol{\phi}_k$  is associated with a class  $c_k \in \{1, \dots, N_C\}$  and a compressor row  $r_k \in \{1, \dots, N_R\}$ . Based on this, the set  $\mathcal{V}$  may be split class- and row-wise into subsets  $\mathcal{V}_c^r$ , which may be again combined as

$$\mathcal{V}_c = \bigcup_{r=1}^{N_R} \mathcal{V}_c^r, \quad \mathcal{V}_r = \bigcup_{c=1}^{N_C} \mathcal{V}_c^r, \quad \mathcal{V} = \bigcup_{c=1}^{N_C} \mathcal{V}_c = \bigcup_{r=1}^{N_R} \mathcal{V}^r \quad (5.8)$$

where  $\mathcal{V}_c$  represents all mode vectors of a specific class  $c$  and  $\mathcal{V}^r$  summarizes all mode vectors of a specific row  $r$ . Thus, a specific vector  $\boldsymbol{\phi}_k \in \mathcal{V}$  is also element of the sets  $\mathcal{V}_{c_k}^{r_k}$ ,  $\mathcal{V}_{c_k}$ ,  $\mathcal{V}^{r_k}$ .

Firstly the effects on cluster separation by redirecting and normalizing eigenmode vectors are analyzed. This may be accomplished by computing the silhouette value (Rousseeuw (1987)) for each vector  $\boldsymbol{\phi}_k$ . It indicates how distinct the membership of a sample  $\boldsymbol{\phi}_k$  to cluster  $c_k$  is compared to the closest neighboring cluster  $c^* \neq c_k$ . Calculation of the silhouette value requires computation of the average distance of  $\boldsymbol{\phi}_k$  to all points of

a cluster  $\mathcal{V}_c$  as

$$\overline{dist}(\boldsymbol{\phi}_k, \mathcal{V}_c) := \frac{1}{|\mathcal{V}_c|} \sum_{\boldsymbol{\phi}_l \in \mathcal{V}_c} \|\boldsymbol{\phi}_k - \boldsymbol{\phi}_l\| . \quad (5.9)$$

The class  $c^* \neq c_k$  of the neighboring cluster  $\mathcal{V}_{c^*}$  with the smallest average distance to  $\boldsymbol{\phi}_k$ , but not containing  $\boldsymbol{\phi}_k$ , i.e.,  $\boldsymbol{\phi}_k \notin \mathcal{V}_{c^*}$ , is given as

$$c^* := \arg \min_{c \neq c_k} \left( \overline{dist}(\boldsymbol{\phi}_k, \mathcal{V}_c) \right) . \quad (5.10)$$

Thus, the silhouette value

$$s(\boldsymbol{\phi}_k) := \frac{\overline{dist}(\boldsymbol{\phi}_k, \mathcal{V}_{c^*}) - \overline{dist}(\boldsymbol{\phi}_k, \mathcal{V}_{c_k})}{\max \left\{ \overline{dist}(\boldsymbol{\phi}_k, \mathcal{V}_{c^*}), \overline{dist}(\boldsymbol{\phi}_k, \mathcal{V}_{c_k}) \right\}} \in [-1, 1] \quad (5.11)$$

compares the average distance of  $\boldsymbol{\phi}_k$  to its own cluster  $\mathcal{V}_{c_k}$  with its average distance to the nearest neighboring cluster  $\mathcal{V}_{c^* \neq c_k}$ . The silhouette value is (+1) if the average distance to its own cluster  $\mathcal{V}_{c_k}$  is zero, i.e.,  $\boldsymbol{\phi}_k$  is the centroid of  $\mathcal{V}_{c_k}$ , and it is (-1) if  $\boldsymbol{\phi}_k$  is the centroid of  $\mathcal{V}_{c^*}$ . The silhouette value is zero if  $\boldsymbol{\phi}_k$  is the combined centroid of both  $\mathcal{V}_{c_k}$  and  $\mathcal{V}_{c^*}$ . If clusters are spheres of similar size, high silhouette values of all  $\boldsymbol{\phi}_k \in \mathcal{V}$  indicate good cluster separation and thus high chance of correct classification. Else, low silhouette values may be a sign that clusters are poorly separated, differ much in size (volume) and/or their shapes differ from that of spheres.

The silhouette values for all vectors  $\boldsymbol{\phi}_k = \mathbf{u}_k \in \mathcal{V}$ , only aligned according to Eq. (5.2), are shown in Fig. 5.2a in descending order within each class (1B, 1T, ..., 3H; see Fig. 2.6). Except for 1B, 1T, and 3H, all other mode clusters show signs of possible intersections by fractions of negative silhouette values. This is especially the case for 1S, 3T, and 3B mode clusters, where modes are more prone to misclassification.

Normalization of the data according to Eq. (5.5), i.e.,  $\boldsymbol{\phi}_k = \tilde{\mathbf{u}}_k$ , obviously improves the data structure especially for the first-stiff-wise eigenmodes (1S), see Fig. 5.2b. This is certainly relevant, because 1S modes are characterized by high displacements in chord direction, but tend to take over other mode characteristics (especially in the scenario of frequency veering) of the perpendicular displacement field to such an extent that they are easily misclassified. Other eigenmode shapes (clusters) which profit from normalization are third torsion (3T) and bending (3B).

Besides  $\boldsymbol{\phi}_k = \tilde{\mathbf{u}}_k$ , eigenmodes may also be described by the normalized perpendicular displacement field  $\boldsymbol{\phi}_k = \tilde{\mathbf{u}}_k^\perp$  only, because except for 1S all other fundamental eigenmode shapes within reference set  $\mathcal{V}$  have been distinguished by human expert based on  $\mathbf{u}_k^\perp$ . As a matter of fact, Fig. 5.2c shows that for  $\boldsymbol{\phi}_k = \tilde{\mathbf{u}}_k^\perp$ , data structure is nearly as good as for  $\boldsymbol{\phi}_k = \tilde{\mathbf{u}}_k$  in Fig. 5.2b. Restriction to  $\boldsymbol{\phi}_k = \tilde{\mathbf{u}}_k^\perp$  reduces dimensionality of the classification problem and, therefore, is one of the possibilities going to be suggested in Section 5.2.

Without normalization, however, the silhouette values for  $\boldsymbol{\phi}_k = \mathbf{u}_k^\perp$  are nearly the

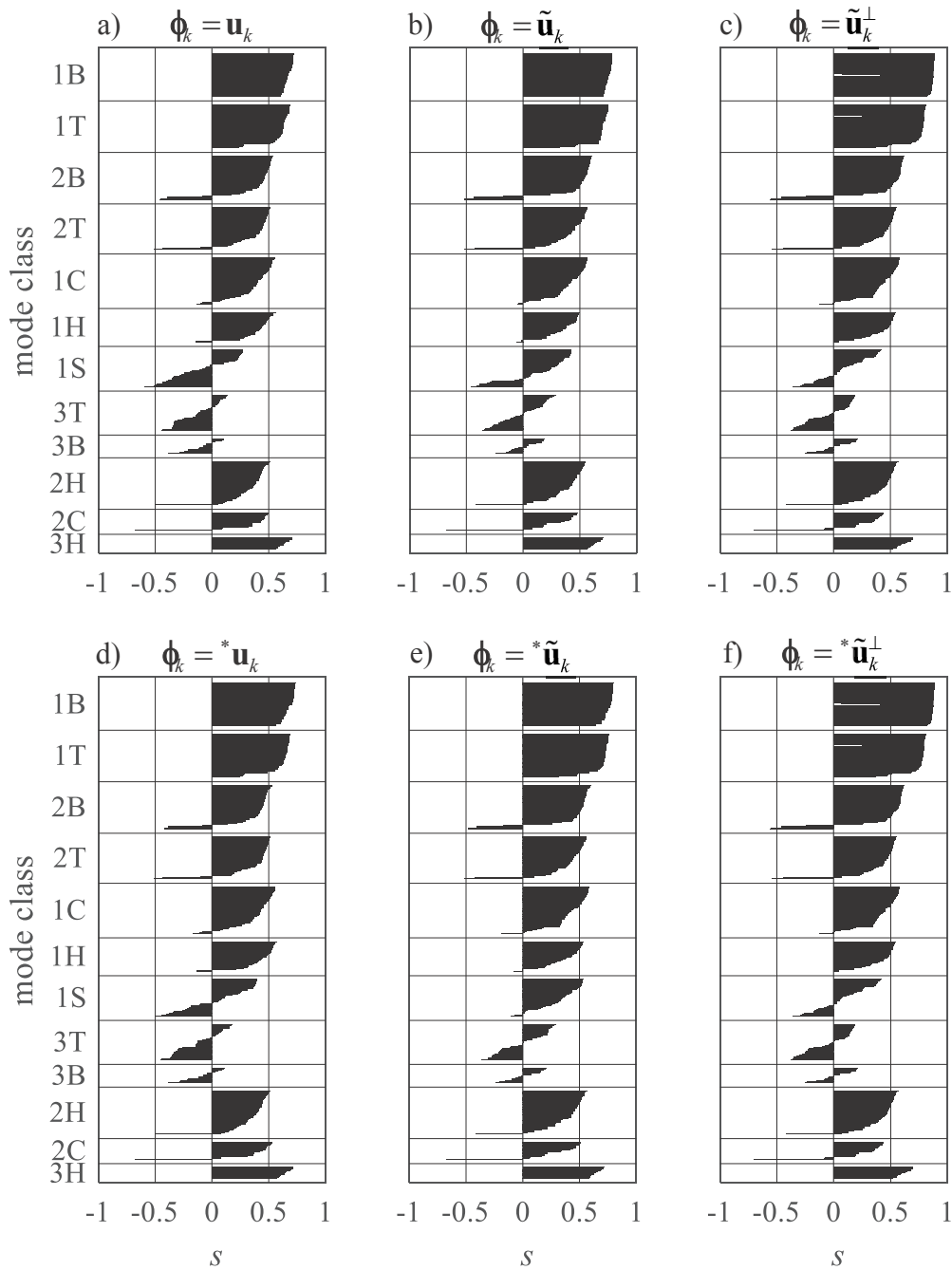


Figure 5.2: Effect of normalization on cluster separation (silhouette values)

same as in Fig. 5.2a, except for the 1S cluster which has only negative silhouette values then (not shown here). Consequently, normalization transfers information from the axial displacement field  $\mathbf{u}_i^\rightarrow$  onto the perpendicular field via the scaling (5.5). Note that also other strategies such as scaling the reference set feature-wise between zero and one have been investigated, but none performed better than strategy (5.5).

In comparison to  $\phi_k = \mathbf{u}_k$  or  $\tilde{\mathbf{u}}_k$ , redirecting the data according to Eq. (5.3) and using  $\phi_k = {}^*\mathbf{u}_k$  or  ${}^*\tilde{\mathbf{u}}_k$ , gives a better data structure for the 1S-mode cluster. The beneficial

effect vanishes, however, for  $\boldsymbol{\phi}_k = * \tilde{\mathbf{u}}_k^\perp$ . This is not surprising, since (5.2) and (5.3) only affect the orientation of the 1S-mode vectors, but not their magnitude, which means that the normalized perpendicular displacement field will hardly be effected.

As already mentioned the silhouette value only allows to judge on cluster separation, if the corresponding clusters are almost shaped as spheres of similar size. However, if they differ as shown in Fig. 5.3a, negative silhouette values occur in Fig. 5.3b, although the clusters are clearly separated. Because of this, the distance ratio  $DR$  is introduced as a measure which grasps the data structure in a less global, but more local sense. It is defined as normalized difference between the distances of  $\boldsymbol{\phi}_k$  to the closest neighbor of its own kind and the closest neighbor of different classification instead of average distance (5.11):

$$DR(\boldsymbol{\phi}_k) := \frac{\text{dist}(\boldsymbol{\phi}_k, \mathcal{V}_{c^\circ}) - \text{dist}(\boldsymbol{\phi}_k, \mathcal{V}_{c_k})}{\max\{\text{dist}(\boldsymbol{\phi}_k, \mathcal{V}_{c^\circ}), \text{dist}(\boldsymbol{\phi}_k, \mathcal{V}_{c_k})\}} \in [-1, 1] \quad (5.12)$$

where

$$\text{dist}(\boldsymbol{\phi}_k, \mathcal{V}_c) := \min_{\boldsymbol{\phi}_l \in \mathcal{V}_c \setminus \{\boldsymbol{\phi}_k\}} \|\boldsymbol{\phi}_k - \boldsymbol{\phi}_l\| \quad (5.13)$$

and

$$c^\circ := \arg \min_{c \notin c_k} (\text{dist}(\boldsymbol{\phi}_k, \mathcal{V}_c)) . \quad (5.14)$$

Figure 5.3c shows that the distance ratio  $DR$  is able to identify that the negative silhouette values in Fig. 5.3b are not caused by cluster intersection (no negative  $DR$ -values). Note: Also the  $DR$ -criterion has its drawbacks, because if cluster data points are insufficiently sparse, this may cause false negative  $DR$ -values!

The evaluation of  $DR$ -values is given in Fig. 5.4. In comparison between Fig. 5.2a for silhouette values and Fig. 5.4a for  $DR$ -values, it can be seen that many of the negative

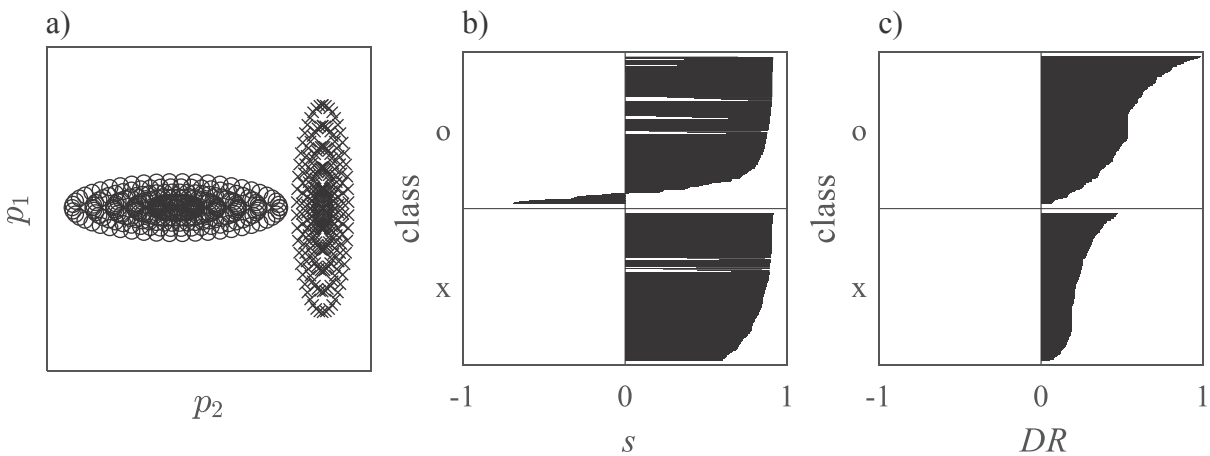


Figure 5.3: Example of a) artificial cluster distribution (marked by x and o, respectively) with corresponding b) silhouette values and c) distance ratios

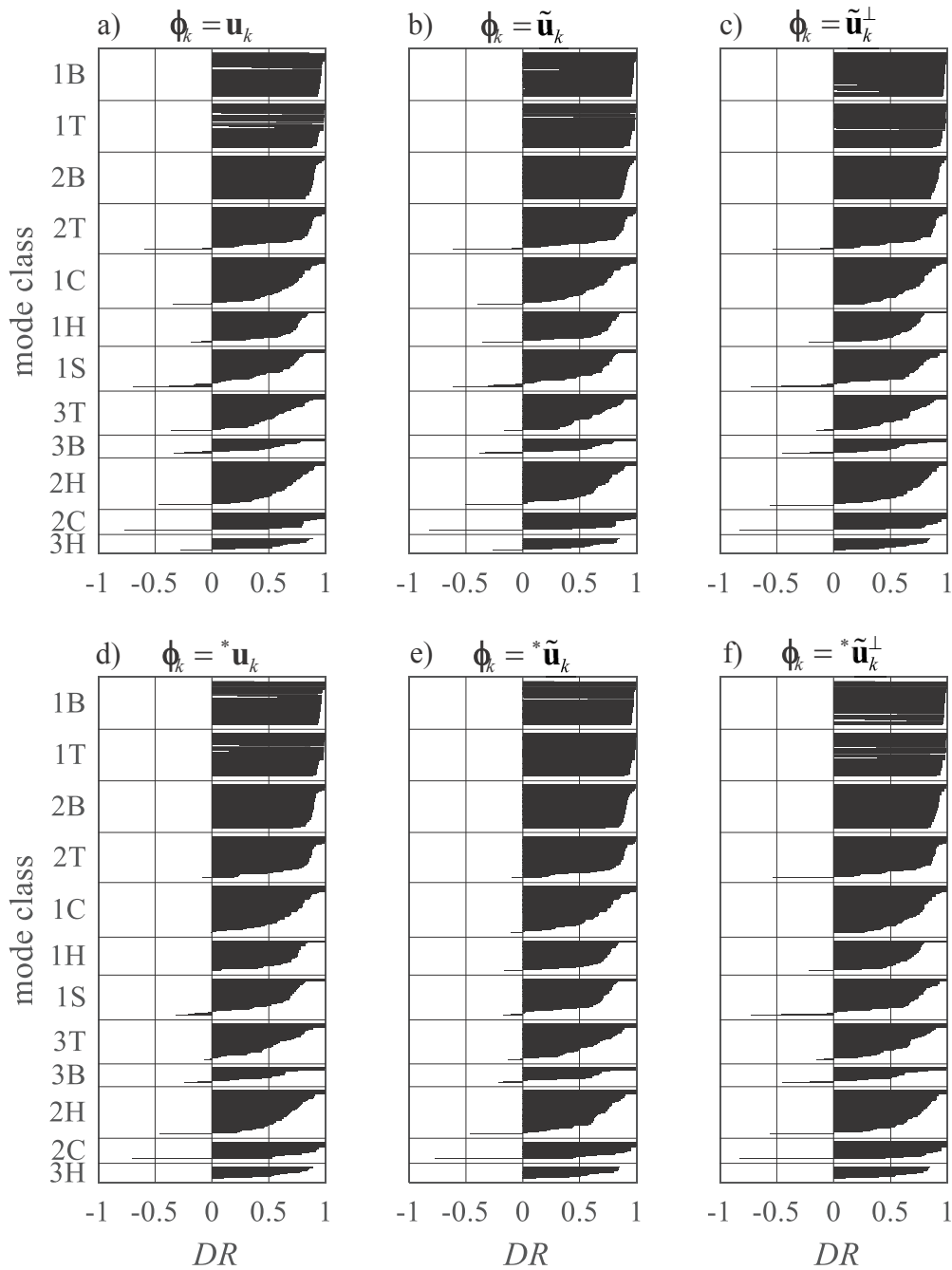


Figure 5.4: Effect of normalization on cluster separation (distance ratio)

silhouette values are actually caused by inhomogeneous clusters (differences in size, shape and orientation) and there are only very few potential intersections with both negative silhouette- and  $DR$ -values. However, there are some cases of ill sampled regions with negative  $DR$ -values but positive silhouette values as well, e.g., 3H cluster for  $\phi_k \hat{=} \mathbf{u}_k$  or  $\tilde{\mathbf{u}}_k$ . Normalization increases homogeneity of the clusters, hence improves silhouette values (Fig. 5.2b), but hardly effects intersections (Fig. 5.4b), whereas the perpendicular displacement field itself causes less intersections (Fig. 5.4c). Alignment with Eq. (5.3) also reduces intersections, see Fig. 5.4d–f.

The silhouette value and distance ratio may give insight into the quality of the clusters and/or their sampling, but no information is gained about the actual neighborhood structure between the clusters. Therefore, in Fig. 5.5 the nearest neighboring mode  $\phi_{p^*}$  of different classification  $c_{p^*} \neq c_k$  is identified for each mode  $\phi_k$  by

$$p^* = \arg \min_{p: \phi_p \in \mathcal{V} \setminus \mathcal{V}_{c_k}} \|\phi_k - \phi_p\|, \quad (5.15)$$

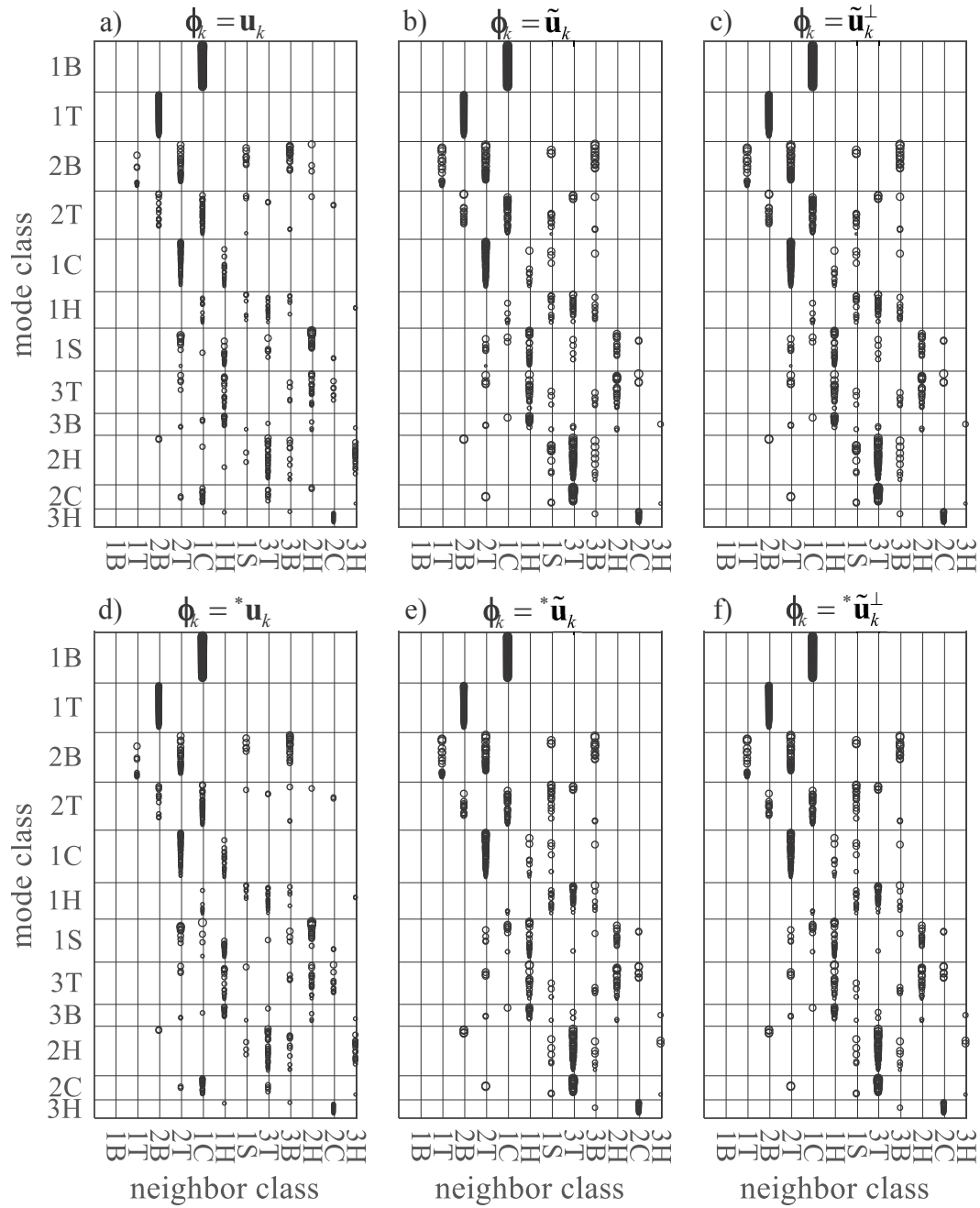


Figure 5.5: Effect of normalization onto the type of cluster neighbors and their distance (indicated by marker size)

and marked on the abscissa, while the size of the marker indicates the Euclidean distance (being scaled w.r.t. minimum and maximum distances between all members of  $\mathcal{V}$ ). The modes within each class are sorted by the distance to the neighboring clusters. For example, Fig. 5.5a then shows that the 1B cluster is remotely located from any other cluster since its only neighbor is the 1C cluster. All other clusters have more or less the same distance to the closest neighboring clusters. The primary effect of normalization in Fig. 5.5b is that distances between the clusters are more equalized, but it also causes changes in the neighborhood structure. Most modes now have fewer direct neighbors such as 2B or 2H, and only 1C has now more. In accordance to Fig. 5.2, Fig. 5.5 shows positive effects by using redirection (5.3) if no normalization is applied, but negligible effects otherwise.

In conclusion, redirecting in accordance to Eq. (5.3) improves cluster separation (Fig. 5.4) and using normalization improves the homogeneity of clusters w.r.t. shape, size, and orientation. Thus, without any further data treatment and depending on the sensitivity of the classifier to the dimensionality of the data,  $\boldsymbol{\phi}_k = {}^* \tilde{\mathbf{u}}_k$  or  $\boldsymbol{\phi}_k = {}^* \tilde{\mathbf{u}}_k^\perp$  should provide the lowest classification errors.

## 5.2 Procedures for Dimension Reduction of Eigenmode Data

Dimension reduction aims at neglecting information which is not needed for classification. The goal is to counteract the curse of dimensionality, a phenomenon of data becoming sparse for a fixed sample size as the volume increases exponentially with the dimension of data vectors. This phenomenon can harm the performance of classifiers because it deteriorates the statistical significance of the training data and/or increases the classifiers number of free parameters to such an extent that generalization w.r.t. new unknown data suffers significantly. In the following, potential methods for dimension reduction of eigenmode data shall be introduced in order to increase the data density by transforming the eigenvector data (5.1) of dimension  $D$  to vectors of lower dimension  $d < D$ :

$$\{\mathbf{u}_1, \dots, \mathbf{u}_{N_M}\}, \mathbf{u}_k \in \mathbb{R}^D \rightarrow \{\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_{N_M}\}, \boldsymbol{\phi}_k \in \mathbb{R}^{d < D}. \quad (5.16)$$

Note, that for matter of generality the transformation (5.16) may include any alignment or normalization strategy introduced in the previous section. Furthermore, for the sake of simplicity, it will only be referred to data alignment according to Eq. (5.2) in the following without harming generality. The effect of the dimension-reduction methods that are going to be introduced here, will be assessed within the subsequent section.

As mentioned in the last section, using the normalized perpendicular displacement field  $\boldsymbol{\phi}_k = \tilde{\mathbf{u}}_k^\perp$  as a representation for  $\mathbf{u}_k$  might be a promising reduction of data size from  $D = 4800$  to  $d = 1600$ . Taking the underlying idea one step further, one may conclude that, except for stiff-wise bending, the number of minima and maxima on the leading and trailing edge, the tip, and the middle vertical line are sufficient to characterize the underlying eigenmode shape, see Fig. 5.6. Thus, a characteristic pixel vector (CPV)

$$\boldsymbol{\phi} = CPV(\tilde{\mathbf{u}}^\perp) := \left[ \tilde{u}_{11}^\perp \dots \tilde{u}_{1J}^\perp \dots \tilde{u}_{IJ}^\perp \dots \tilde{u}_{I1}^\perp, \tilde{u}_{[I/2]J}^\perp \dots \tilde{u}_{[I/2]1}^\perp \right]^T \quad (5.17)$$

may be used for classification, where the order of pixels is irrelevant. For a  $40 \times 40$  pixel image, this reduces dimensionality of the data set to  $d = 157$ .

Another possibility to extract essential distribution of minimum/maximum information is to use Haar-detectors. They are also being used for face detection as suggested by Viola and Jones (2004). Each Haar-detector in Fig. 5.7 provides one characteristic scalar by summing up all gray-scale-pixel values in the black and white regions separately and subtracting the two resulting scalars as

$$\boldsymbol{\phi} = \sum_{h=1}^{16} \left[ \mathbf{e}^h \left( \sum_{(i,j) \in A_h^{\text{black}}} \tilde{u}_{ij}^\perp - \sum_{(i,j) \in A_h^{\text{white}}} \tilde{u}_{ij}^\perp \right) \right] \quad (5.18)$$

where  $\mathbf{e}^h \in \mathbb{R}^{16}$  is the  $h^{\text{th}}$  unit vector and  $A_h^{\text{black}}$ ,  $A_h^{\text{white}}$  refer to the black and white areas of the  $h^{\text{th}}$  Haar-detectors in Fig. 5.7. The Haar-detectors seem to be well suited for detection of eigenmode shapes, because each detector in Fig. 5.7 is sensitive to a specific eigenmode shape. For example, the detectors of the first column are sensitive to bending modes of first till fourth order, the detectors of the second column are sensitive to torsion modes of first till fourth order, and the detectors of the last two columns are sensitive to different orders of higher-order and chord-wise-bending modes. The use of these Haar-detectors reduces the dimensionality of the data set to  $d = \mathbb{R}^{16}$ .

In face recognition, the principle-component analysis (PCA; Jolliffe (2002)) is often used to receive the content of underlying principle eigenfaces of a set of image data (Heseltine et al. (2003)), which also reduces dimension of data significantly. The idea of PCA is sketched in Fig. 5.8a for a two dimensional problem, showing that for a set  $\mathcal{V}$  of vectors the coordinates  $[x_1, x_2]^T$  are expressed by a shift  $\mathbf{m}$  and rotation of the principle coordinate system  $\{\mathbf{e}'_1, \mathbf{e}'_2\}$  aligned along the main variances of  $\mathcal{V}$  in order to provide the new local vector coordinates  $[x'_1, x'_2]^T$ . Since directions (coordinates) with little variance provide only little gain in information, they may be omitted to reduce the data dimension without much loss of information. A detailed description of PCA is given in Appendix C.1. By accounting for 99.8% of the combined data variance (see (C.10) and (C.11)), the method

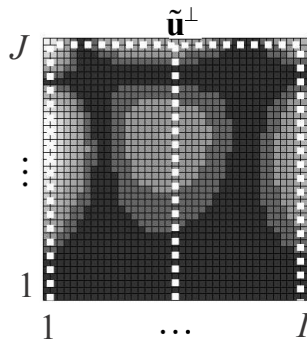


Figure 5.6: Trace of characteristic pixels of the perpendicular displacement field



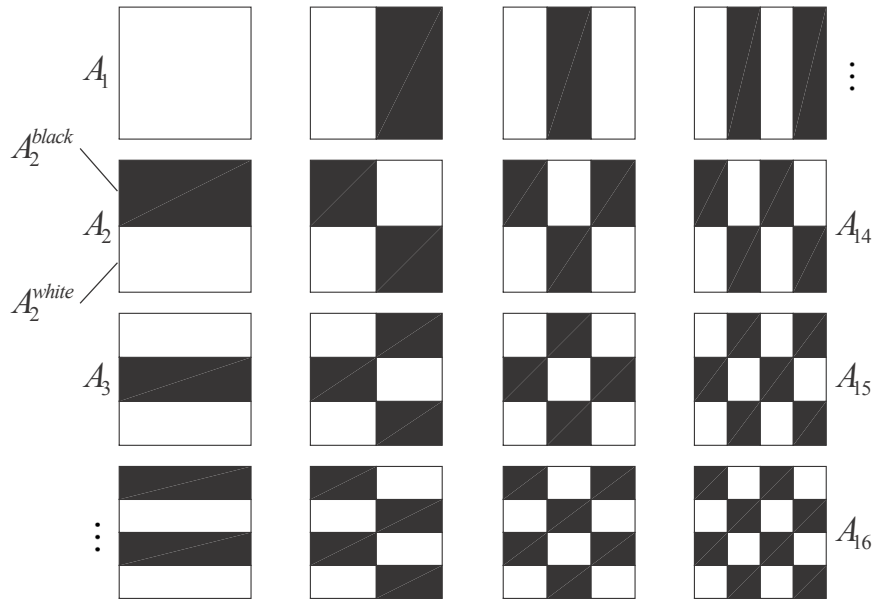


Figure 5.7: Haar-filters used for dimension reduction

reduces the dimension of the airfoil-data set from  $\mathbb{R}^{4800}$  to  $\mathbb{R}^{43}$ . By accounting for 99.99% of the combined data variance, the dimensionality would be  $\mathbb{R}^{186}$  with a slight benefit in classification performance for the methods to be introduced later. The perpendicular displacement fields of the first 50 principle directions are shown in Fig. 5.9, where the first ten principle directions together already account for 93.59% of the combined data variance. These first ten principle directions show more familiarity with some of the fundamental eigenmode shapes than the rest of the principle components. For example, the second direction looks like a 2T mode (see Fig. 2.6), the fourth direction looks like a 2B mode of the strongly twisted blades from the first rows of the compressor, and the sixth and tenth direction look like 1C and 2C modes, respectively. Besides the dimension reduction, PCA is also often used to whiten data (Bishop (2006)), i.e., to scale the data such that the variance becomes equal in all directions, which in many cases is considered to be beneficial

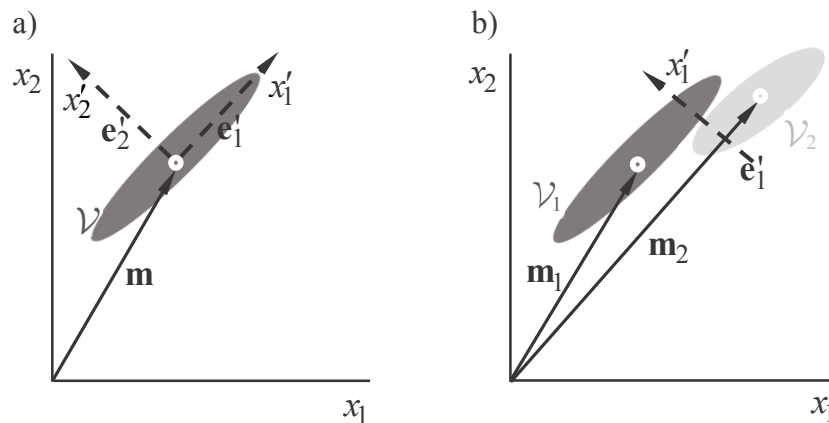


Figure 5.8: PCA projection (a) and MDA projection (b)

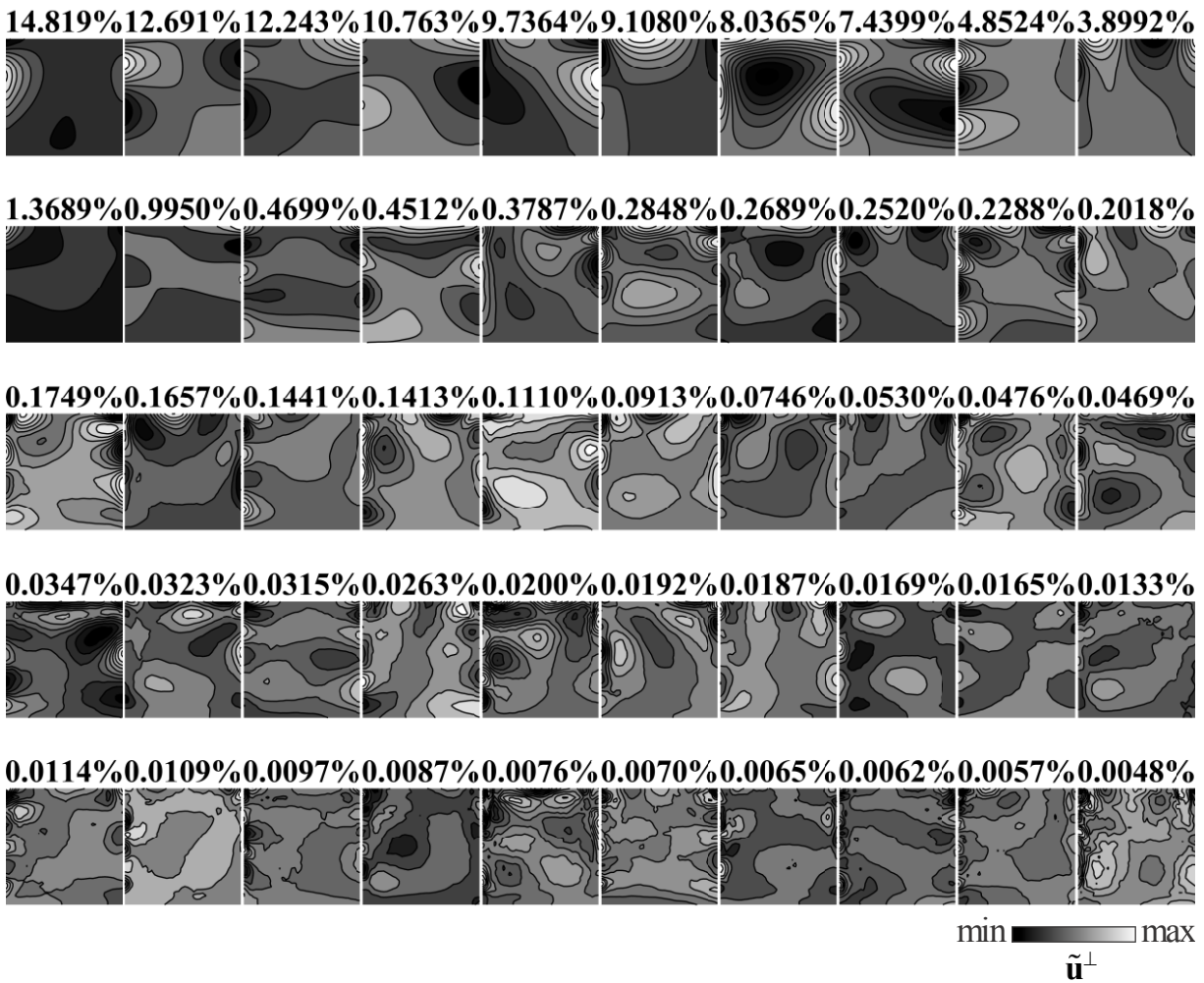


Figure 5.9: Perpendicular displacement fields of the first 50 principle directions with the according percentages of total variance

because of removing linear correlations between features of the data set (Appendix C.1, Eq. (C.12)).

PCA can only detect variances in linear directions, whereas data information with nonlinear correlations will get lost or the dimension of the data might not be reduced to the full possible extend. To overcome these issues, a nonlinear version of PCA using kernel functions (kPCA; see Appendix C.1) was suggested by Schölkopf et al. (1997).

The goal here is to classify data, which is why finding directions of best cluster separation instead of directions with highest variances might be more beneficial. To illustrate this, Fig. 5.8b shows two clusters  $\mathcal{V}_1$  and  $\mathcal{V}_2$  of data with mean vectors  $\mathbf{m}_1$  and  $\mathbf{m}_2$  which are best separated by projecting the data onto direction  $\mathbf{e}^1$  instead of the direction of main variance which is approximately directed along the cluster means. Based on linear Fisher-discriminant analysis (FDA) suggested by Fisher (1936) and explained in Appendix C.2, the multiple discriminant analysis (MDA; e.g. Duda et al. (2000)) finds directions where the distances between cluster means are highest w.r.t. the variances of the clusters in those directions. As suggested by Mika et al. (1999) and Roth and Steinhage

(1999), and in accordance to PCA, MDA can account for nonlinearities by also using kernel functions (kMDA). Using MDA/kMDA, the dimensionality reduces to one less than the number of clusters, i.e.,  $N_C - 1$ . Clearly, discriminant analysis is not just reducing data, but is a classifier as well. However, due to its close relation to PCA it is assessed already in the following section instead of the next chapter.

The perpendicular displacement fields of the first ten linear discriminants are shown in Fig. 5.10. In comparison to the principle components in Fig. 5.9, the discriminants look confusing and it can be suspected that they are sensitive to changes in the data set. Possible reasons for that will be discussed within the next section after the actual performance has been assessed. Another issue to be faced is that the dimension of the dataset is higher than the size of each cluster  $|\mathcal{V}_c|$ . Thus, the within-cluster-scatter matrix becomes non-positive definite and little values of  $10^{-13}$  have to be added to the main diagonal in order to receive results (see Appendix C.2).

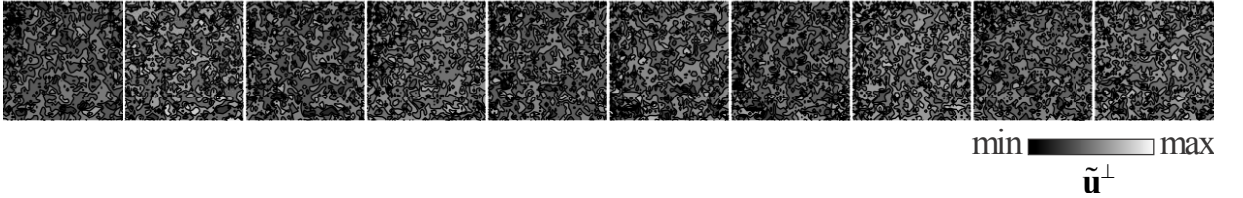


Figure 5.10: Perpendicular displacement fields of the first 10 discriminants

### 5.3 Assessment of Dimension Reduction Procedures

The effect of the different dimension-reduction methods on the data structure will again be evaluated by comparing silhouette values (5.11), distance ratios (5.12), and neighbor modes (5.15). All reduction methods will be applied to  $\tilde{\mathbf{u}}$  and  $^*\tilde{\mathbf{u}}$  in order to examine if the benefits of using redirection (5.3) are preserved after dimension reduction. Evaluation of the different reduction methods except for MDA will be performed using the reference set  $\mathcal{V}$  defined in Section 5.1. Because MDA requires knowledge of the classification of the samples of the data set used for calculation of the discriminates, the performance cannot be assessed by using  $\mathcal{V}$ , since classification is unknown in case of new data (new airfoil geometry i.e. compressor row design). Instead, the performance is assessed by using the individual set  $\mathcal{V} \setminus \mathcal{V}^{r_k}$  for each  $\phi_k \in \mathcal{V}^{r_k}$  where all eigenmodes of the same compressor row  $r_k$  as  $\phi_k$  are dropped from the set of training data (discriminants are calculated based on known data pretending  $\phi_k \in \mathcal{V}^{r_k}$  is new and of unknown class).

