

Learning-Based Manufacturing Deviation Estimation for Robust Computer-Aided Design

Von der Fakultät für Maschinenbau, Elektro- und Energiesysteme der
Brandenburgischen Technischen Universität Cottbus-Senftenberg
zur Erlangung des akademischen Grades
eines Doktors der Ingenieurwissenschaften

genehmigte Dissertation

vorgelegt von

M.Sc. José Mauricio Urbano Caguasango

geboren am 1.7.1991 in Pasto (Kolumbien)

Vorsitzender: Prof. Dr.-Ing. habil. Bernd Beirow

Gutachter: Prof. Dr.-Ing. habil. Hon. Prof. (NUST) Dieter Bestle

Gutachter: Prof. Dr.-Ing. Peter Flassig

Tag der mündlichen Prüfung: 30.9.2022

DOI: <https://doi.org/10.26127/BTUOpen-6154>

Acknowledgments

The present dissertation is the culmination of my activity as Research Assistant at the Institute of Engineering Mechanics and Vehicle Dynamics (TMF) of the Brandenburg Technical University Cottbus-Senftenberg from October 2017 until June 2021. The hereby featured methods and results were developed in close cooperation with Rolls-Royce Deutschland Ltd & Co KG (RRD) as part of the Research Program “Verfahren der Industrie 4.0 für die Triebwerks-Vorentwicklung” (VIT-V), sponsored by the Brandenburg State and the European Regional Development Fund.

I want to express my profound gratitude to my doctoral supervisor Prof. Dr.-Ing. habil. Hon. Prof. (NUST) Dieter Bestle whose unlimited discussion availability, extensive writing guidance and persistent commitment to excellence fundamentally reshaped my approach to engineering. My sincere gratitude also goes to the evaluation committee, Prof. Dr.-Ing. Peter Flassig for convincing me to pursue an industry doctorate in cooperation with RRD and kick-starting the initial phase with inspiring technical discussions, as well as to Prof. Dr.-Ing. habil. Bernd Beirow for eagerly accepting the evaluation committee chair. A special thanks also to Prof. Dr.-Ing. habil. Ronald Mailach and the TU Dresden for the preparation and provision of the surface measurements used in this investigation, which were conducted within the LuFo research project “RobusTurb” (20T0608A) funded by the Bundesministerium für Wirtschaft und Technologie.

Any intense academic undertaking becomes significantly more cheerful in the presence of a supporting team, which is why I am in great debt to my TMF family. Huge thanks to Peggy Fobo for her support on all university formalities and to Lutz Anklam for always giving a swift and experienced helping hand regarding software support. My fellow doctoral researchers at the time, Dr.-Ing. Chen-Xiang Chao, David Göbel, Malte Niehoff, Simon Extra and Dr.-Ing. Dmitriy Ivanov, were invaluable for turning both wins and setbacks into shared, joyful anecdotes: thank you deeply! My big appreciation also goes to Dr.-Ing. Michael Lockan and Timo Bielitz for inspiring me with their team engagement and to Tianyu Zhai and Dr.-Ing. Xue Rui for sharing with me different perspectives in life.

Having started at RRD as a graduate intern, I must acknowledge the tremendous research drive and curiosity I perceived from RRD colleagues since day one. Huge thanks to Dr.-Ing. Rudolf Lück for giving me the opportunity to put my hands on industry-level

research for the first time, and to Dr.-Ing. Andrej Golowin for trusting me with challenging technical responsibilities during my master thesis. I also sincerely thank Dr.-Ing. Marcus Meyer for serving as major research focal point spanning countless topics and institutions, and for putting such rich network into the hands of young researchers.

None of the methods and findings of this dissertation would have been possible without the intense commitment and disposition of my closest RRD fellows, Dr.-Ing. Ulf Gerstberger and Gnanasivam Ramasamy. Thank you Ulf for your genuine interest in R&D and for continuously encouraging discussion environments and technical opportunities where research and industry may meet. My deepest appreciation also goes to Gnanasivam, for your relentless working spirit and for accompanying me in the intense trial-and-error campaigns intrinsic to the R&D trenches. I am in immense debt to you both.

In occasions where doubt and uncertainty may wander, excellent remedy may be found in laughing, better if accompanied by Colombian home-cooked recipes. Therefore I must acknowledge the incredible fortune of having such strong friendships which have filled my life with memorable experiences throughout all these years. My huge thanks to all my friends in Berlin, Vienna, Bonn and Bogotá, your love and support is truly invaluable to me.

As the saying goes, major accomplishments are sometimes only possible by standing on the shoulders of giants. That is why I want to express my deep gratitude and affection to my two Colombian giants, Dolly and Chepe, who have forged our united, supportive family in the midst of adversity, and who never cease to inspire me through their lifetime example. Much love and admiration to my sisters Diana and Alexandra, and to my brother by heart Sebastian, who constantly remind me of strength, courage and perseverance.

Lastly, my deepest gratitude and admiration go to my beloved Elba. You have filled my life with boundless love, companionship and wisdom. It is my true fortune to learn, reflect and grow at your side. This work is dedicated to you.

Berlin, November 2022

José Mauricio Urbano Caguasango

Abstract

Surface variations are an unavoidable byproduct of any manufacturing process and may lead to deviating part performance and even elevated part rejection rates. Because traditional computer aided-design approaches are aimed towards production of idealized, nominal geometric shapes, the wide geometric—and statistical—variability typical for any manufacturing process remains unrepresented and is frequently ignored during design. Thus, the present work aims at a more realistic design approach and, therefore, develops a collection of computer-aided design strategies for accurate representation, statistical analysis and prospective estimation of surface deviations with validation examples on aero engine turbine blades.

The CAD representation of real manufactured surfaces requires the ability to accurately recreate complex geometric shapes. This is achieved by automated re-parametrization of any CAD face of interest as B-spline surface with a rather dense control point grid. Face matching to scanned manufactured samples is then performed by calculating control point displacements, which successfully deliver surface representation errors below typical measurement uncertainties on multiple matching examples from turbine shank and hot-gas faces. Since inference of performance variability due to manufacturing is usually limited by the amount of scanned manufactured parts, a probabilistic model is formulated based on singular-value decomposition of control point displacements and identification of dominant manufacturing modes. This allows generation of an infinite set of synthetic deviating surfaces faithful to experimental deviation patterns.

Nominal geometric features may significantly differ between design iterations and manufacturing modes may not necessarily be transferable between different designs. Thus, deviation estimation may remain infeasible before manufacturing. To enable deviation estimation during the design phase, the present work proposes a machine learning strategy to identify deviation patterns explained by nominal geometric properties—such as relative position and local orientation—and use them for deviation estimation on new designs. This strategy is able to predict realistic stress variability induced by shank deviations of a turbine blade design using only surface deviation information from three given designs, which encourages machine-learning approaches as valuable tool for geometric deviation estimation as part of robust design.

Kurzfassung

Fertigungsabweichungen sind unvermeidbar und tragen signifikant zu unerwünschten Performance-Variationen und erhöhten Ablehnungsraten bei. Konventionelle rechnergestützte Bauteilauslegungsmethoden konzentrieren sich dagegen meist auf die Erstellung von idealisierten, nominalen Geometrien und lassen statistische Fertigungsvariabilitäten in der Auslegungsphase größtenteils unberücksichtigt. Um beides besser miteinander zu verbinden, werden in dieser Arbeit eine Reihe von rechnergestützten Konstruktions- und Modellierungsmethoden vorgestellt, welche eine virtuelle Rekonstruktion, robuste statistische Analyse und prädiktive Abschätzung von Oberflächenabweichungen erlauben, und mittels Validierungsbeispiele für Turbinenschaufeln nachgewiesen.

Die CAD Darstellung von Fertigungsabweichungen erfordert die Fähigkeit, komplexe geometrische Formen akkurat nachzubilden, weshalb in dieser Arbeit die CAD Fläche zunächst als B-Spline Fläche mit hoher Kontrollpunkt-Dichte re-parametrisiert wird. Die Flächenanpassung an eingescannte, gefertigte Bauteile erfolgt dann durch Berechnung von Kontrollpunkt-Verschiebungen. Anhand mehrerer Anpassungsbeispiele von Heißgas- und Schaftflächen einer Turbinenschaufel wird nachgewiesen, dass Anpassungsgenauigkeiten unterhalb typischer Messunsicherheiten möglich sind. Um statistische Abschätzungen der geometrischen Variabilität des Fertigungsprozesses trotz einer limitierten Anzahl an vermessenen Bauteilen zu ermöglichen, wird ein probabilistisches Modell der Fertigungsstreuung auf Basis von Singulärwertzerlegung der Kontrollpunkt-Abweichungen und Identifizierung dominanter Fertigungsmoden entwickelt. Dies ermöglicht die Generierung unendlich vieler, synthetischer fertigungstreuer Abweichungsexemplare.

Diese Darstellung der geometrischen Variabilität ist allerdings auf eine bestimmte Parametrisierung beschränkt, weshalb eine direkte Übertragung auf neue abweichende geometrische Designs nicht korrekt ist. Um eine solche zu ermöglichen, wird eine Strategie zur Erkennung geometrisch-bedingter Fertigungsmuster basierend auf Methoden des maschinellen Lernens formuliert. Diese Strategie ist in der Lage, realistische Abschätzungen von Oberflächen- und Spannungsvariabilität der Schaftflächen einer Turbinenschaufel vorzunehmen, allein durch Verwendung von Oberflächenmessungen dreier bereits gefertigter Turbinenschaufel-Entwürfe. Damit wird das große Potential des maschinellen Lernens bei der Wissensextraktion aus Fertigungsexemplaren bestätigt und ihre Weiterentwicklung innerhalb robuster Entwurfsprozesse ermutigt.

Contents

Abstract	v
Kurzfassung	vii
Nomenclature, Symbols & Abbreviations	xi
1 Introduction	1
1.1 The Cost of Manufacturing Variability	2
1.2 Estimating Geometric Uncertainty for Robust Design	4
1.3 Problem Statement and Thesis Outline	9
2 CAD Surface Parametrization and Measurement	11
2.1 CAD Surface Parametrization	11
2.2 Geometric Properties of B-Spline Surfaces	16
2.3 Parametric Construction of CAD Parts	19
2.4 Surface Measurement with Structured Light	20
3 CAD Representation of Manufacturing Deviations	29
3.1 Re-Parametrization as B-Spline Surface	30
3.2 Selection of Relevant Surface Positions	32
3.3 Selection of Relevant Control Points	34
3.4 B-Spline Morphing to Surface Measurements	38
3.5 Noise Elimination from Morphed Surfaces	41
3.6 Application to HPT Blade Casting Faces	45
3.6.1 Suction-Side Shank Face	45
3.6.2 Hot-Gas Faces	48
4 Probabilistic Representation of Manufacturing Deviations	51
4.1 Manufacturing Modes	51
4.2 Coupled Manufacturing Modes	55
4.3 Representation Capacity of Manufacturing Modes	58
4.4 Probabilistic Description of Surface Deviations	62

4.5	Application to HPT Blade Structural Uncertainty Quantification	66
4.6	Application to Manufacturing Variability of Diverse Shank Designs	73
5	Deviation Estimation for New Geometric Designs	79
5.1	Strategies for Deviation Estimation	80
5.2	Deviation Model Using Convolution Operators	84
5.3	Learning of Deviation Laws	90
5.4	Minimization of Deviation Prediction Divergence	98
5.5	Application to Estimation of HPT Blade Casting Deviations	102
5.6	Integration of Deviation Laws into Robust Design	112
6	Conclusions	117
	Appendix A: Convolutional Neural Networks	121
A.1	Basic Mathematical Background	121
A.2	Pooling, Interpolation and Filtering Operations	124
	List of Figures	125
	References	129

Abbreviations, Nomenclature & Symbols

Abbreviations

ASTM	<u>A</u> merican <u>S</u> ociety for <u>T</u> esting and <u>M</u> aterials
B2P	<u>B</u> lade to <u>P</u> arameter (proprietary software)
BTU	<u>B</u> ritish <u>T</u> hermal <u>U</u> nit
CAD	<u>C</u> omputer- <u>A</u> ided <u>D</u> esign
CFD	<u>C</u> omputational <u>F</u> luid <u>D</u> ynamics
EI	<u>E</u> nergy <u>I</u> ntensive
FE	<u>F</u> inite <u>E</u> lement
GPU	<u>G</u> raphics <u>P</u> rocessing <u>U</u> nit
HPT	<u>H</u> igh- <u>P</u> ressure <u>T</u> urbine
IGES	<u>I</u> nitial <u>G</u> raphics <u>E</u> xchange <u>S</u> pecification
LE	<u>L</u> eading <u>E</u> dge
NURBS	<u>N</u> on- <u>U</u> niform <u>R</u> ational <u>B</u> - <u>S</u> plines
PCA	<u>P</u> rincipal <u>C</u> omponent <u>A</u> nalysis
PS	<u>P</u> ressure <u>S</u> ide
RGB	<u>R</u> ed <u>G</u> reen <u>B</u> lue
SS	<u>S</u> uction <u>S</u> ide
STEP	<u>S</u> Tandard for the <u>E</u> xchange of <u>P</u> roduct Model Data
SVD	<u>S</u> ingular <u>V</u> alue <u>D</u> ecomposition
TE	<u>T</u> railing <u>E</u> dge

Typographical Convention

a	scalar
\mathbf{a}	vector
\mathbf{A}	matrix
\mathbf{A}	higher-order tensor

a	scalar random variable
\mathbf{a}	random vector
\mathcal{A}	set

Mathematical Symbols

$\tilde{\bullet}$	approximated quantity
$\bar{\bullet}$	average
$\hat{\bullet}$	normalization
\bullet^*	optimum
\forall	for all
\in	member of
\subset	subset of
$ a $	absolute value of scalar a
$ \mathcal{A} $	cardinality of set \mathcal{A}
$\ \mathbf{a}\ $	Euclidian norm of vector \mathbf{a}
Δ	difference
\times	cross product
$*$	convolution operator
\odot	element-wise multiplication (Hadamard product)
\mathbb{B}	Boolean set
\mathbb{N}	set of natural numbers
\mathbb{R}	set of real numbers
\mathbf{A}^\top	transpose of matrix \mathbf{A}
\mathbf{A}^{-1}	inverse of matrix \mathbf{A}
$\text{diag}(\mathbf{a})$	diagonal matrix with diagonal entries \mathbf{a}

Latin Letters

d	Mahalanobis distance
d_{th}	distance threshold for relevant normalized positions
d_{cp}	distance threshold for relevant control points
e_S	surface measurement error
e_C	re-parametrization error

e_M	morphing error
e_R	reconstruction error
e_P	process error
e_{th}	deviation threshold
e	loss function between deviation tensors
I	light intensity function
\mathcal{I}	set of relevant control points
\mathcal{M}	set of measured points
\mathcal{N}	set of measured normal vectors
$N_{i,q}$	B-spline basis function
n_{sup}	supporting relevant normalized positions per control point
O	origin of coordinate system
t	number of observable manufacturing modes
\mathcal{U}	set of normalized relevant positions
u, v	normalized B-spline parameters
x, y, z	Cartesian coordinates

Greek Letters

α	deviation scaling factor
δ	prediction divergence
θ_{th}	angular threshold for relevant normalized positions
κ	curvature
$\hat{\kappa}$	normalized curvature
σ	singular value
τ_{th}	tangential threshold for relevant control points
Φ	projected phase map
Ψ	captured phase map

Vectors

\mathbf{b}	mode amplitude vector
\mathbf{b}	random mode amplitude vector
\mathbf{n}	normal vector
\mathbf{p}	control point

Δp	control point deviation
$\Delta \tilde{p}$	approximated control point deviation
$\Delta \hat{p}$	normalized control point deviation
q	control point deviations of entire B-spline surface
\bar{q}	mean control point deviations of entire B-spline surface
\mathbf{q}	probabilistic control point deviations of entire B-spline surface
s	B-spline surface
\hat{s}	morphed B-spline surface
t	tangential vector
x	measured point; deviation vector
θ	neural network weights
μ	mean mode amplitude vector
μ'	robust mean mode amplitude vector

Matrices

C	covariance matrix of mode amplitudes
C'	robust covariance matrix of mode amplitudes
D	matrix of surface deviations
I	identity matrix
N	matrix of basis functions
P	matrix of control point displacements
R	rotation matrix
U	left singular matrix
V	right singular matrix
X	data matrix
Σ	diagonal matrix of singular values

Tensors

C	characteristic tensor
K	convolution Kernel
\hat{P}	measured, normalized control point deviation tensor
\tilde{P}	approximated, normalized control point deviation tensor
P	estimated, non-normalized control point deviation tensor

1 Introduction

It is estimated that a third of global energy consumption is destined to manufacturing, U.S. Energy Information Administration (2021). Despite comprising production of essential goods for human activity such as food, textiles, electronic devices and machinery, manufacturing processes also account for almost a quarter of global carbon dioxide emissions, Ritchie and Roser (2020). This heavy environmental toll is likely to be further exacerbated as global manufacturing energy consumption is projected to increase by 50% till 2050, U.S. Energy Information Administration (2021). Such dramatic growth of energetic activity stays in direct conflict with several national environmental initiatives, which aim at net-zero emissions by the same year, leaving an urgent challenge for increasing energetic and environmental efficiency of manufacturing processes in the immediate future.

Several strategies are needed. Besides massive adoption of renewable energy sources, an increased effort has to be directed at elevating production efficiency, i.e., by reducing rejection rates, material scrap, product defects, and consequently energy requirements per production unit. Among several existing paths to achieve this, innovation of Computer-Aided Design (CAD) towards better understanding of causes, quantification and mitigation of manufacturing variability has progressively gained more attention from both academia and industry. This interest has been largely sparked by current developments in automated surface metrology, increased computational capacity, digital transformations in manufacturing, and most recently by accelerated rise of artificial intelligence. In this context, the present dissertation aims at pushing current boundaries of CAD into an increased awareness of manufacturing variability, particularly with novel ideas regarding quantification and estimation of geometric variability in early stages of product design.

This first chapter provides an introduction to the main dissertation ideas. First, the economic and environmental impact of manufacturing variability will be briefly addressed as contextualization for the problem relevance. Next, a literature review concerning estimation of geometric uncertainty will be presented, which summarizes current tendencies and opportunities from different research sectors. Finally, the dissertation objective will be formalized and a broad manuscript outline will be given.

1.1 The Cost of Manufacturing Variability

According to Creese (1999), manufacturing comprises “[...] integration of product demand, product design, material selection, manufacturing processing, product assembly and management to produce a desired product at a competitive price”. Because it involves such a long chain of human and physical interactions, manufacturing is prone to numerous uncertainty sources (e.g. variations in material properties or different operator skill-levels) which negatively affect the originally intended production outcome (e.g. reduced batch volume or increased machine downtime). Manufacturing variability thus carries large economic and environmental costs, which may be studied as two distinct effects: consequences due to inefficient manufacturing and consequences due to deviating part/component performance.

The economic burden of inefficient manufacturing has awoken such interest in engineering and management that it drove the development of new disciplines such as Six Sigma and Total Quality Management, van Grootel et al. (2019). Several industrial cases demonstrate that correct implementation of such management and statistical strategies may significantly reduce scrap rates, part rejection and rework rates, and thus provide substantial cost savings, Gupta et al. (2018); Gijo and Scaria (2014); and Shokri (2019). In contrast, environmental costs of manufacturing variability have gained significantly less attention, van Grootel et al. (2020). While environmental impact is the focus of policy making, its cost estimation usually follows a rather macroeconomic approach. For example, Egilmez et al. (2013) compare economic output vs. environmental load of several U.S. manufacturing sectors by introducing an eco-efficiency score as relative economic contribution of the sector divided by its relative, normalized environmental impact. This eco-efficiency score considers five environmental variables weighted such that a 100% score implies relative environmental effects being balanced-out by socioeconomic benefits. Based on this framework, the authors determine that aerospace product and part manufacturing needs to reduce energy use by 28% to achieve a 100% eco-efficiency score. Other sectors such as plastics and foundries even require energy reductions by 57% and 83%, respectively. Despite not being exclusively attributed to manufacturing variability, such figures clearly contextualize the role of manufacturing efficiency on mitigating environmental impact. This role is further magnified as current projections of global manufacturing energy consumption estimate a steady increase in the coming decades, Fig. 1.1, U.S. Energy Information Administration (2021). This diagram shows how both Energy Intensive (EI) manufacturing (raw goods such as plastics, chemicals or steel) and non-EI manufacturing (assembled goods such as computers, equipment, appliances or medicines) are expected to grow their total energy consumption by almost 100×10^{15} British Thermal Units (BTU) on 2050.

Manufacturing variability may not only affect energetic efficiency and economic prof-

itability for manufacturers, it may also induce undesired performance variations for the costumer. In this regard, aerospace literature exposes plenty examples of induced part variability due to geometric deviations. Högner et al. (2020) perform a comprehensive probabilistic, thermo-mechanical analysis of manufactured high-pressure turbine blades and reveal ± 100 MPa stress variations due to geometric deviations. Similarly, Bunker (2009) determines $\pm 40^\circ\text{C}$ temperature fluctuations on high-pressure turbine blades after only 10% deviation tolerance on cooling channel diameter. Regarding aero engine compressor, Flassig (2011) and Luo and Liu (2018) report $\pm 0.4\%$ pressure loss and $\pm 0.5\%$ adiabatic efficiency variations due to measured blade deviations, respectively. In addition to disturbing nominal component performance, the effects of manufacturing variability often propagate and interact throughout entire mechanical systems, posing a serious challenge for engineering design. For instance, van Grootel et al. (2020) estimate in a broader life-cycle analysis that an increase in manufacturing variability from 10% to 14% may cause up to three million USD in excess lifetime fuel consumption per Boeing 787, where variability is associated with safety and overdesign factors according to the technical standard ASTM F3114-15.

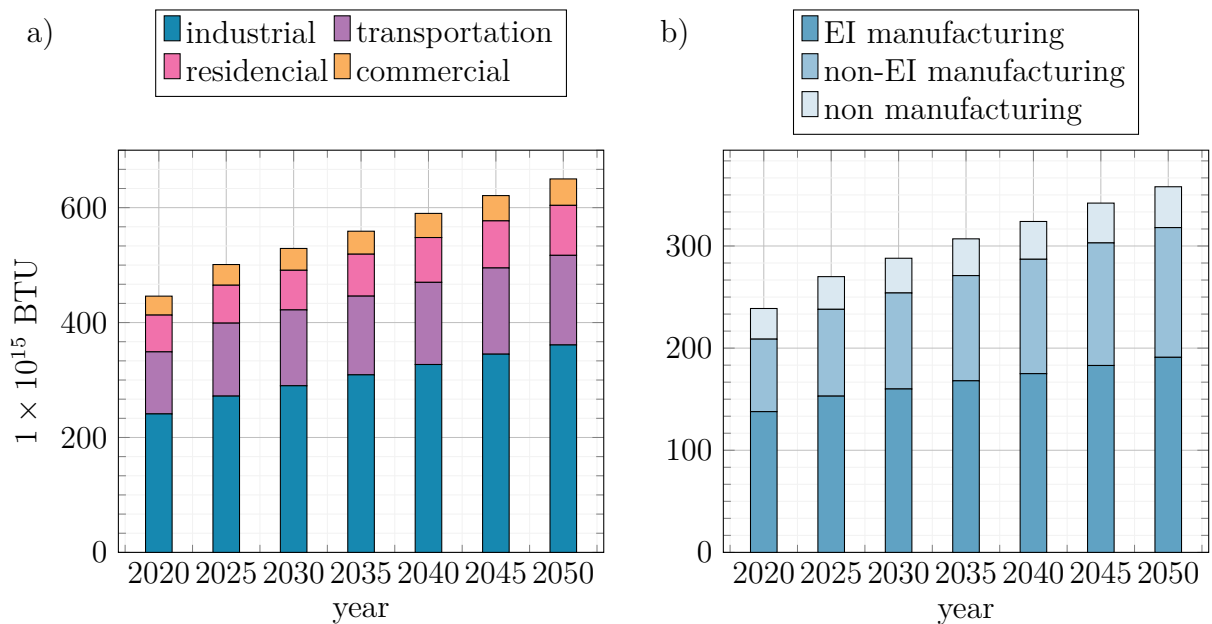


Figure 1.1: Global energy consumption (a) according to different sectors and (b) exclusively industrial sector. Acronyms in the List of Abbreviations.

All previous cases highlight the urgency to better understand and control manufacturing variability in the interest of both manufacturers and customers. Although lean management strategies may contribute to efficiency improvements of the product life-cycle, they need not be the only measure taken to address manufacturing variability. For instance, up to

now manufacturing knowledge remains remarkably ignored for supporting computer-aided design, much less influence geometry definition, Hedberg et al. (2017). This highlights a substantial opportunity to address manufacturing variability—in particular geometric deviations—directly in the design phase with better process understanding and modeling, such that potential deviations may be anticipated before manufacturing, and thus overall production output may be benefited. Current research directions regarding estimation of geometric deviations will be, therefore, the topic of the following section.

1.2 Estimating Geometric Uncertainty for Robust Design

Consideration of manufacturing information within computer-aided design is a very broad topic which has been approached differently depending on specific necessities of each industry sector. In particular, modeling of geometric deviations between manufactured and nominal design enables simulation of performance variability and thus has sparked considerable interest in industry sectors where high shape accuracy is critical for component performance, such as aerospace, automobile or optic industries. Despite successful geometric quantification and performance variability simulation on several literature examples, systematic consideration of geometric uncertainty during design is still far from becoming industrial standard. Most of these technical challenges may be grouped in three categories: CAD representation strategies for manufactured surfaces, probabilistic description of geometric variability and deviation estimation of new geometries.

Virtual representation of manufactured parts has seen a notorious development over the last decades. Motivated by the need of evaluating tolerance variability on mechanical assemblies, initial attempts to model manufacturing effects in CAD aimed at optimizing part tolerances using linear programming and Monte Carlo analysis to minimize assembly violations, e.g. Turner and Wozny (1987) and Turner (1988). Later, advances in surface metrology such as coordinate measurement machines and laser scanning facilitated access to detailed three-dimensional measurements of manufactured parts as dense point clouds, which inspired initial algorithms for B-spline and NURBS surface matching, e.g. Weir et al. (1996) and Yau (1999). With further developments in computation capacity, CAD reconstruction of complex manufactured shapes became increasingly accessible and powerful, even reaching commercial CAD software. Two major paths are currently preferred for translating point-cloud information into virtual representation: via discrete constructions such as mesh morphing or via continuous constructions such as parametric surfaces, Schleich and Wartzack (2017), Fig. 1.2. Mesh morphing essentially aims at deforming the nominal part represented as mesh to fit a series of measurement points at the surface while considering some type of volumetric displacement interpolation such as weighted averaging,

Franciosa and Gerbino (2011), or radial basis functions, Bobrowski et al. (2017). Since the manufactured geometry is directly represented as mesh, this method greatly simplifies the interface to product simulation, e.g. FE or CFD analysis, which in turn accelerates batch simulation by avoiding re-meshing. However, in cases where an analytical representation of manufactured surfaces is required, such as for design optimization, a continuous construction would be rather preferred. Buonamici et al. (2018) classify continuous CAD reconstruction methods as feature-based reconstruction strategies and free-form strategies, where the former attempts to retrieve entire parametric relationships and constraints from the measurement point cloud and the latter aims at fitting free-form surfaces with single or multiple patches.

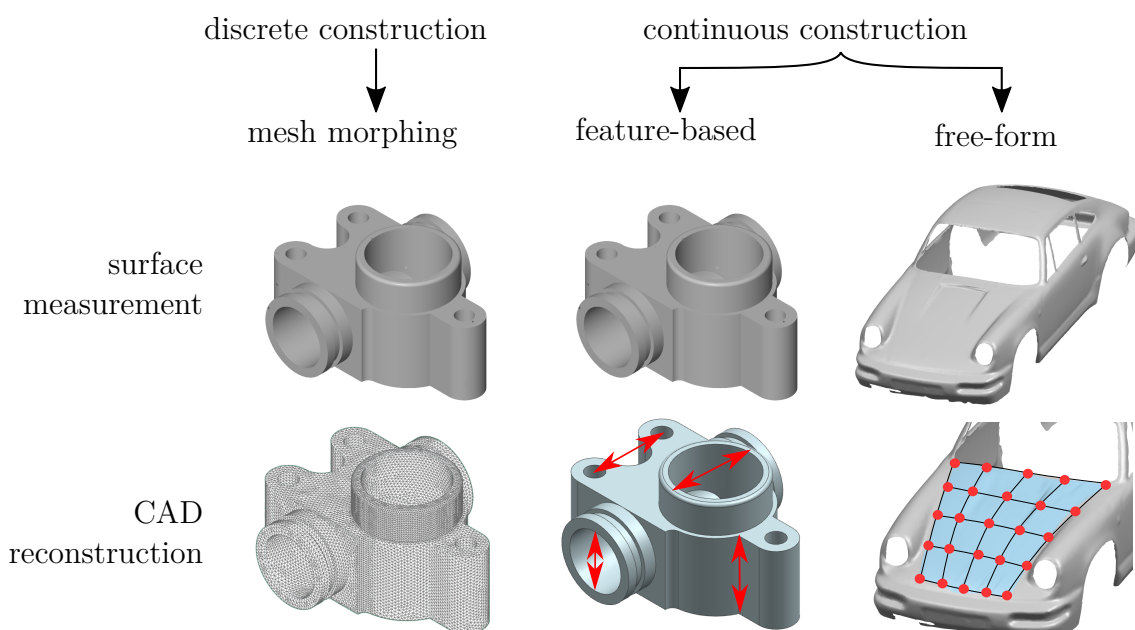


Figure 1.2: Examples of CAD reconstruction methods.

Commercial CAD software already offers general-purpose reconstruction capabilities, which is why industry-specific requirements will likely shape the evolution of future CAD reconstruction methodologies. Two canonical challenges from the aerospace industry in this regard are restricting a fixed parametrization when reconstructing a specific geometric feature from multiple manufacturing samples, and fully automated matching of complex free-form surfaces with wide curvature changes. A fixed parametrization for surface variability is a prerequisite to perform any statistical analysis such as PCA, and stays in conflict with typical commercial CAD reconstruction routines which frequently require additional surface rework and manual patching by the user. In the special case of airfoils, a noteworthy response to this issue has been the deviation parametrization developed by the TU Dresden for compressor and turbine blades, Lange et al. (2009)

and Heinze et al. (2014), which has been intensively exploited by industry and academia for robust blade optimization, Flassig (2011) and Kamenik et al. (2018); probabilistic CFD and FE analysis, Lange et al. (2012) and Högner et al. (2020); modal forcing variability quantification, Gambitta et al. (2021), etc. Reconstruction of more generic free-form surfaces—such as shanks and hot-gas faces, Fig. 1.3—may significantly challenge commercial CAD reconstruction capabilities due to rapid curvature gradients, which is why interest has been given to developing morphing strategies by direct manipulation of nominal B-spline surfaces. To obtain a CAD representation of aerospace components undergoing thermal deformations, Gaun et al. (2014) developed a method to transfer FE displacements to B-spline surface control points, which allows smooth manipulations of high-curvature CAD parts such as compressor blades or exhaust mixers. Inspired by these ideas, Urbano et al. (2019a) implemented B-spline morphing for manufactured surface fitting. In particular, it was demonstrated that re-parametrization as B-spline surface with a dense control point grid may significantly benefit matching accuracy especially for complex surfaces such as turbine shanks, Urbano et al. (2019b). Therefore, such research developments regarding automated matching of complex free-form surfaces exemplify new possibilities for improved parametrization control and high-curvature matching, and may thus signal potential development directions for commercial CAD software.

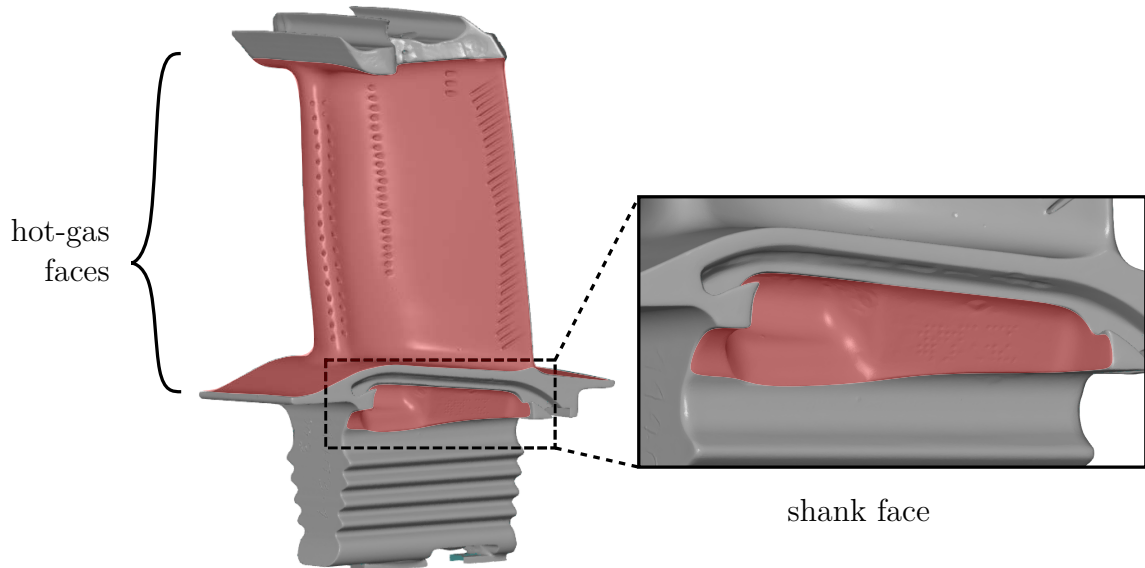


Figure 1.3: Complex free-form surfaces in a high-pressure turbine blade.

The second relevant research topic on geometric uncertainty estimation for product design deals with probabilistic representation of surface variability. This area is likely the most mature and exhibits visible consensus on both research and industry. Since manufacturing is a stochastic process, it is widely accepted that any rigorous robustness assessment must

be performed on a statistical basis. However, since surface measurements are frequently limited in quantity, an important question regards the generation of additional synthetic manufacturing samples which mimic real deviation patterns and thus enhance statistical estimation of performance variability. Two broad method classes appear to be dominant for this purpose: probabilistic modeling either directly from geometric parameters or rather from main deviation components.

Once a representative manufacturing sample has been matched as collection of CAD models, all recorded values for any given geometric parameter may be represented as histogram. Thus, the first approach consists of using a re-sampling scheme which considers individual parameter distributions and their statistical correlation, see e.g. Flassig (2011) and Högner et al. (2020). This is typically accomplished by an adaptive Latin Hypercube Sampling method such as restrictive pairing, where a sample is drawn from all geometric parameters using individually fitted probability density functions and the resulting correlation matrix is adapted incrementally to the measured correlation matrix, Dandekar et al. (2002).

In contrast, the second approach aims at decomposing parametric variability as linear superposition of characteristic deviation patterns via Principal Component Analysis (also named Singular Value Decomposition, Proper Orthogonal Decomposition or Statistical Shape Analysis). This results in unique amplitude histograms for each deviation pattern, e.g. Garzon and Darmofal (2003) and Lange et al. (2012). The sampling strategy then follows similarly as the first approach although using mode amplitudes instead of parameter histograms. As an advantage, this method illustrates the relative contribution of each deviation pattern into overall surface variability and may even allow elimination of non-representative deviation information, Urbano et al. (2019a). However, depending on the specific manufacturing scenario, linear superposition of deviation patterns may fail to capture inter-modal dependencies and may require non-parametric probabilistic representations such as kernel-density estimation, Lamb (2005), which are computationally intensive. Capturing complex non-linear deviation behaviors at reduced computational cost is thus the main challenge in probabilistic representation of geometric variability.

A promising alternative involves data compression strategies from machine learning, such as variational autoencoders, which are a type of neural network trained to retrieve information as non-linear function of a small number of latent variables, Kingma and Welling (2019). As these latent variables may be parametrized by any arbitrary multivariate probability distribution (e.g. standard multivariate normal distribution), generation of synthetic manufacturing instances becomes trivial, and thus the additional autoencoder training time may be partially compensated by the avoided adaptive sampling strategy. Autoencoders have successfully reduced geometric space for design optimization problems,

D’Agostino et al. (2018), and thus may constitute a powerful alternative for compressing non-linear geometric variability.

The third major research area deals with geometric prediction of surface deviations. Although CAD and probabilistic representations of manufacturing variability are useful for understanding the performance impact on a given design, they can hardly be applied to a new design if it is parametrized differently. As a result, detailed deviation information ends up being used mostly for performance impact assessment after the design has been fixed rather than for active variability mitigation in new designs (with the probably exception of airfoils). This leaves major room for opportunity, i.e., by modeling the relationship between surface deviations with nominal geometry and manufacturing process such that deviation estimations for new designs are enabled. In manufacturing literature, this topic is usually addressed as *deviation compensation* and has seen noticeable developments across optics, machining, and—most prominently—additive manufacturing. Initial analytical deviation models were proposed by Huang et al. (2014) for cylindrical additive manufacturing probes. Based on a nonlinear deviation function in cylindrical coordinates, the authors defined a Bayesian model for manufacturing deviations, trained it on multiple cylindrical probes of different diameters, and later used it to compensate deviations from prismatic probes. Despite not accurately predicting radial deviation profiles, mean radial deviation was reduced by a factor of four. An analogous research was performed by Poniatowska (2015) for free-form surface machining, where ten 50 mm × 50 mm aluminum surfaces were milled, measured and later fitted as NURBS surfaces. The mean deviation component between nominal and manufactured surfaces was then used as compensation for ten new corrected samples, which later reported a five-fold average decrease in deviation amplitude when compared to non-compensated surfaces.

Further developments in additive manufacturing refine probabilistic deviation modeling and introduced machine learning strategies. Zhu et al. (2019) demonstrate a comprehensive approach for probabilistic modeling and deviation transfer for cylindrical additive manufacturing probes. The authors perform a design of experiments with 18 cylindrical samples of different radius, layer thickness and scan speed, and observe three relatively simple dominant deviation patterns after additive manufacturing simulations. Gaussian process regression is then implemented to model the relationship between associated mode amplitudes and individual probe parameters, which then correctly predict deviation behavior on two validation probes. Taking a further step, Zhu et al. (2020) enhance the previous deviation modeling approach by generating 1200 additional synthetic deviation samples from the Gaussian process, and use them to train a convolutional neural network for predicting deviation fields from arbitrary parameter and geometry combinations. As previous examples demonstrate, deviation prediction is steadily gaining ground for manu-

facturing of relatively simple geometries. To enable deviation estimation for more complex mechanical parts in the future, coupling of machine learning methods with manufacturing simulations—such as thermo-mechanical analysis of additive, machining or casting processes—seem to be definitely a valuable research direction.

Estimation of geometric deviations for robust design is a multidisciplinary problem which involves accurate and scalable CAD reconstruction techniques, efficient probabilistic representations and physics-based mathematical modeling of geometric variability. Although progress on these areas has been fundamentally shaped by individual needs of industry sectors, it is undeniable that huge advantages have been gained by adopting methods from external disciplines, as exemplified by the vast appropriation of Principal Component Analysis and—more recently—machine-learning methods. Thus, a closer cooperation and exchange of ideas between different industry sectors, such as additive manufacturing, aerospace, and CAD software development—may significantly propel geometric deviation estimation to progressively gain ground as an important industrial standard in a foreseeable future.

1.3 Problem Statement and Thesis Outline

Consideration of geometric variability within product design is indispensable for increasing manufacturing efficiency, and thus minimize economic and environmental impact for both manufacturer and customer. However, in contrast to product design, which comprises mainly a coordination between CAD and simulation software, systematic inclusion of geometric variability into product design involves surface metrology, probability, machine learning modeling plus innovative CAD methods. Currently, some of these processes are available only for special geometry types, such as prismatic specimens or airfoils, or require extensive user interaction, such as CAD fitting of complex surfaces with commercial software. Therefore, the only way to leverage geometric variability as new industrial standard is by automating knowledge extraction from manufacturing. This involves two major automation chains: a process for fitting surface measurements and storing them in a database of probabilistic CAD representations, and a process for using such deviation information for estimating deviations of new geometries.

The present dissertation aims at providing innovative solutions on both fronts. Firstly, a generic CAD surface matching procedure will be introduced, followed by a probabilistic representation of manufacturing variability. However, to allow automation on these areas, several innovative strategies for mitigating measurement noise, CAD inconsistencies and statistical outliers will be introduced. Secondly, since deviation estimation has been largely investigated for simplified geometries, the present dissertation aims at developing a

methodical framework for predicting stochastic geometric deviations for rather complex, generic free-form surfaces. This will be achieved using methods of machine learning, which allow association of surface variability with nominal geometry and identification of common deviation patterns between distinct manufactured designs.

The remaining of the dissertation is structured in four parts. First, the mathematical background of CAD surface parametrization and measurement will be described in Chapter 2, where special emphasis will be given to B-spline surfaces as they will lay the foundation for CAD surface matching and deviation modeling. Subsequent sections will be devoted to explain the subtractive CAD construction approach, which conveniently extends surface matching capabilities from a single face to multiple faces, and to describe the optical principles of surface metrology with structured light. The second major part described in Chapter 3 addresses the process for accurately representing manufactured surfaces as CAD geometry using B-spline morphing. Multiple strategies for automated mitigation of measurement noise and CAD irregularities throughout the morphing process will be provided. These methods will be then demonstrated in a surface matching example considering complex free-form faces of a high-pressure turbine blade. Part three introduced in Chapter 4 deals with probabilistic methods for surface variability once multiple fitted CAD surfaces are available. A probabilistic description of geometric variability based on linear superposition of main deviation patterns—also referred as *manufacturing modes*—will be introduced. Further, a statistic-geometric criterion for identification of observable manufacturing modes will be presented, as well as a statistical approach to remove inconsistent CAD surface outliers from a sample. Two application sections on turbine blades will demonstrate the high fidelity of synthetic deviation instances obtained from this probabilistic approach and also highlight differences in statistical deviation behavior across different turbine blade designs.

The last contribution is presented in Chapter 5 which concerns probabilistic deviation estimation for new geometric designs. Given the technical scope of this chapter, a simplified conceptual background will be introduced first. Then, a deviation model associating surface variations with nominal geometry will be developed using the convolution operation. To estimate deviations on a new geometry, the concept of recurrent deviation behavior repeatedly observed across multiple manufactured designs will be formalized as *deviation law*. Subsequent sections will describe the process of extraction and optimization of deviation laws from different databases of manufactured geometries. An application example will be provided, which illustrates deviation estimation for a turbine shank based on surface measurements of previous turbine blade designs. The integration of deviation laws into a robust design process will be briefly addressed in the last section. To conclude, a brief summary of findings and potential research directions will be provided in Chapter 6.

2 CAD Surface Parametrization and Measurement

Surface deviations consist of discrepancies between design intent and manufactured result. Nominal design intent may be usually represented as CAD geometry, which fundamentally consists of analytical surfaces. Representation of a manufactured part, on the other side, may differ according to the applied metrology approach, but typical examples may include selective point measurements or dense surface point clouds. Therefore, any attempt to quantify surface variability necessitates a thorough understanding of the underlying mathematical description of both CAD and manufactured objects, which is the main purpose of the present chapter. After a general description of CAD surface parametrization approaches, where special emphasis will be given to B-spline surfaces, useful geometric properties of B-spline surfaces such as local orientation and curvature will be derived. Then, a brief description of subtractive parametric CAD construction approaches will be given, and lastly an introduction to experimental surface measurement with structured light will be provided.

2.1 CAD Surface Parametrization

The design, manufacturing and analysis of any mechanical part rely on well-defined surface parametrization. On the most fundamental level, any CAD face may be parametrized either as primitive or free-form face. As the name suggests, primitive faces are generated from elementary geometric shapes such as planes, cylinders, spheres, cones or tori. They are pretty common in mechanical design, for instance as part of firtree grooves or as multiple cylindrical blends of an HPT blade, Fig. 2.1. Free-form faces on the other side, exhibit complex three-dimensional shapes such as HPT airfoil, hub-endwall, fillet radius and shank walls which, therefore, calls for more elaborate parametrization strategies.

Parametrization of free-form faces is mainly determined by design intent and, therefore, extremely diverse. For instance, the fillet radius serves as a smooth transition between two almost perpendicular faces (hub-endwall and airfoil) and may be best described by profile and sweep curves. On the other hand, the airfoil dictates the aerodynamics of the HPT

blade and may be better described by physical parameters such as stagger angle, leading edge radius, maximum thickness, etc. The choice of parametrization also influences the part’s optimization potential. For instance, an appropriate parametrization can rapidly reveal high-efficiency regions in the design space. Despite of the many parametrization possibilities for free-form surfaces, a standardized geometry format is needed to enable data exchange between different CAD and engineering simulation software. This has led to the development of standarized CAD formats, such as IGES, U.S. Product Data Association (1996), and STEP, ISO (2011). These standards enable the translation of any free-form surface, regardless of which parametrization was initially chosen, into its most fundamental mathematical descriptions, i.e., B-spline or Non-Uniform Rational B-Spline (NURBS) surfaces.

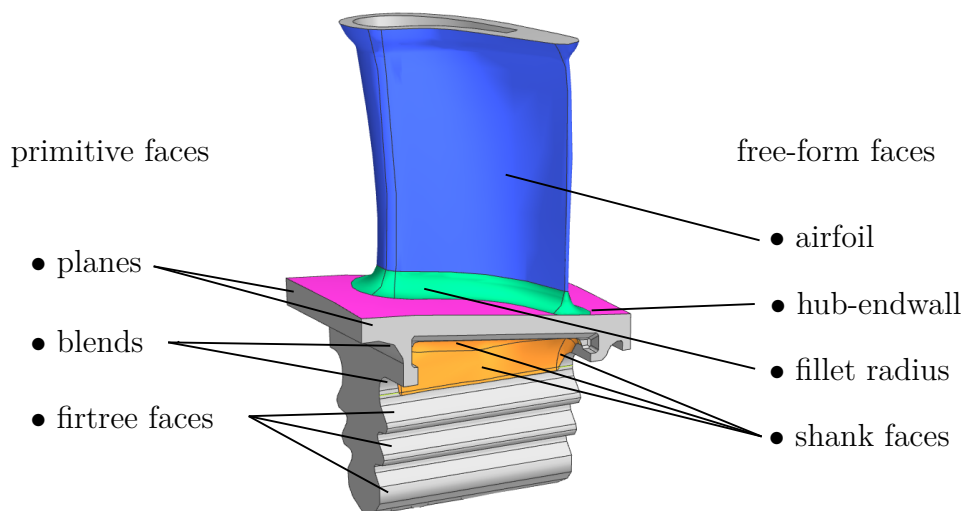


Figure 2.1: Examples of primitive and free-form faces of an HPT blade.

According to Piegl and Tiller (1995), a B-spline curve

$$\mathbf{c}(u) = \sum_{i=0}^n N_{i,q}(u) \mathbf{p}_i, \quad u \in [0, 1] \subset \mathbb{R}, \quad (2.1)$$

is a parametric curve, which weights control points \mathbf{p}_i with q -th order basis functions $N_{i,q}(u)$ evaluated at curve parameter u . The B-spline curve is associated with a non-decreasing knot vector

$$\bar{\mathbf{u}} = \underbrace{[0, \dots, 0]}_{q+1}, \bar{u}_{q+1}, \dots, \bar{u}_{r-q-1}, \underbrace{[1, \dots, 1]}_{q+1}]^T \in \mathbb{R}^{r+1}, \quad \bar{u}_i \leq \bar{u}_{i+1}, \quad r = (n + 1) + q, \quad (2.2)$$

defined according to the number $n + 1$ of control points. If knot values $\bar{u}_q, \dots, \bar{u}_{r-q}$ are equally spaced, the curve is regarded as *uniform*, otherwise as *non uniform*. The basis

functions $N_{i,q}(u)$ share the index i with control points \mathbf{p}_i and are defined recursively as

$$N_{i,0}(u) = \begin{cases} 1 & \text{for } \bar{u}_i \leq u < \bar{u}_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \quad (2.3)$$

$$N_{i,q}(u) = \frac{u - \bar{u}_i}{\bar{u}_{i+q} - \bar{u}_i} N_{i,q-1}(u) + \frac{\bar{u}_{i+q+1} - u}{\bar{u}_{i+q+1} - \bar{u}_{i+1}} N_{i+1,q-1}(u), \quad (2.4)$$

where the value $N_{i,q}(u)$ is interpolated from lower-order values $N_{i,q-1}(u)$ and $N_{i+1,q-1}(u)$ in relation to the position of the curve parameter u .

As an example, consider a B-spline curve in the xy -plane with order $q = 3$ and $n + 1 = 6$ arbitrary control points $\mathbf{p}_i \in \mathbb{R}^2$, Fig. 2.2b. The curve is associated with a uniform knot vector

$$\bar{\mathbf{u}} = [0, 0, 0, 0, 1/3, 2/3, 1, 1, 1, 1]^\top \quad (2.5)$$

having $r + 1 = (6 + 3) + 1 = 10$ components. To compute curve $\mathbf{c}(u)$ all basis functions $N_{i,q}(u)$ must be first evaluated. According to Eq. (2.3), the zero-order basis functions are non-zero for $u \in [\bar{u}_i, \bar{u}_{i+1})$, $i = 3, 4, 5$, respectively, Fig. 2.2a. Next, first-order basis functions $N_{i,1}(u)$ are computed by weighting values $N_{i,0}(u)$ and $N_{i+1,0}(u)$ according to Eq. (2.4) resulting in triangular functions. The B-spline curve $\mathbf{c}(u)$ in Fig. 2.2b results from multiplication of final basis functions $N_{i,3}(u)$ with respective control points \mathbf{p}_i according to Eq. (2.1).

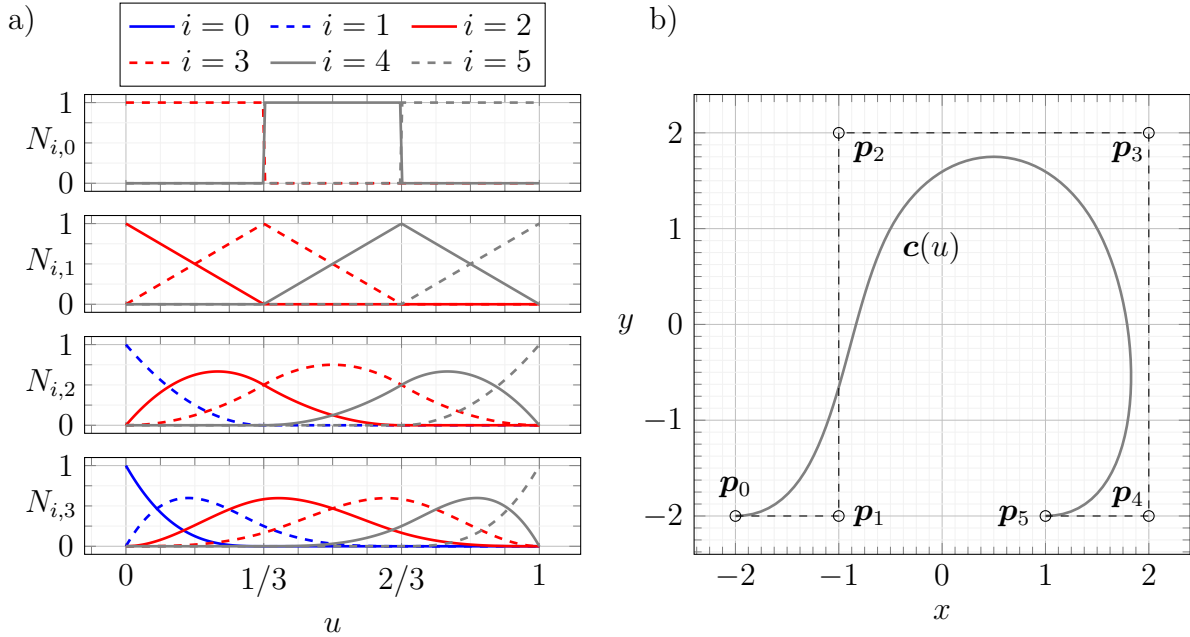


Figure 2.2: Representation of B-spline by a) basis functions $N_{i,q}(u)$ resulting in b) curve $\mathbf{c}(u)$ with six respective control points \mathbf{p}_i (o).

One may extend the concept of B-spline curve to a B-spline surface by using a tensor product of two B-spline curves with two surface parameters u and v . Thus, the B-spline surface

$$\mathbf{s}(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,q}(u) N_{j,q}(v) \mathbf{p}_{i,j}, \quad (u, v) \in [0, 1]^2 \subset \mathbb{R}^2 \quad (2.6)$$

may be defined for a $(n + 1) \times (m + 1)$ grid of control points, which are weighted by basis functions $N_{i,q}(u)$ and $N_{j,q}(v)$. Although a generic B-spline surface definition may assign different basis function orders to parameters u and v , Piegl and Tiller (1995), the present formulation assigns same order q for simplicity. Thus, Eqs. (2.3) and (2.4) are equally valid for both curve parameters u and v . Furthermore, in addition to associated knot vector (2.2), surface (2.6) requires also a knot vector $\bar{\mathbf{v}}$ which may be defined in analogy to Eq. (2.2) as

$$\bar{\mathbf{v}} = [\underbrace{0, \dots, 0}_{q+1}, \bar{v}_{q+1}, \dots, \bar{v}_{s-q-1}, \underbrace{1, \dots, 1}_{q+1}]^T \in \mathbb{R}^{s+1}, \quad \bar{v}_i \leq \bar{v}_{i+1}, \quad s = (m + 1) + q. \quad (2.7)$$

Although knot vectors or surface degree may be used to control the shape of B-spline surfaces, a more efficient surface manipulation is obtained from control point number and position. This is due to the smooth surface behavior after control point variations, which suits design of aerodynamic faces, such as the hub-endwall of an HPT blade in Fig. 2.3.

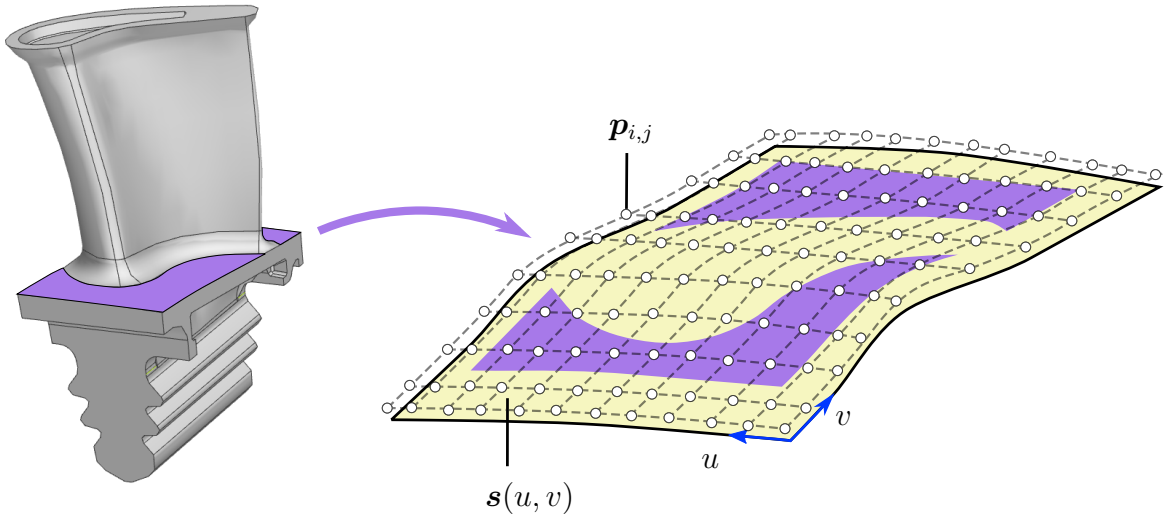


Figure 2.3: Exemplary B-spline surface parametrization for an HPT blade hub-endwall before trimming.

Despite the mathematical simplicity of B-spline surfaces, they are incapable of exactly representing regular geometric shapes, such as cylinders or cones. This has led to a further

surface generalization called Non-Uniform Rational B-Splines (NURBS):

$$\mathbf{s}(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{i,j} \mathbf{p}_{i,j} N_{i,q}(u) N_{j,q}(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{i,j} N_{i,q}(u) N_{j,q}(v)}, \quad (u, v) \in [0, 1]^2 \subset \mathbb{R}^2, \quad (2.8)$$

which basically consists of a rational formulation of basis functions $N_{i,q}(u)$ with additional weight factors $w_{i,j}$.

Both B-spline and NURBS surfaces have been extensively developed and efficiently implemented in CAD software. In particular, automatic re-parametrization of any CAD face as B-spline surface offers important capabilities. As an example, consider the re-parametrization of a half-cylinder as B-spline surface with different numbers of control points in Fig. 2.4. In theory, such a shape would never be possible to be exactly constructed using B-spline surfaces. However, because an elevated number of control points is nowadays numerically inexpensive, a high-fidelity B-spline surface approximation is possible. This is demonstrated in Fig. 2.4, where a finer parametrization reduces surface deviations to the cylinder from $\pm 5 \mu\text{m}$ to below $\pm 1 \mu\text{m}$. For simplicity and versatility, all methods developed in the present work are exclusively formulated for B-spline surfaces, with the fair assumption that any CAD face may, to a great level of accuracy and with minor user involvement, be re-parametrized in such a way.

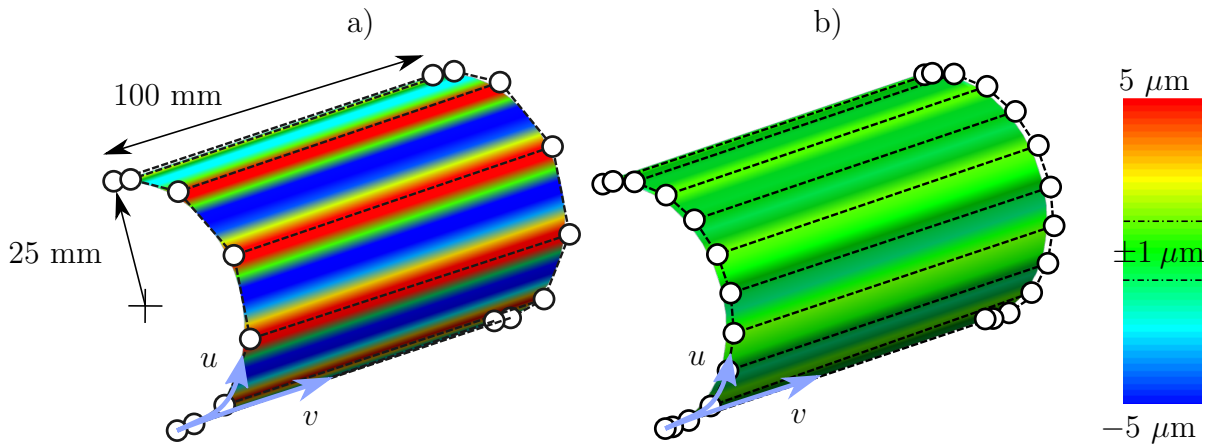


Figure 2.4: Error of B-spline surface re-parametrizations of half-cylinder with a) 8×2 and b) 13×2 control point grids.

2.2 Geometric Properties of B-Spline Surfaces

Throughout this work, geometric properties such as local orientation and curvature will repetitively arise in the formulation of B-spline morphing and geometric deviation modeling. Depending on available information, an estimation of these properties may be derived directly for a B-spline surface position (u, v) or alternatively for a specific control point $\mathbf{p}_{i,j}$. To evaluate orientation and curvature for a position (u, v) , consider Fig. 2.5.

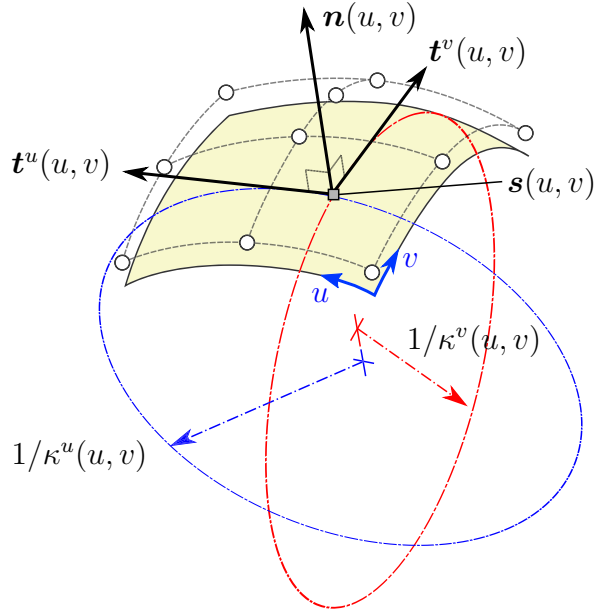


Figure 2.5: Orientation and curvature definitions for a B-spline surface.

At position (u, v) , tangential vectors of surface (2.6) are defined as directional derivatives

$$\mathbf{t}^u(u, v) := \frac{\partial \mathbf{s}}{\partial u}(u, v) \quad \text{and} \quad \mathbf{t}^v(u, v) := \frac{\partial \mathbf{s}}{\partial v}(u, v), \quad (2.9)$$

where the expressions for B-spline derivatives may be found analytically, see Piegl and Tiller (1995), or from numerical libraries for computational geometry, e.g. Bingol and Krishnamurthy (2019). These tangential vectors may be then used to compute the unit normal vector as

$$\mathbf{n}(u, v) := \frac{\mathbf{t}^u(u, v) \times \mathbf{t}^v(u, v)}{\|\mathbf{t}^u(u, v) \times \mathbf{t}^v(u, v)\|}. \quad (2.10)$$

To formulate an expression for curvature, consider first an arbitrary parametric curve $\mathbf{p}(s)$, $s \in [0, 1] \subset \mathbb{R}$. According to Gray et al. (2006), the curvature κ at position s may be

obtained as

$$\kappa(s) = \frac{\left\| \frac{d\mathbf{p}}{ds} \times \frac{d^2\mathbf{p}}{ds^2} \right\|}{\left\| \frac{d\mathbf{p}}{ds} \right\|^3}, \quad (2.11)$$

which has dimension $[L]^{-1}$ and in the case of a circle results in $\kappa = 1/R$. This relationship may be also applied to B-spline surfaces in each direction independently. For example, the directional curvature in u -direction results in

$$\kappa^u(u, v) = \frac{\left\| \frac{\partial \mathbf{s}(u, v)}{\partial u} \times \frac{\partial^2 \mathbf{s}(u, v)}{\partial u^2} \right\|}{\left\| \frac{\partial \mathbf{s}(u, v)}{\partial u} \right\|^3}, \quad (2.12)$$

and analogously for $\kappa^v(u, v)$. Now, there exist several curvature formulations which combine directional curvatures into a single representative curvature value associated with a surface point. One possibility is the Gauß curvature defined as the product of both directional curvatures

$$\kappa(u, v) := \kappa^u(u, v) \kappa^v(u, v) \quad (2.13)$$

having dimension $[L]^{-2}$. In the case of a sphere, multiplication of both directional curvatures at any surface point yields a Gauß curvature $\kappa = 1/R^2$.

When performing operations with control points such as control point selection for B-spline morphing (Sec. 3.3) or geometric deviation laws (Sec. 5.1), estimations of local orientation and curvature associated with a specific control point $\mathbf{p}_{i,j}$ may be very useful. In this case, above relationships are useless since control points $\mathbf{p}_{i,j}$ are no surface positions. An alternative approach consists in interpolating any discrete control point row by a parametric curve $\mathbf{p}(s)$, Fig. 2.6b, and deriving orientation and curvature relationships from this.

Let the first and second order derivatives of the parametric curve $\mathbf{p}(s)$ may be approximated by central finite differences as

$$\frac{d\mathbf{p}}{ds} \approx \frac{\mathbf{p}(s+h) - \mathbf{p}(s-h)}{2h} \quad \text{and} \quad \frac{d^2\mathbf{p}}{ds^2} \approx \frac{\mathbf{p}(s+h) - 2\mathbf{p}(s) + \mathbf{p}(s-h)}{h^2}, \quad (2.14)$$

where h corresponds to a small parametric spacing of curve parameter s . Previous derivatives may also be evaluated at fixed control point positions of a B-spline surface in Fig. 2.6a as

$$\frac{d\mathbf{p}_{i,j}}{ds} \approx \frac{\mathbf{p}_{i,j+1} - \mathbf{p}_{i,j-1}}{2h} \quad \text{and} \quad \frac{d^2\mathbf{p}_{i,j}}{ds^2} \approx \frac{\mathbf{p}_{i,j+1} - 2\mathbf{p}_{i,j} + \mathbf{p}_{i,j-1}}{h^2}, \quad (2.15)$$

where the derivatives have been taken along the j -direction and the parametric spacing has been assumed to be constant as $h = 1/m$, where $m + 1$ equals the number of control

points along j -direction. The definition of a unit tangential vector for the v -direction at position (i, j) may be then formulated as

$$\mathbf{t}_{i,j}^v = \frac{d\mathbf{p}_{i,j}}{ds} / \left\| \frac{d\mathbf{p}_{i,j}}{ds} \right\| \approx \frac{\mathbf{p}_{i,j+1} - \mathbf{p}_{i,j-1}}{\|\mathbf{p}_{i,j+1} - \mathbf{p}_{i,j-1}\|} \quad (2.16)$$

being independent of the parametric spacing h . The calculation of the tangential vector $\mathbf{t}_{i,j}^u$ in u -direction follows by analogy. Tangential vectors may be then used to approximate the unit normal vector (2.10) as

$$\mathbf{n}_{i,j} = \frac{\mathbf{t}_{i,j}^u \times \mathbf{t}_{i,j}^v}{\|\mathbf{t}_{i,j}^u \times \mathbf{t}_{i,j}^v\|}. \quad (2.17)$$

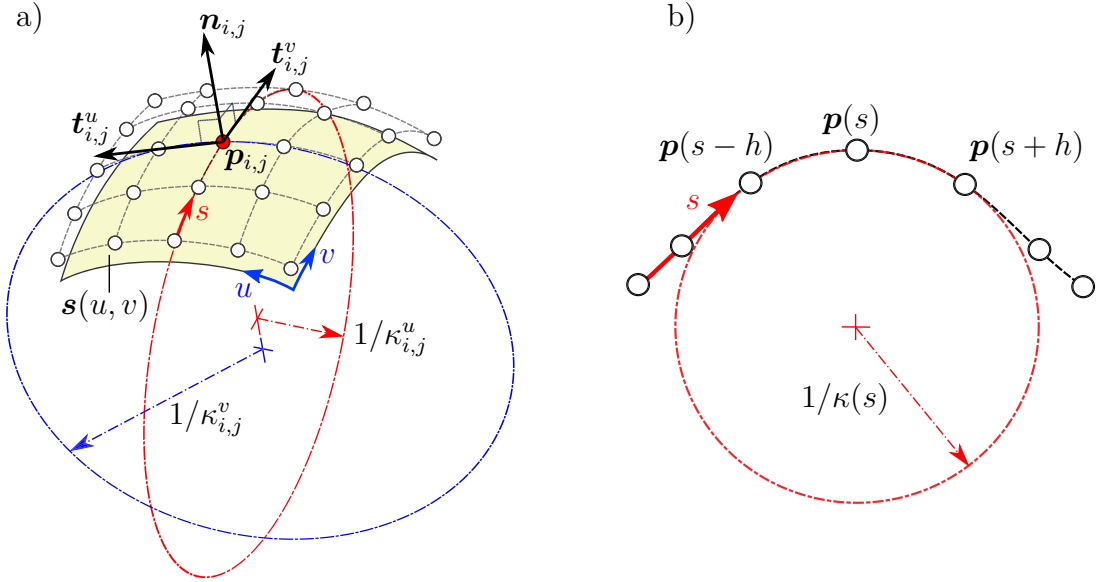


Figure 2.6: Illustration of a) orientation and curvature definitions based on control point grid and b) parametric curve $\mathbf{p}(s)$ from a control point row.

Lastly, the parametric curve $\mathbf{p}(s)$ may be used to estimate curvature from discrete control point rows. By replacing derivative approximations (2.15) in (2.11), a directional curvature approximation for point $\mathbf{p}_{i,j}$ may be obtained as

$$\kappa_{i,j}^v \approx \frac{\|\mathbf{p}_{i,j+1} - \mathbf{p}_{i,j-1}\|/2h \times \|\mathbf{p}_{i,j+1} - 2\mathbf{p}_{i,j} + \mathbf{p}_{i,j-1}\|/h^2}{\|\mathbf{p}_{i,j+1} - \mathbf{p}_{i,j-1}\|^3/8h^3} \quad (2.18)$$

$$= 4 \frac{\|(\mathbf{p}_{i,j+1} - \mathbf{p}_{i,j-1}) \times (\mathbf{p}_{i,j+1} - 2\mathbf{p}_{i,j} + \mathbf{p}_{i,j-1})\|}{\|\mathbf{p}_{i,j+1} - \mathbf{p}_{i,j-1}\|^3}, \quad (2.19)$$

which is again independent of the parametric spacing h . Since this expression was derived for a control point row in j -direction, it results in a directional curvature $\kappa_{i,j}^v$. An analogous procedure may be performed to obtain directional curvature $\kappa_{i,j}^u$ from any control point row in i -direction. A unified curvature associated to point $\mathbf{p}_{i,j}$ may then be defined analogously to the Gauß curvature (2.13) as

$$\kappa_{i,j} := \kappa_{i,j}^u \kappa_{i,j}^v. \quad (2.20)$$

The reader is finally remarked about notation differences in orientation and curvature quantities. Tangential, normal and curvature quantities evaluated for B-spline surface points are distinguished by the arguments (u, v) , e.g. $\mathbf{t}^u(u, v)$, $\mathbf{n}(u, v)$, $\kappa(u, v)$, whereas quantities associated with control point positions carry the subindices i, j , e.g. $\mathbf{t}_{i,j}^u$, $\mathbf{n}_{i,j}$, $\kappa_{i,j}$.

2.3 Parametric Construction of CAD Parts

Complex CAD parts require up to hundreds of geometric parameters, many of which will be iterated dozens of times before final geometry definition. However, due to intrinsic geometric interactions, small parameter modifications may frequently lead to impossible geometric constructions—e.g. conflicting body subtractions or open surfaces—thus hindering fast CAD design explorations. Although a complete overview of parametric constraints may be infeasible in practice, a robust parametric geometry workflow with a clear operation hierarchy may to a great extent reduce CAD reconstruction failures.

A possible strategy consists in implementing a subtractive construction approach which loosely mimics the real manufacturing sequence. As illustration, consider an exemplary CAD construction workflow for an HPT blade foot shown in Fig. 2.7. While the airfoil geometry will be typically constrained by aerodynamic requirements, the blade foot may be entirely constructed by subtraction operations starting from a solid block attached at the bottom (Fig. 2.7a). Each subtraction step is simply defined by a subtraction body (Fig. 2.7b) which contains the parametrization of the faces produced by subtraction, such that after multiple sequential subtractions the desired geometry is finally obtained (Fig. 2.7d). In this manner, any parameter modification will automatically re-run the entire subtraction sequence and produce the desired CAD part, provided that no conflicts between different subtraction operations exist.

In addition, to enable fast CAD design iterations, this approach leverages a single construction operation, i.e., body subtraction, to generate the entire geometry. This characteristic may be a powerful tool for easily scaling manufacturing surface representations throughout multiple CAD faces. As thoroughly explained in the next chapter, CAD rep-

resentation of any manufactured surface may be conveniently accomplished by replacing subtraction bodies with one that has been previously fitted to a surface measurement. By replacing subtraction bodies, this approach allows CAD representation of surface variations to any arbitrary number of faces.

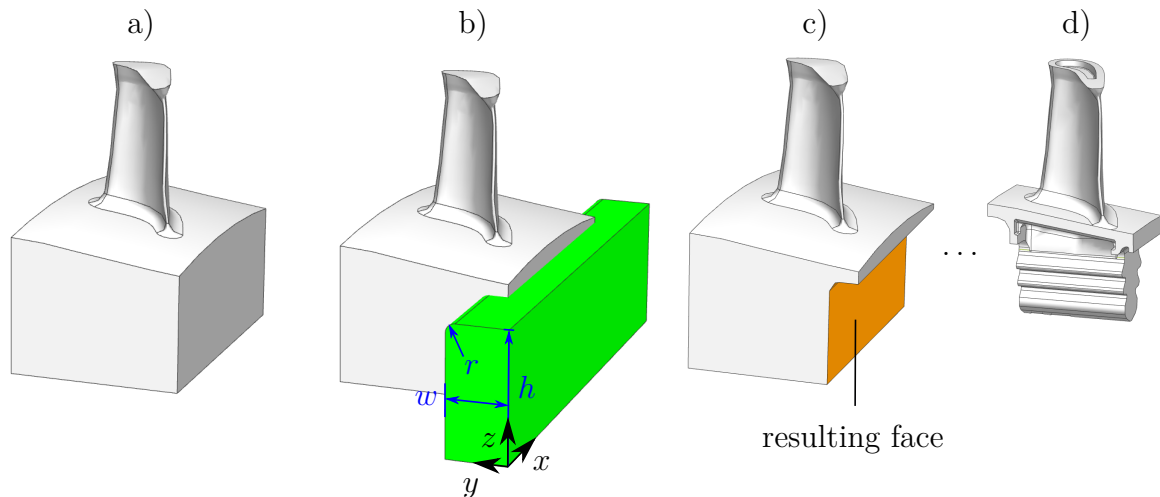


Figure 2.7: Subtractive construction approach for a blade root: a) initial geometry, b) subtraction body with face parametrization, c) resulting face and d) final geometry after several subtraction steps.

2.4 Surface Measurement with Structured Light

Previous sections laid the foundation of nominal geometry parametrization and construction in CAD. To assess manufacturing accuracy and quality, surface measurements of real manufactured samples have to be considered now. As a basis, the present section will explore surface measurement with structured light, in particular, how manufactured surfaces may be digitized as dense point clouds using photogrammetric models.

There exist multiple approaches to assess manufacturing accuracy based on surface measurement. Coordinate-measurement machines are flexible regarding surface and lighting conditions but their measurement speeds may be limited depending on part geometry, Li and Gu (2004). When internal geometric features are of interest, such as turbine blade cooling holes, industrial computed tomography is preferred, where a continuous volumetric representation of the part is constructed from x -ray irradiation, Villarraga-Gómez et al. (2019). An attractive compromise is obtained by surface measurement with structured light, where the object surface is reconstructed as dense point cloud by projecting light patterns from different angles and calculating reflected ray paths, Peng and Gupta (2007). Despite requiring careful part preparation and setup, important efforts have been pursued

in measurement automation and instrument portability. Moreover, compared to sample points or scan volumes, surface point clouds are most useful for re-engineering CAD surfaces.

In the most basic form, a structured light measurement setup consists of a light pattern projector, a reference plane and a camera, Fig. 2.8. The position of projector \mathbf{x}_P , camera \mathbf{x}_C and reference plane with reference point \mathbf{x}_R and normal vector \mathbf{n}_R are known and given with respect to a global coordinate system $\{O; x, y, z\}$. The camera orientation is given by a known rotation matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ defining the coordinate transformation between camera coordinate system $\{O'; x', y', z'\}$ and global coordinate system as

$$[x', y', z']^\top = \mathbf{R} [x, y, z]^\top. \quad (2.21)$$

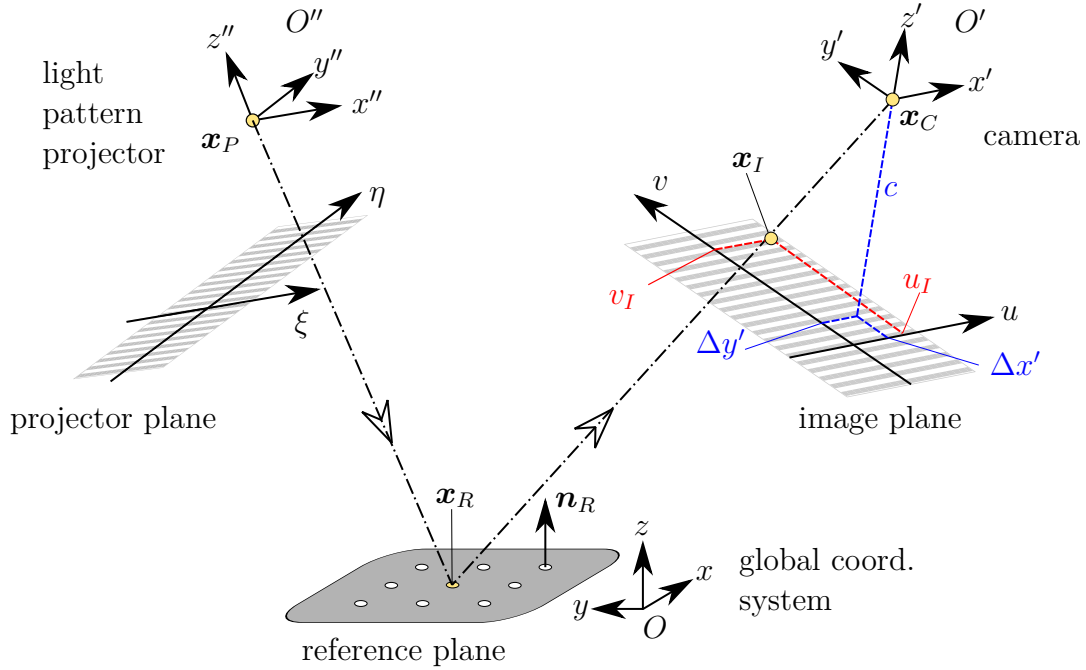


Figure 2.8: Schematic representation of the camera-projector model.

Light is projected from point \mathbf{x}_P through the projector plane onto the reference plane and reflected towards the camera center \mathbf{x}_C to be captured on the image plane by image coordinates (u_I, v_I) . The uv -plane is commonly parallel to the $x'y'$ -plane intersecting the z' -axis at distance $z' = -c$. Due to internal camera distortion, the image plane center is typically shifted from its nominal position by displacements $(\Delta x', \Delta y')$ in the camera coordinate system, Luhmann et al. (2019). The image plane intersection $(-c)$ and the distortion parameters $(\Delta x', \Delta y')$ are properties of the camera and may be obtained by

calibration procedures, e.g. Brown (1971). According to the collinearity principle, the camera center \mathbf{x}_C and a reference point \mathbf{x}_R intersect at a unique point \mathbf{x}_I in the image plane. For a calibrated camera with identified parameters, the three-dimensional coordinates \mathbf{x}_I of any (u_I, v_I) point on the image plane may be computed in the global coordinate system O as

$$\mathbf{x}_I = \mathbf{x}_C + \mathbf{R}^{-1} \begin{bmatrix} u_I - \Delta x' \\ v_I - \Delta y' \\ -c \end{bmatrix} \quad \text{where} \quad \mathbf{R}^{-1} = \mathbf{R}^\top. \quad (2.22)$$

The main goal of surface photogrammetry is to reconstruct object coordinates from their (u, v) image location. For instance, consider the measurement setup shown in Fig. 2.9, where object point \mathbf{x}_M is associated with image location (u_1, v_1) . Even with identified camera calibration parameters and user-defined scaling factors, additional information is still required to compute depth from the photograph. This is the reason why fringe patterns are used. In this method, each light ray in the fringe pattern is labeled with a unique phase value which remains unchanged after reflection. The projected phase values are defined on the projector plane as a function $\Phi(\xi, \eta)$ and, once reflected and distorted by the object surface, they are captured one by one on the image plane as camera phase values $\Psi(u, v)$. Thus, identification of identical phase values on the projector plane (ξ, η) and on the image plane (u, v) enables reconstruction of a light ray's path, and consequently allows computation of the reflection point as intersection of incident and reflected rays. For instance, point \mathbf{x}_M in Fig. 2.9 is the intersection of projected light ray $\mathbf{x}_P + \lambda(\mathbf{x}_2 - \mathbf{x}_P)$ and reflected light ray $\mathbf{x}_C + \mu(\mathbf{x}_1 - \mathbf{x}_C)$.

In this sense, the goal of structured light measurement consists on identifying a map

$$M : (\xi, \eta) \rightarrow (u, v) \quad (2.23)$$

which describes the re-organization of phase values between projector and image planes, thus allowing detailed surface reconstruction of the object using geometric relationships. The initial step consists on defining the projected phase map as a monotonic function in at least one projector plane direction, for instance as

$$\Phi(\xi, \eta) = \varepsilon \eta, \quad (2.24)$$

where ε is arbitrary. This label is then encoded physically into each light ray by modulating projected optical intensity I_P , which measures energy per unit area normal to the direction of light propagation, Hariharan (2007). However, to properly define a light intensity function $I_P(\xi, \eta)$ dependent on projected phase $\Phi(\xi, \eta)$ which efficiently allows reconstruction of map (2.23) after reflection, an initial understanding of intensity modulation and distortion

is needed first.

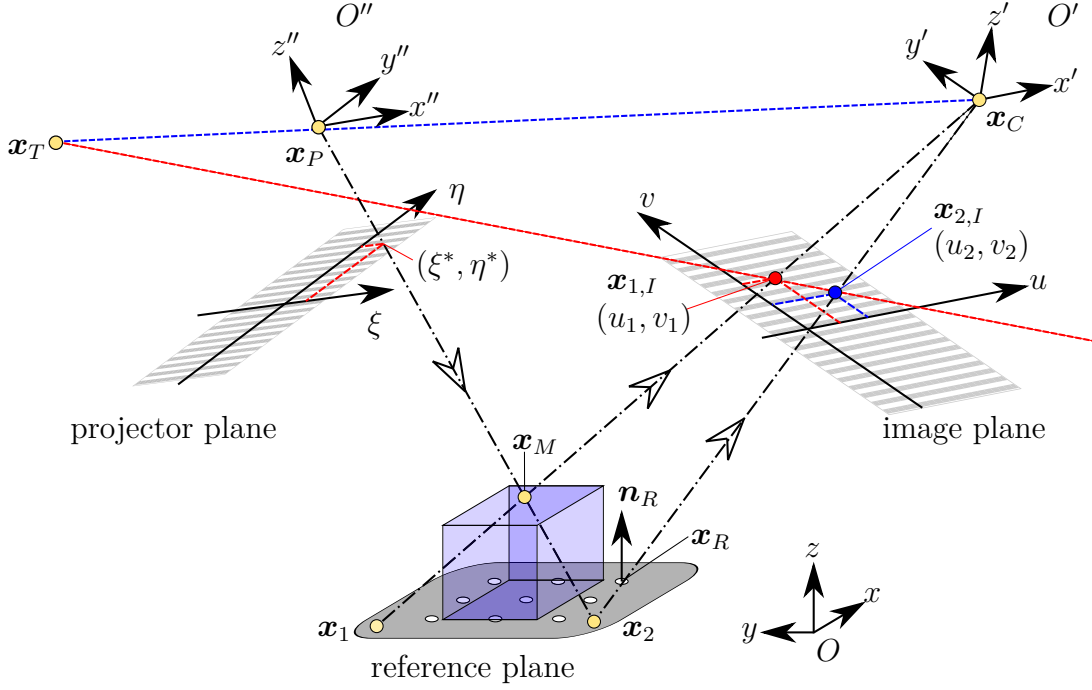


Figure 2.9: Geometric relationships for surface measurement with fringe patterns.

According to Zuo et al. (2018), the light intensity $I_C(u, v)$ of a reflected ray measured by the camera on the image plane differs from its original projected intensity $I_P(\xi, \eta)$ as

$$I_C(u, v) = \alpha(u, v) (I_P(\xi, \eta) + \beta_1(u, v)) + \beta_2(u, v), \quad (2.25)$$

where $\beta_1(u, v)$ corresponds to intensity amplification due to ambient light before reflection, $\alpha(u, v)$ is the local surface reflectivity of the object and $\beta_2(u, v)$ quantifies intensity distortion due to ambient light entering the camera. These three parameters may vary substantially on space, which is why they must be evaluated individually for each ray and thus each position (u, v) on the image plane. Therefore, using projected phase map (2.24) directly as projected intensity function $I_P(\xi, \eta) := \Phi(\xi, \eta)$ and considering the measured intensity $I_C(u, v)$ of each photograph directly as captured phase map $\Psi(u, v) := I_C(u, v)$ would yield

$$\begin{aligned} \Psi(u, v) &= \alpha(u, v) (\Phi(\xi, \eta) + \beta_1(u, v)) + \beta_2(u, v) \\ &= \alpha(u, v)\beta_1(u, v) + \beta_2(u, v) + \alpha(u, v)\Phi(\xi, \eta) \\ &= A(u, v) + B(u, v)\Phi(\xi, \eta). \end{aligned} \quad (2.26)$$

However, since local parameters $A(u, v)$ and $B(u, v)$ are unknown and too complex to be

measured locally with sufficient accuracy, no mapping between projected and captured phase values is thus possible.

A smarter alternative for defining projected and measured intensities is required which circumvents explicit calculation of parameters $A(u, v)$ and $B(u, v)$ for the desired phase mapping. This may be achieved using some variation of the N -step phase shifting algorithm by Srinivasan et al. (1984), which in essence, defines projected light intensity as a cosine function of the projected phase map with an additional phase shift parameter $n \in [1 \dots N] \subset \mathbb{N}$. For example, the four step phase shifting algorithm presented by Peng and Gupta (2007) and simplified here for illustration purposes defines the projected intensity as

$$I_P(\xi, \eta; n) = \frac{I_{\max}}{2} \left(1 + \sin \left(\Phi(\xi, \eta) + \frac{(n-1)\pi}{2} \right) \right), \quad n \in [1 \dots 4] \subset \mathbb{N}. \quad (2.27)$$

With previous formulation and projected phase map (2.24), the projected intensity undulates periodically between 0 and the maximum projector intensity I_{\max} along the projector axis η . Furthermore, because intensity along the axis ξ remains constant, projected intensity (2.27) corresponds to a classical fringe pattern as schematically illustrated on the projector plane in Fig. 2.9. To examine how projected intensity (2.27) is then modulated after reflection and captured by the camera, one may replace it in Eq. (2.25) to obtain the measured intensity

$$\begin{aligned} I_C(u, v; n) &= \alpha \left(\frac{I_{\max}}{2} \left(1 + \sin \left(\Phi(\xi, \eta) + \frac{(n-1)\pi}{2} \right) \right) + \beta_1 \right) + \beta_2 \\ &= \alpha \left(\frac{I_{\max}}{2} + \frac{I_{\max}}{2} \sin \left(\Phi(\xi, \eta) + \frac{(n-1)\pi}{2} \right) \right) + \alpha\beta_1 + \beta_2 \\ &= \alpha \frac{I_{\max}}{2} + \alpha\beta_1 + \beta_2 + \alpha \frac{I_{\max}}{2} \sin \left(\Phi(\xi, \eta) + \frac{(n-1)\pi}{2} \right), \end{aligned} \quad (2.28)$$

where the explicit dependency of parameters α, β_1 and β_2 on the image position (u, v) has been omitted for brevity. Result (2.28) suggests that an appropriate model for the measured intensity $I_C(u, v; n)$ as function of captured phase $\Psi(u, v)$ is thus

$$I_C(u, v; n) = A(u, v) + B(u, v) \sin \left(\Psi(u, v) + \frac{(n-1)\pi}{2} \right) \quad (2.29)$$

with $A(u, v)$ and $B(u, v)$ to be found locally. Now, to compute the captured phase map $\Psi(u, v)$ without addressing parameters $A(u, v)$ and $B(u, v)$, four projections using different phase shift values n are performed delivering four photographs with respective

measured intensities $\{I_C(u, v; n)\}$, $n = 1, 2, 3, 4$. Subsequently, by using subtractions

$$\begin{aligned} I_C(u, v, 1) - I_C(u, v, 3) &= B(u, v) \sin \Psi(u, v) - B(u, v) \sin (\Psi(u, v) + \pi) \\ &= B(u, v) \sin \Psi(u, v) + B(u, v) \sin \Psi(u, v) \\ &= 2B(u, v) \sin \Psi(u, v) \end{aligned} \quad (2.30)$$

and

$$\begin{aligned} I_C(u, v, 2) - I_C(u, v, 4) &= B(u, v) \sin \left(\Psi(u, v) + \frac{\pi}{2} \right) - B(u, v) \sin \left(\Psi(u, v) + \frac{3\pi}{2} \right) \\ &= B(u, v) \cos \Psi(u, v) + B(u, v) \cos \Psi(u, v) \\ &= 2B(u, v) \cos \Psi(u, v) \end{aligned} \quad (2.31)$$

one may obtain the desired captured phase map from the four photographs as

$$\Psi(u, v) = \arctan \left(\frac{I_C(u, v, 1) - I_C(u, v, 3)}{I_C(u, v, 2) - I_C(u, v, 4)} \right). \quad (2.32)$$

In this way, a specific light ray may be localized in the projector and image planes according identical phase values, see for example Fig. 2.9 where $\Phi(\xi^*, \eta^*) = \Psi(u_1, v_1)$.

To demonstrate how surface measurement is performed using phase maps and geometric relationships, consider following example simplified from Peng and Gupta (2007) and explained using the setup shown in Fig. 2.9. First, a light-intensity function $I(\xi, \eta; n)$ is defined and projected onto the reference plane alone (i.e., without measurement object) using four phase shift values $n = 1 \dots 4$. The resulting fringe patterns are photographed and used for computation of a reference phase map $\Psi_R(u, v)$ with Eq. (2.32), i.e., a captured phase map associated with the reference plane. Then, the object of interest is placed and the same fringe patterns are projected leading to a captured phase map $\Psi_C(u, v)$, which is associated with the measurement object and differs from the previously obtained reference phase map $\Psi_R(u, v)$. Consider the ray from camera \mathbf{x}_C through position (u_1, v_1) on the image plane intersecting the measured object at \mathbf{x}_M and the reference plane at \mathbf{x}_1 . Thus, the three-dimensional object coordinates \mathbf{x}_M correspond to the image location (u_1, v_1) and may be obtained from intersection of lines $\mathbf{x}_P + \lambda(\mathbf{x}_2 - \mathbf{x}_P)$ and $\mathbf{x}_C + \mu(\mathbf{x}_1 - \mathbf{x}_C)$.

Line $\mathbf{x}_C + \mu(\mathbf{x}_1 - \mathbf{x}_C)$ may be determined from the camera position \mathbf{x}_C and point $\mathbf{x}_{1,I}$, which also lies on the line and may be obtained using Eq. (2.22) according to the image location (u_1, v_1) . To obtain \mathbf{x}_2 and consequently identify line $\mathbf{x}_P + \lambda(\mathbf{x}_2 - \mathbf{x}_P)$, consider that points \mathbf{x}_M and \mathbf{x}_2 share the same incident ray, and thus the same projected phase $\Phi(\xi^*, \eta^*)$. Hence, the rays reflected from point \mathbf{x}_M onto the object phase map $\Psi_C(u, v)$ and from \mathbf{x}_2 onto the reference phase map $\Psi_R(u, v)$ are labeled with the same captured phase value,

i.e.,

$$\Psi_R(u_2, v_2) = \Psi_C(u_1, v_1), \quad (2.33)$$

where (u_2, v_2) corresponds to the image location associated to point \mathbf{x}_2 . To facilitate computation of (u_2, v_2) , following geometric relationships are considered. Because $\mathbf{x}_P, \mathbf{x}_2, \mathbf{x}_C$ and \mathbf{x}_1 lie in the same plane, $\mathbf{x}_{1,I}$ and $\mathbf{x}_{2,I}$ lie on its intersection line with the image plane. This line may be identified with an additional point \mathbf{x}_T obtained from intersecting line $\mathbf{x}_C + \delta(\mathbf{x}_P - \mathbf{x}_C)$ with the image plane (blue line in Fig. 2.9). All possible image plane coordinates (u, v) lying on line $\mathbf{x}_T + \zeta(\mathbf{x}_{1,I} - \mathbf{x}_T)$ may be then represented in a set

$$\mathcal{L} = \left\{ (u, v) \in \mathbb{R}^2 \mid \|(\mathbf{x}_I(u, v) - \mathbf{x}_T) \times (\mathbf{x}_{1,I} - \mathbf{x}_T)\| = 0 \right\}, \quad (2.34)$$

where $\mathbf{x}_I(u, v)$ corresponds to the Cartesian coordinates associated with position (u, v) according to Eq. (2.22). In this manner, the desired point $\mathbf{x}_{2,I}$ associated to image position (u_2, v_2) must belong to set \mathcal{L} and satisfy Eq. (2.33). Conveniently, given that any phase map is monotonic in one direction, there exists a unique image position satisfying this condition, e.g.

$$(u_2, v_2) = (u, v) \in \mathcal{L} \mid \Psi_R(u, v) = \Psi_C(u_1, v_1). \quad (2.35)$$

Having found position (u_2, v_2) , point $\mathbf{x}_{2,I}$ may be determined with Eq. (2.22) and point \mathbf{x}_2 results from intersecting line $\mathbf{x}_C + \rho(\mathbf{x}_{2,I} - \mathbf{x}_C)$ with the reference plane. Lastly, point \mathbf{x}_2 determines line $\mathbf{x}_P + \lambda(\mathbf{x}_2 - \mathbf{x}_P)$ and point \mathbf{x}_M is finally obtained from the intersection with line $\mathbf{x}_C + \mu(\mathbf{x}_1 - \mathbf{x}_C)$.

Previous procedure delivers a point cloud representation of a single image captured by the camera. To reconstruct the entire object surface, the process is repeated to obtain multiple images from different camera angles and positions. All resulting point clouds may then be aligned using multiple coded targets applied to the object surface as reference. After manual elimination of measured points outside the object of interest, the point cloud may be triangulated to produce a fine surface mesh, where each element is characterized by its centroid \mathbf{x}_i and unit normal vector \mathbf{n}_i . Thus, the measured object may be represented by sets

$$\mathcal{M} = \{\mathbf{x}_1, \dots, \mathbf{x}_M\}, \quad \mathbf{x} \in \mathbb{R}^3 \quad \text{and} \quad \mathcal{N} = \{\mathbf{n}_1, \dots, \mathbf{n}_M\}, \quad \mathbf{x} \in \mathbb{R}^3, \quad (2.36)$$

of measured points and normal vectors, respectively.

Measurement accuracy of structured light depends on several physical factors like layer thickness of reflective titan coating, surface curvature, light exposure or ground vibrations, as well as human factors regarding infrequent calibration, unreliable part fixation, coarse mesh triangulation, etc. Nonetheless, it is well accepted that typical

measurement uncertainties of structured light lie below $50 \mu\text{m}$. For example Li et al. (2021) report a combined repeatability, reproducibility and resolution uncertainty of $11 \mu\text{m}$ for linear distances. A similar repeatability error for low curvature regions was obtained by Backhaus et al. (2017). However, the authors showed deviations up to $27 \mu\text{m}$ when comparing airfoil chord lengths derived from structured light and direct coordinate measurements. Given the highly curved free-form surfaces treated in the present work, a rather conservative surface measurement uncertainty of $e_S = 25 \mu\text{m}$ will be assumed in the following.

3 CAD Representation of Manufacturing Deviations

Optical measurements of manufactured parts offer a practical and precise means to assess manufacturing quality. There are, however, multiple engineering scenarios where a CAD representation of the manufactured part would be more useful than its corresponding surface scan. For instance, when engineering simulations of the manufactured sample are expected, a CAD representation may serve as a unified repository of surface deviations, ensuring consistency for any mesh derived from it and enabling access to further CAD post-processing tools. More importantly, because any part is first designed as CAD model, having a CAD representation provides valuable insight into how to parametrize manufacturing variability so that it might be modeled and recreated. This is the gateway for robust design where, in order to identify the geometry least affected by manufacturing deviations, multiple deviating geometries have to be simulated and analyzed. Thus, the present chapter will describe a series of methods to transfer surface deviation information from surface measurements into CAD models, such that manufacturing scatter is effectively parametrized and may be later used for probabilistic analysis and robust design.

The reach and capacity of the proposed workflow relies on three edges: a structured CAD routine to construct the mechanical part using predominantly Boolean operations (Sec. 2.3), an automated re-parametrization strategy of any face as B-spline surface (Sec. 3.1), and the ability to handle B-spline surfaces for deviation representation (Secs. 3.2 - 3.5). An overview of the proposed workflow, and thus of the present chapter, is shown in Fig. 3.1. The process starts with a B-spline re-parametrization of the nominal surface. Then, both re-parametrized surface and scan will be associated by defining relevant surface positions and control points. Next, control point deviations will be computed using B-spline morphing to fit a re-parametrized surface to the scan, and finally, residual noise of the morphing process will be removed from the resulting surface. The entire process may be repeated for multiple iterations, until the desired fitting accuracy is achieved. To close the chapter, application examples will be provided for the hot-gas and shank faces of an HPT blade.

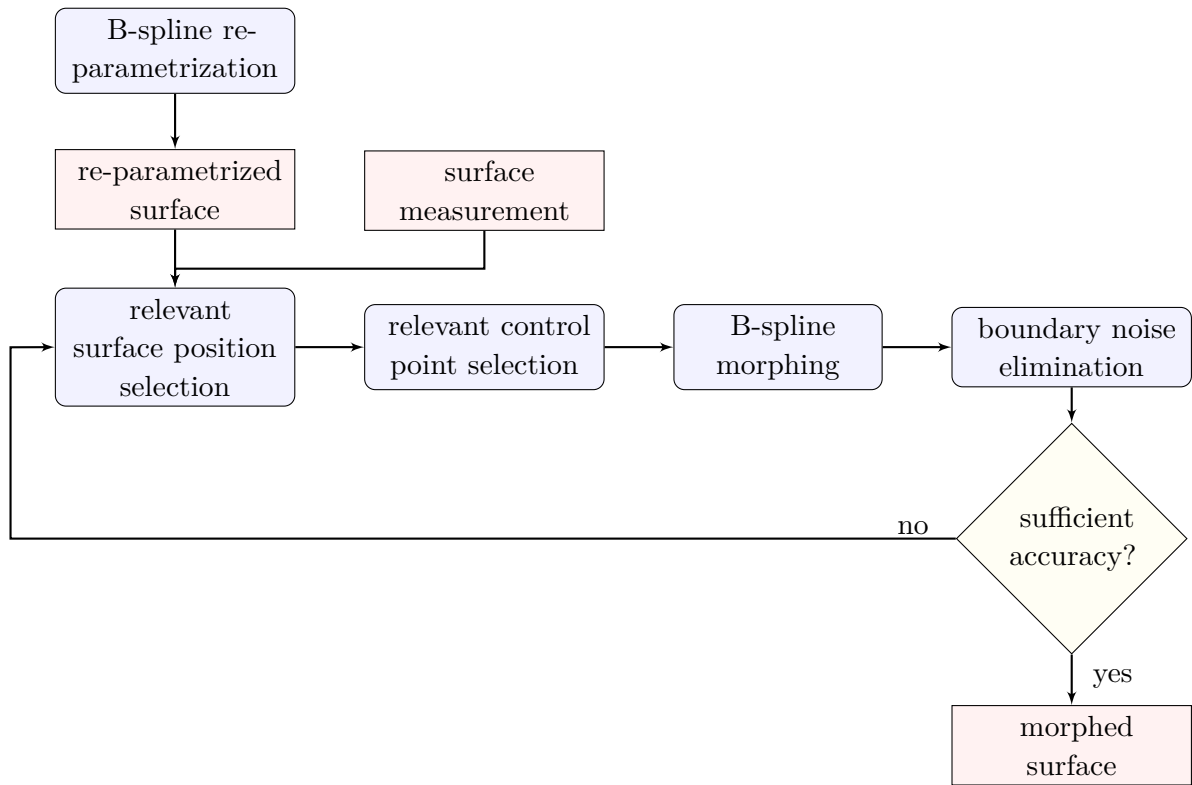


Figure 3.1: Developed workflow for CAD representation of manufacturing deviations.

3.1 Re-Parametrization as B-Spline Surface

Consider a CAD face to be reconstructed from a measured manufactured part, such as the shank wall of an HPT blade, Fig. 3.2a. Once the surface measurement has been performed, it must be aligned according to the desired deviation analysis intent, for instance, to the contact faces during nominal assembly position. As an additional preparation step, the surface scan may be segmented to the area of interest, which eliminates irrelevant scan regions and accelerates deviation calculations. As a result, the segmented surface measurement may be represented as point set with associated normal vectors (2.36).

As discussed in Sec. 2.3, complex CAD parts frequently require standardized construction approaches to efficiently associate up to hundreds of design parameters. Given the already high complexity of nominal geometry generation, inefficient inclusion of additional deviation representation procedures may lead to overly complicated construction workflows. Therefore, integration of deviation representation procedures should exploit already existing nominal CAD design operations as much as possible. In particular, because a large portion of CAD faces may be constructed using subtraction operations, workflow efficiency and scalability may be gained by replacing nominal with morphed subtraction bodies which would contain all deviation features of the associated CAD faces.

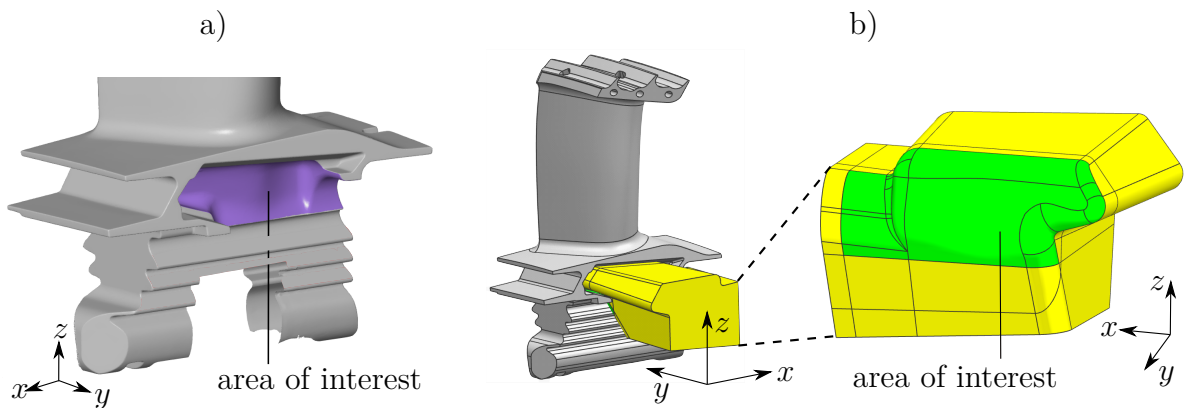


Figure 3.2: Illustration of a) HPT blade surface measurement and b) associated subtraction body for shank construction.

For example, Fig. 3.2b illustrates the CAD construction of a shank face using a subtraction body. To take advantage of the subtractive construction approach, deviation features should be captured in the subtraction body first, and then automatically imprinted in the final CAD part by subtraction. This implies that all CAD faces corresponding to the area of interest should be modified to match the segmented surface measurement, which imposes two challenges. First, assuming that the area of interest is composed of several adjacent primitive and free-form faces, the representation capacity would differ considerably from face to face (a planar face has less representation capacity than a free-form face). Secondly, assuming that each face is independently fitted to the scan, a large number of boundary continuity corrections would be required to guarantee an overall smooth body.

A more efficient alternative for deviation representation, which overcomes previous issues, consists in a re-parametrization of the entire subtraction body as a B-spline surface with a high number of control points, Fig. 3.3. This approach ensures surface smoothness and high representation capacity since B-splines may be accurately fitted to any surface measurement and, therefore, reproduce a high-fidelity CAD version of the manufactured face. Furthermore, the B-spline re-parametrization procedure is commonly available for simple surfaces in commercial CAD programs, and in the case of complex surfaces it may be implemented by constructing a B-spline surface from a user-defined control point grid.

Since the re-parametrized surface $\mathbf{s}(u, v)$ is an approximation of the nominal B-spline surface $\mathbf{s}_N(u, v)$, a small re-parametrization error

$$e_C(u, v) = \|\mathbf{s}(u, v) - \mathbf{s}_N(u, v)\| \quad (3.1)$$

is naturally expected. This error is automatically computed by the CAD software, and with proper control point grid structure may be kept below $5 \mu\text{m}$, see for example Fig. 2.4. Once the surface measurement has been aligned and segmented and the subtraction body

has been re-parametrized as B-spline surface, the association between measured points and B-spline surface points may be performed.