In comparison to Fig. 5.2c, Fig. 5.11a shows that despite the tremendous dimension reduction (5.17),  $\phi_k = \text{CPV}(\tilde{\mathbf{u}}_k^\perp)$  allows nearly as good separation as  $\phi_k = \tilde{\mathbf{u}}_k^\perp$  itself, i.e., most of the relevant information in the perpendicular displacement field is located at the trace (5.17). The Haar-detectors on the other hand ( $\phi_k = \text{Haar}(\tilde{\mathbf{u}}_k^\perp)$ ) cause slightly decreasing silhouette values in most of the clusters (Figure 5.11b), i.e., some useful information was lost. Improvement of data separation is given, if normalized eigenvectors

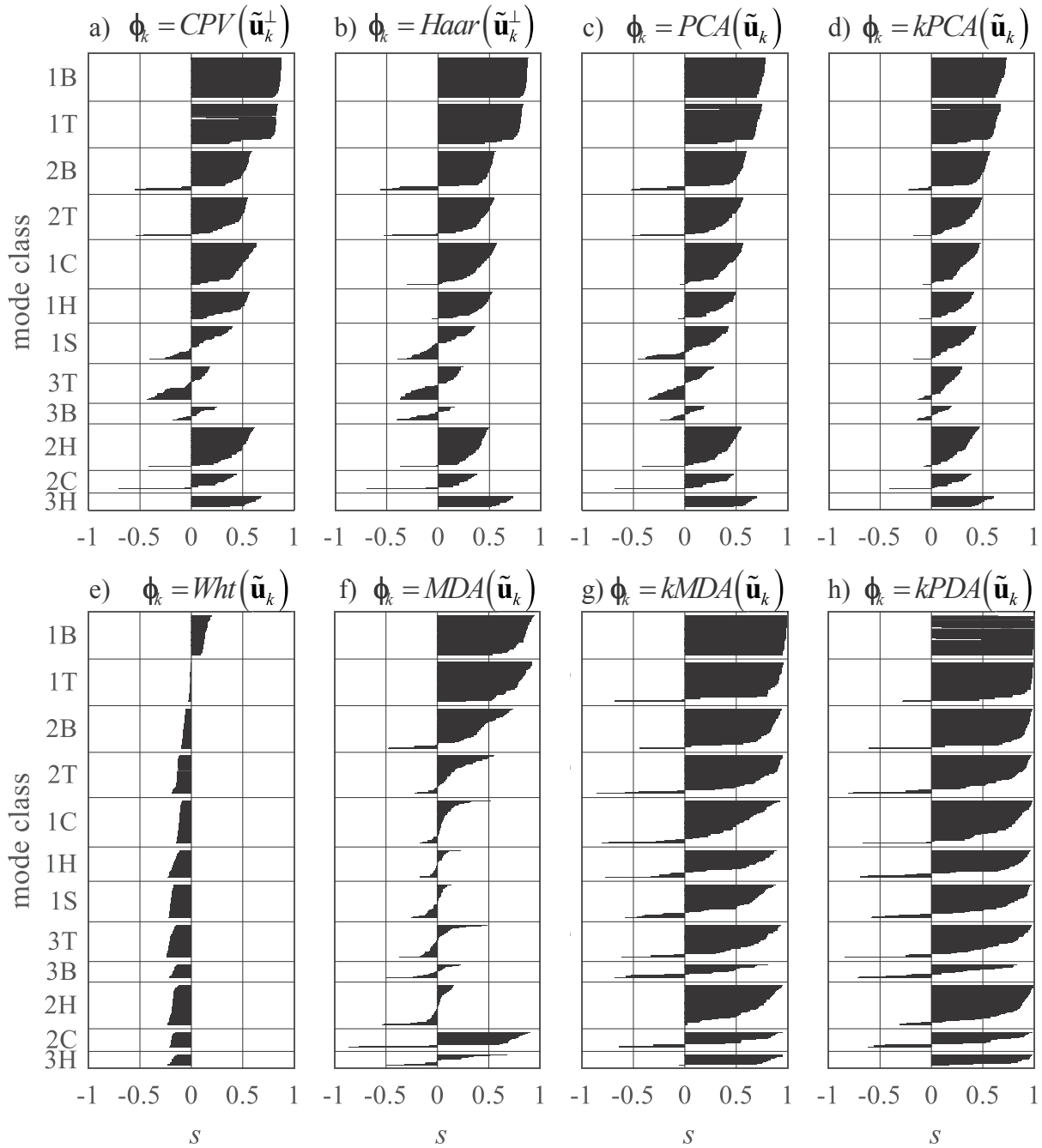


Figure 5.11: Silhouette plots of the different dimension reduction methods

(5.5) are projected onto directions of highest variances ( $\phi_k = \text{PCA}(\tilde{\mathbf{u}}_k)$ ), see Fig. 5.11c, neglecting information (features) of little variance, i.e., little information value. However, a significant improvement is achieved when nonlinear PCA ( $\phi_k = \text{kPCA}(\tilde{\mathbf{u}}_k)$ ) is used with a quadratic polynomial kernel function (polynomial kernel functions with higher order than quadratic or Gaussian kernel function did not perform better), see Fig. 5.11d.

The benefit of kPCA is lost if the data are whitened, see Fig. 5.11e. To understand the reason, a lower dimensional view on the data structure for a test case is given in Fig. 5.12. It is known from Appendix B that the eigenmodes are distinguished by the amplitude

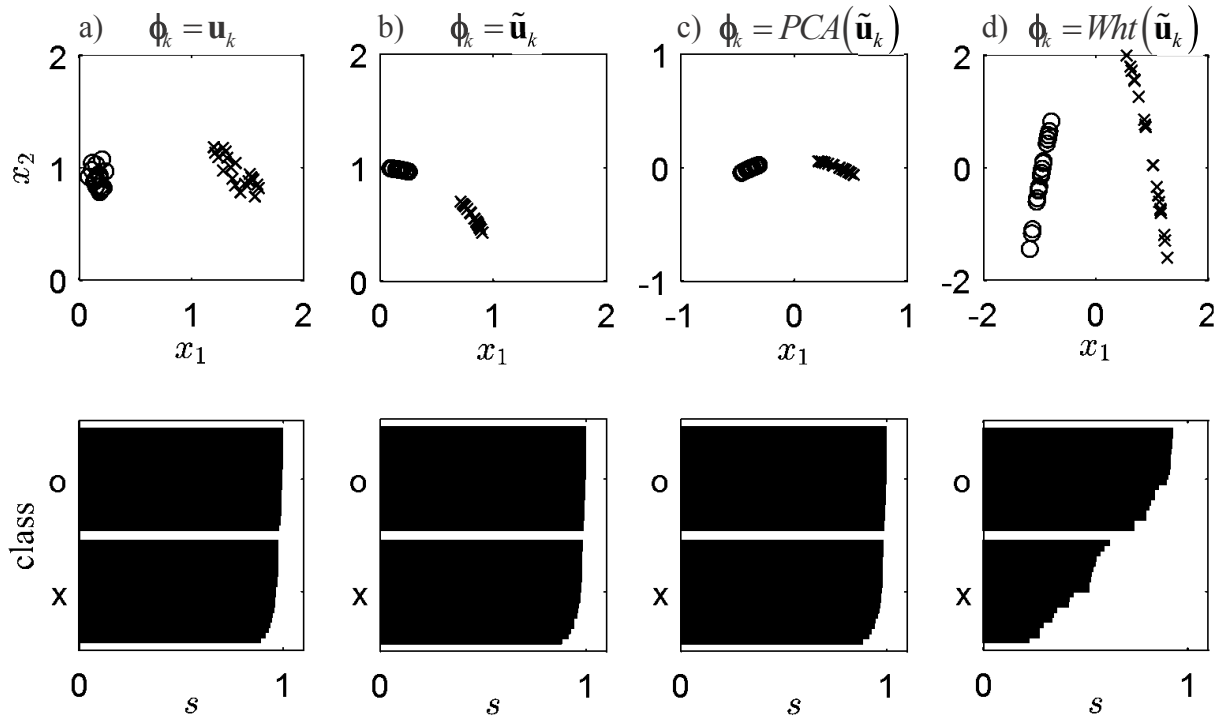


Figure 5.12: Effect of b) normalization, c) PCA, and d) whitening on data structure (a) with corresponding silhouette plots

ratio of the displacement field, but not by the absolute value of the amplitudes. Therefore, the test case in Fig. 5.12a contains two clusters representing two different eigenmode classes, one cluster with little variance in the amplitude ratios but larger variance in amplitude level (unambiguous mode type), and one cluster of visa versa characteristic. The silhouette values show that the test data are well separated and separation is improved by normalization (Fig. 5.12b; data is located on a sphere of radius one) and PCA (Fig. 5.12c; data are centered and rotated into direction of main variance). Thus, the reason for the good performance of PCA and also kPCA in the low-dimensional example as well as in the high-dimensional application (Fig. 5.11) is that after redirection and normalization (data relocated on one half of a hypersphere) the main variance directions roughly coincide with good discriminating directions. But in case of data whitening (Fig. 5.12d), the possible set of discriminates reduces due to the increased variance in ordinate direction, i.e., the separation of the data deteriorates. The reason is that the direction of highest variance is also the direction of best separation and whitening adapts the variances of the data set to become all equal to one. Hence, the distinguishing amplitude ratio becomes less and the negligible amplitude level more dominant.

Interestingly, using MDA to filter for the information that best separate the clusters ( $\phi_k = \text{MDA}(\tilde{\mathbf{u}}_k)$ ), see Fig. 5.11f, has a poor effect on cluster separation, as well. Contrariwise,  $\phi_k = \text{kMDA}(\tilde{\mathbf{u}}_k)$  in Fig. 5.11g gives a significant improvement of the data structure. The reason is that in the present case kMDA is much less prone to changes in the data set used to calculate the discriminates. It shall be recapitulated that calculation

of the discriminates requires knowledge of the classification of each sample within the reference set used for the analysis. Therefore, the silhouette value for each  $\boldsymbol{\phi}_k \in \mathcal{V}^{r_k}$  of a compressor row  $r_k$  has to be calculated based on the MDA projection received from  $\mathcal{V} \setminus \mathcal{V}^{r_k}$ . Nonetheless, a calculation of the silhouette values based on the MDA projection received from  $\mathcal{V}$  was performed and the values are all close to one, which means that the data set  $\mathcal{V}$  is actually linearly separable, but the discriminates are too sensitive to changes in the data set for the appropriately applied MDA to be performing well. In order to analyze the sensitivity of MDA to changes in the data set, the modal-assurance criterion (C.54) between the discriminants  $\mathbf{v}^i$ , based on the whole reference set  $\mathcal{V}$  using (C.39) and the discriminants  $\mathbf{v}_r^i$  calculated from  $\mathcal{V} \setminus \mathcal{V}^r$  is evaluated as follows:

$$MAC_r^i := MAC(\mathbf{v}^i, \mathbf{v}_r^i) \in [0, 1] . \quad (5.19)$$

The MAC-value is bounded between zero and one, where a value of zero means that  $\mathbf{v}^i$  and  $\mathbf{v}_r^i$  are perpendicular and a value of one means that they are fully aligned (Appendix C.2). The results in Fig. 5.13 indicate that especially compressor rows 6, 8, 13, and 27 contain eigenmodes which strongly affect the cluster means and variances and, therefore, the discriminants. Unfortunately the same evaluation is not possible for kMDA because the size of the scatter matrices in Eq. (C.51) is equal to the number of samples in  $\mathcal{V}$  respectively  $\mathcal{V} \setminus \mathcal{V}^r$  which is why the size of the eigenvectors is not consistent.

Due to the good performance of kPCA and kMDA the question arises if their combination, i.e.,  $\boldsymbol{\phi}_k = \text{kMDA}(\text{kPCA}(\tilde{\mathbf{u}}_k)) \doteq \text{kPDA}(\tilde{\mathbf{u}}_k)$  would give further improvement of the data structure. As shown in Fig. 5.11h, this method causes higher positive but also

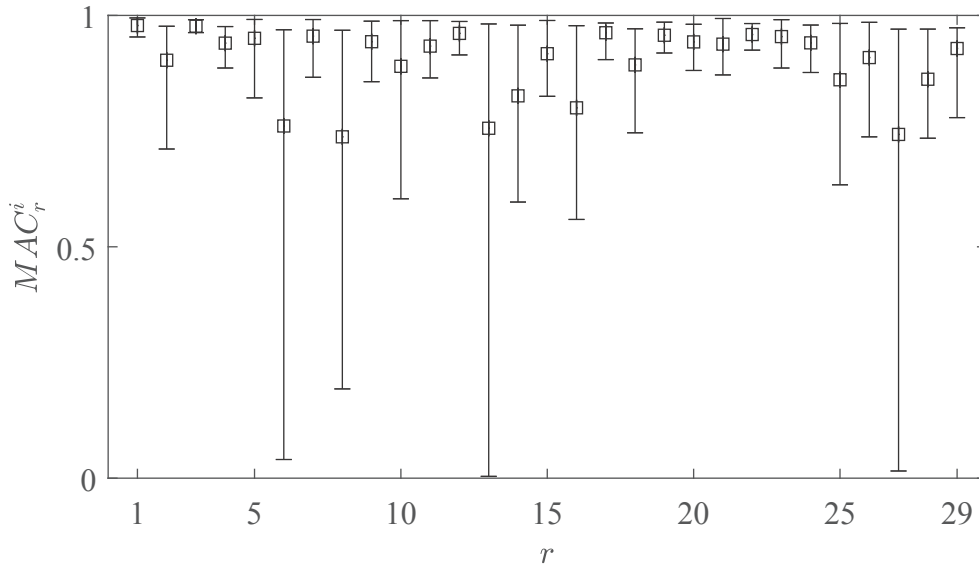


Figure 5.13: Mean value (□) and the range between minimum and maximum MAC-value for discriminants calculated based on reference set  $\mathcal{V}$  and reduced reference set  $\mathcal{V} \setminus \mathcal{V}_r$

negative silhouette values and judging whether this means an improvement or not has to be made by evaluating the classification performance later on.

All the described effects of dimension reduction onto the data structure w.r.t. redirecting the eigenvectors in accordance to (5.2) also hold for (5.3), see Fig. A.3. In direct comparison to Fig. 5.11, the beneficial effects by using (5.3) almost vanish for all dimension-reduction methods except for  $\boldsymbol{\phi}_k = \text{PCA}(*\tilde{\mathbf{u}}_k)$  and the 1S-mode cluster.

In Section 5.1, based on the distance ratio (5.12), it was shown that the majority of the negative silhouette values is caused by cluster imbalance and not by cluster intersection. The effect of dimension-reduction onto the distance ratio may be observed in Fig. 5.14. Comparing Fig. 5.14a–d with Fig. 5.4b–c shows that those dimension-reduction methods have negligible effect onto cluster intersection, which seems to be surprising but is reasonable, since all those methods preserve most of the relevant content of the data (w.r.t. variances). For data whitening and MDA (Fig. 5.14e–f), however, there is noticeable deterioration in cluster separation. Using nonlinear MDA instead (Fig. 5.14g–h) hardly effects the number of cluster intersections (negative distance ratios), but slightly increases those that are existent. Using redirection method (5.3) instead of (5.2) decreases those intersections but has negligible influence on the distance ratio otherwise, see Fig. A.4.

A more detailed insight into the effects of dimension reduction onto the neighborhood structure of the clusters is given in Fig. 5.15. No noteworthy influence can be seen by using  $\boldsymbol{\phi}_k = \text{CPV}(\tilde{\mathbf{u}}_k)$  and  $\boldsymbol{\phi}_k = \text{Haar}(\tilde{\mathbf{u}}_k)$  (Fig. 5.15a-b) instead of  $\boldsymbol{\phi}_k = \tilde{\mathbf{u}}_k$  (Fig. 5.5b), but the dimensionality has decreased significantly from  $D = 4800$  to  $d = 157$  and  $d = 16$ , respectively. A clear improvement onto the homogeneity of the distances between the clusters is achieved by using  $\boldsymbol{\phi}_k = \text{PCA}(\tilde{\mathbf{u}}_k)$  and especially  $\boldsymbol{\phi}_k = \text{kPCA}(\tilde{\mathbf{u}}_k)$  (Fig. 5.15c&d). Interestingly the neighborhood structure is nearly identical for both PCA and kPCA. Whitening (Fig. 5.15e), however, seems to cause intersections of all clusters with the 1B cluster. The poor silhouette results for  $\boldsymbol{\phi}_k = \text{MDA}(\tilde{\mathbf{u}}_k)$  are caused by small distances between the clusters (comparably small markers in Fig. 5.15f) and higher number of neighbors for each cluster. Even the 1B and 1T clusters, which have good silhouette values, have been shifted closer to other clusters, such that misclassification becomes more likely. The improved silhouette results for  $\boldsymbol{\phi}_k = \text{kMDA}(\tilde{\mathbf{u}}_k)$  and  $\boldsymbol{\phi}_k = \text{kPDA}(\tilde{\mathbf{u}}_k)$  do not result from a more homogeneous distance between the clusters but from fewer direct neighbors (especially 2B). The combination of kPCA and kMDA reduction, see Fig. 5.15h, gives fewer direct neighbors than just using kMDA, but those left are located closer and no clear advantage can be determined.

The effects onto neighborhood structure by using (5.3) instead of (5.2) are negligible for any method not related to MDA, see Fig. A.5. However, based on the conclusions drawn from the investigations here, the best combination of cluster separation and homogeneity is achieved with kPCA. Therefore, using either (5.2) or (5.3) makes no notable difference.

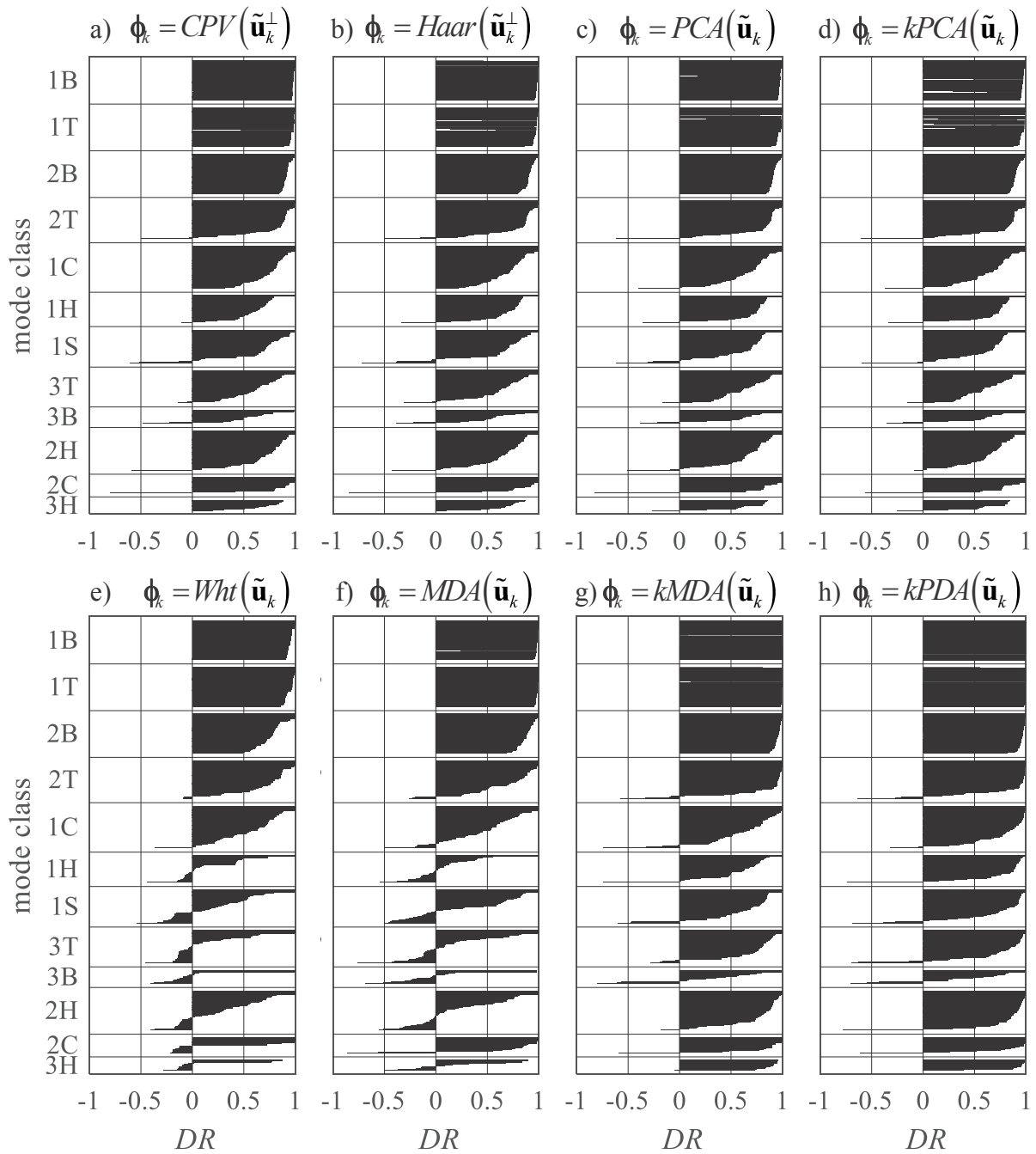


Figure 5.14: Effect of different dimension reduction methods onto the distance ratio



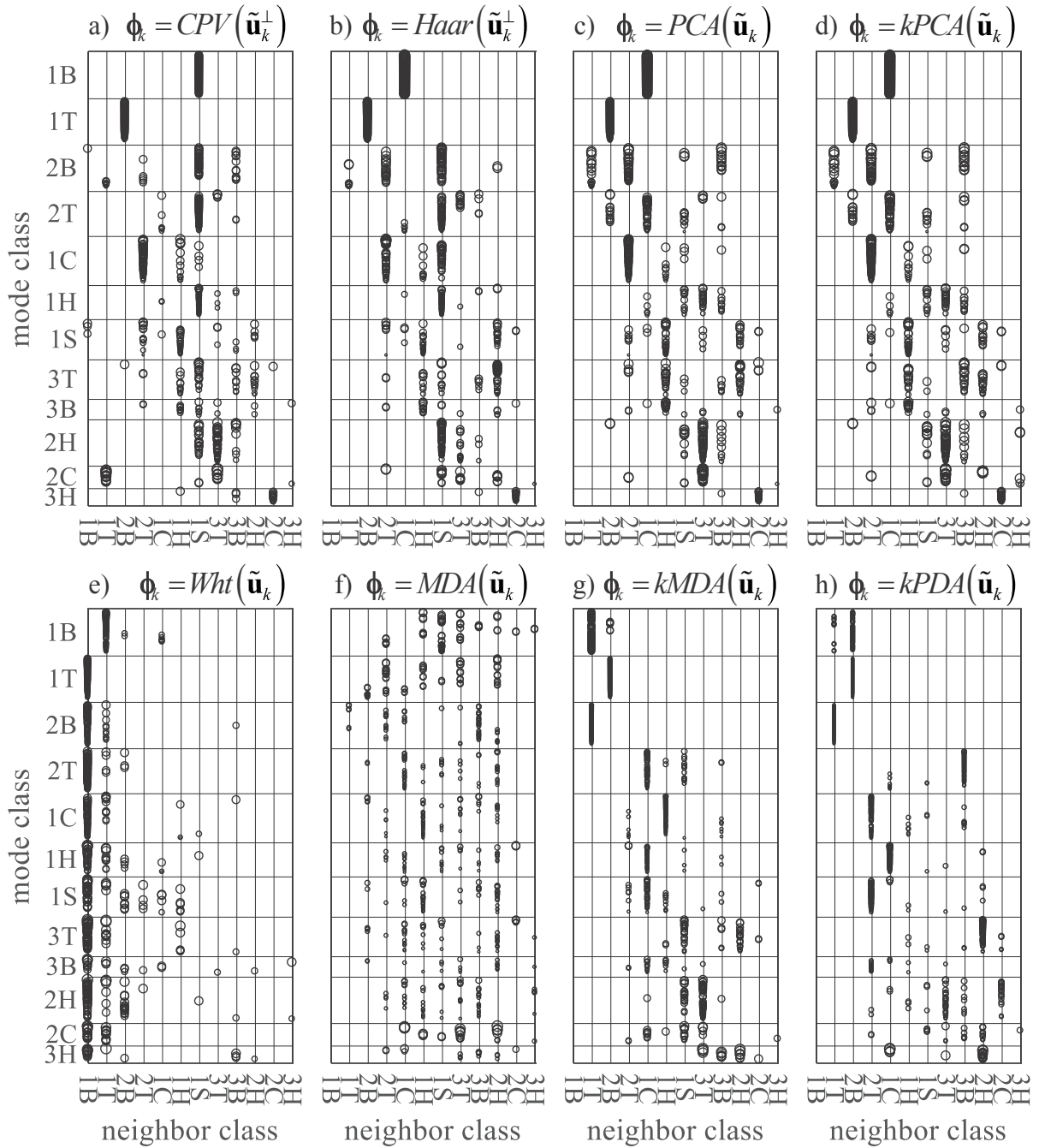


Figure 5.15: Effect of different dimension reduction methods onto the type of cluster neighbors and distance (size of marker)



# 6 Classification of Eigenmodes and Assignment of Frequency Bands

In this chapter it shall be investigated which classification algorithm from a selection of some common algorithms results in the highest probability to assign the correct class to an unknown eigenmode shape. In order to increase the data density and improve data structure for the classification algorithms, eigenmode data have been redirected, normalized, and reduced in dimension by various methods described in previous Chapter 5. However, according to the "no free lunch theorem" described by Wolpert (1996), classification performance will depend on how well the data structure fits to the classification algorithm, i.e., a dimension reduction method with poorer effect on cluster separation does not necessarily perform poor in classification and visa versa. Thus, in order to identify eigenmode shapes with lowest risk of misclassification, assessing the performance of different classification algorithms has to be done in combination with the different dimension reduction methods from Chapter 5.

Despite classification of eigenmode shapes with possibly low error rate, assigning reasonable eigenfrequency bands may be challenging due to the possibility that specific eigenmode shapes may not be existent at both lower and upper limit of predicted eigenfrequencies. Based on the assessment in Section 3.4 a procedure for assigning proper eigenfrequency bands to specific mode shapes will be introduced which incorporates the best suited classification method identified. This procedure will be introduced within the last section of this chapter, after different classification algorithms have been introduced and their performance has been assessed.

## 6.1 Classification of Eigenmode Shapes

In order to be able to assign a specific mode-shape class  $\tilde{c}$  to a new unclassified sample  $\tilde{\phi}$ , all classifiers require a training set  $\mathcal{V}_T = \{\phi_k\}$  of vectors with known classification  $\{c_k\}$ . The training intends to adjust a classifiers free parameters in such a way that classification error becomes minimal. In case of iteratively trained classifiers, overfitting of the free parameters to the training set is prevented by stopping the training after it performs with minimum classification error on an independent validation set  $\mathcal{V}_V$  i.e.  $\mathcal{V}_V \cap \mathcal{V}_T = \emptyset$  containing elements of known classification. Because the choice of the training and validation set influences the classification performance, their definition is explained before the performance of the

classifiers is assessed.

### 6.1.1 Nearest-Neighbors Algorithm

The nearest-neighbor algorithm is one of the simplest classifiers. A new sample  $\tilde{\phi}$  with unknown class  $\tilde{c}$  is assigned with the class  $c_{k^*}$  of the closest neighboring training sample  $\phi_{k^*}$  from the training set  $\mathcal{V}_T$ :

$$k^* := \arg \min_{k: \phi_k \in \mathcal{V}_T} d_{st}(\tilde{\phi}, \phi_k) \rightarrow \tilde{c} := c_{k^*} \quad (6.1)$$

where  $d_{st}(\tilde{\phi}, \phi_k)$  is a distance metric calculating the distance between  $\tilde{\phi}$  and  $\phi_k$  which will be explained later. A graphical interpretation of the nearest-neighbor classifier may be given by a Voronoi diagram where any new  $\tilde{\phi}$  within the Voronoi region

$$\mathcal{R}_k := \left\{ \tilde{\phi} \mid d_{st}(\tilde{\phi}, \phi_k \in \mathcal{V}_T) < d_{st}(\tilde{\phi}, \phi_j \in \mathcal{V}_T) \quad \forall j \neq k \right\} \quad (6.2)$$

of  $\phi_k$  is assigned with the class  $c_k$  (Lee (1982)). Figure 6.1a shows Voronoi cells for data that have been randomly sampled inside and outside a quarter of a circle (dotted line) giving two sets of samples defined as class one and two. It can be seen that the decision boundary of the nearest neighbor classifier is formed by borders of the Voronoi cells (solid black lines).

An extension of Eq. (6.1) to majority voting is the k-nearest-neighbors algorithm (kNN) which assigns the most frequent class among the  $k$  nearest neighbors in  $\mathcal{V}_T$  to  $\tilde{\phi}$ . This includes  $k = 1$  as nearest-neighbor algorithm. Using  $k > 1$  reduces the risk of overfitting the data and may help to receive a model that generalizes better, see Fig 6.1b. If, however, the number of neighbors  $k$  is chosen too high, the decision border will be pushed towards the less dense cluster at first, see Fig 6.1c, but as  $k$  approaches the total number of samples

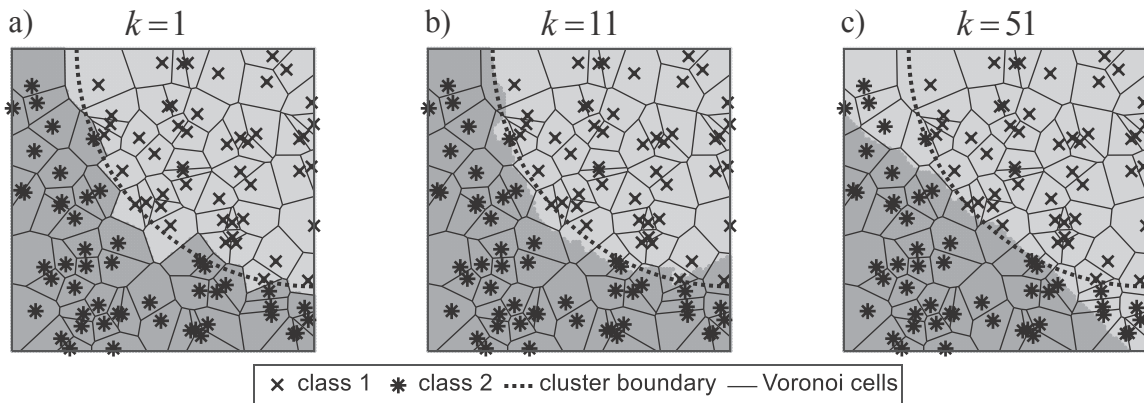


Figure 6.1: Binary classification (different shades of gray; 50 samples per cluster) with kNN algorithm for different numbers of neighbors  $k$  ( $k = 1$  represents nearest-neighbor algorithm)

(combined of both classes) the decision will be in favor of the cluster with the most samples. In order to suppress this undesired behavior, the votes may be weighted by the distance of each neighbor to  $\tilde{\boldsymbol{\phi}}$  (Hechenbichler and Schliep (2004)). However, this strategy is not applied in the present investigation.

The performance of the kNN-algorithm depends on the type of distance metric and is sensitive to the local structure of  $\mathcal{V}_T$ . Considering the continuous nature of the eigenmode data, two different distance metrics seem possible, the Euclidean distance

$$d_{st}(\tilde{\boldsymbol{\phi}}, \boldsymbol{\phi}_k) := \sqrt{(\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}_k)^T (\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}_k)} \quad (6.3)$$

and the cosine distance

$$d_{st}(\tilde{\boldsymbol{\phi}}, \boldsymbol{\phi}_k) := 1 - \frac{\tilde{\boldsymbol{\phi}}^T \boldsymbol{\phi}_k}{\|\tilde{\boldsymbol{\phi}}\| \|\boldsymbol{\phi}_k\|} . \quad (6.4)$$

The cosine distance measures the angle between two vectors and is, therefore, promising w.r.t. to the raw data where no normalization and dimension reduction has been applied to. From the study in Appendix B it is known that eigenmodes are distinguished by their amplitude ratios not their amplitude level. For this reason the cosine distance is commonly used in modal analysis, but scaled reversely, to be used as a correlation measure for mode agreement, the so called modal-assurance criterion (C.54), see Appendix C.3. Applied to mode classification by nearest-neighbor algorithm, the goal is to find the training vector  $\boldsymbol{\phi}_{k^*}$  with the highest MAC-value to  $\tilde{\boldsymbol{\phi}}$  and then to assign the corresponding class  $c_{k^*}$ :

$$MAC_k := MAC(\tilde{\boldsymbol{\phi}}, \boldsymbol{\phi}_k) \in [0, 1] \quad \forall \boldsymbol{\phi}_k \in V_T \rightarrow k^* = \arg \max_k MAC_k \rightarrow \tilde{c} := c_{k^*} . \quad (6.5)$$

Because using cosine distance (6.4) in Eq. (6.1) is equivalent to (6.5), the assessment in the following section will refer to  $MAC_k$  only. Improvements of the performance of the nearest-neighbor algorithm may be achieved by learning a metric, e.g., using neighborhood-components analysis (Goldberger et al. (2004)), but such methods will not be investigated here.

Instead of evaluating MAC between  $\tilde{\boldsymbol{\phi}}$  and the samples in  $\mathcal{V}_T$ , one may also find the class  $\tilde{c}$  by evaluating the correlation between  $\tilde{\boldsymbol{\phi}}$  and the cluster means  $\mathbf{m}_c = (1/|\mathcal{V}_c|) \sum_{k:\boldsymbol{\phi}_k \in \mathcal{V}_c} \boldsymbol{\phi}_k$  where  $\mathcal{V}_c$  is defined in Eq. (5.8). Then, the assignment analogously to Eq. (6.5) is as follows:

$$MAC^c := MAC(\tilde{\boldsymbol{\phi}}, \mathbf{m}_c) \in [0, 1] \rightarrow \tilde{c} = \arg \max_c MAC^c . \quad (6.6)$$

## 6.1.2 Support-Vector Machine

A support-vector machine (SVM) by Vapnik and Lerner (1963) is an algorithm which seeks an optimal discriminating hyperplane  $\mathbf{n}^T \tilde{\boldsymbol{\phi}} + b = 0$  between data points of two different classes, see Fig. 6.2a. The resulting hyperplane is optimal in the sense that it represents

the border with widest margin between the cluster bounds. The algorithm filters for those data points which actually influence the decision on the optimal hyperplane; these data points are referred to as support vectors. The SVM algorithm can only find linear hyperplanes (discriminants) and in case of non-separable data a trade-off parameter (slack) has to be defined to balance between maximization of the margin  $2/|\mathbf{n}|$  and minimization of violations (Fig. 6.2b). However, indirectly the algorithm can account for nonlinear directions of separation by implementing the kernel trick (Boser et al. (1992)), see Fig. 6.2c. For more details, a description of the mathematics of the SVM algorithm is given in Appendix C.4.

The goal of SVM is very similar to MDA/kMDA, but the advantage of SVM over MDA/kMDA is that SVM finds the discriminate based on the local data structure and the decision is influenced by the support vectors only. In contrast to this, MDA/kMDA uses the statistical representation of the data via the scatter matrices which can cause poor results in case the shape of the clusters deviates much from a sphere and the sample distribution within the clusters is inhomogeneous. Additionally, the resulting discriminants can be very sensitive to changes in the data set even if those changes do not affect the boundary region between the clusters.

As SVM is a binary classifier, application to multiple class problems requires to calculate multiple binary classifiers. Thereby, two approaches are commonly used, the one-vs-rest (also known as one-vs-all) and the one-vs-one approach. The one-vs-rest approach requires  $N_C$  binary classifiers, one for each class, where the opponent class is formed by the combined samples of all other classes. A new  $\hat{\phi}$  is then assigned with the class that belongs to the binary classifier with the highest score. On the contrary, the one-vs-one approach requires  $N_C(N_C - 1)/2$  binary classifiers (one for each possible pairing of the classes) and the class with the highest score on votes from all binary classifiers is assigned to a new data point. The performance of both methods on test data from the UCI machine learning repository (Asuncion and Newman (2007)) has been investigated by Hsu and Lin

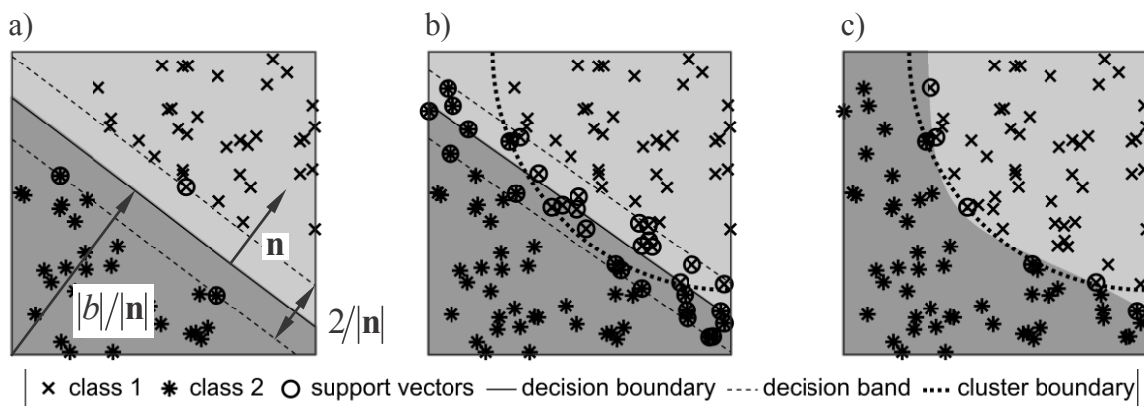


Figure 6.2: Binary classification (different shades of gray; 50 samples per cluster) with SVM algorithm: a) linear for linearly separable data, b) linear for nonlinearly separable data (with slack), and c) nonlinear for nonlinearly separable data

(2002) as well as Duan and Keerthi (2005) but the results are contradictorily w.r.t. which method performs better. Along with different cross-validation strategies, the main reason for the inconsistencies may be the usage of a different training set size which has major influence on the results as shown by Duan and Keerthi (2005). Without further analyses of the data structure of the UCI test-data sets and considering that in general the decision boundaries will be much more complex with the one-vs-rest approach, the one-vs-one method should be preferred. For example, imagine an arbitrary constellation of pairwise linearly separable clusters, e.g., several clusters located on a sphere and one in the middle, then the decision boundary between one arbitrary cluster and the rest will be nonlinear but linear for any one-vs-one classifier. Therefore, the data set used throughout this thesis, where the amplitude ratio of the eigenmodes is the distinguishing information, contains a cluster structure that should benefit from the one-vs-one approach, and a direct comparison actually showed a slightly better performance compared to the one-vs-rest approach.

### 6.1.3 Feed-Forward Neural Network

Feed-forward-neural networks (FFNN) in combination with back-propagation training (back-propagation-neural network (BPNN)) are one of the most powerful learning algorithms and are widely used in autonomous driving, face recognition, and other pattern-recognition tasks. A specially designed BPNN suggested by Ciregan et al. (2012) even outperforms humans on the task of traffic sign recognition. Such networks belong to supervised learning algorithms, i.e., data with known outcome are used to train the weights of network synapses. A simple design of a fully connected three-layer FFNN with input (IL), hidden (HL), and output (OL) layer, as described by Bishop (2006), is shown in Fig. 6.3. Each layer contains neurons, each of which processes a signal according to its transfer function ( $h_l$  for input-layer neurons;  $f_j$  for hidden-layer neurons;  $g_c$  for output-layer neurons) and is connected to all neurons of the previous and subsequent layer.

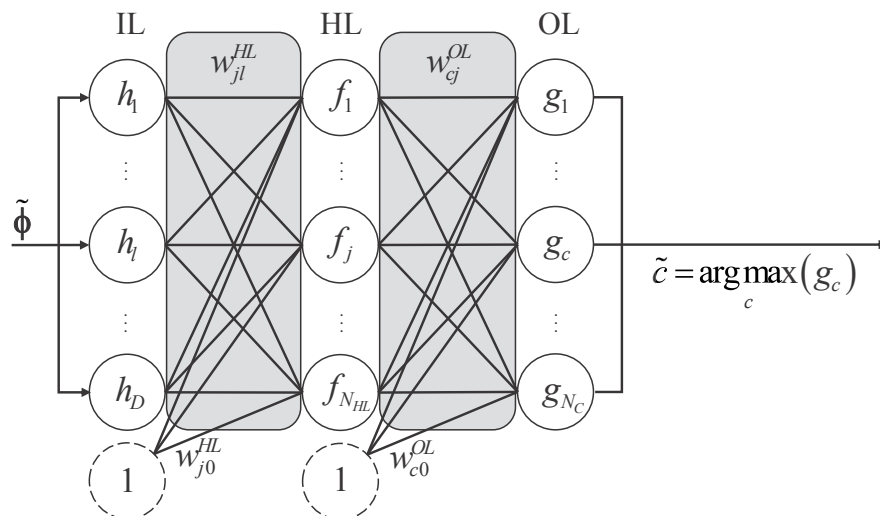


Figure 6.3: Fully connected feed-forward-neural network with three layers

The connections, also called synapses, are associated with weights  $w_{jl}^{HL}$  connecting the  $l^{\text{th}}$  input-layer neuron with the  $j^{\text{th}}$  hidden-layer neuron and  $w_{cj}^{OL}$  connecting the  $j^{\text{th}}$  hidden-layer neuron with the  $c^{\text{th}}$  output-layer neuron. These weights account for the strength of each connection. The hidden and output-layer neurons are additionally connected to bias neurons, with constant output one, by connection weights  $w_{j0}^{HL}$  and  $w_{c0}^{OL}$ .

The network processes a signal by first splitting an input vector  $\tilde{\boldsymbol{\phi}} \in \mathbb{R}^D$  into its components and each component  $\tilde{\phi}_l$  is the input to the transfer function  $h_l$  of the  $l^{\text{th}}$  input-layer neuron. Commonly  $h_l$  is set to be a linear function without offset or shift (Duda et al. (2000)) such that the input  $\tilde{\phi}_l$  is simply transmitted to the output of the  $l^{\text{th}}$  input-layer neuron. The output of the  $l^{\text{th}}$  input-layer neuron is then linked to all neurons of the hidden layer where the strength of the output processing is controlled via the weights  $w_{jl}^{HL}$ . Thus, the input  $t_j^{HL}$  to each neuron of the hidden layer (net activation) is a specific weighted sum of the outputs of all input-layer neurons plus a bias  $w_{j0}^{HL}$ , i.e.,

$$t_j^{HL} := \sum_{l=1}^D (h_l w_{jl}^{HL}) + w_{j0}^{HL} \quad (6.7)$$

or

$$\mathbf{t}^{HL} = \mathbf{W}^{HL} \mathbf{h} + \mathbf{w}_0^{HL} = \left[ \mathbf{W}^{HL} \mid \mathbf{w}_0^{HL} \right] \begin{bmatrix} \mathbf{h} \\ 1 \end{bmatrix} \quad (6.8)$$

where  $\mathbf{h} = [h_1, \dots, h_D]^T$  is the output vector of the input layer,  $\mathbf{W}^{HL} = [w_{jl}^{HL}]$  is a connection matrix,  $\mathbf{w}_0^{HL}$  is the bias vector of the hidden layer, and  $\mathbf{t}^{HL} = [t_1^{HL}, \dots, t_{N_{HL}}^{HL}]^T$  is the activation vector of the hidden layer. Accordingly, the net activations  $t_c^{OL}$  of the neurons of the output layer are weighted sums of the hidden-layer outputs  $f_j := f(t_j^{HL})$ , i.e.,

$$t_c^{OL} := \mathbf{f}^T \mathbf{w}_c^{OL} + w_{c0}^{OL} = \left[ \mathbf{w}_c^{OLT} \mid w_{c0}^{OL} \right] \begin{bmatrix} \mathbf{f} \\ 1 \end{bmatrix}, \quad (6.9)$$

where  $\mathbf{f} = [f_1, \dots, f_{N_{HL}}]$  is the output vector of the hidden layer and  $\mathbf{w}_c^{OL} = [w_{c1}^{OL}, \dots, w_{cN_{HL}}^{OL}]^T$  is the weight vector of the links from the hidden layer to the  $c^{\text{th}}$  neuron of the output layer. In classification tasks, the index  $c$  of the output layer neuron with the maximum output determines the class  $\tilde{c}$  assigned to the input signal  $\tilde{\boldsymbol{\phi}}$ .