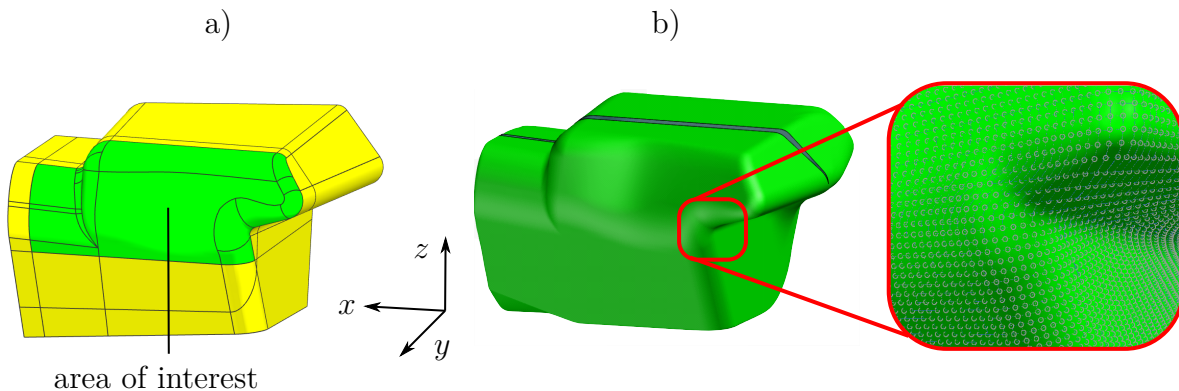


Figure 3.3: Illustration of a) nominal subtraction body and b) associated re-parametrized B-spline surface with a 135×96 control point grid.

3.2 Selection of Relevant Surface Positions

There exist several approaches to construct B-spline surfaces from discrete point information. In particular, if a nominal B-spline surface relatively close to a given point cloud is already available, this procedure is considerably simplified. For instance, Gaun et al. (2014) demonstrate how thermal deformations of free-form geometries may be represented in CAD by adding control point deviations to a given B-spline surface representing the nominal geometry. In a similar way, Urbano et al. (2019a) utilize the same approach to accurately represent manufacturing deviations from surface measurements as morphed B-spline surfaces. Both studies highlight the applicability of B-spline morphing with numerous case studies from aero engine components. In the particular case of manufacturing deviations, however, no special emphasis has been given to increase morphing robustness against surface measurement noise. This is a cardinal topic, since the limiting factor for larger statistical studies and batch CAD reconstruction is frequent user involvement to manually handle scan noise and individual morphing imperfections. The present work will build upon B-spline morphing as presented by Urbano et al. (2019a) and describe additional strategies to mitigate scan and morphing irregularities.

The first phase of B-spline surface fitting consists of local association between B-spline surface $\mathbf{s}(u, v)$ and the surface measurement set \mathcal{M} in Eq. (2.36). In particular, all normalized surface positions (u, v) in the vicinity of the area of interest have to be associated with respective closest measured points $\mathbf{x} \in \mathcal{M}$. For illustration, consider Fig. 3.4, where B-spline surface $\mathbf{s}(u, v)$ has to be fitted to a horizontal area of interest within a segmented

scan \mathcal{M} . In this illustration, however, the segmented scan accidentally extends beyond the area of interest and includes additional unwanted regions, such as a perpendicular border of the measured object (shown as gray area). Although these undesired regions may be easily eliminated manually, such an approach would limit automation of batch surface fitting, which is why automatic noise removal strategies will be emphasized. To begin the association, a fine grid

$$\mathcal{G} = \{u_1, \dots, u_\mu, \dots, u_{n'}\} \times \{v_1, \dots, v_\nu, \dots, v_{m'}\} \quad (3.2)$$

of surface parameter values (u, v) is introduced, which samples the entire B-spline surface $\mathbf{s}(u, v)$ into a dense structured mesh of points. Next, function

$$\phi(u, v) = \underset{\eta: \mathbf{x}_\eta \in \mathcal{M}}{\operatorname{argmin}} \|\mathbf{s}(u, v) - \mathbf{x}_\eta\|, \quad (u, v) \in \mathcal{G}, \quad \phi: \mathcal{G} \rightarrow [1, M] \subset \mathbb{N} \quad (3.3)$$

is defined, which provides the index η of the closest measured point \mathbf{x}_η to a specific surface point $\mathbf{s}(u, v)$. In Fig. 3.4, this is sketched by the three dashed lines.

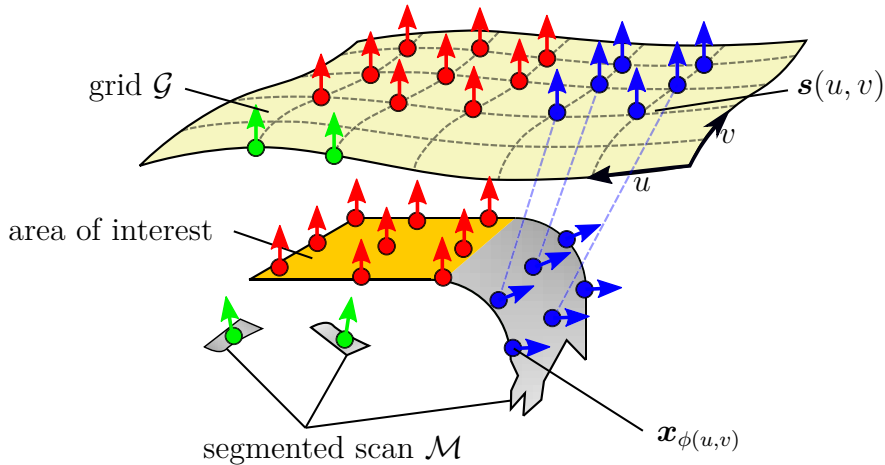


Figure 3.4: B-spline surface $\mathbf{s}(u, v)$ (control points not shown) with associated segmented scan \mathcal{M} .

Because both grid \mathcal{G} and surface scan \mathcal{M} may contain a very large number of points, an efficient distance minimization strategy is required to solve minimization problem (3.3). A simple approach consists in performing a k -d tree search, Bentley (1975), where k stands for search space dimensionality, in the present case $k = 3$ Cartesian dimensions. This method recursively divides the search space into clusters. In case of point cloud \mathcal{M} , first according to the median value of x -coordinates of all measured points, producing two clusters. Then, by the median value of y -coordinates producing four total clusters,

finally division across the z -dimension delivers eight clusters. After this the process is repeated. At each step, all median values are stored as nodes in a tree-like structure, such that the closest measured point $\mathbf{x}_{\phi(u,v)}$ to a given surface point $\mathbf{s}(u,v)$ may be efficiently computed by navigating across the nodes of the tree.

Having implemented function (3.3), all relevant normalized positions (u,v) in the vicinity of the area of interest may now be identified. This is achieved by comparing distances $\|\mathbf{s}(u,v) - \mathbf{x}_{\phi(u,v)}\|$ to a user-defined threshold d_{th} . All normalized positions (u,v) in the area of interest are stored in the index set

$$\mathcal{U}_d = \left\{ (u,v) \in \mathcal{G} \mid \|\mathbf{s}(u,v) - \mathbf{x}_{\phi(u,v)}\| \leq d_{\text{th}} \right\}. \quad (3.4)$$

In Fig. 3.4, a clustering of different regions is shown as red, blue and green points for both scan and B-spline surface (the respective normal vectors are also sketched). As illustrated, the threshold-based selection may erroneously identify unwanted point pairs, such as blue and green points, which highlights the need for refinement strategies.

A simple and useful refinement method is based on normal vector deviations, which discards all associations where measured normal vectors significantly deviate from associated B-spline normal vectors. Since B-spline normal vectors $\mathbf{n}(u,v)$ are analytically available from Eq. (2.10) and measured normal vectors $\mathbf{n}_{\phi(u,v)}$ can be estimated from structured light measurement as described by Eq. (2.36), this refinement method is computationally inexpensive. Thus, the normalized positions from the previous distance-threshold set \mathcal{U}_d are further reduced to the set

$$\mathcal{U} = \left\{ (u,v) \in \mathcal{U}_d \mid \mathbf{n}_{\phi(u,v)}^T \mathbf{n}(u,v) \geq \cos \theta_{\text{th}} \right\}, \quad (3.5)$$

of relevant surface positions where only (u,v) positions with normal vector deviations below a user-defined angle θ_{th} are accepted. For the example in Fig. 3.4, this refinement step would eliminate all blue point associations, leaving a refined area of interest consisting of red and green points. The algorithmic parameters d_{th} and θ_{th} have to be defined according to the largest expected deviations, see Sec. 3.6 with application examples.

3.3 Selection of Relevant Control Points

To simplify the fitting process of the B-spline surface, a selection step is also performed for control points of $\mathbf{s}(u,v)$ having direct influence on the interesting region. Because this selection is performed individually for each surface measurement, it is prone to errors due to scan imperfections. Therefore, refinement strategies for exclusion of unwanted control points are required as well.

Consider Fig. 3.5a corresponding to a re-parametrization of the subtraction body in Fig. 3.3a, where only control points at the intersection of a cutting plane are illustrated. Similarly as in the previous section, the identification of control points near the surface scan starts with a new user-defined distance threshold d_{cp} , now defined between control points and measured points. This criterion alone is, however, not robust for two reasons. A selection based on a small threshold d_{cp} would identify correctly all relevant control points (red points at the scan boundary in Fig. 3.5b), however, it may ignore relevant control points at the interior if large deviations are present (white points). On the other hand, a more conservative selection would correctly identify all control points at the interior, but would probably also include some unwanted control points at the scan boundary as in Fig. 3.5c. Therefore, the first selection refinement shall eliminate all unwanted control points at the scan boundary.

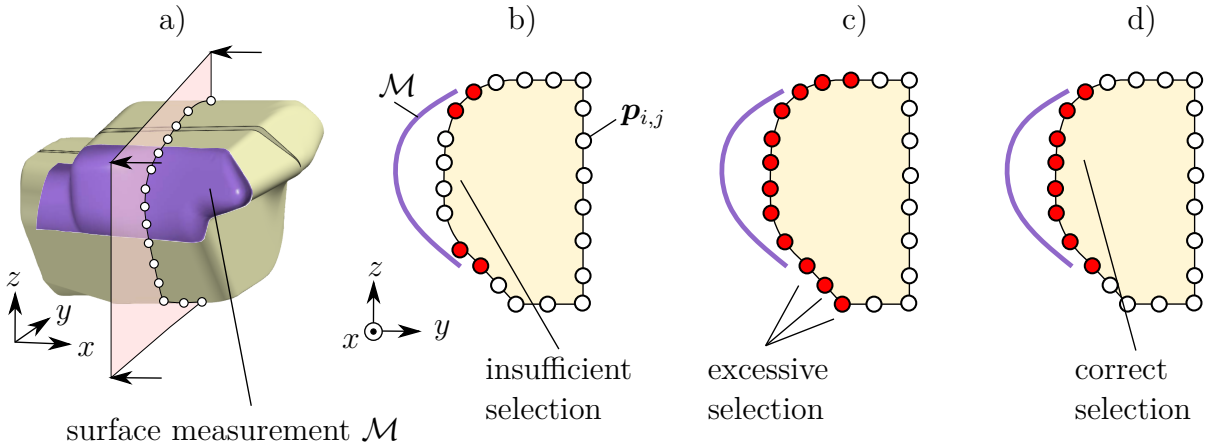


Figure 3.5: Exemplary control point selection for a) B-spline surface section using b) a small distance threshold d_{cp} and c) a large distance threshold d_{cp} , as well as d) a large distance threshold d_{cp} with boundary control.

To identify control points based on a distance threshold, function

$$\psi(i, j) = \operatorname{argmin}_{\eta: \mathbf{x}_\eta \in \mathcal{M}} \|\mathbf{p}_{i,j} - \mathbf{x}_\eta\|, \quad (i, j) \in [0, n] \times [0, m] \subset \mathbb{N}_0^2,$$

$$\psi : [0, n] \times [0, m] \rightarrow [1, M] \subset \mathbb{N}, \quad (3.6)$$

similar to Eq. (3.3) is first defined, which identifies the closest measured point \mathbf{x}_η to each control point $\mathbf{p}_{i,j}$. This function may also be evaluated using a k -d tree search. In particular, since surface measurement \mathcal{M} is already represented as k -d tree during relevant surface position selection, it may be reused to query control points $\mathbf{p}_{i,j}$ instead. Now all control point positions (i, j) having a closest measured point $\mathbf{x}_{\psi(i,j)}$ to control

point $\mathbf{p}_{i,j}$ below a distance d_{cp} are stored in an index set

$$\mathcal{I}_d = \left\{ (i, j) \in [0, n] \times [0, m] \subset \mathbb{N}_0^2 \mid \|\mathbf{p}_{i,j} - \mathbf{x}_{\psi(i,j)}\| \leq d_{\text{cp}} \right\}, \quad (3.7)$$

where threshold d_{cp} may be conservatively defined such that all control points close to the area of interest of the surface measurement \mathcal{M} are correctly selected. Notice that the closest points $\mathbf{x}_{\psi(i,j)}$ are not required to belong to the relevant surface positions (3.5).

The elimination of unwanted control points at the periphery of scan \mathcal{M} is now performed by defining a tangential deviation metric. In general, as control points are located further away from surface measurement \mathcal{M} , the distance to the closest measured point tends to be more tangential to the control point grid rather than normal, Fig. 3.6. Thus, the projected distance $\|(\mathbf{x}_{\psi(i,j)} - \mathbf{p}_{i,j})^\top \mathbf{t}_{i,j}^v\|$ from closest measured point $\mathbf{x}_{\psi(i,j)}$ to a particular control point $\mathbf{p}_{i,j}$ is useful as elimination criterion. Notice that the tangential vector $\mathbf{t}_{i,j}^v$ along direction v is associated to control point position (i, j) rather than to surface position (u, v) , e.g. it must be approximated using finite differences (2.16). Since a user-defined threshold based on tangential distance is closely dependent on control point density, it would require manual calibration each time the control point density is adjusted. Therefore, the elimination of unwanted boundary control points is better performed with a normalized tangential deviation metric

$$\tau_{i,j}^v = \frac{\|(\mathbf{x}_{\psi(i,j)} - \mathbf{p}_{i,j})^\top \mathbf{t}_{i,j}^v\|}{\|\mathbf{p}_{i,j+1} - \mathbf{p}_{i,j}\|}, \quad (i, j) \in \mathcal{I}_d, \quad (3.8)$$

which relates the tangential distance to the closest measured point with the distance to the neighboring control point $\mathbf{p}_{i,j+1}$ in the grid, Fig. 3.6. Thus, the previous selection \mathcal{I}_d based on distance threshold d_{cp} may now be refined to exclude unwanted periphery control points and form a new index set

$$\mathcal{I}_p = \left\{ (i, j) \in \mathcal{I}_d \mid \max \left\{ \tau_{i,j}^u, \tau_{i,j}^v \right\} < \tau_{\text{th}} \right\}, \quad (3.9)$$

where only control point positions (i, j) with both tangential deviation metrics $\tau_{i,j}^u$ and $\tau_{i,j}^v$ below a user-defined threshold τ_{th} are accepted.

During B-spline morphing, deviations between relevant B-spline surface positions $\mathbf{s}(u, v)$, $(u, v) \in \mathcal{U}$, and respective closest measured points $\mathbf{x}_{\phi(u,v)}$ will be minimized by applying control point shifts to all relevant control points. Because each relevant control point may only affect a limited region in the B-spline surface, sufficient deviation information must be available in that region to enable accurate computation of control point displacements. If, for example, only very few relevant measured and B-spline point

associations exist in a particular region, the control point shift computation may be inexact and even lead to noise in the morphed surface due to insufficient information. For a sufficiently dense grid \mathcal{G} and small surface deviations, such problematic cases may be mostly located at the scan periphery rather than in the interior. Therefore, to prevent potential morphing noise, an additional refinement strategy consisting in elimination of all insufficiently supported control points needs to be formulated.

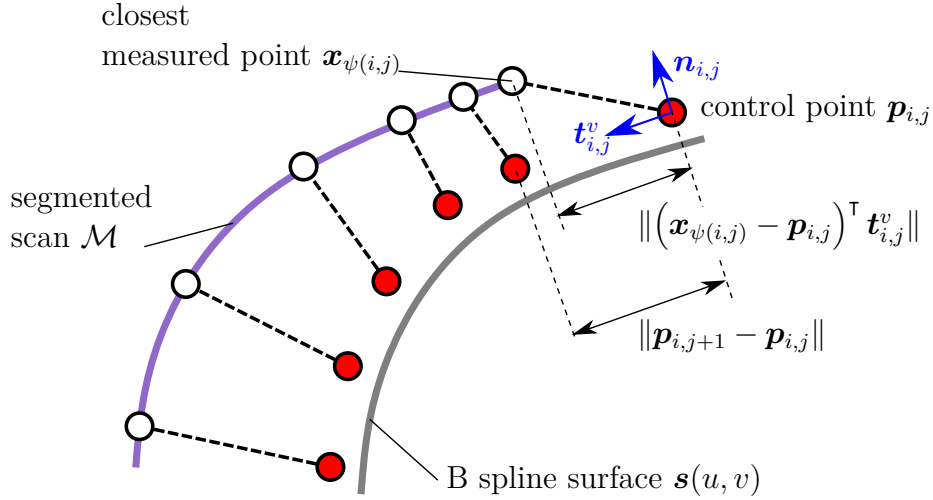


Figure 3.6: Calculation of tangential control point deviation $d_{i,j}^j$.

The support of a control point $\mathbf{p}_{i,j}$ may be specified as the number of relevant B-spline surface points $\mathbf{s}(u, v)$, $(u, v) \in \mathcal{U}$, located closer to control point $\mathbf{p}_{i,j}$ than to any other control point. First, the distance from a specific control point $\mathbf{p}_{i,j}$ to the closest control point neighbor is introduced as a reference distance

$$h_{i,j} := \min_{\substack{k \in \{-1,1\} \\ l \in \{-1,1\}}} \|\mathbf{p}_{i,j} - \mathbf{p}_{i+k,j+l}\|, \quad (3.10)$$

associated to a particular control point position (i, j) . Then, the number of support points associated to a given control point position (i, j) may be defined as

$$\varepsilon(i, j) = \left| \left\{ (u, v) \in \mathcal{U} \mid \|\mathbf{p}_{i,j} - \mathbf{s}(u, v)\| < h_{i,j} \right\} \right|, \quad (i, j) \in \mathcal{I}_p \subset \mathbb{N}_0^2, \quad \varepsilon : \mathcal{I}_p \rightarrow [0, |\mathcal{U}|] \subset \mathbb{N}_0, \quad (3.11)$$

where the set collects all relevant surface grid points $\mathbf{s}(u, v)$ with a distance lower than $h_{i,j}$ from control point $\mathbf{p}_{i,j}$, and $\varepsilon(i, j)$ returns the corresponding set size. For illustration, consider the example in Fig. 3.7, where all support points associated with control point $\mathbf{p}_{i,j}$

are colored in dark blue. In this manner, the last selection refinement step isolates all relevant control points with enough support in the index set

$$\mathcal{I} = \left\{ (i, j) \in \mathcal{I}_p \mid \varepsilon(i, j) > n_{\text{sup}} \right\}, \quad (3.12)$$

where the user-defined number n_{sup} represents the minimal accepted support.

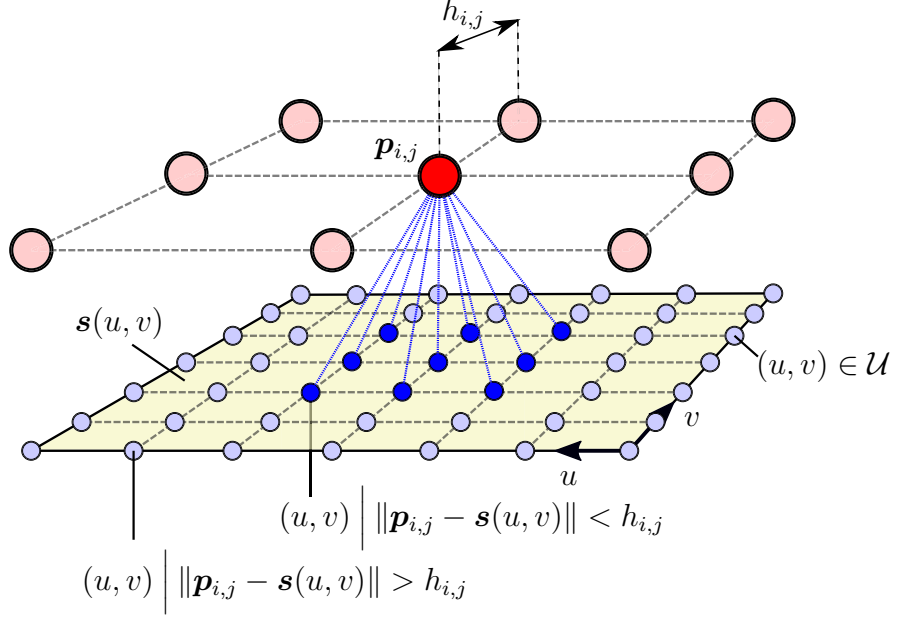


Figure 3.7: Schematic representation of control point support (dark blue).

By using present selection refinement steps, the relevant control point positions \mathcal{I} may be robustly identified regardless of most scan irregularities, which effectively prevents morphing noise and facilitates automated batch reconstruction. Similar to the previous section, the method parameters d_{cp} , τ_{th} and n_{sup} have to be manually adjusted according to various scan samples to ensure a robust and consistent control point selection.

3.4 B-Spline Morphing to Surface Measurements

Once the association between re-parametrized B-spline surface $s(u, v)$ and surface measurement \mathcal{M} has been performed and all relevant control point positions have been identified in set \mathcal{I} , the surface fitting may proceed. The goal of this step is to compute control point shifts $\Delta p_{i,j}$ which minimize differences between the B-spline surface and a given scan. This process is denominated B-spline morphing and will be described next following the work of Gaun et al. (2014) and Urbano et al. (2019a).

Let the re-parametrized B-spline surface $s(u, v)$ be given by Eq. (2.6). The aim is to

obtain a morphed surface

$$\begin{aligned}\hat{\mathbf{s}}(u, v) &= \sum_{i=0}^n \sum_{j=0}^m N_{i,q}(u) N_{j,q}(v) (\mathbf{p}_{i,j} + \Delta \mathbf{p}_{i,j}) \\ &\equiv \mathbf{s}(u, v) + \sum_{i=0}^n \sum_{j=0}^m N_{i,q}(u) N_{j,q}(v) \Delta \mathbf{p}_{i,j}\end{aligned}\quad (3.13)$$

which closely matches a given surface scan \mathcal{M} . By imposing that irrelevant control points should not carry any control point deviation, i.e., $\Delta \mathbf{p}_{i,j} = 0 \forall (i, j) \notin \mathcal{I}$, Eq. (3.13) may be reduced to

$$\hat{\mathbf{s}}(u, v) \approx \mathbf{s}(u, v) + \sum_{(i,j) \in \mathcal{I}} N_{i,q}(u) N_{j,q}(v) \Delta \mathbf{p}_{i,j}. \quad (3.14)$$

In order to find proper control point displacements $\Delta \mathbf{p}_{i,j}$, equality between morphed surface points $\hat{\mathbf{s}}(u, v)$ and respective closest measured points $\mathbf{x}_{\phi(u,v)} \in \mathcal{M}$ at the relevant normalized positions $(u, v) \in \mathcal{U}$ defined by Eq. (3.5) is imposed:

$$\mathbf{s}(u, v) + \sum_{(i,j) \in \mathcal{I}} N_{i,q}(u) N_{j,q}(v) \Delta \mathbf{p}_{i,j} \stackrel{!}{=} \mathbf{x}_{\phi(u,v)} \quad \forall (u, v) \in \mathcal{U}. \quad (3.15)$$

This may be rearranged as

$$\sum_{(i,j) \in \mathcal{I}} N_{i,q}(u) N_{j,q}(v) \Delta \mathbf{p}_{i,j} = \mathbf{x}_{\phi(u,v)} - \mathbf{s}(u, v) =: \mathbf{d}_{(u,v)} \quad \forall (u, v) \in \mathcal{U}. \quad (3.16)$$

Eq. (3.16) highlights how localized manufacturing deviations $\mathbf{d}_{(u,v)}$ at a particular surface position (u, v) are smoothly distributed over the control point grid according to B-spline basis functions. Furthermore, it is a vector equation which relates vector control point deviations $\Delta \mathbf{p}_{i,j} = [\Delta p_{i,j}^x, \Delta p_{i,j}^y, \Delta p_{i,j}^z]^\top$ to corresponding manufacturing deviations $\mathbf{d}_{(u,v)} = [d_{(u,v)}^x, d_{(u,v)}^y, d_{(u,v)}^z]^\top$. For instance, the x -coordinate of Eq. (3.16) reads as

$$\sum_{(i,j) \in \mathcal{I}} N_{i,q}(u) N_{j,q}(v) \Delta p_{i,j}^x = d_{(u,v)}^x \quad \forall (u, v) \in \mathcal{U}. \quad (3.17)$$

These equations can be summarized columnwise as

$$\mathbf{N} \mathbf{P} = \mathbf{D} \quad (3.18)$$

where

$$\mathbf{N} := \begin{bmatrix} \vdots \\ \cdots & N_{i,q}(u) N_{j,q}(v) & \cdots \\ \vdots \end{bmatrix} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}, \quad (3.19)$$

$$\mathbf{P} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \Delta p_{i,j}^x & \Delta p_{i,j}^y & \Delta p_{i,j}^z \\ \vdots & \vdots & \vdots \end{bmatrix} =: [\Delta \mathbf{p}^x, \Delta \mathbf{p}^y, \Delta \mathbf{p}^z] \in \mathbb{R}^{|\mathcal{I}| \times 3}, \quad (3.20)$$

$$\mathbf{D} = \begin{bmatrix} \vdots & \vdots & \vdots \\ d_{(u,v)}^x & d_{(u,v)}^y & d_{(u,v)}^z \\ \vdots & \vdots & \vdots \end{bmatrix} =: [\mathbf{d}^x, \mathbf{d}^y, \mathbf{d}^z] \in \mathbb{R}^{|\mathcal{U}| \times 3}. \quad (3.21)$$

In general Eq. (3.18) it is overdetermined since the number $|\mathcal{U}|$ of relevant B-spline positions, and thus equations, is larger than the number $|\mathcal{I}|$ of relevant control points, and thus unknowns. The control point positions \mathbf{P} may, therefore, be computed as least-square regressions from

$$\min_{\Delta \mathbf{p}^j \in \mathbb{R}^{|\mathcal{I}|}} \|\mathbf{N} \Delta \mathbf{p}^j - \mathbf{d}^j\|, \quad j \in \{x, y, z\}, \quad (3.22)$$

in each coordinate direction separately, where ordinary linear regression methods may be applied. In particular, given that matrix \mathbf{N} is sparse, computational efficiency may be gained by considering sparse least-squares solvers, such as LSQR, Paige and Saunders (1982). The resulting control point shifts $\Delta \mathbf{p}^j$ may then be used in Eq. (3.14) to produce the morphed surface $\hat{\mathbf{s}}(u, v)$.

Since control point displacements $\Delta \mathbf{p}_{i,j}$ are obtained from regression, they are prone to residuals and the accuracy of B-spline morphing may be quantified by distance vectors

$$\mathbf{f}(u, v) = \hat{\mathbf{s}}(u, v) - \mathbf{x}_{\phi(u,v)} \quad (3.23)$$

between morphed surface points $\hat{\mathbf{s}}(u, v)$, $(u, v) \in \mathcal{U}$ and corresponding measured points $\mathbf{x}_{\phi(u,v)}$, Fig. 3.8. Using Eq. (3.13) and definition (3.16), previous equation may be expanded as

$$\begin{aligned} \mathbf{f}(u, v) &= \mathbf{s}(u, v) + \sum_{(i,j) \in \mathcal{I}} N_i^q(u) N_j^q(v) \Delta \mathbf{p}_{i,j} - \mathbf{x}_{\phi(u,v)} \\ &= \sum_{(i,j) \in \mathcal{I}} N_i^q(u) N_j^q(v) \Delta \mathbf{p}_{i,j} + (\mathbf{s}(u, v) - \mathbf{x}_{\phi(u,v)}) \\ &= \sum_{(i,j) \in \mathcal{I}} N_i^q(u) N_j^q(v) \Delta \mathbf{p}_{i,j} - \mathbf{d}_{(u,v)}. \end{aligned} \quad (3.24)$$

This corresponds to the difference between the left and right hand side of regression conditions (3.16) or (3.18), respectively.

To formulate a morphing error comparable to the measurement error e_S in Sec. 2.4 or the re-parametrization error e_C in Eq. (3.1), distance $\mathbf{f}(u, v)$ may be projected onto the B-spline surface normal vector $\mathbf{n}(u, v)$ defined by Eq.(2.10). In this manner, the local

morphing error normal to the B-spline surface for every relevant surface position $(u, v) \in \mathcal{U}$ is given as the absolute value

$$e_M(u, v) = \left| \mathbf{f}^\top(u, v) \mathbf{n}(u, v) \right|, \quad (u, v) \in \mathcal{U}. \quad (3.25)$$

The global evaluation of morphing quality may be obtained on a statistical basis, for example, by considering percentiles from all evaluated surface positions, see application examples in Sec. 3.6.

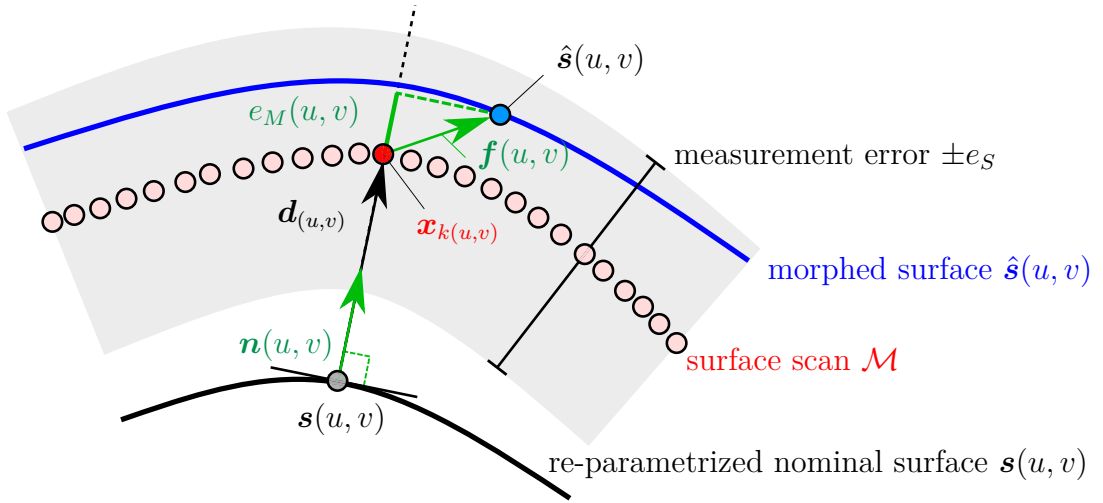


Figure 3.8: Definition of morphing error e_M .

3.5 Noise Elimination from Morphed Surfaces

After robust selection of surface positions and control points, B-spline morphing of re-parametrized surface $\mathbf{s}(u, v)$ may offer an accurate CAD representation of scan \mathcal{M} . In the case of larger manufacturing deviations, however, abrupt surface discontinuities may appear at the boundary between relevant and irrelevant regions. These discontinuities may manifest as sudden peaks and undulations in control point deviations and may potentially lead to CAD construction issues. Since control point deviations at the surface interior have been accurately matched, any noise elimination strategy must be exclusively applied on an individual basis. Hence, any traditional surface smoothing, such as Gaussian blur applied over the entire morphed surface, may negatively impact the overall morphing accuracy. For this reason, the present section illustrates an efficient approach to treat unusually large control point deviations near the scan boundary.

The first of four total steps consists of identification of control point positions (i, j) at the scan boundary. For this, consider a representation of the relevant control point

region \mathcal{I} as binary matrix

$$\mathbf{C} \in \mathbb{B}^{(n+1) \times (m+1)} \quad \text{where} \quad c_{i,j} := \begin{cases} 1 & \text{if } (i,j) \in \mathcal{I}, \\ 0 & \text{otherwise.} \end{cases} \quad (3.26)$$

The boundary region may then be defined by following the concept of erosion in mathematical morphology, Haralick et al. (1987). Essentially, a convolution between binary matrix \mathbf{C} and a $(2h+1) \times (2h+1)$ kernel \mathbf{K} is performed, where the convolution at element (i,j) is defined as

$$(\mathbf{C} * \mathbf{K})_{i,j} = \sum_{\mu=-h}^h \sum_{\nu=-h}^h c_{i+\mu, j+\nu} k_{\mu+h+1, \nu+h+1}. \quad (3.27)$$

The kernel is centered at position (i,j) of matrix \mathbf{C} and performs an element-wise multiplication and addition of overlapping elements. In case of an $\alpha \times \alpha$ matrix

$$\mathbf{J}_\alpha := \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{\alpha \times \alpha} \quad (3.28)$$

of ones as kernel, the operation returns the total number of elements covered by the kernel, Fig. 3.9a–b. In this manner, the boundary region may be defined as the set of relevant positions (i,j) where the kernel is not fully covered by relevant elements. Thus, these boundary positions may be stored in set

$$\mathcal{I}_{\text{bound}} = \left\{ (i,j) \in \mathcal{I} \mid (\mathbf{C} * \mathbf{J}_\alpha)_{i,j} < \alpha^2 \right\}, \quad (3.29)$$

where a position (i,j) is accepted only if the number of overlapping elements with kernel \mathbf{J}_α is strictly less than its area α^2 . Thus, the user-defined kernel size α serves for width control of the boundary region, for example in Fig. 3.9c $\alpha = 3$.

The second step deals with the identification of unusually large control point deviations in the scan boundary. Such deviation peaks may be characterized by abrupt deviation discontinuities in the control point grid, or equivalently large partial derivatives with respect to a given grid coordinate. Thus, a useful approximation of control point deviation derivatives may be obtained using central-differences (2.15), e.g.

$$\frac{\partial \Delta \mathbf{p}_{i,j}}{\partial u} \approx \frac{\Delta \mathbf{p}_{i+1,j} - \Delta \mathbf{p}_{i-1,j}}{2h_u} \quad \text{and} \quad \frac{\partial \Delta \mathbf{p}_{i,j}}{\partial v} \approx \frac{\Delta \mathbf{p}_{i,j+1} - \Delta \mathbf{p}_{i,j-1}}{2h_v}, \quad (3.30)$$

where $h_u = 1/n$ and $h_v = 1/m$ correspond to parametric spacings assuming equidistant

control points. For fine control point grids, previous approximation is sufficient; however, in case of significant spacing discrepancies, finite differences for unevenly spaced grids may become more appropriate, see Fornberg (1988). Thus, the location (i, j) of peak deviations at the scan boundary may be collected in set

$$\mathcal{I}_{\text{noisy}} = \left\{ (i, j) \in \mathcal{I}_{\text{bound}} \left| \left\| \frac{\partial \Delta \mathbf{p}_{i,j}}{\partial u} \right\| > \left\{ \left\| \frac{\partial \Delta \mathbf{p}}{\partial u} \right\| \right\}_\rho \wedge \left\| \frac{\partial \Delta \mathbf{p}_{i,j}}{\partial v} \right\| > \left\{ \left\| \frac{\partial \Delta \mathbf{p}}{\partial v} \right\| \right\}_\rho \right. \right\}, \quad (3.31)$$

where thresholds $\left\{ \left\| \frac{\partial \Delta \mathbf{p}}{\partial u} \right\| \right\}_\rho$ and $\left\{ \left\| \frac{\partial \Delta \mathbf{p}}{\partial v} \right\| \right\}_\rho$ correspond to the ρ -th percentile of directional derivative magnitudes evaluated at all boundary locations $(i, j) \in \mathcal{I}_{\text{bound}}$.

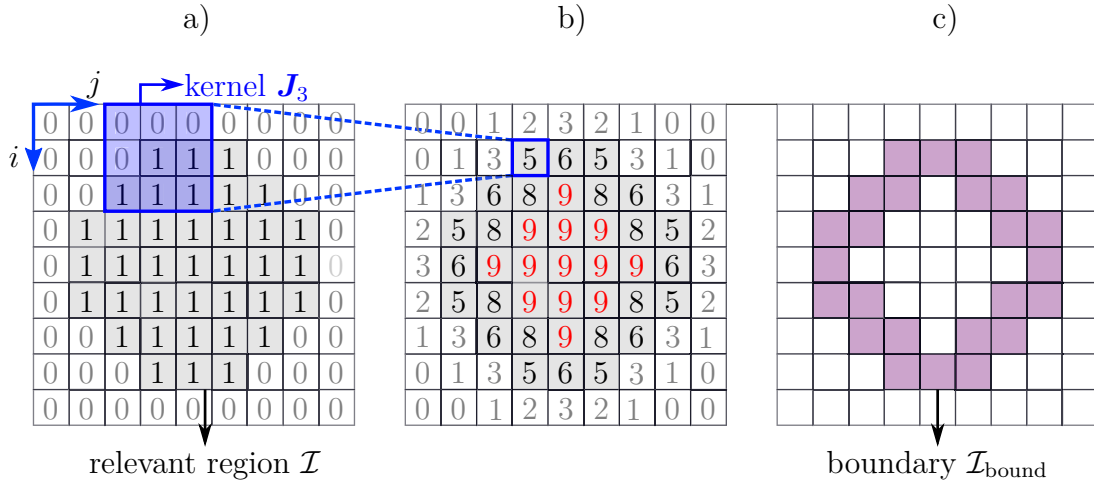


Figure 3.9: Illustration of a) exemplary binary matrix \mathbf{C} , b) convolution $\mathbf{C} * \mathbf{J}_3$ and c) resulting boundary positions using $\alpha = 3$.

In the third step, a smoothing strategy is formulated for these noisy control point deviations in the scan boundary. Basically, each control point shift is replaced by a local average of its valid neighbor deviations. This operation is best described by considering Cartesian directions separately. For example by summarizing the x -control point deviations $\Delta p_{i,j}^x$ in a matrix $\mathbf{P}^x \in \mathbb{R}^{(n+1) \times (m+1)}$, a local deviation average at position (i, j) may be simply obtained as $(\mathbf{P}^x * \mathbf{J}_\beta)_{i,j} / \beta^2$, where the $\beta \times \beta$ kernel \mathbf{J}_β corresponding to Eq. (3.28) collects and adds all surrounding deviations, and the final sum is divided by the kernel area. This simple approach would, however, treat any other peak covered by the kernel as a valid deviation, including position (i, j) itself. Furthermore, this deviation average may provide excessively low values in the case where irrelevant elements with zero deviation are covered. To resolve these issues, a matrix $\mathbf{E} \in \mathbb{R}^{(n+1) \times (m+1)}$ is defined as

$$e_{i,j} := \begin{cases} 1 & \text{if } (i, j) \in \mathcal{I} - \mathcal{I}_{\text{noisy}}, \\ 0 & \text{otherwise.} \end{cases} \quad (3.32)$$

An element-wise product $\mathbf{E} \odot \mathbf{P}^x$ then discards any irrelevant or noisy element. The sum of all surrounding valid elements covered by the kernel at position (i, j) can be obtained as $((\mathbf{E} \odot \mathbf{P}^x) * \mathbf{J}_\beta)_{i,j} - \Delta p_{i,j}^x$ and the total number of valid surrounding elements as $(\mathbf{E} * \mathbf{J}_\beta)_{i,j} - 1$. In this manner, control point deviations with corrected peaks may be defined as new variables $\Delta q_{i,j}^x$ and a corresponding matrix representation $\mathbf{Q}^x \in \mathbb{R}^{(n+1) \times (m+1)}$ as

$$\Delta q_{i,j}^x := \begin{cases} \frac{((\mathbf{E} \odot \mathbf{P}^x) * \mathbf{J}_\beta)_{i,j} - \Delta p_{i,j}^x}{(\mathbf{E} * \mathbf{J}_\beta)_{i,j} - 1} & \text{if } (i, j) \in \mathcal{I}_{\text{noisy}}, \\ \Delta p_{i,j}^x & \text{otherwise.} \end{cases} \quad (3.33)$$

Although previous operation heals deviation peaks, a large jump may still be present at the interface with irrelevant control points. Therefore, the fourth and final post-processing step consists in irrelevant control point smoothing. In this case, it suffices to replace irrelevant control point locations with a local average from previous corrected deviations \mathbf{Q}^x . Thus, the final smoothed deviation field $\Delta r_{i,j}^x$ with a corresponding matrix representation $\mathbf{R}^x \in \mathbb{R}^{(n+1) \times (m+1)}$ may be obtained as

$$\Delta r_{i,j}^x := \begin{cases} (\mathbf{Q}^x * \mathbf{J}_\beta)_{i,j} / \beta^2 & \text{if } (i, j) \notin \mathcal{I}, \\ \Delta q_{i,j}^x & \text{otherwise,} \end{cases} \quad (3.34)$$

where only irrelevant locations $(i, j) \notin \mathcal{I}$ are changed, see Fig. 3.10 for an example. Of course, steps three and four are repeated analogously for Cartesian coordinates y and z . The method parameters α , ρ and β have to be chosen depending on a case study, see next section.

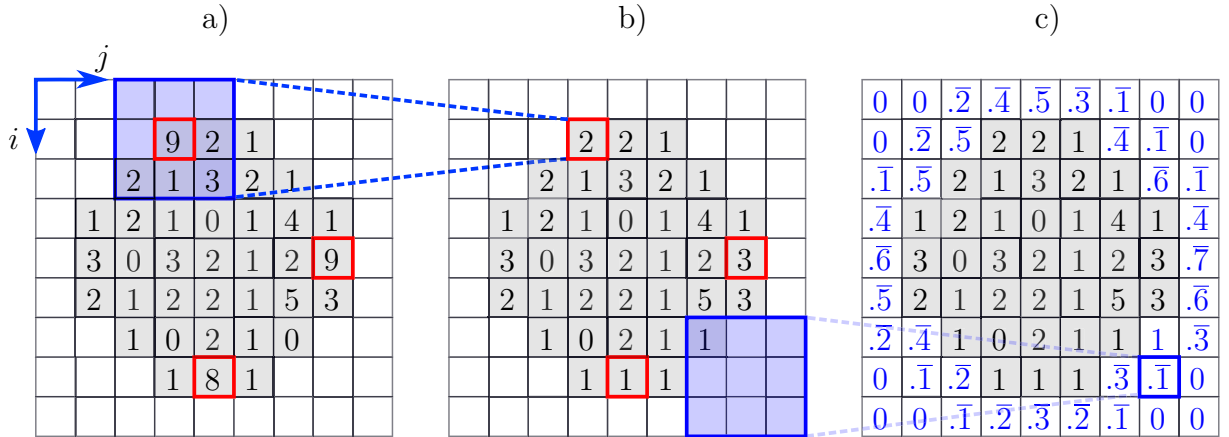


Figure 3.10: Smoothing of a) exemplary deviation matrix \mathbf{P}^x with gray relevant region and red noisy elements by b) defining matrix \mathbf{Q}^x with smoothed relevant region finally resulting in c) matrix \mathbf{R}^x with smoothed irrelevant region where $\beta = 3$.

3.6 Application to HPT Blade Casting Faces

The present section will demonstrate how B-spline morphing may accurately represent complex manufactured surfaces in CAD. For this purpose, the casting faces of an HPT blade will be considered, Fig. 3.11. In contrast to machined primitive faces, free-form casted faces may exhibit rather complex deviation patterns, which is why they are appropriate for testing above morphing methods. First, the procedure will be exemplified for the suction-side shank surface and later for the hot-gas faces.

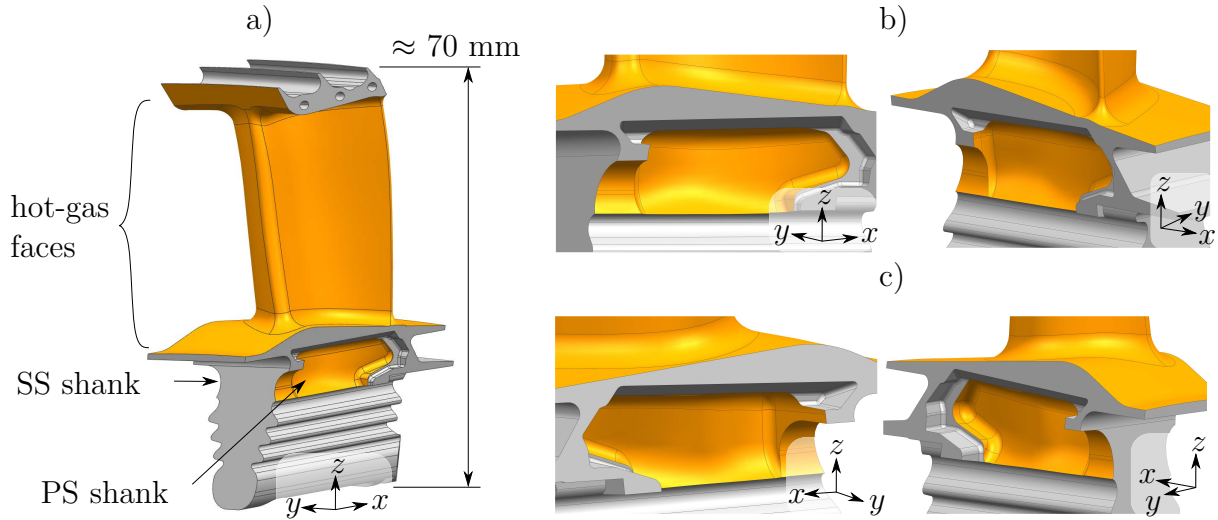


Figure 3.11: Illustration of a) HPT-blade casting faces with details of b) pressure side and c) suction side shank faces.

3.6.1 Suction-Side Shank Face

The suction-side shank face of the HPT blade is constructed using a subtraction body, which is first re-parametrized as B-spline surface with $n \times m = 135 \times 96$ control points, Fig. 3.3. A surface measurement of the manufactured blade is aligned to the blade root and then segmented to the shank profile, Fig. 3.12. For B-spline surface association, a uniform grid \mathcal{G} of $3n \times 3m = 405 \times 288$ surface positions is defined. The first surface position selection is performed using a distance threshold of $d_{th} = 0.50$ mm, which leads to a total of $|\mathcal{U}_d| = 32\,970$ relevant surface positions (3.4). Next, a refinement using a maximum normal deviation angle of $\theta_{th} = 50^\circ$ is performed leading to a total of $|\mathcal{U}| = 31\,640$ relevant surface positions (3.5), Fig. 3.12a. To select relevant control points, a first selection using a distance threshold $d_{cp} = 0.50$ mm is done, which leads to a total of $|\mathcal{I}_d| = 3\,500$ control points (3.7). Next, a periphery refinement with a tangential threshold $\tau_{th} = 2.0$ is applied, which reduces the control point selection to set (3.9) with $|\mathcal{I}_p| = 3\,286$ control points.

Lastly, after considering only control points having a minimum support (3.12) of $n_{\text{sup}} = 4$ measured points, the total number of relevant control points is reduced to $|\mathcal{I}| = 3\,243$, Fig. 3.12b.

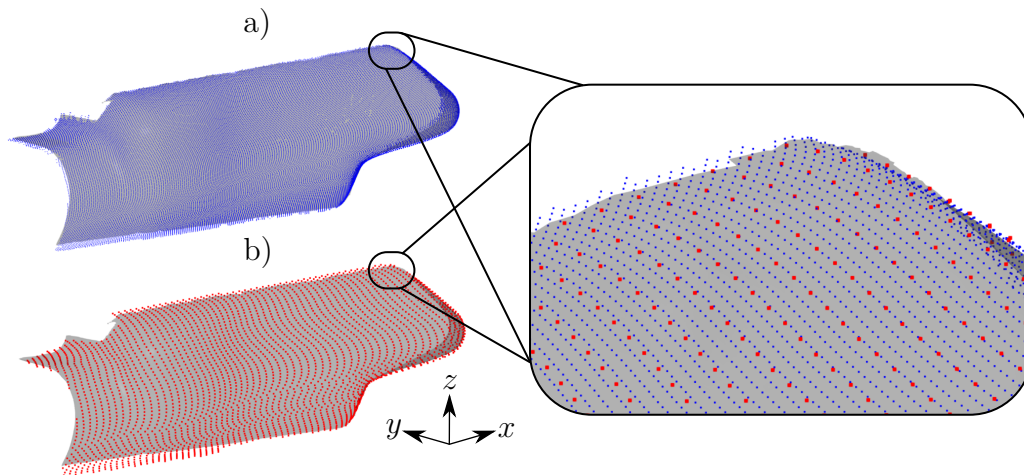


Figure 3.12: Segmented scan of a suction-side shank with a) relevant surface points $\mathbf{s}(u, v)$, $(u, v) \in \mathcal{U}$, and b) control points $\mathbf{p}_{i,j}$, $(i, j) \in \mathcal{I}$.

After these association steps, B-spline morphing is performed to produce the morphed surface shown in Fig. 3.13a. As it may be observed, considerable surface undulations are present at the boundary of the relevant region, which may potentially corrupt CAD subtraction. These imperfections may be significantly reduced by using boundary noise elimination. Here, the width parameter of the boundary region (3.29) is set to $\alpha = 3$, the noise threshold (3.31) is taken from the 60th percentile and the local averaging factor (3.33) is set to $\beta = 3$. As seen in Fig. 3.13b, the resulting post-processed surface shows a much smoother boundary without affecting the interior region.

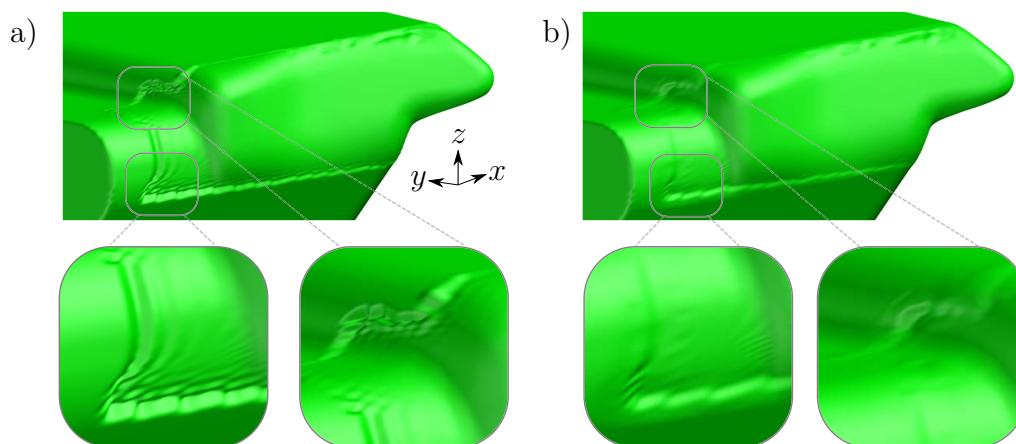


Figure 3.13: Morphed surfaces a) without and b) with boundary noise elimination.

To assess the morphing quality, consider Fig. 3.14 where the morphing error (3.25) is represented as percentile of all relevant surface positions $(u, v) \in \mathcal{U}$. The diagram shows the effect of repeating the entire surface association and fitting procedure several times according to Fig. 3.1. For instance, a second morphing iteration may be considered as beneficial, since it reduces the 95th percentile morphing error from 44 μm to 30 μm (arrow in Fig. 3.14). A third morphing iteration, however, does not seem to provide any additional benefit. For the present validation example, each surface fitting iteration, including surface association, morphing and post-processing, takes about 2 minutes on a personal laptop.