The outputs of the hidden- and output-layer neurons w.r.t. their net activations is controlled by the transfer function  $f_j$  and  $g_c$ . Those transfer functions have to be bounded in order to prevent the weights of the network links (synapses) from diverging during training (see Appendix C.5 for more details). Additionally, it is desired that the hidden-layer transfer functions are linear in some place in order to keep the model simple wherever possible and nonlinear where more complexity is required (Duda et al. (2000)). This is



achieved by choosing the hyperbolic-tangent function

$$f_j := f(t_j^{HL}) = \frac{2}{1 + e^{-2t_j^{HL}}} - 1, f : \mathbb{R} \rightarrow [-1, 1]. \quad (6.10)$$

In contrast to this, the output-layer transfer functions for classification task shall scale the output w.r.t. the competing outputs which is commonly achieved via the softmax function

$$g_c := g(t_1^{OL}, \dots, t_{N_C}^{OL}) = \frac{e^{t_c^{OL}}}{\sum_{s=1}^{N_C} e^{t_s^{OL}}}, g : \mathbb{R}^{N_C} \rightarrow (0, 1]. \quad (6.11)$$

By considering Eqns. (6.8) and (6.9), it becomes clear that the biases  $w_{j_0}^{HL}$  and  $w_{c_0}^{OL}$  allow to control the left-right shift of the transfer functions. Thus,  $w_{j_0}^{HL}$  allows to tune the outputs  $f_j$  between linear for  $|t_j^{HL}|$  and nonlinear w.r.t. the inputs  $t_j^{HL}$ , and  $w_{c_0}^{OL}$  allows to tune the average classification decision of a specific class  $c$ .

For the adaption of the network weights  $w_{jl}^{HL}$ ,  $w_{j_0}^{HL}$ ,  $w_{cj}^{OL}$ , and  $w_{c_0}^{OL}$  during the training of such networks, commonly gradient-based optimization algorithms are applied where the training-error gradient is back-propagated through the network as described in Appendix C.5. Because of the large number of parameters/weights, these algorithms perform better than most others. However, gradient-descent algorithms are likely to get stuck in a local optimum. Therefore, the initial setting of the weights is usually generated randomly and the procedure of initialization and training the network is repeated multiple times until a setting is found with a sufficiently small value of the local optimum. Alternatively, it is suggested in the following to initialize the weights close to a state that resembles the desired operation of the network such that optimization is more effectively driven to a minimum (may still be local) which gives satisfactory performance of the network.

Based on the previous chapter it is known that PCA has a positive effect on the separation and structure of the clusters of the reference data. Furthermore, it was suggested that the maximal MAC between the cluster means  $\mathbf{m}_c$  and a test vector  $\tilde{\boldsymbol{\phi}} \in \mathbb{R}^D$  may be used to assign the class  $\tilde{c}$ , see Eq. (6.6). Considering that the net activations of the neurons of the input layer (6.8) and output layer (6.9) are scalar products of the previous layer outputs  $\mathbf{h} = [h_1, \dots, h_D]^T$  and  $\mathbf{f} = [f_1, \dots, f_{N_{HL}}]^T$  with the weight vectors  $\mathbf{w}_j^{HL} = [w_{j_1}^{HL}, \dots, w_{j_D}^{HL}]^T$  and  $\mathbf{w}_c^{OL} = [w_{c_1}^{OL}, \dots, w_{c_{N_{HL}}}^{OL}]^T$ , both PCA and a correlation criterion similar to  $MAC^c$  can be incorporated into the network structure in an elegant manner described by Martin and Bestle (2018).

The PCA transformation according to Eq. (C.10) can be represented by the first two layers and their connections. Therefore, the transfer functions  $h_l$  of the input layer are used to centralize the data by choosing

$$\mathbf{h}(\tilde{\boldsymbol{\phi}}) = [h_1, \dots, h_D]^T := \tilde{\boldsymbol{\phi}} - \mathbf{m} =: \tilde{\tilde{\boldsymbol{\phi}}} \quad (6.12)$$

where the mean vector  $\mathbf{m} \approx \boldsymbol{\mu}$  is calculated from the training samples  $\boldsymbol{\phi}_k \in V_T$  using

Eq. (C.1).

The projection (C.10) of  $(\tilde{\boldsymbol{\phi}} - \mathbf{m})$  onto the principle components in  $\mathbf{V}_R$  is represented by the connections between the input- and hidden-layer through setting the initial connection matrix  $\mathbf{W}^{HL} = [w_{jl}^{HL}] := \mathbf{V}_R^T$ . Hence, the calculation of the net activations (6.8) coincides with the calculation of the PCA projection (C.10), i.e.,

$$\mathbf{t}^{HL} = \underbrace{\mathbf{W}^{HL}}_{=:\mathbf{V}_R^T} \mathbf{h} \stackrel{(6.12)}{=} \mathbf{V}_R^T (\tilde{\boldsymbol{\phi}} - \mathbf{m}) =: \tilde{\boldsymbol{\phi}}'_R \quad (6.13)$$

where  $N_{HL} := d$  and the hidden-layer bias weights are set to  $\mathbf{w}_0^{HL} := \mathbf{0}$ . A motivation of this initial setting may also be derived from the work of Baldi and Hornik (1989) who have shown that for symmetric neural networks with linear transfer functions the weights at the global optimum are the principle components.

Due to the normalization of the network inputs  $\tilde{\boldsymbol{\phi}}$  according to Eq. (5.5), any input vector is located on an unit sphere. Therefore, the subtraction of the training data mean  $\mathbf{m}$  within the input layer according to Eq. (6.12) will cause the net activations of the hidden layer (6.8) to become  $t_j^{HL} < 1$  since  $w_{j0}^{HL} = 0$ . This is true regardless the projection onto principle components via  $\mathbf{V}_R^T$ . In conclusion, net activations are within the linear region of the transfer functions  $f_j$  (see Eq. (6.10)) causing the hidden-layer outputs to equal the inputs, i.e.,  $\mathbf{f} \approx \mathbf{t}^{HL} = \tilde{\boldsymbol{\phi}}'_R$ . For this reason it is legitimate to utilize the output-layer weights  $w_{cj}^{OL}$  for measuring the correlation between the projected test vector  $\tilde{\boldsymbol{\phi}}'_R = \mathbf{t}^{HL} \approx \mathbf{f} = [f_1, \dots, f_{N_{HL}}]$  and the projected cluster means

$$\mathbf{m}'_c := \frac{1}{|\mathcal{V}_c|} \sum_{k:\boldsymbol{\phi}_k \in \mathcal{V}_c} \tilde{\boldsymbol{\phi}}'_{R,k} \text{ where } \tilde{\boldsymbol{\phi}}'_{R,k} = \mathbf{V}_R^T (\boldsymbol{\phi}_k - \mathbf{m}) \text{ ,} \quad (6.14)$$

which have to be calculated from a training set  $\mathcal{V}_T = \bigcup_{c=1}^{N_C} \mathcal{V}_c$ . The correlation may be assessed by scalar products  $\tilde{\boldsymbol{\phi}}_R'^T \mathbf{m}'_c$  similar to the MAC in Eq. (6.6). By setting  $\mathbf{w}_c^{OL} := \mathbf{m}'_c$ , this assessment is performed by the connections between the hidden and output layer, and net activations of the output layer (6.9) become

$$t_c^{OL} = \underbrace{\mathbf{f}^T}_{\approx \tilde{\boldsymbol{\phi}}_R'^T} \underbrace{\mathbf{w}_c^{OL}}_{=:\mathbf{m}'_c} \approx \tilde{\boldsymbol{\phi}}_R'^T \mathbf{m}'_c \text{ , } c = 1, \dots, N_C \text{ ,} \quad (6.15)$$

where the biases of the OL are again set to  $w_{c0}^{OL} := 0$ . Consequently, the neuron of the output layer with the highest net activation stands for the cluster with the highest correlation to  $\tilde{\boldsymbol{\phi}}$  and, thus, determines the class  $\tilde{c}$  of  $\tilde{\boldsymbol{\phi}}$  as

$$\tilde{c} = \arg \max_c \left( \tilde{\boldsymbol{\phi}}_R'^T \mathbf{m}'_c \right) \approx \arg \max_c \left( t_c^{OL} \right) = \arg \max_c (g_c) \quad (6.16)$$

where the transfer function  $g_c$  of the output-layer neurons is a softmax function (6.11)

which scales the net activation  $t_c^{OL}$  such that  $0 < g_c \leq 1$  and  $\sum g_c = 1$ .

In accordance to Eq. (C.24), kPCA can be used for initialization in a similar manner when using  $\tilde{\boldsymbol{\Phi}} := \bar{\mathbf{k}}_{\tilde{\Phi}}$  as inputs, setting transfer functions as  $\mathbf{h} := \tilde{\boldsymbol{\Phi}}$ , and hidden-layer weights as  $\mathbf{W}^{HL} := \mathbf{B}_R \boldsymbol{\Lambda}^+$ . The output-layer weights would then be defined as  $\mathbf{w}_c^{OL} := \hat{\mathbf{m}}'_c(\mathbf{f}'_k)$  similar to Eq. (6.14).

Up to now it was defined that  $w_{j_0}^{HL} := 0$  and  $w_{c_0}^{OL} := 0$ , and initializing the bias weights in such a way seems reasonable, since the approximate desired functionality of the network has already been specified by the suggested weight initializations in Eq. (6.13) and Eq. (6.15). However, tests have shown that the classification performance may be significantly improved if the bias weights are initialized as

$$\mathbf{w}_0^{HL} = -\mathbf{V}_R^T [1, \dots, 1]^T \quad \text{and} \quad w_{c_0}^{OL} = -\mathbf{m}_c'^T [1, \dots, 1]^T . \quad (6.17)$$

Starting from this initial setting, the network is able to fine tune the weights  $w_{j_l}^{HL}$ ,  $w_{j_0}^{HL}$ ,  $w_{c_j}^{OL}$ , and  $w_{c_0}^{OL}$  w.r.t. a training set  $\mathcal{V}_T$ . A detailed explanation of this training process is provided in Appendix C.5.

Because the setting (6.17) lacks reasonable justification at this point, it shall be further investigated. Before this investigation of the distributions of the biases and the correlation between initial setting before and adapted setting after the training of the network, a training and validation set have to be defined. The basis shall be reference set  $\mathcal{V}$  received from an industrial compressor as described in Section 5.1. This set is an union of  $N_R$  sets  $\mathcal{V}^r$  where each contains eigenmode vectors of known classification from a specific compressor row  $r$ , see Eq. (5.8). From reference set  $\mathcal{V}$  row-specific training and validation sets are received which allows to use cross-validation for assessing the average behavior of the network to training and classification data. This is done in such a way that the validation set  $\mathcal{V}_V^r$  for a row  $r$  contains the eigenmodes of two neighboring, geometrically similar rotors/stators (upstream and downstream), respectively, i.e.,

$$\mathcal{V}_V^r := \begin{cases} \bigcup_{i=r-2}^{N_R} \mathcal{V}^i \setminus \mathcal{V}^r & \text{if } r \geq N_R - 1 , \\ \bigcup_{i=r+2}^{r+2} \mathcal{V}^i \setminus \mathcal{V}^r & \text{if } r \leq 2 , \\ \bigcup_{i=r-2}^{i=1} \mathcal{V}^i \setminus \mathcal{V}^r & \text{else .} \end{cases} \quad (6.18)$$

The training set  $\mathcal{V}_T^r$  for a specific row  $r$  is defined as

$$\mathcal{V}_T^r := \mathcal{V} \setminus (\mathcal{V}_V^r \cup \mathcal{V}^r) . \quad (6.19)$$

First the bias distribution resulting from Eq. (6.17) shall be understood as well as the changes due to training of the network. For comparison, results with  $w_{j_0}^{HL} = 0$ ,  $w_{c_0}^{OL} = 0$  and randomly chosen bias weights (setting with best performance out of 20 random settings) will be evaluated. The first represents the initial approach from Eq. (6.13) and

Eq. (6.15), the latter common practice. Representative for results of all compressor rows, only that case will be shown where the data of the last row  $N_R$  are considered as unknown by using  $\mathcal{V}_T^{N_R}$  and  $\mathcal{V}_V^{N_R}$ . For now the elements of these sets are defined as  $\boldsymbol{\phi}_k = \tilde{\mathbf{u}}_k$  in accordance to Eq. (5.5). In Fig. 6.4 the bias distributions before and after the training are indicated by black solid and red dashed lines, respectively. Despite the three settings mentioned already and shown in Fig. 6.4a–c, a fourth setting is introduced in Fig. 6.4d and will be explained later. In general it can be seen that the bias adaptations due to the training are quite small. Actually they have about the same magnitude in all four cases

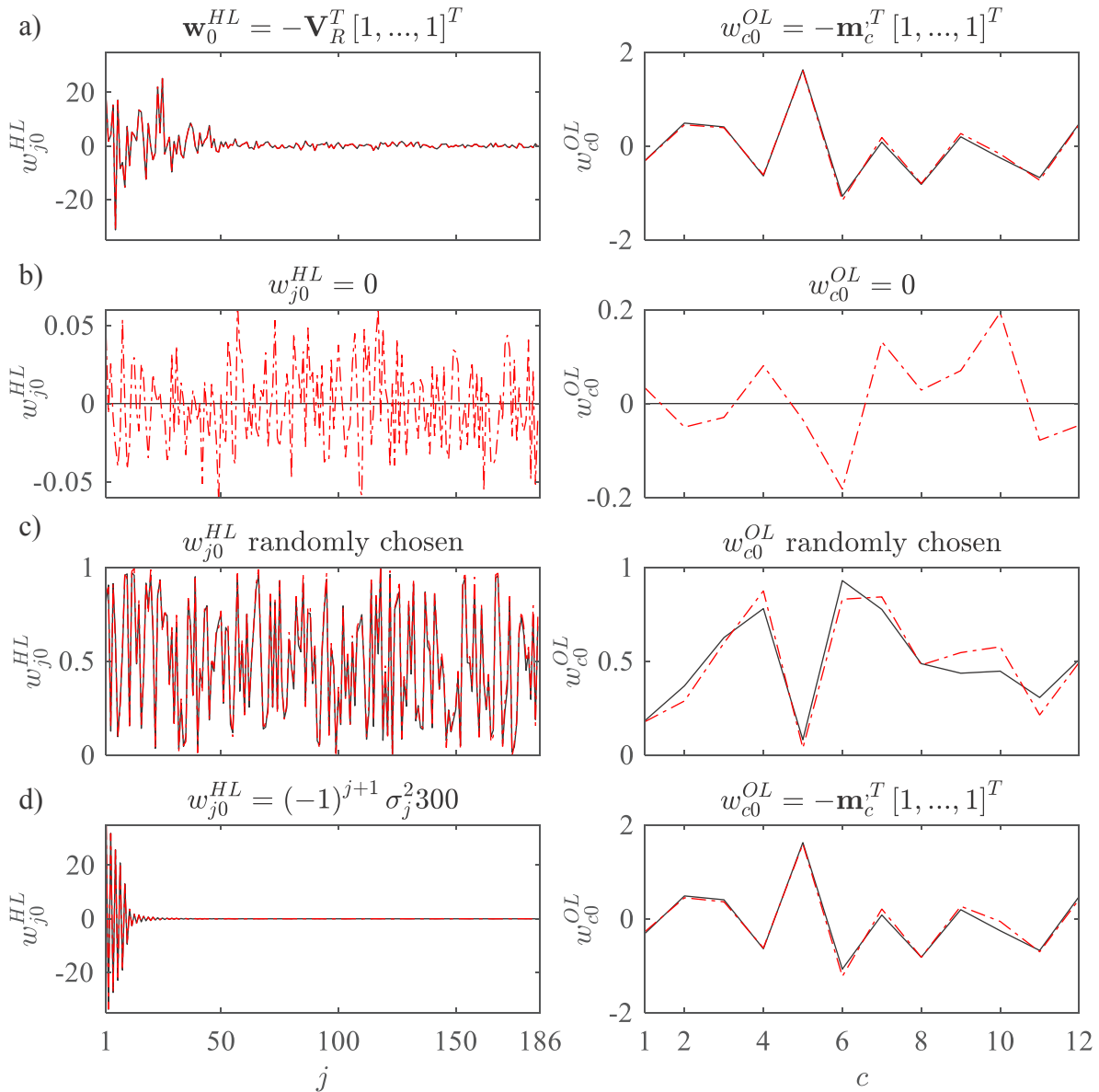


Figure 6.4: Biases  $w_{j0}^{HL}$  and  $w_{c0}^{OL}$  before (black line) and after training (red dashed line) for following initializations a) according to Eq. (6.17), b) all weights set to be zero, c) random initialization (best performance out of 20 random settings), and d) HL weights w.r.t. variances and OL according to Eq. (6.17)

despite the different axis scaling. However, initialization (6.17) has a favorable effect on classification performance which may be explained with the alternating characteristic and variance dependency in case of  $\mathbf{w}_0^{HL} = -\mathbf{V}_R^T [1, \dots, 1]^T$  in Fig. 6.4a. This dependency is visible in the larger fluctuations of the first 50 values which have been identified to contain  $\approx 99.8\%$  of data variance, see Fig. 5.9. For this reason a fourth setting has been tested in Fig. 6.4d which accounts for the variance associated with the principle components in an alternating manner. With this setting the classification performance improves for all rows in case of  $\boldsymbol{\phi}_k = \tilde{\mathbf{u}}$ . However, for input data that have been dimensionally reduced by some of the suggested methods in Section 5.2, initialization of the biases according to (6.17) gave slightly better classification performance.

At this point it shall be evaluated if setting (6.17) reflects a local optimum compared to random initialization for any of the introduced dimension reduction methods and if it performs well for all stages of the compressor. In order to clarify this, the MAC-value (C.54) between the initial bias weight vectors and the final ones is evaluated in Fig. 6.5. The correlation of the bias weights is the highest in Fig. 6.5a, d, f, and h which indicates that the network training gets directly trapped in a local optimum right from the start. Additionally, it might seem odd that Fig. 6.5a&d are very much alike, but it is rather logical because the HL weights  $w_{jl}^{HL}$  have been initialized with the principle components from PCA; so in case of Fig. 6.5d PCA is applied twice, first to the reference data and after within the HL of the network. The only effect is that the 99.99% filter on the principle components is applied twice in case that  $\boldsymbol{\phi}_k = \text{PCA}(\tilde{\mathbf{u}}_k)$  resulting in viewer principle components to be considered and slightly poorer classification performance which will be shown in the next section. The next section will also show that a dimension reduction method with high correlations w.r.t. Eq. (6.17) in Fig. 6.5 will have the largest improvement in classification performance when choosing (6.17) over random initialization. However, Fig. 6.5b, c, and g also make clear that initialization (6.17) is not an optimal choice in every case.

So far the investigation has shown that good performance w.r.t. initialization (6.17) is achieved in cases with high correlation between initial and final state of the bias weights. In conclusion more significant changes to the initial hidden- and output layer weights in Eq. (6.13) and Eq. (6.15) must be introduced by the training in order to explain the learning effect of the network and better performance over MAC as will be shown in the next section. That this is actually the case, can be seen in Fig. 6.6 where the hidden- and output-layer weight matrices have been deflated to vectors  $[w_{jl}^{HL}]$  and  $[w_{cj}^{OL}]$  in order to measure the MAC-correlation between initial state before and final state after training. However, the degree of correlation does not correlate with the classification performances evaluated in the next section which is no issue since the intention of the invented initialization is not to trap the network in this state, but to provide a good starting point for exploring a proper optimum.

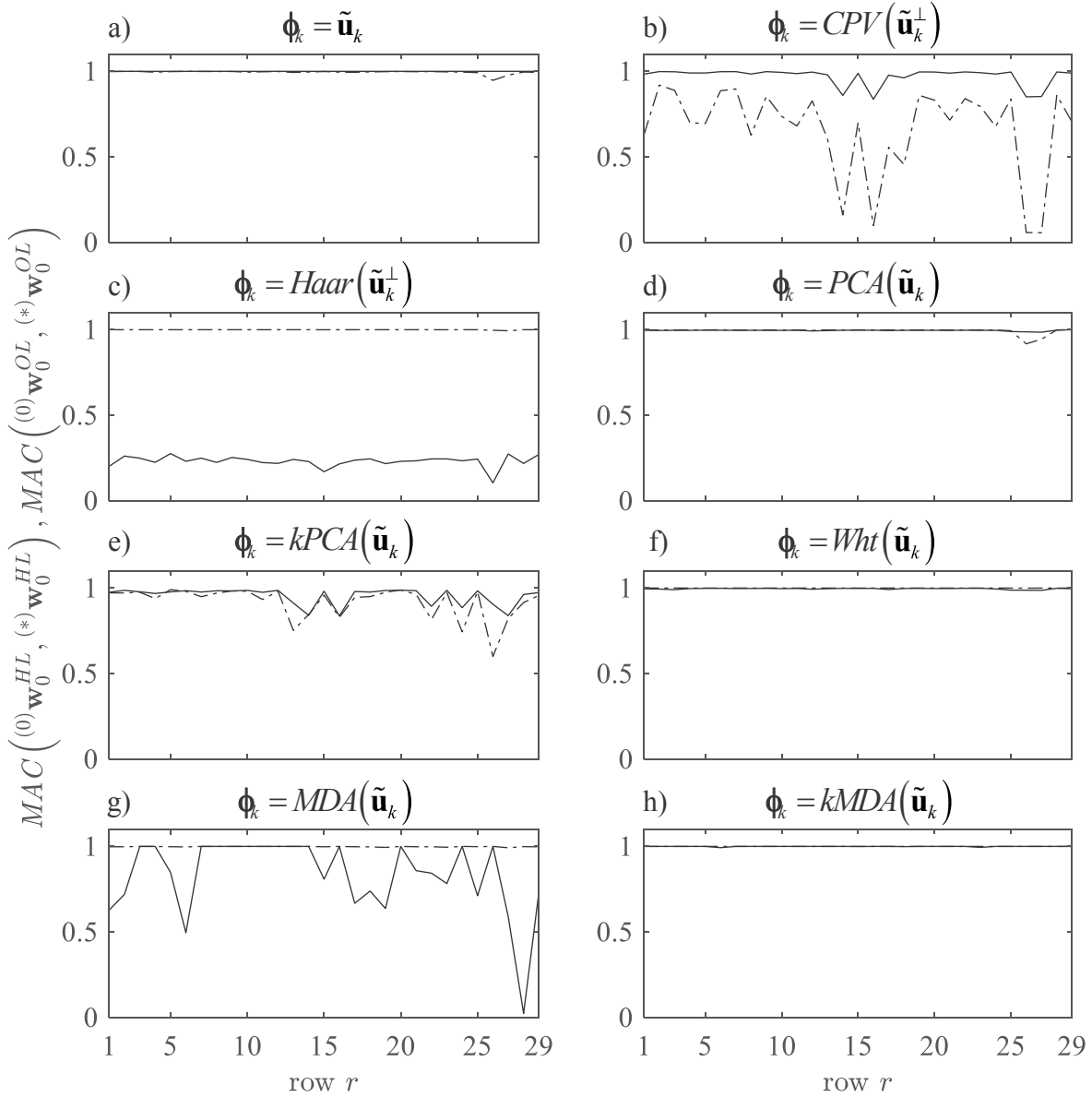


Figure 6.5: MAC between the initial  $\mathbf{w}_0^{(0)}$  and the final  $\mathbf{w}_0^{(*)}$  setting of the bias-weight vector of the hidden layer  $\mathbf{w}_0^{HL}$  (full line) and output layer  $\mathbf{w}_0^{OL}$  (dashed line) for training the BPNN with the set (6.19) for each compressor row  $r$

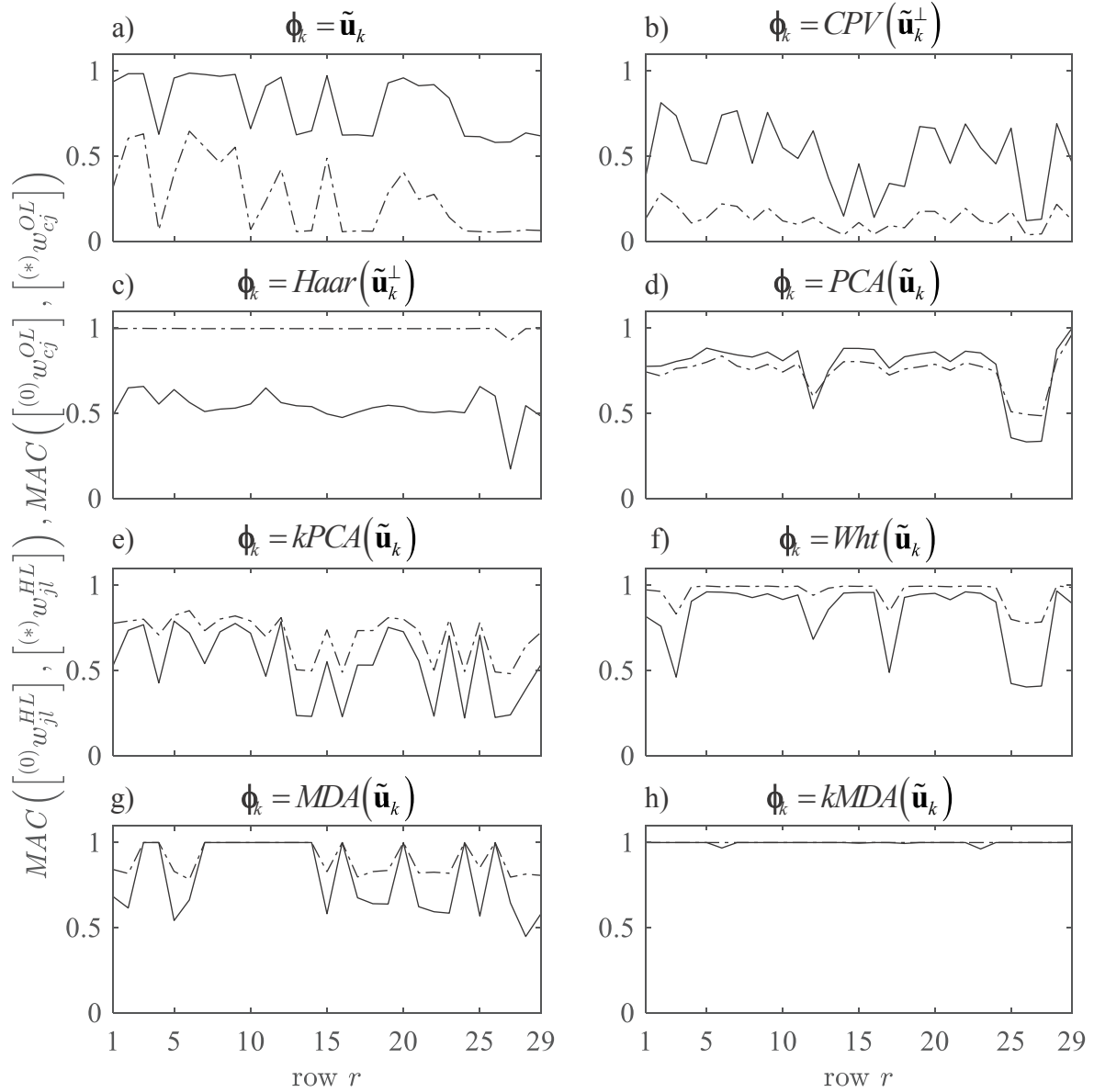


Figure 6.6: MAC between the initial  $(^{(0)})$  and the final  $(^{(*)})$  setting of the hidden-layer  $w_{jl}^{HL}$  (full line) and output-layer weights  $w_{cj}^{OL}$  (dashed line) for training the BPNN with the set (6.19) for each compressor row  $r$

## 6.2 Assessment of Classification Performance

Although it is rather obvious that a dimension-reduction method (Section 5.2) with better effect on the separation and structure of the clusters will also improve performance of a specific classifier, this is not certain, because the performance of a classifier will greatly depend on how well the data structure fits to the classification algorithm and involved assumptions. Therefore, the classification performance of each classifier in Section 6.1 in combination with most of the dimension-reduction methods<sup>1</sup> in Section 5.2 will be assessed through cross-validation on the reference set  $\mathcal{V}$  of eigenmode vectors of known classification, which is split row-wise in subsets  $\mathcal{V}^r$  according to Eq. (5.8). These subsets  $\mathcal{V}^r$  are the test sets for evaluating the classification performance of each classifier. The training of each classifier in Section 6.1, except for the BPNN, will be performed on row-specific training sets  $\mathcal{V}_T^r := \mathcal{V} \setminus \mathcal{V}^r$ . In case of the BPNN classifier the training set and necessary validation set of a specific row  $r$  are defined according to Eqns. (6.18) and (6.19).

By using the Kronecker-delta  $\delta_{ij}$  ( $\delta_{ij} = 1$  for  $i = j$  and  $\delta_{ij} = 0$  for  $i \neq j$ ) and comparing the assigned classes  $\tilde{c}_k$  with the true classes  $c_k$  assigned by a human expert, the number of misclassifications on each compressor row  $r$  can be calculated as

$$N^r = |\mathcal{V}^r| - \sum_{k: \boldsymbol{\phi}_k \in \mathcal{V}^r} \delta_{\tilde{c}_k c_k} . \quad (6.20)$$

The general classification performance can then be assessed by summing up the number of misclassifications of all rows  $r = 1, \dots, N_R$ , i.e.,  $\sum_{r=1}^{N_R} N^r$ . Additionally, the following characteristics may be defined for assessing the percentage of correct classifications:

$$\eta_{avg} = \frac{1}{N_R} \sum_{r=1}^{N_R} \left( 1 - \frac{N^r}{|\mathcal{V}^r|} \right), \eta_{max} = \max_r \left( 1 - \frac{N^r}{|\mathcal{V}^r|} \right), \eta_{min} = \min_r \left( 1 - \frac{N^r}{|\mathcal{V}^r|} \right) . \quad (6.21)$$

The performances of the nearest-neighbor (1NN) and 3-nearest-neighbors classifiers are given in Table 6.1. In alignment with the improved cluster separation evaluated in Sections 5.1 and 5.2, it can be seen that normalization (5.5) and kPCA improve classification performance of 1NN significantly (1 & 2 compared to 3 & 4 and 3–5 compared to 11–13, respectively). Interestingly, the performance of the 1NN-classifier using redirection (5.2) is significantly better than using (5.3) (1, 3, 8, 11 vs. 2, 4, 9, 12, respectively) although, the silhouette values in Fig. 5.2 are only slightly better. Especially applied to kPCA on the whole normalized displacement field ( $\boldsymbol{\phi} = kPCA(*\tilde{\mathbf{u}})$ ) gives the best performance for 1NN. If only the perpendicular displacement field is utilized, i.e.,  $\boldsymbol{\phi} = \tilde{\mathbf{u}}^\perp, *\tilde{\mathbf{u}}^\perp$ , the performance is the same for both redirection methods which is also the case for any application of additional dimension-reduction methods. Note, this is not self-evident, since the vector

<sup>1</sup>Data whitening and MDA will not be considered because they performed exceptionally poor on cluster separation compared to other methods. Furthermore, most of the dimension-reduction methods can be combined for further reduction. Such possibilities are also not documented here due to little or no improvement and the vast number of possibilities which would have made documentation less clear.



Table 6.1: Classification performance of 1NN and 3NN classifiers

Combination	Data $\phi$	Method	Dim.	$\sum_r N^r$	$\eta_{min}$ [%]	$\eta_{avg}$ [%]	$\eta_{max}$ [%]
1	$\mathbf{u}$	1NN	4800	40	55.6	92.4	100
2	$*\mathbf{u}$	1NN	4800	28	66.7	94.6	100
3	$\tilde{\mathbf{u}}$	1NN	4800	32	62.5	93.9	100
4	$*\tilde{\mathbf{u}}$	1NN	4800	23	77.8	95.6	100
5	$\tilde{\mathbf{u}}^\perp, * \tilde{\mathbf{u}}^\perp$	1NN	1600	28	75.0	94.7	100
6	$CPV(\tilde{\mathbf{u}}^\perp)$	1NN	157	30	75.0	94.4	100
7	$Haar(\tilde{\mathbf{u}}^\perp)$	1NN	16	40	55.6	92.4	100
8	$PCA(\tilde{\mathbf{u}})$	1NN	186	32	62.5	93.9	100
9	$PCA(*\tilde{\mathbf{u}})$	1NN	184	24	77.8	95.5	100
10	$PCA(\tilde{\mathbf{u}}^\perp)$	1NN	76	28	75.0	94.7	100
11	$kPCA(\tilde{\mathbf{u}})$	1NN	370	30	62.5	94.3	100
12	$kPCA(*\tilde{\mathbf{u}})$	1NN	368	<b>22</b>	<b>77.8</b>	<b>95.8</b>	100
13	$kPCA(\tilde{\mathbf{u}}^\perp)$	1NN	242	23	77.8	95.7	100
14	$kMDA(\tilde{\mathbf{u}})$	1NN	11	53	44.4	90.0	100
15	$kMDA(*\tilde{\mathbf{u}})$	1NN	11	49	55.6	90.8	100
16	$kMDA(\tilde{\mathbf{u}}^\perp)$	1NN	11	96	44.4	82.2	100
17	$\mathbf{u}$	3NN	4800	35	62.5	93.3	100
18	$*\mathbf{u}$	3NN	4800	34	62.5	93.5	100
19	$\tilde{\mathbf{u}}$	3NN	4800	33	62.5	93.7	100
20	$*\tilde{\mathbf{u}}$	3NN	4800	28	75.0	94.7	100
21	$\tilde{\mathbf{u}}^\perp, * \tilde{\mathbf{u}}^\perp$	3NN	1600	28	75.0	94.7	100
22	$CPV(\tilde{\mathbf{u}}^\perp)$	3NN	157	30	75.0	94.4	100
23	$Haar(\tilde{\mathbf{u}}^\perp)$	3NN	16	46	55.6	91.3	100
24	$PCA(\tilde{\mathbf{u}})$	3NN	186	33	62.5	93.7	100
25	$PCA(*\tilde{\mathbf{u}})$	3NN	184	28	75.0	94.7	100
26	$PCA(\tilde{\mathbf{u}}^\perp)$	3NN	76	28	75.0	94.7	100
27	$kPCA(\tilde{\mathbf{u}})$	3NN	370	30	62.5	94.3	100
28	$kPCA(*\tilde{\mathbf{u}})$	3NN	368	24	75.0	95.5	100
29	$kPCA(\tilde{\mathbf{u}}^\perp)$	3NN	242	23	77.8	95.7	100
30	$kMDA(\tilde{\mathbf{u}})$	3NN	11	53	44.4	90.0	100
31	$kMDA(*\tilde{\mathbf{u}})$	3NN	11	49	55.6	90.8	100
32	$kMDA(\tilde{\mathbf{u}}^\perp)$	3NN	11	96	44.4	82.2	100

norm differs and, therefore, normalization affects the perpendicular displacement field differently. However, here those differences seem to be negligible which was also concluded from the investigations in Chapter 5. Also in agreement with the results from Chapter 5, the performance of kMDA (comb. 14–16) is poor compared to other methods.

Majority voting using kNN with  $k > 1$  decreases the classification performance for any number of nearest neighbors except three. In fact, using 3NN does not improve the performance for any of the dimension-reduction methods, but for some methods performance is just the same. In such cases 3NN should be preferred over 1NN, because it will make the decision boundary more smooth and may give better generalization for some data.

In Section 6.1 it was explained that the  $MAC_k$  classifier is an 1NN classifier with cosine-distance measure which should be, according to the investigation in Appendix B, the natural distance measure for eigenmodes that have not been dimensionally reduced (eigenmodes are distinguished by their amplitude ratios, not the level of the amplitudes). This is confirmed by the performance data in Table 6.2 which states a better performance for  $\hat{\phi} \hat{=}\mathbf{u}, * \mathbf{u}, \tilde{\mathbf{u}}, * \tilde{\mathbf{u}}$  (comb. 1–4) with cosine distance than with Euclidean in Table. 6.1. Interestingly the performance also improves if PCA and kPCA are applied to  $* \tilde{\mathbf{u}}$  (comb. 9&12). This is not surprising in the way that it is known from Fig. 5.12 that, if applied to normalized vector data, PCA and kPCA maintain a data structure that favors the cosine distance. However, the fact that the performance improves for  $* \tilde{\mathbf{u}}$ , but not for  $\tilde{\mathbf{u}}$ , indicates that (5.3) causes better alignment of vectors on the unit hypersphere so that directions of main variances are also better discriminants, which is comprehensible.

The performance of the SVM classifier strongly depends on how the chosen kernel function suits the data as described in Appendix C.4. The choice of the kernel function is in agreement with kPCA; hence, also with SVM the polynomial kernel (C.18) with  $q = 2$  performs best on the given reference set of data (compared to other values for  $q$  or types of kernels, e.g., Gaussian kernel), if data have not been dimensionally reduced with kPCA or kMDA. For these exceptions the linear kernel with  $q = 1$  performs best. The box constraint  $C$  in Eq. (C.80) gives best results when set to 200000 for the polynomial (quadratic) kernel and 1 for the linear kernel. These settings have been deployed in order to calculate the performance of SVM in regards to the different dimension reduction methods in Table 6.3. In comparison to Table 6.1 and 6.2, it shows that SVM, although being the more sophisticated classifier, does not outperform the kNN and MAC in general. The performance is significantly poorer when the data have not been normalized (comb. 1&2) or have been dimensionally reduced using CPV and Haar features (comb. 6&7). However, SVM gives the best performance so far when applied to normalized data redirected by Eq. (5.2) and dimensionally reduced with kPCA (comb. 11).

Finally, the performance of the BPNN classifier is assessed in Table 6.4. As shown before in Table 6.2, using the cluster-correlation measure  $MAC^c$  (comb. 17–32) does not give a competitive classifier. However, the results are sufficient enough to justify the implementation of cluster-correlation into the initial output-layer weights  $w_{cj}^{OL}$  of the BPNN as suggested in Section 6.1. By training the network, this initialization together

Table 6.2: Classification performance of  $MAC_k$  and  $MAC^c$  classifier

Combination	Data $\phi$	Method	Dim.	$\sum_r N^r$	$\eta_{min}$ [%]	$\eta_{avg}$ [%]	$\eta_{max}$ [%]
1	$\mathbf{u}$	$MAC_k$	4800	29	62.5	94.5	100
2	$*\mathbf{u}$	$MAC_k$	4800	22	77.8	95.8	100
3	$\tilde{\mathbf{u}}$	$MAC_k$	4800	29	62.5	94.5	100
4	$*\tilde{\mathbf{u}}$	$MAC_k$	4800	22	77.8	95.8	100
5	$\tilde{\mathbf{u}}^\perp, * \tilde{\mathbf{u}}^\perp$	$MAC_k$	1600	32	75.0	94.0	100
6	$CPV(\tilde{\mathbf{u}}^\perp)$	$MAC_k$	157	36	75.0	93.3	100
7	$Haar(\tilde{\mathbf{u}}^\perp)$	$MAC_k$	16	41	55.6	92.3	100
8	$PCA(\tilde{\mathbf{u}})$	$MAC_k$	186	34	62.5	93.5	100
9	$PCA(*\tilde{\mathbf{u}})$	$MAC_k$	184	25	75.0	95.2	100
10	$PCA(\tilde{\mathbf{u}}^\perp)$	$MAC_k$	76	34	66.7	93.6	100
11	$kPCA(\tilde{\mathbf{u}})$	$MAC_k$	370	31	62.5	94.1	100
12	$kPCA(*\tilde{\mathbf{u}})$	$MAC_k$	368	<b>21</b>	<b>77.8</b>	<b>96.1</b>	100
13	$kPCA(\tilde{\mathbf{u}}^\perp)$	$MAC_k$	242	29	66.7	94.5	100
14	$kMDA(\tilde{\mathbf{u}})$	$MAC_k$	11	59	44.4	88.9	100
15	$kMDA(*\tilde{\mathbf{u}})$	$MAC_k$	11	47	66.7	91.2	100
16	$kMDA(\tilde{\mathbf{u}}^\perp)$	$MAC_k$	11	130	25.0	75.8	100
17	$\mathbf{u}$	$MAC^c$	4800	33	77.8	93.9	100
18	$*\mathbf{u}$	$MAC^c$	4800	34	77.8	93.8	100
19	$\tilde{\mathbf{u}}$	$MAC^c$	4800	36	77.8	93.4	100
20	$*\tilde{\mathbf{u}}$	$MAC^c$	4800	34	77.8	93.7	100
21	$\tilde{\mathbf{u}}^\perp, * \tilde{\mathbf{u}}^\perp$	$MAC^c$	1600	82	55.6	84.8	94.7
22	$CPV(\tilde{\mathbf{u}}^\perp)$	$MAC^c$	157	77	62.5	85.8	95.0
23	$Haar(\tilde{\mathbf{u}}^\perp)$	$MAC^c$	16	84	62.5	84.5	100
24	$PCA(\tilde{\mathbf{u}})$	$MAC^c$	186	37	77.8	93.2	100
25	$PCA(*\tilde{\mathbf{u}})$	$MAC^c$	184	44	75.0	91.9	100
26	$PCA(\tilde{\mathbf{u}}^\perp)$	$MAC^c$	76	70	75.0	87.2	100
27	$kPCA(\tilde{\mathbf{u}})$	$MAC^c$	370	35	66.7	93.5	100
28	$kPCA(*\tilde{\mathbf{u}})$	$MAC^c$	368	37	66.7	93.0	100
29	$kPCA(\tilde{\mathbf{u}}^\perp)$	$MAC^c$	242	44	75.0	91.8	100
30	$kMDA(\tilde{\mathbf{u}})$	$MAC^c$	11	59	44.4	88.9	100
31	$kMDA(*\tilde{\mathbf{u}})$	$MAC^c$	11	47	66.7	91.2	100
32	$kMDA(\tilde{\mathbf{u}}^\perp)$	$MAC^c$	11	132	25.0	75.5	100

Table 6.3: Classification performance of SVM classifier

Combination	Data $\phi$	Method	Dim.	$\sum_r N^r$	$\eta_{min}$ [%]	$\eta_{avg}$ [%]	$\eta_{max}$ [%]
1	$\mathbf{u}$	SVM	4800	49	50.0	90.8	100
2	$*\mathbf{u}$	SVM	4800	37	50.0	93.0	100
3	$\tilde{\mathbf{u}}$	SVM	4800	27	75.0	94.9	100
4	$*\tilde{\mathbf{u}}$	SVM	4800	22	77.8	95.9	100
5	$\tilde{\mathbf{u}}^\perp, *\tilde{\mathbf{u}}^\perp$	SVM	1600	28	66.7	94.8	100
6	$CPV(\tilde{\mathbf{u}}^\perp)$	SVM	157	38	55.6	92.9	100
7	$Haar(\tilde{\mathbf{u}}^\perp)$	SVM	16	109	12.5	79.6	100
8	$PCA(\tilde{\mathbf{u}})$	SVM	186	30	66.7	94.4	100
9	$PCA(*\tilde{\mathbf{u}})$	SVM	184	27	66.7	95.5	100
10	$PCA(\tilde{\mathbf{u}}^\perp)$	SVM	76	28	66.7	94.8	100
11	$kPCA(\tilde{\mathbf{u}})$	SVM	370	<b>20</b>	75.0	<b>96.2</b>	100
12	$kPCA(*\tilde{\mathbf{u}})$	SVM	368	22	72.2	95.9	100
12	$kPCA(\tilde{\mathbf{u}}^\perp)$	SVM	242	31	66.7	94.2	100
14	$kMDA(\tilde{\mathbf{u}})$	SVM	11	61	55.6	88.7	100
15	$kMDA(*\tilde{\mathbf{u}})$	SVM	11	53	55.6	90.0	100
16	$kMDA(\tilde{\mathbf{u}}^\perp)$	SVM	11	179	33.3	67.0	94.1

with PCA-initialization (6.13) should enable the network to surpass the performance of  $MAC^c$ . This, however, is barely the case as can be seen in Table 6.4. This is an issue of the network training getting stuck in a local minimum and of the correlation (6.15) being unscaled unlike (6.6). However, in most cases the suggested initialization enables the neural network to find better local optima than with random initialization for an equal number of hidden-layer neurons  $N_{HL}$  and the best setting out of 10 random initializations per row. For the comparison, the number of hidden-layer neurons  $N_{HL}$  was determined by the PCA-initialization for 99.99% of data variance. Except for  $\phi = kPCA(\tilde{\mathbf{u}})$  (comb. 22) the overall performance of the suggested BPNN is less competitive compared to the other classifiers. In this case, Fig. 6.6 and Fig. 6.5 show good agreement between initial and final state of the network, but with notable modification such that training leads to successful exploration of a proper optimum. This is in contrast to most of the other input-data configurations in Table 6.4, where the network gets stuck in a local minimum and no noticeable changes are made to the weights resulting in a poor optimum of the network weights. The alternative initialization with kPCA instead of PCA generally improves the performance except for  $\phi = kPCA(\tilde{\mathbf{u}})$ . Because the best performing kPCA-initialized BPNN does not perform better than combination 22 in Table 6.4, results are not shown explicitly. Also not shown are the results of using MDA projection as input for the neural network, because results were extremely poor. The reason might be that input-layer size of  $D = N_C - 1$ , hidden-layer size of  $N_{HL} \leq N_C - 1$ , and output-layer size of  $N_{OL} = N_C$  is a bad setting for the network with too few free parameters (synapses).

Table 6.4: Classification performance of BPNN classifier

Comb.	Data $\phi$	Method	Dim.	$\sum_r N^r$	$\eta_{min}$ [%]	$\eta_{avg}$ [%]	$\eta_{max}$ [%]
1	$\mathbf{u}$	$N_{HL} = 186$ , rand. init.	4800	49	50.0	90.7	100
2	$\mathbf{u}$	$N_{HL} = 186$ , init. with PCA	4800	48	55.6	90.9	100
3	$^*\mathbf{u}$	$N_{HL} = 184$ , rand. init.	4800	53	44.4	90.0	100
4	$^*\mathbf{u}$	$N_{HL} = 184$ , init. with PCA	4800	84	44.4	84.4	100
5	$\tilde{\mathbf{u}}$	$N_{HL} = 186$ , rand. init.	4800	53	44.4	90.0	100
6	$\tilde{\mathbf{u}}$	$N_{HL} = 186$ , init. with PCA	4800	45	66.7	91.6	100
7	$^*\tilde{\mathbf{u}}$	$N_{HL} = 184$ , rand. init.	4800	55	44.4	89.6	100
8	$^*\tilde{\mathbf{u}}$	$N_{HL} = 184$ , init. with PCA	4800	35	66.7	93.4	100
9	$\tilde{\mathbf{u}}^\perp, ^*\tilde{\mathbf{u}}^\perp$	$N_{HL} = 76$ , rand. init.	1600	46	55.6	91.3	100
10	$\tilde{\mathbf{u}}^\perp, ^*\tilde{\mathbf{u}}^\perp$	$N_{HL} = 76$ , init. with PCA	1600	35	62.5	93.4	100
11	$CPV(\tilde{\mathbf{u}}^\perp)$	$N_{HL} = 35$ , rand. init.	157	31	75.0	94.2	100
12	$CPV(\tilde{\mathbf{u}}^\perp)$	$N_{HL} = 35$ , init. with PCA	157	29	66.7	94.6	100
13	$Haar(\tilde{\mathbf{u}}^\perp)$	$N_{HL} = 14$ , rand. init.	16	53	44.4	90.0	100
14	$Haar(\tilde{\mathbf{u}}^\perp)$	$N_{HL} = 14$ , init. with PCA	16	69	44.4	87.0	100
15	$PCA(\tilde{\mathbf{u}})$	$N_{HL} = 145$ , rand. init.	186	101	55.6	81.5	100
16	$PCA(\tilde{\mathbf{u}})$	$N_{HL} = 145$ , init. with PCA	186	67	62.5	87.5	95.0
17	$PCA(^*\tilde{\mathbf{u}})$	$N_{HL} = 144$ , rand. init.	184	89	33.3	83.5	95.0
18	$PCA(^*\tilde{\mathbf{u}})$	$N_{HL} = 144$ , init. with PCA	184	62	44.4	88.3	100
19	$PCA(\tilde{\mathbf{u}}^\perp)$	$N_{HL} = 50$ , rand. init.	76	90	44.4	83.4	100
20	$PCA(\tilde{\mathbf{u}}^\perp)$	$N_{HL} = 50$ , init. with PCA	76	65	55.6	87.9	100
21	$kPCA(\tilde{\mathbf{u}})$	$N_{HL} = 36$ , rand. init.	370	30	50.0	92.4	100
22	$kPCA(\tilde{\mathbf{u}})$	$N_{HL} = 36$ , init. with PCA	370	<b>23</b>	<b>77.8</b>	<b>95.8</b>	100
23	$kPCA(^*\tilde{\mathbf{u}})$	$N_{HL} = 36$ , rand. init.	368	29	66.7	94.5	100
24	$kPCA(^*\tilde{\mathbf{u}})$	$N_{HL} = 36$ , init. with PCA	368	25	75.0	95.3	100
25	$kPCA(\tilde{\mathbf{u}}^\perp)$	$N_{HL} = 21$ , rand. init.	242	31	62.5	94.1	100
26	$kPCA(\tilde{\mathbf{u}}^\perp)$	$N_{HL} = 21$ , init. with PCA	242	35	62.5	93.4	100

In conclusion, the suggested BPNN rarely outperforms the other classifiers, and w.r.t. the best result of each of the investigated classifiers, BPNN only outperforms  $MAC^c$ . However, the suggested initialization enables to find promising local optima faster than random initialization and further potential may be exploited if a more complex network design is set up.

After the comparison of the classifier performances, the SVM based on  $\phi = kPCA(\tilde{\mathbf{u}})$  seems to be the best choice in regards to the overall performance. However, w.r.t. the airfoil design problem described in Section 2.3, misclassification of eigenmode shapes that are more likely to cause failure is more critical than misclassifying less critical modes. In

regards to the fixation study in Section 3.4 it should also be considered that eigenmode shapes with a wider frequency range due to higher fixation uncertainties (frequency veering) should be given more concern w.r.t. misclassification. Therefore, an additional detailed study on the confusion, defined by type and number of misclassifications between a human expert and the best strategies of each of the assessed classifiers is given in Fig. 6.7. First, it is apparent that SVM and BPNN show some confusion between second-bending (2B) and first-torsion modes (1T) w.r.t. human expert judgment. This issue is not critical, since these misclassifications occur at front rotor rows equipped with dovetail roots, which as explained in Section 3.2 suffer only little installation uncertainties due to minor effects by changing temperature and centrifugal loading. Therefore, assigning the correct eigenfrequency bands (correlating eigenmode shapes) is no problem. In this case it is also not problematic that an incorrect penalty might be assigned, because both mode shapes usually share similar penalty factors  $w_i^E$  in Eq. (2.15). The matter is quite different w.r.t the first stiff-wise mode (1S), which is considered to be critical in regards to failure and is also prone to frequency veering, where misclassification may result in underestimation of the width of the eigenfrequency. From this point of view it might be worthwhile to prefer the BPNN classifier over the others due to its perfect detection rate of 1S modes

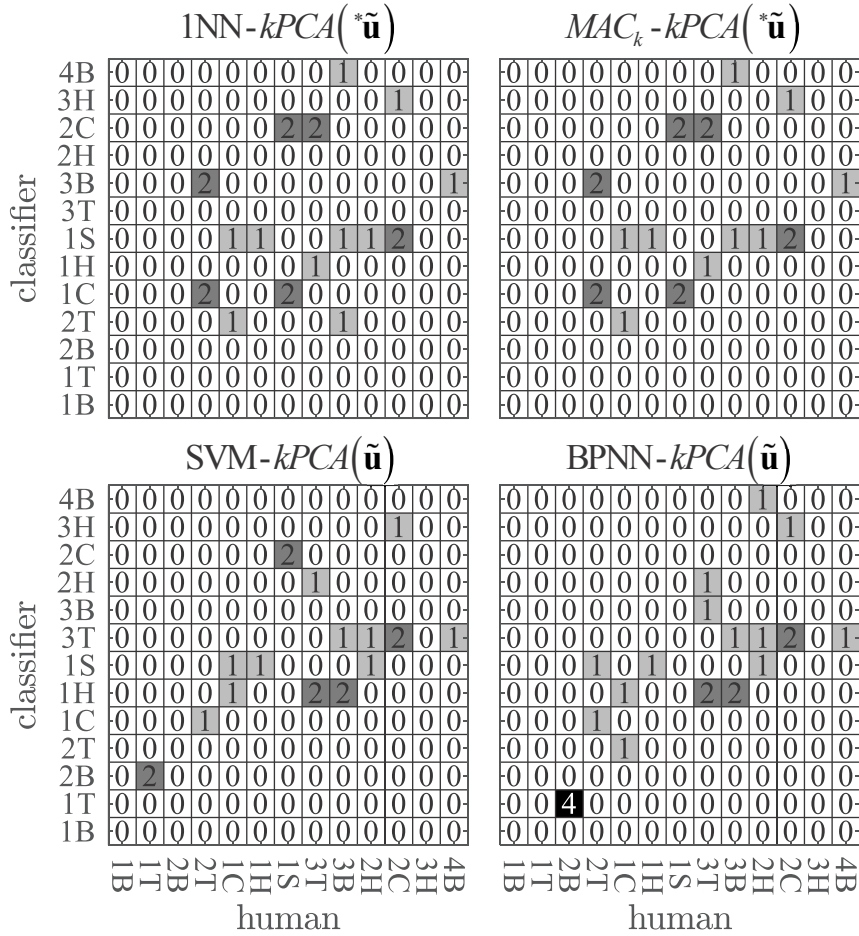


Figure 6.7: Confusion of best performing 1NN,  $MAC_k$ , SVM, and BPNN classifier w.r.t. the judgment of a human expert

and negligible performance drawback compared to SVM. Hence, the BPNN classifier will be used further on.

## 6.3 Procedure for Assigning Eigenmode Bands

After having elaborated the most suitable classification method within the previous section, assigning the shape specific eigenfrequency bands required for the calculation of the *MRF* value (see Section 2.3) seems straight forward. However, pairing the correct eigenfrequencies between their fixation and operation extrema becomes challenging when dealing with misclassification, missing pairing options, and unknown eigenmode shapes. Therefore, the elaborated classification method from Section 6.2 is embedded into a heuristic decision procedure which is developed based on the knowledge gained from the fixation-variation studies in Section 3.4.

The first step is the calculation of eigenmodes for a new airfoil design at both the lower and upper prediction limit of the eigenfrequencies. As described in Section 3.2.2, the upper limit for a hammer-foot root is obtained with tight root fixation at idle engine temperature, whereas the lower limit is obtained with loose root fixation at nominal engine temperature. These eigenmodes are projected onto standard rectangular squares as described in Section 4, formed to eigenvectors (5.1), redirected by Eq. (5.2), normalized by Eq. (5.5), reduced according to Eq. (C.16), and then split into one set  $\mathcal{V}_L$  of eigenvectors at loose-nominal and one set  $\mathcal{V}_U$  at tight-idle fixation conditions. In order to ensure that each of these two sets contains all eigenmode shapes of interest, the number of modes per set has to be larger than the number of actually relevant modes. Based on service-experience and research on industrial gas-turbines, 11 eigenmode shapes have been identified as being relevant, which are the first 11 modes in Fig. 2.6. For the test compressor used to build the eigenmode-reference set  $\mathcal{V}$ , 15 eigenmodes per fixation type were calculated. The result is two sets of mode vectors  $\mathcal{V}_L = \{\boldsymbol{\phi}_1^L, \dots, \boldsymbol{\phi}_{15}^L\}$  and  $\mathcal{V}_U = \{\boldsymbol{\phi}_1^U, \dots, \boldsymbol{\phi}_{15}^U\}$  which have to be classified by the BPNN (Section 6.1) to assign classes  $c_1^L, \dots, c_{15}^L$  and  $c_1^U, \dots, c_{15}^U$ , respectively.

Since a classifier will assign only classes it has been trained for, it is clear that eigenmodes not contained in the training set  $\mathcal{V}_T$  will be falsely assigned to any class contained in  $\mathcal{V}_T$ . This issue becomes critical due to the circumstance that  $\mathcal{V}_L$  and  $\mathcal{V}_U$  contain more eigenmodes than there are different eigenmode classes within  $\mathcal{V}_T$ . To relax this problem, two more modes 3H and 4B (see Fig. 2.6) are added to the set of modes to be detected, because they were found in the reference set among the first 15 modes at numerous rows. Additionally, eigenvectors of a new airfoil design  $\tilde{\boldsymbol{\phi}} \in \mathcal{V}_L \cup \mathcal{V}_U =: \mathcal{V}_C$  which do not correlate sufficiently with any of the eigenvectors in  $\mathcal{V}_T$  are excluded from classification. Sufficient correlation is tested by evaluating  $MAC_k$  in Eq. (6.5) between each  $\tilde{\boldsymbol{\phi}} \in \mathcal{V}_C$  and  $\boldsymbol{\phi}_k \in \mathcal{V}_T$ . In case of  $MAC_k < 0.4 \forall k = 1, \dots, |\mathcal{V}_T|$ , the eigenvector  $\tilde{\boldsymbol{\phi}}$  is assumed to be insufficiently represented by any element in  $\mathcal{V}_T$  and, therefore, is deleted from  $\mathcal{V}_C$ , i.e.,  $\mathcal{V}_C := \mathcal{V}_C \setminus \{\tilde{\boldsymbol{\phi}}\}$ . Accordingly new reduced sets of eigenmodes at the upper and lower frequency limit are defined as  $\mathcal{V}'_L := \mathcal{V}_L \cap \mathcal{V}_C$  and  $\mathcal{V}'_U := \mathcal{V}_U \cap \mathcal{V}_C$ .

So far only the issue of eigenmodes associated with an unknown class w.r.t. the

reference set  $\mathcal{V}$  has been addressed. A possible solution to the other two issues, namely misclassification and missing pairing options, is embedded in the following procedure of assigning eigenmode shapes:

1. After the sets  $\mathcal{V}'_L$  and  $\mathcal{V}'_U$  have been calculated as described above and the eigenmodes  $\boldsymbol{\phi}_i^L$  and  $\boldsymbol{\phi}_j^U$  have been sorted in the order of the corresponding eigenfrequencies, respectively, the initial step of allocating eigenfrequency bands is to pair most unambiguous modes first. This reduces the risk of confusion caused by multiple detections of the same mode shape due to frequency veering or misclassification. Based on the MAC-values (C.54)

$$MAC_{ij}^{L-U} := MAC(\boldsymbol{\phi}_i^L, \boldsymbol{\phi}_j^U) \text{ where } \boldsymbol{\phi}_i^L \in \mathcal{V}'_L, \boldsymbol{\phi}_j^U \in \mathcal{V}'_U, \quad (6.22)$$

correlations between the elements of the sets  $\mathcal{V}'_L$  and  $\mathcal{V}'_U$  are used to pair each  $\boldsymbol{\phi}_i^L \in \mathcal{V}'_L$  with that  $\boldsymbol{\phi}_j^U \in \mathcal{V}'_U$  which has the same class and sufficiently high MAC-value, resulting in a set  $\{\boldsymbol{\phi}_j^U \in \mathcal{V}'_U | c_j^U = c_i^L \wedge MAC_{ij}^{L-U} \geq 0.6\}$ . If several possibilities exist, the one with the lowest eigenfrequency is chosen. After a pair has been identified, the paired eigenmodes are deleted from  $\mathcal{V}'_L$  and  $\mathcal{V}'_U$ :

$$\mathcal{V}'_L := \mathcal{V}'_L \setminus \{\boldsymbol{\phi}_i^L\} \text{ and } \mathcal{V}'_U := \mathcal{V}'_U \setminus \{\boldsymbol{\phi}_j^U\}, \quad (6.23)$$

before the next pair is going to be identified.

2. After, each remaining  $\boldsymbol{\phi}_i^L \in \mathcal{V}'_L$  is paired with that  $\boldsymbol{\phi}_j^U \in \{\boldsymbol{\phi}_j^U \in \mathcal{V}'_U | c_j^U = c_i^L\}$  which has the lowest eigenfrequency. However, the pairing shall start from the lowest eigenfrequency and the paired modes are deleted according to Eq. (6.23) before the next pairing is established. Note, the difference between this step and the previous one is that here only classification is used without the combination with the MAC-value. Hence, first the most ambiguous modes are paired and afterwards the less ambiguous ones with lower MAC-values.
3. Next, each  $\boldsymbol{\phi}_i^L \in \mathcal{V}'_L$  is paired with that  $\boldsymbol{\phi}_j^U \in \{\boldsymbol{\phi}_j^U \in \mathcal{V}'_U | MAC_{ij}^{L-U} \geq 0.6\}$  which has the lowest eigenfrequency, because these modes are rather similar, w.r.t. MAC, thus, have most likely been misclassified by the classifier. Each resulting mode band is assigned with that class of the two partners which corresponds to the higher risk of failure. Both partners are also deleted according to (6.23) before the next pairing is identified.
4. If there are  $\boldsymbol{\phi}_i^L \in \mathcal{V}'_L$  ( $\boldsymbol{\phi}_j^U \in \mathcal{V}'_U$ ) left that share the same class  $c_i^L = c_{i\pm 1}^L$  ( $c_j^U = c_{j\pm 1}^U$ ) as a neighbor  $\boldsymbol{\phi}_{i\pm 1}^L$  ( $\boldsymbol{\phi}_{j\pm 1}^U$ ) which has been paired already, i.e.,  $\boldsymbol{\phi}_{i\pm 1}^L \notin \mathcal{V}'_L$  ( $\boldsymbol{\phi}_{j\pm 1}^U \notin \mathcal{V}'_U$ ), then this mode is assigned to the same mode band of this neighbor, because that phenomenon may occur due to frequency veering as described in Section 3.4. These modes are then also deleted using Eq. (6.23).
5. For all remaining  $\boldsymbol{\phi}_i^L \in \mathcal{V}'_L$  remaining  $\boldsymbol{\phi}_j^U \in \mathcal{V}'_U$  (no partner was found within the previous steps) the decision is made according to the conclusions a-d in Section 3.4.



It should be noted that step 1 to 3 are different in the way that first both correlation and classification are used to make a decision, then only classification, and finally only correlation. As each step possibly reduces the number of elements in  $\mathcal{V}'_L$  and  $\mathcal{V}'_U$ , the assignment of eigenmode bands starts with the most ambiguous pairs and moves on to the less ambiguous ones step by step.

The pairing decisions made by the procedure for three exemplary rotors are shown in Fig. 6.8, where the number at links denotes the decision step within the procedure above. Geometrically these three examples coincide with the ones in Section 3.4, but here the temperature influence is taken into account as it is done during an actual design evaluation. It can be seen that wherever no pairing can be established based on classification, the decision strategy evaluated based on Section 3.4 is able to assign reasonable frequency bands. In Fig. 6.8a, mode 3B at tight fixation (nominal temperature conditions) is misclassified as 1H which causes assignment of wider and intersecting frequency bands. However, this is assumed to be acceptable, because in case of resonance with such bands the optimization process should drive design changes to tighter frequency bands reducing uncertainties.

In the three examples not all the decision steps of the proposed procedure are required. However, evaluation of all assignments for all compressor rows results in the statistics shown in Table 6.5 with the percentage of each decision rule being applied, where most assignments (81.12%) are made by the classifier in step 1 and 2.

Table 6.5: Execution statistics for the steps of the assignment procedure

step	1	2	3	4	5a	5b	5c	5d
exec. [%]	46,55	34,57	3,55	3,39	0,39	7,28	1,48	2,78

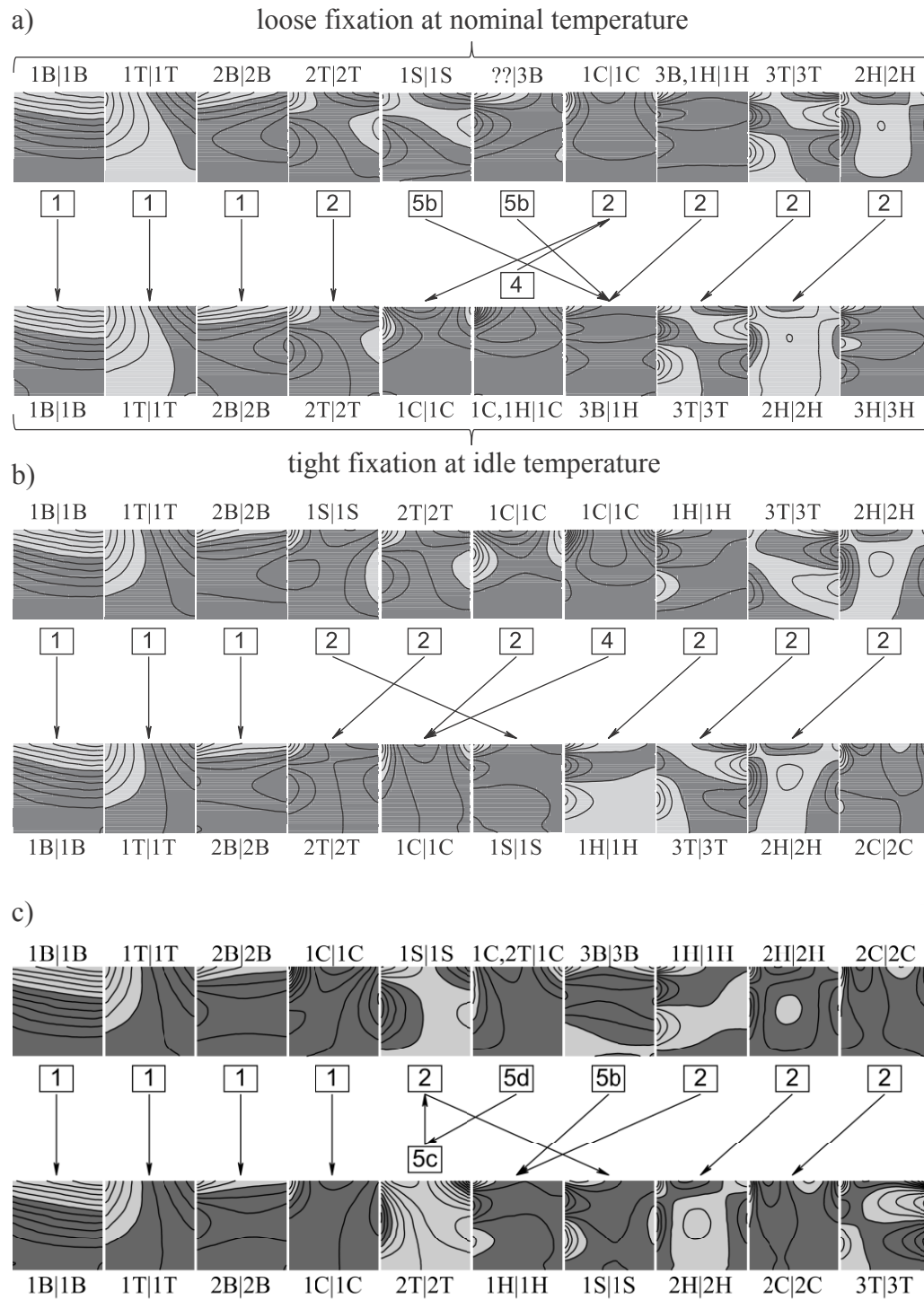


Figure 6.8: Assignment of eigenmode shapes for a) front rotor blade of stage 5, b) middle rotor blade of stage 8, and c) rear rotor blade of stage 15; numbers show the corresponding decision step; human-expert classification (left of |) is compared to the procedure result (right of |)

# 7 Structural Optimization Results for a Rotor Blade

The primary scope of this thesis is to improve automated airfoil optimization. The introduced methods for mode shape specific tuning of eigenfrequency bands, direct hot-to-cold geometry transformation, and assessment of production tolerances in the context of design robustness shall now be tested to prove their capability within an automated design-optimization process described in Chapter 2. Due to limited computational resources, the pre-blading process in Fig. 2.1 will be skipped and the blading process is applied to only a single compressor row. However, this compressor row, which is taken from an industrial test-compressor, is chosen to be rather challenging by changing the material to a cheaper one with lower elastic modulus. This causes the reference design to violate all structural constraints in Section 2.3 except for the maximal overall stress  $\sigma_{max}$  in Eq. (2.18). As the optimization process pushes design changes towards a reduction of constraint violation (2.9), it simultaneously seeks to improve the aerodynamic efficiency (see Eq. (2.10)).

In the following, first it will be validated that the parameter model in Section 2.2 is able to sufficiently describe the reference geometry. Then the process will be applied to the deterministic optimization problem (2.10), and it will be investigated how the two different weight functions (2.13) and (2.14) for calculating the mode-resonance-factor criterion (2.15) affect the optimal design solution. Finally, the process will be applied to the probabilistic optimization problem (2.23) which includes assessment of production tolerances.

## 7.1 Parameter Fitting of an Actual Design

For two reasons it is desirable that the parameter model in Section 2.2 is able to represent a given reference design sufficiently accurately. First, the reference design for the optimization test case in this section originates from an actual commercial compressor design which underwent a manual version of the design process presented in Chapter 2. For the matter of comparability it is therefore important to ensure that the reference design is located in the design space of the parameter model. Second, the setting of the parameter bounds depends on the fitting of the optimization parameters and adverse accuracy would cause a search for optimal designs within an unindented design region. Therefore, it must be tested whether the setting of the parameter model and especially the degree of the splines is suited to allow the parameter model to find a sufficient representation of the

reference design.

The agreement between the one-dimensional optimization parameters and their corresponding reference distribution, i.e.,  $\Delta(\bullet) := (\bullet) - (\bullet)_{ref}$ , is enforced by optimization and results are shown in Fig. 7.1a. While parameters such as the leading-edge radius  $R_L$ , trailing-edge radius  $R_T$ , and theta shift  $\theta_S$  (lean) have been fitted almost perfectly, other parameters, namely leading-edge metal angle  $\beta_L$ , trailing-edge metal angle  $\beta_T$ , maximal thickness  $t^{max}$ , and axial shift  $\tilde{x}_S$  (sweep) show visible but acceptable deviations from the reference distributions. The two-dimensional distribution of the dimensionless chamber-line angle  $\tilde{\beta}$  in Fig. 7.1b and thickness  $\tilde{t}$  in Fig. 7.1c deviate to larger degrees in areas with steep gradients. The global effect onto the deviation of the actual airfoil surface, however, is small, see Fig. 7.2a–b. While the stress distribution (not shown) is not noticeably affected, the eigenfrequency bands show differences which, however, are insignificant. Therefore, the parameter model may be considered as sufficient to describe the reference geometry.

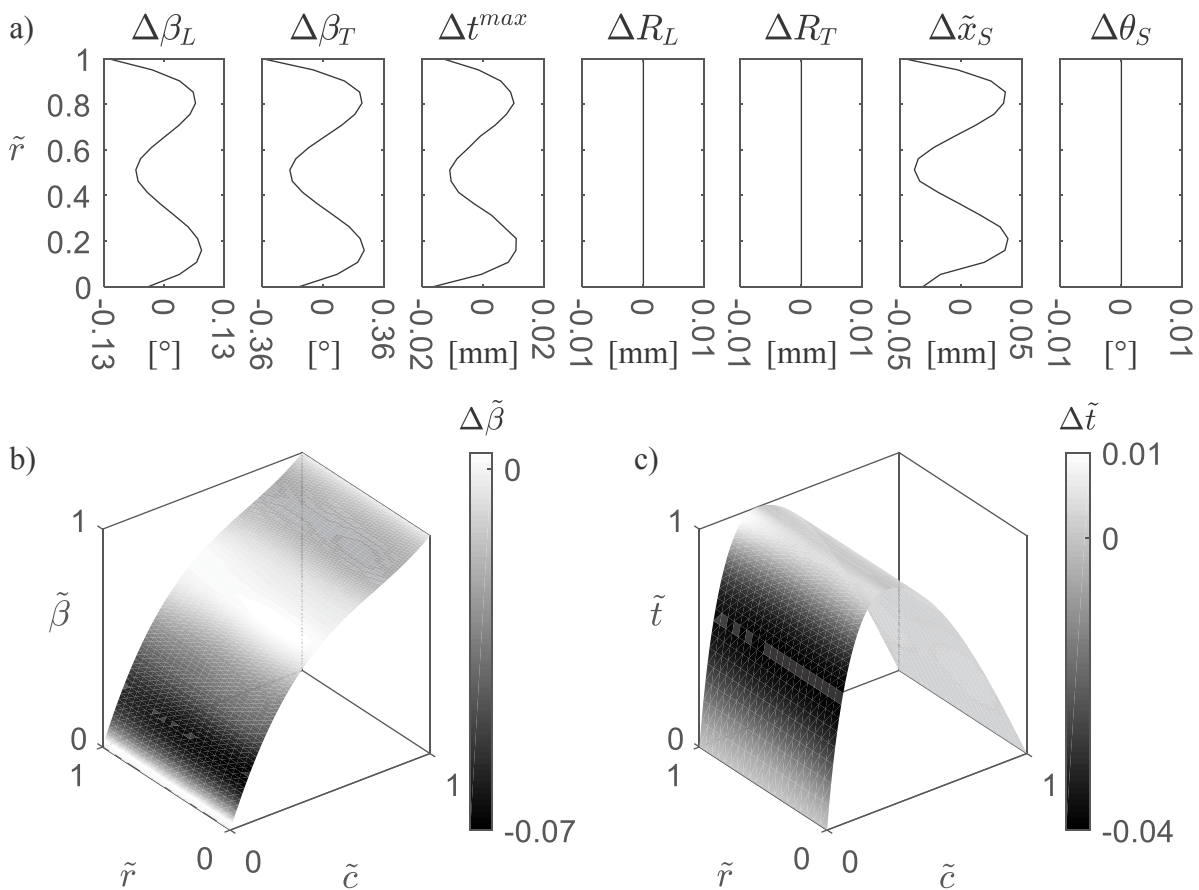


Figure 7.1: Difference fitted parameter distributions and the reference design for a) one dimensional distributions, b) normalized chamber-line-angle distribution, c) and normalized thickness distribution

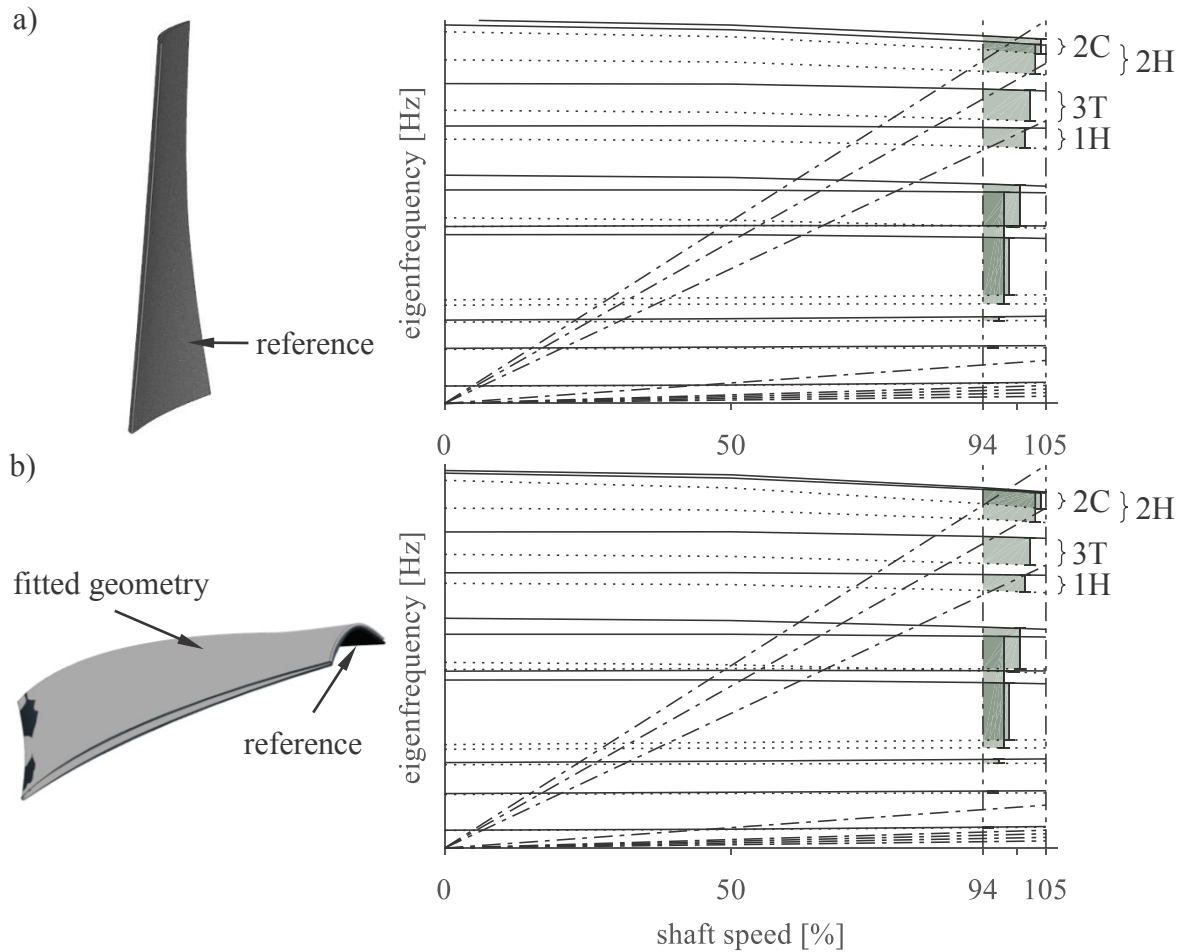


Figure 7.2: Geometry and Campbell diagram with engine orders and eigenmode-shape bands of a) the reference design and b) the design resulting from the fitting of the parameter model

## 7.2 Results of Deterministic Optimization

Besides showing that the introduced methods and criteria are capable of improving the airfoil design when implemented in an automatic design optimization process, it shall also be investigated how the type of weight function  $d(n)$  used for calculating the mode-resonance factor (2.15) influences the optimization results. Therefore, a DoE (design of experiments) of 3000 samples is created within a specified design range about the fitted parameter setting of the industrial compressor blade in Fig. 7.2. Based on this DoE, an optimization is performed which utilizes the normally distributed weight function (2.13). This optimization takes about 1500 iterations until no noteworthy improvements are achieved by the optimizer anymore. Then, the DoE and the additional designs from this first optimization are reassessed using the constant weight function (2.14) and a second optimization is started based on the combined set of samples (about 4500) which continues for about 500 iterations until again no noteworthy improvements are achieved by the optimizer.

As described by Bestle (1994) for multi-objective optimization problems such as Eq. (2.10), there will be no single optimal solution, but a front of Pareto-optimal designs. On this Pareto-front it is not possible to improve one objective without deteriorating at least one other. For further evaluation of the quality of the optimization process, one distinctive design is chosen from the Pareto-front. This design has the lowest constraint penalty factor (2.9) (no design with  $C = 0$  was found), because ensuring aerodynamic feasibility and structural integrity has priority. The selected results from both optimizations with the different weight functions have reduced aerodynamic losses (see Eq. (2.10)) compared to the reference design and significantly better structural integrity than that of the reference design made from the cheaper material.

The Campbell diagram of the mode shape specific eigenfrequency bands are shown in Fig. 7.3 along with an overlay of the reference and optimal geometry. Clearly, the different weight functions cause rather different optimal geometries compared to each other and the reference design. In comparison to the reference design, the usage of constant weight

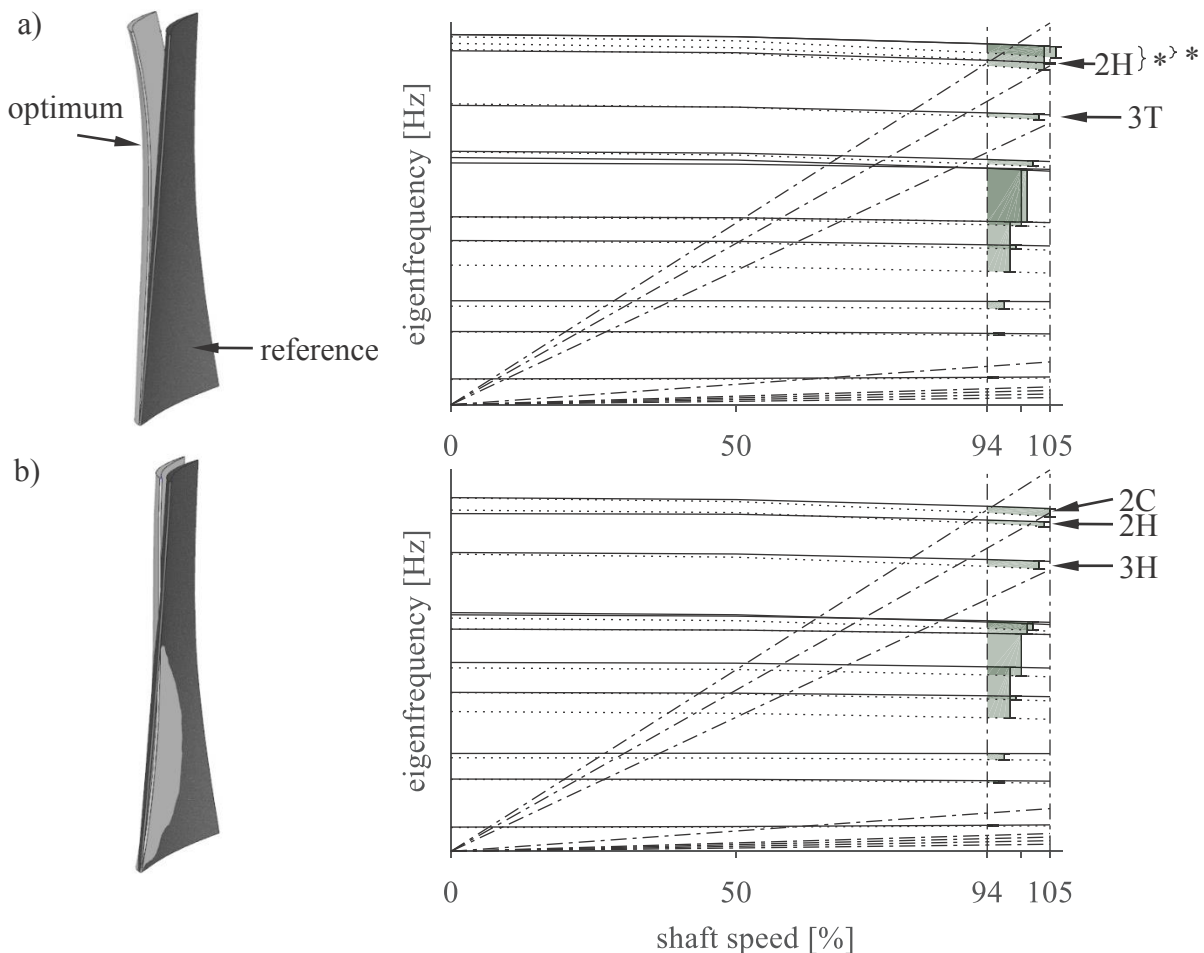


Figure 7.3: Geometry and Campbell diagram with eigenmode-shape bands of optimum obtained with a) constant weight function (2.14) and b) normal distribution (2.13) where \* marks unknown eigenmode shapes not contained by the set of relevant mode shapes in Fig. 2.6

function (2.14) in Fig. 7.3a leads to a better structural-dynamical design. Unlike the reference design which has intersections of the 2H and 2C eigenmode bands with the last two engine orders, see Fig. 7.2, the optimal design only has one intersection of relevance, namely the last engine order (lower penalty, i.e. risk of failure, according to Table 2.2) with 2H mode. The intersections with unknown eigenmodes (marked with \*) are not considered to be HCF critical. Because the process was also able to tighten the 2H mode band to a thickness of only a few Hertz, the speed range of intersection, hence, the probability of excitation has been reduced significantly. Throughout the optimization, the process explored several designs with resonance between lower EO's and eigenmode bands, but successfully avoided those during the later convergence due to the higher penalties which cause high MRF values according to Eq. (2.15).

The usage of normally distributed weight function (2.13) leads the optimization process to a structural-dynamical better design w.r.t. the reference design, but not w.r.t. the constant weight function. Although the intersections with 1H and 3T modes in Fig. 7.2 have been removed, see Fig. 7.3b, the last two engine orders still intersect with 2H and 3H eigenmode bands. The ranges of these intersections, however, have reduced which is beneficial as described above. Considering that the optimization with the constant weight function was carried out on all the designs generated during optimization with the normally distributed weight function (after it was converged), and that the optimum found is different although the design in Fig. 7.3a should be favored by the normally distributed weight function as well, it may be concluded that the constant weight function leads to a less complex response surface of the MRF criterion and thus to better designs.

It has to be acknowledged that, equivalently to the optimization using the constant weight function, all other structural and aerodynamic constraints that are violated by the reference design are passed by the optimal design which has been received using the normally distributed weight function. For example, the arrows in Fig. 7.4a point at regions where the maximum allowed leading and trailing edge stresses are exceeded by the reference design, but not by the optimal designs in Fig. 7.4b–c. The surface stresses of the optimal design on the suction side (SS) in Fig. 7.4b (constant weight function) are the highest as a result of the strong lean of the design causing increased but permissible bending stresses, see Fig. 7.3b.