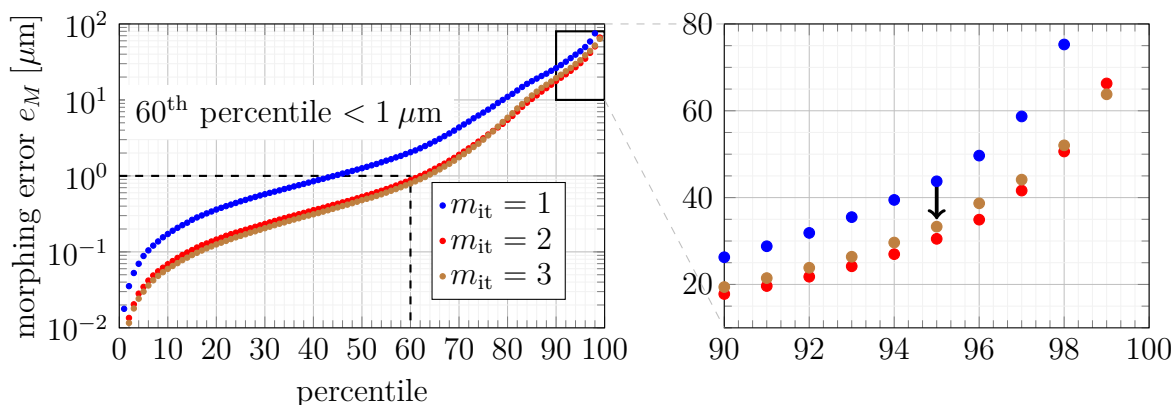


Figure 3.14: Morphing error e_M as percentile of all $|\mathcal{U}| = 31\,640$ relevant surface positions for different morphing iterations m_{it} .

To investigate the spatial distribution of morphing errors, consider Fig. 3.15 which illustrates the morphing error (3.25) throughout the entire segmented surface scan after $m_{it} = 2$ morphing iterations. As it can be observed, the highest morphing errors are located near the scan boundary, which is reasonable since considerable surface discontinuities are expected at this area. On the contrary, all internal regions are morphed with an accuracy well below a typical measurement error $e_S = 25\ \mu\text{m}$, which is a satisfactory result since they typically are the most relevant for analysis.

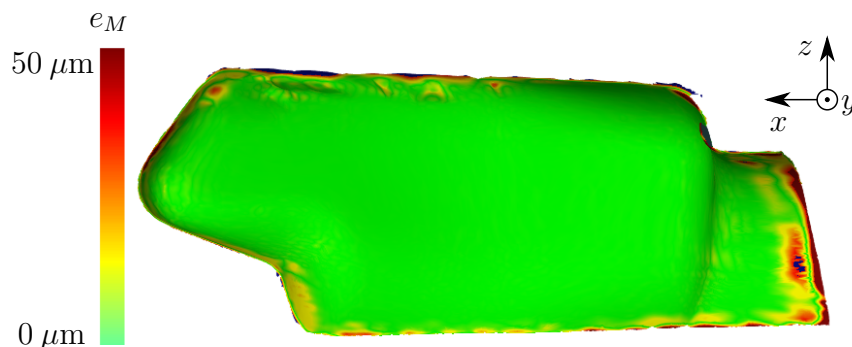


Figure 3.15: Morphing error e_M after $m_{it} = 2$ morphing iterations for the SS shank.

3.6.2 Hot-Gas Faces

The next application example deals with the hot-gas faces of an HPT-blade typically consisting of hub and shroud endwalls, two fillets and the airfoil, Fig. 3.16a. Due to high airfoil curvature at the trailing edge, a scan association based on closest elements may here be insufficient to correctly identify the real trailing edge positions. Since direct B-spline morphing from nominal hot-gas faces may not be sufficiently robust, the fillets and airfoil are first pre-fitted to the surface scan using the program B2P from TU Dresden, Voigt et al. (2017). This approach is based on fitting transverse scan sections to airfoil profiles and has been demonstrated valid for compressor airfoil reconstructions previously, Lange et al. (2009). The resulting pre-fitted hot-gas faces, however, may be further matched to the surface scan to construct a higher-fidelity CAD representation of the HPT blade containing real manufacturing features.

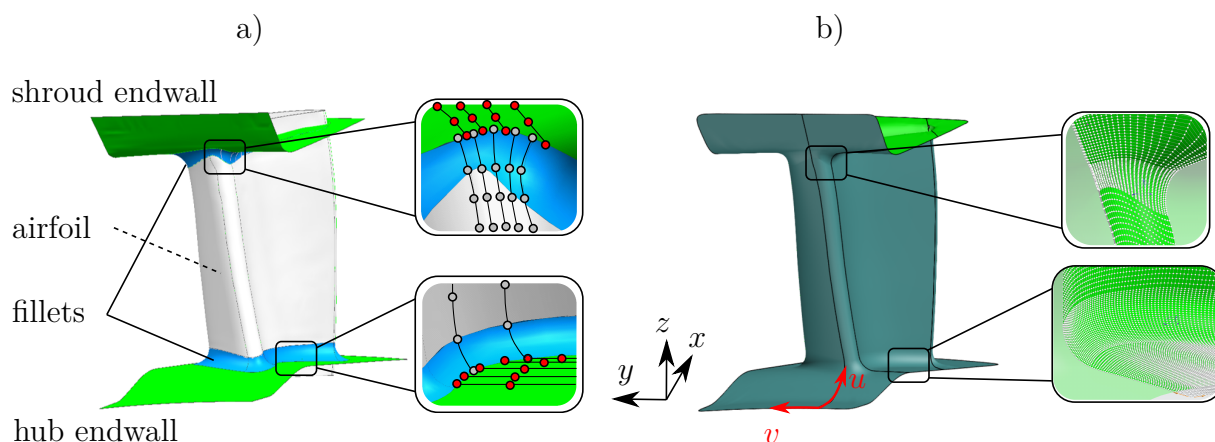


Figure 3.16: Illustration of a) typical hot-gas faces with exemplary control point structure and b) associated re-parametrization as four adjacent B-spline surfaces.

Because each hot-gas face may be designed differently (for instance endwalls vs airfoils), the parametrization, and ultimately the control point structure, may significantly differ from one to another. For deviation representation, however, an inconsistent control point structure at face boundaries may lead to discontinuous face transitions after morphing, Fig. 3.16a. Therefore, all pre-fitted hot-gas faces are re-parametrized as four independent B-spline surfaces corresponding to leading and trailing edges as well as pressure and suction side areas. For this, fine control point grids are constructed, where the number and position of the control points is defined such that similar spacings along u and v directions are guaranteed. Thus, each surface patch is constructed with the same number $n = 246$ of control points along the shared u -direction, and a different number m of control points along the v -direction, i.e., $m = 25, 26, 81$ and 76 for the leading, trailing, pressure and suction areas, respectively, Fig. 3.16b. In this manner, a consistent and smooth control

point structure is guaranteed from hub to shroud, resulting in only four total surface transitions to be treated after morphing. In the present example, each B-spline surface is associated directly to an entire scan without performing individual segmentations. Given the higher complexity of surface deviations compared to the suction-side shank, a finer $6n \times 4m$ grid \mathcal{G} of surface positions is defined for each hot-gas patch, however, the selection of relevant surface positions and control points is performed sequentially using the same method parameters as in the previous example. The settings considered for surface position selection are $d_{\text{th}} = 2.0$ mm and $\theta_{\text{th}} = 50^\circ$, for control point selection $d_{\text{cp}} = 2.0$ mm, $\tau_{\text{th}} = 1.0$ and $n_{\text{min}} = 3$, and for noise elimination $\alpha = 3$, $\rho = 60$ and $\beta = 3$. Finally, the surface transitions are post-processed using the approach described by Gaun et al. (2014), which basically computes small corrections for the two closest control point rows of any surface transition to achieve $G1$ continuity. Two total morphing iterations are performed, each one taking approximately 30 minutes on a personal laptop.

The resulting morphed surfaces in Fig. 3.17 show smooth transitions and rather small surface undulations induced by the airfoil cooling holes. This highlights the versatility of B-spline morphing, making it suitable to accurately capture complex, real manufacturing features. Additionally, B-spline morphing accuracy is excellent throughout the entire interior of hot-gas faces, where a morphing error of $e_M(u, v) < 10 \mu\text{m}$ is observed, Fig. 3.18. Similar to the shank face, minor surface undulations may appear at the scan boundary, such as the endwall edges, but will not interfere in further CAD processing steps.

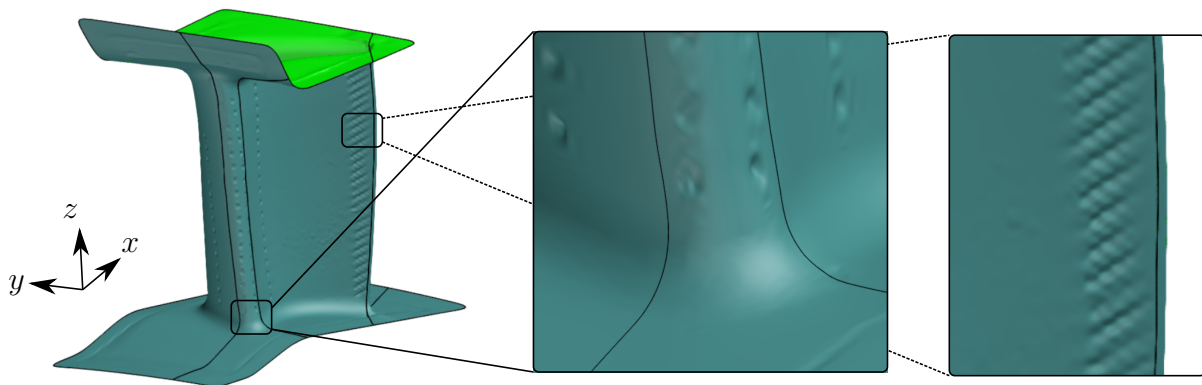


Figure 3.17: Morphed and post-processed hot-gas faces.

Both examples illustrate that accurate CAD representation of complex manufacturing deviations is possible with B-spline re-parametrization and morphing, achieving matching errors well below measurement uncertainty at the face interiors. Furthermore, by taking advantage of a subtractive CAD construction approach, this method may be easily applied to multiple faces so that all relevant deviation information from the scan is effectively transferred to the CAD part. Since each step in the workflow can be automated, large scan

batches may be processed, leading to multiple morphed surfaces of the same nominal CAD face. Statistical methods may then be applied to discover common deviation patterns, which will be the main topic of the next chapter.

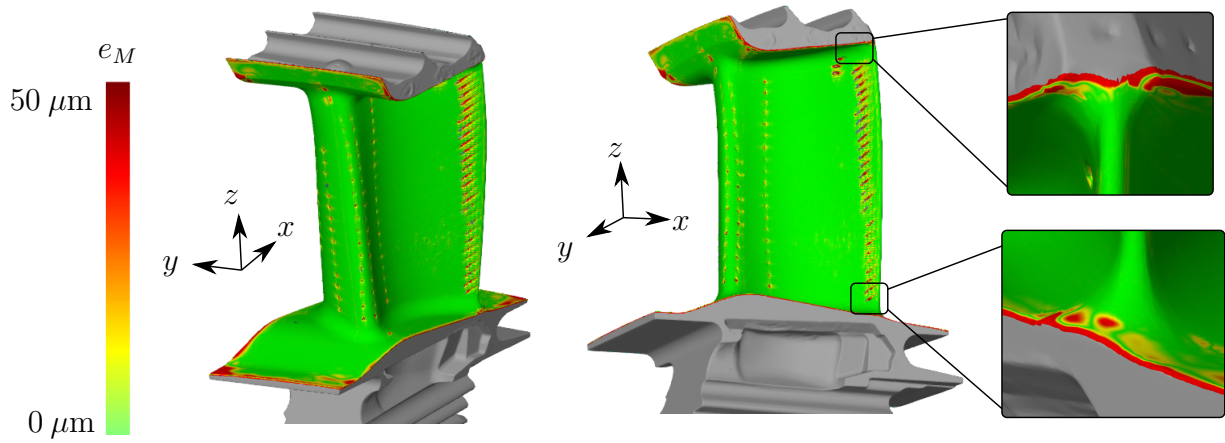


Figure 3.18: Morphing error e_M after $m_{it} = 2$ morphing iterations for the hot-gas faces.

4 Probabilistic Representation of Manufacturing Deviations

In the previous chapter, a series of methods was developed to obtain high-fidelity CAD representations of manufactured parts. Despite allowing graphic interpretation of surface variability, a single manufactured part constitutes only a single realization of a complex, stochastic manufacturing process. Thus, real insight into systematic deviation patterns will only result from using statistical methods. In particular, it is not sufficient to extract deviation patterns directly from statistical analysis of surface measurements; since the ultimate goal is to use deviation information for robust CAD design, an interface between statistical deviation information and CAD representation is required. This leads to the definition of *manufacturing modes*, i.e., a CAD representation of fundamental deviation patterns associated with a manufacturing process, see Sec. 4.1. The manufacturing process typically influences several CAD faces which is why coupled deviation behavior might be expected, leading to the definition of *coupled manufacturing modes* in Sec. 4.2. These deviation patterns extracted from measurement and B-spline morphing procedures are, however, prone to numerous uncertainty sources. Thus, a method will be derived in Sec. 4.3 to distinguish observable manufacturing modes from non-informative deviation patterns. An important step in Sec. 4.4 will be the generation of a probabilistic deviation model allowing to synthetically generate realistic morphed B-spline surfaces without requiring further surface measurements. All these methods will be validated in Sec. 4.5 using shank and hot-gas face deviation information from high-pressure turbine blade measurements.

4.1 Manufacturing Modes

To reveal characteristic surface variability associated with a specific manufacturing process, statistical analysis of multiple manufactured examples is required. Consider, for instance, a CAD face with associated re-parametrized B-spline surface $\mathbf{s}(u, v)$ and multiple associated manufactured instances represented as surface scans $\mathcal{M}^{(1)} \dots \mathcal{M}^{(S)}$. To construct comparable CAD representations of the real surfaces, all scans have to be individually processed according to Chapter 3. In particular, due to individual scan variability, the

scan-surface association process of Secs. 3.2 - 3.3 must be performed individually on each example, thereby leading to different sets \mathcal{U} and \mathcal{I} of relevant surface positions and control points. As a result, each individual CAD representation may be condensed as a control point displacement matrix

$$\mathbf{P} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \Delta p_{i,j}^x & \Delta p_{i,j}^y & \Delta p_{i,j}^z \\ \vdots & \vdots & \vdots \end{bmatrix} =: [\Delta \mathbf{p}^x, \Delta \mathbf{p}^y, \Delta \mathbf{p}^z] \in \mathbb{R}^{G \times 3}, \quad (4.1)$$

where $G = (n + 1)(m + 1)$ corresponds to the total number of control points of the re-parametrized B-spline surface $\mathbf{s}(u, v)$. To facilitate statistical treatment, each control point deviation matrix \mathbf{P} may be reordered as deviation vector

$$\mathbf{q} = [(\Delta \mathbf{p}^x)^\top, (\Delta \mathbf{p}^y)^\top, (\Delta \mathbf{p}^z)^\top]^\top \in \mathbb{R}^{3G} = [\dots \Delta p_{i,j}^x \dots | \dots \Delta p_{i,j}^y \dots | \dots \Delta p_{i,j}^z \dots]^\top, \quad (4.2)$$

where all Cartesian deviation components of the entire B-spline surface have been concatenated sequentially, such that the scan population $\mathcal{M}^{(1)} \dots \mathcal{M}^{(S)}$ is ultimately represented by multiple control point deviation vectors $\mathbf{q}^{(1)} \dots \mathbf{q}^{(S)}$.

To begin a statistical analysis, all surface deviation vectors $\mathbf{q}^{(s)}$ are expressed as a sum of mean $\bar{\mathbf{q}}$ and random deviations $\mathbf{x}^{(s)}$, i.e.,

$$\mathbf{q}^{(s)} = \bar{\mathbf{q}} + \mathbf{x}^{(s)}, \quad (4.3)$$

where the first part may be estimated by the sample mean

$$\bar{\mathbf{q}} = \frac{1}{S} \sum_{s=1}^S \mathbf{q}^{(s)} \in \mathbb{R}^{3G} \quad (4.4)$$

and the remaining deviation component

$$\mathbf{x}^{(s)} := \mathbf{q}^{(s)} - \bar{\mathbf{q}}, \quad (4.5)$$

is individual to every manufactured instance and represents the surface variability of the manufacturing process. Due to the elevated number of control points and manufactured samples, any graphical comparison of individual deviation vectors would provide only limited insight into global deviation patterns. Therefore, a more efficient alternative is needed to derive conclusions about manufacturing variability.

Since all manufactured instances arise from the same manufacturing process, it may be expected that most surface variability is governed by a common set of independent

deviation patterns. Hence, a useful way to analyze dynamic deviation vectors is through identification of the dominant deviation patterns, for example by following the approach described by Urbano et al. (2019a). By assuming that all surface variability of the manufacturing process is contained in the measured sample, the authors propose to represent any arbitrary dynamic deviation vector \mathbf{x} as linear superposition

$$\mathbf{x} := a_1 \mathbf{x}^{(1)} + \dots + a_S \mathbf{x}^{(S)} =: \mathbf{X} \mathbf{a} \quad (4.6)$$

of all measured deviation vectors $\mathbf{x}^{(s)}$ summarized in the data matrix

$$\mathbf{X} = \left[\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(S)} \right] \in \mathbb{R}^{3G \times S}, \quad (4.7)$$

where $\mathbf{a} \in \mathbb{R}^S$ corresponds to their individual contributions to the superposition. To discover the main deviation patterns, data matrix \mathbf{X} may be interpreted as a $3G$ -dimensional point cloud centered around zero, and all orthogonal directions being rich in variability may be searched. These directions correspond to an orthonormal basis for the deviation vector set $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(S)}\}$, which may be found by singular-value decomposition of data matrix \mathbf{X} , i.e.,

$$\mathbf{X} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top \quad (4.8)$$

where the left (\mathbf{U}) and right (\mathbf{V}) singular matrices are orthogonal and square:

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_{3G}] \in \mathbb{R}^{3G \times 3G}, \quad \mathbf{U}^\top \mathbf{U} = \mathbf{I}, \quad (4.9)$$

$$\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_S] \in \mathbb{R}^{S \times S}, \quad \mathbf{V}^\top \mathbf{V} = \mathbf{I}. \quad (4.10)$$

Further,

$$\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_r, 0 \dots 0) \in \mathbb{R}^{3G \times S}, \quad \sigma_1 \geq \dots \geq \sigma_r > 0, \quad (4.11)$$

is a diagonal matrix of singular values σ_i in descending order. The number r of non-zero singular values corresponds to the rank $r = \text{rank}(\mathbf{X}) \leq \min\{3G, S\}$ of the data matrix and, therefore, to the maximum number of basis vectors or rank-one matrices ($\mathbf{u}_i \mathbf{v}_i^\top$) required to completely represent the data matrix, Golub and van Loan (2013):

$$\mathbf{X} = [\mathbf{u}_1 \mathbf{u}_2 \dots] \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \vdots \end{bmatrix} = \sum_{i=1}^S \sigma_i \mathbf{u}_i \mathbf{v}_i^\top \equiv \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top. \quad (4.12)$$

Substitution in superposition (4.6) results in

$$\begin{aligned}\mathbf{x} = (\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top) \mathbf{a} &= \sum_{i=1}^r (\sigma_i \mathbf{u}_i \mathbf{v}_i^\top) \mathbf{a} \equiv \sum_{i=1}^r (\sigma_i \mathbf{v}_i^\top \mathbf{a}) \mathbf{u}_i \\ &=: \sum_{i=1}^r b_i \mathbf{u}_i,\end{aligned}\tag{4.13}$$

which demonstrates that any deviation vector \mathbf{x} may be represented in a dimension-reduced space spanned by the orthonormal basis $\{\mathbf{u}_1, \dots, \mathbf{u}_r\}$. According to Urbano et al. (2019a), each normalized basis vector \mathbf{u}_i may be interpreted as a control point deviation vector of the re-parametrized B-spline surface $\mathbf{s}(u, v)$, and thus it represents an independent deviation pattern considered as *manufacturing mode*. Specifically, Eq. (4.13) shows that the same set of manufacturing modes may be utilized to represent any arbitrary displacement vector sufficiently by its *mode magnitudes* b_i .

To compute the mode magnitudes directly from the data matrix, one may collect all manufacturing modes into a reduced left-singular matrix

$$\mathbf{U}_r = [\mathbf{u}_1, \dots, \mathbf{u}_r] \in \mathbb{R}^{3G \times r}, \quad \mathbf{U}_r^\top \mathbf{U}_r = \mathbf{I},\tag{4.14}$$

which is also orthogonal. Then the superposition (4.13) may be simplified to

$$\mathbf{x} = \mathbf{U}_r \mathbf{b} \quad \text{where} \quad \mathbf{b} = [b_1, \dots, b_r]^\top.\tag{4.15}$$

Pre-multiplication with \mathbf{U}_r^\top and use of orthogonality (4.14) results in

$$\mathbf{U}_r^\top \mathbf{x} = \mathbf{U}_r^\top \mathbf{U}_r \mathbf{b} = \mathbf{I} \mathbf{b} \equiv \mathbf{b}\tag{4.16}$$

or equivalently

$$b_i := \mathbf{u}_i^\top \mathbf{x}, \quad i = 1 \dots r,\tag{4.17}$$

which states that the mode amplitude b_i of any given deviation vector \mathbf{x} is equivalent to its projection onto the i -th manufacturing mode \mathbf{u}_i .

With (4.13), the representation (4.3) of any particular control point deviation vector $\mathbf{q}^{(s)}$ may be specified as

$$\begin{aligned}\mathbf{q}^{(s)} &= \bar{\mathbf{q}} + \mathbf{x}^{(s)}, \\ &= \bar{\mathbf{q}} + \sum_{i=1}^r b_i^{(s)} \mathbf{u}_i,\end{aligned}\tag{4.18}$$

where the deviation $\mathbf{x}^{(s)} = \mathbf{q}^{(s)} - \bar{\mathbf{q}}$ has been replaced by linear superposition (4.15) using

individual manufacturing mode amplitudes

$$b_i^{(s)} := \mathbf{u}_i^T \mathbf{x}^{(s)}. \quad (4.19)$$

In contrast to representation (4.3), Eq. (4.18) shows that any control point deviation (4.2) results as a combination of manufacturing process properties (mean deviation component $\bar{\mathbf{q}}$ and manufacturing modes $\{\mathbf{u}_i, \dots, \mathbf{u}_r\}$) and coefficients associated with the particular manufacturing instance (mode amplitudes $b_i^{(s)}$). In particular, the mean deviation component as well as any manufacturing mode may be reorganized as control point grid of the re-parametrized B-spline surface $\mathbf{s}(u, v)$ which allows CAD visualization of these deviation patterns.

4.2 Coupled Manufacturing Modes

Manufacturing of any mechanical part typically involves a sequence of various manufacturing steps chosen to provide different surface characteristics according to the design intent. Some manufacturing steps, may treat a specific subset of part faces, thereby making them vulnerable to the same sources of manufacturing uncertainty and probably inducing common deviation patterns. In the case of an HPT blade, for instance, the geometry of shank and hot-gas faces is largely defined by the same casting process, which is why some sort of coupled deviation behavior may be expected. Therefore, the present section introduces a concept of coupled manufacturing modes as deviation patterns occurring simultaneously in multiple faces.

Consider two different CAD faces A and B of the same mechanical part sharing the same history of production steps. As in previous section, assume that a representative sample of S manufactured instances has been selected and measured to investigate their surface deviation behavior. To obtain a CAD representation of the surface measurements, both faces have to be re-parametrized as B-spline surfaces $\mathbf{s}^A(u, v)$ and $\mathbf{s}^B(u, v)$ and processed according to Chapter 3 to obtain control point displacements associated with each re-parametrized B-spline surface. For a particular surface measurement, the resulting control point deviations at both faces may be represented as deviation matrices (4.1), i.e.,

$$\mathbf{P}^A =: [\Delta \mathbf{p}^{x,A}, \Delta \mathbf{p}^{y,A}, \Delta \mathbf{p}^{z,A}] \in \mathbb{R}^{G \times 3}, \quad (4.20)$$

$$\mathbf{P}^B =: [\Delta \mathbf{p}^{x,B}, \Delta \mathbf{p}^{y,B}, \Delta \mathbf{p}^{z,B}] \in \mathbb{R}^{H \times 3}, \quad (4.21)$$

where

$$G = (n^A + 1)(m^A + 1) \quad \text{and} \quad H = (n^B + 1)(m^B + 1) \quad (4.22)$$

correspond to the total number of control points of the re-parametrized B-spline surfaces $\mathbf{s}^A(u, v)$ and $\mathbf{s}^B(u, v)$, respectively. To facilitate statistic analysis, both deviation matrices may be joined together as a single joint deviation vector (4.2), i.e.,

$$\mathbf{q} := \left[\left(\Delta \mathbf{p}^{x,A} \right)^\top, \left(\Delta \mathbf{p}^{y,A} \right)^\top, \left(\Delta \mathbf{p}^{z,A} \right)^\top, \right. \\ \left. \left(\Delta \mathbf{p}^{x,B} \right)^\top, \left(\Delta \mathbf{p}^{y,B} \right)^\top, \left(\Delta \mathbf{p}^{z,B} \right)^\top \right]^\top \in \mathbb{R}^{3(G+H)}, \quad (4.23)$$

where all columns of the deviation matrices have been joined first according to Cartesian direction as in Eq. (4.2) and then according to face order. As a result, each joint deviation vector (4.23) contains all deviation information of a particular manufacturing instance, such that the entire surface variability of all measured manufactured samples is contained in deviation vectors $\mathbf{q}^{(1)} \dots \mathbf{q}^{(S)}$.

The statistic representation of multiple faces is analogous to that of a single-face. First, measured joint deviation vectors $\mathbf{q}^{(s)}$ are decomposed into mean and individual components according to Eq. (4.3), however, in this case the mean component

$$\bar{\mathbf{q}} = \frac{1}{S} \sum_{s=1}^S \mathbf{q}^{(s)} = \begin{bmatrix} \bar{\mathbf{q}}^A \\ \bar{\mathbf{q}}^B \end{bmatrix} \quad (4.24)$$

summarizes mean deviation vectors

$$\bar{\mathbf{q}}^A := [\bar{q}_1, \dots, \bar{q}_{3G}]^\top \in \mathbb{R}^{3G}, \quad (4.25)$$

$$\bar{\mathbf{q}}^B := [\bar{q}_{3G+1}, \dots, \bar{q}_{3(G+H)}]^\top \in \mathbb{R}^{3H} \quad (4.26)$$

and the individual deviation components

$$\mathbf{x}^{(s)} := \mathbf{q}^{(s)} - \bar{\mathbf{q}} = \begin{bmatrix} \mathbf{x}^{A,(s)} \\ \mathbf{x}^{B,(s)} \end{bmatrix} \quad (4.27)$$

summarize face-specific deviation vectors

$$\mathbf{x}^{A,(s)} := [x_1^{(s)}, \dots, x_{3G}^{(s)}]^\top \in \mathbb{R}^{3G}, \quad (4.28)$$

$$\mathbf{x}^{B,(s)} := [x_{3G+1}^{(s)}, \dots, x_{3(G+H)}^{(s)}]^\top \in \mathbb{R}^{3H}. \quad (4.29)$$

The collection $\mathbf{x}^{(1)} \dots \mathbf{x}^{(S)}$ of deviation vectors represents the surface variability arising from the manufacturing process. As explained in Sec. 4.1, these deviation components are best interpreted as superposition of fundamental deviation patterns. Since both faces A and B are expected to exhibit coupled deviation behavior, independent treatment of

face-specific dynamic deviation vectors would destroy any existing correlation between both faces, thereby eliminating valuable information. Therefore, during statistical analysis of multiple CAD faces, each individual deviation vector (4.27) has to be considered as an indivisible process realization, and only after identification of coupled deviation patterns may be separated for independent CAD examination.

Hence, data matrix (4.7) may be now constructed with joint deviation vectors (4.27) and application of singular value decomposition (4.8)–(4.13) allows representation of any individual deviation vector \mathbf{x} as superposition of orthonormal basis vectors

$$\mathbf{x} =: \sum_{i=1}^r b_i \mathbf{u}_i = \sum_{i=1}^r b_i \begin{bmatrix} \mathbf{u}_i^A \\ \mathbf{u}_i^B \end{bmatrix}. \quad (4.30)$$

In this case, each vector \mathbf{u}_i contains a characteristic deviation pattern represented as control point deviations of both re-parametrized B-spline surfaces $\mathbf{s}^A(u, v)$ and $\mathbf{s}^B(u, v)$ and consequently may be regarded as *coupled manufacturing mode*. The face-specific sections of each coupled manufacturing mode may be extracted according to the face ordering in Eq. (4.23) as

$$\mathbf{u}_i^A := [u_{1,i}, \dots, u_{3G,i}]^\top, \quad (4.31)$$

$$\mathbf{u}_i^B := [u_{3G+1,i}, \dots, u_{3(G+H),i}]^\top \quad (4.32)$$

and utilized for face-specific CAD visualization. It's worth pointing out that, although each coupled manufacturing mode \mathbf{u}_i is normalized, the face-specific sections (4.31) and (4.32) obtained by vector partitioning do not necessarily have unitary magnitude. Furthermore, because the deviation pattern on both faces is coupled, a single mode amplitude b_i is sufficient to describe the deviation state of both faces by Eq. (4.30).

Once the singular-value decomposition of individual deviation components $\mathbf{x}^{(1)} \dots \mathbf{x}^{(S)}$ has been completed, the representation of any arbitrary joint deviation vector $\mathbf{q}^{(s)}$ may be reformulated as

$$\mathbf{q}^{(s)} = \bar{\mathbf{q}} + \mathbf{x}^{(s)} = \bar{\mathbf{q}} + \sum_{i=1}^r b_i^{(s)} \mathbf{u}_i = \begin{bmatrix} \bar{\mathbf{q}}^A \\ \bar{\mathbf{q}}^B \end{bmatrix} + \sum_{i=1}^r b_i \begin{bmatrix} \mathbf{u}_i^A \\ \mathbf{u}_i^B \end{bmatrix}, \quad (4.33)$$

where the face-specific deviation components $\bar{\mathbf{q}}^A$, $\bar{\mathbf{q}}^B$ and manufacturing modes \mathbf{u}_i^A , \mathbf{u}_i^B may be reorganized as control point deviations of re-parametrized B-spline surfaces $\mathbf{s}^A(u, v)$ and $\mathbf{s}^B(u, v)$ and used for CAD visualization of coupled deviation patterns.

4.3 Representation Capacity of Manufacturing Modes

Characterization of manufacturing variability as mean deviation plus superposition of manufacturing modes already reveals valuable information about manufacturing uncertainty. As a further step, the relative contribution of each manufacturing mode to the overall geometric variability may be now quantified. This mode representation capacity is crucial for understanding the statistical complexity of surface deviations, i.e., the amount of dominant manufacturing modes responsible for surface variability. Hence, this section will present an estimation of mode representation capacity based on the approximation error obtained by truncating superposition (4.33) to a smaller subset of manufacturing modes.

Until now, the representation capacity of any given manufacturing mode has been estimated only for individual surface measurements using mode amplitudes (4.17). However, since they are specifically associated to a given surface measurement, they are rather inadequate to quantify the overall mode contribution. Instead, singular values $\sigma_1 \dots \sigma_r$ in Eq. (4.11) obtained from singular-value decomposition (4.8) of data matrix \mathbf{X} and associated with manufacturing modes $\mathbf{u}_1 \dots \mathbf{u}_r$ correspond somehow to a global measure of control point variability of the measured sample. In particular, the i -th singular value σ_i is associated with the statistical variance of dot products $\{\mathbf{u}_i^\top \mathbf{x}^{(1)} \dots \mathbf{u}_i^\top \mathbf{x}^{(S)}\}$, i.e., the projections of all computed control point deviations $\mathbf{x}^{(1)} \dots \mathbf{x}^{(S)}$ onto the i -th manufacturing mode \mathbf{u}_i as will be shown in Sec. 4.4. Unfortunately, because control points have to be weighted by basis functions $N_{i,q}(u)N_{j,q}(v)$ to produce surface deviations (3.14), control point variability does not directly correspond to Cartesian variability of morphed B-spline surfaces. Hence, singular values remain sub-optimal for estimating mode representation capacity.

A reasonable approach consists of incrementally truncating manufacturing mode superposition (4.33) and observing its impact onto the approximation accuracy of morphed surfaces, Urbano et al. (2019a). Recall that any given surface scan may be accurately represented in CAD by a vector $\mathbf{q}^{(s)}$ of control point displacements which produce a morphed B-spline surface $\hat{\mathbf{s}}(u, v)$ according to Eqs. (4.2) and (3.14). The same control point displacement vector $\mathbf{q}^{(s)}$ may be represented as linear superposition (4.33), where a total number r of manufacturing modes describe all measured surface deviations completely. To determine the representation capacity of a smaller subset consisting of the first t manufacturing modes only, a truncated control point displacement vector (4.18) may be defined as

$$\mathbf{q}^{(s), t} := \bar{\mathbf{q}} + \sum_{i=1}^t b_i^{(s)} \mathbf{u}_i = \bar{\mathbf{q}} + \mathbf{U}_t \mathbf{b}^{(s)}, \quad t \leq r, \quad (4.34)$$

where t corresponds to an arbitrary truncation order, r to the full number of manufacturing modes and \mathbf{U}_t to the first t columns of the left singular matrix (4.14). In this manner, the difference between truncated $\mathbf{q}^{(s),t}$ and precise control point displacement vector $\mathbf{q}^{(s)}$ may be considered as a measure of the representation strength of the first t manufacturing modes. Since manufacturing modes \mathbf{u}_i are sorted according to non-increasing singular values σ_i , it is expected that the difference $\|\mathbf{q}^{(s)} - \mathbf{q}^{(s),t}\|$ reduces as t approaches r , such that the mode representation capacity may be effectively analyzed incrementally.

To physically interpret the mode reconstruction capacity, the control point difference $\|\mathbf{q}^{(s)} - \mathbf{q}^{(s),t}\|$ may be translated into a Cartesian difference between morphed surface $\hat{\mathbf{s}}(u, v)$ (corresponding to the best CAD representation possible for the given surface measurement) and a truncated B-spline surface obtained from the reduced set of t manufacturing modes. Using reordering (4.23), the truncated control point displacements $\Delta\mathbf{p}_{i,j}^{(s),t}$ may be deduced for a specific manufacturing instance (s) from the truncated vector (4.34) to build up the truncated B-spline surface

$$\hat{\mathbf{s}}^{(s),t}(u, v) := \mathbf{s}(u, v) + \sum_{i=0}^n \sum_{j=0}^m N_{i,q}(u)N_{j,q}(v) \left(\mathbf{p}_{i,j} + \Delta\mathbf{p}_{i,j}^{(s),t} \right). \quad (4.35)$$

The surface difference between morphed and truncated B-spline surfaces may be then formulated locally as a reconstruction error

$$e_R^{(s),t}(u, v) := \left| \left(\hat{\mathbf{s}}^{(s)}(u, v) - \hat{\mathbf{s}}^{(s),t}(u, v) \right)^\top \mathbf{n}(u, v) \right| \forall (u, v) \in \mathcal{U}, \quad (4.36)$$

as absolute value of the projected distance onto the normal vector $\mathbf{n}(u, v)$ taken from the nominal surface, Fig. 4.1.

Since reconstruction error $e_R^{(s),t}(u, v)$ quantifies the representation capacity of the first t manufacturing modes for an individual surface measurement at a specific surface location (u, v) , it is still not informative about overall mode participation in the whole measured sample. Therefore, to obtain a global measure for the representation capacity, reconstruction error (4.36) must be evaluated statistically for all S surface measurements and r truncation orders. This results in a nested approach, where for a fixed truncation order t , local reconstruction errors are first quantified for a single B-spline surface and later for the entire measured sample, Fig. 4.2. The internal error quantification loop is performed for each surface measurement by computing reconstruction errors (4.36) for all relevant surface positions $(u, v) \in \mathcal{U}$ and a given truncation order t , resulting in a total of $|\mathcal{U}|$ error values for each surface measurement. These resulting errors are then quantified with a percentile ρ leading to a single reconstruction error metric $\bar{e}_R^{(s)}$ for each surface measurement. In the external error quantification loop all error values $\bar{e}_R^{(1)} \dots \bar{e}_R^{(S)}$ obtained

from all surface measurements are then quantified with another percentile ρ' to produce a global reconstruction error $\bar{e}_R(t)$ associated with a specific truncation order t . This metric may be computed for various truncation values $t \in [1, r] \subset \mathbb{N}$ and may thus illustrate how much representation capacity is gained by a growing number of manufacturing modes.

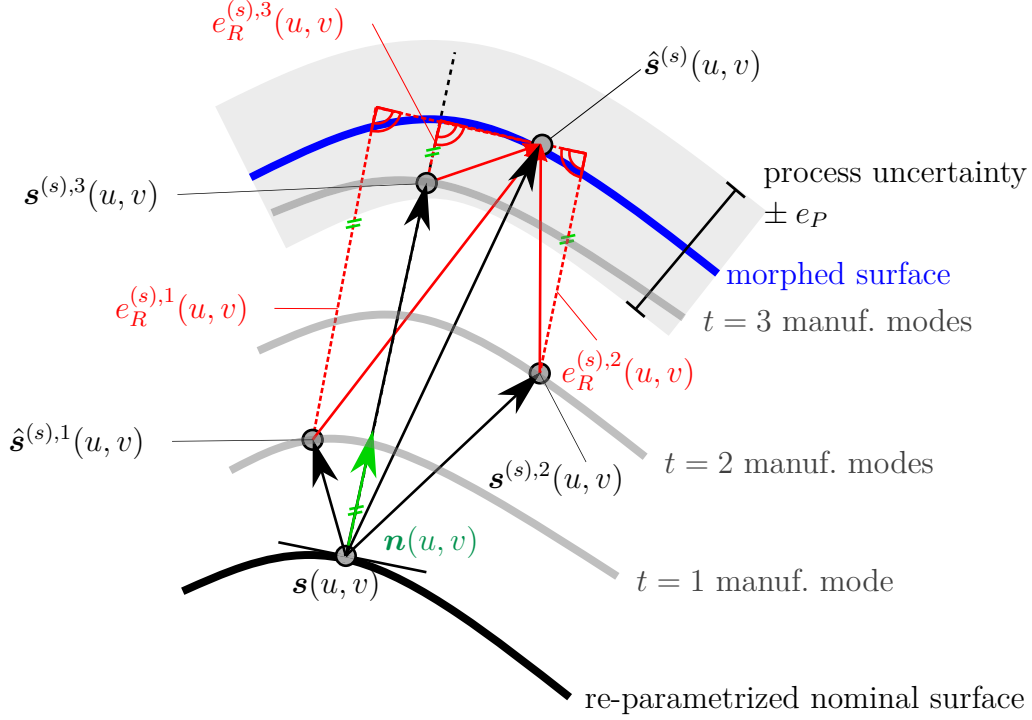


Figure 4.1: Reconstruction error $e_R^{(s),t}$ for t manufacturing modes.

Before determining which manufacturing mode contributes most to the surface variability, it is important to recall that all morphed B-spline surfaces have been influenced by measurement, re-parametrization and morphing uncertainties. Thus, each local position of the morphed surface $\hat{\mathbf{s}}(u, v)$ must be associated with an uncertainty interval approximately equivalent to the compounding uncertainties of measurement error e_S discussed in Sec. 2.4, re-parametrization error $e_C(u, v)$ defined by Eq. (3.1) and morphing error $e_M(u, v)$ given by (3.25). However, since re-parametrization error $e_C(u, v)$ and morphing error $e_M(u, v)$ were defined for a particular surface location $(u, v) \in \mathcal{U}$, new representative global error metrics \bar{e}_C and \bar{e}_M associated to both processes are required, which may be also computed via percentiles in the same fashion as reconstruction error $\bar{e}_R(t)$ in Fig. 4.2. In this manner, the uncertainty propagation due to measurement, re-parametrization and morphing processes may be condensed into an overall process error e_P approximated by the error propagation rule for surface metrology according to Li et al. (2021), i.e.,

$$e_P := \sqrt{e_S^2 + \bar{e}_C^2 + \bar{e}_M^2}. \quad (4.37)$$

Because the global reconstruction error $\bar{e}_R(t)$ resulting from manufacturing mode truncation may be interpreted as the largest surface discrepancy when considering only t manufacturing modes, it may be directly compared to the process error e_P . As a consequence, not every increase in reconstruction capacity is physically relevant, since any point located below a distance $\pm e_P$ from the morphed surface belongs to the process uncertainty boundary and is, therefore, equally valid. In Fig. 4.1, for example, this knowledge threshold is surpassed after the first three truncation modes. This allows a definition of dominant (or observable) manufacturing modes as the number

$$t^* := \min t \quad \text{s.t.} \quad \bar{e}_R(t) \leq e_P \quad (4.38)$$

of manufacturing modes required to obtain a global reconstruction error below the process uncertainty.

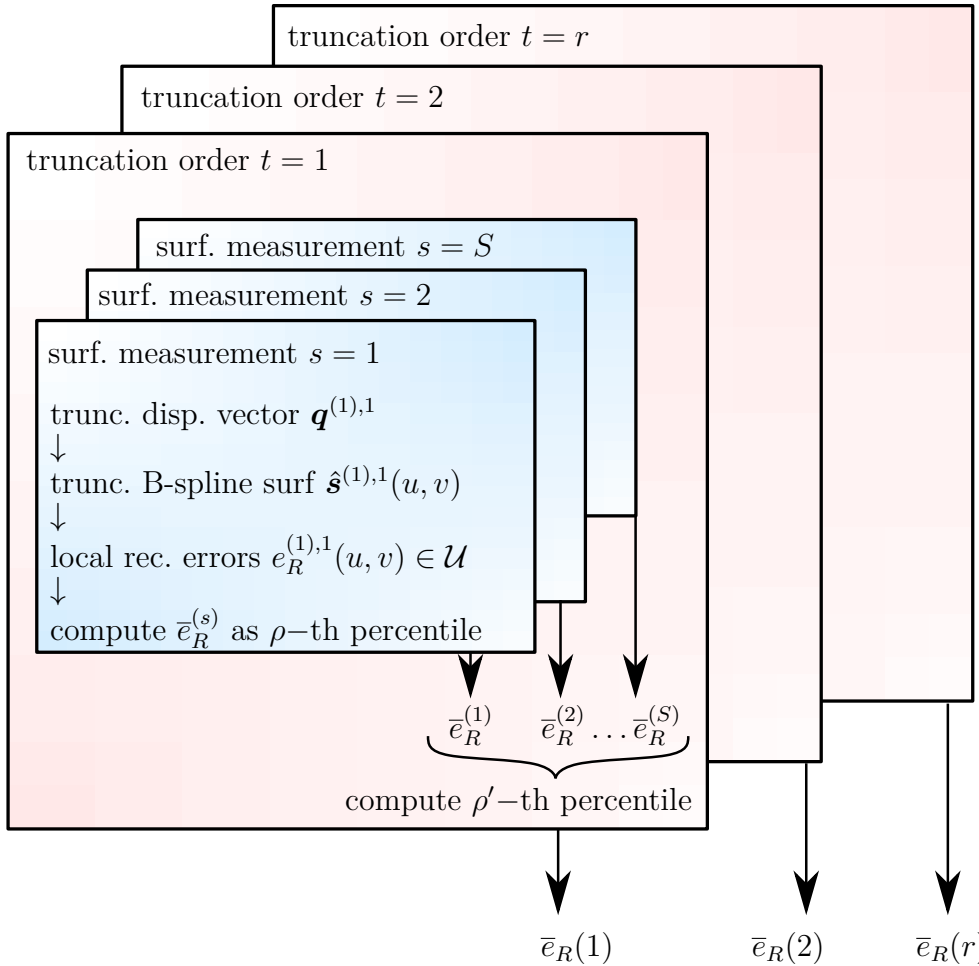


Figure 4.2: Computation of global reconstruction error $\bar{e}_R(t)$ using percentiles.

This information reduction process effectively separates dominant deviation patterns from any potential experimental and numerical noise leaked throughout the measurement and morphing processes. Moreover, the number t^* of dominant manufacturing modes serves as an appropriate metric for statistical complexity of surface deviations between different part populations. For instance, when comparing batches of multiple manufacturing processes, the process least affected by mode interactions and thus with less surface variability may be identified by the smallest number of dominant manufacturing modes.

4.4 Probabilistic Description of Surface Deviations

Aside from identifying main surface deviation patterns, understanding and quantifying the impact of surface variability on part performance is essential for ensuring reliability. Such an analysis may be executed statistically by selecting a large sample of manufactured geometries, creating corresponding CAD representations and running individual performance simulations. Quantification of performance scatter is then possible via histograms, percentiles or statistical moments of the resulting performance distribution. This statistical study—referred as uncertainty quantification—is the cornerstone of any robust design process and relies heavily on sample size for accurate identification of rare events. To avoid underestimation of performance variability due to lack of sufficient surface measurements, the present section describes how experimental mode amplitudes may be used to obtain a probabilistic description of surface deviations. This procedure will facilitate generation of realistic additional manufacturing instances by sampling from the obtained probability distributions, thus enhancing performance variability quantification without requiring further surface measurements.

To elaborate a probabilistic description of surface deviations, it is necessary to understand where and how specifically randomness arises in the truncated representation (4.34) of control point deviations. Since the mean deviation component $\bar{\mathbf{q}}$ and manufacturing modes \mathbf{u}_i are properties of the manufacturing process, they are valid for all surface measurements and, therefore, may be treated as constants in a probabilistic representation. Mode amplitudes $b_i^{(s)}$, on the other hand, correspond to unique realizations for every manufacturing instance and may even exhibit correlations across different mode orders. Thus, they may be approached by a multivariate probabilistic model.

This probabilistic modeling problem has been addressed in numerous experimental studies, particularly involving airfoil manufacturing deviations. In their investigations, Lamb (2005) and later Lange et al. (2012) conducted surface measurements of manufactured airfoils and represented surface variability differently, the former using a series of physical geometric parameters (such as chord length, leading/trailing edge thickness, chord angles,

etc.) and the latter using three-dimensional displacements at different radial sections. The parameter differences were then processed via Principal Component Analysis (PCA), such that decomposition into manufacturing modes with respective mode amplitudes was possible. In both investigations, independent sampling from experimental mode amplitudes and subsequent airfoil reconstruction led to non-realistic airfoil shapes, despite considering experimental correlation matrices. This raised the hypothesis that additional inter-modal interactions might exist which have been ignored by the PCA and motivated the later use of Parzen windows, i.e., a computationally expensive Kernel-based method to fit non-parametric multi-dimensional probability distributions. The authors further hypothesized that the measurement datasets might have been rather inhomogeneous, i.e., involve surface measurements of varying manufacturing characteristics, which may possibly explain the presence of several multi-modal amplitude distributions.

Both studies are immensely valuable as they highlight the importance of careful judgment of dataset quality. Based on these ideas, three major considerations are formulated before computing probabilistic mode amplitudes for B-spline surfaces: First, despite having formulated a robust B-spline morphing process, the construction of morphed surfaces is inevitably prone to further measurement, re-parametrization and morphing errors, which may manifest as rare, inexact morphing surfaces within the measured sample, i.e., sample outliers. Second, since the main focus of probabilistic surface modeling should be a generative manufacturing process instead of individual manufacturing realizations, any attempt to construct a probabilistic distribution must consider the possibility of existence of such outliers within the measurement sample. This paradigm significantly elevates the resulting probability distribution representativeness because it avoids measurement errors to propagate back into the probabilistic manufacturing model. Third, all surface measurements considered in the modeling procedure should arise from exactly the same manufacturing process, which may be interpreted as same manufacturing batch, supplier, construction parameters, etc. If this is not the case, an inhomogeneous manufacturing sample might lead to multi-modal or skewed mode amplitudes, which may further give rise to additional interactions irreproducible with conventional covariance matrices.

As a consequence, an essential ingredient of the present probabilistic modeling consists of robust quantification of mode covariance in the potential presence of morphed surface outliers within the dataset. Since visual inspection is typically infeasible for outlier identification given a large number of morphed surfaces, a statistical method is formulated based on the scatter produced by mode amplitude comparisons. Consider the scatter diagram in Fig. 4.3, where exemplary first and second mode amplitudes b_1 and b_2 of a sample \mathcal{S} of $|\mathcal{S}| = 55$ morphed surfaces have been represented as blue and red dots. It is apparent that the morphed surfaces corresponding to red dots have a disproportionate

mode participation, which may possibly indicate contaminated surface measurements. To further quantify their discrepancy, principal directions of statistical variance may be found using Principal Component Analysis. The first step consists of introducing estimators for sample mean and covariance for experimental mode amplitudes $\mathbf{b}^{(s)}$:

$$\boldsymbol{\mu} := \frac{1}{S} \sum_{s=1}^S \mathbf{b}^{(s)}, \quad (4.39)$$

$$\mathbf{C} := \frac{1}{S-1} \sum_{s=1}^S (\mathbf{b}^{(s)} - \boldsymbol{\mu}) (\mathbf{b}^{(s)} - \boldsymbol{\mu})^\top, \quad (4.40)$$

where mode amplitudes $\mathbf{b}^{(s)} \in \mathbb{R}^t$ may be truncated to a chosen number t of truncation modes, in the present example $t = 2$. According to Jolliffe (2002), the directions of maximum variance are equivalent to the eigenvectors $\mathbf{e}_1, \dots, \mathbf{e}_t$ of covariance matrix (4.40), and their eigenvalues $\lambda_1 \dots \lambda_t$ are equivalent to the sample variance along each principal direction. As a consequence, the scatter range may be conveniently represented graphically as an ellipse with semi-axes a_1 and a_2 being proportional to the sample standard deviations along each principal axis, e.g. $a_1 = 3\sqrt{\lambda_1}$ and $a_2 = 3\sqrt{\lambda_2}$.

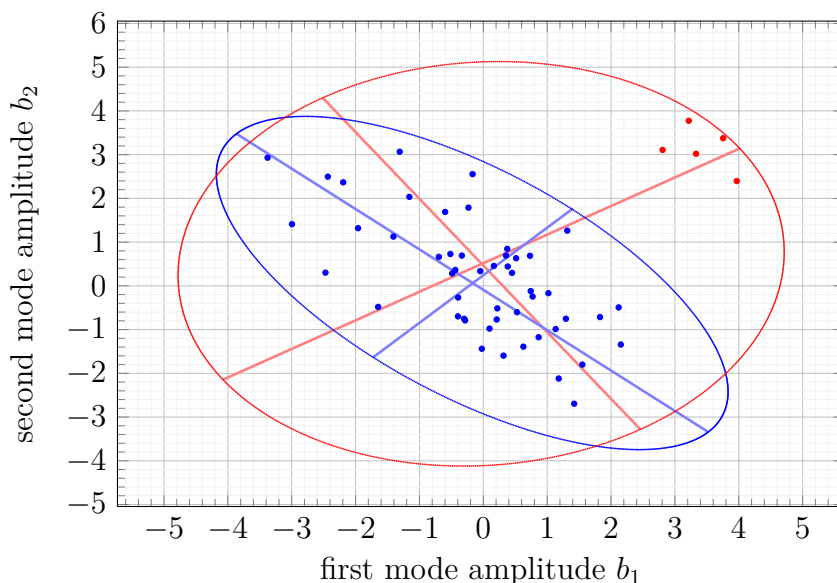


Figure 4.3: Exemplary mode amplitudes with outliers marked as red.

As demonstrated by the red ellipse and red semi-axes in Fig. 4.3, if the red points are included in the covariance estimation (4.40), they will predict an obviously erroneous relationship between first and second modes. In contrast, disregarding these five noisy observations will produce the blue ellipse resulting in a more concentrated scatter and a better description of the true mode correlation.

Therefore, the desired method for outlier elimination should identify a sample subset $\mathcal{S}' \subset \mathcal{S}$ which maximally concentrates scatter throughout all mode amplitude dimensions given a desired robust sample size $|\mathcal{S}'|$. This may be achieved by the Minimum Covariance Determinant method developed by Rousseeuw (1984), which may be explained using the enclosing ellipse analogy. Since the objective of the Minimum Covariance Determinant is to identify the subset of points which maximally concentrates scatter, this is equivalent to calculating the ellipsoid with smallest semi-axes, which may be conveniently obtained by minimizing the product of all covariance eigenvalues, i.e., the covariance matrix determinant $\det(\mathbf{C}) = \lambda_1 \lambda_2 \dots \lambda_{t-1} \lambda_t$. An efficient algorithm for this purpose was presented by Rousseeuw and van Driessen (1999), where the minimization problem was solved iteratively by drawing a random subset $\mathcal{S}' \subset \mathcal{S}$, computing its sample mean $\boldsymbol{\mu}'$ and covariance matrix \mathbf{C}' , and quantifying a standardized distance from all $|\mathcal{S}|$ original observations to the new centroid $\boldsymbol{\mu}'$ by means of the Mahalanobis distance

$$d(s) := \sqrt{(\mathbf{b}^{(s)} - \boldsymbol{\mu}')^\top (\mathbf{C}')^{-1} (\mathbf{b}^{(s)} - \boldsymbol{\mu}')} \quad \forall s \in [1, S] \subset \mathbb{N}, \quad (4.41)$$

which is essentially a multivariate extension of the standard score $d = (b - \mu)/\sigma$, where μ and σ correspond to univariate sample mean and standard deviation, respectively. After sorting all points $\mathbf{b}^{(s)}$ according to increasing distance, the first S' observations are considered as best cluster. The process is repeated several times and the subset \mathcal{S}' with minimum covariance determinant is finally chosen.