After the effect of the different weight functions onto structural properties has been analyzed, a more detailed view on the changes of optimization parameters and resulting geometry parameter distributions shall be given. For the matter of simplicity, the optimal solution using constant weight function (2.14) will be referred to as design A, and the other one using normally distributed weight function (2.13) as design B. The comparison of the one dimensional geometry-parameter distributions in Fig. 7.5a for design A and Fig. 7.6a for design B shows that for both designs the thickness  $t^{max}$  is increased towards the hub compared to the reference design. This raises eigenfrequencies of lower eigenmodes while eigenfrequencies of some higher-order modes are even decreased. Since the cord length was not modified in order to maintain the stage splitting defined by the Meanline-Throughflow process, the thicker blades will experience flow separation further upstream. However, flow turning is one of the aerodynamic constraints, hence, the metal angles  $\beta_L$  is increased

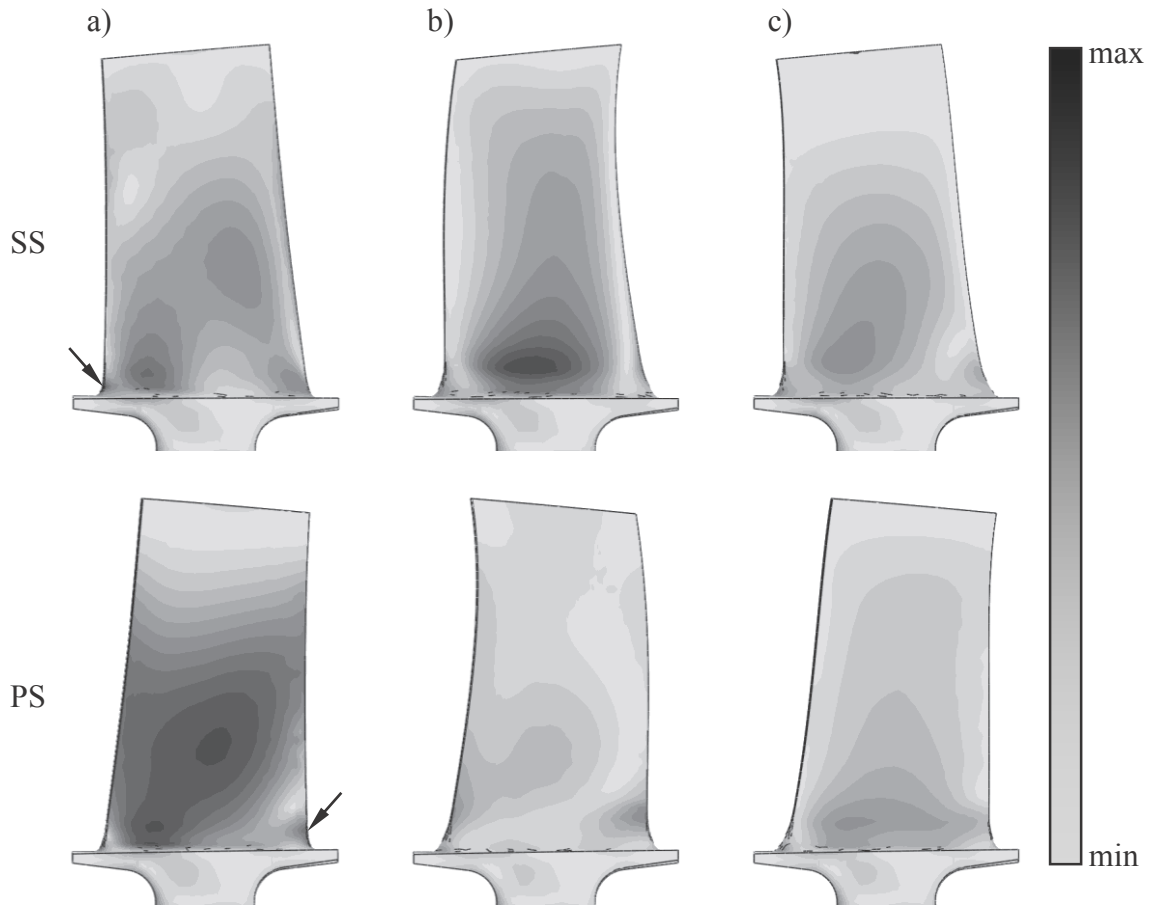


Figure 7.4: Von Mises-stress distribution of a) reference design, b) optimum with constant weight function (2.14), and c) with normal distribution (2.13)

in order to maintain the required turning defined by the Meanline-Throughflow process. The fact that  $\beta_T$  is increased as well, which reduces airfoil curvature as per Fig. 2.2, indicates that this is a countermeasure for flow separation in order to reduce pressure wake losses. The reason that  $\Delta\beta_T$  of design B is larger than that of design A, may be the different direction of sweep  $\tilde{x}_s$  (axial shift). Interestingly the changes of the leading- and trailing-edge radii  $R_L$  and  $R_T$  are negligible despite the need to fulfill the leading- and trailing-edge-stress constraints. The reason is that increased airfoil thickness has a larger influence on the leading- and trailing-edge stresses than the corresponding radii.

Both designs are able to slightly improve aerodynamic efficiency of the reference design which may be founded by the application of lean and sweep as well as by changes to the two-dimensional distributions of the dimensionless chamber-line angle  $\tilde{\beta}$  and thickness  $\tilde{t}$ . For both designs A and B,  $\tilde{\beta}$  is reduced between front and middle as well as middle and rear of the airfoil, whereas it is about similar to the reference design at the middle, see Fig. 7.5b and Fig. 7.6b. Furthermore,  $\tilde{t}$  is mainly reduced in the front region of the airfoil designs and increased towards the trailing edge for design A while being nearly unchanged w.r.t. the reference for design B, see Fig. 7.5c and Fig. 7.6c. The position of the maximum  $\tilde{t}$  is hardly shifted for both designs which means that the position of the shock front hardly



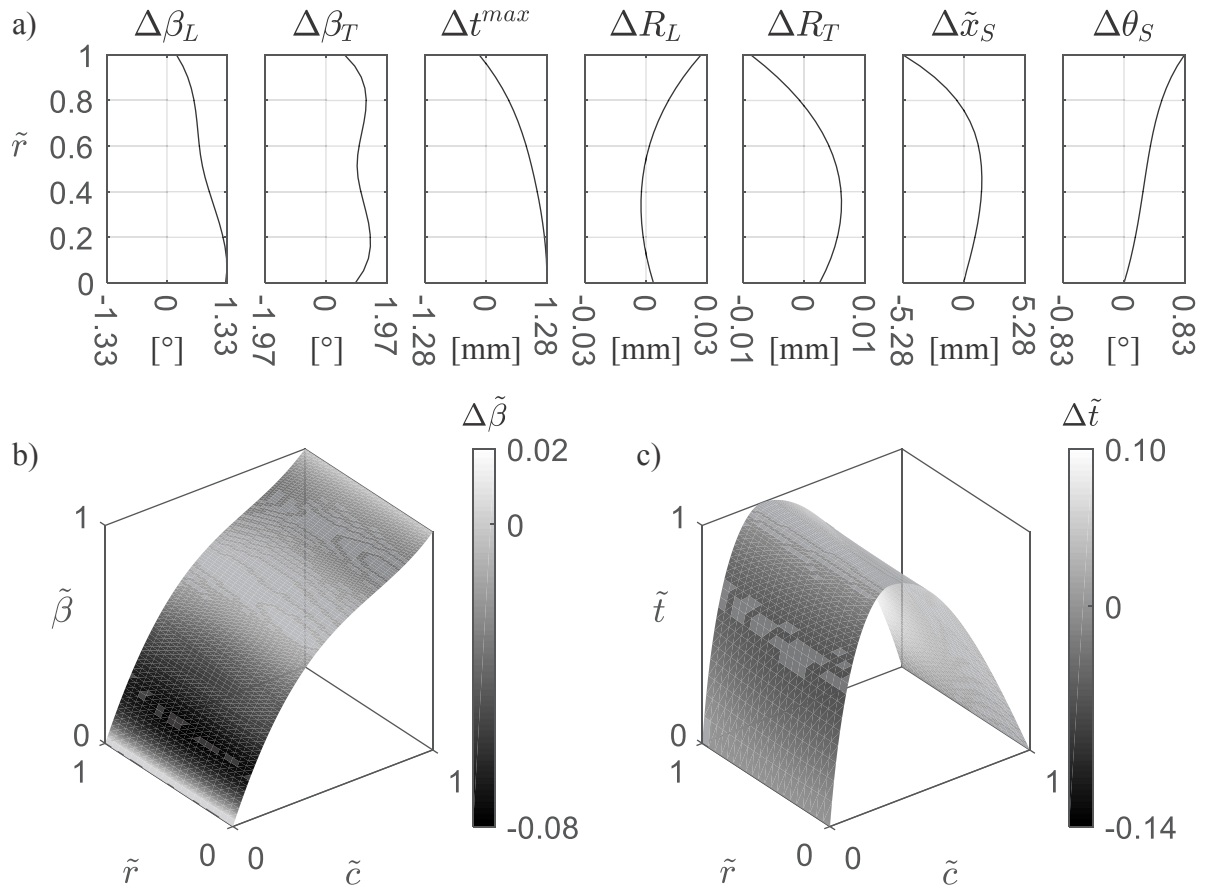


Figure 7.5: Difference between the parameter distributions of design A and reference design for a) one dimensional distributions, b) normalized chamber-line-angle distribution, and c) normalized thickness distribution

changes as well.

So far it was shown that for the given example tuning and tightening eigenmode bands such as 2H and 2C is possible, but it has not been clarified how much design freedom there exists to separately tune and tighten all eigenmode bands of the given example independently. Therefore, based on the DoE data it will be analyzed which parameters influence the position of the  $i^{\text{th}}$  eigenmode band, i.e.,  $\bar{f}_i^E := (\hat{f}_i^E + \check{f}_i^E)/2$ , and which ones influence the bandwidth  $\Delta f_i^E := \hat{f}_i^E - \check{f}_i^E$ . This analysis is performed using the regression-tree based nonlinear sensitivity-analysis method suggested by Lockan et al. (2014) which uses the leave-one-out method to successively identify the parameter which, if neglected, causes the smallest error between associated regression-tree models, and thus orders the parameter set according to importance. Figure 7.7a shows the results for frequency mean  $\bar{f}_i^E$  where the set of the ten most important parameters is plotted for each of the first 11 eigenmodes. It can be seen that none of the eigenmodes share the same order in parameter influence which means that parameter changes will affect the eigenfrequency bands differently. The same observation can be made for bandwidth  $\Delta f_i^E$  in Fig. 7.7b. The difference between the parameter IDs in Fig. 7.7c shows that the position and width of the

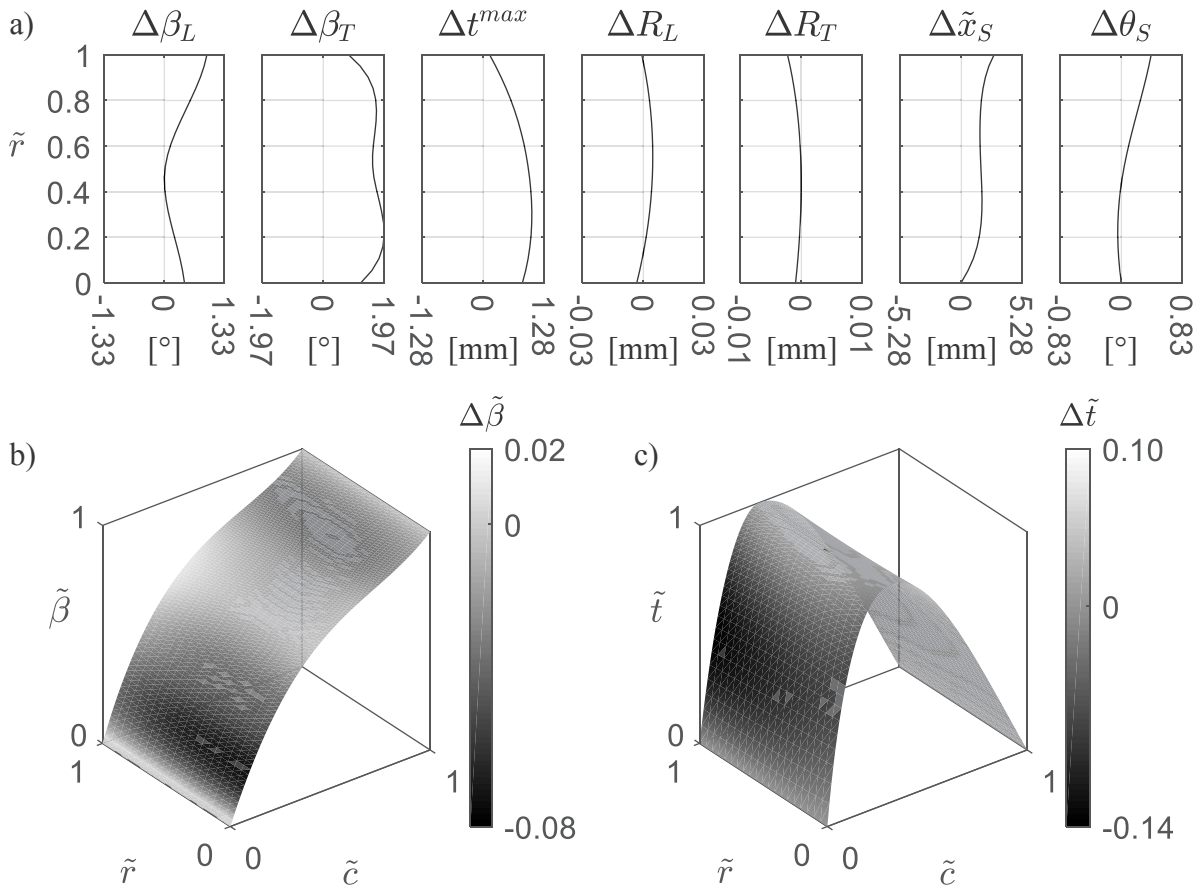


Figure 7.6: Difference between the parameter distributions of design B and reference design for a) one dimensional distributions, b) normalized chamber-line-angle distribution, and c) normalized thickness distribution

eigenfrequency bands are differently affected by parameter changes as well. In conclusion, to a certain degree there is design freedom to tune and tighten individual eigenfrequency bands differently.

### 7.3 Robust Design Results

Based on the previous deterministic optimization, the same DoE ( $\approx 3000$  samples) and  $\approx 2000$  designs from the deterministic optimization with constant weight function (2.14) are utilized for solving the probabilistic optimization problem (2.23) by reassessing production-tolerance criterion (2.22) for all designs. These results are used to create the initial RSMs for the probabilistic assessment as shown in Fig. 2.1. The optimization was stopped after about 1500 design evaluations when no noteworthy improvements were observable. The results, however, do not contain any design where  $\Delta s_{\perp}^{max} \neq 0$  or  $C = 0$  meaning that, due to the cheaper material, the design task is too challenging for finding a completely valid design which is producible within a tolerance of least  $\pm 0.01\text{mm}$ . However, reassessing all designs with a reduced tolerance step  $\Delta s_{\perp} = 0.001\text{mm}$  and choosing the design with the

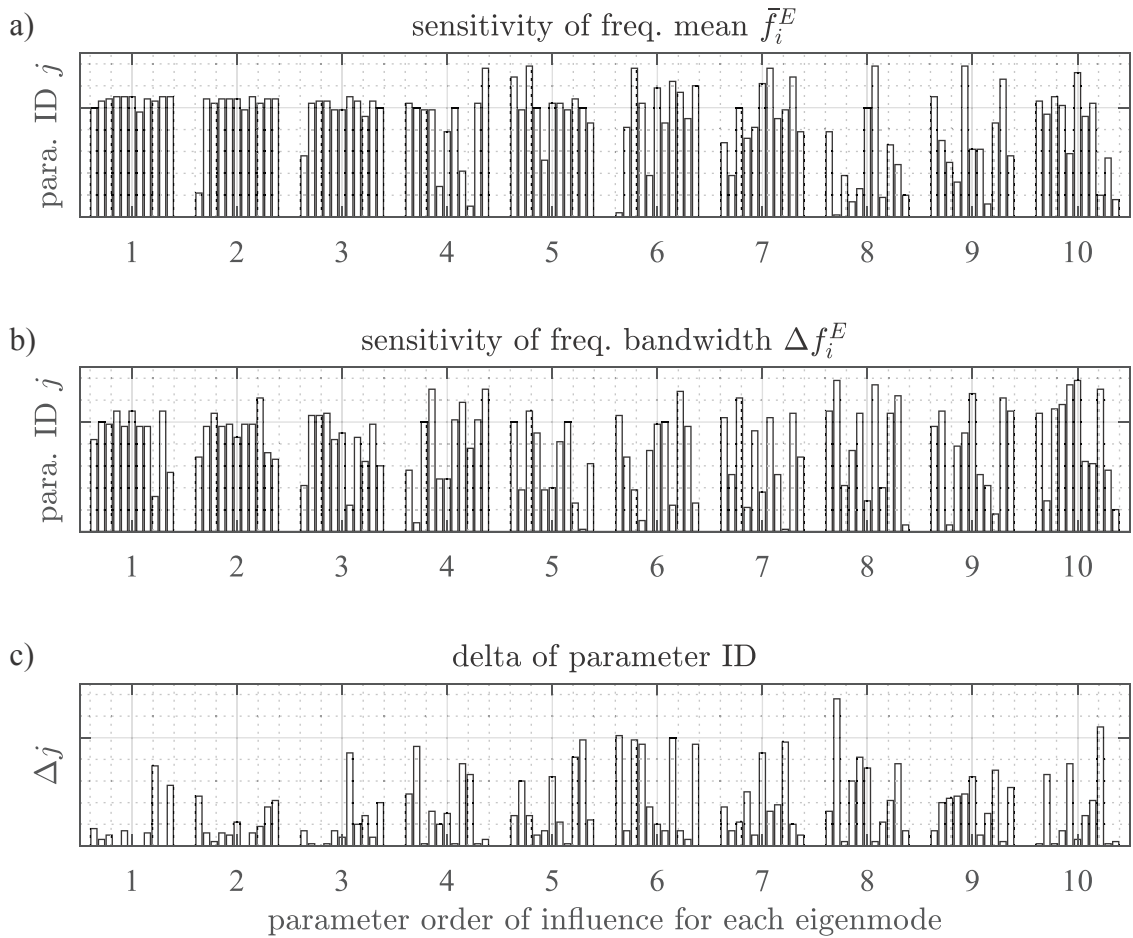


Figure 7.7: First ten most influential optimization parameters for the first 11 eigenmodes regarding a) frequency mean  $\bar{f}_i^E$ , b) frequency delta  $\Delta f_i^E$ , and the difference in parameter ID between those two

lowest weighted sum of constraints (2.9) gives a design that is producible within  $\pm 0.004\text{mm}$ . It violates only one less relevant eigenmode band and fulfills all other structural constraints. However, it suffers from a slightly less efficient aerodynamics than the reference design, but fulfills the required aerodynamic turning and all other aerodynamic constraints. In conclusion, the human expert has to decide now whether the challenging production tolerance negates the economy of the cheaper material or not, and in case it doesn't, if the savings justify the reduced aerodynamic efficiency, provided that the forced response of a less critical eigenmode doesn't harm service life.

A more detailed insight into the structural dynamic characteristic of the robust design and its deviation from the reference geometry is given in Fig. 7.8. Although the change of the geometry w.r.t. the reference design is smaller compared to results obtained from the deterministic optimizations in Fig. 7.3, the improvement of the structural dynamic behavior is still significant compared to Fig. 7.2. Two eigenmode bands intersect with one engine order within the operating speed range where one of the associated eigenmode

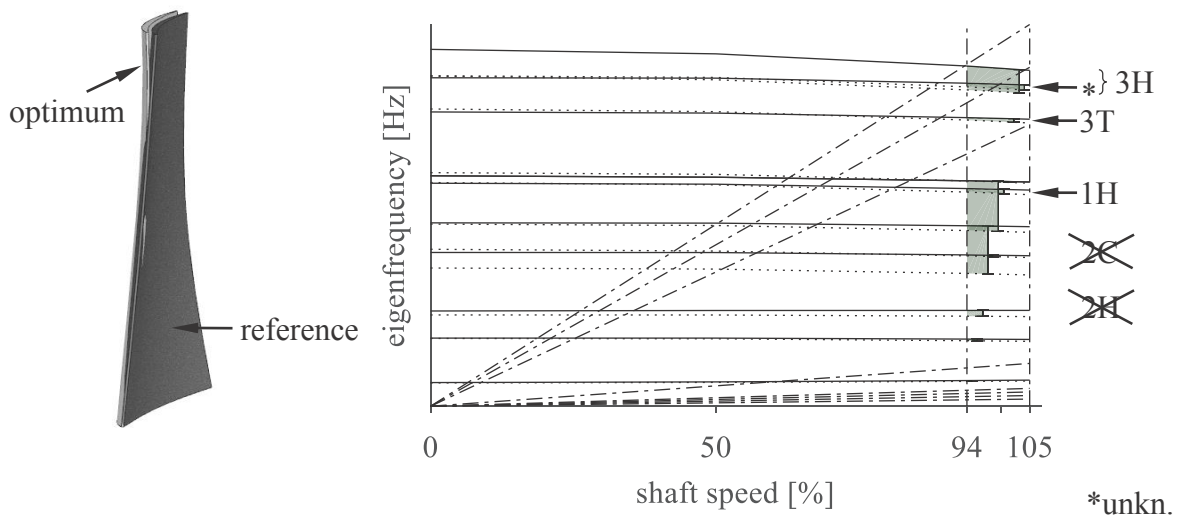


Figure 7.8: Geometry and Campbell diagram for the robust optimal design

shapes is unknown and therefore considered to be non-critical. The other 3H eigenmode shape suffers a low risk of HCF which, however, should be further assessed w.r.t. the service-life time using a high-fidelity tool.

Besides the dynamic behavior, also the maximum von Mises stresses are reduced to tolerable levels, see Fig. 7.9, compared to the reference design in Fig. 7.4a. The main reason for this is again the increased maximum thickness and not the leading and trailing edge radii which are increased insignificantly compared to the reference geometry, see Fig. 7.10a. In agreement with the deterministic optimization results, also the probabilistic optimization problem leads to an increased metal angle at the leading edge ( $\Delta\beta_L$ ) and reduced one at the trailing edge ( $\Delta\beta_T$ ). Accordingly, the values of the camber-line distribution in Fig. 7.10b and thickness distribution in Fig. 7.10c are reduced around the leading edge and nearly unchanged around the trailing edge. Additionally, axial shift  $\Delta\tilde{x}_S$  and theta shift  $\Delta\Theta_S$  (bow) are slightly increased.

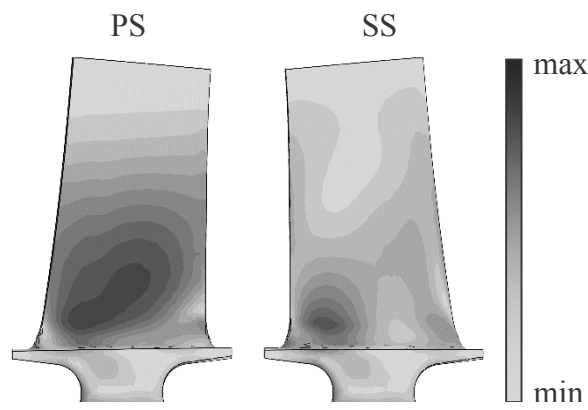


Figure 7.9: Von Mises-stress distribution of the robust optimum

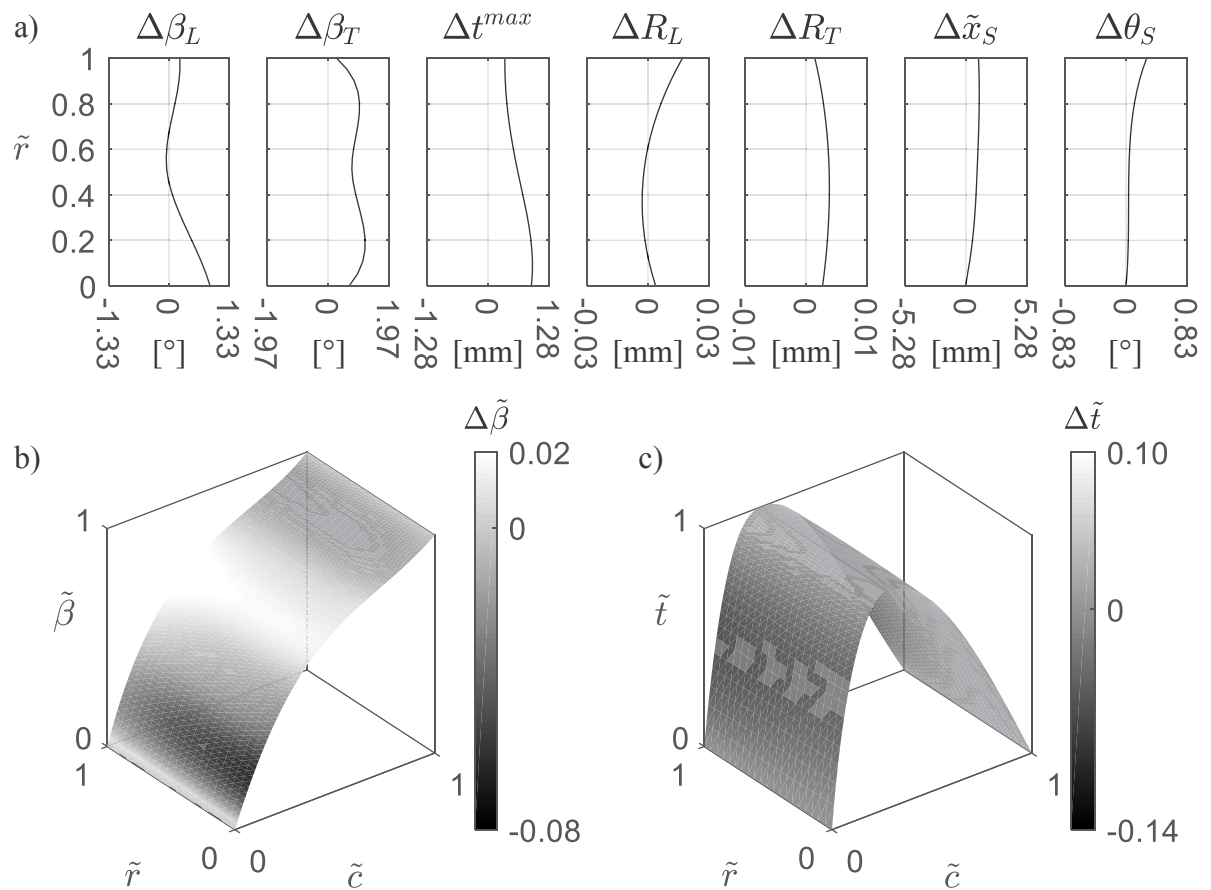


Figure 7.10: Difference between the optimal parameter distributions of the robust design and the distributions of the reference design for a) one-dimensional distributions, b) normalized chamber-line-angle distribution, and c) normalized thickness distribution



## 8 Conclusions and Outlook

In the introduction, several shortfalls of current multidisciplinary airfoil-design-optimization processes were identified and the goal was formulated to develop methods which allow to tackle those shortfalls in order to speed up the process and improve the quality of results. The focus of this thesis is the improvement of structural design evaluation, where special attention is paid to high-cycle fatigue which impairs the lifetime of airfoils. It was decided to utilize a penalty strategy suggested by Hecker et al. (2011) which assigns individual penalties to eigenmode shapes and engine orders w.r.t. their specific risk of failure in the scenario of forced response. Since this risk depends on the mode shape, procedures for eigenmode classification and eigenfrequency band assignment were developed for incorporation into an automated design-evaluation process. As the performance of each classification algorithm depends on the data structure, several studies were conducted to identify methods of data normalization and dimension reduction which are most beneficial to cluster separation. Based on a newly introduced distance ratio, which gives a better insight into data separation than the commonly used silhouette value, the nonlinear principle-component analysis (kPCA) is favored over various alternative methods including multiple-discriminate analysis (MDA). It was shown that the weakness of MDA are underlying statistical assumptions and it may be suggested to use discriminant directions calculated by support-vector machines (SVM) instead. In contrast to MDA, the SVM discriminates consider the local structure of cluster boundaries.

For the final classification of eigenmodes w.r.t. their shape, a feed-forward-neural network (FFNN) with error back-propagation and a newly developed analytic initialization of synapses weights based on principle component analysis is found to perform best. Eigenfrequency bands accounting for uncertain operating conditions are assigned based on the identified eigenmode shapes and a heuristic approach, which was developed on the basis of a study on the behavior of eigenfrequencies and eigenmode shapes under varying operating conditions. This study was conducted for some test airfoils equipped with hammer foot roots and revealed that frequency veering may appear as frequency crossing in regards to the eigenmode shapes.

As classification of eigenmode data for changing airfoil designs during optimization requires a standard of comparison, a projection method based on a self-organizing map (SOM) was developed which projects surface-mesh data of arbitrary geometries and types of finite-element meshes onto rectangular standard surfaces. It is shown that structuring FE-mesh data in an appropriate manner improves the quality of projection at edges of

surfaces significantly, and thus also the detection rate of relevant eigenmode shapes. The reason is that the most characteristic surface displacements of eigenmodes are located at the airfoil edges.

In order to receive proper optimal designs from a design optimization, structural design evaluation has to be coupled with an aerodynamic one. However, since the aerodynamic design is a loaded hot geometry resulting from its unloaded cold geometry after exposure to loads and temperatures at operating conditions, it is recognized that a fast loaded-to-unloaded transformation of airfoil geometries is needed. Otherwise, the structural properties of a produced airfoil will significantly deviate from the prediction during optimization as shown in a study. This led to the development of a new transformation method which employs negative density to reverse the centrifugal loads. Due to slender airfoil structures, this method has to perform linear rather than nonlinear deformation analysis in order to prevent buckling. The frequency and stress error due to the assumption of linear deformations is found to be negligible even for large airfoils of front stages. The advantage of the method over the conventional iterative approach described by Goerke et al. (2012) is its simplicity of implementation within the FEA of an airfoil and its superior computational speed.

Because optimal designs of a deterministic design-optimization process will most likely be located at the borders of design regions, even minor manufacturing deviations may violate constraints. Therefore it is essential to change from a deterministic to a probabilistic optimization problem where uncertainties in production tolerances are accounted for. In this thesis it is shown that the usually minimized failure rate is an insufficient objective for robust design because of its erroneous relation to reliability and uselessness as a manufacturing requirement. Therefore, a new approach was developed which assesses the possible production tolerance of a design. This approach relies on response surface models (RSMs) for design evaluation causing minimal additional computational effort, since the multidisciplinary design-optimization process already employs individual RSMs for different disciplines. This approach of individual RSMs increases process efficiency through excessive parallelization of subprocesses.

The methods and processes developed in this thesis have been validated on the basis of a rotor blade of an industrial compressor. Since the reference design is valid w.r.t. the defined constraints, a cheaper material with lower strength was chosen in order to challenge the optimization process. This makes the reference design violate nearly all structural constraints. The probabilistic optimization process, however, was able to find designs of only slightly decreased aerodynamic efficiency but validity w.r.t. all structural and aerodynamic constraints. The process was also able to shift and tighten eigenfrequency bands to such a degree that the optimal designs show minor risk of forced response resonance and flutter. Due to the probabilistic approach, designs were found which perform robust against production tolerances within the predicted range. For the mode shifting a new criterion was defined beforehand which involves a smooth intersection measure weighted with a function that may account for runtime probability within the gas turbines speed range. The effect of two different functions, one constant and one normal distribution, onto resulting optimal designs was investigated with the result that



the constant function performs better in mode shifting.

Due to limited project time and computational resources, the introduced methods still have some potential for improvement. For example, the simple three-layer FFNN used for mode-shape classification may be extended by additional hidden layers (deep learning) in order to be able to learn more complex features from the data. Furthermore, parallel layers with fixed synapses weights set to account for other type of filters, such as Haar, would most likely improve the classification error below the 4% achieved so far. One might also argue that a larger set of training data may reduce the classification error, which could be accomplished without human support by artificial data expansion. Also the data projection method using SOMs may be improved further if combined with k-means clustering as suggested by Kohonen (2014). Better computational resources regarding the number of CPUs would help to improve predictions of the production-tolerance-assessment method by allowing to include the whole parameter model (not just the reduced set of parameters considered here) and testing the actual CAD models for intersections. Additionally, improvement can be gained by including the chord length as an optimization parameter. Furthermore, the structure of the optimization process with a master and arbitrary slave processes allows to incorporate other high-fidelity tools than the ones considered here, e.g. aeroelasticity calculations for more accurate flutter assessment.



# A Additional Figures

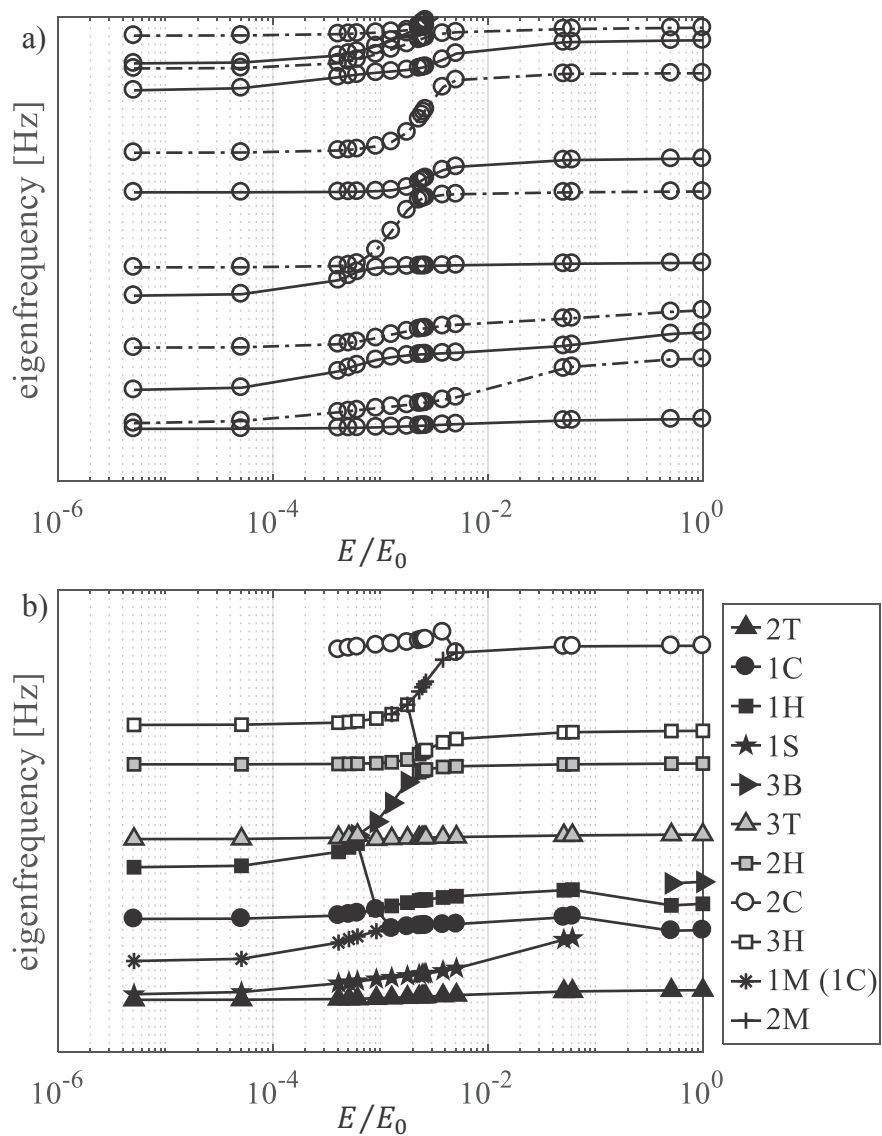


Figure A.1: Blade-fixation study of a front rotor (stage 5) with corresponding eigenfrequencies sorted by a) frequency values and b) eigenmode shapes

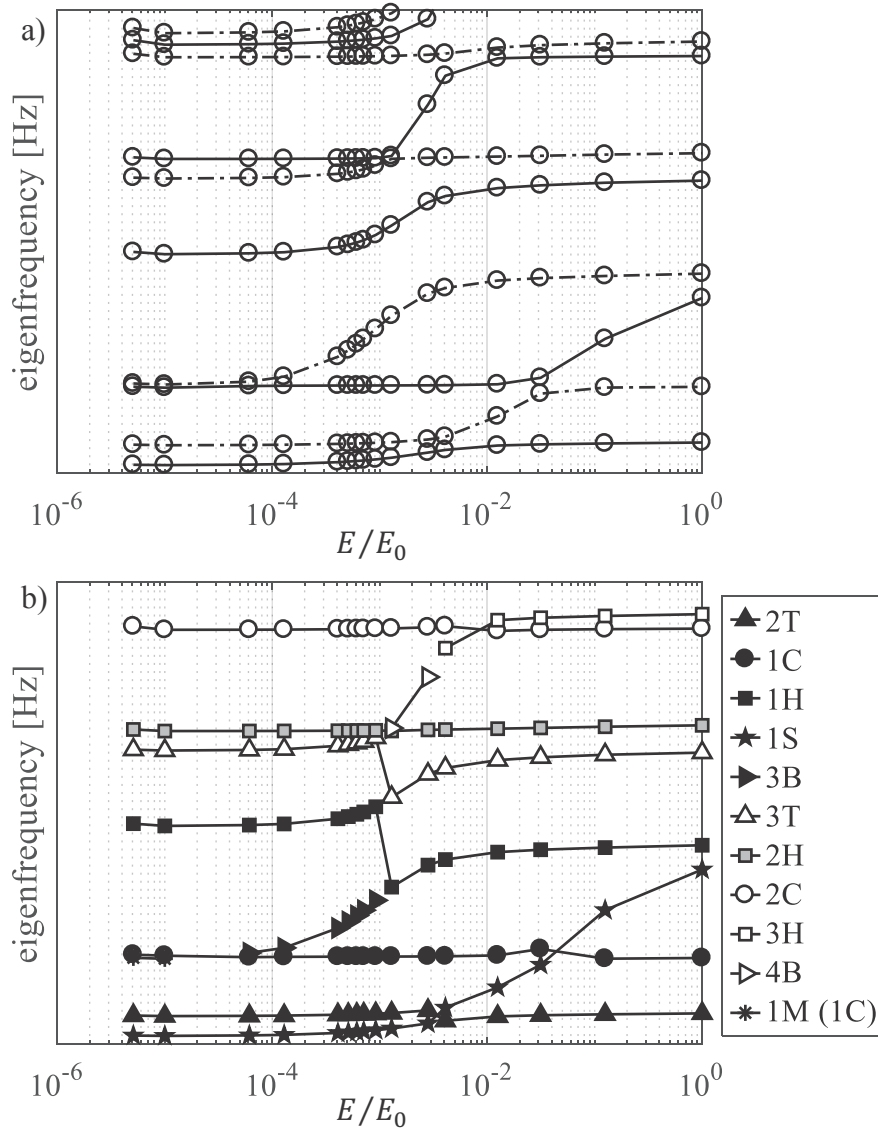


Figure A.2: Blade-fixation study of a middle rotor (stage 8) with corresponding eigenfrequencies sorted by a) frequency values and b) eigenmode shapes

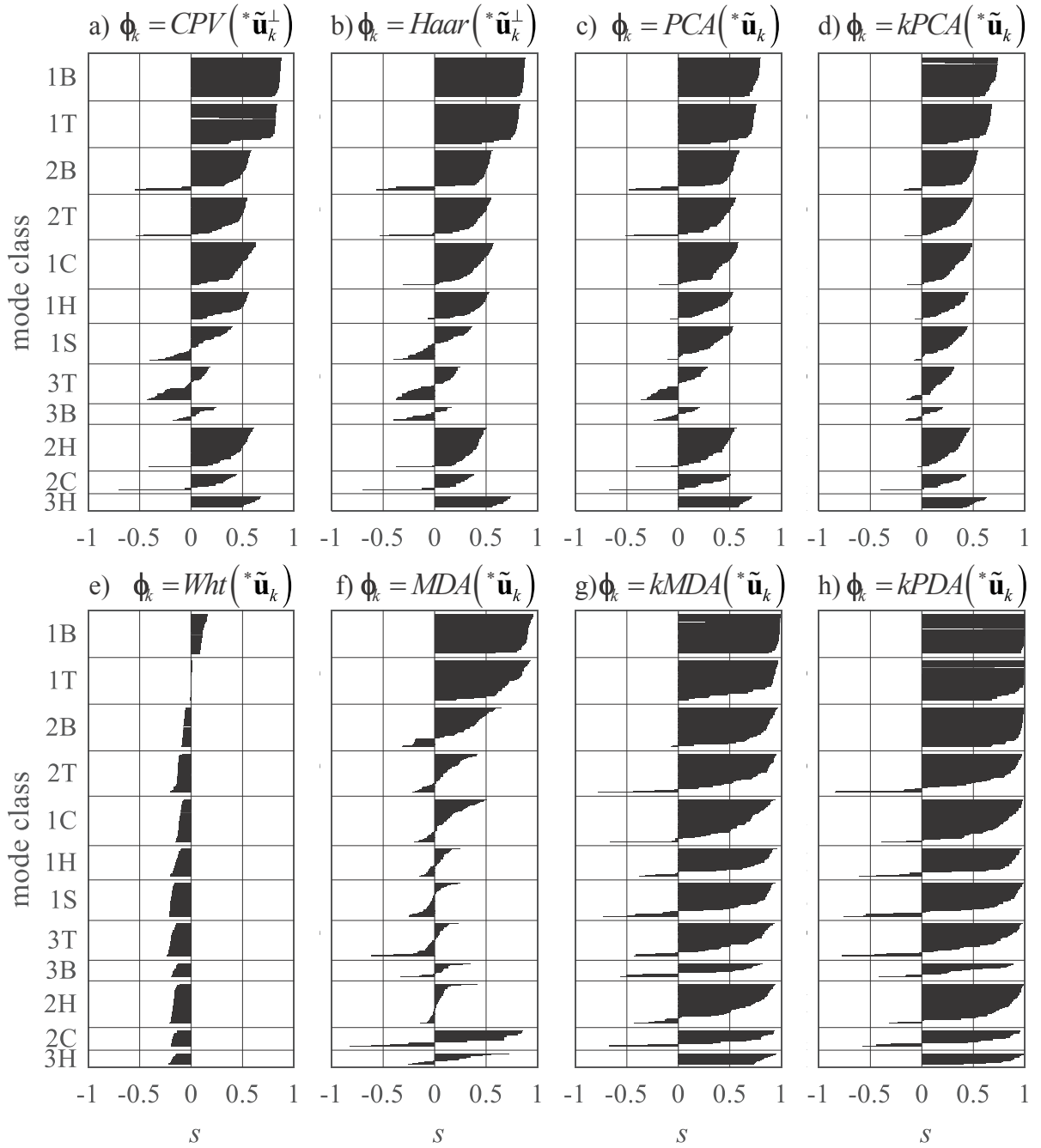


Figure A.3: Silhouette plots of the different dimension reduction methods

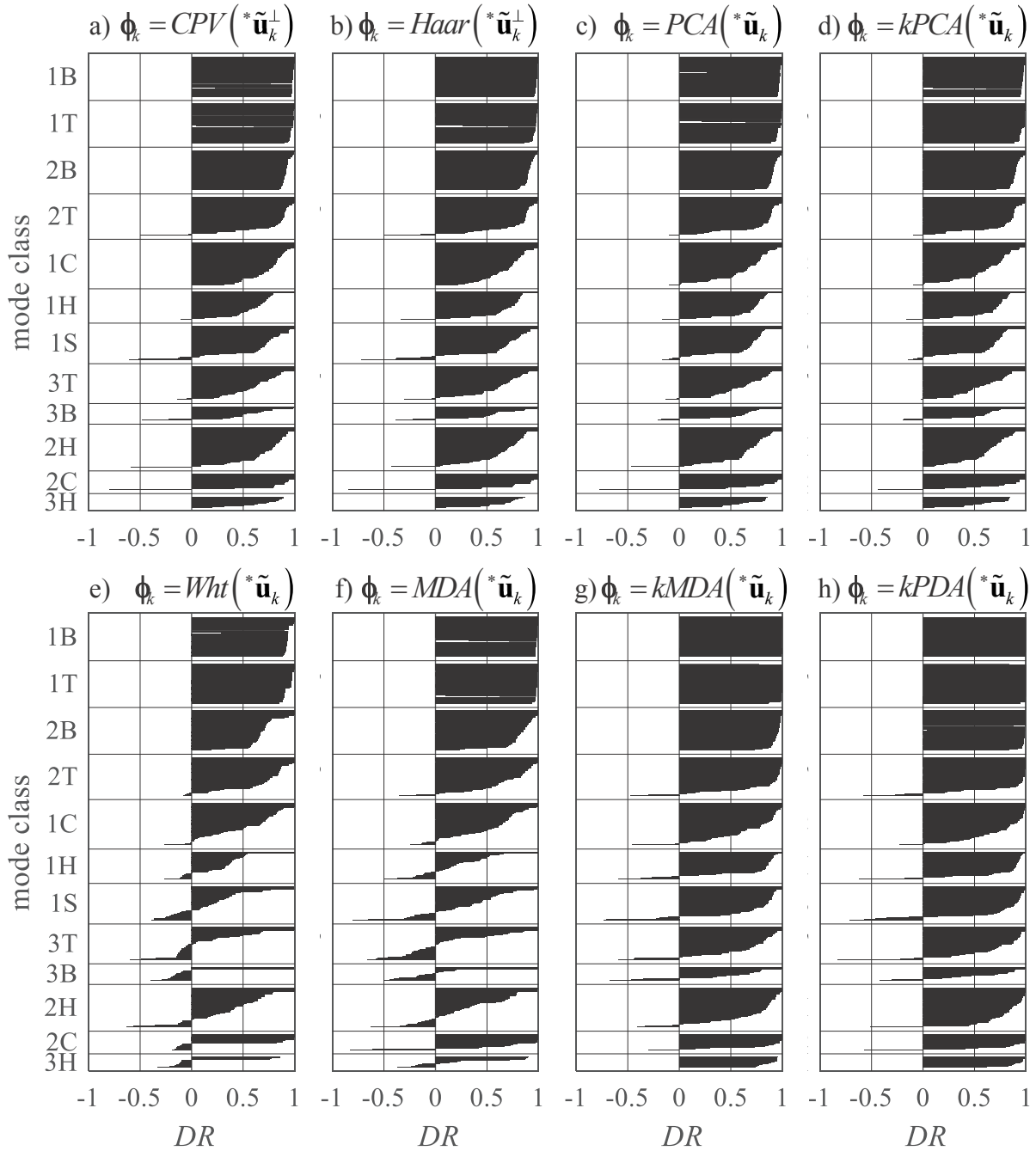


Figure A.4: Effect of different dimension reduction methods onto the distance ratio