Once the subset \mathcal{S}' of uncontaminated surface measurements has been identified, an associated probability distribution may be fitted. For the case of deviation modeling via B-spline control point deviations and after multiple test cases on turbine blade geometries, it has been repeatedly verified that the vast majority of observable mode amplitudes may be assumed to be normally distributed, see validation examples in Sec. 4.5. Thus, all mode amplitude vectors $\mathbf{b}^{(s)}$ may be represented as realizations of a random vector \mathbf{b} with a multivariate Gaussian distribution:

$$\mathbf{b} \sim N(\boldsymbol{\mu}', \mathbf{C}'), \quad (4.42)$$

where the robust sample mean $\boldsymbol{\mu}'$ and robust covariance matrix \mathbf{C}' have been calculated from mode amplitudes $\mathbf{b}^{(s)}$ of the subset \mathcal{S}' of uncontaminated scans. Consequently, a probabilistic representation of manufacturing variability may be obtained by replacing the random amplitude vector (4.42) in truncated mode superposition (4.34):

$$\mathbf{q} = \bar{\mathbf{q}} + \mathbf{U}_t \mathbf{b} = \bar{\mathbf{q}} + b_1 \mathbf{u}_1 + \dots + b_i \mathbf{u}_i + \dots + b_t \mathbf{u}_t = \bar{\mathbf{q}} + \sum_{i=1}^t b_i \mathbf{u}_i \quad (4.43)$$

using random mode amplitudes b_i up to the desired truncation order t . In this manner, the probabilistic surface variability for a given manufacturing process is sufficiently described by mean deviation vector $\bar{\mathbf{q}}$, observable manufacturing modes \mathbf{U}_{t^*} and normally distributed mode amplitudes \mathbf{b} defined by robust mean vector $\boldsymbol{\mu}'$ and robust covariance matrix \mathbf{C}' .

The fact that mode amplitudes may be efficiently modeled as multivariate Gaussian distribution has several advantages, particularly for statistical sampling and deviation modeling. First, sampling from probabilistic models which involve different distribution families requires expensive iterative correlation correction steps, such as restrictive pairing, Dandekar et al. (2002), which may be entirely spared when sampling from traditional multivariate Gaussian models. Second, the probability of occurrence of any surface measurement given its mode amplitudes may be conveniently approximated by the probability density

$$p(\mathbf{b}; \boldsymbol{\mu}', \mathbf{C}') = \frac{\exp\left(-\frac{1}{2}(\mathbf{b} - \boldsymbol{\mu}')^\top (\mathbf{C}')^{-1} (\mathbf{b} - \boldsymbol{\mu}')\right)}{\sqrt{(2\pi)^t \det(\mathbf{C}')}}. \quad (4.44)$$

This is a useful feature when evaluating predictive deviation models, i.e., when predicting manufacturing modes and mode distributions for new geometric designs, Chapter 5. There, the predicted deviation model will undergo a cross-validation step by comparing estimated and measured surface variability. Because the probability of occurrence of individual manufacturing instances is available according to (4.44), the statistical divergence between estimated and real manufacturing processes may be quantified, thus allowing a quantitative stochastic deviation model validation, see Sec. 5.4.

4.5 Application to HPT Blade Structural Uncertainty Quantification

The following demonstrates the probabilistic modeling of surface variability for hot-gas and shank faces of the HPT blade previously analyzed in Sec. 3.6, see Fig. 3.11. Re-parametrized B-spline surfaces for suction-side shank and hot-gas faces are taken from Figs. 3.3 and 3.16, respectively, and the pressure side shank is re-parametrized similarly as the suction side with a finer 203×163 control point grid. These three re-parametrized surfaces are then fitted to a total of $S = 56$ turbine blade surface measurements leading to three morphed surfaces for each scan. All resulting control point displacements associated to the same scan are then joined together as a large joint deviation vector $\mathbf{q} \in \mathbb{R}^{291\ 651}$ as defined in Eq. (4.23). Subtraction (4.27) of mean deviation $\bar{\mathbf{q}}$ and concatenation results in a large data matrix $\mathbf{X} \in \mathbb{R}^{291\ 651 \times 56}$. Singular-value decomposition delivers the desired coupled manufacturing modes. To assess the mode representation capacity, global reconstruction errors $\bar{e}_R(i)$ are

computed following Fig. 4.2 for several truncation orders t , using a percentile $\rho = 99$ for error quantification within relevant normalized pairs $(u, v) \in \mathcal{U}$ and a percentile $\rho' = 90$ for quantification within the entire measurement sample. To define the overall process uncertainty, global metrics for measurement, re-parametrization and morphing errors are needed. Measurement uncertainty was already estimated in Sec. 2.4 as $e_S = 25 \mu\text{m}$. Given the higher control point density of shank and hot-gas faces compared to the simple re-parametrization example in Fig. 2.4, a global re-parametrization error can be confidently estimated as $\bar{e}_C = 2 \mu\text{m}$. Lastly, the global morphing error is estimated as the 90th morphing error percentile from the shank example in Fig. 3.14, e.g. $\bar{e}_M = 20 \mu\text{m}$. As a result, process uncertainty is estimated as $e_P = 32 \mu\text{m}$ according to Eq. (4.37).

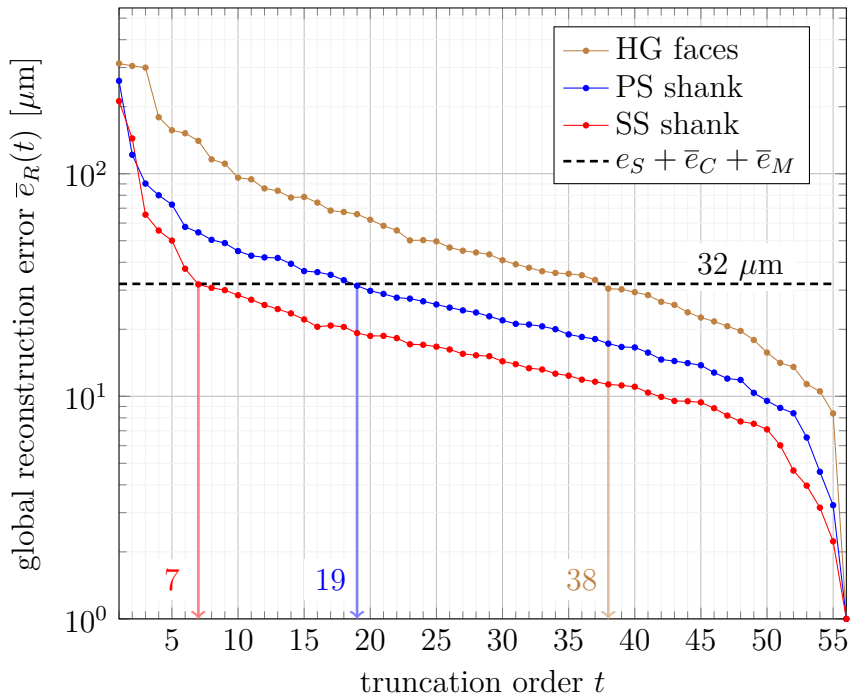


Figure 4.4: Global reconstruction error $\bar{e}_R(t)$ for different truncation orders t .

Although most casted surfaces typically exhibit complex surface deviations, the error reconstruction diagram in Fig. 4.4 suggests different deviation complexities between shanks and hot-gas faces. The suction-side shank (red curve), for instance, is rapidly described by just 7 manufacturing modes, whereas hot-gas faces (brown) require 38 modes to provide an appropriate CAD reconstruction below process uncertainty e_P . This complexity results from the fact that hot-gas faces account not only for airfoil but also blend-radii and endwall deviations, Fig. 3.16, leading thus to a much larger deviation space. To validate the reconstruction error computation, an arbitrary surface measurement is reconstructed from Eq. (4.34) with all $t^* = 38$ observable manufacturing modes, such that local errors $e_R^{(s),t}(u, v)$ can be computed from Eq. (4.36) for all relevant measured

points. As shown in Fig. 4.5, the reconstruction accuracy effectively falls below the process uncertainty e_P for all major regions of the casted surfaces, such that the number $t^* = 38$ of observable modes is correct.

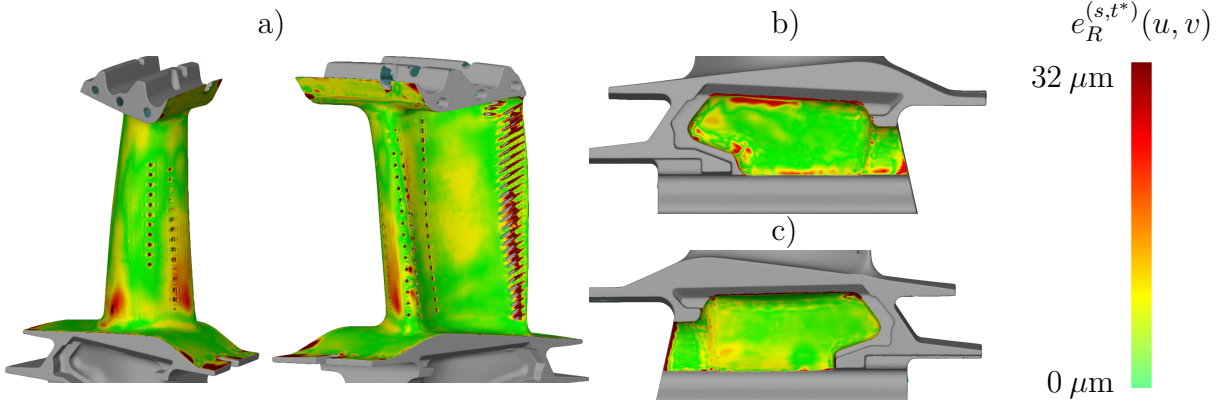


Figure 4.5: Local reconstruction error $e_R^{(s),t^*}(u,v)$ using $t^* = 38$ observable modes for a) hot-gas, b) suction- and c) pressure-side shank faces.

The probabilistic modeling of surface deviations starts with accurate identification of mode correlation. By directly considering all surface measurements as noise-free process realizations, potential correlations may be erroneously hidden by the presence of measurement or morphing outliers. Consider Fig. 4.6a, where all surface measurements are plotted as blue and red circles according to their 3rd and 8th mode amplitudes. PCA of all measurements produces the red, large enclosing ellipse with semi-axes parallel to the coordinate system, thus indicating no apparent mode correlation. Alternatively, the Minimum Covariance Determinant method is utilized to identify a subset \mathcal{S}' of $|\mathcal{S}'| = 50$ realizations from all 56 truncated mode amplitudes $\mathbf{b}^{(s)} \in \mathbb{R}^{38}$ which concentrates multidimensional scatter best. The obtained optimal set of uncontaminated observations with its corresponding enclosing ellipse are colored blue in Fig. 4.6a, whereas the potential outliers are shown as red circles. The resulting principal scatter directions (shown as blue semi-axes) indicate a more realistic estimation of mode correlation, and two evidently outlying observations originating from surface measurement irregularities were correctly identified. Although the remaining four noisy observations might seem mislabeled, this is not necessarily the case since Fig. 4.6a shows only a projection of the 38–dimensional space onto two dimensions, whereas the Minimum Covariance Determinant considers scatter along all possible mode combinations. Thus, these four realizations may possibly display a disproportionate participation on other mode combinations.

To better illustrate relative modal dependencies between i –th mode amplitude $b_i^{(s)}$ and j –th mode amplitude $b_j^{(s)}$, the covariance matrix \mathbf{C} may be normalized according to

variances $C_{i,i}$ leading to the Pearson correlation matrix \hat{C} with elements

$$\hat{C}_{i,j} := \frac{C_{i,j}}{\sqrt{C_{i,i} C_{j,j}}}. \quad (4.45)$$

This quantity is a measure of linear correlation between any mode combination and may be severely affected by the presence of outliers. For instance, uncontaminated 3rd and 8th mode amplitudes have a Pearson correlation of $\hat{C}_{3,8} = -0.4$, see blue ellipse in Fig. 4.6a and its associated correlation value shown in Fig. 4.6b as framed square. On the other side, consideration of outliers aligns the ellipse semi-axes to the coordinate system (red ellipse), erroneously implying mode independence and therefore a false correlation $\hat{C}_{3,8} = 0$. In fact, the Pearson correlation matrix resulted numerically identical to the identity matrix in case all surface measurements were considered, erroneously implying that no modal dependencies were observed. However, after using exclusively the set \mathcal{S}' of uncontaminated observations, several mode correlations were effectively revealed. This can be verified by the symmetric, positive-definite Pearson correlation matrix in Fig. 4.6b.

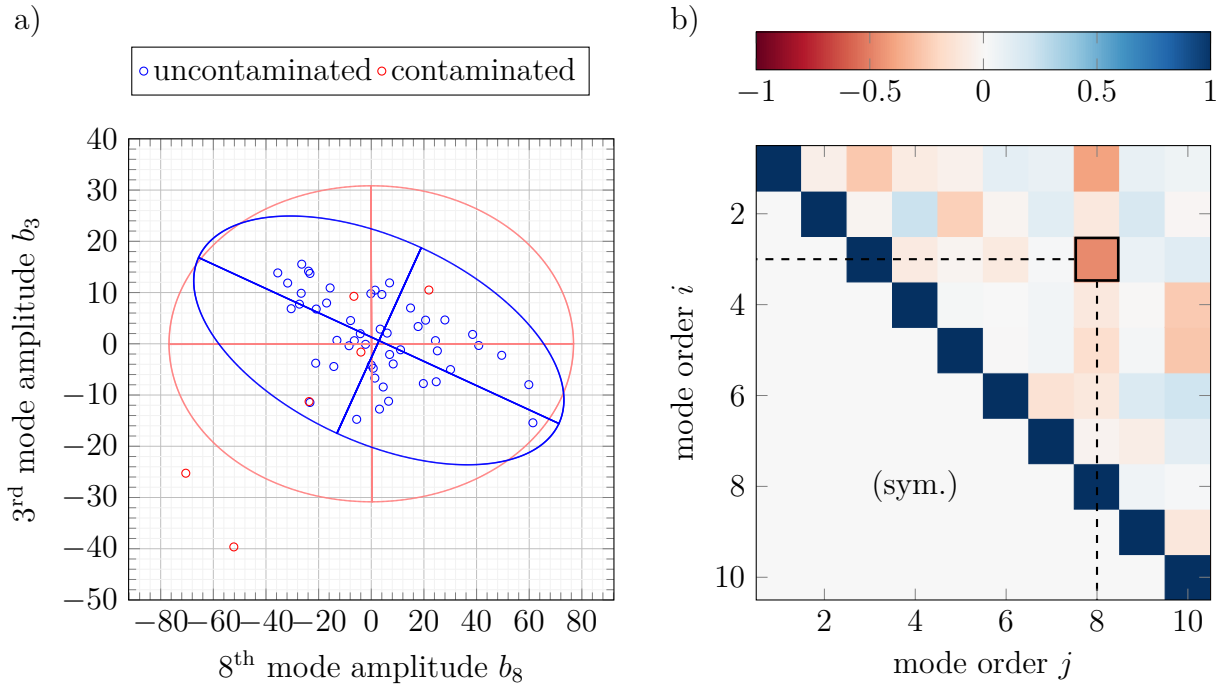


Figure 4.6: Correlation analysis using a) scatter diagram for 8th and 3rd mode amplitudes and b) Pearson correlation matrix for first 10 modes.

The elimination of measurement outliers not only improves correlation identification, it typically enhances mode amplitude normality. This might be verified by testing all $|\mathcal{S}|$ amplitude realizations $b_i^{(s)}$ of the same mode order i with a test for statistical normality.

Here, a characteristic value (also referred as test statistic) associated to the experimental sample is computed, which measures the strength of normality. This statistic is then used to compute a p -value, which quantifies the probability of obtaining an equal or better statistic from the experimental sample assuming it is normally distributed, Cooksey (2020). If the p -value falls below a certain level of significance, e.g. 0.05, it is extremely unlikely that the test statistic may arise from the dataset and thus the normality hypothesis must be rejected. In the case, however, where the p -value is greater, no significant evidence for rejecting the normality hypothesis is found. This test is performed using the statistic described by D’Agostino (1971), first considering all mode amplitudes obtained from all 56 original surface measurements, Fig. 4.7a, and then only with the subset of 50 uncontaminated surface measurements, Fig. 4.7b. In the first case, six modes (represented as red dots) fail the test for normality, whereas in the second case, only three modes below the 15th order fail the test. A fourth failure is also present, but closer to the p -value threshold. Additionally, the overall p -value seems increased for many mode orders, which again emphasizes the positive impact of outlier elimination. A detailed observation of normality-test failures, when considering uncontaminated measurements, is given in Fig. 4.7c. Despite showing abnormal occurrences at the distribution tails, such as increased extreme observations (2nd and 14th mode amplitudes) or reduced ones (6th mode amplitude), the distributions are mostly uni-modal and concentrated towards their centers. This is the reason why in these cases a Gaussian approximation (shown as black line) is accepted.

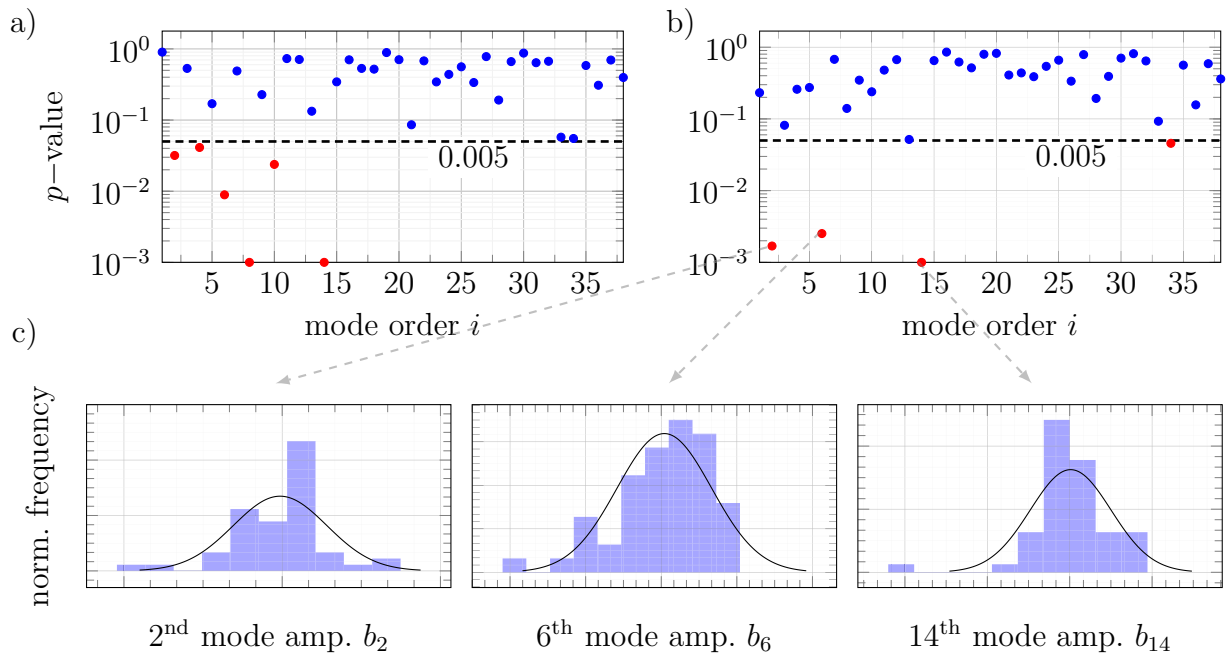


Figure 4.7: Test for mode normality using a) original vs. b) uncontaminated surface measurements and c) corresponding test failure histograms.

Having justified a multivariate Gaussian model for surface variability, robust mean $\boldsymbol{\mu}'$ and covariance \mathbf{C}' estimators for mode amplitudes may be obtained from Eqs. (4.39) and (4.40) using uncontaminated observations \mathcal{S}' . Now, additional synthetic morphed surfaces may be generated by sampling mode amplitudes from the multivariate Gaussian distribution (4.42), introducing them into superposition (4.34), and using the obtained control point displacements to generate synthetic morphed B-spline surfaces with Eq. (3.14). A broad geometric comparison of synthetic vs. experimental surfaces is given for three high-curvature turbine blade sections in Fig. 4.8, i.e., airfoil leading and trailing edges, and hub fillet radius (see nomenclature in Fig. 4.9). Contrary to Lange et al. (2012) who obtained unrealistic reconstructed airfoil geometries after truncated superposition of principal deviation components, all synthetic airfoils in the current approach are realistic and smooth, which validates the present mode correlation estimation and consequently minimizes the possibility of having relevant inter-modal dependencies being ignored by singular-value decomposition.

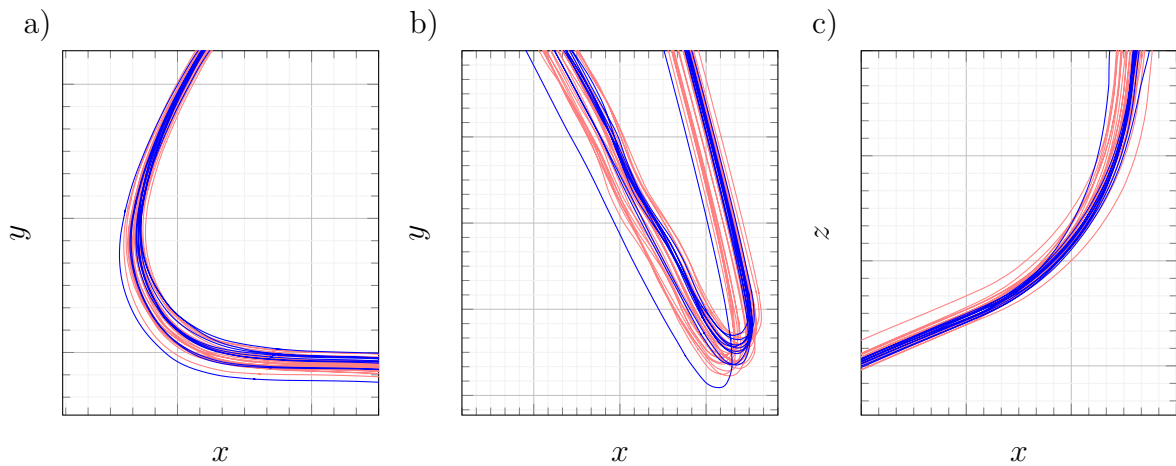


Figure 4.8: Comparison between 20 synthetic (pink) vs. 10 real (blue) morphed surfaces for a) airfoil leading-edge, b) airfoil trailing edge and c) hub fillet-radius.

A final validation of the probabilistic model for surface variability is obtained by propagating manufacturing deviations into a performance model of the turbine blade. In the present case, the impact of shank and hot-gas surface deviations on structural integrity is investigated by means of finite-element analysis. For each measured or synthetic manufacturing instance of hot-gas and two shank faces, the entire CAD geometry of the turbine blade is reconstructed and meshed, where refinement zones to neck, shank and airfoil fillets are assigned, Fig. 4.9. The mesh consists of approximately 0.7 million second-order tetrahedral elements with average element volume of 0.19 mm^3 and average element distortion (ratio circumscribed/inscribed sphere) of 4.3. The simulation considers maximum engine take-off conditions and starts with a thermal model of the turbine

blade, where temperatures and mass flows of hot and cooling gas are used to compute a detailed three-dimensional temperature field on the HPT blade by means of conductive and convective heat transfer. Temperature-dependent material properties of nickel alloy CMSX-4 are applied. The obtained temperature field is then considered in a second-order elastic analysis, where additional loads caused by part deformation are considered. Boundary conditions consist of previously calculated element-displacements of the turbine disk applied to the blade root, gas pressures acting on hot-gas faces and cooling channels, and rotational speed of about 16 000 rpm.

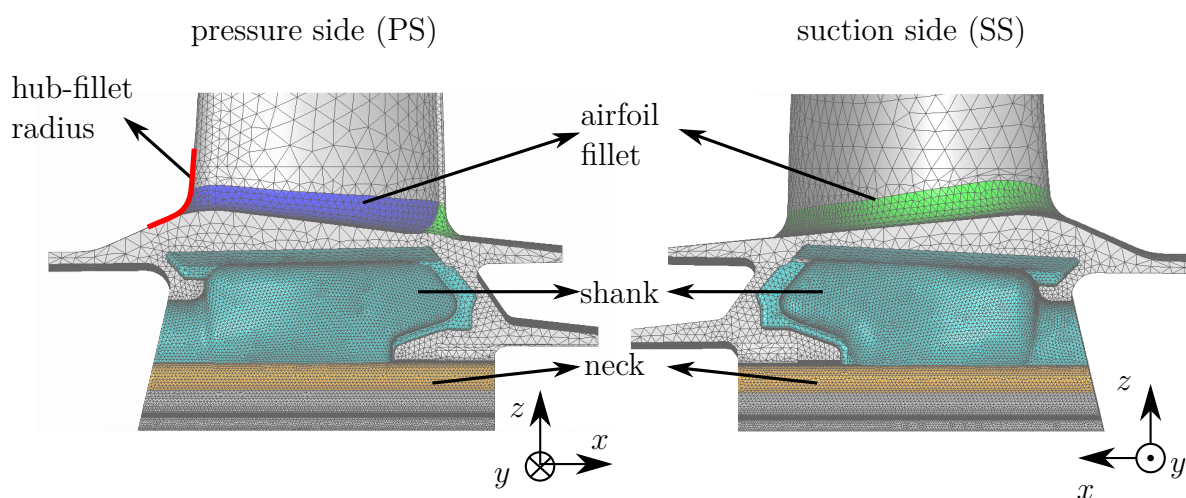


Figure 4.9: Finite-element mesh with domain details.

A comparison of stress variability between 50 uncontaminated scans and 200 synthetic morphed surfaces for different turbine blade positions is shown in Fig. 4.10. Each diagram displays maximum stress/strain at selected mesh domains, see nomenclature in Fig. 4.9. For the three cases shown—SS airfoil fillet, PS neck and SS shank—synthetic morphed surfaces successfully refine frequency distributions of induced stress variability. In general, the most frequent value (mode) of each stress histogram may be loosely associated with mean deviation components of morphed and synthetic surfaces, whereas scatter amplitude around the mode is rather associated with manufacturing modes. It may seem, therefore, remarkable that modeling mode amplitudes as multivariate normal distribution does not alter natural skewness of induced stress variability. It is, however, reminded that in any complex geometry normally distributed geometric deviations may easily lead to disproportionate stresses if unfavorably located. In case of a turbine blade, maximum stresses at the shank walls are severely affected by sectional wall thickness. Thus, shank deviations resulting in cross-section area reduction may induce disproportionate stress increments. Similarly, the scatter diagram in Fig. 4.10 shows how correlated manufacturing deviations between both shank walls also provoke correlated mechanical strains. This

relationship is correctly amplified using synthetic morphed surfaces and demonstrates how adverse effects of manufacturing variability are worst in the pressure shank, where maximum strain is typically elevated.

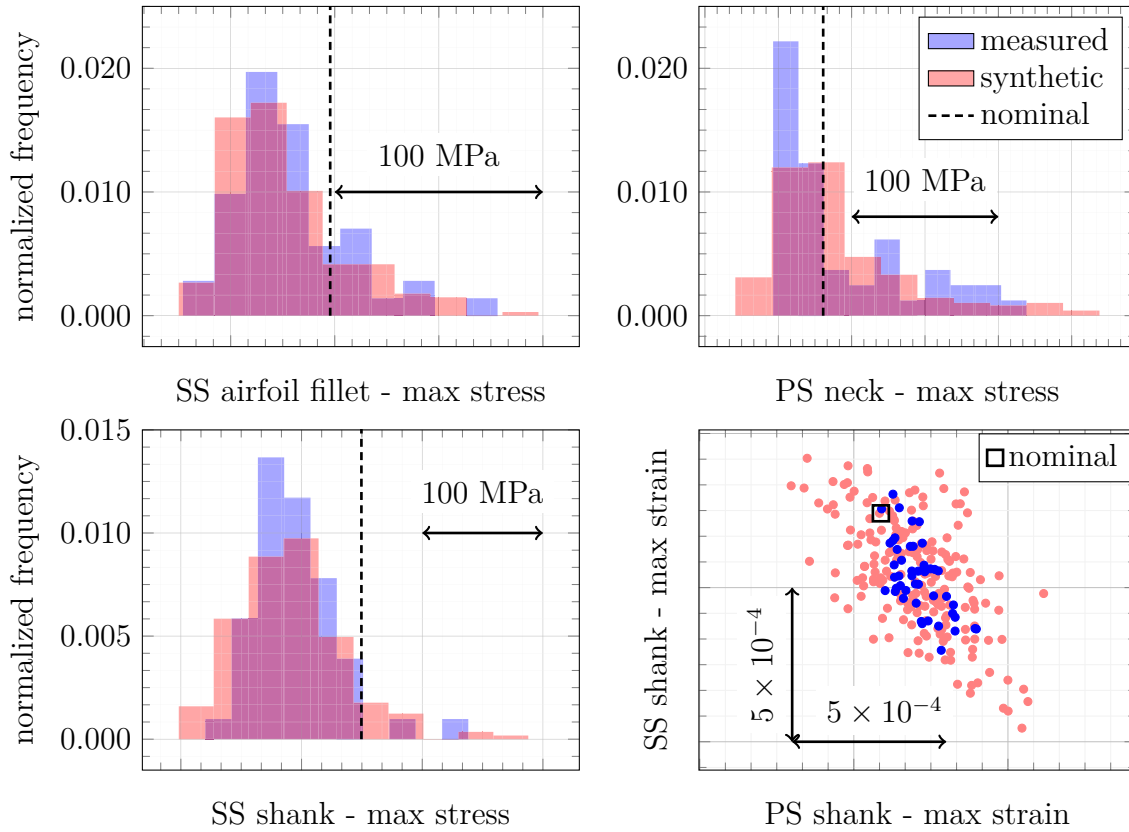


Figure 4.10: Stress variability for measured and synthetic morphed surfaces.

Accurate statistical reconstructions of induced stress and strain in Fig. 4.10 successfully validate the present probabilistic deviation modeling approach via manufacturing modes. Consequently, this approach may be efficiently used for improved statistical quantification of rare manufacturing events, i.e., extreme induced stress/strain values at the distribution tail, and thus provide a better assessment of the overall manufacturing quality.

4.6 Application to Manufacturing Variability of Diverse Shank Designs

To investigate how surface deviations vary with nominal design and manufacturing process, the current section demonstrates manufacturing mode decompositions of three different shank designs, Fig. 4.11. Designs *A* and *C* correspond to first stage high-pressure turbine blades, having similar dimensions as the shank geometry studied in the previous section

(from this point on referred as design *D*). Design *B*, on the other hand, corresponds to a low-pressure turbine stage and is, therefore, slightly larger. Since no machining operations are performed on shanks, their final geometry is entirely associated with the casting process. Thus, despite potential differences in individual manufacturing settings, surface variability across different shank designs may be effectively visualized and compared by their manufacturing modes.

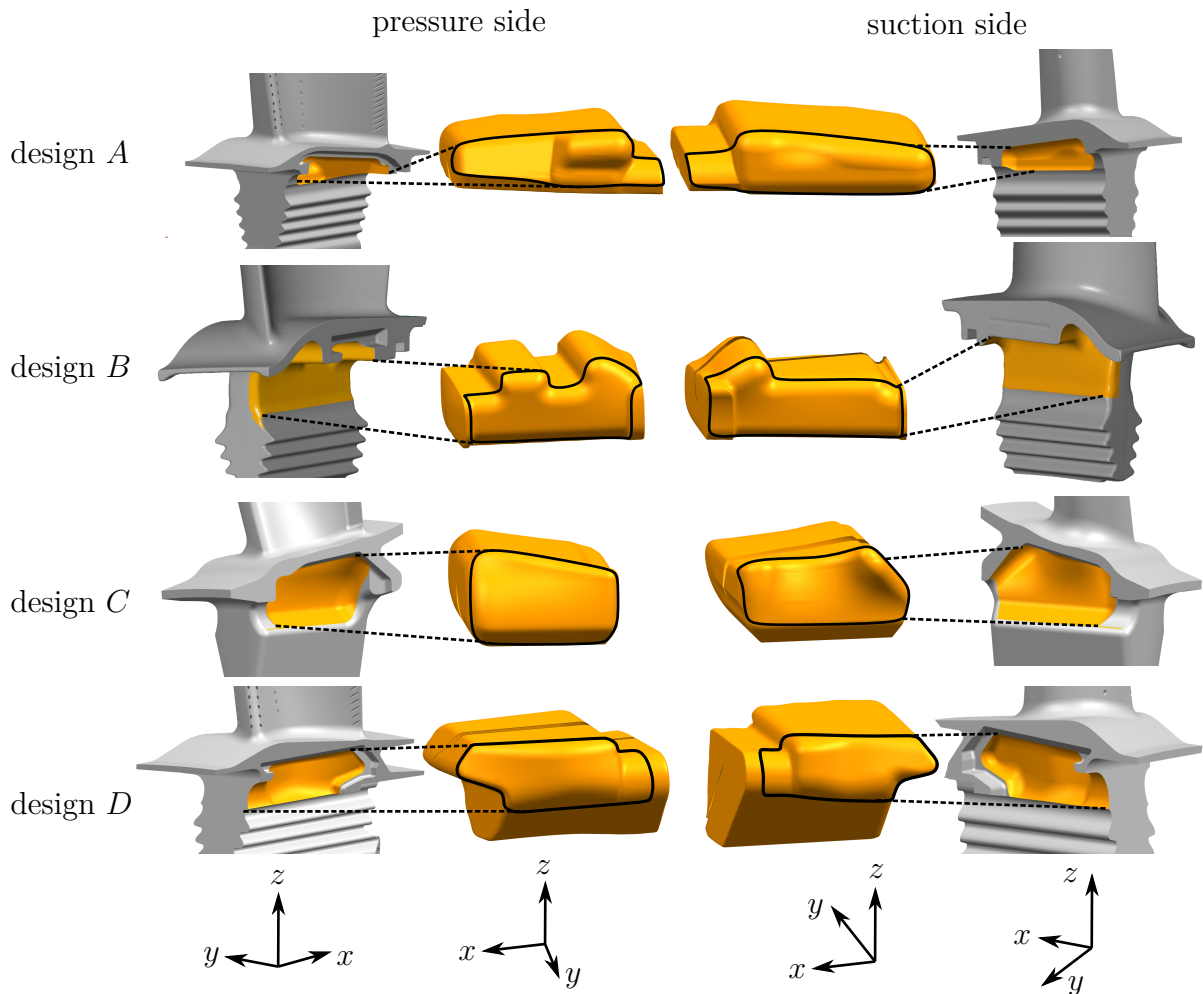


Figure 4.11: B-spline surfaces for various shank designs with associated relevant control points.

Using fine control point grids, complex surface features such as sharp blends and high curvature regions may be accurately captured with dense re-parametrized B-spline surfaces, Fig. 4.11. To fit them to real manufactured instances, surface measurements are obtained, aligned and segmented for each design. The scan alignment, just as for previously analyzed design *D*, is based on the pressure faces of the turbine blade firtree, such that surface deviations correspond to the state “as-mounted”. B-spline morphing of re-parametrized

surfaces and subsequent singular-value decomposition yield coupled manufacturing modes for pressure and suction shank walls with representation capacity illustrated as global reconstruction errors in Fig. 4.12. This reconstruction error is computed similar to the previous section, i.e., using a percentile $\rho = 99$ for error quantification within relevant normalized pairs $(u, v) \in \mathcal{U}$ and a percentile $\rho' = 90$ for quantification within the design-specific scan sample. To compare manufacturing deviation complexity, it is essential to consider the highly different number of available surface measurements per shank design ($|\mathcal{S}^A| = 204, |\mathcal{S}^B| = 38, |\mathcal{S}^C| = 74, |\mathcal{S}^D| = 56$). Surfaces of high variability are identified as those requiring large number of truncation modes for accurate reconstruction below measurement and morphing uncertainties. From Fig. 4.12 it is apparent that complexity of manufacturing variability differs tremendously across different designs: The ratio of highest number of observable modes per design to total number of manufacturing modes results as $41/204 \approx 20\%$ for design *A*, $24/38 \approx 63\%$ for design *B*, $48/74 \approx 65\%$ for design *C* and $19/56 \approx 34\%$ for design *D*, which implies that design *C* has the highest number of deviation patterns and design *A* the lowest.

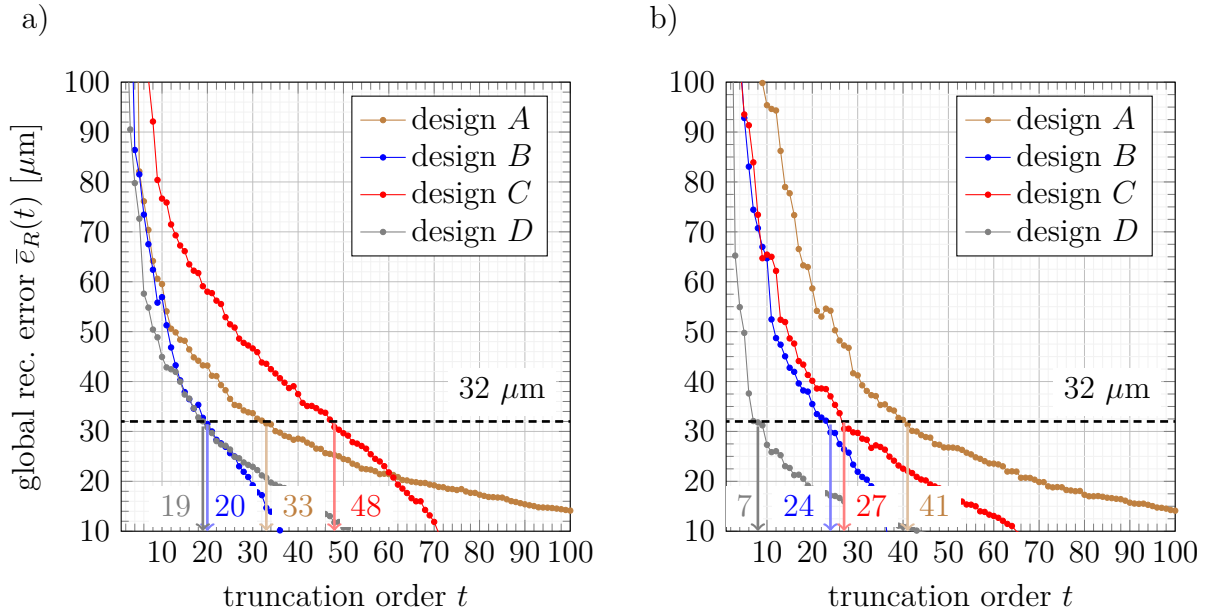


Figure 4.12: Global reconstruction error $\bar{e}_R(i)$ for a) pressure and b) suction sides of different shank designs.

A visual interpretation of static deviation component and first three manufacturing modes is presented for each design in Fig. 4.13. Two deviation limits are plotted for each manufacturing mode: the first one scales the manufacturing mode by the 5th percentile of its associated mode amplitude and the second using the 95th percentile. Additionally, the distance between manufacturing mode shape and re-parametrized surface is shown

as color scale, where blue indicates deformation directed outwards and red inwards the nominal surface. For better visualization, all images are scaled specifically for each design and mode order; therefore, only qualitative comparisons are possible. Although the mean deviation component appears to be unique for each design, Fig. 4.13 reveals that dominant manufacturing modes tend to be rigid body motions, such as y - and z -displacements, particularly in the case of designs A , B and D . Because all scans are aligned to the turbine blade firtree, which is machined after the casting body has been produced, these dominant deviations might be explained as datum offsets between casting and machining operations. Design C , however, tends to display more independent behaviors between pressure and suction shanks, which may rather suggest the prevalence of casting-related deviations. It is notable, nonetheless, that no absolute correspondence between mode order across different designs exists, e.g. z -displacement manifests as 2nd order in designs A and C , 1st order in design B and is non-existent in design D . This result suggests that deviation estimation by direct mode transfer across designs is a too general simplification of real deviation behavior.

As Fig. 4.13 demonstrates, graphical analysis of dominant manufacturing modes enables quick identification of potential fabrication issues within any particular design. Despite some modes may appear rigid and coupled between pressure and suction sides, the majority of them may still exhibit rather complex deformations, which emphasizes unavoidable geometric variability due to the casting process. Although these higher modes are indispensable for accurate CAD reconstruction, great value may be still gained by automating mode classification, at least for modes where a logical, interpretable behavior may be recognized. This process has been previously pursued by Martin and Bestle (2018), and may further facilitate identification and tracing of deviation sources within the manufacturing process in the future.

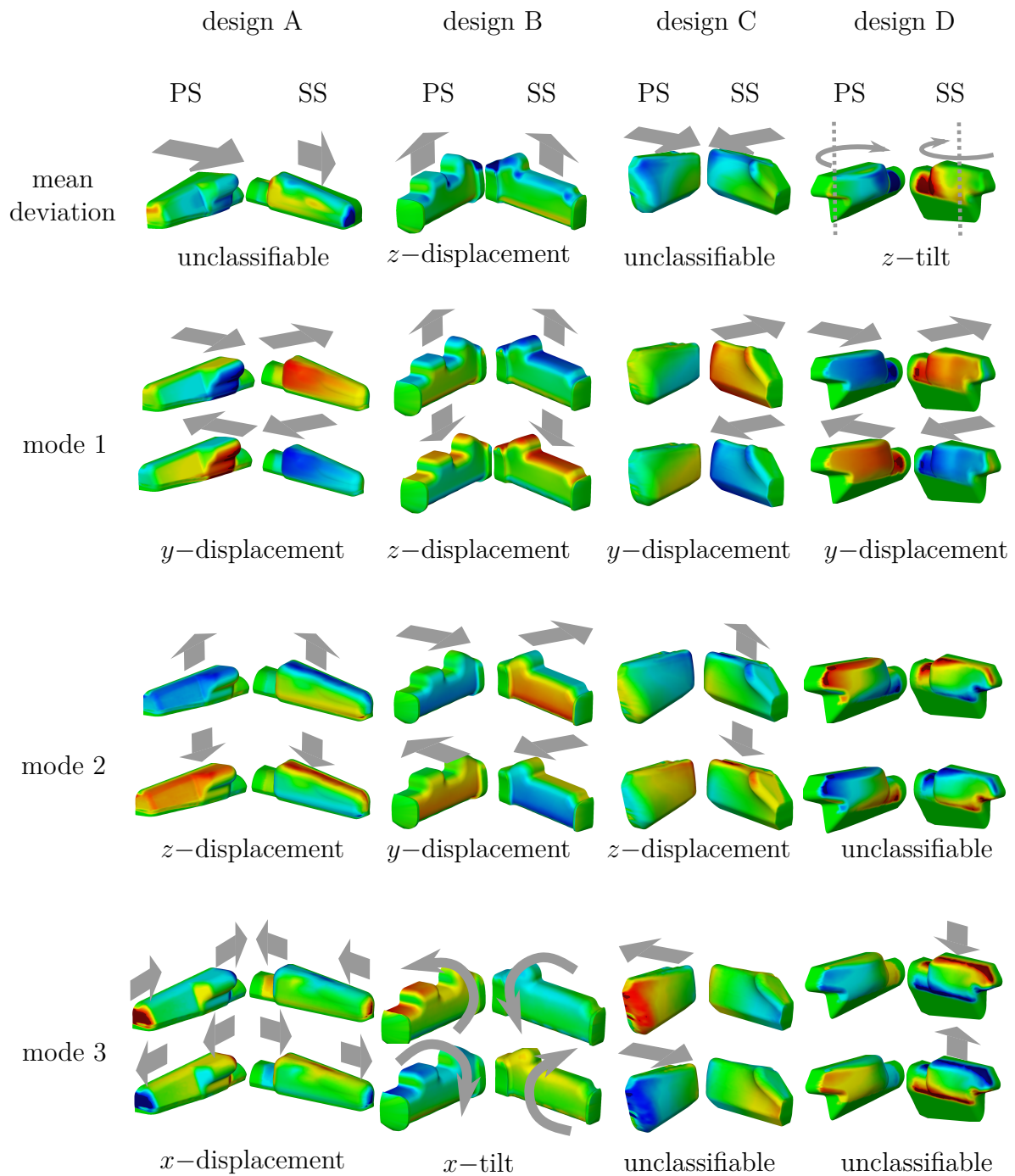


Figure 4.13: Mean deviation and manufacturing mode comparison for different shank designs.

5 Deviation Estimation for New Geometric Designs

The strategies presented in Chapter 4 require several manufactured instances of a nominal B-spline surface to set up a probabilistic representation of manufacturing variability. This is evidently possible once the geometric design has been manufactured. In the case of a robust design process, however, it is desired to estimate manufacturing deviations for a surface *before* it has been manufactured, thus limiting the strategies presented in the previous chapter. Furthermore, because any surface of interest may undergo severe geometric changes for different design versions, their associated surface parametrizations are not necessarily topologically equivalent, and thus their manufacturing modes are not transferable, see e.g., the different shank walls in Fig. 4.11 and their associated manufacturing modes in Fig. 4.13. Thus, the required deviation estimation procedure must be able to compute smooth displacement fields for arbitrary B-spline surfaces and—due to the randomness inherent to any manufacturing process—deliver a probabilistic prediction of manufacturing variability, such as manufacturing modes and amplitudes.

Given the considerable geometric and probabilistic requirements of the deviation estimation problem, the present chapter describes a novel strategy based on local relationships between surface deviation and nominal geometric properties. Rather than proposing a definitive solution to the problem, the following strategies aim to provide a robust framework coherent with the technical requirements of geometric and probabilistic surface variability representations, while at the same time exposing governing assumptions and limitations. In this manner, Sec. 5.1 presents a first intuition behind the concept of *deviation law*, i.e., a deviation-geometry relationship observed in multiple manufactured designs. This concept is formally elaborated for B-spline surfaces using the convolution operation in Sec. 5.2. Next, the process of extracting deviation laws from different manufactured designs is introduced in Sec. 5.3 as a *learning task*, which explains why only recurrent surface deviations may be successfully transferred to new designs. Once deviation estimates are available for new geometries, a probabilistic error quantification is necessary to assess prediction quality. This is achieved by defining a probabilistic divergence metric in Sec. 5.4. Then, Sec. 5.5 offers a comprehensive example of deviation prediction for shank variability

using different manufactured designs. Finally, Sec. 5.6 presents how deviation laws may be integrated in a robust design process.

5.1 Strategies for Deviation Estimation

The first consideration of deviation modeling concerns the nature of information used by the approach. Since surface deviations result from complex stochastic interaction of physical processes, an obvious approach consists of a physics-informed deviation model, which associates physical properties of the manufacturing process in question, such as casting temperatures, cooling rates or cutting-tool wear, with observed surface deviations. Despite the attractiveness of a causality relationship grounded on physical principles, this approach may be limited by the extensive knowledge required on manufacturing process parameters, some of which may be only available to the product supplier.

A likely less information-demanding approach arises from the insight that surface deviations may, to some extent, be influenced by nominal geometric properties, for example high curvature regions. If such a relationship is demonstrated repeatedly in multiple design versions, it is then likely that it may also be responsible for surface deviations of new designs. Thus, an alternative deviation model may be developed by identifying valid associations between nominal geometry with surface deviations and using them to produce predictions on new designs. Because a model of this kind does not require manufacturing inputs, the information requirements are dramatically reduced. As a trade-off, since not all surface deviations may successfully be associated to geometric properties, the obtained predictions would represent only a fraction of the overall surface variability and thus lead to a reduced, but still useful, geometric scatter. Due to the availability of several nominal HPT blade geometries with corresponding surface measurements at the time of the present investigation, this approach was chosen for further development.

The remaining of this section is devoted to develop a first intuition of the deviation estimation approach, which will be fully addressed in the subsequent sections. Consider an arbitrary CAD face which is present on different geometric designs of the same mechanical part, for instance the hub endwall of several hypothetical HPT blade designs, Fig. 5.1. In particular, assume that designs A , B and C have already been manufactured and investigated according to Chapters 3 and 4, such that probabilistic descriptions of surface variability for each of the designs are available. The ultimate goal of a deviation estimation procedure is to provide a probabilistic representation of potential surface deviations for a new, not-yet-manufactured design N .

As Fig 5.1 suggests, there are two paths to obtain this. The first path—shown as blue and pink arrows—consists of deviation estimation based on process parameters, i.e., on

constructing functional relationships for mean deviation vector $\bar{\mathbf{q}}^D \approx \tilde{\bar{\mathbf{q}}}^D(\mathbf{p}_{i,j}^D)$, observable manufacturing modes $\mathbf{U}_{t^*}^D \approx \tilde{\mathbf{U}}_{t^*}^D(\mathbf{p}_{i,j}^D)$, mean mode amplitude vector $\boldsymbol{\mu}^D \approx \tilde{\boldsymbol{\mu}}^D(\mathbf{p}_{i,j}^D)$ and mode covariance matrix $\mathbf{C}^D \approx \tilde{\mathbf{C}}^D(\mathbf{p}_{i,j}^D)$ depending on their associated nominal control point grid $\mathbf{p}_{i,j}^D$ for any manufactured design $D \in \{A, B, C\}$. Once such relationships have been modeled, they might be used to generate deviation predictions for a new design N , for example as $\bar{\mathbf{q}}^N = \tilde{\bar{\mathbf{q}}}^D(\mathbf{p}_{i,j}^N)$. This approach would offer the advantage of a direct probabilistic deviation representation for design N at the expense of three potential obstacles. First, the enormous mismatch between process parameter size (for example $\bar{\mathbf{q}} \in \mathbb{R}^{291 \times 651}$, Sec. 3.6) vs. training data (3 nominal geometries only); second, a different number of observable manufacturing modes between designs leading to different sizes for parameters \mathbf{U}_{t^*} , $\boldsymbol{\mu}$ and \mathbf{C} ; and third, the orthogonality requirement of manufacturing modes \mathbf{U}_{t^*} .