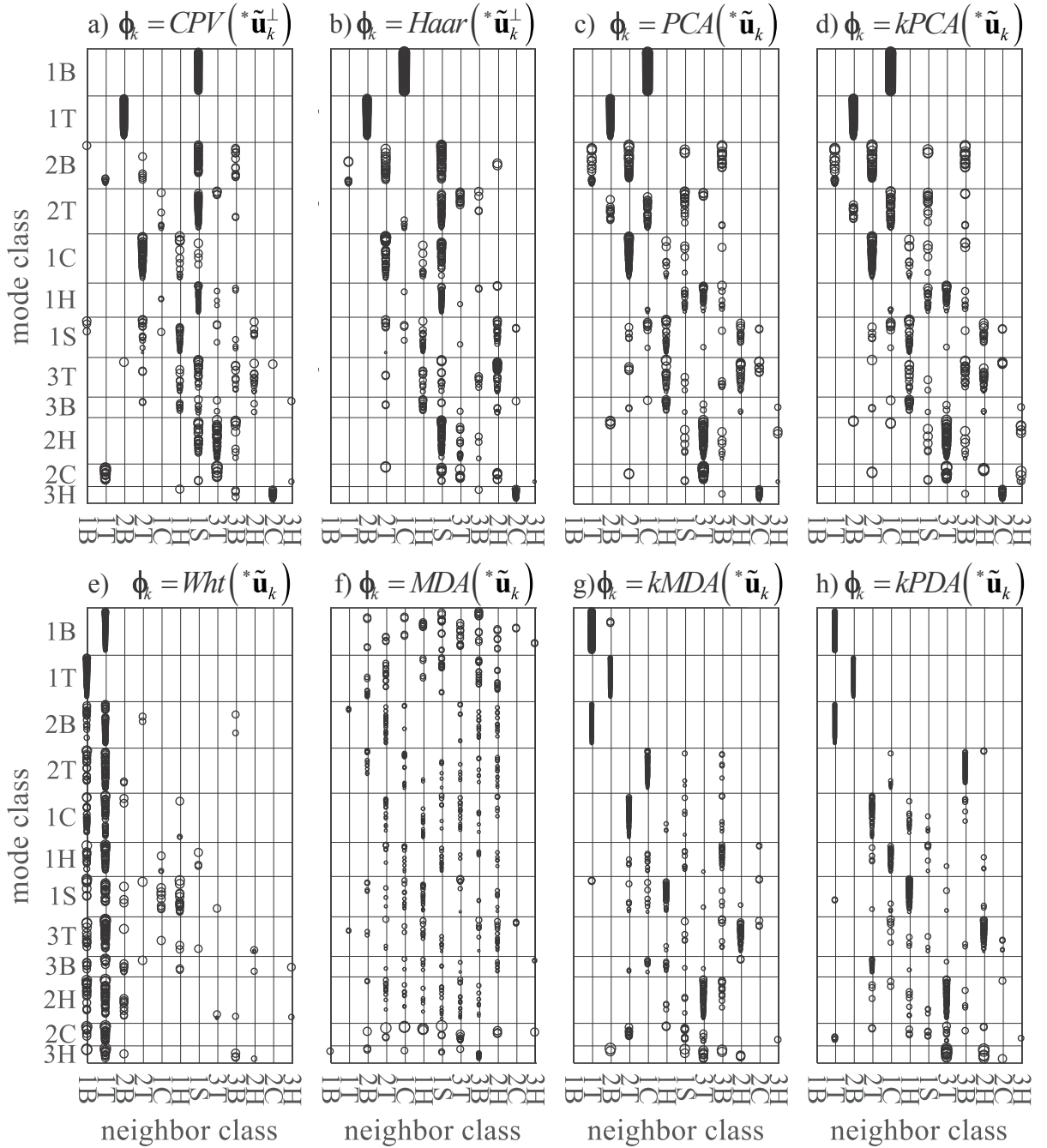


Figure A.5: Effect of different dimension reduction methods onto the type of cluster neighbors and distance (size of marker)

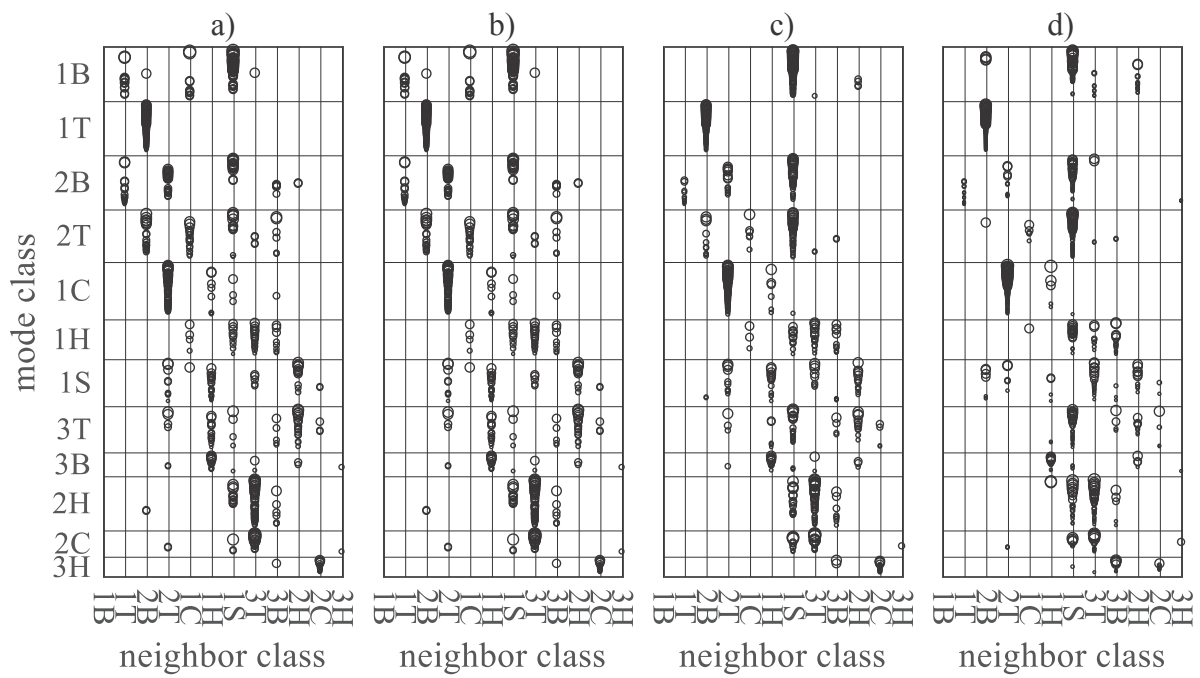


Figure A.6: Effect of BPNN data processing onto the type of cluster neighbors and distance (size of marker): a) before HL without biases, b) before HL with biases, c) after HL, and d) before OL with biases



## B Frequency Veering

An explanation of the frequency-veering phenomenon may be given by analyzing an oscillator consisting of two spring-coupled masses (Novotny (2010)) as shown in Fig. C.1. The equations of motion for each mass read as

$$\begin{aligned} m_1 \ddot{x}_1 &= -(k_1 + k_c) x_1 + k_c x_2 \\ m_2 \ddot{x}_2 &= k_c x_1 - (k_2 + k_c) x_2 \end{aligned} \quad (\text{B.1})$$

or in matrix form with  $\mathbf{x} = [x_1, x_2]^T$

$$\underbrace{\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}}_{\mathbf{M}} \ddot{\mathbf{x}} + \underbrace{\begin{bmatrix} (k_1 + k_c) & -k_c \\ -k_c & (k_2 + k_c) \end{bmatrix}}_{\mathbf{K}^L} \mathbf{x} = \mathbf{0} \quad (\text{B.2})$$

where  $\mathbf{M}$  and  $\mathbf{K}^L$  are the mass and linear stiffness matrix, respectively. Substitution of  $x$  with

$$\mathbf{x} = \boldsymbol{\phi} \exp(\lambda t) \quad (\text{B.3})$$

yields a generalized eigenproblem

$$(\lambda^2 \mathbf{M} + \mathbf{K}^L) \boldsymbol{\phi} = \mathbf{0} \quad (\text{B.4})$$

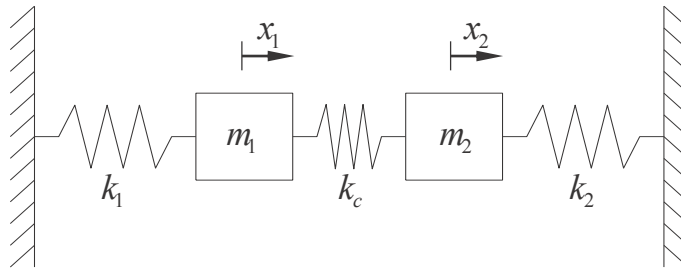


Figure C.1: Two spring-coupled masses

with four eigenvalues  $\lambda_{1,\dots,4}^2$  and nontrivial eigenvectors  $\boldsymbol{\phi}_{1,\dots,4} \neq \mathbf{0}$  if  $(\lambda^2 \mathbf{M} + \mathbf{K}^L)$  is singular, i.e.,

$$\det(\lambda^2 \mathbf{M} + \mathbf{K}^L) = m_1 m_2 \lambda^4 + \lambda^2 (m_1 (k_2 + k_c) + m_2 (k_1 + k_c)) + k_1 k_2 + k_c (k_1 + k_2) = 0. \quad (\text{B.5})$$

The solutions are

$$\lambda_{\pm}^2 = -\frac{1}{2} \left( \underbrace{\omega_1^2 + \omega_2^2}_{\text{I}} \mp \underbrace{\sqrt{(\omega_1^2 - \omega_2^2)^2 + 4 \frac{k_c^2}{m_1 m_2}}}_{\text{II}} \right) \quad (\text{B.6})$$

where  $\omega_{1,2} = \sqrt{(k_{1,2} + k_c) / m_{1,2}}$ . Because term I > II for  $k_1 k_2 + k_c (k_1 + k_2) > 0$ , it follows that  $\lambda_{\pm}^2 < 0$  which means there are four imaginary eigenvalues

$$\lambda_{1,2} = \sqrt{\lambda_{\pm}^2} = i\sqrt{-\lambda_{\pm}^2} \text{ and } \lambda_{3,4} = -\sqrt{\lambda_{\pm}^2} = -i\sqrt{-\lambda_{\pm}^2}. \quad (\text{B.7})$$

The corresponding eigenvectors are received by substituting the eigenvalues into Eq. (B.4). Due to Eq. (B.5), both equations are linearly dependent and the solution is ambiguous with a free remaining parameter accounting for the so far undetermined initial state of the system. Additionally, it may be noticed that due to (B.7)  $\lambda_{1,3}$  and  $\lambda_{2,4}$  will pairwise share the same eigenvector. The eigenvectors may be calculated from the first row of Eq. (B.4) e.g. as

$$\boldsymbol{\phi}_{1,3} = \begin{bmatrix} 1 \\ \frac{m_1 \lambda_{\pm}^2 + (k_1 + k_c)}{k_c} \end{bmatrix} =: \boldsymbol{\phi}_+, \quad \boldsymbol{\phi}_{2,4} = \begin{bmatrix} 1 \\ \frac{m_1 \lambda_{\mp}^2 + (k_1 + k_c)}{k_c} \end{bmatrix} =: \boldsymbol{\phi}_-. \quad (\text{B.8})$$

The final solution of differential equations (B.1) is a linear superposition of the four, linearly independent solutions (B.3), i.e.,

$$\mathbf{x}(t) = \alpha_1 \boldsymbol{\phi}_1 e^{\lambda_1 t} + \alpha_2 \boldsymbol{\phi}_2 e^{\lambda_2 t} + \alpha_3 \boldsymbol{\phi}_3 e^{\lambda_3 t} + \alpha_4 \boldsymbol{\phi}_4 e^{\lambda_4 t}. \quad (\text{B.9})$$

The eigenvalues (B.7) may be written as  $\lambda_{1,2} = i|\lambda_{\pm}|$  and  $\lambda_{3,4} = -i|\lambda_{\pm}|$  where  $|\lambda_{\pm}| = \sqrt{-\lambda_{\pm}^2}$ . This allows to rewrite Eq. (B.9) as

$$\begin{aligned} \mathbf{x}(t) = & (\alpha_1 + \alpha_3) \boldsymbol{\phi}_+ \cos(|\lambda_+| t) + i(\alpha_1 - \alpha_3) \boldsymbol{\phi}_+ \sin(|\lambda_+| t) \\ & + (\alpha_2 + \alpha_4) \boldsymbol{\phi}_- \cos(|\lambda_-| t) + i(\alpha_2 - \alpha_4) \boldsymbol{\phi}_- \sin(|\lambda_-| t) \end{aligned} \quad (\text{B.10})$$

using Euler's formula. Because for real initial conditions for  $\mathbf{x}(t) \in \mathbb{R}^2$  the solution shall be real as well, the coefficients of Eq. (B.10) have to be real, i.e.,  $(\alpha_1 + \alpha_3) \in \mathbb{R}$ ,  $i(\alpha_1 - \alpha_3) \in \mathbb{R}$ ,  $(\alpha_2 + \alpha_4) \in \mathbb{R}$ , and  $i(\alpha_2 - \alpha_4) \in \mathbb{R}$ . This means that a real solution requires conjugate complex coefficients  $\alpha_1 = \bar{\alpha}_3$ ,  $\alpha_2 = \bar{\alpha}_4$  and using the trigonometric addition rule gives the

final solution

$$\mathbf{x}(t) = \alpha_+ \boldsymbol{\phi}_+ \cos(|\lambda_+|t + \theta_+) + \alpha_- \boldsymbol{\phi}_- \cos(|\lambda_-|t + \theta_-) , \quad (\text{B.11})$$

where the constant amplitude factors  $\alpha_{\pm}$  and phase shifts  $\theta_{\pm}$  are defined through the initial state of the system. Equation (B.11) states that the system will respond to disturbances with an oscillation which is a superposition of the two eigenfrequencies  $|\lambda_{\pm}|$  and their corresponding eigenvectors  $\boldsymbol{\phi}_{\pm}$ .

Analyzing the frequency veering phenomenon can be done by evaluating Eq. (B.6) and Eq. (B.8) for a varying stiffness of either  $k_1$  or  $k_2$  and a varying coupling stiffness  $k_c$ . Figure C.2 shows such an evaluation, where the stiffness  $k_1 := 4\pi\text{N/m}$  is constant and  $k_2 := k_1 + \Delta k$  is varied depending on  $-k_1 \leq \Delta k \leq k_1$ ; the coupling stiffness is set to be  $k_c \in \{0, 1, 2, 4\} \text{N/m}$ , and the masses are chosen to be  $m_1 = m_2 := 1\text{kg}$ . For zero coupling ( $k_c = 0$ ) the eigenfrequencies in Fig. C.2a cross because  $|\lambda_-| = \omega_1 = \sqrt{k_1/m_1}$  is not affected by the increase of  $k_2$  and only  $|\lambda_+| = \omega_2 = \sqrt{k_2/m_2}$  rises. With coupling ( $k_c \neq 0$ ), however, both eigenfrequencies are affected by the increasing stiffness  $k_2$ , and the two eigenfrequencies of the system veer away when  $k_1 = k_2$  ( $\Delta k = 0$ ). The distance of closest approach (for  $m_1/m_2 = 1$  at  $\Delta k/k_1 = 0$ ) increases with increasing coupling stiffness  $k_c$ , and it would decrease in case of increasing  $m_1$  or  $m_2$  (not shown in Fig. C.2a where  $m_1 = m_2$ ), which may be shown using Eq. (B.6) as follows:

$$\lambda_-^2 - \lambda_+^2 = \sqrt{(\omega_1^2 - \omega_2^2)^2 + 4 \frac{k_c^2}{m_1 m_2}} \quad (\text{B.12})$$

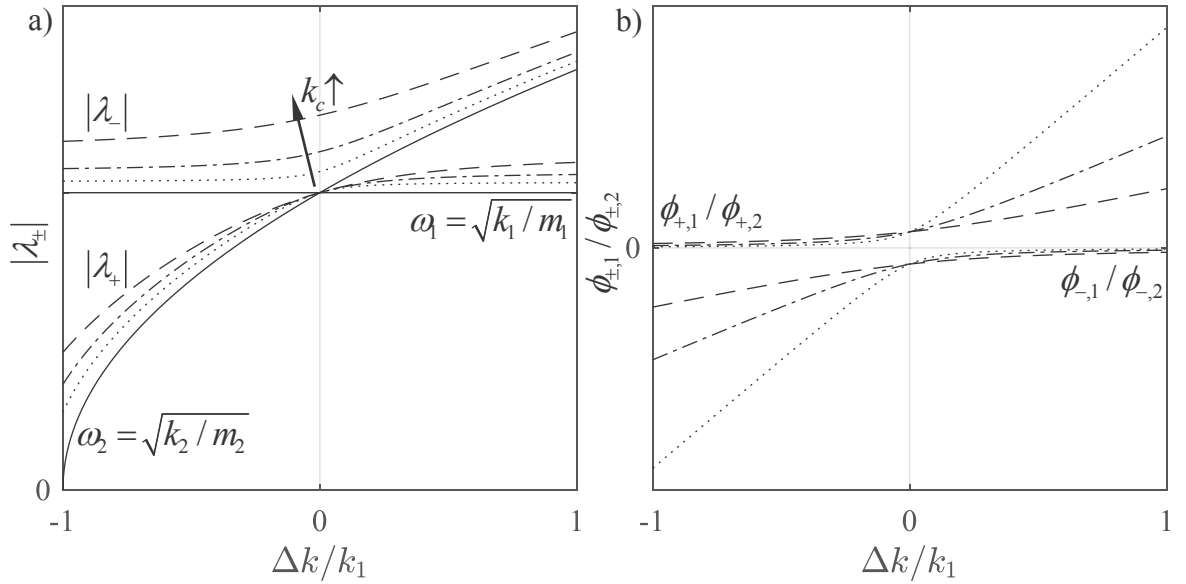


Figure C.2: Eigenfrequencies (a) and amplitude ratios (b) of coupled masses as variations over the difference in stiffness of the wall connected springs

$$0 \stackrel{!}{=} \frac{\partial (\lambda_-^2 - \lambda_+^2)}{\partial \Delta k} = \frac{\partial (\lambda_-^2 - \lambda_+^2)}{\partial \omega_2} \frac{\partial \omega_2}{\partial k_2} \frac{\partial k_2}{\partial \Delta k} = \frac{(\omega_2^2 - \omega_1^2)}{m_2 \sqrt{(\omega_1^2 - \omega_2^2)^2 + 4 \frac{k_c^2}{m_1 m_2}}} \quad (\text{B.13})$$

which has one solution  $(\omega_2^2 - \omega_1^2) = 0$  leading to

$$\min [\lambda_-^2 - \lambda_+^2] = 2k_c / \sqrt{m_1 m_2} \quad \text{at} \quad \frac{\Delta k}{k_1} = \left( \frac{k_c}{k_1} + 1 \right) \left( \frac{m_2}{m_1} - 1 \right). \quad (\text{B.14})$$

Thus, the position of veering  $\Delta k/k_1$  depends on the mass ratio  $m_2/m_1$  and stiffness ratio  $k_c/k_1$ . The distance only becomes zero ( $\hat{=}$ crossing) if  $k_c = 0$ , else veering will occur, where distance increases with  $k_c$ .

The effect of frequency veering onto the eigenmode shapes, i.e., the amplitude ratios  $\phi_{\pm,1}/\phi_{\pm,2}$  with  $\boldsymbol{\phi}_{\pm} = [\phi_{\pm,1}, \phi_{\pm,2}]^T$ , is shown in Figure C.2b where the eigenmodes undergo a significant change within the region of frequency veering. Interestingly, the character of the first eigenmode  $\boldsymbol{\phi}_+$  with  $|\phi_{+,2}| \gg |\phi_{+,1}|$ , is taken over by the second eigenmode  $\boldsymbol{\phi}_-$  after frequency veering, which means that according to the eigenmode shape frequency veering may be interpreted as frequency crossing as seen in Fig. 3.6b&c. The second eigenmode, however, changes its character from the two masses oscillating in opposite directions to oscillating in same direction, both at a comparable level.

Concluding, it can be said that due to frequency veering, eigenmodes may appear to cross and change their order or they may transform into different new mode shapes. In case an eigenmode changes its order due to frequency crossing, the level of eigenfrequency change is proportional to the strength of mode coupling. The position of frequency veering (in the example here  $\Delta k/k_1 = 0$ ) is affected by the mass- as well as stiffness distribution of the system.

# C Mathematics of Chosen Methods

## C.1 Principal Component Analysis

This section describes the mathematics of principle component analysis (PCA). The main purpose of PCA is to find main directions of variance, called principle components, in a set of data points. While these directions are linear the application of the kernel trick allows to find nonlinear directions. Both PCA and kernel-PCA will be described in the following.

### C.1.1 Linear PCA

A set of data points  $\boldsymbol{\phi}_k \in \mathbb{R}^D$ ,  $k = 1, \dots, N$ , may be interpreted as a sample of a random vector  $\boldsymbol{\phi}$  with mean vector  $\boldsymbol{\mu} = E[\boldsymbol{\phi}]$  which can be approximated as

$$\boldsymbol{\mu} \approx \mathbf{m} = \frac{1}{N} \sum_{k=1}^N \boldsymbol{\phi}_k . \quad (\text{C.1})$$

If the data set is centered, i.e.,  $\boldsymbol{\phi} - \boldsymbol{\mu}$ , the variance of the data set in a specific direction given by unit vector  $\mathbf{v}$ ,  $\mathbf{v}^T \mathbf{v} = 1$ , can be calculated from the projection  $(\boldsymbol{\phi} - \boldsymbol{\mu})^T \mathbf{v}$  as follows:

$$\sigma^2 = E \left[ \left( (\boldsymbol{\phi} - \boldsymbol{\mu})^T \mathbf{v} \right)^2 \right] \equiv E \left[ \mathbf{v}^T (\boldsymbol{\phi} - \boldsymbol{\mu}) (\boldsymbol{\phi} - \boldsymbol{\mu})^T \mathbf{v} \right] = \mathbf{v}^T \boldsymbol{\Sigma} \mathbf{v} \quad (\text{C.2})$$

where  $\boldsymbol{\Sigma}$  is the symmetric covariance matrix of  $\boldsymbol{\phi}$  which can be estimated as

$$\boldsymbol{\Sigma} = E \left[ (\boldsymbol{\phi} - \boldsymbol{\mu}) (\boldsymbol{\phi} - \boldsymbol{\mu})^T \right] \approx \frac{1}{N-1} \sum_{k=1}^N (\boldsymbol{\phi}_k - \mathbf{m}) (\boldsymbol{\phi}_k - \mathbf{m})^T . \quad (\text{C.3})$$

The direction  $\mathbf{v}$  of highest variance maximizing Eq. (C.2) under the constraint  $\mathbf{v}^T \mathbf{v} = 1$  can be found by the necessary condition of a vanishing derivative of the Lagrange function  $L$ :

$$\frac{\partial L}{\partial \mathbf{v}} = \frac{\partial \left( \mathbf{v}^T \boldsymbol{\Sigma} \mathbf{v} - \lambda (\mathbf{v}^T \mathbf{v} - 1) \right)}{\partial \mathbf{v}} = 2 (\boldsymbol{\Sigma} \mathbf{v} - \lambda \mathbf{v}) \stackrel{!}{=} \mathbf{0} \quad (\text{C.4})$$

where the Lagrange multiplier  $\lambda$  is the eigenvalue to the eigenvector  $\mathbf{v}$  in the eigenvalue problem

$$\mathbf{\Sigma}\mathbf{v} = \lambda\mathbf{v} . \quad (\text{C.5})$$

Thus, the eigenvectors  $\mathbf{v}$  and eigenvalues  $\lambda$  of the covariance matrix  $\mathbf{\Sigma}$  are the local maxima/minima of (C.2), and substituting (C.5) into Eq. (C.2) yields  $\mathbf{v}^T\mathbf{\Sigma}\mathbf{v} \equiv \lambda\mathbf{v}^T\mathbf{v} \equiv \lambda$ . Hence, the Lagrange multipliers  $\lambda$  are equal to the variances  $\sigma^2$  in the principle-component directions  $\mathbf{v}$  of  $\mathbf{\Sigma}$ . Because of (C.3),  $\mathbf{\Sigma} \in \mathbb{R}^{D \times D}$  is  $D$ -dimensional due to  $\boldsymbol{\phi} \in \mathbb{R}^D$ , symmetric and at least positive-semidefinite. Thus, there are  $D$  orthonormal principle components  $\mathbf{v}^i$  with corresponding variances  $\sigma_i^2 \geq 0$ , which may be ordered as  $\sigma_1^2 \geq \sigma_2^2 \geq \dots \geq \sigma_D^2 \geq 0$ .

The symmetric positive-semidefinite covariance matrix  $\mathbf{\Sigma}$  can be decomposed by using singular value decomposition (Golub and Reinsch (1971)) such that

$$\mathbf{\Sigma} = \mathbf{V}\mathbf{D}\mathbf{V}^T = \mathbf{V}\mathbf{D}^{1/2}\mathbf{D}^{1/2}\mathbf{V}^T \quad (\text{C.6})$$

where  $\mathbf{V} = [\mathbf{v}^1, \dots, \mathbf{v}^D]$ ,  $\mathbf{D} = \text{diag}(\sigma_j^2)$ , and  $\mathbf{V}^T\mathbf{V} = \mathbf{I}$  or  $\mathbf{v}^{iT}\mathbf{v}^j = \delta_{ij}$  (Kronecker delta). Because  $\mathbf{D}$  is a straining and  $\mathbf{V}$  a rotating matrix, the distribution of  $\boldsymbol{\phi}$  may be obtained from standard normally distributed vectors  $\mathbf{x}$  via

$$\boldsymbol{\phi} := \boldsymbol{\mu} + \mathbf{V}\mathbf{D}^{1/2}\mathbf{x} = \boldsymbol{\mu} + \sum_{i=1}^D \sigma_i x_i \mathbf{v}^i \text{ where } x_i \sim \mathcal{N}\{0, 1\} . \quad (\text{C.7})$$

The proof can be given by substituting transformation (C.7) in (C.3) and comparing the result to (C.6):

$$\text{E}[(\boldsymbol{\phi} - \boldsymbol{\mu})(\boldsymbol{\phi} - \boldsymbol{\mu})^T] = \text{E}[\mathbf{V}\mathbf{D}^{1/2}\mathbf{x}\mathbf{x}^T\mathbf{D}^{1/2}\mathbf{V}^T] = \mathbf{V}\mathbf{D}^{1/2} \underbrace{\text{E}[\mathbf{x}\mathbf{x}^T]}_{=\mathbf{I}} \mathbf{D}^{1/2}\mathbf{V}^T \equiv \mathbf{\Sigma} . \quad (\text{C.8})$$

Because the size of each term in Eq. (C.7) is mainly determined by the standard deviation  $\sigma_i$ , since  $\|\mathbf{v}^i\| = 1$  and  $x_i \sim \mathcal{N}\{0, 1\}$ , terms with small  $\sigma_i$  may be neglected. The sorting  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_D$  allows to write an approximation with  $d$  major terms as

$$\boldsymbol{\phi} \approx \boldsymbol{\mu} + \sum_{i=1}^d \sigma_i x_i \mathbf{v}^i = \boldsymbol{\mu} + \mathbf{V}_R \mathbf{D}_R^{1/2} \mathbf{x}_R \quad (\text{C.9})$$

where  $\mathbf{V}_R = [\mathbf{v}^1, \dots, \mathbf{v}^d]$ ,  $\mathbf{D}_R = \text{diag}\{\sigma_1^2 \dots \sigma_d^2\}$ , and  $x_{Ri} \sim \mathcal{N}\{0, 1\}$ . Accordingly, the major information can be maintained at reduced dimension  $d < D$  with a projection

$$\bar{\boldsymbol{\phi}}_R' := \mathbf{V}_R^T(\boldsymbol{\phi} - \boldsymbol{\mu}) = \underbrace{\mathbf{V}_R^T \mathbf{V}_R}_{=\mathbf{I}} \mathbf{D}_R^{1/2} \mathbf{x}_R = \mathbf{D}_R^{1/2} \mathbf{x}_R \sim \mathcal{N}\{\mathbf{0}, \mathbf{D}_R\} \quad (\text{C.10})$$

onto the directions  $\mathbf{v}^i$  with highest variances. Therefore, the maximum amount of data

variance the reduction shall account for, determines the number  $d$  of directions  $\mathbf{v}^i$  to be considered, i.e., the dimensionality of the reduction. For example, in case of 99.99% one may find  $d$  as

$$d = \min_{I \in \mathbb{N}} I \quad \text{s.t.} \quad \sum_{i=1}^I \sigma_i^2 / \sum_{i=1}^D \sigma_i^2 \geq 99.99\% . \quad (\text{C.11})$$

The PCA method is often used for whitening data which means that the projection (C.10) is adapted in such a way that the coordinate variances of  $\bar{\boldsymbol{\Phi}}'_R$  are all equal to one. Therefor, either  $\mathbf{D}_R^{1/2}$  is set to be the identity matrix  $\mathbf{I}$  or each eigenvector is scaled such that

$$\mathbf{v}^{iT} \mathbf{v}^i = \frac{1}{\sigma_i^2} . \quad (\text{C.12})$$

### C.1.2 Nonlinear PCA with Kernel-trick

Because linear PCA is limited to finding only straight directions of main variances, it was suggested by Schölkopf et al. (1997) and inspired by Boser et al. (1992) to execute the linear PCA in a nonlinear feature space  $\mathbb{R}^F$  of such a high dimension (even infinite if necessary, i.e.,  $F \rightarrow \infty$ ) that nonlinearities in the data set disappear. The data are transformed into the feature space therefor, using nonlinear functions

$$\mathbf{f} : \mathbb{R}^D \rightarrow \mathbb{R}^{F > D}, \quad \hat{\boldsymbol{\Phi}} = \mathbf{f}(\boldsymbol{\Phi}) \quad (\text{C.13})$$

where, and just for now, it is assumed that the transformed data points are centered, i.e.,  $\sum_{k=1}^N \mathbf{f}(\boldsymbol{\Phi}_k) = \mathbf{0}$ . Consequently, the scatter matrix (C.3), i.e., the estimator of the covariance matrix, becomes

$$\hat{\boldsymbol{\Sigma}} \approx \frac{1}{N-1} \sum_{k=1}^N \mathbf{f}(\boldsymbol{\Phi}_k) \mathbf{f}^T(\boldsymbol{\Phi}_k) = \frac{1}{N-1} \mathbf{F} \mathbf{F}^T \quad \text{where } \mathbf{F} := [\mathbf{f}_1, \dots, \mathbf{f}_N], \quad \mathbf{f}_k := \mathbf{f}(\boldsymbol{\Phi}_k) . \quad (\text{C.14})$$

In accordance to linear PCA, the principle components and main variances are received by solving the eigenvalue problem (diagonalization of scatter matrix)

$$\hat{\boldsymbol{\Sigma}} \hat{\mathbf{v}}^i = \hat{\sigma}_i^2 \hat{\mathbf{v}}^i, \quad \|\hat{\mathbf{v}}^i\| = 1 \quad (\text{C.15})$$

similarly to Eq. (C.5) and ordering eigenvalues as  $\hat{\sigma}_1^2 \geq \hat{\sigma}_2^2 \geq \dots \geq \hat{\sigma}_F^2$ . Due to the unknown transformation (C.13) and thus unknown  $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{F \times F}$ , no practical solution can be obtained directly from this eigenproblem. However, by realizing that the final goal is to project all  $\mathbf{f}_k$  onto the directions  $\hat{\mathbf{v}}^i$  of main variances  $\hat{\sigma}_i^2$  analogously to Eq. (C.10), the eigenproblem (C.15) can be used to calculate this projection in a subspace of  $\mathbb{R}^{F \times F}$  by using the kernel

trick which will be explained later. First, similar to (C.10) this projection is specified as

$$\mathbf{f}_k'^T := \mathbf{f}_k^T \hat{\mathbf{V}}_R = [\mathbf{F}^T \hat{\mathbf{V}}_R]_k =: [\mathbf{B}_R]_k \quad \text{where } \hat{\mathbf{V}}_R = [\hat{\mathbf{v}}^1, \dots, \hat{\mathbf{v}}^d], \mathbf{B}_R = [\boldsymbol{\beta}^1, \dots, \boldsymbol{\beta}^d] \quad (\text{C.16})$$

and according to (C.9) those  $\hat{\mathbf{v}}^{i>d}$  with negligible variances, especially  $\hat{\sigma}_i^2 = 0$ , may be omitted and  $[\mathbf{B}_R]_k$  is the  $k^{\text{th}}$  row of  $\mathbf{B}_R$ . Its column  $\boldsymbol{\beta}^i$  is the projection of all  $\mathbf{f}_k$  onto a specific eigenvector  $\hat{\mathbf{v}}^i$ , i.e.,  $\mathbf{F}^T \hat{\mathbf{v}}^i =: \boldsymbol{\beta}^i$ , which can be calculated using Eq. (C.14) by multiplying (C.15) with  $\mathbf{F}^T$  and solving the following eigenproblem:

$$\mathbf{F}^T \hat{\Sigma} \hat{\mathbf{v}}^i \equiv \frac{1}{N-1} \underbrace{\mathbf{F}^T \mathbf{F}}_{=: \mathbf{K}} \underbrace{\mathbf{F}^T \hat{\mathbf{v}}^i}_{\boldsymbol{\beta}^i} = \hat{\sigma}_i^2 \underbrace{\mathbf{F}^T \hat{\mathbf{v}}^i}_{\boldsymbol{\beta}^i} \rightarrow \mathbf{K} \boldsymbol{\beta}^i = (N-1) \hat{\sigma}_i^2 \boldsymbol{\beta}^i \rightarrow \mathbf{K} \mathbf{B} = \mathbf{B} \boldsymbol{\Lambda} \quad (\text{C.17})$$

where  $\boldsymbol{\Lambda} = \text{diag}((N-1) \hat{\sigma}_i^2)$  and  $\mathbf{B} = [\boldsymbol{\beta}^1, \dots, \boldsymbol{\beta}^N]$ . The matrix  $\mathbf{K} \in \mathbb{R}^{N \times N}$  with elements  $K_{ij} = \mathbf{f}_i^T \mathbf{f}_j$  is called the kernel matrix. Due to the symmetry of  $\mathbf{K}$ , the eigenvectors  $\boldsymbol{\beta}^i$  are orthogonal and may be normalized such that  $\mathbf{B}^T \mathbf{B} = \mathbf{I}$ .

It becomes apparent that by solving Eq. (C.17) instead of Eq. (C.15), the problem is transferred into a subspace  $\mathbb{R}^{N \times N}$  of  $\mathbb{R}^{F \times F}$  and that the dimension of the projection (C.16) can be  $d = N$  at maximum. The motivation for rewriting Eq. (C.15) as Eq. (C.17) is to receive a formulation where the unknown transformation function  $\mathbf{f}$  only occurs in a scalar product  $\mathbf{f}_i^T \mathbf{f}_j$  with itself. This scalar product, as suggested by Aizerman et al. (1964), can actually be calculated through a representative simpler function named kernel function. The transformation  $\mathbf{f}_i$  itself, and hence dimension  $F$  and the subspace Eq. (C.17) is solved in, is defined through the choice of the kernel function. This kernel trick requires that the kernel function actually defines a scalar product of some transformation  $\mathbf{f}_i$  in a feature space  $\mathbb{R}^F$ , which is given if it fulfills Mercer's condition (Mercer (1909)).

An example of a valid kernel function is the polynomial kernel

$$K_{ij} = \mathbf{f}_i^T \mathbf{f}_j = (\boldsymbol{\phi}_i^T \boldsymbol{\phi}_j)^q \quad (\text{C.18})$$

where  $q$  is a freely chosen exponent which determines the dimension  $F$  of the transformation  $\mathbf{f}(\boldsymbol{\phi})$ . For  $q = 1$ , kernel-PCA becomes linear PCA. By being able to solve the eigenproblem (C.17) for  $\boldsymbol{\beta}^i$ , the projection (C.16) of  $\mathbf{f}_k$  onto the eigenvectors  $\hat{\mathbf{V}}_R = [\hat{\mathbf{v}}^1, \dots, \hat{\mathbf{v}}^d]$  can be evaluated. The decision on the number  $d$  of principle components to be considered can be made analogously to (C.11) by substituting  $\sigma_i^2$  with  $\hat{\sigma}_i^2$ .

However, before the projection (C.16) is actually calculated, it has to be ensured that the previously made assumptions  $\hat{\mathbf{v}}^{iT} \hat{\mathbf{v}}^i = 1$  and  $\sum_{k=1}^N \mathbf{f}(\boldsymbol{\phi}_k) = \mathbf{0}$  are met in order to receive reasonable results. The normalization condition  $\hat{\mathbf{v}}^{iT} \hat{\mathbf{v}}^i = 1$  can be met by a specific normalization of  $\boldsymbol{\beta}^i$ . Multiplication of the normalization condition with  $\hat{\sigma}_i^4$  and proper



substitution gives the following results:

$$\begin{aligned}
1 = \hat{\mathbf{v}}^{iT} \hat{\mathbf{v}}^i &\rightarrow \hat{\sigma}_i^4 \equiv \hat{\sigma}_i^2 \hat{\mathbf{v}}^{iT} \hat{\sigma}_i^2 \hat{\mathbf{v}}^i \stackrel{\text{(C.15)}}{\equiv} \hat{\mathbf{v}}^{iT} \hat{\Sigma}^T \hat{\Sigma} \hat{\mathbf{v}}^i \stackrel{\text{(C.14)}}{\equiv} \frac{1}{(N-1)^2} \underbrace{\hat{\mathbf{v}}^{iT} \mathbf{F}}_{=\boldsymbol{\beta}^{iT}} \underbrace{\mathbf{F}^T \mathbf{F}}_{=K} \underbrace{\mathbf{F}^T \hat{\mathbf{v}}^i}_{=\boldsymbol{\beta}^i} \\
&\stackrel{\text{(C.17)}}{=} \frac{1}{(N-1) \hat{\sigma}_i^2} \boldsymbol{\beta}^i \boldsymbol{\beta}^{iT} \stackrel{\text{(C.19)}}{=} \frac{N-1}{\hat{\sigma}_i^2} .
\end{aligned} \tag{C.19}$$

Let  $\boldsymbol{\beta}^i$  be an arbitrary solution of (C.17), then the normalized vector

$$\boldsymbol{\beta}^{(i)} := \sqrt{\frac{N-1}{\hat{\sigma}_i^2}} \frac{\boldsymbol{\beta}^i}{\|\boldsymbol{\beta}^i\|} \text{ for } \hat{\sigma}_i^2 \neq 0 \tag{C.20}$$

will fulfill the required normalization condition (C.19). The restriction to non-zero eigenvalues  $\hat{\sigma}_i^2$  can be made, because directions  $\hat{\mathbf{v}}^i$  with zero variances are of no interest for the projection  $\boldsymbol{\beta}^i = \mathbf{F}^T \hat{\mathbf{v}}^i$  as discussed above in the context of Eq. (C.9). Anyway, due to Eq. (C.18), the symmetric kernel matrix  $\mathbf{K} = \mathbf{F}^T \mathbf{F}$  will only be rank deficient if redundant samples  $\boldsymbol{\phi}_i = \boldsymbol{\phi}_j$  for any  $i \neq j$  are present, which can easily be avoided by deleting such redundancies from the data set.

The validity of the second assumption  $\sum_k \mathbf{f}_k = \mathbf{0}$  can be ensured by including the required shift of the data points  $\mathbf{f}_k$  into the kernel matrix by defining a new kernel matrix  $\bar{K}_{ij} := \bar{\mathbf{f}}_i^T \bar{\mathbf{f}}_j$  based on the centered data points  $\bar{\mathbf{f}}_i := \mathbf{f}_i - (1/N) \sum_{k=1}^N \mathbf{f}_k$  or  $\bar{\mathbf{F}} := \mathbf{F} - \mathbf{F}\mathbf{J}/N$  where elements of  $\mathbf{J} \in \mathbb{R}^{N \times N}$  are all ones, i.e.,  $J_{ij} = 1 \forall i, j$ . Thus, a centered kernel matrix

$$\bar{\mathbf{K}} = \bar{\mathbf{F}}^T \bar{\mathbf{F}} = \mathbf{K} - \frac{\mathbf{J}\mathbf{K}}{N} - \frac{\mathbf{K}\mathbf{J}}{N} + \frac{\mathbf{J}\mathbf{K}\mathbf{J}}{N^2} \tag{C.21}$$

is calculated as a substitute for  $\mathbf{K}$  in Eq. (C.17) and the kPCA algorithm applied to a training set  $\mathcal{V}_T = \{\boldsymbol{\phi}_k\}$  then works as follows:

1. according to Eq. (C.18) calculate the kernel-matrix  $\mathbf{K}$  based on all  $\boldsymbol{\phi}_k \in \mathcal{V}_T$
2. according to Eq. (C.21) center the kernel-matrix
3. according to Eq. (C.17) evaluate the eigenvectors  $\boldsymbol{\beta}^i$  of  $\bar{\mathbf{K}}$
4. according to Eq. (C.20) normalize the eigenvectors  $\boldsymbol{\beta}^i$
5. according to Eq. (C.16) extract the projections  $\mathbf{f}'_k$

If the projection  $\bar{\mathbf{f}}' = \hat{\mathbf{V}}_R^T \bar{\mathbf{f}}(\boldsymbol{\phi})$  of a new data point  $\boldsymbol{\phi} \notin \mathcal{V}_T$  is explicitly required, a modification of Eq. (C.16) has to be found which provides the necessary information. First, from Eqns. (C.15) and (C.14) it is obtained that

$$\hat{\sigma}_i^2 \hat{\mathbf{v}}^i = \hat{\Sigma} \hat{\mathbf{v}}^i = \frac{1}{(N-1)} \mathbf{F} \underbrace{\mathbf{F}^T \hat{\mathbf{v}}^i}_{\boldsymbol{\beta}^i} = \frac{1}{(N-1)} \mathbf{F} \boldsymbol{\beta}^i \tag{C.22}$$

or

$$\hat{\mathbf{V}}_R = \mathbf{F}\mathbf{B}_R\mathbf{\Lambda}^+ \quad (\text{C.23})$$

where  $\mathbf{\Lambda}^+ := 1/(N-1) \text{diag}(1/\hat{\sigma}_1^2, \dots, 1/\hat{\sigma}_d^2)$  and  $d \leq r$ , where  $r$  is the rank of kernel matrix  $\bar{\mathbf{K}}$ , i.e.,  $\hat{\sigma}_i^2 \neq 0$  for  $i = 1, \dots, r$ . Remember that the  $\boldsymbol{\beta}^i$  in  $\mathbf{B}_R$  have to be normalized according to Eq. (C.20)). Now the projection  $\bar{\mathbf{f}}'$  of a data point  $\boldsymbol{\phi} \notin \mathcal{V}_T$  can be calculated as

$$\bar{\mathbf{f}}'^T = \bar{\mathbf{f}}(\boldsymbol{\phi})^T \hat{\mathbf{V}}_R = \bar{\mathbf{k}}_\phi^T \mathbf{B}_R \mathbf{\Lambda}^+ \quad (\text{C.24})$$

with  $\bar{\mathbf{k}}_\phi^T = \mathbf{k}_\phi^T - 1/N[1, \dots, 1]\mathbf{K} - 1/N\mathbf{k}_\phi^T\mathbf{J} + 1/N^2[1, \dots, 1]\mathbf{K}\mathbf{J}$  where  $\mathbf{k}_\phi^T = \mathbf{f}(\boldsymbol{\phi})^T \mathbf{F}$ .