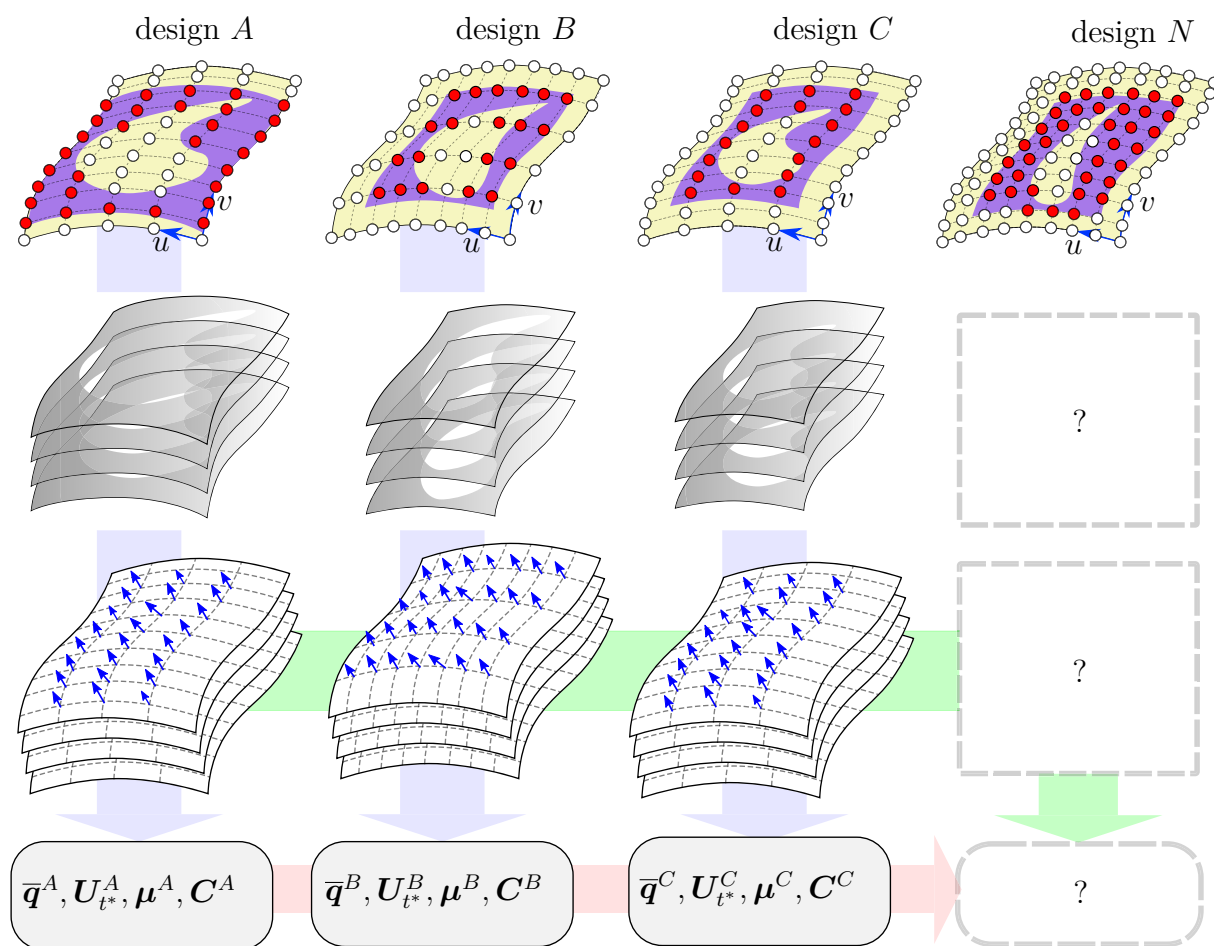


Figure 5.1: Calculation of deviation parameters according to surface measurements (blue path) and deviation estimation for a new design N based on deviation parameters (pink path) vs. based on process realizations (green path).

Instead of enforcing a standard number of observable modes and actively controlling mode orthogonality in the prediction, an alternative modeling approach consists of deviation

estimation based on manufacturing instances, see green path in Fig. 5.1. The main idea here is to establish functional relationships between deterministic control point deviations $\Delta \mathbf{p}_{i,j}^D \approx \Delta \tilde{\mathbf{p}}_{i,j}^D(\mathbf{p}_{i,j}^D)$ with corresponding nominal geometries $\mathbf{p}_{i,j}^D$ for all manufactured designs $D \in \{A, B, C\}$. In other words, the objective is to construct an artificial set of morphed surfaces for design N from observed geometric relationships between control point deviations and nominal geometry, such that all stochastic deviation parameters may be naturally derived from singular-value decomposition of the artificial surface set. These geometric relationships would have to be extracted from a large sample of morphed surfaces for all manufactured designs. However, because probabilistic description of surface variability is already available for them, realistic generation of synthetic morphed surfaces is possible without any significant computational cost.

The extraction of these geometry-deviation relationships is further illustrated in Fig. 5.2. The process is iterative in nature and starts with sampling one synthetic morphed surface from each manufactured design (three in this example). Then it is tested if it is possible to construct a functional relationship between control point deviations $\Delta \mathbf{p}_{i,j}^D$ and nominal geometry $\mathbf{p}_{i,j}^D$, for example by finding coefficients $\boldsymbol{\theta}$ of approximation functions such that

$$\Delta \mathbf{p}_{i,j}^D \approx \Delta \tilde{\mathbf{p}}_{i,j}^D(\mathbf{p}_{i,j}^D; \boldsymbol{\theta}) := \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \\ \theta_7 & \theta_8 & \theta_9 \end{bmatrix} \mathbf{p}_{i,j}^D \quad \forall D \in \{A, B, C\} \quad (5.1)$$

for each morphed surface on the triplet. If such a relationship can be demonstrated, then Eq. (5.1) may be treated as *deviation law* and the coefficients $\boldsymbol{\theta}$ may be stored in a database. Because any deviation law (5.1) is deterministic, it will always provide the same deviation prediction for any given nominal geometry. To reflect the intrinsic process randomness, a new synthetic morphed surface triplet may be sampled and tested for geometric similarities until several deviation laws with associated coefficients $\boldsymbol{\theta}^{(1)} \dots \boldsymbol{\theta}^{(L)}$ have been identified. It is important to emphasize that the functional relationship (5.1) is fixed for all deviation laws; the only variation regards the unique coefficients $\boldsymbol{\theta}$ for each surface triplet. After collection of multiple coefficients $\boldsymbol{\theta}$, they may then be used to generate L control point predictions on the new design as $\Delta \mathbf{p}_{i,j}^{N;(l)} = \Delta \tilde{\mathbf{p}}_{i,j}^D(\mathbf{p}_{i,j}^N; \boldsymbol{\theta}^{(l)})$ for $l \in [1, L] \subset \mathbb{N}$, which may then be processed via singular-value decomposition to generate the desired probabilistic estimation of manufacturing variability.

Because recurrent geometry-deviation relationships observed across all manufactured designs have the highest likelihood to manifest on new designs, it is expected that only a rather small fraction of morphed surface triplets may be effectively modeled as (5.1) and therefore yield deviation laws. This approach is, nonetheless, still preferred as modeling each stochastic process parameter as a function, whereby the impact of not finding a valid

relationship may entirely inhibit overall deviation estimation. Additionally, the sampling performed in this second approach certainly allows a different number of observable modes and, because the manufacturing modes are later computed from singular-value decomposition, no orthogonality control is required. For real deviation estimation purposes, a deviation model such as (5.1) is evidently simplified, thus a more appropriate functional relationship will be formally introduced in Sec. 5.2. Similarly, a sampling procedure, where one morphed surface triplet is tested at a time as shown in Fig. 5.2, is highly inefficient. Therefore, a more realistic procedure will be described in Sec. 5.3.

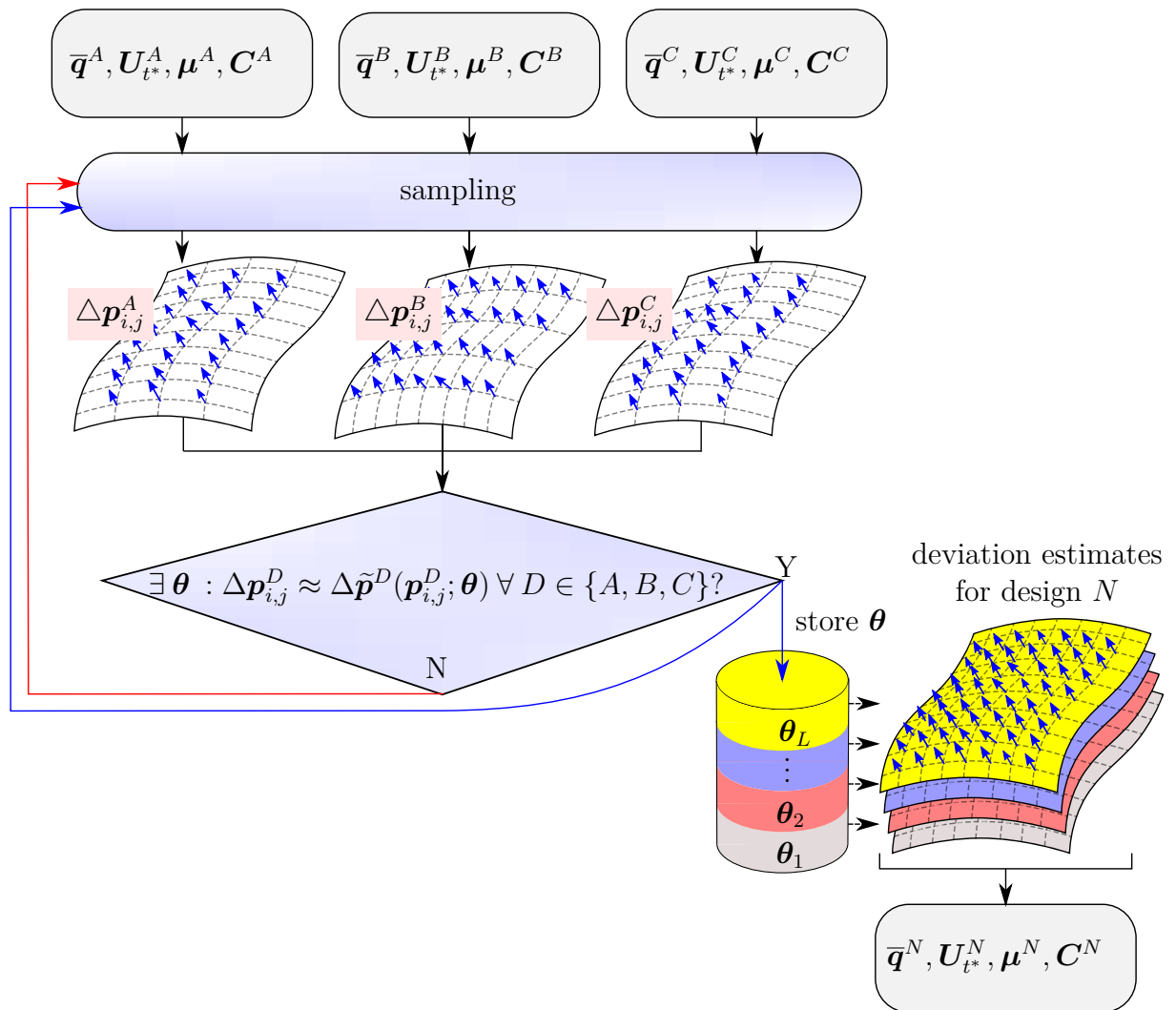


Figure 5.2: Extraction of deviation laws from manufacturing realizations with similar deviation behavior.

5.2 Deviation Model Using Convolution Operators

The extraction of geometry-deviation relationships from multiple designs poses the challenge of handling different surface parametrizations. As opposed to airfoils which suitably enable deviation transfer as parametric increments, other geometric features may exhibit wider topological changes which hinder any standardized parametrization, see different shank geometries in Fig. 4.11. To allow differing control point structures in a generic deviation model, manufacturing variability may be treated as a local phenomenon, where control point deviations $\Delta \mathbf{p}_{i,j}$ observed at any arbitrary control point position (i, j) may be associated with nominal geometric properties evaluated at the surrounding control point positions. In this manner, a local deviation model

$$\Delta \mathbf{p}_{i,j} \approx \Delta \tilde{\mathbf{p}}_{i,j}(\mathbf{p}_{k,l}, \kappa_{k,l}, \mathbf{n}_{k,l}), \quad (k, l) \in [i-h, i+h] \times [j-h, j+h] \subset \mathbb{N}^2, \quad (5.2)$$

is proposed, where approximated surface deviations $\Delta \tilde{\mathbf{p}}_{i,j}$ result from local control point positions $\mathbf{p}_{k,l}$, curvatures $\kappa_{k,l}$ and normal vectors $\mathbf{n}_{k,l}$ obtained in the surrounding $(2h+1) \times (2h+1)$ positions of (i, j) w.r.t. the nominal control point grid.

To enforce a comparable participation of control point deviation, position, curvature and orientation in the model, a normalization procedure is required. Since synthetic morphed surfaces from different geometric designs will serve as input for the model, it is additionally required that relative parametric differences between designs remain unaffected by this procedure. Thus, the desired normalization coefficients are not design-specific, but must be quantified using all designs. Consider, e.g., the deviation transfer problem illustrated in Fig. 5.1, where morphed surfaces from manufactured designs A, B and C are used to predict morphed surfaces for a new design N . To normalize control point deviations of any morphed surface, a reference deviation

$$\Delta p_\infty := \left\{ \Delta p_{i,j}^{D,k,(s)}; D \in \{A, B, C\}, k \in \{x, y, z\}, s \in [1, S] \subset \mathbb{N} \right\}_{90} \quad (5.3)$$

may be defined and calculated as follows: first a total of S synthetic morphed surfaces from each manufactured design $D \in \{A, B, C\}$ is constructed, e.g. four morphed surfaces per design in Fig. 5.3. Then, all Cartesian deviation components are extracted from all $3S$ surfaces and gathered in a single, large deviation set. Finally, the 90th percentile is extracted from all deviation values and used as reference deviation Δp_∞ . For normalization of control-point deviations, a large percentile is preferred over maximum deviation values because it avoids excessive distortion provoked by eventual deviation spikes. It is also important that the normalization procedure preserves relative differences between Cartesian deviations, which is why this percentile is taken combining all Cartesian directions. Additionally,

since division of control point deviations by (5.3) may yield values greater than one, the hyperbolic tangent function is applied leading to normalized control point deviations

$$\Delta \hat{p}_{i,j}^k := \tanh \left(\frac{\Delta p_{i,j}^k}{\Delta p_\infty} \right), \quad -1 < \Delta \hat{p}_{i,j}^k < 1, \quad k \in \{x, y, z\}. \quad (5.4)$$

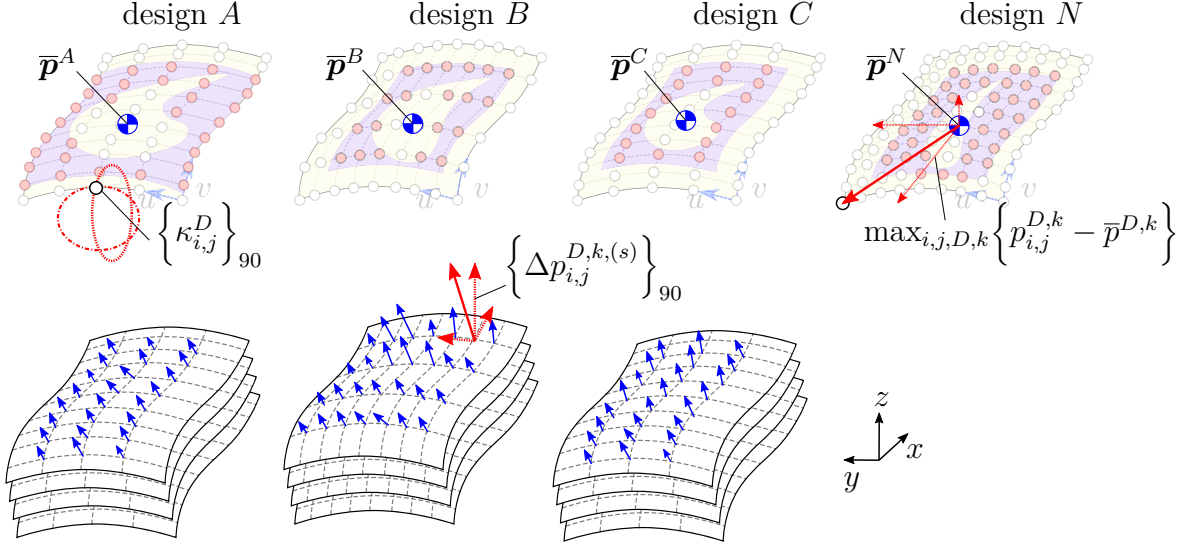


Figure 5.3: Normalization coefficients for deviation modeling.

The remaining normalization rules concern nominal geometric properties associated with each manufactured and new design, i.e., they are computed once for each re-parametrized nominal B-spline surface (not morphed surfaces). To normalize control point positions, it is first useful to define a design-specific nominal centroid

$$\bar{\mathbf{p}}^D := \frac{1}{|\mathcal{I}^D|} \sum_{(i,j) \in \mathcal{I}^D} \mathbf{p}_{i,j}^D, \quad k \in \{x, y, z\}, \quad (5.5)$$

computed as mean value of relevant control points of any design D , see the four blue colored circles in Fig. 5.3. A reference control point value p_∞ may be then defined as largest difference of any Cartesian control point component (relevant or irrelevant) from its corresponding nominal centroid as

$$p_\infty := 1.10 \times \max_{i,j,D,k} \left\{ p_{i,j}^{D,k} - \bar{p}^{D,k} \right\}, \quad D \in \{A, B, C, N\}, \quad k \in \{x, y, z\}, \quad (5.6)$$

where the quantification is performed by considering all manufactured designs as well as the new design. For example, in Fig. 5.3 the largest distance from any control point to its respective centroid is found in the new design N , where the x -component is largest. If the new design is derived from an optimization process, the final surface dimensions may be

still unknown. Given that the reference control point value (5.6) represents the maximum dimension of any possible design surface, an additional factor 1.10 is introduced to consider cases where larger re-parametrized B-spline surfaces are designed. Now, normalized control point positions $\hat{\mathbf{p}}_{i,j}$ may be defined as

$$\hat{\mathbf{p}}_{i,j} := \frac{\mathbf{p}_{i,j} - \bar{\mathbf{p}}}{p_\infty}, \quad -1 < \hat{p}_{i,j}^k < 1, \quad k \in \{x, y, z\}. \quad (5.7)$$

The Gauß curvature $\kappa_{i,j}$ defined for control point positions by Eq. (2.20) serves as a good starting point for curvature quantification. However, to prevent normalization distortion by extreme curvature values due to unexpected surface irregularities, a reference value κ_∞ may be taken as the 90th percentile of Gauß curvatures of all re-parametrized B-spline surfaces, i.e.,

$$\kappa_\infty := \left\{ \kappa_{i,j}^D; D \in \{A, B, C, N\} \right\}_{90}, \quad (5.8)$$

where all manufactured designs and the new design are considered. Since curvature $\kappa_{i,j}$ attains only positive values, a zero-centered formulation would be more consistent with previous normalized quantities. Additionally, since it has been observed that the vast majority of curvature values from present examples lies below $\kappa_{i,j} < 0.5$, the fourth root is taken to enhance curvature resolution on these regions. This leads to the definition of normalized curvature as

$$\hat{\kappa}_{i,j} := \tanh \left(\frac{2(\kappa_{i,j})^{1/4} - (\kappa_\infty)^{1/4}}{(\kappa_\infty)^{1/4}} \right), \quad -1 < \hat{\kappa}_{i,j} < 1, \quad (5.9)$$

where the hyperbolic tangent function has been again used to enforce the desired normalization limits. Lastly, the orientation of at any control point position may be obtained by its normal vector $\mathbf{n}_{i,j}$ approximated by finite differences according to Eq. (2.17), which is already consistently normalized.

The geometry-deviation relationship may now be modeled using a convolution operation. Consider, e.g., nominal design A and an associated synthetic morphed surface represented by normalized control point deviations $\Delta\hat{\mathbf{p}}_{i,j}$. The first step consists of summarizing all nominal geometric properties of design A , i.e., normalized positions $\hat{\mathbf{p}}_{i,j}$, orientations $\mathbf{n}_{i,j}$ and curvatures $\hat{\kappa}_{i,j}$ by a characteristic tensor $\mathbf{C} \in \mathbb{R}^{(n+1) \times (m+1) \times 7}$ where

$$\begin{aligned} \mathbf{C}_{i,j,0} &= \hat{p}_{i,j}^x, & \mathbf{C}_{i,j,1} &= \hat{p}_{i,j}^y, & \mathbf{C}_{i,j,2} &= \hat{p}_{i,j}^z, \\ \mathbf{C}_{i,j,3} &= n_{i,j}^x, & \mathbf{C}_{i,j,4} &= n_{i,j}^y, & \mathbf{C}_{i,j,5} &= n_{i,j}^z, & \mathbf{C}_{i,j,6} &= \hat{\kappa}_{i,j}, \end{aligned} \quad (5.10)$$

and the associated normalized control point deviations $\Delta\hat{\mathbf{p}}_{i,j}$ are summarized as a measured

deviation tensor $\hat{\mathbf{P}} \in \mathbb{R}^{(n+1) \times (m+1) \times 3}$ where

$$\hat{\mathbf{P}}_{i,j,0} = \Delta \hat{p}_{i,j}^x, \quad \hat{\mathbf{P}}_{i,j,1} = \Delta \hat{p}_{i,j}^y, \quad \hat{\mathbf{P}}_{i,j,2} = \Delta \hat{p}_{i,j}^z. \quad (5.11)$$

This tensor construction is illustrated in Fig. 5.4a for a nominal control point geometry and the corresponding control point deviations.

Next, the association between characteristic tensor \mathbf{C} and measured deviation tensor $\hat{\mathbf{P}}$ may be performed by means of a convolution operation which is calculated independently for each Cartesian coordinate by applying a three-dimensional kernel $\mathbf{K} \in \mathbb{R}^{(2h+1) \times (2h+1) \times 7}$ to the characteristic tensor \mathbf{C} . For instance, convolution along the x -coordinate at position (i, j) yields the approximated normalized deviation

$$\Delta \tilde{p}_{i,j}^x = (\mathbf{C} * \mathbf{K}^x)_{i,j} \equiv \sum_{k=1}^7 \sum_{\mu=-h}^h \sum_{\nu=-h}^h c_{i+\mu, j+\nu, k} \mathbf{K}_{\mu+h+1, \nu+h+1, k}^x, \quad (5.12)$$

which computes the sum-product of all overlapping elements at each (i, j) position, Fig. 5.4b. Similar operations apply to y - and z -directions using different kernels \mathbf{K}^y and \mathbf{K}^z . The resulting approximated surface deviations $\Delta \tilde{p}_{i,j}$ may then be represented by an approximated deviation tensor $\tilde{\mathbf{P}} \in \mathbb{R}^{(n+1) \times (m+1) \times 3}$ where

$$\tilde{\mathbf{P}}_{i,j,0} = \Delta \tilde{p}_{i,j}^x, \quad \tilde{\mathbf{P}}_{i,j,1} = \Delta \tilde{p}_{i,j}^y, \quad \tilde{\mathbf{P}}_{i,j,2} = \Delta \tilde{p}_{i,j}^z. \quad (5.13)$$

The kernels $\mathbf{K}^x, \mathbf{K}^y, \mathbf{K}^z$ need to be trained to minimize the differences between measured and approximated deviation tensors $\hat{\mathbf{P}}$ and $\tilde{\mathbf{P}}$, Fig. 5.4c. This minimization of model errors requires the definition of a loss function. Since control point displacements are Cartesian vectors, the most reasonable deviation criterion consists of squared differences along each dimension, i.e., $(\Delta \tilde{p}_{i,j}^k - \Delta \hat{p}_{i,j}^k)^2$ for $k \in \{x, y, z\}$. This difference alone, however, is not informative unless it is put into relation with the magnitude of targeted control point displacements. In other words, larger differences may be expected by synthetic morphed surfaces with larger measured control point deviations. Thus, the difference may be normalized by the measured control point deviation variance $\sum_{i=0}^n \sum_{j=0}^m \sum_{k \in \{x, y, z\}} (\Delta \hat{p}_{i,j}^k - \Delta \bar{p}^k)^2$ which is exclusively associated with the morphed surface in question and uses the mean control point deviation vector

$$\Delta \bar{\mathbf{p}} := \frac{1}{|\mathcal{I}|} \sum_{(i,j) \in \mathcal{I}} \Delta \hat{\mathbf{p}}_{i,j}, \quad (5.14)$$

which is also computed for each morphed surface. Now, a normalized loss function may be

defined as

$$e(\tilde{\mathbf{P}}, \hat{\mathbf{P}}) = \frac{\sum_{i=0}^n \sum_{j=0}^m \sum_{k \in \{x,y,z\}} (\Delta \tilde{p}_{i,j}^k - \Delta \hat{p}_{i,j}^k)^2}{\sum_{i=0}^n \sum_{j=0}^m \sum_{k \in \{x,y,z\}} (\Delta \hat{p}_{i,j}^k - \Delta \bar{p}^k)^2}, \quad (5.15)$$

which may be effectively interpreted regardless of geometric design or manufacturing instance.

In this manner, kernels \mathbf{K}^x , \mathbf{K}^y and \mathbf{K}^z are adapted to minimize error (5.15) below a certain threshold, consequently leading to an accurate relationship between nominal geometric properties of any design and a specific morphed surface by means of the convolution operation (5.12). In practice, however, a single convolution operation may be insufficient to accurately model control point deviations. This is why the next section extends the present model and introduces a deviation learning technique implementing convolutional neural networks.

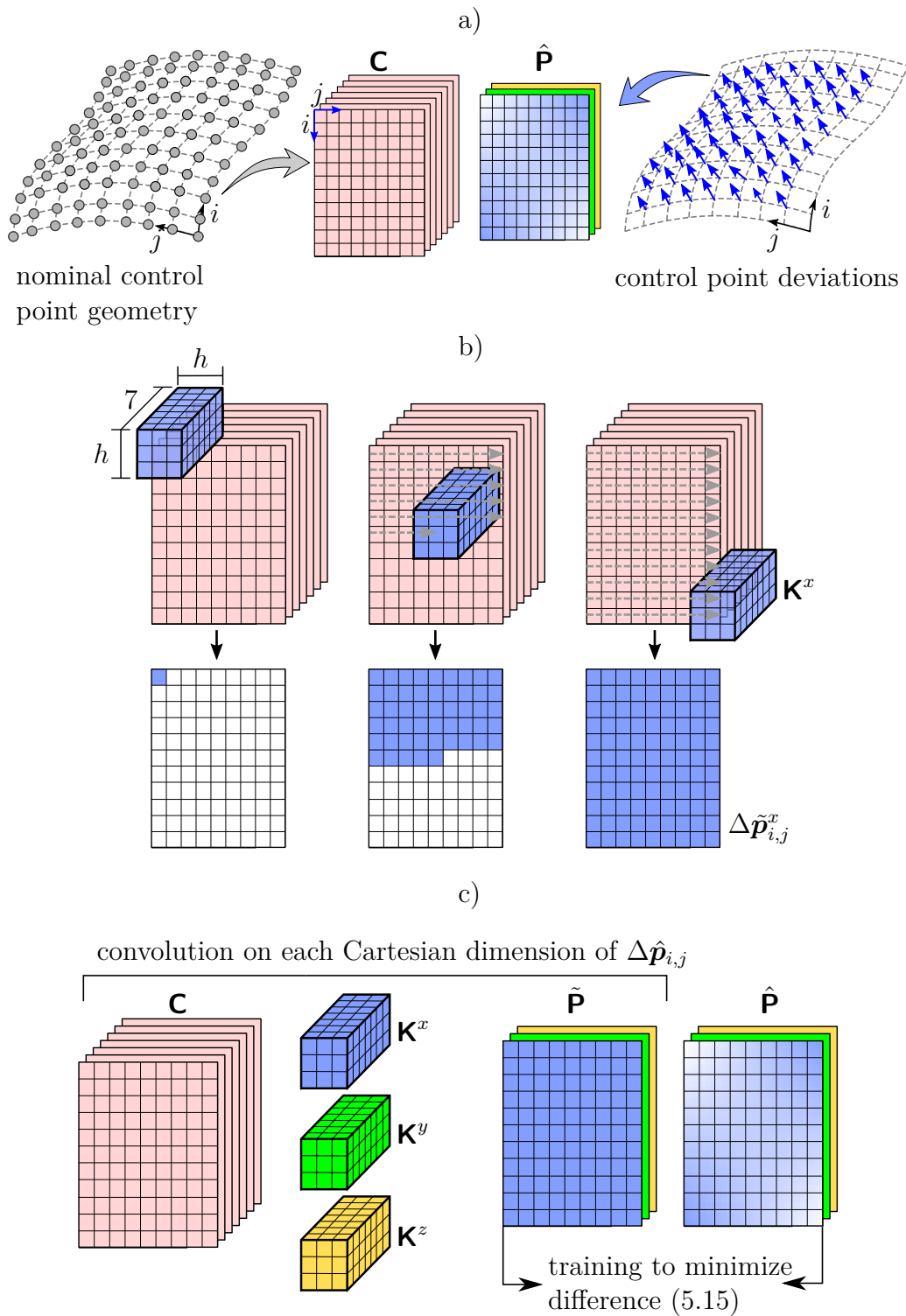


Figure 5.4: Deviation model using convolution operation: a) construction of characteristic and deviation tensors, b) convolution operation, and c) kernel training on the three Cartesian dimensions.

5.3 Learning of Deviation Laws

The process of extracting and transferring deviation information from manufactured designs into new designs requires a manufacturing deviation model as derived in the previous section and an optimal deviation learning strategy delivering the best possible interpretation of surface deviations for new designs. To achieve the latter, it is first necessary to distinguish between model regression and learning strategies. Although both approaches are used to fit over-determined systems of equations to a set of observations, each follows a different paradigm. Deterministic model regression, for instance, treats observations as the totality of possible outcomes of any given process and, therefore, aims at producing a rigid representation of the past. A learning strategy, on the other side, aims at building a process representation which will remain truthful after new observations.

An example for model regression is B-spline morphing, where distances between measurement and surface points are treated as every possible manifestation of manufacturing variability of a particular surface measurement and, therefore, lead to unique control point deviations obtained by linear regression (3.22). On the other hand, computation of deviation laws necessitates a learning strategy because geometry-deviation relationships must be formulated such that they are also valid for new, not yet manufactured designs. In this sense, computation of kernels \mathbf{K}^x , \mathbf{K}^y and \mathbf{K}^z by *only* minimizing loss function (5.15) with respect to a unique design is wrong, because it entirely ignores how accurate the kernels may reproduce deviations on any other nominal geometry. Therefore, to assess the ability of a model to perform predictions on data not seen during training, i.e., its generalization capacity, a learning strategy typically splits available observations into training and test datasets. The training observations (roughly 70 – 80% of all available observations) are used to incrementally train the model, whereas the remaining test observations are used to progressively assess how the model will react on new data.

To illustrate these concepts, consider the following example where a probabilistic process

$$y(x; \mathbf{z}) := p(x) + \mathbf{z}, \quad p(x) = 1 - x + 9x^2 - 3x^3, \quad \mathbf{z} \sim N(0, 10) \quad (5.16)$$

is constructed by adding Gaussian noise \mathbf{z} to a third-degree polynomial $p(x)$. A deterministic approximation model

$$\hat{y}(x; \boldsymbol{\theta}) := \sum_{n=0}^5 \theta_n x^n, \quad (5.17)$$

shall be build up by first using linear regression and then a learning strategy for adapting model parameters θ_n . Assume 30 available realizations of process (5.16) to be given and stored in a dataset

$$\mathcal{D} := \{(x, y)^{(1)}, \dots, (x, y)^{(30)}\} \quad (5.18)$$

shown as blue and red dots in Fig. 5.5a. Both methods require a definition of a loss function to fit the model, where in this case the mean squared error is chosen:

$$L(\mathcal{D}; \boldsymbol{\theta}) := \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} (y - \hat{y}(x; \boldsymbol{\theta}))^2. \quad (5.19)$$

Linear regression treats available observations as all possible process outcomes and, therefore, aims at producing a model which best fits all available information. In the present example, linear regression of polynomial (5.17) using observations (5.18) and loss function (5.19) may be solved explicitly producing coefficients $\boldsymbol{\theta}_R$ and leading to model $\hat{y}(x; \boldsymbol{\theta}_R)$ shown as black line in Fig. 5.5a.

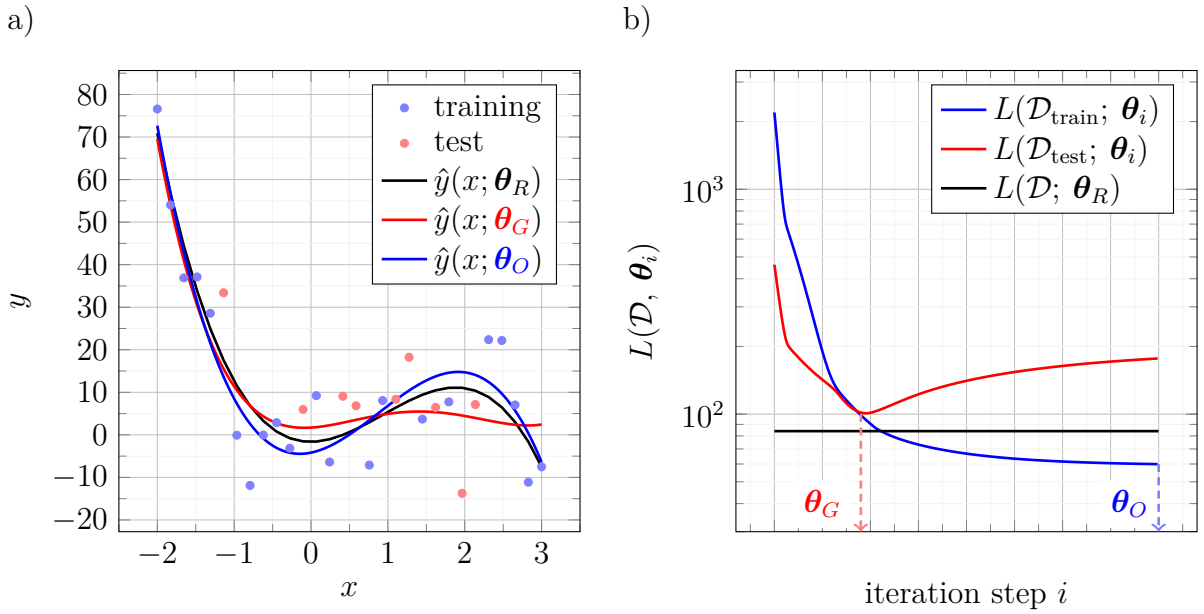


Figure 5.5: Diagrams for a) dataset approximation and b) loss function (5.19).

The learning strategy, on the other side, assumes that further process observations may be expected and aims at producing a model representation which best fits new data. Therefore, the observation set \mathcal{D} is first randomly separated into training and test datasets $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$, containing 21 and 9 observations, respectively, see blue and red dots in Fig. 5.5a. To offer continuous assessment of model accuracy, loss function (5.19) may be solved iteratively. Due to its simplicity, its gradient

$$\nabla L = \left[\frac{\partial L}{\partial \theta_0} \cdots \frac{\partial L}{\partial \theta_5} \right]^\top \quad (5.20)$$

with respect to model parameters $\boldsymbol{\theta}$ may be computed analytically as

$$\frac{\partial L}{\partial \theta_n} = -\frac{2}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} (y - \hat{y}(x; \boldsymbol{\theta})) x^n, \quad n \in [0, 5] \subset \mathbb{N}_0, \quad (5.21)$$

which enables an iterative solution by gradient descent as

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \varepsilon \nabla L(\mathcal{D}_{\text{train}}; \boldsymbol{\theta}_i), \quad (5.22)$$

where the constant learning rate ε may be chosen arbitrarily, e.g. $\varepsilon = 0.01$. It is relevant to notice that the calculation of model parameters $\boldsymbol{\theta}$ is only driven by the training dataset $\mathcal{D}_{\text{train}}$. Thus, evaluation of $L(\mathcal{D}_{\text{test}}; \boldsymbol{\theta}_i)$ remains a valid estimation of model accuracy on new observations.

The result of this learning process is shown in Fig. 5.5b. Blue and red lines represent the model error with respect to training and test datasets, whereas the black line corresponds to the model error using linear regression. It is evident that an increasing number of iterations benefits training accuracy but not necessarily generalization capacity. Here, the model test error $L(\mathcal{D}_{\text{test}}; \boldsymbol{\theta}_i)$ reaches its global minimum at $\boldsymbol{\theta}_G$ corresponding to the best performance on new observations, thus delivering optimal generalization. Continuing the training beyond this point would improve the model accuracy w.r.t training observations, but would increasingly damage performance on the test dataset (overfitting). In particular, the model parameters $\boldsymbol{\theta}_O$ would produce the best training accuracy, but almost worst generalization. This result highlights the purpose of any learning strategy: not merely concentrating on information retrieval but on identification of patterns useful for solving new problems.

The present manufacturing deviation model associates nominal geometric properties with surface deviations. The simplest architecture introduced in the previous section corresponds to a single convolution operation, where each individual kernel reproduces the relationship between characteristic tensor \mathbf{C} and approximated deviation tensor $\tilde{\mathbf{P}}$ along each Cartesian direction, e.g. Eq. (5.12) for x -direction. To enable extraction of more complex geometrical relationships, this architecture may be further extended to multiple sequential convolution operations leading to a convolutional neural network, see Appendix A.1. Therefore, approximated normalized control point deviations $\tilde{\mathbf{P}}^D$ from any particular design D may be modeled as a convolutional neural network

$$\tilde{\mathbf{P}}^D = f(\mathbf{C}^D; \boldsymbol{\theta}) \quad (5.23)$$

which transforms the design-specific characteristic tensor \mathbf{C}^D according to neural network weights $\boldsymbol{\theta}$ into the desired tensor. In this case, since the architecture of the convolutional

neural network f (e.g. number of convolution operations, kernel dimensions, etc.) remains fixed throughout the problem, notation $f(\mathbf{C}; \boldsymbol{\theta})$ will refer to the output tensor obtained by forward propagation of characteristic tensor \mathbf{C} into network f using weights $\boldsymbol{\theta}$, i.e., it implies *evaluation* of neural network f rather than functional dependency, see Appendix A.1.

In this manner, the objective consists on identifying a set of network weights $\boldsymbol{\theta}^{(1)} \dots \boldsymbol{\theta}^{(L)}$ which is capable of reproducing realistic control point deviations on all manufactured designs available for training. To reproduce the random characteristics of any manufacturing process, these deviation laws are extracted from synthetic morphed surfaces sampled from each manufactured design, Fig. 5.2. However, in contrast to this simplified scheme, where only one synthetic surface per design was investigated at each sampling iteration, the present section describes a more efficient approach. This process for extracting deviation laws will be first described for two different manufactured designs and later generalized for multiple designs.

Because extraction of deviation patterns requires at least one manufactured design to continuously test the learning process, the minimum number of available manufactured designs required is two. To increase the probability of finding synthetic morphed surfaces with similar deviation patterns, a larger number S of synthetic surfaces should be extracted from the probabilistic representation of each design A and B leading to sets

$$\mathcal{A} = \left\{ \hat{\mathbf{P}}^{A,(1)}, \dots, \hat{\mathbf{P}}^{A,(S)} \right\} \quad \text{and} \quad \mathcal{B} = \left\{ \hat{\mathbf{P}}^{B,(1)}, \dots, \hat{\mathbf{P}}^{B,(S)} \right\} \quad (5.24)$$

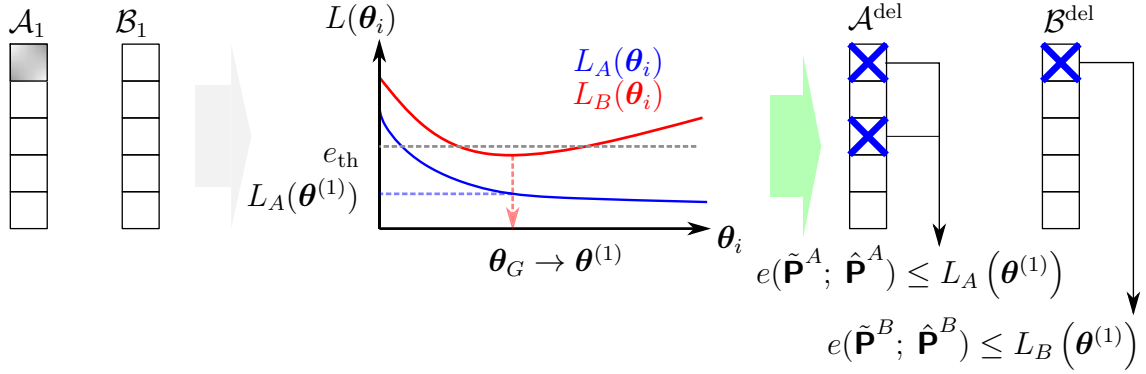
of measured deviation tensors normalized according to Eq. (5.4). The proposed method for deviation law extraction will be described in the following by referring constantly to Fig. 5.6, where sets \mathcal{A} and \mathcal{B} are shown as white columns with five instances each, respectively.

The goal is to pair deviation tensors from both designs which share unique geometry-deviation characteristics. Because several geometry-deviation relationships may be expected, this pairing process is performed iteratively such that in each pairing iteration only unpaired deviation tensors are considered (in Fig. 5.6 the pairing process is complete after three iterations, where already paired elements of sets \mathcal{A} and \mathcal{B} are represented as gray crosses at the start of each pairing iteration). Each pairing iteration j starts by associating the first unpaired deviation tensor $\hat{\mathbf{P}}^{A,(1)}$ (shown as gray square in Fig. 5.6) from the current set \mathcal{A}_j of unpaired deviation tensors with its characteristic tensor \mathbf{C}^A by means of the training loss function

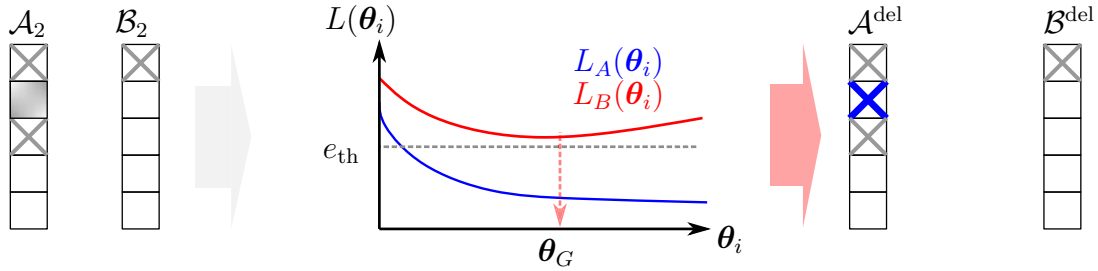
$$L_A(\boldsymbol{\theta}_i) := e \left(\tilde{\mathbf{P}}^A; \hat{\mathbf{P}}^{A,(1)} \right), \quad \tilde{\mathbf{P}}^A = f \left(\mathbf{C}^A; \boldsymbol{\theta}_i \right), \quad (5.25)$$

where $e(\tilde{\mathbf{P}}, \hat{\mathbf{P}})$ corresponds to distance (5.15) and $\tilde{\mathbf{P}}^A$ corresponds to the output of the neural network at the training iteration i with characteristic tensor \mathbf{C}^A as input.

pairing iteration $j = 1$



pairing iteration $j = 2$



pairing iteration $j = 3$

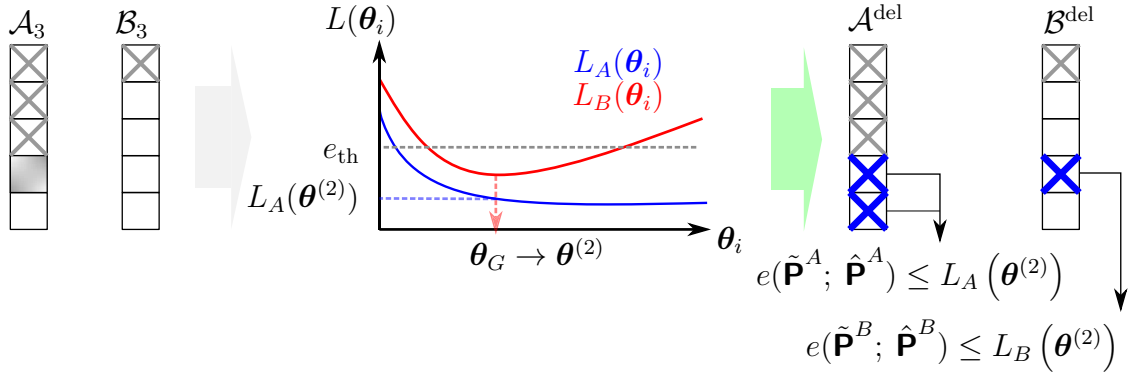


Figure 5.6: Extraction of two deviation laws from five manufacturing instances of two designs A and B .

Since this loss function alone may lead to model overfitting as explained in Fig. 5.5, a test case must be defined to assess if the neural network $f(\mathbf{C}; \theta_i)$ generalizes to new geometric designs. Therefore, a generalization error may be defined by asking how accurate

parameters $\boldsymbol{\theta}_i$ would reproduce manufacturing deviations of design B , i.e.:

$$L_B(\boldsymbol{\theta}_i) := \min_{\hat{\mathbf{P}}^B \in \mathcal{B}_j} e\left(\tilde{\mathbf{P}}^B; \hat{\mathbf{P}}^B\right), \quad \tilde{\mathbf{P}}^B = f\left(\mathbf{C}^B; \boldsymbol{\theta}_i\right), \quad (5.26)$$

where the estimated deviation tensor $\tilde{\mathbf{P}}^B$ at the current state $\boldsymbol{\theta}_i$ is compared to each measured (unpaired) deviation tensor in set \mathcal{B}_j using distance $e(\tilde{\mathbf{P}}, \hat{\mathbf{P}})$. Because distance $e(\tilde{\mathbf{P}}, \hat{\mathbf{P}})$ has been normalized according to the targeted measured deviation tensor $\hat{\mathbf{P}}$, training (5.25) and test (5.26) losses may be effectively compared, see blue and red curves in Fig. 5.6. Now, the neural network may be trained using stochastic gradient descent, where the weights are incrementally corrected as

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \varepsilon \nabla L_A(\boldsymbol{\theta}_i) \quad (5.27)$$

and gradient $\nabla L_A(\boldsymbol{\theta}_i)$ may be evaluated from the neural network using back-propagation, see Appendix A.1.

Once the training has converged, existence of a deviation law between tensor $\hat{\mathbf{P}}^{A,(1)}$ and any tensor $\hat{\mathbf{P}} \in \mathcal{B}_j$ may be tested. For this, the model parameters producing optimum generalization

$$\boldsymbol{\theta}_G := \underset{\boldsymbol{\theta}}{\operatorname{argmin}} L_B(\boldsymbol{\theta}) \quad (5.28)$$

are identified, and if at this stage both training (5.25) and test (5.26) errors fall below a certain threshold e_{th} (such as for first and third iterations in Fig. 5.6) the network weights approximate realistic manufacturing deviations on both designs and, therefore, may be accepted as first deviation law, i.e.,

$$\boldsymbol{\theta}_G =: \boldsymbol{\theta}^{(1)} \quad \text{if} \quad \max_{D \in \{A,B\}} L_D(\boldsymbol{\theta}_G) < e_{\text{th}}. \quad (5.29)$$

The training error $L_A(\boldsymbol{\theta}^{(1)})$ evaluated for the optimum generalization parameters corresponds to the distance between deviation estimation $\tilde{\mathbf{P}}^A = f(\mathbf{C}^A; \boldsymbol{\theta}^{(1)})$ and first deviation tensor $\hat{\mathbf{P}}^{A,(1)}$. Because this distance may be treated as a tolerance, any other deviation tensor in sample \mathcal{A}_j with an associated error below $L_A(\boldsymbol{\theta}^{(1)})$ may also be explained by this deviation law. Thus, before starting a new pairing iteration, all measured deviation tensors covered by the current deviation law may be extracted in a design-specific set

$$\mathcal{A}^{\text{del}} := \left\{ \hat{\mathbf{P}} \in \mathcal{A}_j \mid e\left(\tilde{\mathbf{P}}; \hat{\mathbf{P}}\right) \leq L_A\left(\boldsymbol{\theta}^{(1)}\right) \right\}, \quad \tilde{\mathbf{P}} = f\left(\mathbf{C}^A; \boldsymbol{\theta}^{(1)}\right). \quad (5.30)$$

This may also be evaluated for design B to compute \mathcal{B}^{del} (or for any other design D) by using the correspondent tensor set \mathcal{B}_j , error $L_B(\boldsymbol{\theta}^{(1)})$ and replacing $\boldsymbol{\theta}^{(1)}$ with the current

deviation law (in Fig. 5.6 all deviation tensors within tolerance found in the current pairing iteration are represented as blue crosses, those found on the previous iteration as gray crosses).

Now, deviation tensor sets (5.24) may be contracted for the next pairing iteration $j + 1$ by deleting all paired deviation tensors found at iteration j , i.e.,

$$\mathcal{A}_{j+1} := \mathcal{A}_j \setminus \mathcal{A}_j^{\text{del}} \quad \text{and} \quad \mathcal{B}_{j+1} := \mathcal{B}_j \setminus \mathcal{B}_j^{\text{del}}, \quad (5.31)$$

and the process may be repeated from Eq. (5.25). In case no deviation law is found, i.e., Eq. (5.29) does not hold, the chosen deviation tensor $\hat{\mathbf{P}}^{A, (1)}$ cannot be explained by geometric relationships and, therefore, it must be discarded from its dataset before starting a new iteration, i.e., $\mathcal{A}_{j+1} := \mathcal{A}_j \setminus \{\hat{\mathbf{P}}^{A, (1)}\}$, see iteration 2 in Fig. 5.6.

In this manner, further pairing iterations may be performed until all deviation tensors in set \mathcal{A} have been tested. It is expected that only a small fraction of deviation tensors effectively leads to deviation laws, which is why the entire process may be performed again using new synthetic morphed surface samples from designs A and B (i.e. bootstrapping from probabilistic deviation descriptions of designs A and B) until sufficient deviation laws $\boldsymbol{\theta}^{(1)} \dots \boldsymbol{\theta}^{(L)}$ have been found and stored in a database. In Fig. 5.6 for example, all deviation tensors in dataset \mathcal{A} were tested after three pairing iterations, first and third leading to deviation laws explaining four total manufacturing instances of design A and two of design B .

Extraction of deviation laws may analogously be formulated for any arbitrary number of manufactured designs. This may be desired, for instance, when extracting coupled deviation patterns from multiple casted surfaces, e.g. pressure- and suction-side shanks of diverse turbine blade designs. In this case, it is again imperative to leave at least one manufactured design for testing and the remaining designs for training. The process starts again by creating samples of normalized measured deviation tensors for all designs of interest, for instance \mathcal{A} , \mathcal{B} , \mathcal{C} and \mathcal{D} in case of four manufactured designs. For each pairing iteration j , the first deviation tensor $\hat{\mathbf{P}}^{A, (1)}$ of current tensor set \mathcal{A}_j may be used to define loss function $L_A(\boldsymbol{\theta}_i)$ according to Eq. (5.25). However, after surpassing tolerance e_{th} at a parameter setting

$$\boldsymbol{\theta}_A := \underset{\boldsymbol{\theta}}{\operatorname{argmax}} L_A(\boldsymbol{\theta}_i) \text{ s.t. } L_A(\boldsymbol{\theta}_i) < e_{\text{th}}, \quad (5.32)$$

the training process may now be extended to consider a measured deviation tensor $\hat{\mathbf{P}}^{B, *}$ from design B which best matches the deviation prediction at this stage, i.e.,

$$\hat{\mathbf{P}}^{B, *} := \underset{\hat{\mathbf{P}}^B \in \mathcal{B}_j}{\operatorname{argmin}} e\left(\tilde{\mathbf{P}}^B; \hat{\mathbf{P}}^B\right), \quad \tilde{\mathbf{P}}^B = f\left(\mathbf{C}^B; \boldsymbol{\theta}_A\right). \quad (5.33)$$

Deviation tensor $\hat{\mathbf{P}}^{B,*}$ constitutes, therefore, a potential candidate for a manufacturing instance of current deviation law in design B , and thus may be used to define a second training loss function as

$$L_B(\boldsymbol{\theta}_i) := e\left(\tilde{\mathbf{P}}^B; \hat{\mathbf{P}}^{B,*}\right), \quad \tilde{\mathbf{P}}^B = f\left(\mathbf{C}^B; \boldsymbol{\theta}_i\right). \quad (5.34)$$

In this manner, model parameters $\boldsymbol{\theta}_i$ should now minimize both loss functions $L_A(\boldsymbol{\theta}_i)$ and $L_B(\boldsymbol{\theta}_i)$ associated with measured deviation tensors $\hat{\mathbf{P}}^{A,(1)}$ and $\hat{\mathbf{P}}^{B,*}$, which may be achieved by alternating gradient descent corrections, i.e.,

$$\boldsymbol{\theta}_a = \boldsymbol{\theta}_i - \varepsilon \nabla L_A(\boldsymbol{\theta}_i) \quad (5.35)$$

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_a - \varepsilon \nabla L_B(\boldsymbol{\theta}_a). \quad (5.36)$$

Training may then continue until both loss functions fall below threshold e_{th} , a point where an additional candidate from design C may be considered in analogy to (5.33) and consequently as an extra loss function $L_C(\boldsymbol{\theta}_i)$ leading to three sequential gradient descent corrections per iteration, Fig. 5.7. Because all training designs have now been paired, the testing process may initiate exactly as in the case of two designs, e.g. by constructing a test loss function $L_D(\boldsymbol{\theta}_i)$ which returns the smallest difference between prediction and measured deviation tensors in analogy to (5.26). As Fig. 5.7 suggests, in case of all training and test loss functions falling below threshold e_{th} , the parameter setting of optimum generalization $\boldsymbol{\theta}_G$ will be declared as deviation law $\boldsymbol{\theta}^{(1)}$ and all measured deviation tensors explained by it may be eliminated before next pairing iteration.

The present approach for deviation law identification may be interpreted as a clustering problem, where design-specific errors $L_D(\boldsymbol{\theta}_G)$ are used to separate deviation tensors into two classes: “explained by current deviation law” or “not explained”. Consequently, each pairing iteration yields a cluster of deviation tensors uniquely characterized by its deviation law. More generally, the problem of pairing tensors may be regarded as an unsupervised machine learning problem, because the total number of deviation laws for a given sample size is not known a priori. Evidently, this clustering approach is dependent on the order in which deviation tensors are drawn from first tensor set \mathcal{A} . This is, however, an attractive advantage, because it transfers the random nature of experimental manufacturing variability into deviation laws. In particular, several bootstrapping iterations may increase deviation law diversity and enable a probabilistic deviation estimation via singular value decomposition, as initially explained in Fig. 5.2. Equally attractive is the fact that the entire deviation learning process is driven exclusively by the deviation law threshold e_{th} , see Figs. 5.6 and 5.7. As expected, a small law threshold e_{th} would result in a much more accurate but also more demanding pairing process, which may ultimately impact the

total number of deviation laws found. This parameter may be chosen according to the expected manufacturing differences between designs; a larger variability may require a higher threshold, and vice-versa, see application example in Sec. 5.5.

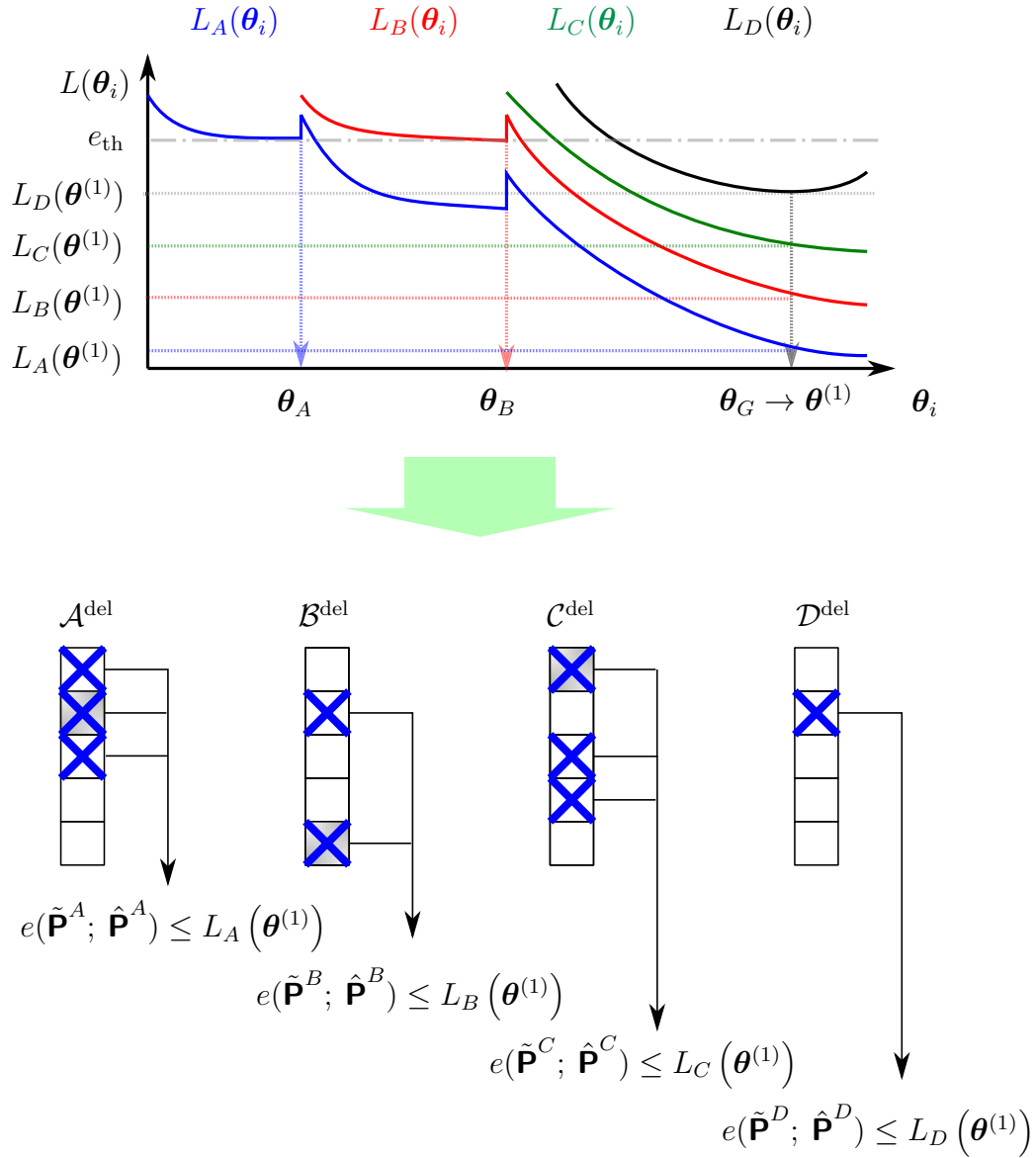


Figure 5.7: Extraction of one deviation law from five manufacturing instances of designs A, B, C and D .

5.4 Minimization of Deviation Prediction Divergence

Having defined a deviation model and a learning procedure, two final steps are lastly required to conclude the present deviation estimation strategy. Since deviation predictions

computed by deviation laws are normalized, an inverse-normalization rule must be first developed. Secondly, since deviation laws will produce numerous deterministic deviation estimations for any new design, a probabilistic representation of surface variability may be found via singular-value decomposition and minimum-covariance determinant as explained in Chapter 4. This variability representation allows the formulation of a probabilistic measure of error between measured and estimated surface variability referred to as *model divergence*. This metric will provide a general assessment of deviation prediction accuracy and thus evaluate the overall learning process. These two additional steps are performed simultaneously, first by introducing a scaling factor α in the inverse-normalization rule, and later by using it to minimize the deviation prediction divergence.

Deviation laws $\boldsymbol{\theta}^{(1)} \dots \boldsymbol{\theta}^{(L)}$ obtained from the previous learning process may be used to estimate normalized control point deviations of a new design N as $\tilde{\mathbf{P}}^{(l)} = f(\mathbf{C}^N; \boldsymbol{\theta}^{(l)})$ using associated characteristic tensor \mathbf{C}^N . A normalization rule may be formulated for deviation estimates $\tilde{\mathbf{P}}$ just as for measured deviations $\hat{\mathbf{P}}$ in Eq. (5.4), however, including an additional scaling factor α , i.e.,

$$\tilde{\mathbf{P}} := \tanh \left(\frac{1}{\alpha} \frac{\mathbf{P}}{\Delta p_\infty} \right), \quad (5.37)$$

where the reference control point deviation Δp_∞ is the same as used in the learning process and the hyperbolic tangent function is evaluated element-wise for the non-normalized estimated deviation tensor \mathbf{P} . Consequently, the desired Cartesian deviation predictions are obtained from the inverse-normalization rule

$$\mathbf{P} := \alpha \Delta p_\infty \tanh^{-1} f(\mathbf{C}; \boldsymbol{\theta}), \quad (5.38)$$

where the normalized estimated deviation tensor $\tilde{\mathbf{P}}$ has now been replaced by the convolutional neural network result $f(\mathbf{C}; \boldsymbol{\theta})$. Because the scaling factor α does not alter model parameters $\boldsymbol{\theta}$ and applies equally to any geometric design, it may be arbitrarily manipulated without violating any deviation law. Therefore, it constitutes uniquely a Cartesian magnification factor which may be exploited to amplify predicted geometric scatter without violating geometry-deviation relationships. This is an attractive feature, since it will result in a more refined performance variability estimation due to manufacturing deviations, thus enhancing uncertainty quantification in robust design. However, without any further model restrictions, this may suggest that any scaling factor is equally justified, which is clearly wrong.

Despite not violating geometry-deviation relationships, scaling factor α does influence deviation prediction accuracy. More precisely, since manufacturing variability is modeled

as a probabilistic process, the impact of scaling factor α on any manufactured design may be assessed by measuring the divergence between two probabilistic processes: the *measured* manufacturing variability (obtained from surface measurements) and the *learned* manufacturing variability (obtained from deviation laws). In this way, the scaling factor α may be chosen such that prediction divergence is minimal, thus increasing amplitude and likelihood of deviation predictions on new designs. Essentially, this prediction divergence may be calculated for any manufactured design and given scaling factor α by generating Cartesian deviation estimates from all deviation laws and comparing the probability of each estimate using measured and learned process parameters, Fig. 5.8. First, each non-normalized deviation estimate $\mathbf{P}^{(l)}$ is rearranged as a column vector $\mathbf{q}^{(l)}$ using Eq. (4.2). For computation of *learned* probabilities (here distinguished by the tilde), the learned static deviation vector $\tilde{\mathbf{q}}$ (4.4) is first computed as mean of deviation estimates and then used to construct individual deviation components $\tilde{\mathbf{x}}^{(l)} = \mathbf{q}^{(l)} - \tilde{\mathbf{q}}$. These may then be processed with singular-value decomposition (4.13) to obtain learned manufacturing modes $\tilde{\mathbf{U}}_r$ (4.14) and mode amplitudes $\tilde{\mathbf{b}}^{(l)}$ (4.19). Using the minimum-covariance method, Sec. 4.4, mode amplitudes $\tilde{\mathbf{b}}^{(l)}$ may be finally approximated by a multivariate Gaussian distribution with robust mean vector $\tilde{\boldsymbol{\mu}}'$ and covariance matrix $\tilde{\mathbf{C}}'$, such that individual probabilities $\tilde{p}^{(l)}$ are available for each deviation estimate using its learned mode amplitude vector $\tilde{\mathbf{b}}^{(l)}$ in Eq. (4.44). These values correspond to the probability with which any deviation estimate $\mathbf{P}^{(l)}$ is predicted to occur in the new geometric design.

Measured probabilities, on the other side, quantify the probability that a given deviation estimate $\mathbf{P}^{(l)}$ would occur from the real process parameters, i.e., from the measured static deviation $\bar{\mathbf{q}}$, observable manufacturing modes \mathbf{U}_{t^*} , robust mean mode amplitude vector $\boldsymbol{\mu}'$ and covariance matrix \mathbf{C}' . In this case, each deviation estimate is first interpreted as a manufacturing process realization, i.e., it is first centered using measured static deviation vector $\bar{\mathbf{q}}$ as $\mathbf{x}^{(l)} = \mathbf{q}^{(l)} - \bar{\mathbf{q}}$ and then projected onto observable experimental manufacturing modes as $\mathbf{b}^{(l)} = \mathbf{U}_t^T \mathbf{x}^{(l)}$ according to (4.16). The resulting mode amplitudes $\mathbf{b}^{(l)}$ correspond to the required values such that deviation estimates $\mathbf{q}^{(l)}$ may occur as real manufacturing instances. Due to the multivariate Gaussian approximation of manufacturing variability, it is possible to compute an associated probability $p^{(l)}$ using Eq. (4.44) with the real process parameters $\boldsymbol{\mu}'$ and \mathbf{C}' estimated directly from surface measurements.

Now, prediction divergence may be initially estimated as difference between learned and measured probabilities $\tilde{p}^{(l)} - p^{(l)}$ for each deviation estimate $\mathbf{q}^{(l)}$. This, however, may cause underflow errors due to low values of the multivariate Gaussian probabilities. Therefore, a better approach consists in taking the log-probability being always negative and normalizing it by the measured probability. This leads to a design-specific, normalized

prediction divergence

$$\delta = \frac{1}{L} \sum_{l=1}^L \left| \frac{\log \tilde{p}^{(l)} - \log p^{(l)}}{\log p^{(l)}} \right|, \quad (5.39)$$

where the absolute value is considered to keep divergence positive. Since the manufacturing process is not exclusively dependent on geometric relationships, the probability p that a deviation estimate is actually observed from measured manufacturing parameters is typically much lower than the learned probability \tilde{p} , i.e. $0 < p < \tilde{p} < 1$, or equivalently $0 > \log \tilde{p} > \log p$. Subtraction of $\log p$ from previous inequality reveals the probability difference bounds as $-\log p > \log \tilde{p} - \log p > 0$, and consequently the prediction divergence bounds as $0 < \delta < 1$. In this manner, a collection of deviation laws with associated divergence δ may be roughly interpreted as follows: on average, the estimated probability of a predicted manufacturing instance deviates by $\delta \times \log p$ orders of magnitude from its real probability. For example $\tilde{p} = 1 \times 10^{-8}$ and $p = 1 \times 10^{-10}$ yield $\delta = 0.2$.

Since magnitudes of multivariate Gaussian probabilities are severely influenced by dimensionality, it is imperative to consider a rather small (≤ 5) number of truncation modes for both measured and learned probabilistic representations. Fortunately, since divergence calculation contains mostly singular-value decomposition operations, it is rather numerically inexpensive, and thus may be evaluated for a dense scaling interval to provide visual assessment of prediction divergence progression for different manufactured designs. An optimal value α^* may then be chosen based on individual judgment, for instance, by minimizing prediction divergence on the training design which most closely resembles new design's manufacturing process, or rather by minimizing the maximum design divergence, e.g.

$$\alpha^* = \operatorname{argmin}_{\alpha} \max_D \delta_D(\alpha), \quad \alpha \in \{1, 1.1, \dots, 4.9, 5\}, \quad (5.40)$$

where divergence $\delta_D(\alpha)$ is evaluated for all manufactured designs D considered in the learning process.

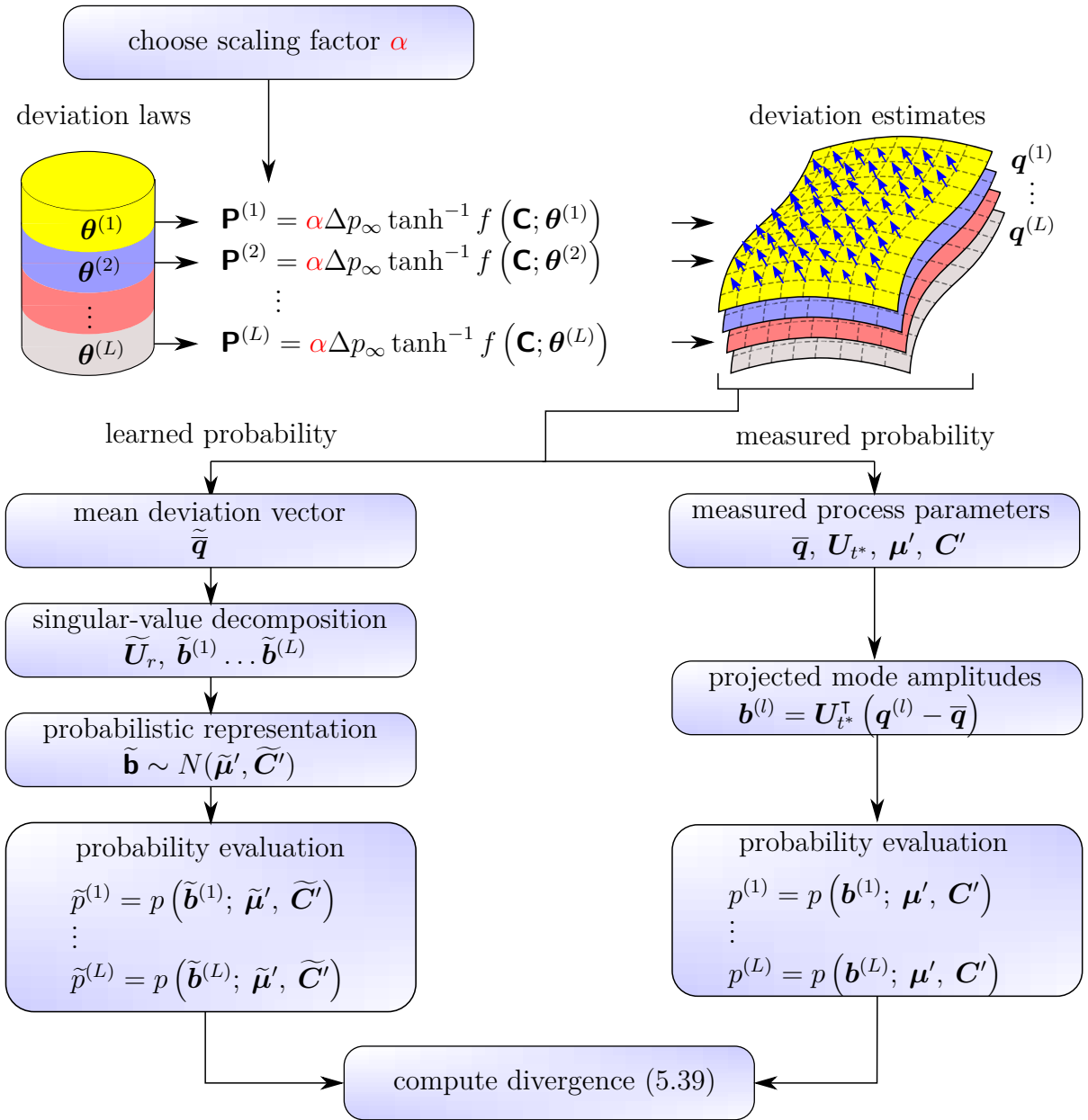


Figure 5.8: Calculation of deviation prediction divergence.

5.5 Application to Estimation of HPT Blade Casting Deviations

The present section will apply the proposed deviation estimation procedure to shank wall pairs of different turbine blade designs. Particularly, the aim consists in learning common deviation patterns from designs A, B and C in Fig. 4.11 and estimate deviations for design D . Since there exist sufficient surface measurements for all four designs with associated manufacturing modes and stochastic mode amplitudes (Secs. 4.5 and 4.6),

deviation predictions for design D may be conveniently compared to the actual experimental deviations. Each B-spline surface has a unique control point grid, which is why the first step consists of refinement of each relevant region to a standardized grid. This forward grid standardization is simply performed by enclosing each relevant area on the respective (i, j) -grid by the smallest rectangle and refining all necessary geometric quantities to a standardized size by means of bilinear interpolation (see Appendix A.2), in this case to a 240×213 grid, Fig. 5.9. This procedure will be then reversed once deviation laws have been found and Cartesian control point deviations are desired.

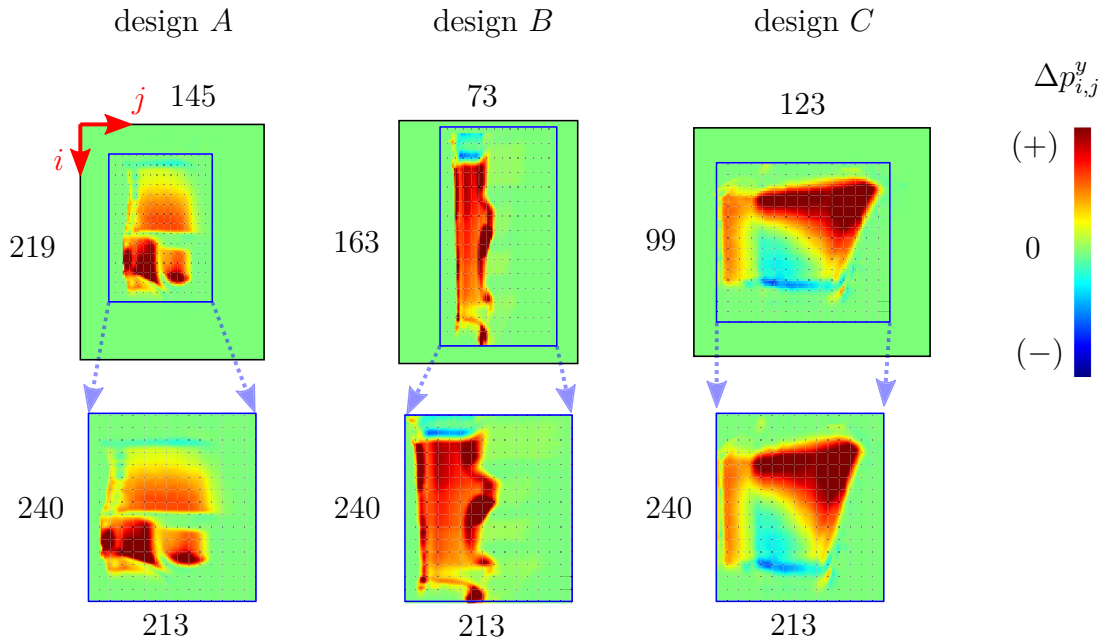


Figure 5.9: Forward grid standardization using bilinear interpolation of control point displacements $\Delta p_{i,j}^y$ for three different pressure side shank designs.

Characteristic tensors may now be created for each of the eight standardized grids (pressure- and suction-side shanks of totally four shank designs). Normalized control point positions $\hat{\mathbf{p}}_{i,j}$ (5.7) may be represented relative to design-specific centroids $\bar{\mathbf{p}}^D$, which are obtained by averaging relevant control points from both pressure- and suction-side shanks of the same design. This parametrization efficiently informs the position of any control point relative to the shank centroid of the turbine blade, and thus may facilitate identification of potential deviation relationships associated to individual shank sides. Normal vectors $\mathbf{n}_{i,j}$ and curvatures $\kappa_{i,j}$ may then be computed directly from standardized control point grids using Eqs. (2.17), (2.20) and (5.9). Reference quantities for control point position p_∞ and curvature κ_∞ may be extracted using quantification rules (5.6) and (5.8) considering all eight refined grids, thus enabling normalization of all nominal geometric quantities. Finally, all irrelevant control point positions are set to zero on each

geometric quantity to isolate the domain of interest, see characteristic tensor for pressure shank of design D , Fig. 5.10.

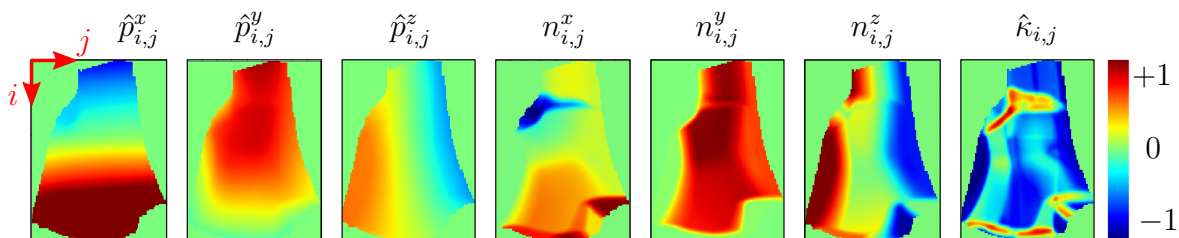


Figure 5.10: Characteristic tensor \mathbf{C}^D for pressure shank of design D .

Normalization of control point displacements needs a reference displacement Δp_∞ which is computed with Eq. (5.3) using a sample of $S = 250$ synthetic morphed surfaces from all training designs (A, B, C), and kept constant for deviation normalization throughout the entire deviation learning procedure. The convolutional neural network architecture chosen for deviation modeling is shown in Fig. 5.11 consisting essentially of three subsequent convolution operations activated by the hyperbolic tangent function. The first two convolutions are followed by average-pooling operations performed on each channel independently with 2×2 windows and stride lengths of 2 cells for the first and second tensor dimensions, which reduces them by half, see Appendix A.2. The third convolution operation reduces the number of channels from 20 to 3, and a subsequent bilinear interpolation is then implemented to refine the first two dimensions to their original size 240×213 . Lastly, the irrelevant region is set to zero and a Gaussian blur is implemented to smooth transitions from relevant to irrelevant areas, see Appendix A.2. The resulting estimated deviation tensor $\tilde{\mathbf{P}}$ may be finally compared to the targeted experimental deviation tensor $\hat{\mathbf{P}}$ by means of loss function (5.15), which is used to update the neural network weights with back-propagation and the Adam optimizer with a learning rate of $\varepsilon = 0.001$, see Kingma and Ba (2015).

The extraction of deviation laws is then performed according to Fig. 5.7, but only with designs A and B for training and design C for testing. Each bootstrapping iteration samples $S = 250$ manufacturing instances from the three designs, where each instance corresponds to coupled deviating shank walls for pressure- and suction-sides. Because each manufacturing instance consists of two surfaces, alternating gradient corrections are necessary, where both characteristic tensor \mathbf{C} and targeted deviation tensor $\hat{\mathbf{P}}$ are alternated in the neural network according to shank side, Fig. 5.12. When loss function $e(\tilde{\mathbf{P}}; \hat{\mathbf{P}})$ surpasses law threshold $e_{\text{th}} = 0.80$ on both shank sides for a given manufacturing instance of design A , the closest manufacturing instance of design B is chosen for additional gradient corrections. This stage is clearly identified as loss jumps in typical convergence diagrams, see iteration steps 25 and 50 in Figs. 5.12a and 5.12b, respectively. As soon as the neural network

describes both manufacturing instances of designs A and B —i.e. four total surfaces—with an accuracy below $e_{\text{th}} < 0.80$, testing evaluations may be performed for design C at selected training steps.

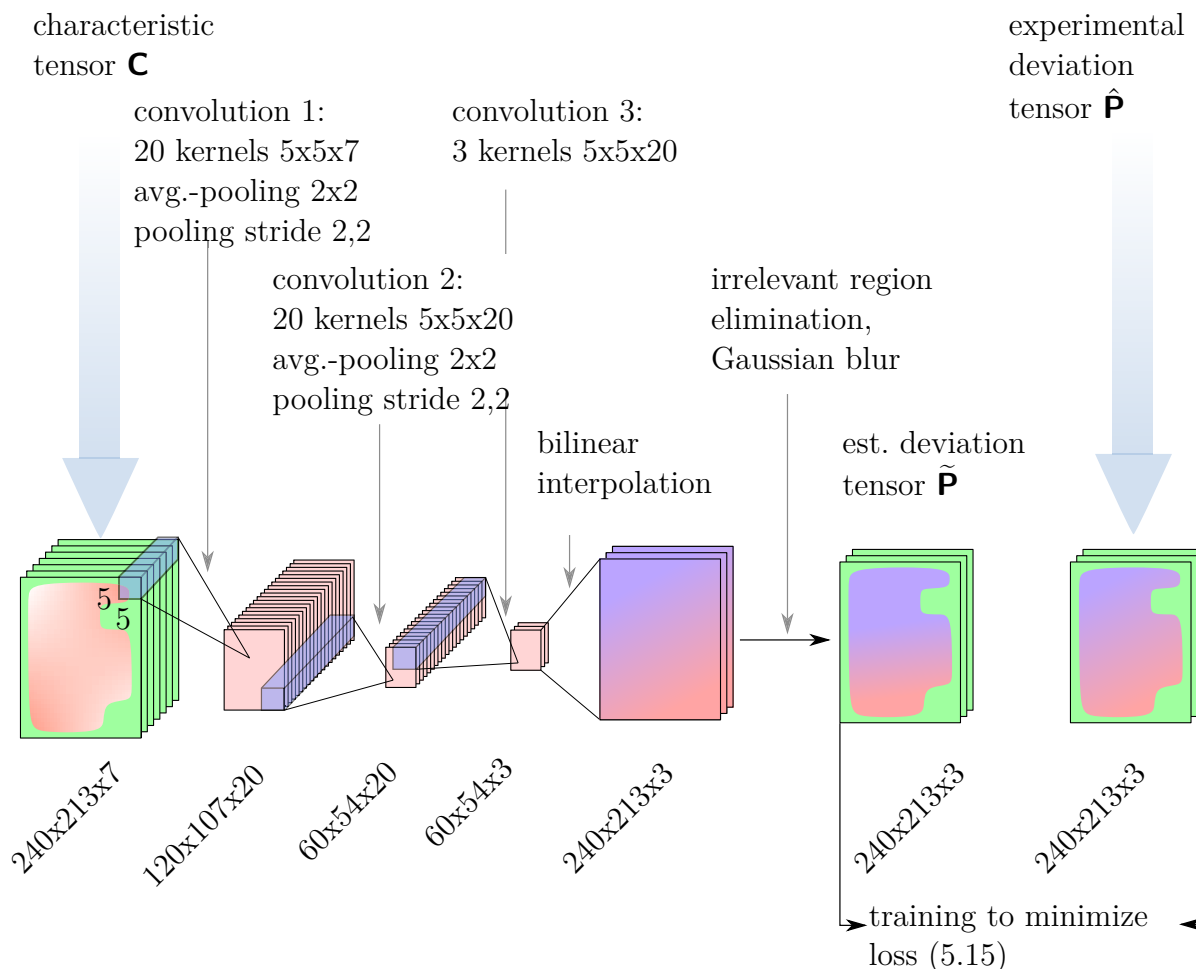


Figure 5.11: Convolutional neural network architecture.

Fig. 5.12a showcases a successful law extraction, where the optimal generalization is found at a training iteration $i = 199$, and demonstrates how the model starts to overfit beyond this point. On the other side, a failed law extraction is represented in Fig. 5.12b, where the test error never reaches the deviation law threshold. The selected threshold $e_{\text{th}} = 0.80$ implies that the mean-squared difference between measurement and prediction images must lie below 80% of the measurement image variance to consider the prediction as valid and, therefore, it is rather high. At this value, approximately four deviation laws are found for each bootstrapping iteration taking about seven hours on a Tesla Volta 100S GPU server with 64 cores at 2.3 GHz. A smaller threshold e_{th} would make the surface pairing process more selective resulting in fewer deviation laws per bootstrapping iteration and ultimately increasing substantially the total computation time. Such a generous threshold,

however, does not inhibit identification of useful deviation patterns from large surface samples, which is why it is considered acceptable in this context.

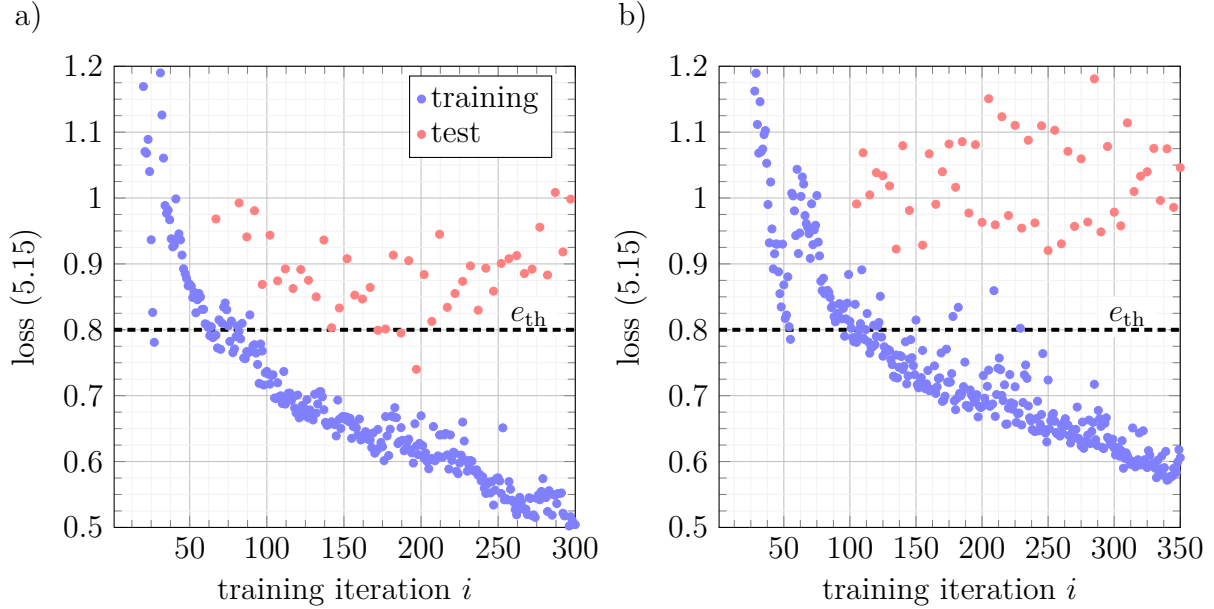


Figure 5.12: Typical convergence diagrams for a) successful and b) failed deviation law extraction.

The deviation learning process was repeated for 31 bootstrapping iterations, leading to $L = 112$ deviation laws, stored as weight vectors of the convolutional neural network. Deviation prediction quality on design D may be now assessed in three ways: first by calculating probabilistic divergence, second by comparing predicted vs. measured manufacturing modes and third by comparing induced stress variability. Because surface measurements are available for every design, learned and measured probabilities of each deviation estimate can be computed using five truncation modes for a scaling interval $\alpha \in [1, 5] \subset \mathbb{R}$, Sec. 5.4. The resulting prediction divergence is shown in Fig. 5.13a. and demonstrates the significant influence of scaling factors on probabilistic accuracy. In general, moderate increases in scaling are beneficial for prediction accuracy, whereas larger values tend to exaggerate deviation predictions and thus amplify divergence. Designs A and B are directly observed during training, and accordingly achieve the lowest divergence values, i.e., their deviation estimates have the highest likelihood among all designs. Although its losses are not propagated through the neural network, design C actively discriminates which deviation laws are preserved and eliminated, and consequently also displays comparatively low divergence values. On the other hand, design D is not seen during training and, therefore, its prediction divergence is at least twice as large as any other training design (for instance $\delta_D(2.5) = 0.8$ vs. $\delta_C(2.5) = 0.4$). This divergence implies that the deviation

estimates for design D are in fact extremely unlikely to be observed in manufacturing. For example, a deviation estimate with learned probability $\tilde{p} = 1 \times 10^{-2}$ would have a measured probability of $p = 1 \times 10^{-10}$.

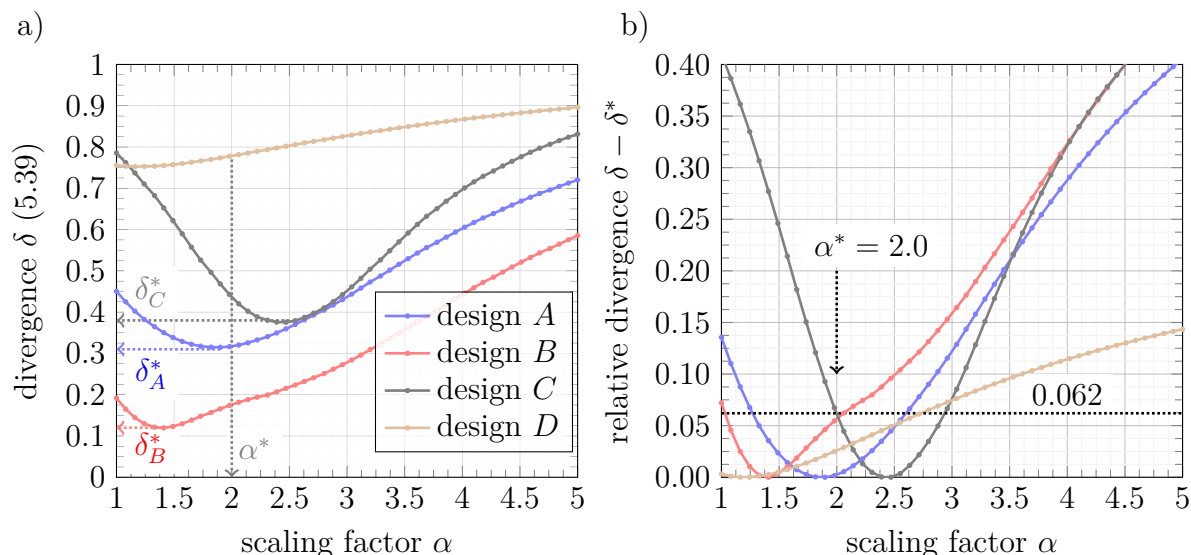


Figure 5.13: Prediction divergence δ a) for different scaling factors α and b) relative to design-specific minima δ^* .

This poor prediction accuracy may be explained by the extremely heterogeneous dataset, i.e., large variability in both manufacturing setting and geometric complexity. Although all turbine blade designs were casted, the specific manufacturing settings—i.e. casting temperatures, mold setup and probably even manufacturing supplier—where different across designs. This is why even common geometric deviation patterns may still display large prediction divergence in new designs due to design-specific manufacturing biases. Furthermore, each geometric design displays significant differences in shape and surface variability (recall that the observable modes ratio for design C was 65% whereas for design A only 20%, Fig. 4.12). It is thus not surprising that the intersection space of surface variability only represents a tiny fraction of real design-specific deviation behavior. However, before formulating potential areas of improvement, the second quality analysis of present deviation prediction will be presented, i.e., manufacturing mode comparison.

Figs. 5.14 and 5.15 show mean deviation components and manufacturing modes for all designs, where blue color indicates displacement directed outwards and red inwards, and each manufacturing mode is weighted by 5th and 95th percentiles of associated mode amplitudes. Mean deviation predictions across all designs show comparable displacement fields (see blue displacements in the blue bounding boxes), which again confirms successful extraction of a common deviation pattern, i.e., y -displacement of the pressure shank and

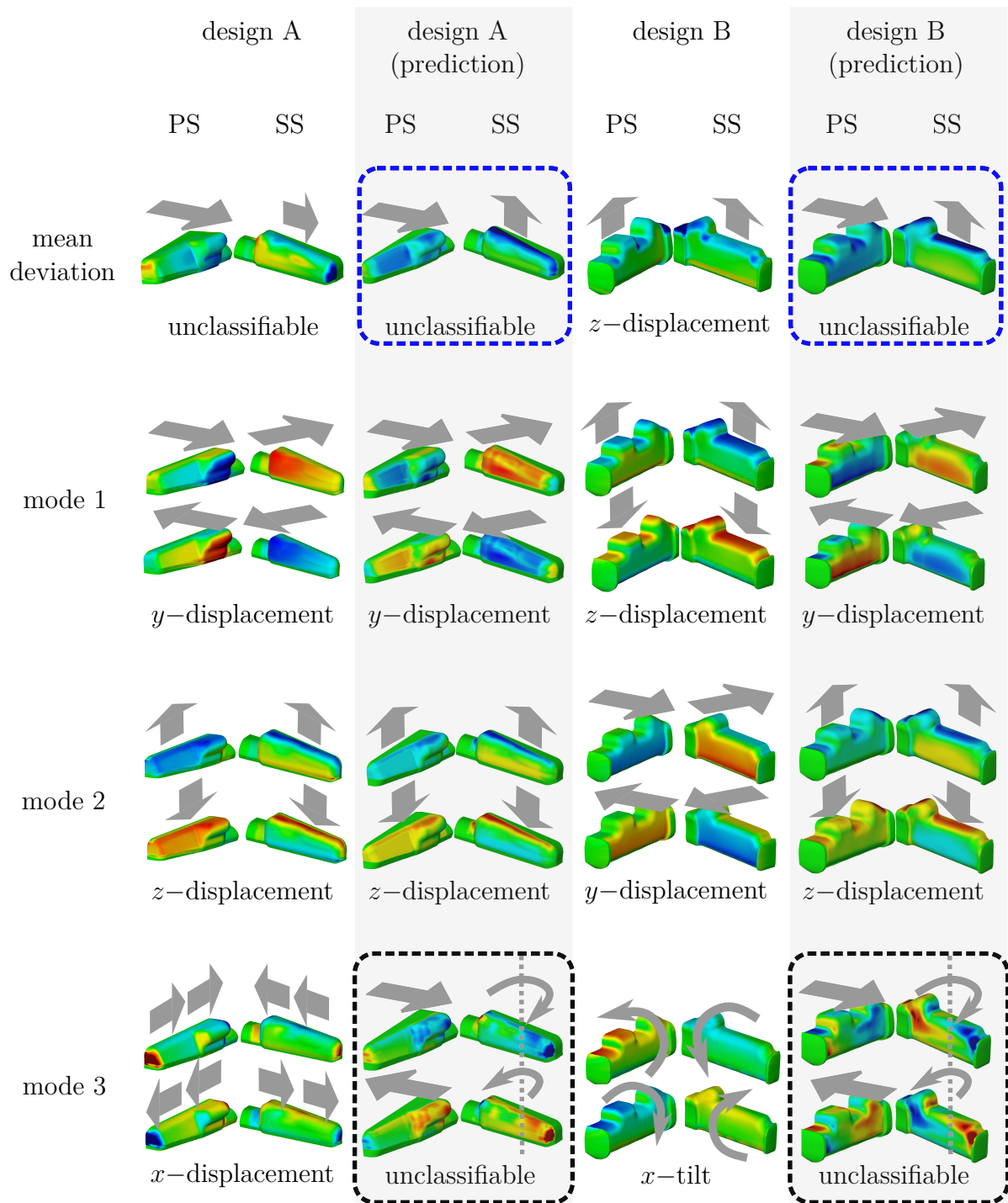


Figure 5.14: Manufacturing mode comparison for designs A and B .

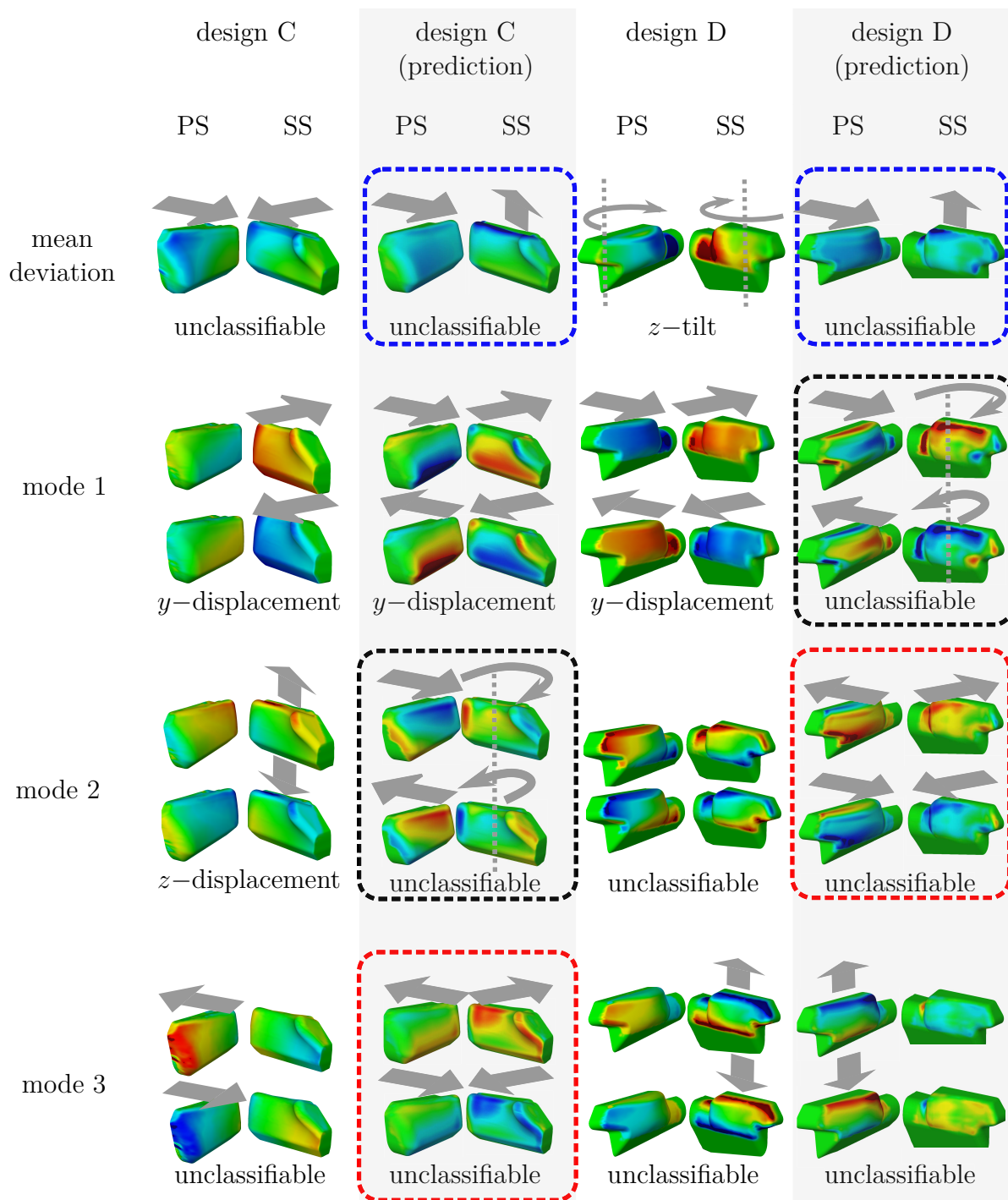


Figure 5.15: Manufacturing mode comparison for designs C and D .

z -displacement of the suction shank. This is a highly satisfactory result since it shows a common deviation pattern smoothly transferred between different designs. In this sense, predicted mean deviation of design D is entirely consistent with the learned deviations from designs A , B and C , but fails at representing reality because design D is the only one showing z -tilt as mean deviation. This result again illustrates the contrast between expected deviation patterns and design-specific manufacturing biases.

Designs A and B are characterized by clear y - and z -displacements as first or second measured manufacturing modes, whereas design C shows this pattern only on the suction side. As a result, none of these patterns is effectively learned and transferred to design D , which again reflects the original intent of extracting only recurrent deviation patterns. Although remaining predicted manufacturing modes cannot be directly associated with any other measured manufacturing modes, a notable resemblance is still found between them. First predicted manufacturing mode on design D , for instance, is also seen on all other designs (black bounding boxes) and corresponds to a mixed displacement-rotation. A similar case is also observed on the second predicted manufacturing mode on design D (red bounding box), which shows some type of contraction-expansion behavior. It must be mentioned, however, that predicted manufacturing modes must be interpreted mainly as statistical characteristics of shared surface variability rather than actual physical processes (as opposed to measured manufacturing modes). This distinction arises from the independence of deviation laws from individual manufacturing process, i.e., by pairing surface measurements from different manufacturing setups, the traceability to a physical process is lost. Nonetheless, this probabilistic representation is equally useful for uncertainty quantification studies and robust optimization.

The third prediction quality analysis compares induced stress variability on design D using predicted and measured manufacturing deviations. To recreate Cartesian deviations, however, an optimum scaling factor α^* minimizing prediction divergence must be chosen. This selection has to be made based exclusively on prediction performance of designs A , B and C , since any new design would not have available divergence values. By observing Fig. 5.13a, it is clear that each design reaches minimum divergence at different scaling factors, which forces formulation of a multi-objective optimization problem. By referencing design-specific divergence to design-specific divergence minima $\delta_E - \delta_E^*$, $\forall E \in \{A, B, C, D\}$ in Fig. 5.13b, a convenient measure of approximation cost is obtained, which may be interpreted as sacrificed divergence given a scaling factor α . On this view, a scaling factor of $\alpha = 2.0$ equally penalizes designs B and C by a small divergence margin of 0.062 while minimizing divergence at design A , which is why $\alpha^* := 2.0$ is chosen as optimum scaling. Although divergence of design D shows least variability, its divergence is still notably close to its minimum at this optimum scaling, which validates this selection criterion.

Using optimum scaling α^* , Cartesian deviation estimates are produced for design D using inverse normalization rule (5.38) and then simulated in the finite-element model described in Sec. 4.5. Maximum stress values for pressure- and suction-side shanks are shown in Fig. 5.16, where 60 random deviation estimates are compared against 58 real deviating geometries for scaling factors $\alpha = 1.0$ and $\alpha = 2.0$. These stress variability histograms may be interpreted as estimated performance scatter in the case where common deviation patterns of designs A, B and C are also observed in design D .

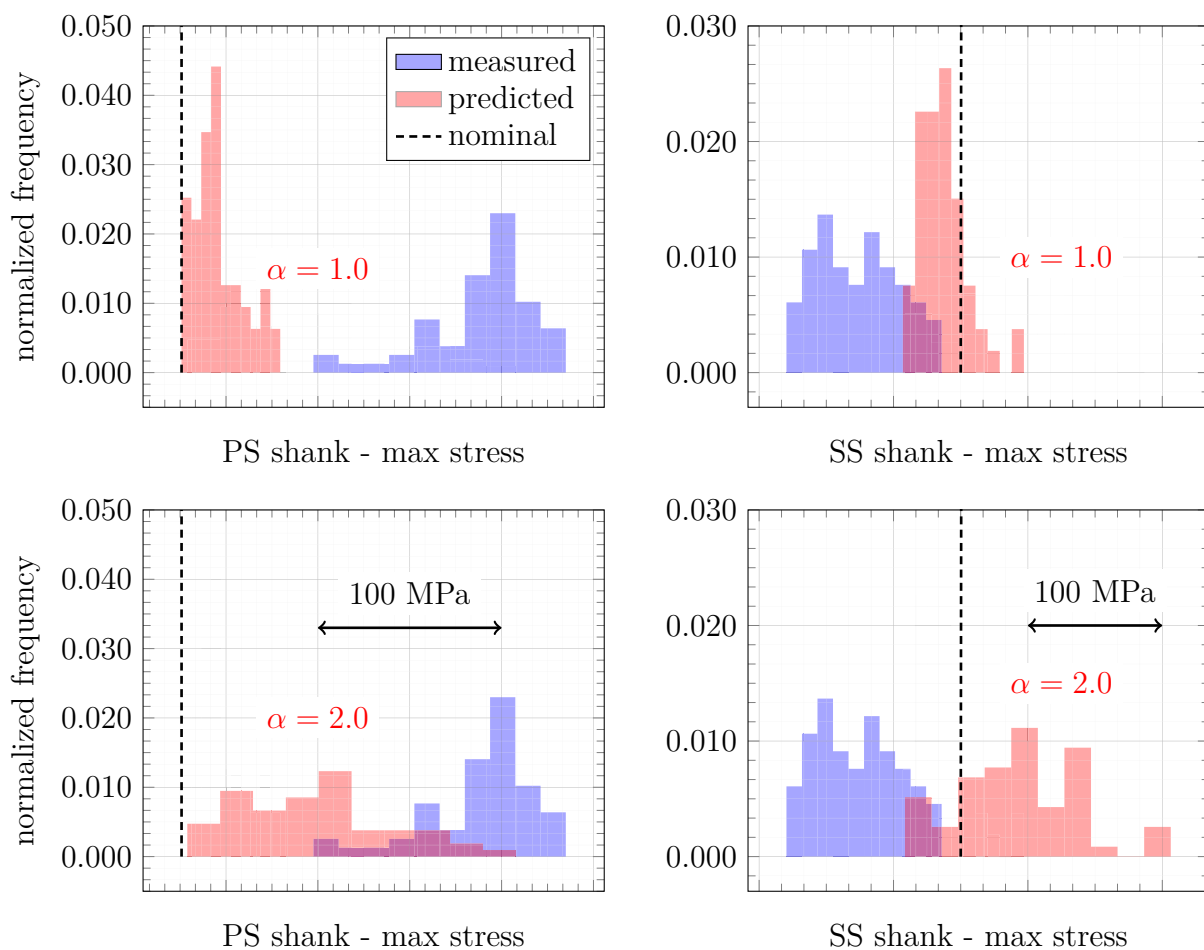


Figure 5.16: Stress variability for design D using different scaling factors α .

Although effectively condensing all recurrent deviation patterns in the learning process, important performance differences are seen between prediction and measurement, which further highlights the major role of manufacturing context. Despite being consistent with training designs, mean deviation prediction on design D significantly differs from the measurement, which further induces different horizontal shifts in the stress histograms: mean stress shift on pressure shank is underestimated (+71.7 MPa vs. +162.6 MPa) while on the suction side overestimated (+46.0 MPa vs. -77.8 MPa), Table 5.1. On the

other hand, optimization of scaling factor α not only reduces prediction divergence, it also amplifies stress standard deviation approximately 40% more on the pressure side (42.0 MPa vs. 29.8 MPa) and 70% more on the suction side (50.4 MPa vs. 29.9 MPa), Table 5.1.