## C.2 Multiple Discriminant Analysis

The multiple discriminant analysis (MDA) is the generalization of the Fisher-discriminant analysis (FDA) applied to multiple classes. Therefore, FDA will be explained first for matter of simplicity, and then the linear and nonlinear MDA will be described.

### C.2.1 Fisher-Discriminant Analysis

For a set of data points  $\boldsymbol{\phi}_k \in \mathbb{R}^D$ ,  $k = 1, \dots, N$ , with known classification  $c_k \in \{1, 2\}$  into two classes, FDA seeks to find the direction  $\mathbf{v} \in \mathbb{R}^D$  where a projection

$$\phi'_k = \mathbf{v}^T \boldsymbol{\phi}_k \quad (\text{C.25})$$

causes maximum separation between the two clusters  $\mathcal{V}_c = \{\boldsymbol{\phi}_k | c_k = c\}$ ,  $c = 1, 2$ , with mean vectors

$$\mathbf{m}_c = \frac{1}{|\mathcal{V}_c|} \sum_{\boldsymbol{\phi}_k \in \mathcal{V}_c} \boldsymbol{\phi}_k. \quad (\text{C.26})$$

The level of separation may be defined by the distance between the projected cluster means  $m'_c$  with respect to the sum of class scatters  $s_c'^2$  of the projected data samples, i.e.,

$$\frac{(m'_1 - m'_2)^2}{(s_1'^2 + s_2'^2)}, \quad (\text{C.27})$$

where

$$m'_c = \mathbf{v}^T \mathbf{m}_c \quad (\text{C.28})$$

and

$$\begin{aligned} s_c'^2 &= \sum_{\boldsymbol{\phi}_k \in \mathcal{V}_c} (\mathbf{v}^T \boldsymbol{\phi}_k - \mathbf{v}^T \mathbf{m}_c)^2 = \sum_{\boldsymbol{\phi}_k \in \mathcal{V}_c} (\mathbf{v}^T (\boldsymbol{\phi}_k - \mathbf{m}_c) \mathbf{v}^T (\boldsymbol{\phi}_k - \mathbf{m}_c)) \\ &= \sum_{\boldsymbol{\phi}_k \in \mathcal{V}_c} (\mathbf{v}^T (\boldsymbol{\phi}_k - \mathbf{m}_c) (\boldsymbol{\phi}_k - \mathbf{m}_c)^T \mathbf{v}) \\ &= \mathbf{v}^T \underbrace{\sum_{\boldsymbol{\phi}_k \in \mathcal{V}_c} ((\boldsymbol{\phi}_k - \mathbf{m}_c) (\boldsymbol{\phi}_k - \mathbf{m}_c)^T)}_{=:\boldsymbol{\Sigma}_c} \mathbf{v}. \end{aligned} \quad (\text{C.29})$$

Using Eq. (C.28) and Eq. (C.29), Eq. (C.27) may be written as

$$\frac{(m'_1 - m'_2)^2}{(s_1'^2 + s_2'^2)} = \frac{(\mathbf{v}^T \mathbf{m}_1 - \mathbf{v}^T \mathbf{m}_2)^2}{\mathbf{v}^T \left( \sum_{c=1}^2 \Sigma_c \right) \mathbf{v}} = \frac{\overbrace{\mathbf{v}^T (\mathbf{m}_1 - \mathbf{m}_2) (\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{v}}^{=: \Sigma_B}}{\underbrace{\mathbf{v}^T \left( \sum_{c=1}^2 \Sigma_c \right) \mathbf{v}}_{=: \Sigma_W}} = \frac{\mathbf{v}^T \Sigma_B \mathbf{v}}{\mathbf{v}^T \Sigma_W \mathbf{v}} \quad (\text{C.30})$$

where  $\Sigma_c$ ,  $\Sigma_B$ , and  $\Sigma_W$  are the symmetric class-scatter, between-class-scatter, and within-class-scatter matrices of  $\mathcal{V}_1$  and  $\mathcal{V}_2$ . Since the scaling of  $\mathbf{v}$  does not affect the separation measure (C.30), one may as well scale the solution to fulfill  $\mathbf{v}^T \Sigma_W \mathbf{v} = 1$ . The vector  $\mathbf{v}$  which maximizes (C.30) w.r.t. this constraint can be received from a vanishing derivative of the Lagrange function  $L$ :

$$\mathbf{0} \stackrel{!}{=} \frac{\partial L}{\partial \mathbf{v}} = \frac{\partial \left( \mathbf{v}^T \Sigma_B \mathbf{v} - \lambda \left( \mathbf{v}^T \Sigma_W \mathbf{v} - 1 \right) \right)}{\partial \mathbf{v}} = 2 \left( \Sigma_B \mathbf{v} - \lambda \Sigma_W \mathbf{v} \right) . \quad (\text{C.31})$$

The solutions are the eigenvectors  $\mathbf{v}$  and eigenvalues  $\lambda$  of the general eigenvalue problem

$$\Sigma_B \mathbf{v} = \lambda \Sigma_W \mathbf{v} . \quad (\text{C.32})$$

Substituting Eq. (C.32) into Eq. (C.30) shows that the ratio of between-class scatter to within-class scatter in direction of  $\mathbf{v}$  is equal to the eigenvalue  $\lambda$ . From definition (C.30) it can be seen that the dyadic product  $\Sigma_B$  has only rank one, which is why there is only one eigenvalue  $\lambda \neq 0$ . This is reasonable, because two clusters are maximally separated by one discriminant only.

## C.2.2 Linear MDA

According to Fukunaga (1990), the generalization of FDA to multiple classes  $N_C > 2$  may be accomplished by defining appropriate scatter matrices  $\Sigma_B$  and  $\Sigma_W$  for usage in Eq. (C.32). This is straight forward for the within-class scatter matrix in Eq. (C.30) which can be defined as

$$\Sigma_W := \sum_{c=1}^{N_C} \sum_{\boldsymbol{\phi}_k \in \mathcal{V}_c} \left( (\boldsymbol{\phi}_k - \mathbf{m}_c) (\boldsymbol{\phi}_k - \mathbf{m}_c)^T \right) . \quad (\text{C.33})$$

The formulation of the between-class scatter matrix  $\Sigma_B$ , however, first requires to realize that the overall mean vector  $\mathbf{m}$  can be expressed by the cluster means as follows:

$$\mathbf{m} := \frac{1}{|\mathcal{V}|} \sum_{\boldsymbol{\phi}_k \in \mathcal{V}} \boldsymbol{\phi}_k = \frac{1}{|\mathcal{V}|} \sum_{c=1}^{N_C} \sum_{\boldsymbol{\phi}_k \in \mathcal{V}_c} \boldsymbol{\phi}_k = \frac{1}{|\mathcal{V}|} \sum_{c=1}^{N_C} (|\mathcal{V}_c| \mathbf{m}_c) . \quad (\text{C.34})$$

Assuming only two clusters where  $|\mathcal{V}| := |\mathcal{V}_1| + |\mathcal{V}_2|$ ,  $\Sigma_B$  may be rewritten using definition (C.34) such that

$$\begin{aligned}
\Sigma_B &:= (\mathbf{m}_1 - \mathbf{m}_2) (\mathbf{m}_1 - \mathbf{m}_2)^T \\
&= \frac{1}{2} (\mathbf{m}_1 - \mathbf{m}_2) (\mathbf{m}_1 - \mathbf{m}_2)^T + \frac{1}{2} (\mathbf{m}_1 - \mathbf{m}_2) (\mathbf{m}_1 - \mathbf{m}_2)^T \\
&\equiv \frac{1}{2} (\mathbf{m}_1 - \mathbf{m}_2) (\mathbf{m}_1 - \mathbf{m}_2)^T + \frac{1}{2} (\mathbf{m}_2 - \mathbf{m}_1) (\mathbf{m}_2 - \mathbf{m}_1)^T \\
&= \frac{|\mathcal{V}|^2}{2|\mathcal{V}_2|^2} (\mathbf{m}_1 - \mathbf{m}) (\mathbf{m}_1 - \mathbf{m})^T + \frac{|\mathcal{V}|^2}{2|\mathcal{V}_1|^2} (\mathbf{m}_2 - \mathbf{m}) (\mathbf{m}_2 - \mathbf{m})^T \\
&= \sum_{c=1}^2 \frac{|\mathcal{V}|^2}{2(|\mathcal{V}| - |\mathcal{V}_c|)^2} \left( (\mathbf{m}_c - \mathbf{m}) (\mathbf{m}_c - \mathbf{m})^T \right) .
\end{aligned} \tag{C.35}$$

In case of three clusters, calculating the overall between-class scatter matrix as the sum of the between-class scatter matrices (C.35) between the first and second, second and third, and third and first cluster, and defining  $\mathcal{V} := \bigcup_{c=1}^{N_C} \mathcal{V}_c$ , makes it apparent that Eq. (C.35) can be generalized to multiple clusters as

$$\Sigma_B := \sum_{c=1}^{N_C} \left( \frac{|\mathcal{V}|^2}{2(|\mathcal{V}| - |\mathcal{V}_c|)^2} (\mathbf{m}_c - \mathbf{m}) (\mathbf{m}_c - \mathbf{m})^T \right) . \tag{C.36}$$

In accordance to Eqns. (C.30)–(C.32) maximal ratios of between-class scatter to within-class scatter can be obtained by solving the general eigenvalue problem

$$\Sigma_B \mathbf{V} = \Sigma_W \mathbf{V} \Lambda \quad \text{or} \quad \Sigma_B \mathbf{v}^i = \lambda_i \Sigma_W \mathbf{v}^i \tag{C.37}$$

where  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_D)$  with  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_D$  and  $\mathbf{V} = [\mathbf{v}^1, \dots, \mathbf{v}^D]$ .

In a reasonable application,  $\Sigma_B$  is a combination of  $N_C$  linearly independent cluster means  $\mathbf{m}_c$ , while  $\mathbf{m}$  is a linear combination of all  $\mathbf{m}_c$  (Eq. (C.34)). Therefore, the between-class-scatter matrix  $\Sigma_B$  has the rank  $r = N_C - 1$  with only the first  $N_C - 1$  eigenvalues being nonzero. Substituting Eq. (C.37) into Eq. (C.30) gives  $\Lambda$ , hence, the ratio (C.30) in direction of  $\mathbf{v}^i$  is equal to  $\lambda_i$ . This means that only for the largest first  $N_C - 1$  eigenvalues  $\lambda_i \neq 0$ , the corresponding eigenvectors  $\mathbf{V}_R = [\mathbf{v}^1, \dots, \mathbf{v}^{N_C-1}]$  will maximize Eq. (C.30). Thus, only those eigenvectors need to be considered for dimension reduction of data. The projection of a data point  $\boldsymbol{\phi}$  can be received as

$$\boldsymbol{\phi}'_R = \mathbf{V}_R^T \boldsymbol{\phi} . \tag{C.38}$$

If  $\Sigma_W$  is calculated from a reference set  $\mathcal{V}$  with dimensionality  $D$  of the samples  $\boldsymbol{\phi}_k \in \mathcal{V}$  larger than the number of samples  $|\mathcal{V}|$  (this is the case for the reference eigenmode set used

within this thesis), the problem (C.37) becomes ill-posed ( $\Sigma_W$  has not full rank and  $\Sigma_W^{-1}$  may not exist). It is, therefore, common practice to make  $\Sigma_W$  invertible by adding a small value  $\varepsilon$  to the main diagonal (Friedman (1989)) such that  $\Sigma_W + \varepsilon \mathbf{I}$  becomes non-singular and a sufficiently accurate estimation  $\mathbf{V}^\varepsilon$  of the solution to (C.37) is received by solving the new eigenproblem

$$(\Sigma_W + \varepsilon \mathbf{I})^{-1} \Sigma_B \mathbf{V}^\varepsilon = \mathbf{V}^\varepsilon \Lambda^\varepsilon . \quad (\text{C.39})$$

Throughout this thesis  $\varepsilon = 10^{-13}$  has proven to be sufficient. Alternatively, Howland et al. (2003) developed a method to solve Eq. (C.37) without inverting  $\Sigma_W$  by using generalized singular-value decomposition.

### C.2.3 Nonlinear MDA

Searching for linear discriminating directions is not sufficient in many applications, which is why Mika et al. (1999) introduced a method of finding nonlinear directions by implementation of the kernel trick. First, a nonlinear transformation function  $\mathbf{f}(\boldsymbol{\phi})$  is defined which transforms  $\boldsymbol{\phi} \in \mathbb{R}^D$  into a feature space of dimension  $F > D$  analogues to Eq. (C.13). In accordance to Eqns. (C.36) and (C.33) one can then calculate

$$\hat{\Sigma}_B := \sum_{c=1}^{N_C} \left( \frac{|\mathcal{V}|^2}{2(|\mathcal{V}| - |\mathcal{V}_c|)^2} (\hat{\mathbf{m}}_c - \hat{\mathbf{m}}) (\hat{\mathbf{m}}_c - \hat{\mathbf{m}})^T \right), \quad (\text{C.40})$$

$$\hat{\Sigma}_W := \sum_{c=1}^{N_C} \sum_{k:\boldsymbol{\phi}_k \in \mathcal{V}_c} \left( (\mathbf{f}_k - \hat{\mathbf{m}}_c) (\mathbf{f}_k - \hat{\mathbf{m}}_c)^T \right) \quad (\text{C.41})$$

where  $\mathbf{f}_k := \mathbf{f}(\boldsymbol{\phi}_k)$  and

$$\hat{\mathbf{m}} := \frac{1}{|\mathcal{V}|} \sum_{k:\boldsymbol{\phi}_k \in \mathcal{V}} \mathbf{f}_k, \quad \hat{\mathbf{m}}_c := \frac{1}{|\mathcal{V}_c|} \sum_{k:\boldsymbol{\phi}_k \in \mathcal{V}_c} \mathbf{f}_k . \quad (\text{C.42})$$

Now, best linear discriminants may be found by calculating the extrema of

$$\frac{\hat{\mathbf{v}}^{iT} \hat{\Sigma}_B \hat{\mathbf{v}}^i}{\hat{\mathbf{v}}^{iT} \hat{\Sigma}_W \hat{\mathbf{v}}^i} \quad (\text{C.43})$$

in accordance to Eq. (C.30). Similar to Eq. (C.37), the solution is obtained by solving the eigenvalue problem

$$\hat{\Sigma}_B \hat{\mathbf{V}} = \hat{\Sigma}_W \hat{\mathbf{V}} \hat{\Lambda}, \quad \hat{\mathbf{V}} := [\hat{\mathbf{v}}^1, \dots, \hat{\mathbf{v}}^F], \quad (\text{C.44})$$

which may be impossible for large or even infinite  $F$ . As a work-around the kernel trick (as introduced in Section C.1.2) is implemented. From Aronszajn (1950) it is known that

the solution must lie in the span of  $\{\mathbf{f}_k\}$ . Thus, there are coefficients  $\alpha_k^i$  such that

$$\hat{\mathbf{v}}^i = \sum_{k=1}^N (\alpha_k^i \mathbf{f}_k) = \mathbf{F} \boldsymbol{\alpha}^i \rightarrow \hat{\mathbf{V}} = \mathbf{F} \mathbf{A} \quad (\text{C.45})$$

where  $\mathbf{F} := [\mathbf{f}_1, \dots, \mathbf{f}_N]$  and  $\mathbf{A} := [\boldsymbol{\alpha}^1, \dots, \boldsymbol{\alpha}^N]$ . Substituting Eq. (C.45) into Eq. (C.43) gives

$$\frac{\overbrace{\boldsymbol{\alpha}^{iT} \mathbf{F}^T \hat{\boldsymbol{\Sigma}}_B \mathbf{F} \boldsymbol{\alpha}^i}^{=: \hat{\boldsymbol{\Sigma}}_B^*}}{\underbrace{\boldsymbol{\alpha}^{iT} \mathbf{F}^T \hat{\boldsymbol{\Sigma}}_W \mathbf{F} \boldsymbol{\alpha}^i}_{=: \hat{\boldsymbol{\Sigma}}_W^*}}. \quad (\text{C.46})$$

Hence,

$$\begin{aligned} \hat{\boldsymbol{\Sigma}}_B^* &\stackrel{(\text{C.40})}{=} \mathbf{F}^T \left( \sum_{c=1}^{N_C} \left( \frac{|\mathcal{V}|^2}{2(|\mathcal{V}| - |\mathcal{V}_c|)^2} (\hat{\mathbf{m}}_c - \hat{\mathbf{m}}) (\hat{\mathbf{m}}_c - \hat{\mathbf{m}})^T \right) \right) \mathbf{F} \\ &= \sum_{c=1}^{N_C} \left( \frac{|\mathcal{V}|^2}{2(|\mathcal{V}| - |\mathcal{V}_c|)^2} (\mathbf{F}^T \hat{\mathbf{m}}_c - \mathbf{F}^T \hat{\mathbf{m}}) (\mathbf{F}^T \hat{\mathbf{m}}_c - \mathbf{F}^T \hat{\mathbf{m}})^T \right). \end{aligned} \quad (\text{C.47})$$

By using definition (C.18) of the kernel matrix and defining its  $k^{\text{th}}$  column as  $[\mathbf{K}]^k$ , one can write

$$\mathbf{F}^T \hat{\mathbf{m}}_c \stackrel{(\text{C.42})}{=} \frac{1}{|\mathcal{V}_c|} \sum_{k: \boldsymbol{\phi}_k \in \mathcal{V}_c} \underbrace{\mathbf{F}^T \mathbf{f}_k}_{[\mathbf{K}]^k} \quad \text{and} \quad \mathbf{F}^T \hat{\mathbf{m}} \stackrel{(\text{C.42})}{=} \frac{1}{|\mathcal{V}|} \sum_{k: \boldsymbol{\phi}_k \in \mathcal{V}} \mathbf{F}^T \mathbf{f}_k = \frac{1}{|\mathcal{V}|} \sum_{k: \boldsymbol{\phi}_k \in \mathcal{V}} [\mathbf{K}]^k. \quad (\text{C.48})$$

Using this allows rewriting the between-cluster-scatter matrix Eq. C.47 as

$$\begin{aligned} \hat{\boldsymbol{\Sigma}}_B^* &= \sum_{c=1}^{N_C} \left( \left( \frac{|\mathcal{V}|^2}{2|\mathcal{V}_c|(|\mathcal{V}| - |\mathcal{V}_c|)^2} \sum_{k: \boldsymbol{\phi}_k \in \mathcal{V}_c} [\mathbf{K}]^k - \frac{|\mathcal{V}|}{2(|\mathcal{V}| - |\mathcal{V}_c|)^2} \sum_{k: \boldsymbol{\phi}_k \in \mathcal{V}} [\mathbf{K}]^k \right) \right. \\ &\quad \left. \left( \frac{|\mathcal{V}|^2}{2|\mathcal{V}_c|(|\mathcal{V}| - |\mathcal{V}_c|)^2} \sum_{k: \boldsymbol{\phi}_k \in \mathcal{V}_c} [\mathbf{K}]^k - \frac{|\mathcal{V}|}{2(|\mathcal{V}| - |\mathcal{V}_c|)^2} \sum_{k: \boldsymbol{\phi}_k \in \mathcal{V}} [\mathbf{K}]^k \right)^T \right). \end{aligned} \quad (\text{C.49})$$

Accordingly, the within-class-scatter matrix can be written as

$$\begin{aligned} \hat{\boldsymbol{\Sigma}}_W^* &\stackrel{(\text{C.41})}{=} \mathbf{F}^T \sum_{c=1}^{N_C} \sum_{\boldsymbol{\phi}_k \in \mathcal{V}_c} ((\mathbf{f}_k - \hat{\mathbf{m}}_c) (\mathbf{f}_k - \hat{\mathbf{m}}_c)^T) \mathbf{F} \\ &= \sum_{c=1}^{N_C} \sum_{\boldsymbol{\phi}_k \in \mathcal{V}_c} \left( \left( [\mathbf{K}]^k - \frac{1}{|\mathcal{V}_c|} \sum_{\boldsymbol{\phi}_k \in \mathcal{V}_c} [\mathbf{K}]^k \right) \left( [\mathbf{K}]^k - \frac{1}{|\mathcal{V}_c|} \sum_{\boldsymbol{\phi}_k \in \mathcal{V}_c} [\mathbf{K}]^k \right)^T \right). \end{aligned} \quad (\text{C.50})$$

Now, as explained in Section C.1.2, knowing  $\mathbf{f}$  is not required, instead, it is implicitly defined using a kernel function (C.18) for the scalar product  $\mathbf{f}^T \mathbf{f}$ . Therefore, the quality of the discriminates  $\hat{\mathbf{V}}$  will depend on the suitability of the kernel function w.r.t. to the data structure.

Due to Eq. (C.46), the eigenproblem to be solved changes from (C.44) to

$$\hat{\Sigma}_B^* \mathbf{A} = \hat{\Sigma}_W^* \mathbf{A} \hat{\Lambda} . \quad (\text{C.51})$$

The approach of Howland et al. (2003) to solve Eq. (C.37) without having to invert  $\Sigma_W^*$  by using generalized singular-value decomposition can also be applied to solve Eq. (C.51) as suggested by Park and Park (2005). Else, as for MDA,  $\hat{\Sigma}_W^*$  would have to be made non-singular by adding a small value  $\varepsilon$  to its main diagonal resulting in

$$\left( \hat{\Sigma}_W^* + \varepsilon \mathbf{I} \right)^{-1} \hat{\Sigma}_B^* \mathbf{A}^\varepsilon = \mathbf{A}^\varepsilon \hat{\Lambda}^\varepsilon . \quad (\text{C.52})$$

The desired projection of a data point  $\boldsymbol{\phi}$  is calculated as

$$\boldsymbol{\phi}'_R = \hat{\mathbf{V}}^T \underbrace{\mathbf{f}(\boldsymbol{\phi})}_{\mathbf{f}_\phi} \stackrel{(\text{C.45})}{=} (\mathbf{F}\mathbf{A})^T \mathbf{f}_\phi = \mathbf{A}^T \underbrace{\mathbf{F}^T \mathbf{f}_\phi}_{\mathbf{k}_\phi} = \mathbf{A}^T \mathbf{k}_\phi \quad (\text{C.53})$$

where  $[\mathbf{k}_\phi]_i = \left( \boldsymbol{\phi}_i^T \boldsymbol{\phi} \right)^q$  refers to the  $i^{\text{th}}$  element of  $\mathbf{k}_\phi$  in accordance to Eq. (C.18).



### C.3 Modal-Assurance Criterion

At various points in this thesis the modal-assurance criterion (MAC) is used to measure the correlation between two vectors. Introduced by Allemang and Brown (1982), it was intended as a quality measure for experimental eigenvectors originating from frequency-response functions, but became popular for validating dynamic FE-models through experimentally measured eigenmodes (Allemang (2002)). Today, commercial-software tools such as Abaqus use MAC for eigenmode tracking during automated design optimization. In order to assess the necessity for re-performing flutter analysis based on the similarity between two designs w.r.t. their eigenmode shapes, Blocher and Aulich (2013) used MAC during an automatized design process.

Let  $\boldsymbol{\phi}_i$  and  $\boldsymbol{\phi}_j$  be two non-zero vectors; then the MAC-value

$$MAC(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) := \frac{(\boldsymbol{\phi}_i^T \boldsymbol{\phi}_j)^2}{\|\boldsymbol{\phi}_i\|^2 \|\boldsymbol{\phi}_j\|^2} = \frac{[\|\boldsymbol{\phi}_i\| \|\boldsymbol{\phi}_j\| \cos(\angle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j)]^2}{\|\boldsymbol{\phi}_i\|^2 \|\boldsymbol{\phi}_j\|^2} = \cos^2(\angle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j) \in [0, 1] \quad (\text{C.54})$$

is a bounded measure of the linear correlation between these two vectors. As shown by Morales (2005) and Eq. (C.54), MAC is simply the squared cosine of the angle between the two vectors. Thus, the coefficient is equal to one, if the two vectors only differ in length but have the same or opposite orientation, and is equal to zero if they are orthogonal. Hence, higher values indicate better agreement between the two eigenvectors being compared. In Appendix B it was shown that the direction of eigenvectors is the distinguishing feature, not the length, which is why MAC is a suitable measure of mode agreement. However, MAC has been criticized for mainly three reasons which gave motivation to the development of numerous other correlation criteria (Heylen and Avitabile (1998), Morales (2005), Allemang (2002)).

One issue is that under strict consideration MAC is no orthogonality measure in the context of structure analysis, because eigenmodes of structures are typically orthogonal w.r.t. the mass matrix  $\mathbf{M}$  (or stiffness matrix  $\mathbf{K}^{L/NL}$ ), i.e.,  $\boldsymbol{\phi}_i^T \mathbf{M} \boldsymbol{\phi}_j = 0$  for  $i \neq j$ . To prove this, the symmetric and positive definite mass matrix may be decomposed by Cholesky decomposition as

$$\mathbf{M} = \mathbf{L}\mathbf{L}^T \quad (\text{C.55})$$

where  $\mathbf{L}$  is a nonsingular lower triangular matrix. Then the problem (3.9), i.e.,

$$\mathbf{K}^{L/NL} \boldsymbol{\phi}_i = \omega_i^2 \mathbf{M} \boldsymbol{\phi}_i \quad (\text{C.56})$$

can be rewritten as

$$\mathbf{L}^{-1} \mathbf{K}^{L/NL} \mathbf{L}^{-T} \mathbf{v}_i = \omega_i^2 \mathbf{v}_i, \quad (\text{C.57})$$

where the eigenvectors  $\mathbf{v}_i := \mathbf{L}^T \boldsymbol{\phi}_i$  are orthogonal due to symmetry of  $\mathbf{L}^{-1} \mathbf{K}^{L/NL} \mathbf{L}^{-T}$ , i.e.,

$$\mathbf{v}_i^T \mathbf{v}_j = 0 = \boldsymbol{\phi}_i^T \mathbf{L} \mathbf{L}^T \boldsymbol{\phi}_j = \boldsymbol{\phi}_i^T \mathbf{M} \boldsymbol{\phi}_j \text{ for } i \neq j . \quad (\text{C.58})$$

This might be an issue when calibrating numeric models with experimental results, but not for the purpose of measuring mode agreement within this thesis. Anyway, calculation of correlations between eigenmodes taking into account the mass matrix is not possible if eigenmodes of different structures with different mass matrices shall be compared.

MAC is also often criticized for its nonlinearity of representing the squared cosine between two vectors. Therefore, Morales (2005) defined a linear MAC (LMAC) as

$$LMAC(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) := 1 - \frac{2}{\pi} \arccos \frac{|\boldsymbol{\phi}_i^T \boldsymbol{\phi}_j|}{\|\boldsymbol{\phi}_i\| \|\boldsymbol{\phi}_j\|} \in [0, 1] . \quad (\text{C.59})$$

However, the usage of LMAC over MAC has only little influence on the relative scaling of some plots within this thesis and none onto classification performance. Therefore, for matter of simplicity, eigenvector correlations are measured with MAC throughout this thesis.

## C.4 Support-Vector Machines

In general, a support-vector machine (SVM), first contemplated by Vapnik and Lerner (1963) and refined by Boser et al. (1992) and Cortes and Vapnik (1995), is a binary classifier which maximizes the margin  $\varepsilon$  of a linear discriminate  $\mathcal{H}$  (hyperplane) between two data sets  $\mathcal{V}_c$ ,  $c = \{1, 2\}$ , of vectors  $\boldsymbol{\phi}_k$  with classification  $c_k$  where  $\boldsymbol{\phi}_k \in \mathcal{V}_{c_k}$ . The hyperplane can be defined as

$$\mathcal{H} : \mathbf{n}^T (\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}^*) = 0 \quad (\text{C.60})$$

where  $\tilde{\boldsymbol{\phi}}$  is an arbitrary data vector,  $\boldsymbol{\phi}^*$  is a point on the hyperplane ( $\boldsymbol{\phi}^* \in \mathcal{H}$ ), and  $\mathbf{n}$  is a nonzero vector perpendicular to  $\mathcal{H}$ . Rewriting Eq. (C.60) as

$$\mathcal{H} : \mathbf{n}^T \tilde{\boldsymbol{\phi}} - \underbrace{\mathbf{n}^T \boldsymbol{\phi}^*}_{=:b} = 0 \quad (\text{C.61})$$

and choosing  $\mathbf{n}$  and  $b$  in such a way that

$$\mathbf{n}^T \boldsymbol{\phi}_k + b = \begin{cases} > 0 & \text{for } \boldsymbol{\phi}_k \in \mathcal{V}_1, \\ < 0 & \text{for } \boldsymbol{\phi}_k \in \mathcal{V}_2, \end{cases} \quad (\text{C.62})$$

gives a discriminate that allows to assign a class  $\tilde{c}$  to any unknown data point  $\tilde{\boldsymbol{\phi}}$  by the sign of Eq. (C.62) when substituting  $\tilde{\boldsymbol{\phi}}$  for  $\boldsymbol{\phi}_k$ . However, since the choice of  $\mathbf{n}$  and  $b$  is not unique, i.e., there exist multiple possible hyperplanes  $\mathcal{H}$  separating the data, an optimal setting is defined as the one which maximizes the margin

$$\zeta := \min_{\boldsymbol{\phi}_k \in \mathcal{V}_1} \frac{|\mathbf{n}^T \boldsymbol{\phi}_k + b|}{|\mathbf{n}|} + \min_{\boldsymbol{\phi}_k \in \mathcal{V}_2} \frac{|\mathbf{n}^T \boldsymbol{\phi}_k + b|}{|\mathbf{n}|}. \quad (\text{C.63})$$

Here, Eq. (C.62) is normalized by  $|\mathbf{n}|$  in order to provide the true distance of  $\boldsymbol{\phi}_k$  from  $\mathcal{H}$ . The margin is defined by the two points closest to  $\mathcal{H}$  on both sides of  $\mathcal{H}$ , respectively. Additionally both points shall have the same distance to  $\mathcal{H}$  which, by proper scaling of  $\mathbf{n}$ , results in the condition

$$\min_{k:\boldsymbol{\phi}_k \in \mathcal{V}_1} |\mathbf{n}^T \boldsymbol{\phi}_k + b| = \min_{k:\boldsymbol{\phi}_k \in \mathcal{V}_2} |\mathbf{n}^T \boldsymbol{\phi}_k + b| \stackrel{!}{=} 1. \quad (\text{C.64})$$

The margin (C.63) then becomes

$$\zeta := \frac{2}{|\mathbf{n}|} \quad (\text{C.65})$$

as shown in Fig. 6.2. Hence, maximizing the margin  $\zeta$  requires minimizing  $|\mathbf{n}|$  subject to the constraint (C.62). By defining a class identifier  $y_k \in \{-1, 1\}$  for the training data  $\boldsymbol{\phi}_k \in \mathcal{V}_1 \cup \mathcal{V}_2$  (e.g.  $-1$  for  $\boldsymbol{\phi}_k \in \mathcal{V}_1$  and  $1$  for  $\boldsymbol{\phi}_k \in \mathcal{V}_2$ ) and due to Eq. (C.64), the constraint

(C.62) for linearly separable data may be rewritten as

$$y_k (\mathbf{n}^T \boldsymbol{\phi}_k + b) \geq 1 . \quad (\text{C.66})$$

Thus, the problem of finding optimal  $\mathbf{n}$  and  $b$  for linearly separable data may be stated as

$$\min_{\mathbf{n}, b} \|\mathbf{n}\| \hat{=} \min_{\mathbf{n}, b} \left( \frac{1}{2} \|\mathbf{n}\|^2 \right) = \min_{\mathbf{n}, b} \left( \frac{1}{2} \mathbf{n}^T \mathbf{n} \right) \text{ s.t. } y_k (\mathbf{n}^T \boldsymbol{\phi}_k + b) \geq 1 \forall k . \quad (\text{C.67})$$

However, in case two clusters of data with different classification overlap or if they are not linearly separable, problem (C.67) has no solution. Consequently, violation of the constraint (C.66) has to be allowed and minimizing the degree of violation has to be included into the problem definition (C.67). This is usually achieved by relaxing the constraint by introducing a slack variable  $\xi_k \geq 0$ , i.e.,

$$y_k (\mathbf{n}^T \boldsymbol{\phi}_k + b) \geq 1 - \xi_k , \quad (\text{C.68})$$

and redefining the problem (C.67) as

$$\min_{\mathbf{n}, b, \xi_k} \left( \frac{1}{2} \mathbf{n}^T \mathbf{n} + C \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} \xi_k \right) \text{ s.t. } (1 - \xi_k) - y_k (\mathbf{n}^T \boldsymbol{\phi}_k + b) \leq 0, \xi_k \geq 0 , \quad (\text{C.69})$$

where  $C$  is a weight parameter controlling the trade-off between increasing the size of the margin (C.65) and reducing the degree of violating Eq. (C.66). Note, that if the data are linearly separable, problem (C.69) becomes (C.67) by  $\xi_k = 0$ . However, if data are not linearly separable, the optimal slack  $\xi_k$  for any  $\boldsymbol{\phi}_k$  violating constraint (C.66) will be just as large to fulfill constraint (C.68) in order to increase the objective in (C.69) as little as possible. Thus, the optimal slacks would be

$$\xi_k^* := \max \{0, 1 - y_k (\mathbf{n}^T \boldsymbol{\phi}_k + b)\} \geq 0 \quad (\text{C.70})$$

The solution of problem (C.69) can be improved by allowing for nonlinear hyperplanes  $\mathcal{H}$ . This is achieved by using the kernel trick (similar to Appendix C.1). Therefore, formulation (C.69) has to be modified in a way that only scalar products of  $\boldsymbol{\phi}_k$  occur. First, the Lagrangian of the problem (C.69) is written as

$$\begin{aligned} L(\mathbf{n}, b, \xi_k, \alpha_k, \gamma_k) := & \frac{1}{2} \mathbf{n}^T \mathbf{n} + C \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} \xi_k + \\ & \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} \left( \alpha_k \left( (1 - \xi_k) - y_k (\mathbf{n}^T \boldsymbol{\phi}_k + b) \right) \right) + \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} \left( \gamma_k (-\xi_k) \right), \end{aligned} \quad (\text{C.71})$$

where  $\alpha_k, \gamma_k \geq 0$  are Karush-Kuhn-Tucker multipliers (KKT multipliers; Lagrange multi-

pliers apply to equality constraints only). As shown by (Kuhn and Tucker (1951)), the optimal solution to the convex problem (C.69) is at the minimum of the Lagrangian (C.71) w.r.t. to the primal variables  $\mathbf{n}$ ,  $b$ ,  $\xi_k$ , and at the maximum w.r.t. the KKT multipliers. Thus, for the primal variables the solution has to fulfill following conditions:

$$\frac{\partial L}{\partial \mathbf{n}} = \mathbf{0} = \mathbf{n} - \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_k y_k \boldsymbol{\phi}_k) \quad (\text{C.72})$$

$$\frac{\partial L}{\partial b} = 0 = - \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_k y_k) \quad (\text{C.73})$$

$$\frac{\partial L}{\partial \xi_k} = 0 = C - \alpha_k - \gamma_k . \quad (\text{C.74})$$

These stationary conditions are used to modify parts of Eq. (C.71) as follows:

$$\frac{1}{2} \mathbf{n}^T \mathbf{n} \stackrel{(\text{C.72})}{=} \frac{1}{2} \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_k y_k \boldsymbol{\phi}_k^T) \sum_{l=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_l y_l \boldsymbol{\phi}_l) = \frac{1}{2} \sum_{k,l=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_k \alpha_l y_k y_l \boldsymbol{\phi}_k^T \boldsymbol{\phi}_l) , \quad (\text{C.75})$$

$$\sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_k y_k \mathbf{n}^T \boldsymbol{\phi}_k) \equiv \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_k y_k \boldsymbol{\phi}_k^T) \mathbf{n} \stackrel{(\text{C.72})}{=} \sum_{k,l=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_k \alpha_l y_k y_l \boldsymbol{\phi}_k^T \boldsymbol{\phi}_l) , \quad (\text{C.76})$$

$$\sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_k y_k b) = b \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_k y_k) \stackrel{(\text{C.73})}{=} 0 , \quad (\text{C.77})$$

$$C \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} \xi_k - \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_k \xi_k) - \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\gamma_k \xi_k) = \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} ((C - \alpha_k - \gamma_k) \xi_k) \stackrel{(\text{C.74})}{=} 0 . \quad (\text{C.78})$$

Thus, applying the stationary conditions (C.72)–(C.74) to Eq. (C.71) gives the dual formulation

$$L_D(\alpha_k) := \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} \alpha_k - \frac{1}{2} \sum_{k,l=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_k \alpha_l y_k y_l \boldsymbol{\phi}_k^T \boldsymbol{\phi}_l) . \quad (\text{C.79})$$

Since this function is negative quadratic in  $\alpha_k$ , the optimal solution  $\alpha_k^*$  can be received by maximizing (C.79) w.r.t. the constraints (C.73) and  $0 \leq \alpha_k \leq C$ , where the latter is received from  $\alpha_k, \gamma_k \geq 0$  and (C.74). Thus, the final problem to be solved is the dual

$$\max_{\alpha_k} L_D(\alpha_k) \text{ s.t. } 0 \leq \alpha_k \leq C, \quad \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_k y_k) = 0 . \quad (\text{C.80})$$

This may be done by using e.g. gradient-based optimization algorithms, since the problem is convex, i.e., a local maximum must be the global one. This is the reason why SVM often outperforms other classifiers such as artificial-neural networks which easily get stuck in local minima.

The optimal solution to problem (C.80) will fulfill complementary condition (Kuhn and Tucker (1951))

$$\alpha_k \left( (1 - \xi_k) - y_k (\mathbf{n}^T \boldsymbol{\phi}_k + b) \right) = 0 . \quad (\text{C.81})$$

Therefore, and due to Eq. (C.70), all points  $\boldsymbol{\phi}_k$  which violate Eq. (C.68) or are located on the margin bounds ( $|\mathbf{n}^T \boldsymbol{\phi}_k + b| = 1$ ) will cause the parenthesis to become zero and  $\alpha_k \neq 0$ , and visa versa. Thus, only those  $\boldsymbol{\phi}_k$  for which  $\alpha_k^* \neq 0$  will influence the solution to (C.80), hence, the orientation  $\mathbf{n}$  of the discriminate  $\mathcal{H}$ , see Eq. (C.72). These vectors are therefore referred to as support vectors.

In order to calculate  $\mathcal{H}$  from the solution to (C.80), Eq. (C.72) is used to receive  $\mathbf{n}^*$ . The offset  $b$  is evaluated by identifying two vectors  $\boldsymbol{\phi}_{i^\circ, j^\circ}$  from the set of support vectors  $\{\boldsymbol{\phi}_k | \alpha_k^* \neq 0\}$  for which

$$\{i^\circ, j^\circ\} := \arg \max_{(i,j): \alpha_i \neq 0, \alpha_j \neq 0} \mathbf{n}^{*T} (\boldsymbol{\phi}_i - \boldsymbol{\phi}_j) . \quad (\text{C.82})$$

This simply means that the two identified vectors belong to different classes and fulfill  $y_{i^\circ} (\mathbf{n}^{*T} \boldsymbol{\phi}_{i^\circ} + b^*) = y_{j^\circ} (\mathbf{n}^{*T} \boldsymbol{\phi}_{j^\circ} + b^*) = 1$  (see Eq. (C.66)) which can be used to calculate the optimal shift  $b^*$  (Boser et al. (1992)) as follows:

$$b^* = -\frac{1}{2} (\mathbf{n}^{*T} (\boldsymbol{\phi}_{i^\circ} + \boldsymbol{\phi}_{j^\circ})) \quad (\text{C.83})$$

$$\stackrel{(\text{C.72})}{=} -\frac{1}{2} \sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} (\alpha_k^* y_k (\boldsymbol{\phi}_k^T \boldsymbol{\phi}_{i^\circ} + \boldsymbol{\phi}_k^T \boldsymbol{\phi}_{j^\circ})) . \quad (\text{C.84})$$

Putting it all together, and in accordance to Eq. (C.62), the decision on the classification of an unassigned data point  $\tilde{\boldsymbol{\phi}}$  can be made on

$$\mathbf{n}^{*T} \tilde{\boldsymbol{\phi}} + b^* = \begin{cases} > 0 & \Rightarrow \tilde{c} = 1 , \\ < 0 & \Rightarrow \tilde{c} = 2 . \end{cases} \quad (\text{C.85})$$

The relevant equation (C.79) contains only scalar products of the data. As described in the context of nonlinear PCA (Appendix C.1), also SVM can account for nonlinear directions (hyperplanes) by transforming the data into a higher dimensional space  $\mathbb{R}^F$  (infinite dimensional possibly), see Eq. (C.13), where the data might be linearly separable (Boser et al. (1992)). In order to avoid the costly calculation of high-dimensional scalar products, the kernel trick is implemented (Schölkopf and Smola (2002)), where it is assumed that high-dimensional scalar products can be calculated as a functional of the

scalar products in the original space. The functional has to fulfill the Mercer's condition (Mercer (1909)) therefor and is commonly called a kernel. An example of such a kernel is the polynomial kernel (C.18) where the exponent  $q$  can be chosen freely, deciding on the dimension and shape of the transformation (C.13). In conclusion, by substituting each scalar product in Eqns. (C.79) and (C.86) with a kernel, e.g., Eq. (C.18), the linear SVM-classification algorithm is able to account for nonlinear hyperplanes  $\mathcal{H}$ , see Fig. 6.2c. In order to assign a class to a new data point  $\tilde{\boldsymbol{\phi}}$ , Eq. (C.85) has to be rewritten in a form using scalar products of data only, by substituting Eq. (C.72) and Eq. (C.84) such that the class to be assigned is determined by the sign of

$$\sum_{k=1}^{|\mathcal{V}_1 \cup \mathcal{V}_2|} \left( \alpha_k^* y_k \left( \boldsymbol{\phi}_k^T \tilde{\boldsymbol{\phi}} - \frac{1}{2} \left( \boldsymbol{\phi}_k^T \boldsymbol{\phi}_{i^\circ} + \boldsymbol{\phi}_k^T \boldsymbol{\phi}_{j^\circ} \right) \right) \right) . \quad (\text{C.86})$$

## C.5 Training of a Three-Layer Feed-Forward Neural Network

In the following, training of a three-layer FFNN as shown in Fig. 6.3 and described in Section 6.1.3 will be explained. For adapting the weights of a FFNN via supervised learning, a reference set  $\mathcal{V}$  of samples  $\boldsymbol{\phi}_k$  with known classification  $c_k$  is required. However, in order to prevent overfitting, the network weights are adapted based on a smaller training set  $\mathcal{V}_T \subset \mathcal{V}$  and training stops, i.a., when the performance of the network does not improve on a validation set  $\mathcal{V}_V \subseteq \mathcal{V} \setminus \mathcal{V}_T$  any further. The training may be performed either sample-wise or batch-wise where the latter usually gives better performance. For batch-wise training the samples in  $\mathcal{V}_T$  are all presented to the network before the performance of this batch is calculated and network weights are adapted accordingly. The performance of a neural network in the field of classification is commonly measured using the cross-entropy

$$E^S(\mathcal{V}_T) := -\frac{1}{|\mathcal{V}_T|} \sum_{k|\boldsymbol{\phi}_k \in \mathcal{V}_T} \sum_{c=1}^{N_C} (\delta_{cc_k} \ln(g_c)) = -\frac{1}{|\mathcal{V}_T|} \sum_{k|\boldsymbol{\phi}_k \in \mathcal{V}_T} \ln(g_{c_k}) , \quad (\text{C.87})$$

which may be seen as a measure for the confidence in the correct classification of reference set (Hagan et al. (1996)). Unless the mean-squared-error-performance function<sup>1</sup> it does not assess the distance between the target vector  $[\delta_{1c_k}, \dots, \delta_{N_C c_k}]^T$  and the output vector  $[g_1, \dots, g_{N_C}]^T$ , because why would, figuratively said, mistaking e.g. a car as a giraffe be less/more accurate than mistaking it as a tree.