Table 5.1: Measured and predicted shank stress variability [MPa].

		measured	predicted for $\alpha = 1.0$	predicted for $\alpha = 2.0$
pressure side	mean stress shift	+162.6	+20.1	+71.7
	stress std. deviation	29.8	13.3	42.0
suction side	mean stress shift	-77.8	-13.5	+46.0
	stress std. deviation	29.9	17.7	50.4

The above results highlight the high sensitivity of induced stress variability with scaling factor α and the relative importance of mean deviation component vs. manufacturing modes. Although most predicted manufacturing modes cannot be directly associated with measured ones, their induced stress variability is comparable to the real one and, therefore, they may deliver conservative approximations of variability amplitude. The mean deviation component, on the other side, plays a larger role in overall performance variability as it defines mean stress shift, and thus even small geometric differences may disproportionately penalize performance prediction. These findings may guide future research in two directions. Although it is indisputable that manufacturing context must be considered in deviation modeling, significant value may be also gained by simplifying manufacturing and geometric space. Similarly as in Zhu et al. (2019) and Zhu et al. (2020), isolated manufacturing parameter modification and simplified specimen testing might better help the current deviation strategy to underline fundamental deviation effects and increase prediction confidence within a controlled range of manufacturing settings. Similarly, the convolutional neural network may be enhanced by including simulated information of the manufacturing process, such as temperatures and stress fields. Finally, because upper investigations demonstrate the greatest effect in performance variability, the second research direction regards a greater emphasis to mean deviation prediction. This could be achieved, for example, by isolating static deviation component in a separated prediction strategy.

5.6 Integration of Deviation Laws into Robust Design

Previous sections describe the extraction of surface variability from different manufactured designs into universal deviation laws. Application to shank variability prediction

demonstrates that, despite their limitations, deviation laws may provide acceptable estimations of induced performance variability. As a consequence, they might serve as a first approximation of manufacturing variability within a robust design process, where the geometry is optimized iteratively with respect to its expected performance scatter. Such a workflow is sketched in Fig. 5.17 and summarizes many of the methods described in present dissertation.

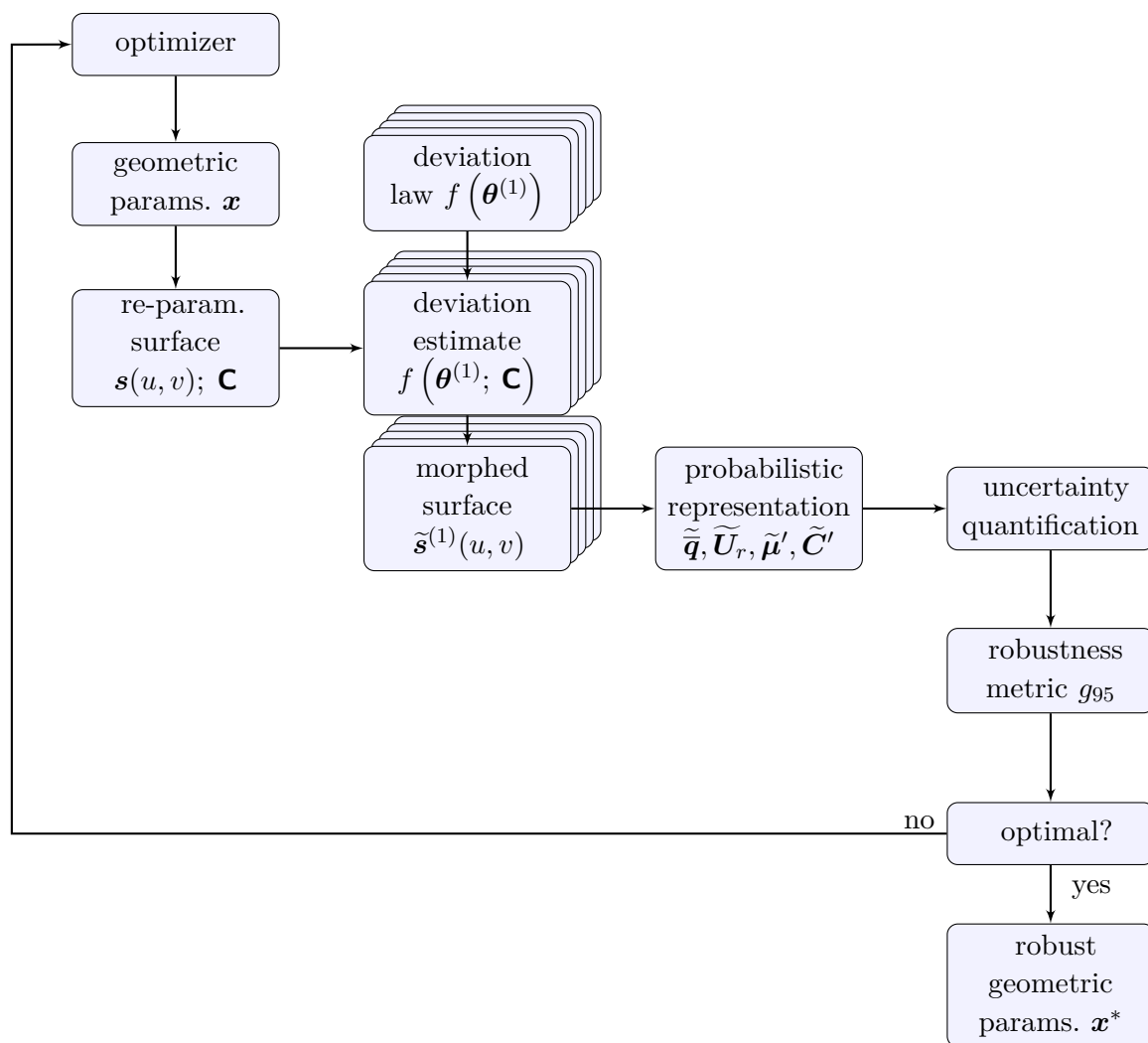


Figure 5.17: Integration of deviation laws into a robust design process.

To enable such process, it is first necessary to extract deviation laws $\boldsymbol{\theta}^{(1)} \dots \boldsymbol{\theta}^{(L)}$ from previously manufactured versions with the same geometric features and an optimized scaling factor α^* , Sec. 5.5. Having collected sufficient deviation laws in a database of convolutional neural network weights, the optimization strategy may be defined. For manageable computational costs, the nominal geometry is typically parametrized with relatively few parameters, for instance by considering only selected radial airfoil sections

in the case of a turbine blade. The associated n geometric parameters may be stored in a vector $\mathbf{x} \in \mathbb{R}^n$, such that for each optimization iteration a new nominal CAD face—not necessarily a B-spline surface—is constructed. To represent potential manufacturing deviations of the nominal face, it is necessary to re-parametrize it as dense B-spline surface $\mathbf{s}(u, v)$, just as previously done for shank or hot-gas faces, Figs. 3.3 and 3.16, respectively. Furthermore, to recreate possible manufacturing deviations using convolutional neural networks, the re-parametrized surface $\mathbf{s}(u, v)$ has to be standardized and later represented as normalized characteristic tensor \mathbf{C} .

Characteristic tensor \mathbf{C} may now serve as input for all deviation laws, each one producing standardized, normalized deviation estimates $f(\boldsymbol{\theta}^{(1)}; \mathbf{C}) \dots f(\boldsymbol{\theta}^{(L)}; \mathbf{C})$. These may be then interpolated back to its original control point grid and scaled using optimal factor α^* to produce Cartesian control point deviations, which may be then considered in the re-parametrized surface $\mathbf{s}(u, v)$ to produce morphed surfaces $\tilde{\mathbf{s}}^{(1)}(u, v) \dots \tilde{\mathbf{s}}^{(L)}(u, v)$. Because a discrete set of deviation estimates does not suffice for accurate uncertainty quantification, a probabilistic representation may be performed, where singular-value decomposition, outlier elimination by minimum-covariance determinant and normal distribution fitting produce a probabilistic representation of manufacturing scatter, Chap. 4. Such a probabilistic description, consisting of predicted mean deviation vector $\tilde{\mathbf{q}}$, manufacturing modes $\tilde{\mathbf{U}}_r$, robust mean amplitude vector $\tilde{\boldsymbol{\mu}}'$ and amplitude covariance matrix $\tilde{\mathbf{C}}'$, may now allow diverse uncertainty quantification methods, such as Monte-Carlo sampling of further deviation estimates. To propagate manufacturing uncertainty into performance variability, each deviation estimate may be analyzed according to the relevant physics—for instance using aerodynamic or structural models—to obtain objective function g . This uncertainty propagation might induce important computational costs, which is why numerical methods such as adaptive response surfaces or multi-fidelity Monte-Carlo may become necessary. As a result of this process, a single (or multiple) robustness metrics may be extracted from all objective function evaluations—such as the 95th percentile g_{95} —and associated to the nominal geometry \mathbf{x} of current iteration. In this manner, the optimizer may use obtained robustness metrics of the current geometry to continuously propose better geometry candidates until a robust optimal design vector \mathbf{x}^* is found.

A robust design process is undoubtedly computationally expensive. The additional computational cost of including deviation laws, however, is marginal compared to the unavoidable uncertainty quantification process. Because deviation laws are prepared in advance and may be re-used throughout the entire optimization, no neural network training is further involved. Thus, the remaining associated costs involve running the neural network repeatedly to produce deviation estimates and importing morphed surfaces as CAD files to create the probabilistic representation. Furthermore, because deviation laws are a function

of nominal curvature, size and orientation, deviation estimates are created differently for each nominal geometry, which constitutes a unique feature of the proposed robust design strategy. Specifically in design problems with large geometric freedom, deviation laws may certainly offer more realistic representations of potential manufacturing variability than traditional approaches, where parametric uncertainties are typically held constant throughout the entire optimization process.

6 Conclusions

Due to several physical and human interactions, manufacturing may be regarded a stochastic process where any production output is expected to deviate at a certain degree from the original design intent. Failure to correctly estimate and handle this manufacturing variability has severe consequences in complex production setups, as typically seen by elevated rejection rates, high material scrap and, therefore, increased economic and environmental costs. From all types of manufacturing variability, surface deviations have sparked particular research interest as they—besides driving rejection rates—may propagate through entire mechanical systems and severely affect overall component performance. However, because it involves several disciplines such as metrology, CAD parametrization and deviation modeling, current attempts to consider surface variability in product design have either been limited to very specific geometries—such as prismatic specimens or airfoils—or rather involve extensive user interaction—such as CAD surface fitting of complex free-form surfaces. As a response, the present dissertation proposes a collection of methods for CAD representation, probabilistic description and prospective estimation of geometric variability for generic free-form surfaces.

The path towards automated virtual representation of complex manufactured surfaces requires a seamless interface with commercial CAD software representations as well as robustness against potential surface scan irregularities. Furthermore, because geometric complexity of manufacturing deviations may frequently be higher than associated nominal face parametrizations, a more capable CAD representation model different from the nominal one is needed. Therefore, with the major goals of high matching accuracy and method generalization, the present dissertation proposes re-parametrization of any CAD face of interest as B-spline surface with dense control point grid and subsequent matching to measurement scans via B-spline morphing. Such dense control point re-parametrization demonstrates high versatility to rebuild high-curvature turbine blade free-form faces and also shows outstanding matching accuracy with corresponding surface measurements. As downside, dense control point grids require additional strategies to prevent boundary irregularities after B-spline morphing, which are significantly alleviated by local area-averaging of boundary control points. Overall, B-spline re-parametrization coupled with B-spline morphing constitutes a robust, generic and accurate approach for CAD representation

of complex manufactured surfaces and, when paired with a subtractive CAD construction approach, may offer extensive automation potential for geometric reconstruction of complete parts.

Analysis of a single manufacturing instance is, however, not sufficient to derive systematic deviation information from any manufacturing process. This is why common statistical analysis methods, such as principal component analysis, have gained considerable popularity among industry and academia for enabling identification of manufacturing deviation patterns. However, because most studies use discrete geometric representations such as meshes or point clouds, a direct interface between a probabilistic model and continuous, commercial CAD representation formats is missing. In this context, the present dissertation takes advantage of the established interface between surface measurement and CAD representation and proposes probabilistic modeling directly from B-spline control points. Specifically, this approach consists of singular value decomposition of a representative sample of morphed B-spline surfaces into manufacturing modes, computation of statistical mode representation capacity, elimination of unobservable modes, and modeling mode amplitudes as multivariate Gaussian probability distributions with robust covariance matrix. Although probabilistic model selection is highly dependent on the individual manufacturing dataset, synthetic high-pressure turbine blade surfaces generated by the present probabilistic analysis demonstrate excellent resemblance with actual surface measurements, both by deviation morphology and by induced stress variability. However, because multi-modal amplitude distributions and nonlinear mode interactions have been observed in manufacturing literature, the proposed strategy may be treated only as an additional, valuable approach for probabilistic deviation modeling.

In several industrial applications, historic manufacturing deviations remain largely sub-utilized for robust design of new parts, fundamentally due to potential geometric and parametrization differences. Because current robust design strategies address exclusively particular geometries, such as cylindrical specimens or airfoils, the present dissertation aims at formulating a general approach for free-form surface deviation extraction based on machine learning. As demonstrated by shank variability analysis of different turbine blade models, mean deviation component and manufacturing modes may significantly differ between designs, thereby hindering deviation modeling based on shared manufacturing modes. Instead, deviation modeling is developed using individual manufacturing instances, i.e., by associating a particular morphed surface with its corresponding nominal surface via a convolutional neural network. Here, normalized curvature, orientation and position are derived from nominal B-spline surfaces and associated with local control point displacements, such that deviation modeling is possible for any free-form surface independent of control point structure. Additionally, a learning strategy is proposed to identify and

collect convolutional neural networks which correctly model manufacturing deviations on different nominal designs. As a result, stochastic surface variability is captured as ensemble of deviation laws, which may be used to estimate morphed surfaces for any new design and consequently construct a probabilistic representation based on estimated manufacturing modes.

Application to shank surface variability demonstrates successful extraction of over a hundred deviation laws from three different turbine blade designs. Prediction quality and usefulness is evaluated by comparing mean and standard deviation of induced stress variability. Although prediction of the mean deviation component for the control shank design is highly consistent with predictions for shank designs used in training, it fails to reflect actual manufacturing evidence of the test design, which was consequently manifested as representative errors in mean stress shift prediction. This result likely arises from important differences in manufacturing settings between training and test designs. Partially due to the same reason, stress standard deviation is overestimated. These results highlight an important drawback of exclusive geometric deviation modeling, i.e., limited prediction accuracy due to avoidance of manufacturing context.

Future improvements of deviation estimation strategies are certainly possible. Perhaps a major advancement may arise in an environment where direct access and influence to the underlying manufacturing process is facilitated. This would allow design of larger test campaigns with simplified nominal geometries and controlled manufacturing parameters, and thus help to refine the relationship between deviation, manufacturing and nominal geometry. Similarly, deviation modeling may be extended to consider physical parameters of the manufacturing process, such as temperatures, residual stresses, or displacements, which are typically available in commercial manufacturing simulation software for casting or additive processes. Because this information is distributed spatially, it may be conveniently included in the convolutional neural network model by adding additional channels to the characteristic tensor; therefore, its further integration into machine learning models is highly encouraged. Additionally, since deviation forecasts are complex in both geometry and probabilistic domains, further efforts may be spend in developing more intuitive metrics for better quantification of prediction divergence. This would facilitate integration and adoption of deviation modeling approaches into product design by a larger technical audience.

Appendix A

Convolutional Neural Networks

A convolutional neural network is a special mathematical model used for pattern extraction from structured information arrays, such as B-spline surfaces. In the following, their basic mathematical background and some relevant operations will be introduced.

A.1 Basic Mathematical Background

Artificial neural networks are mathematical models inspired by the human brain's learning process. Essentially, they consist of an extensive arrangement of simple mathematical relationships performed on the model input $\mathbf{x} \in \mathbb{R}^m$ to approximate a desired output $\mathbf{y} \in \mathbb{R}^n$. In a neural network, each mathematical relationship is interconnected hierarchically with the others by means of layers, such that the model capacity is efficiently controlled by the neural network depth. The most fundamental units of an artificial neural network are perceptrons, see gray circles in Fig. A.1, which produce a single value h from all incoming connections \mathbf{x} as $h = f(\mathbf{w}^T \mathbf{x} + b)$. Here, quantities \mathbf{w} and b correspond to perceptron weights and biases, and an activation function f is chosen to prevent diverging network values and also provide some model non-linearity. Typical activation function examples include sigmoid function $f(a) = 1/(1 + e^{-a})$, hyperbolic tangent function $f(a) = \tanh(a)$ and rectified linear unit $f(a) = \max\{0, a\}$.

In a feed-forward neural network, complex relationships between model input \mathbf{x} and output \mathbf{y} may be mapped using multiple interconnected neuron arrays, i.e., hidden neuron layers, such that any hidden layer $\mathbf{h}^{(i)}$ is related to its predecessor layer $\mathbf{h}^{(i-1)}$ by

$$\mathbf{h}^{(i)} = f(\mathbf{W}^{(i)} \mathbf{h}^{(i-1)} + \mathbf{b}^{(i)}), \quad i = 1, \dots, D, \quad (\text{A.1})$$

and the final output layer $\hat{\mathbf{y}}$ is related to the last hidden layer $\mathbf{h}^{(D)}$ as

$$\hat{\mathbf{y}} = f(\mathbf{W}^{(D+1)} \mathbf{h}^{(D)} + \mathbf{b}^{(D+1)}) \quad (\text{A.2})$$

where D corresponds to the network depth.

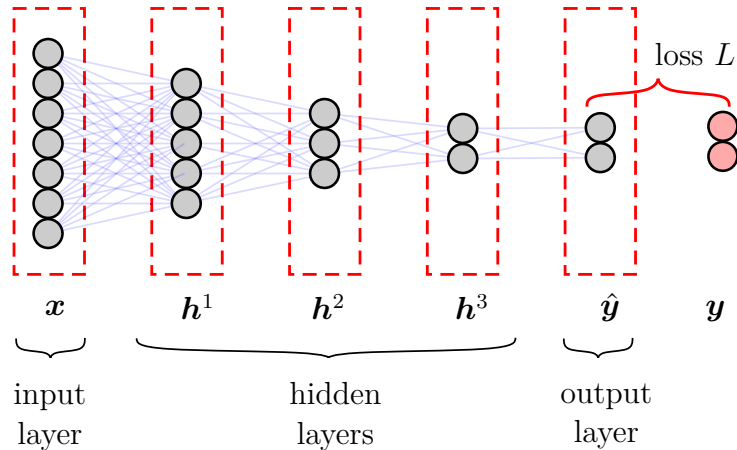


Figure A.1: Artificial feed-forward neural network.

Training a neural network to compute predictions $\hat{\mathbf{y}}$ closely matching observations \mathbf{y} may seem a strenuous task considering the extensive number of network parameters $\{\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(D+1)}, \mathbf{b}^{(1)}, \dots, \mathbf{b}^{(D+1)}\}$ summarized as $\boldsymbol{\theta}$. However, due to the hierarchical construction of hidden layers (A.1), computation of network parameters $\boldsymbol{\theta}$ is possible with gradient-based optimization methods. Typically, a loss function $L = \|\hat{\mathbf{y}} - \mathbf{y}\|^2$ is defined and network parameters $\boldsymbol{\theta}$ are optimized iteratively using gradient descent, e.g.

$$\mathbf{W}_{j+1}^{(i)} = \mathbf{W}_j^{(i)} - \varepsilon \frac{\partial L}{\partial \mathbf{W}^{(i)}} \Big|_{\boldsymbol{\theta}_j} \quad \text{and} \quad \mathbf{b}_{j+1}^{(i)} = \mathbf{b}_j^{(i)} - \varepsilon \frac{\partial L}{\partial \mathbf{b}^{(i)}} \Big|_{\boldsymbol{\theta}_j} \quad (\text{A.3})$$

where the sub-index j corresponds to the iteration step, ε is a small learning rate and $\partial L / \partial \mathbf{W}^{(i)}$ and $\partial L / \partial \mathbf{b}^{(i)}$ are partial derivatives of loss function L w.r.t. weights and biases of the i -th layer.

Because neural networks are computational graphs composed of simple mathematical functions, propagation of model errors into network parameters may significantly benefit from the chain rule of calculus and automatic differentiation. This procedure is also referred as back-propagation, Goodfellow et al. (2016), and essentially traverses the network backwards performing successive applications of the chain rule for each encountered neuron. By computing gradients following the network paths, redundant operations and sub-expressions are significantly reduced, which in turn minimizes total memory requirements. Currently, gradient calculation by back-propagation is already implemented in many computational frameworks for artificial neural networks, such as TensorFlow, Abadi et al. (2015), or PyTorch, Paszke et al. (2019). Furthermore, pure gradient descent (A.3)—although conceptually correct—is almost always an inefficient network update scheme given the large dimensionality of the problem. That is why current methods for neural network training consider robust gradient estimators by averaging gradient information

from random mini-batches of the training sample (stochastic gradient descent) and may also consider adaptative learning rates and refined initialization strategies.

In applications where model inputs follow a grid-like data structure—such as images or B-spline surfaces, the convolution operation is typically used in the neural network as it may efficiently perform image manipulations while minimizing new additional network parameters, Goodfellow et al. (2016). The convolution operation between a tensor $\mathbf{C} \in \mathbb{R}^{(n+1) \times (m+1) \times c}$ and a kernel $\mathbf{K} \in \mathbb{R}^{(2h+1) \times (2h+1) \times c}$ is defined for element (i, j) as

$$(\mathbf{C} * \mathbf{K})_{i,j} \equiv \sum_{k=1}^c \sum_{\mu=-h}^h \sum_{\nu=-h}^h C_{i+\mu, j+\nu, k} K_{\mu+h+1, \nu+h+1, k}, \quad (\text{A.4})$$

i.e. as element-wise multiplication and sum of overlapping elements, where c is the number of channels on both tensors. An illustration is given in Fig. 5.4b, where a kernel \mathbf{K}^x operates on a characteristic tensor \mathbf{C} to produce estimated deviations $\Delta \tilde{\mathbf{p}}_{i,j}^x$.

An artificial neural network performing several sequential convolution operations on an input tensor \mathbf{C} is thus referred as convolutional neural network, Fig. A.2. Each convolution operation may be treated as an individual hidden layer where each channel of the resulting tensor is obtained by convolution with a different kernel. For example, the input tensor \mathbf{C} (pink) is convoluted (symbol $*$) with the green kernel to produce the green channel of hidden layer 1. The result tensor $\tilde{\mathbf{P}}$ of a convolutional neural network f may be described using the notation $\tilde{\mathbf{P}} = f(\mathbf{C}; \boldsymbol{\theta})$, which implies execution of network f on input tensor \mathbf{C} according to network weights $\boldsymbol{\theta}$, i.e., the collection of all kernel parameters. Similar to multi-layer perceptrons, Fig. A.1, convolutional neural networks can be trained using back propagation and stochastic gradient descent to optimize kernel values such that the differences between output tensor $\tilde{\mathbf{P}}$ and a reference tensor \mathbf{P} are minimized.

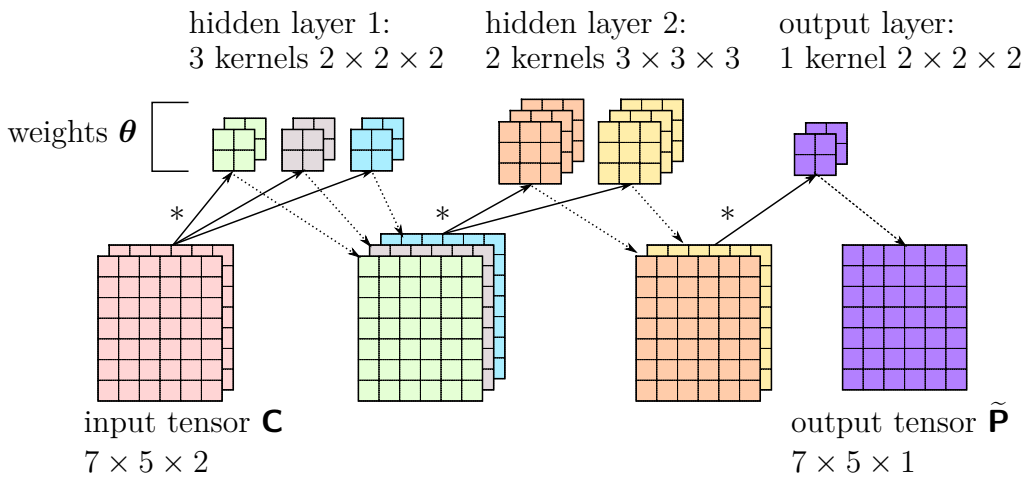


Figure A.2: Convolutional neural network.

A.2 Pooling, Interpolation and Filtering Operations

Historically, the convolutional neural networks in Appendix A.1 have profited from traditional image processing concepts, such as compression, interpolation and filtering, which has given place to analogous mathematical operations compatible with neural network architectures and back-propagation. Consider the convolutional neural network in Fig. A.2, where the first two tensor dimensions (7×5) remain constant throughout the entire graph. In several applications information compression may be desired, for example to force extraction of dominant image patterns, which is why two main alternatives are typically considered. The first one consists of performing the convolution operation with a custom stride length, i.e., by shifting the kernel for more than one cell at a time. This modification substantially reduces dimensionality of the resulting tensor without adding trainable parameters. A second alternative is using a pooling operation after the convolution, which computes moving statistics of the input array, such as maximum, minimum or average values, and may also be defined with a custom stride length, Fig. A.3a. The selection of stride lengths and pooling operations for any convolutional layer may differ according to specific information contraction and array shape requirements, however, it has become common practice to pair each convolution operation with a successive pooling step, Goodfellow et al. (2016).

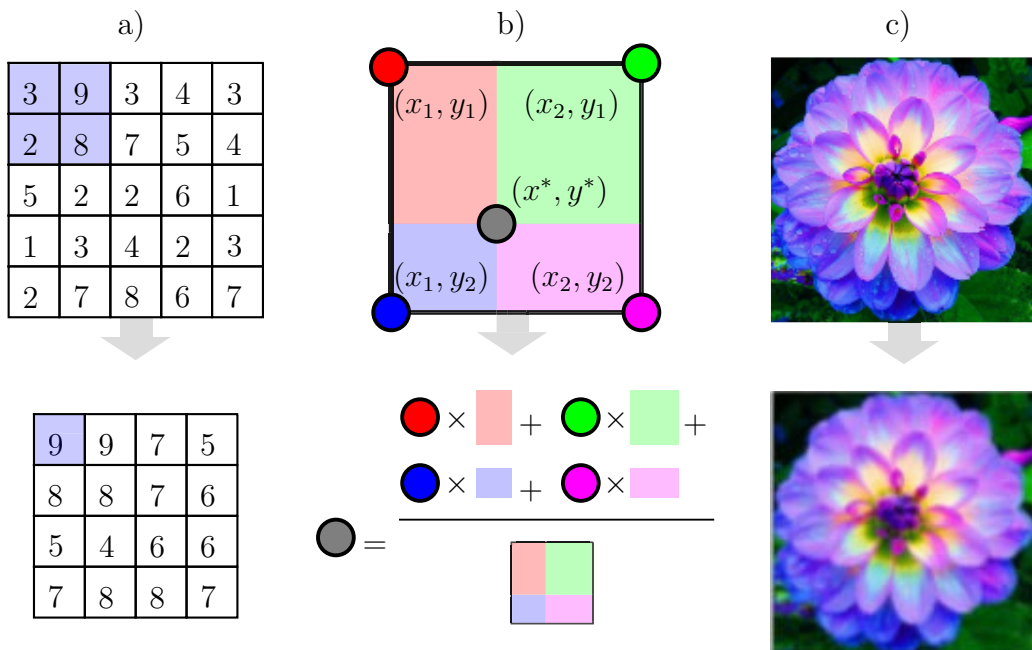


Figure A.3: Illustration of a) 2×2 pooling as maximum of elements in overlapping regions with unitary stride length, b) schematic bilinear interpolation, and c) Gaussian blur.

Contrarily, in cases where image size is fixed and upsampling is required, inexpensive resolution enhancement may be obtained by bilinear interpolation, Fig. A.3b, where the image value at a new pixel position (x^*, y^*) may be approximated by weighting all surrounding pixels with their relative positions, i.e.,

$$f(x^*, y^*) \approx w_{11}f(x_1, y_1) + w_{12}f(x_1, y_2) + w_{21}f(x_2, y_1) + w_{22}f(x_2, y_2) \quad (\text{A.5})$$

where

$$\begin{aligned} w_{11} &= \frac{(x^* - x_1)(y^* - y_1)}{(x_2 - x_1)(y_2 - y_1)}, & w_{21} &= \frac{(x_2 - x^*)(y^* - y_1)}{(x_2 - x_1)(y_2 - y_1)} \\ w_{12} &= \frac{(x^* - x_1)(y_2 - y^*)}{(x_2 - x_1)(y_2 - y_1)}, & w_{22} &= \frac{(x_2 - x^*)(y_2 - y^*)}{(x_2 - x_1)(y_2 - y_1)} \end{aligned}$$

correspond to the relative square area associated to each pixel. This operation may be conveniently written using matrix algebra and thus considered in the neural network architecture without interrupting back-propagation of errors.

In cases where a tensor has been selectively modified and, therefore, exhibits irregular gradients, general purpose filtering such as blurring or sharpening may be useful. These procedures have also been adapted for neural networks as convolutional layers with constant, predefined kernels. For example, blurring is based on the kernel

$$K(x, y; \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma^2}\right), \quad (\text{A.6})$$

derived from the bivariate normal probability distribution, parametrized by a standard deviation σ and referenced to a center position (x_0, y_0) . An example is given in Fig. A.3c where a 144×144 RGB image is blurred by a 5×5 kernel with standard deviation $\sigma = 1.0$.

List of Figures

1.1	Global energy consumption (a) according to different sectors and (b) exclusively industrial sector. Acronyms in the List of Abbreviations.	3
1.2	Examples of CAD reconstruction methods.	5
1.3	Complex free-form surfaces in a high-pressure turbine blade.	6
2.1	Examples of primitive and free-form faces of an HPT blade.	12
2.2	Representation of B-spline by a) basis functions $N_{i,q}(u)$ resulting in b) curve $\mathbf{c}(u)$ with six respective control points \mathbf{p}_i (\circ).	13
2.3	Exemplary B-spline surface parametrization for an HPT blade hub-endwall before trimming.	14
2.4	Error of B-spline surface re-parametrizations of half-cylinder with a) 8×2 and b) 13×2 control point grids.	15
2.5	Orientation and curvature definitions for a B-spline surface.	16
2.6	Illustration of a) orientation and curvature definitions based on control point grid and b) parametric curve $\mathbf{p}(s)$ from a control point row.	18
2.7	Subtractive construction approach for a blade root: a) initial geometry, b) subtraction body with face parametrization, c) resulting face and d) final geometry after several subtraction steps.	20
2.8	Schematic representation of the camera-projector model.	21
2.9	Geometric relationships for surface measurement with fringe patterns.	23
3.1	Developed workflow for CAD representation of manufacturing deviations.	30
3.2	Illustration of a) HPT blade surface measurement and b) associated subtraction body for shank construction.	31
3.3	Illustration of a) nominal subtraction body and b) associated re-parametrized B-spline surface with a 135×96 control point grid.	32
3.4	B-spline surface $\mathbf{s}(u, v)$ (control points not shown) with associated segmented scan \mathcal{M}	33
3.5	Exemplary control point selection for a) B-spline surface section using b) a small distance threshold d_{cp} and c) a large distance threshold d_{cp} , as well as d) a large distance threshold d_{cp} with boundary control.	35

3.6	Calculation of tangential control point deviation $d_{i,j}^j$	37
3.7	Schematic representation of control point support (dark blue).	38
3.8	Definition of morphing error e_M	41
3.9	Illustration of a) exemplary binary matrix \mathbf{C} , b) convolution $\mathbf{C} * \mathbf{J}_3$ and c) resulting boundary positions using $\alpha = 3$	43
3.10	Smoothing of a) exemplary deviation matrix \mathbf{P}^x with gray relevant region and red noisy elements by b) defining matrix \mathbf{Q}^x with smoothed relevant region finally resulting in c) matrix \mathbf{R}^x with smoothed irrelevant region where $\beta = 3$	44
3.11	Illustration of a) HPT-blade casting faces with details of b) <u>p</u> ressure <u>s</u> ide and c) <u>s</u> uction <u>s</u> ide shank faces.	45
3.12	Segmented scan of a suction-side shank with a) relevant surface points $\mathbf{s}(u, v)$, $(u, v) \in \mathcal{U}$, and b) control points $\mathbf{p}_{i,j}$, $(i, j) \in \mathcal{I}$	46
3.13	Morphed surfaces a) without and b) with boundary noise elimination.	46
3.14	Morphing error e_M as percentile of all $ \mathcal{U} = 31\,640$ relevant surface positions for different morphing iterations m_{it}	47
3.15	Morphing error e_M after $m_{it} = 2$ morphing iterations for the SS shank.	47
3.16	Illustration of a) typical hot-gas faces with exemplary control point structure and b) associated re-parametrization as four adjacent B-spline surfaces.	48
3.17	Morphed and post-processed hot-gas faces.	49
3.18	Morphing error e_M after $m_{it} = 2$ morphing iterations for the hot-gas faces.	50
4.1	Reconstruction error $e_R^{(s),t}$ for t manufacturing modes.	60
4.2	Computation of global reconstruction error $\bar{e}_R(t)$ using percentiles.	61
4.3	Exemplary mode amplitudes with outliers marked as red.	64
4.4	Global reconstruction error $\bar{e}_R(t)$ for different truncation orders t	67
4.5	Local reconstruction error $e_R^{(s),t^*}(u, v)$ using $t^* = 38$ observable modes for a) hot-gas, b) suction- and c) pressure-side shank faces.	68
4.6	Correlation analysis using a) scatter diagram for 8 th and 3 rd mode amplitudes and b) Pearson correlation matrix for first 10 modes.	69
4.7	Test for mode normality using a) original vs. b) uncontaminated surface measurements and c) corresponding test failure histograms.	70
4.8	Comparison between 20 synthetic (pink) vs. 10 real (blue) morphed surfaces for a) airfoil leading-edge, b) airfoil trailing edge and c) hub fillet-radius.	71
4.9	Finite-element mesh with domain details.	72
4.10	Stress variability for measured and synthetic morphed surfaces.	73
4.11	B-spline surfaces for various shank designs with associated relevant control points.	74

4.12	Global reconstruction error $\bar{e}_R(i)$ for a) pressure and b) suction sides of different shank designs.	75
4.13	Mean deviation and manufacturing mode comparison for different shank designs.	77
5.1	Calculation of deviation parameters according to surface measurements (blue path) and deviation estimation for a new design N based on deviation parameters (pink path) vs. based on process realizations (green path). . . .	81
5.2	Extraction of deviation laws from manufacturing realizations with similar deviation behavior.	83
5.3	Normalization coefficients for deviation modeling.	85
5.4	Deviation model using convolution operation: a) construction of characteristic and deviation tensors, b) convolution operation, and c) kernel training on the three Cartesian dimensions.	89
5.5	Diagrams for a) dataset approximation and b) loss function (5.19).	91
5.6	Extraction of two deviation laws from five manufacturing instances of two designs A and B	94
5.7	Extraction of one deviation law from five manufacturing instances of designs A, B, C and D	98
5.8	Calculation of deviation prediction divergence.	102
5.9	Forward grid standardization using bilinear interpolation of control point displacements $\Delta p_{i,j}^y$ for three different pressure side shank designs.	103
5.10	Characteristic tensor \mathbf{C}^D for pressure shank of design D	104
5.11	Convolutional neural network architecture.	105
5.12	Typical convergence diagrams for a) successful and b) failed deviation law extraction.	106
5.13	Prediction divergence δ a) for different scaling factors α and b) relative to design-specific minima δ^*	107
5.14	Manufacturing mode comparison for designs A and B	108
5.15	Manufacturing mode comparison for designs C and D	109
5.16	Stress variability for design D using different scaling factors α	111
5.17	Integration of deviation laws into a robust design process.	113
A.1	Artificial feed-forward neural network.	122
A.2	Convolutional neural network.	123
A.3	Illustration of a) 2×2 pooling as maximum of elements in overlapping regions with unitary stride length, b) schematic bilinear interpolation, and c) Gaussian blur.	124

References

- ABADI, M. ; AGARWAL, A. ; BARHAM, P. ; BREVDO, E. ; CHEN, Z. ; CITRO, C. ; CORRADO, G. S. ; DAVIS, A. ; DEAN, J. ; DEVIN, M. ; GHEMAWAT, S. ; GOODFELLOW, I. ; HARP, A. ; IRVING, G. ; ISARD, M. ; JIA, Y. ; JOZEFOWICZ, R. ; KAISER, L. ; KUDLUR, M. ; LEVENBERG, J. ; MANÉ, D. ; MONGA, R. ; MOORE, S. ; MURRAY, D. ; OLAH, C. ; SCHUSTER, M. ; SHLENS, J. ; STEINER, B. ; SUTSKEVER, I. ; TALWAR, K. ; TUCKER, P. ; VANHOUCHE, V. ; VASUDEVAN, V. ; VIÉGAS, F. ; VINYALS, O. ; WARDEN, P. ; WATTENBERG, M. ; WICKE, M. ; YU, Y. ; ZHENG, X.: *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. URL <https://www.tensorflow.org/>, 2015. Accessed November 28, 2022.
- BACKHAUS, T. ; SCHRAPE, S. ; HARDING, M. ; VOIGT, M. ; MAILACH, R.: Validation Methods for 3D Digitizing Precision Concerning Jet Engine BLISKs. *Proceedings of Deutscher Luft- und Raumfahrtkongress 2017, Munich, Germany*. (2017) 1–9.
- BENTLEY, J. L.: Multidimensional Binary Search Trees Used for Associative Searching. *Communications of the ACM*. 18 (1975) 509–517.
- BINGOL, O. R. ; KRISHNAMURTHY, A.: NURBS-Python: An Open-Source Object-Oriented NURBS Modeling Framework in Python. *SoftwareX*. 9 (2019) 85–94.
- BOBROWSKI, K. ; FERRER, E. ; VALERO, E. ; BARNEWITZ, H.: Aerodynamic Shape Optimization Using Geometry Surrogates and Adjoint Method. *AIAA Journal*. 55 (2017) 3304–3317.
- BROWN, D. C.: Close-Range Camera Calibration. *Photogrammetric Engineering*. 37 (1971) 855–866.
- BUNKER, R. S.: The Effects of Manufacturing Tolerances on Gas Turbine Cooling. *ASME Journal of Turbomachinery*. 131 (2009) 1–11.
- BUONAMICI, F. ; CARFAGNI, M. ; FURFERI, R. ; GOVERNI, L. ; LAPINI, A. ; VOLPE, Y.: Reverse Engineering Modeling Methods and Tools: A Survey. *Computer-Aided Design and Applications*. 15 (2018) 443–464.

- COOKSEY, R. W.: *Inferential Statistics for Hypothesis Testing. In: Illustrating Statistical Procedures: Finding Meaning in Quantitative Data*, Springer, Cham, 2020, pp: 241–451.
- CREESE, R.: *Introduction to Manufacturing Processes and Materials*. CRC Press, Boca Raton, 1999.
- D’AGOSTINO, D. ; SERANI, A. ; DIEZ, M. ; CAMPANA, E. F.: Deep Autoencoder for Off-Line Design-Space Dimensionality Reduction in Shape Optimization. *Proceedings of AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Kissimmee, USA*. (2018) 1–14.
- D’AGOSTINO, R. B.: An Omnibus Test of Normality for Moderate and Large Size Samples. *Biometrika*. 58 (1971) 341–348.
- DANDEKAR, R. A. ; COHEN, M. ; KIRKENDALL, N.: *Sensitive Micro Data Protection Using Latin Hypercube Sampling Technique. In: Inference Control in Statistical Databases, From Theory to Practice*, Springer Verlag, Berlin, 2002, pp: 117–125.
- EGILMEZ, G. ; KUCUKVAR, M. ; TATARI, O.: Sustainability Assessment of U.S. Manufacturing Sectors: An Economic Input Output-Based Frontier Approach. *Journal of Cleaner Production*. 53 (2013) 91–102.
- FLASSIG, P.: *Unterstützende Optimierungsstrategien zur robusten aerodynamischen Verdichterschaufelauslegung, Doctoral Dissertation*. Shaker, Aachen, 2011.
- FORNBERG, B.: Generation of Finite Difference Formulas on Arbitrarily Spaced Grids. *Mathematics of Computation*. 51 (1988) 699–706.
- FRANCIOSA, P. ; GERBINO, S.: Simulation of Variational Compliant Assemblies with Shape Errors Based on Morphing Mesh Approach. *The International Journal of Advanced Manufacturing Technology*. 53 (2011) 47–61.
- GAMBITTA, M. ; KÜHHORN, A. ; BEIROW, B. ; SCHRAPE, S.: Stator Blades Manufacturing Geometrical Variability in Axial Compressors and Impact on the Aeroelastic Excitation Forces. *ASME Journal of Turbomachinery*. 144 (2021) 1–10.
- GARZON, V. E. ; DARMOFAL, D. L.: Impact of Geometric Variability on Axial Compressor Performance. *ASME Journal of Turbomachinery*. 125 (2003) 692–703.
- GAUN, L. ; BESTLE, D. ; HUPPERTZ, A.: Hot-to-Cold CAD Geometry Transformation of Aero Engine Parts Based on B-Spline Morphing. *Proceedings of ASME Turbo Expo: Power for Land, Sea and Air, Düsseldorf, Germany*. (2014) 1–12.

- GIJO, E. V. ; SCARIA, J.: Process Improvement Through Six Sigma with Beta Correction: A Case Study of Manufacturing Company. *The International Journal of Advanced Manufacturing Technology*. 71 (2014) 717–730.
- GOLUB, G. H. ; VAN LOAN, C. F.: *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 2013.
- GOODFELLOW, I. ; BENGIO, Y. ; COURVILLE, A.: *Deep Learning*. MIT Press, Cambridge, 2016.
- GRAY, A. ; ABBENA, E. ; SALAMON, S.: *Modern Differential Geometry of Curves and Surfaces with Mathematica*. Chapman & Hall/CRC, Boca Raton, 2006.
- GUPTA, V. ; JAIN, R. ; MEENA, M. L. ; DANGAYACH, G. S.: Six-Sigma Application in Tire-Manufacturing Company: A Case Study. *Journal of Industrial Engineering International*. 14 (2018) 511–520.
- HARALICK, R. M. ; STERNBERG, S. R. ; ZHUANG, X.: Image Analysis Using Mathematical Morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. PAMI-9 (1987) 532–550.
- HARIHARAN, P.: *Basics of Interferometry*. Academic Press, Burlington, 2007.
- HEDBERG, Jr. T. D. ; HARTMAN, N. W. ; ROSCHE, P. ; FISCHER, K.: Identified Research Directions for Using Manufacturing Knowledge Earlier in the Product Life Cycle. *International Journal of Production Research*. 55 (2017) 819–827.
- HEINZE, K. ; MEYER, M. ; SCHARFENSTEIN, J. ; VOIGT, M. ; VOGELER, K.: A Parametric Model for Probabilistic Analysis of Turbine Blades Considering Real Geometric Effects. *CEAS Aeronautical Journal*. 5 (2014) 41–51.
- HÖGNER, L. ; VOIGT, M. ; MAILACH, R. ; MEYER, M. ; GERSTBERGER, U.: Probabilistic FE-Analysis of Cooled High Pressure Turbine Blades—Part B: Probabilistic Analysis. *ASME Journal of Turbomachinery*. 142 (2020) 1–10.
- HUANG, Q. ; NOURI, H. ; XU, K. ; CHEN, Y. ; SOSINA, S. ; DASGUPTA, T.: Statistical Predictive Modeling and Compensation of Geometric Deviations of Three-Dimensional Printed Products. *ASME Journal of Manufacturing Science and Engineering*. 136 (2014) 1–10.
- ISO: *Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 203: Application Protocol: Configuration Controlled 3D Design of*

- Mechanical Parts and Assemblies (ISO 10303-203:2011)*. URL <https://www.iso.org/standard/44305.html>, 2011. Accessed November 28, 2022.
- JOLLIFFE, I.: *Principal Component Analysis*. Springer Verlag, New York, 2002.
- KAMENIK, J. ; VOUTCHKOV, I. ; TOAL, D. J. J. ; KEANE, A. J. ; HÖGNER, L. ; MEYER, M. ; BATES, R.: Robust Turbine Blade Optimization in the Face of Real Geometric Variations. *Journal of Propulsion and Power*. 34 (2018) 1479–1493.
- KINGMA, D. P. ; BA, J. L.: Adam: A Method for Stochastic Optimization. *Proceedings of 3rd International Conference on Learning Representations 2015, San Diego, USA*. (2015) 1–15.
- KINGMA, D. P. ; WELLING, M.: An Introduction to Variational Autoencoders. *Foundations and Trends in Machine Learning*. 12 (2019) 307–392.
- LAMB, C. M.: *Probabilistic Performance-Based Geometric Tolerancing of Compressor Blades, Master's Thesis*. Massachusetts Institute of Technology, 2005.
- LANGE, A. ; VOGELER, K. ; GÜMMER, V. ; SCHRAPP, H. ; CLEMEN, C.: Introduction of a Parameter Based Compressor Blade Model for Considering Measured Geometry Uncertainties in Numerical Simulation. *Proceedings of the ASME Turbo Expo: Power for Land, Sea and Air, Orlando, USA*. (2009) 1113–1123.
- LANGE, A. ; VOIGT, M. ; VOGELER, K. ; JOHANN, E.: Principal Component Analysis on 3D Scanned Compressor Blades for Probabilistic CFD Simulation. *Proceedings of AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, USA*. (2012) 1–16.
- LI, S. ; WANG, Z. ; GUAN, J. ; WANG, J.: Uncertainty Evaluation in Surface Structured Light Measurement. *Proceedings of 2021 IEEE 15th International Conference on Electronic Measurement Instruments, Nanjing, China*. (2021) 395–400.
- LI, Y. ; GU, P.: Free-Form Surface Inspection Techniques State of the Art Review. *Computer-Aided Design*. 36 (2004) 1395–1417.
- LUHMANN, T. ; ROBSON, S. ; KYLE, S. ; BOEHM, J.: *Close-Range Photogrammetry and 3D Imaging*. De Gruyter, Berlin, 2019.
- LUO, J. ; LIU, F.: Statistical Evaluation of Performance Impact of Manufacturing Variability by an Adjoint Method. *Aerospace Science and Technology*. 77 (2018) 471–484.

- MARTIN, I. ; BESTLE, D.: Automated Eigenmode Classification for Airfoils in the Presence of Fixation Uncertainties. *Engineering Applications of Artificial Intelligence*. 67 (2018) 187–196.
- PAIGE, C. C. ; SAUNDERS, M. A.: LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Transactions on Mathematical Software*. 8 (1982) 43–71.
- PASZKE, A. ; GROSS, S. ; MASSA, F. ; LERER, A. ; BRADBURY, J. ; CHANAN, G. ; KILLEEN, T. ; LIN, Z. ; GIMELSHEIN, N. ; ANTIGA, L. ; DESMAISON, A. ; KOPF, A. ; YANG, E. ; DEVITO, Z. ; RAISON, M. ; TEJANI, A. ; CHILAMKURTHY, S. ; STEINER, B. ; FANG, L. ; BAI, J. ; CHINTALA, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. *33rd Conference on Neural Information Processing Systems, Vancouver, Canada*. (2019) 1–12.
- PENG, T. ; GUPTA, S. K.: Model and Algorithms for Point Cloud Construction Using Digital Projection Patterns. *ASME Journal of Computing and Information Science in Engineering*. 7 (2007) 372–381.
- PIEGL, L. ; TILLER, W.: *The NURBS Book*. Springer-Verlag, Berlin, 1995.
- PONIATOWSKA, M.: Free-Form Surface Machining Error Compensation Applying 3D CAD Machining Pattern Model. *Computer-Aided Design*. 62 (2015) 227–235.
- RITCHIE, H. ; ROSER, M.: *Our World in Data: CO₂ and Greenhouse Gas Emissions*. URL <https://ourworldindata.org/emissions-by-sector>, 2020. Accessed November 28, 2022.
- ROUSSEEUW, P. J.: Least Median of Squares Regression. *Journal of the American Statistical Association*. 79 (1984) 871–880.
- ROUSSEEUW, P. J. ; VAN DRIESSEN, K.: A Fast Algorithm for the Minimum Covariance Determinant Estimator. *Technometrics*. 41 (1999) 212–223.
- SCHLEICH, B. ; WARTZACK, S.: Challenges of Geometrical Variations Modelling in Virtual Product Realization. *Procedia CIRP*. 60 (2017) 116–121.
- SHOKRI, A.: Reducing the Scrap Rate in Manufacturing SMEs through Lean Six Sigma Methodology: An Action Research. *IEEE Engineering Management Review*. 47 (2019) 104–117.
- SRINIVASAN, V. ; LIU, H. C. ; HALIOUA, M.: Automated Phase-Measuring Profilometry of 3-D Diffuse Objects. *Applied Optics*. 23 (1984) 3105–3108.

- TURNER, J. U.: New Methods for Tolerance Analysis in Solid Modeling. *Proceedings of International Conference on Computer Integrated Manufacturing, Troy, USA*. (1988) 306–314.
- TURNER, J. U. ; WOZNY, M. J.: Tolerances in Computer-Aided Geometric Design. *The Visual Computer*. 3 (1987) 214–226.
- URBANO, J. ; BESTLE, D. ; GERSTBERGER, U. ; FLASSIG, P.: Low-Order Representation of Manufacturing Variations Based on B-Spline Morphing. *Proceedings of ASME International Mechanical Engineering Congress and Exposition, Salt Lake City, USA*. 2B (2019a) 1–11.
- URBANO, J. ; BESTLE, D. ; GERSTBERGER, U. ; MEYER, M.: Explanation of Surface Deviations by Manufacturing Modes. *Proceedings of 11th Dresden Probabilistic Workshop, Dresden, Germany*. (2019b) 1–17.
- U.S. ENERGY INFORMATION ADMINISTRATION: *International Energy Outlook 2021 With Projections to 2050*. URL https://www.eia.gov/outlooks/ieo/pdf/IE02021_ChartLibrary_full.pdf, 2021. Accessed November 28, 2022.
- U.S. PRODUCT DATA ASSOCIATION: *Initial Graphics Exchange Specification IGES 5.3*. U.S. Product Data Association, North Charleston, USA, 1996.
- VAN GROOTEL, A. ; CHANG, J. ; OLIVETTI, E.: *The Role of Manufacturing Variability on Environmental Impact*. In: *REWAS 2019 Manufacturing the Circular Materials Economy*, Springer, Cham, 2019, pp: 19–32.
- VAN GROOTEL, A. ; CHANG, J. ; WARDLE, B. L. ; OLIVETTI, E.: Manufacturing Variability Drives Significant Environmental and Economic Impact: The Case of Carbon Fiber Reinforced Polymer Composites in the Aerospace Industry. *Journal of Cleaner Production*. 261 (2020) 1–9.
- VILLARRAGA-GÓMEZ, H. ; HERAZO, E. L. ; SMITH, S. T.: X-Ray Computed Tomography: From Medical Imaging to Dimensional Metrology. *Precision Engineering*. 60 (2019) 544–569.
- VOIGT, P ; HÖGNER, L ; BACKHAUS, T: *Blade2Parameter: Geometry Reconstruction User Guide*. TU Dresden, Dresden. 2017.
- WEIR, D. J. ; MILROY, M. J. ; BRADLEY, C. ; VICKERS, G. W.: Reverse Engineering Physical Models Employing Wrap-Around B-Spline Surfaces and Quadrics. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*. 210 (1996) 147–157.

- YAU, H.: A Model-Based Approach to Form Tolerance Evaluation Using Non-Uniform Rational B-Splines. *Robotics and Computer-Integrated Manufacturing*. 15 (1999) 283–295.
- ZHU, Z. ; ANWER, N. ; MATHIEU, L.: Geometric Deviation Modeling With Statistical Shape Analysis in Design for Additive Manufacturing. *Procedia CIRP*. 84 (2019) 496–501.
- ZHU, Z. ; FERREIRA, K. ; ANWER, N. ; MATHIEU, L. ; GUO, K. ; QIAO, L.: Convolutional Neural Network for Geometric Deviation Prediction in Additive Manufacturing. *Procedia CIRP*. 91 (2020) 534–539.
- ZUO, C. ; FENG, S. ; HUANG, L. ; TAO, T. ; YIN, W. ; CHEN, Q.: Phase Shifting Algorithms for Fringe Projection Profilometry: A Review. *Optics and Lasers in Engineering*. 109 (2018) 23–59.