During training of the network, weights are adapted such that  $E^S(\mathcal{V}_T)$  is reduced with each epoch of the training. One epoch means one evaluation of Eq. (C.87) for the batches/sets  $\mathcal{V}_T$  and  $\mathcal{V}_V$ . Any optimization scheme can be used for the training, but the large number of parameters due to the generally large number of synapses (connections between neurons) causes that only a few actually show good performance.

<sup>1</sup>  $\frac{1}{|\mathcal{V}_T|} \sum_{k|\boldsymbol{\phi}_k \in \mathcal{V}_T} \sum_{c=1}^{N_C} (\delta_{cc_k} - g_{c_k})^2$

Most commonly applied are methods which use the gradient of a function describing the network performance w.r.t the network weights. Different types of gradient-based methods have been developed, some using the Jacobian matrix, others using conjugated gradients, but in the present application gradient-descent with adaptive learning rate has given the best performance. In the following the method will be explained in detail.

The goal of training a FFNN is to iteratively adapt the weights as

$${}^{(k+1)}w_{jl}^{HL} := {}^{(k)}w_{jl}^{HL} + \Delta w_{jl}^{HL}, \quad {}^{(k+1)}w_{j0}^{HL} := {}^{(k)}w_{j0}^{HL} + \Delta w_{j0}^{HL}, \quad (\text{C.88})$$

$${}^{(k+1)}w_{cj}^{OL} := {}^{(k)}w_{cj}^{OL} + \Delta w_{cj}^{OL}, \quad {}^{(k+1)}w_{c0}^{OL} := {}^{(k)}w_{c0}^{OL} + \Delta w_{c0}^{OL}, \quad (\text{C.89})$$

such that with each epoch ( $k$ ) the performance function (C.87) is reduced. Calculation of the required adaptations  $\Delta w_{jl}^{HL}$  and  $\Delta w_{cj}^{OL}$  may be done by calculating the gradient of the performance function w.r.t. the network weights as

$$\Delta w_{jl}^{HL} := -L_R \frac{\partial E^S(\mathcal{V}_T)}{\partial w_{jl}^{HL}}, \quad \Delta w_{j0}^{HL} := -L_R \frac{\partial E^S(\mathcal{V}_T)}{\partial w_{j0}^{HL}}, \quad (\text{C.90})$$

$$\Delta w_{cj}^{OL} := -L_R \frac{\partial E^S(\mathcal{V}_T)}{\partial w_{cj}^{OL}}, \quad \Delta w_{c0}^{OL} := -L_R \frac{\partial E^S(\mathcal{V}_T)}{\partial w_{c0}^{OL}}, \quad (\text{C.91})$$

where  $L_R$  is a user-defined learning rate (wherever the epoch is not specified, values refer to epoch ( $k$ )). Applying the chain rule to Eq. (C.91) and calculating the required derivatives of Eqns. (C.87), (6.11), and (6.9) gives

$$\begin{aligned} \Delta w_{cj}^{OL} &= -L_R \frac{\partial E^S(\mathcal{V}_T)}{\partial g_c} \frac{\partial g_c}{\partial t_c^{OL}} \frac{\partial t_c^{OL}}{\partial w_{cj}^{OL}} = L_R \underbrace{\frac{1}{|\mathcal{V}_T|} \sum_{k|\phi_k \in \mathcal{V}_T} \delta_{cc_k} g_c}_{=-\frac{\partial E^S(\mathcal{V}_T)}{\partial g_c}} \underbrace{\frac{g_c^2 \sum_{s \neq c}^{N_C} e^{t_s^{OL}}}{e^{t_c^{OL}}}}_{=\frac{\partial g_c}{\partial t_c^{OL}}} \underbrace{f_j}_{=\frac{\partial t_c^{OL}}{\partial w_{cj}^{OL}}} \\ &= \frac{L_R}{|\mathcal{V}_T|} \sum_{k|\phi_k \in \mathcal{V}_T} \left( \delta_{cc_k} \frac{g_c \sum_{s \neq c}^{N_C} e^{t_s^{OL}}}{e^{t_c^{OL}}} f_j \right), \end{aligned} \quad (\text{C.92})$$

$$\Delta w_{c0}^{OL} = \frac{L_R}{|\mathcal{V}_T|} \sum_{k|\phi_k \in \mathcal{V}_T} \left( \delta_{cc_k} \frac{g_c \sum_{s \neq c}^{N_C} e^{t_s^{OL}}}{e^{t_c^{OL}}} \right). \quad (\text{C.93})$$

Hence, the batch-wise adaptation is just the sum of sample-wise adaptations. In an equivalent manner, Eq. (C.90) can be rewritten using Eqns. (C.87), (6.11), (6.10), and (6.7):

$$\Delta w_{jl}^{HL} := -L_R \frac{\partial E^S(\mathcal{V}_T)}{\partial g_c} \frac{\partial g_c}{\partial t_c^{OL}} \frac{\partial t_c^{OL}}{\partial f_j} \frac{\partial f_j}{\partial t_j^{HL}} \frac{\partial t_j^{HL}}{\partial w_{jl}^{HL}}$$



$$\begin{aligned}
&= L_R \underbrace{\frac{1}{|\mathcal{V}_T|} \sum_{k|\boldsymbol{\phi}_k \in \mathcal{V}_T} \delta_{cc_k}}_{=-\frac{\partial E^S(\mathcal{V}_T)}{\partial g_c}} \underbrace{\frac{g_c^2 \sum_{s \neq c}^{N_C} e^{t_s^{OL}}}{e^{t_c^{OL}}}}_{=\frac{\partial g_c}{\partial t_c^{OL}}} \underbrace{w_{cj}^{OL}}_{=\frac{\partial t_c^{OL}}{\partial f_j}} \underbrace{\frac{4e^{-2t_j^{HL}}}{(1+e^{-2t_j^{HL}})^2}}_{=\frac{\partial f_j}{\partial t_j^{HL}}} \underbrace{h_l}_{=\frac{\partial t_j^{HL}}{\partial w_{jl}^{HL}}} \\
&= \frac{4L_R}{|\mathcal{V}_T|} \sum_{k|\boldsymbol{\phi}_k \in \mathcal{V}_T} \left( \delta_{cc_k} w_{cj}^{OL} \frac{e^{-2t_j^{HL}} h_l g_c \sum_{s \neq c}^{N_C} e^{t_s^{OL}}}{e^{t_c^{OL}} (1+e^{-2t_j^{HL}})^2} \right), \tag{C.94}
\end{aligned}$$

$$\Delta w_{j0}^{HL} := \frac{4L_R}{|\mathcal{V}_T|} \sum_{k|\boldsymbol{\phi}_k \in \mathcal{V}_T} \left( \delta_{cc_k} w_{cj}^{OL} \frac{e^{-2t_j^{HL}} g_c \sum_{s \neq c}^{N_C} e^{t_s^{OL}}}{e^{t_c^{OL}} (1+e^{-2t_j^{HL}})^2} \right). \tag{C.95}$$

Back-propagation training of the network using the adaptive gradient-decent algorithm is then performed as follows:

1. Initialization: Define training set  $\mathcal{V}_T$  and validation set  $\mathcal{V}_V$ , set epoch  $k := 1$  and counter of poor adaptations  $n_F := 0$ . Initialize network weights  ${}^{(k)}w_{jl}^{HL}$ ,  ${}^{(k)}w_{j0}^{HL}$ ,  ${}^{(k)}w_{cj}^{OL}$ ,  ${}^{(k)}w_{c0}^{OL}$  which is usually done randomly. Choose maximal number of epochs  $N_B$ , maximum number of poor adaptations  $N_F$ , maximum performance increase  $\Delta_{max}^+ E^S > 1$ , minimum performance increase  $1 \leq \Delta_{min}^+ E^S \leq \Delta_{max}^+ E^S$ , initial learning rate  $L_R$ , and increase/decrease  $L_R^{+/-}$  of the learning rate.
2. Feed every sample  $\boldsymbol{\phi}_k \in \mathcal{V}_T$  and  $\boldsymbol{\phi}_k \in \mathcal{V}_V$  to the network and calculate the initial performances  ${}^{(k)}E^S(\mathcal{V}_T)$  and  ${}^{(k)}E^S(\mathcal{V}_V)$  using Eq. (C.87).
3. Evaluate Eqns. (C.88)–(C.95) based on  ${}^{(k)}E^S(\mathcal{V}_T)$  in order to calculate the new weights  ${}^{(k+1)}w_{jl}^{HL}$ ,  ${}^{(k+1)}w_{j0}^{HL}$ ,  ${}^{(k+1)}w_{cj}^{OL}$ ,  ${}^{(k+1)}w_{c0}^{OL}$ .
4. Feed every sample  $\boldsymbol{\phi}_k \in \mathcal{V}_T$  and  $\boldsymbol{\phi}_k \in \mathcal{V}_V$  to the network and calculate the new performances  ${}^{(k+1)}E^S(\mathcal{V}_T)$  and  ${}^{(k+1)}E^S(\mathcal{V}_V)$  using Eq. (C.87).
5. If  ${}^{(k+1)}E^S(\mathcal{V}_T) / {}^{(k)}E^S(\mathcal{V}_T) < \Delta_{max}^+ E^S$  set  $L_R := L_R L_R^+$ .  
 If  ${}^{(k+1)}E^S(\mathcal{V}_T) / {}^{(k)}E^S(\mathcal{V}_T) \geq \Delta_{max}^+ E^S$  set  $L_R := L_R L_R^-$ , undo weight changes, i.e.  ${}^{(k+1)}w_{jl}^{HL} := {}^{(k)}w_{jl}^{HL}$ ,  ${}^{(k+1)}w_{cj}^{OL} := {}^{(k)}w_{cj}^{OL}$ , and go to step 3.  
 If  ${}^{(k+1)}E^S(\mathcal{V}_V) / {}^{(k)}E^S(\mathcal{V}_V) \leq \Delta_{min}^+ E^S$ , set  $n_F := n_F + 1$ , else  $n_F := 0$ .
6. Stop training if
  - I.  $k = N_B$  or
  - II.  $n_F = N_F$ ,
 else set  $k := k + 1$  and go to step 3.



# List of Figures

1.1	Compressor, combustor, and turbine of ALSTOM gas turbine GT13E2 (printed by courtesy of General Electric Switzerland) . . . . .	2
1.2	Compressor design process . . . . .	4
2.1	Multidisciplinary compressor-design-optimization process with an initial design guess of airfoils calculated by the pre-blading process and optimized by the blading process . . . . .	12
2.2	Geometry model with a) plane of constant average radius, b) geometry parameters, and c) fillet parameters . . . . .	14
2.3	B-Spline representation of a) geometry parameters with only radial dependency, b) camber-line angle, and c) thickness distribution . . . . .	16
2.4	Campbell diagram with a) multiple intersections (o) between eigenfrequencies $f_i^E$ and engine orders $f_j^A$ and b) an intersection of a single engine-order with an eigenfrequency band due to uncertainties . . . . .	18
2.5	Rotor compressor airfoils with a) hammer-foot root (radially fixed at A and circumferentially at B by neighboring airfoils) and b) dovetail root (radially and circumferentially fixed at C) . . . . .	19
2.6	Major eigenmodes to be identified, where the naming convention is as follows: a mode counter is followed by mode specifier B for bending, T for torsion, C for chord-wise bending, H for higher-order, S for stiff-wise-bending mode (Note: shown eigenmodes originate not from same geometry). . . . .	21
2.7	Haigh diagram with endurance limit marked by red line . . . . .	24
2.8	Robustness assessment with a) failure rate and b) production tolerance $\pm s_{\perp}$ . . . . .	25
2.9	Average objective errors (calculated from all nominal designs of a DoE with 3000 samples) for different sample sizes $N_S$ w.r.t. sampling with $N_S := 5000$ . . . . .	28
3.1	Finite-element mesh of a) dovetail root and b) hammer-foot root . . . . .	33
3.2	Error between linear and nonlinear analysis of first 10 eigenfrequencies for three different stages of the test compressor at loose fixation condition . . . . .	35
3.3	Iterative loaded-to-unloaded transformation (a), sketch of airfoil loads (b), and reversed loads (c) . . . . .	36
3.4	Error of first 10 eigenfrequencies computed with linear loaded-to-unloaded transformation (L2U) and without (L2L+) for a) the first, b) middle, and c) last rotor of an industrial gas-turbine . . . . .	37

3.5	First ten eigenmode shapes and eigenfrequencies of a compressor blade (middle stage; dark regions show high absolute displacements) at a) loose fixation and nominal temperature, b) tight fixation and nominal temperature, and c) tight fixation and idle temperature . . . . .	38
3.6	Material for fixation study (a) and corresponding eigenfrequencies sorted by b) frequency values and c) eigenmode shapes of a rear rotor blade (stage 15)	39
3.7	Change of the absolute displacement field of the 4th till 13th eigenmode of a rear compressor blade due to varying fixation stiffness (the markers and data point numbers refer to Fig. 3.6)c . . . . .	42
4.1	Blade surface extraction and projection onto standard square . . . . .	43
4.2	Unstructured FE-mesh (a) transferred to structured mesh (b) and projected onto square (c) . . . . .	44
4.3	Neural network with best matching neuron $\mathbf{x}_{(i,j)^*}^{(k)}$ and neighborhood radius $R^{(k)}$ . . . . .	45
4.4	Surface error of the mapping depending on shape parameter $\sigma$ of the neighborhood function . . . . .	47
4.5	Adaption of mesh to airfoil surface during the training of the SOM starting from random initialization of the neurons . . . . .	48
4.6	Final airfoil surface mesh compared to the initial mesh before training (both normalized) . . . . .	49
4.7	FE-mesh nodes ( $\bullet$ ) and the fitted regular mesh ( $-$ ) after $N_T = 100,000$ iterations with index sets a) (4.8), b) (4.14), and c) (4.19) . . . . .	50
5.1	Extracted projection of the characteristic displacement fields of a compressor blade . . . . .	54
5.2	Effect of normalization on cluster separation (silhouette values) . . . . .	57
5.3	Example of a) artificial cluster distribution (marked by x and o, respectively) with corresponding b) silhouette values and c) distance ratios . . . . .	58
5.4	Effect of normalization on cluster separation (distance ratio) . . . . .	59
5.5	Effect of normalization onto the type of cluster neighbors and their distance (indicated by marker size) . . . . .	60
5.6	Trace of characteristic pixels of the perpendicular displacement field . . . . .	62
5.7	Haar-filters used for dimension reduction . . . . .	63
5.8	PCA projection (a) and MDA projection (b) . . . . .	63
5.9	Perpendicular displacement fields of the first 50 principle directions with the according percentages of total variance . . . . .	64
5.10	Perpendicular displacement fields of the first 10 discriminants . . . . .	65
5.11	Silhouette plots of the different dimension reduction methods . . . . .	66
5.12	Effect of b) normalization, c) PCA, and d) whitening on data structure (a) with corresponding silhouette plots . . . . .	67
5.13	Mean value ( $\square$ ) and the range between minimum and maximum MAC-value for discriminants calculated based on reference set $\mathcal{V}$ and reduced reference set $\mathcal{V} \setminus \mathcal{V}_r$ . . . . .	68
5.14	Effect of different dimension reduction methods onto the distance ratio . . . . .	70

5.15	Effect of different dimension reduction methods onto the type of cluster neighbors and distance (size of marker) . . . . .	71
6.1	Binary classification (different shades of gray; 50 samples per cluster) with kNN algorithm for different numbers of neighbors $k$ ( $k = 1$ represents nearest-neighbor algorithm) . . . . .	74
6.2	Binary classification (different shades of gray; 50 samples per cluster) with SVM algorithm: a) linear for linearly separable data, b) linear for nonlinearly separable data (with slag), and c) nonlinear for nonlinearly separable data	76
6.3	Fully connected feed-forward-neural network with three layers . . . . .	77
6.4	Biases $w_{j0}^{HL}$ and $w_{c0}^{OL}$ before (black line) and after training (red dashed line) for following initializations a) according to Eq. (6.17), b) all weights set to be zero, c) random initialization (best performance out of 20 random settings), and d) HL weights w.r.t. variances and OL according to Eq. (6.17)	82
6.5	MAC between the initial $^{(0)}\bullet$ and the final $^{(*)}\bullet$ setting of the bias-weight vector of the hidden layer $\mathbf{w}_0^{HL}$ (full line) and output layer $\mathbf{w}_0^{OL}$ (dashed line) for training the BPNN with the set (6.19) for each compressor row $r$	84
6.6	MAC between the initial $^{(0)}\bullet$ and the final $^{(*)}\bullet$ setting of the hidden-layer $w_{jl}^{HL}$ (full line) and output-layer weights $w_{cj}^{OL}$ (dashed line) for training the BPNN with the set (6.19) for each compressor row $r$	85
6.7	Confusion of best performing 1NN, $MAC_k$ , SVM, and BPNN classifier w.r.t. the judgment of a human expert . . . . .	92
6.8	Assignment of eigenmode shapes for a) front rotor blade of stage 5, b) middle rotor blade of stage 8, and c) rear rotor blade of stage 15; numbers show the corresponding decision step; human-expert classification (left of  ) is compared to the procedure result (right of  ) . . . . .	96
7.1	Difference fitted parameter distributions and the reference design for a) one dimensional distributions, b) normalized chamber-line-angle distribution, c) and normalized thickness distribution . . . . .	98
7.2	Geometry and Campbell diagram with engine orders and eigenmode-shape bands of a) the reference design and b) the design resulting from the fitting of the parameter model . . . . .	99
7.3	Geometry and Campbell diagram with eigenmode-shape bands of optimum obtained with a) constant weight function (2.14) and b) normal distribution (2.13) where * marks unknown eigenmode shapes not contained by the set of relevant mode shapes in Fig. 2.6 . . . . .	100
7.4	Von Mises-stress distribution of a) reference design, b) optimum with constant weight function (2.14), and c) with normal distribution (2.13) . .	102
7.5	Difference between the parameter distributions of design A and reference design for a) one dimensional distributions, b) normalized chamber-line-angle distribution, and c) normalized thickness distribution . . . . .	103
7.6	Difference between the parameter distributions of design B and reference design for a) one dimensional distributions, b) normalized chamber-line-angle distribution, and c) normalized thickness distribution . . . . .	104

7.7	First ten most influential optimization parameters for the first 11 eigenmodes regarding a) frequency mean $\bar{f}_i^E$ , b) frequency delta $\Delta f_i^E$ , and the difference in parameter ID between those two . . . . .	105
7.8	Geometry and Campbell diagram for the robust optimal design . . . . .	106
7.9	Von Mises-stress distribution of the robust optimum . . . . .	106
7.10	Difference between the optimal parameter distributions of the robust design and the distributions of the reference design for a) one-dimensional distributions, b) normalized chamber-line-angle distribution, and c) normalized thickness distribution . . . . .	107
A.1	Blade-fixation study of a front rotor (stage 5) with corresponding eigenfrequencies sorted by a) frequency values and b) eigenmode shapes . . . . .	113
A.2	Blade-fixation study of a middle rotor (stage 8) with corresponding eigenfrequencies sorted by a) frequency values and b) eigenmode shapes . . . . .	114
A.3	Silhouette plots of the different dimension reduction methods . . . . .	115
A.4	Effect of different dimension reduction methods onto the distance ratio . . . . .	116
A.5	Effect of different dimension reduction methods onto the type of cluster neighbors and distance (size of marker) . . . . .	117
A.6	Effect of BPNN data processing onto the type of cluster neighbors and distance (size of marker): a) before HL without biases, b) before HL with biases, c) after HL, and d) before OL with biases . . . . .	118
C.1	Two spring-coupled masses . . . . .	119
C.2	Eigenfrequencies (a) and amplitude ratios (b) of coupled masses as variations over the difference in stiffness of the wall connected springs . . . . .	121

# List of Tables

2.1	Penalty factors assigned to eigenmode types shown in Fig. 2.6 . . . . .	22
2.2	Engine-order penalty factors . . . . .	22
3.1	Description of analyses performed for structural design evaluation of airfoils with hammer-foot root . . . . .	34
6.1	Classification performance of 1NN and 3NN classifiers . . . . .	87
6.2	Classification performance of $MAC_k$ and $MAC^c$ classifier . . . . .	89
6.3	Classification performance of SVM classifier . . . . .	90
6.4	Classification performance of BPNN classifier . . . . .	91
6.5	Execution statistics for the steps of the assignment procedure . . . . .	95





# Bibliography

- Aizerman, M. A., Braverman, E. A., Rozonoer, L., 1964. Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning. *Automation and Remote Control*, Vol. 25, pp. 821-837.
- Allemang, R. J., 2002. The Modal Assurance Criterion - Twenty Years of Use and Abuse. In: *Proc. of IMAC-XX: Conference & Exposition on Structural Dynamics 2002*, Los Angeles, pp. 397-405.
- Allemang, R. J., Brown, D. L., 1982. A Correlation Coefficient for Modal Vector Analysis. In: *Proc. of the First International Modal Analysis Conference 1982*, Orlando, pp. 110-116.
- Aronszajn, N., 1950. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, Vol. 68, pp. 337-404.
- Astrua, P., Piola, S., Silingardi, A., Bonzani, F., 2012. Multi-Objective Constrained Aero-Mechanical Optimization of an Axial Compressor Transonic Blade. In: *Proc. of ASME Turbo Expo 2012*, Copenhagen, GT2012-68993.
- Asuncion, A., Newman, D. J., 2007. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences.
- Aulich, A.-L., Goerke, D., Blocher, M., Nicke, E., Kocian, F., 2013. Multidisciplinary Automated Optimization Strategy on a Counter Rotating Fan. In: *Proc. of ASME Turbo Expo 2013*, San Antonio, GT2013-94259.
- Baldi, P., Hornik, K., 1989. Neural Networks and Principle Component Analysis: Learning from Examples Without Local Minima. *Neural Networks*, Vol. 2, pp. 53-58.
- Bathe, K.-J., 1996. *Finite Element Procedures*. Prentice-Hall, New Jersey.
- Bestle, D., 1994. *Analyse und Optimierung von Mehrkörpersystemen: Grundlagen und rechnergestützte Methoden*. Springer, Berlin.
- Bishop, C. M., 2006. *Pattern Recognition and Machine Learning*. Springer, New York.
- Blocher, M., Aulich, A.-L., 2013. Flutter Susceptibility Approximation via Curve Fitting and MAC-Analysis in an Automated Optimization Design Process. In: *Proc. of ASME Turbo Expo 2013*, San Antonio, GT2013-94577.

- Blocher, M., Fernández, I. E. G., 2014. Time-Linearized Forced Response Analysis of a Counter Rotating Fan; Part I: Theoretical Concept of a Fully Time-Linear Forced Response Analysis. In: Proc. of ASME Turbo Expo 2014, Düsseldorf, GT2014-25833.
- Boser, B. E., Guyon, I. M., Vapnik, V. N., 1992. A Training Algorithm for Optimal Margin Classifiers. In: Proc. of the 4th Workshop on Computational Learning Theory 1992, pp. 144–152.
- Bouhlel, M. A., Bartoli, N., Otsmane, A., Morlier, J., 2016. Improving Kriging Surrogates of High-Dimensional Design Models by Partial Least Squares Dimension Reduction. *Structural and Multidisciplinary Optimization*, Vol. 53(5), pp. 935–952.
- Bräunling, W. J., 2015. *Flugzeugtriebwerke: Grundlagen, Aero-Thermodynamik, ideale und reale Kreisprozesse, Thermische Turbomaschinen, Komponenten, Emissionen und Systeme*. Springer, Berlin.
- Buske, C., Krumme, A., Schmidt, T., Dresbach, C., Zur, S., Tiefers, R., 2016. Distributed Multidisciplinary Optimization of a Turbine Blade Regarding Performance, Reliability and Castability. In: Proc. of ASME Turbo Expo 2016, Seoul, GT2016-56079.
- Campbell, W., 1924. Protection of Steam Turbine Disk Wheels from Axial Vibration. *Trans. of the ASME* 1924, Vol. 46, pp. 31–160.
- Ciregan, D., Meier, U., Masci, J., Schmidhuber, J., 2012. Multi-Column Deep Neural Network for Traffic Sign Classification. *Neural Networks*, Vol. 32, pp. 333–338.
- Cortes, C., Vapnik, V., 1995. Support-vector networks. *Machine Learning*, Vol. 20(3), pp. 273–297.
- Cumpsty, N., 2004. *Compressor Aerodynamics*. Krieger Publishing Company, Malabar.
- de Haller, P., 1953. Das Verhalten von Tragflügeln in Axialverdichtern und im Windkanal. In: *Brennstoff-Wärme-Kraft 5*, VDI-Verlag, Düsseldorf, pp. 333–336.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, Vol. 6(2), pp. 182–197.
- Diener, O. H. F., von Backström, T. W., van der Spuy, S. J., Hildebrandt, T., 2016. Multi-Disciplinary Optimization of a Mixed-Flow Compressor Impeller. In: Proc. of ASME Turbo Expo 2016, Seoul, GT2016-57008.
- Dierckx, P., 1993. *Curve and Surface Fitting with Splines*. Oxford University Press, New York.
- Dow, E. A., Wang, Q., 2015. The Implications of Tolerance Optimization on Compressor Blade Design. *Journal of Turbomachinery*, Vol. 137, doi:10.1115/1.4030791.

- Du, X., Sudjianto, A., Chen, W., 2004. An Integrated Framework for Optimization Under Uncertainty Using Inverse Reliability Strategy. *Journal of Mechanical Design*, Vol. 126, pp. 562–570.
- Duan, K.-B., Keerthi, S. S., 2005. Which Is the Best Multiclass SVM Method? An Empirical Study. In: *Proc. of Multiple Classifier Systems: 6th International Workshop 2005*, Seaside.
- Duda, R. O., Hart, P. E., Stork, D. G., 2000. *Pattern Classification*. John Wiley & Sons, New York.
- Dutta, A. K., 2011. *An Automated Multi-Objective Optimization Approach for Aerodynamic Compressor Blade Designs*. Shaker, Aachen.
- Fedorov, I., Szwedowicz, J., Kappis, W., Putschkov, I., 2010. Reliable FE Modeling of Chord-Wise Blade Modes in Compressor Design Process. In: *Proc. of ASME Turbo Expo 2010*, Glasgow, GT2010-22619.
- Fernández, I. E. G., Blocher, M., 2014. Time-Linearized Forced Response Analysis of a Counter Rotating Fan; Part II: Analysis of the DLR CRISP2 Model. In: *Proc. of ASME Turbo Expo 2014*, Düsseldorf, GT2014-25838.
- Fisher, R. A., 1936. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics* 1936, Vol. 7(2), pp. 179–188.
- Flassig, P. M., 2011. *Unterstützende Optimierungsstrategien zur robusten aerodynamischen Verdichterschaufelauslegung*. Shaker, Aachen.
- Forrester, A. I. J., Sobester, A., Keane, A. J., 2008. *Engineering Design via Surrogate Modelling: A Practical Guide*. John Wiley & Sons, New York.
- Friedman, J. H., 1989. Regularized Discriminant Analysis. *Journal of the American Statistical Association* 1989, Vol. 84(405), pp. 165–175.
- Fukunaga, K., 1990. *Introduction to Statistical Pattern Recognition*, 2nd Edition. Academic Press, San Diego.
- Goerke, D., Le Denmat, A.-L., Schmidt, T., Kocian, F., Nicke, E., 2012. Aerodynamic and Mechanical Optimization of CF/PEEK Blades of a Counter Rotating Fan. In: *Proc. of ASME Turbo Expo 2012*, Copenhagen, GT2012-68797.
- Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R., 2004. Neighbourhood Components Analysis. In: *Proc. of Advances in Neural Information Processing Systems 17*, pp. 513–520. MIT Press.
- Golub, G., Reinsch, C., 1971. Singular Value Decomposition and Least Squares Solutions. *Handbook for Automatic Computation*, Vol. 186, pp. 134–151.
- Golub, G. H., Welsch, J. H., 1969. Calculation of Gauss Quadrature Rules. *Mathematics of Computation*, Vol. 23, pp. 221–230.

- Grieb, H., 2009. Verdichter für Turbo-Flugtriebwerke. Springer, Berlin.
- Hagan, M. T., Demuth, H. B., Beale, M. H., 1996. Neural Network Design. PWS Publishing Co., Boston.
- Hartwig, L., Bestle, D., 2016. Enhancing Low-Fidelity Compressor Design by Utilizing a Surrogate Model Based on Higher-Fidelity Information. In: Proc. of 7th IC-SCCE, 2016, Athens.
- Hartwig, L., Bestle, D., 2017. Compressor Blade Design for Stationary Gas Turbines Using Dimension Reduced Surrogate Modeling. In: Proc. of IEEE Congress on Evolutionary Computation (CEC), 2017, San Sebastián, E-17366.
- Hechenbichler, K., Schliep, K. P., 2004. Weighted k-Nearest-Neighbor Techniques and Ordinal Classification. Discussion Paper 399, SFB 386, Ludwig-Maximilians University Munich.
- Hecker, P., Delimar, D., Brandl, H., Lötzerich, M., 2011. Process Integration and Automated Numerical Design Optimization of an Eigenfrequency Analysis of a Compressor Blade. In: Proc. of ASME Turbo Expo 2011, Vancouver, GT2011-4548.
- Heseltine, T., Pears, N., Austin, J., Chen, Z., 2003. Face Recognition: A Comparison of Appearance-Based Approaches. In: Proc. of VIIth Digital Image Computing: Techniques and Applications 2003, Sydney.
- Heylen, W., Avitabile, P., 1998. Correlation Considerations - Part 5 (Degree of Freedom Correlation Techniques). In: Proc. of SPIE - The International Society for Optical Engineering 1, pp. 207–214.
- Howland, P., Jeon, M., Park, H., 2003. Structure Preserving Dimension Reduction for Clustered Text Data Based on the Generalized Singular Value Decomposition. SIAM Journal on Matrix Analysis and Applications 2003, Vol. 25(1), pp. 165–179.
- Hsu, C.-W., Lin, C.-J., 2002. A Comparison of Methods for Multi-class Support Vector Machines. IEEE Transactions on Neural Networks 2002, Vol. 13(2), pp. 415–425.
- Janke, C., Goller, M., Martin, I., Gaun, L., Bestle, D., 2016. 3D CFD Compressor Map Computation of a Multi-Stage Axial Compressor with Off-Design Adjusted Rotor Geometries. In: Proc. of ASME Turbo Expo 2016, Seoul, GT2016-56745.
- Jolliffe, I. T., 2002. Principle Component Analysis. Springer, New York.
- Joly, M. M., Verstraete, T., Panlagua, G., 2014. Multidisciplinary Design Optimization of a Compact Highly Loaded Fan. Structural and Multidisciplinary Optimization, Vol. 49, pp. 471–483.
- Keskin, A., 2007. Process Integration and Automated Multi-Objective Optimization Supporting Aerodynamic Compressor Design. Shaker, Aachen.

- Kohonen, T., 1981. Automatic Formation of Topological Maps of Patterns in a Self-Organizing System. Proc. of 2nd Scand. Conf. on Image Analysis 1981, pp. 214–220.
- Kohonen, T., 1990. The Self-Organizing Map. Proc. of the IEEE 1990, Vol. 78, pp. 1464–1480, doi: 10.1109/5.58325.
- Kohonen, T., 2014. MATLAB Implementations and Applications of the Self-Organizing Map. Unigrafia Oy, Helsinki.
- Kreyszig, E., Kreyszig, H., Norminton, E. J., 2011. Advanced Engineering Mathematics, 10th Edition. John Wiley & Sons, New York.
- Krige, D. G., 1951. A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. Journal of the Chemical, Metallurgical, and Mining Society of South Africa, Vol. 52(6), pp. 119–139.
- Kuhn, H. W., Tucker, A. W., 1951. Nonlinear Programming. In: Proc. of the Second Berkeley Symposium on Mathematical Statistics and Probability, pp. 481–492.
- Lakshminarayana, B., 1996. Fluid Dynamics and Heat Transfer of Turbomachinery. John Wiley & Sons, Inc., Hoboken.
- Lanczos, C., 1950. An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators. Journal of Research of the National Bureau of Standards, Vol. 45, pp. 255–282.
- Lee, D.-T., 1982. On k-Nearest Neighbor Voronoi Diagrams in the Plane. IEEE Transactions on Computers 1982, Vol. C-31(6), pp. 478–487.
- Leissa, A. W., 1974. On a Curve Veering Aberration. Journal of Applied Mathematics and Physics, Vol. 25, pp. 99–111.
- Lieblein, S., Schwenk, F. C., Broderick, R. L., 1953. Diffusion Factor for Estimating Losses and Limiting Blade Loadings in Axial Flow Compressor Blade Elements. NACA Research Memorandum RM E53D01, Washington.
- Lockan, M., Amtsfeld, P., Bestle, D., Meyer, M., 2014. Non-Linear Sensitivity Analysis Based on Regression Trees with Application to 3D Aerodynamic Optimization of Gas Turbine Blades. In: Proc. of OPT-i - An International Conference on Engineering and Applied Sciences Optimization 2014, Kos Island.
- Lockan, M., Bestle, D., Janke, C., Meyer, M., 2017. Optimization of Coupled System Components Using Approximations of Interface Quantities. In: Proc. of ASME Turbo Expo 2017, Charlotte, GT2017-64135.
- MacNeal, R. H., Faulkner, L. L., 1994. Finite Elements: Their Design and Performance. Marcel Dekker Inc., New York.

- Martin, I., Bestle, D., 2016. Automated Mode Identification of Airfoil Geometries to be Used in an Optimization Process. In: Proc. of ASME Turbo Expo 2016, Seoul, GT2016-56987.
- Martin, I., Bestle, D., 2018. Automated Eigenmode Classification for Airfoils in the Presence of Fixation Uncertainties. *Engineering Applications of Artificial Intelligence* 2018, Vol. 67, pp. 187–196.
- Martin, I., Hartwig, L., Bestle, D., 2019. A Multi-Objective Optimization Framework for Robust Axial Compressor Airfoil Design. *Journal of Structural and Multidisciplinary Optimization*, doi: <https://doi.org/10.1007/s00158-018-2164-3>.
- McKay, M. D., Beckman, R. J., Conover, W. J., 1979. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, Vol. 21(2), pp. 239–245.
- Mercer, J., 1909. Functions of Positive and Negative Type, and Their Connection with the Theory of Integral Equations. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, Vol. 209 (441–458), pp. 415–446.
- Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Müller, K. R., 1999. Fisher Discriminant Analysis with Kernels. In: Proc. of the 1999 IEEE Signal Processing Society Workshop, pp. 41–48.
- Mises, R. v., 1913. *Mechanik der festen Körper in plastisch-deformablen Zustand*. Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse 4 (1913), pp. 582–592.
- Morales, C. A., 2005. Comments on the MAC and the NCO, and a Linear Modal Correlation Coefficient. *Journal of Sound and Vibration*, Vol. 282(1-2), pp. 529–537.
- Mudd, M. J., 1995. General Configuration of PFBC Plants. In: Cuenca M.A., Anthony E.J. (Eds) *Pressurized Fluidized Bed Combustion*. Springer, Dordrecht, pp 121-134.
- Nicholas, T., 2006. *High Cycle Fatigue: A Mechanics of Materials Perspective*. Elsevier Science, Amsterdam.
- Novotny, L., 2010. Strong Coupling, Energy Splitting, and Level Crossings: A Classical Perspective. *American Journal of Physics*, Vol. 78, pp. 1199–1202.
- Otto, D., Bestle, D., 2007. Multi-Disciplinary Blading Design by Means of Multi-Objective Optimization. In: Proc. of 1st European Air and Space Conference (CEAS), Berlin.
- Park, C. H., Park, H., 2005. Nonlinear Discriminant Analysis Using Kernel Functions and the Generalized Singular Value Decomposition. *SIAM Journal on Matrix Analysis and Applications* 2005, Vol. 27(1), 2005, pp. 87–102.
- Piegl, L., Tiller, W., 1997. *The NURBS Book: Second Edition*. Springer, Berlin.

- Pierret, S., Coelho, R. F., Kato, H., 2007. Multidisciplinary and Multiple Operating Points Shape Optimization of Three-Dimensional Compressor Blades. *Structural and Multidisciplinary Optimization*, Vol. 33, pp. 61–70.
- Pöhlmann, F., Bestle, D., 2012. Multi-Objective Compressor Design Optimization Using Multi-Design Transfer between Codes of Different Fidelity. In: *Proc. of ASME Turbo Expo 2012, Copenhagen*, GT2012-68577.
- Pugachev, A. O., Sheremetyev, A. V., Tykhomirov, V. V., Petrov, A. V., 2014. Structural and Dynamic Optimization of a Single Axial Compressor Blade Using the Gradient-Based Method. In: *Proc. of ASME Turbo Expo 2014, Düsseldorf*, GT2014-25249.
- Rosipal, R., Krämer, N., 2006. Overview and Recent Advances in Partial Least Squares. Subspace, Latent Structure and Feature Selection. *Lecture Notes in Computer Science 2006*, Vol. 3940, pp. 34–51.
- Roth, V., Steinhage, V., 1999. Nonlinear Discriminant Analysis Using Kernel Functions. In: *Proc. of Advances in Neural Information Processing Systems 1999*, pp. 568–574.
- Rousseeuw, P. J., 1987. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*, Vol. 20, pp. 53–65.
- Rühle, T., Bestle, D., 2010. Compressor Design and Optimisation Based on Throughflow Calculation. In: *Proc. of ADOS Automatic Design Optimization Seminar 2010, Derby*.
- Russel D'A, H., 1973. *Thermodynamic Cycles and Processes (Introductory Engineering)*. Prentice-Hall, New Jersey.
- Schölkopf, B., Smola, A., 2002. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge.
- Schölkopf, B., Smola, A., Müller, K.-R., 1997. Kernel Principal Component Analysis. In: *Proc. of 7th International Conference of Artificial Neural Networks 1997, Lausanne*, pp. 583–588.
- Schörner, R., Bestle, D., 2012. Coupled Optimization of Compressor Blade Roots and Rotor Grooves for Industrial Gas-Turbines. In: *Proc. of 5th IC-SCCE International Conference: From Scientific Computing to Computational Engineering, Athens*.
- Seppälä, J., Hupfer, A., 2014. Topology Optimization in Structural Design of a LP Turbine Guide Vane: Potential of Additive Manufacturing for Weight Reduction. In: *Proc. of ASME Turbo Expo 2014, Düsseldorf*, GT2014-25637.
- Sibson, R., 1981. A Brief Description of Natural Neighbor Interpolation, In: V. Barnett (Ed.) *Interpreting Multivariate Data*, John Wiley & Sons, New York, pp. 21–36.
- Siller, U., Marcel, A., 2010. Multidisciplinary 3D-Optimization of a Fan Stage Performance Map with Consideration of the Static and Dynamic Rotor Mechanics. In: *Proc. of ASME Turbo Expo 2010, Glasgow*, GT2010-22792.

- Singh, M. P., Vargo, J. J., Schiffer, D. M., Dello, J. D., 1988. SAFE Diagram - A Design and Reliability Tool for Turbine Blading. In: Proc. of 17th Turbomachinery Symposium, Turbomachinery Laboratory, Texas A&M University, College Station 1988, pp. 93–101.
- Strehlau, U., Kühhorn, A., 2010. Experimental and Numerical Investigation of HPC Blisks with a Focus on Travelling Waves. In: Proc. of ASME Turbo Expo 2010, Glasgow, GT2010-22463.
- Strouhal, V., 1878. Über eine besondere Art der Tonerregung. *Annalen der Physik und Chemie*, Vol. 5, pp. 216–251.
- Vapnik, V., Lerner, A., 1963. Pattern Recognition Using Generalized Portrait Method. *Automation and Remote Control*, Vol. 24, pp. 774–780.
- Vinogradov, K. A., Didenko, R. A., Kretinin, G. V., Karelin, D. V., Otryahina, K. V., Shmotin, Y. N., 2016. Robust Optimization of the HPT Blade Cooling and Aerodynamic Efficiency. In: Proc. of ASME Turbo Expo 2016, Seoul, GT2016-56195.
- Viola, P., Jones, M. J., 2004. Robust Real-Time Face Detection. *International Journal of Computer Vision*, Vol. 57(2), pp. 137–154.
- Wold, H., 1985. Partial Least Squares. *Encyclopedia of the Statistical Sciences* 1985, Vol. 6, pp. 581–591.
- Wolpert, D. H., 1996. The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Computation* 1996, Vol. 8(7), pp. 1341–1390.
- World Energy Council, 2016. Coal World Energy Resources 2016. [https://www.worldenergy.org/wp-content/uploads/2017/03/WEResources\\_Coal\\_2016.pdf](https://www.worldenergy.org/wp-content/uploads/2017/03/WEResources_Coal_2016.pdf) (visited on 23/11/2017).
- Wu, C.-H., 1952. A General Theory of Three-Dimensional Flow in Subsonic and Supersonic Turbomachines of Axial-, Radial-, and Mixed-Flow Types. NACA Report TN-2604, NASA, Cleveland.
- Zienkiewicz, O. C., Hinton, E., 1976. Reduced Integration, Function Smoothing and Non-Conformity in Finite Element Analysis (with Special Reference to Thick Plates). *Journal of the Franklin Institute*, Vol. 302(5-6), pp. 443–461.