Digital Twins in Self Compensating Assembly Processes

Thesis approved by
the Department of Computer Science
Technische Universität Kaiserslautern
for the award of the Doctoral Degree
Doctor of Engineering (Dr.-Ing.)

to

Felix Claus

Date of Defense:	29^{th} of July 2022
Dean:	Prof. Dr. Jens Schmitt
Chair:	Prof. Dr. Klaus Schneider
Reviewer:	Prof. Dr. Hans Hagen
Reviewer:	Prof. Dr. Bernd Hamann
Reviewer:	Prof. Dr. Ralf Müller

DE-386

Felix Claus

Education

2018-2022	PhD. Student Computer Science (TUK) Topic:“Digital Twins in Self Compensating Assembly Processes”.	
2017-2018	Qualification Studies Computer Science (TUK)	
2011-2017	Student Mechanical Engineering (TUK) Specifications: vehicle technology, design theory	Dipl.-Ing.
2002-2011	High-Scool Paul-von-Denis Schifferstadt	Abitur

Work Experience

since 2019	Research associate at TUK
2015-2019	Research Assistant
2015-2016	Trainee RD&D engine testing at Daimler AG Untertürkheim (6 Months)

Industrial Projects

2020-2021	Development of a prototype tool for gravity compensation in cooperation with Porsche AG - based on [2]
2019-2021	Research cooperation with Q-Das GmbH
2017-2018	Applicability of virtual cubing to aluminum parts - in coopertation with Porsche- and Volkswagen-AG
2015-2019	"Virtueller Meisterbock"(VMB) in cooperation with Volkswagen AG

Further Education

2022	Seminar Conflict Management
2022	Seminar Agile Methods
2021	Quality Assurance Representative (TÜV)
2020	Seminar Leadership
2020	Seminar Projectmanagement
2019	Seminar Time Management
2019	Seminar Presentation Skills

Acknowledgements

I want to thank my supervisors Hans Hagen, Bernd Hamann and Ralf Müller for the fantastic support. I am thankful for this constellation of supervises that could help along with any kind of problems I had. Also, they had a great impact on my development as scientific associate and improvement of skills.

Next, I want to appreciate that I was able to be part of the IRTG2057. Whithin this program I had the great opportunity to collaborate with people from different disciplines and nationalities which greatly improved my work.

I want to thank again Hans Hagen and also Achim Ebert for giving me the opportunity to participate at industrial projects where I could get valuable insights that greatly accelerated my PhD studies.

I want to thank all colleagues and former colleagues of the Computergraphics and HCI Group for the great time and valuable exchanges. I enjoyed being part of this unit and thank the heads of this group - Christoph Garth, Heike Leitte, Achim Ebert, and Hans Hagen for the great leadership.

A special thanks to the volunteers Franca-Alexandra Rupprecht, Vera Marie Memmesheimer, and Andrea Schnorr for providing editorial suggestions on an early stage of this dissertation.

Last, I like to thank my family and partner who supported me during the whole time of my studies whenever needed.

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 252408385 – IRTG 2057. I also thank Q-DAS GmbH for providing CAD models.

Abstract

This dissertation describes the implementation, validation, and troubleshooting of “Digital Twins” in assembly processes of thin structures like parts from the automotive and aerospace industry. As requirements in terms of cost, weight, and human (pedestrian) safety are increasing for modern vehicles, thinner materials are used for exterior components. By that, components become softer but less stable which is challenging for the assembly processes and impacts the resulting quality. The most critical quality measures are gap and flushness as these are affecting aesthetics, wind noise, and fuel consumption of the final vehicle. To compensate for geometrical deviations, parts have adjustable mechanical interfaces which are used to tune in gaps and flushness for each individual assembly. For the components being assembled, individual process parameters depending on the geometry of the actual physical part must be defined. This is a challenging task that cannot be solved in a straightforward manner. However, assembly quality can be predicted by setting up individual FEM[1] simulation models for each part being assembled. These simulation models are called DTs[2] as they are enriched with measured properties from the actual physical part. By that, precise predictions can be made and optimal assembly parameters for automated processes are derived. The demonstration use case in this dissertation is the assembly process of exterior car components made from sheet metals. For this kind of process, the geometrical deviations of individual components are crucial and have to be considered by the DT. To capture geometrical deviations, 3D-scanning is employed which provides a high-resolution point cloud representation of the actual physical part. This point cloud is processed further to obtain the DT that preserves the measured geometry.

This dissertation tackles the following challenges: (a) setting up DTs on different level of details, (b) correctly post-processing 3D-scanned data to remove systematical measurement errors, (c) automatically morphing meshes to derive simulation models from measured point clouds, and (d) troubleshooting DTs with human-in-the-loop approaches. For all approaches, validations are provided that underline applicability and benefits. All methods and results are discussed on a high-level perspective and connections as well as the interplay between methods are elaborated. Each method either improves or extends existing approaches or provides benefits, i.e. higher precision, compared to existing solutions.

[1] Finite Element Method

[2] Digital Twin

The major contributions of this work are:

(1) Framework for DT-based self-compensating assembly system for exterior car components.

(2) Method for calculating the gravity-free shape of 3D-scanned non-rigid parts. Verification and validation with experimental data of presented method.

(3) Automated mesh morphing approach for surface meshes to generate geometrical DTs. Verification and validations using experimental data of real car body parts.

(4) Interactive troubleshooting tool for deriving countermeasures based on measured deviations. Verification and validations using experimental data of real car body parts.

Core References

[1] Felix Claus, Franca-Alexandra Rupprecht, and Hans Hagen. Online simulation considering production uncertainties to improve assembly quality. NAFEMS, 2019

[2] Felix Claus, Hans Hagen, and Bernd Hamann. Calculating the gravity-free shape of sheet metal parts. *The International Journal of Advanced Manufacturing Technology*, 113(11-12):3401–3417, 2021

[3] Felix Claus, Bernd Hamann, and Hans Hagen. Calculating stress-free shapes of sheet metal parts measured with over-constrained fixtures. NAFEMS, 2021

[4] Felix Claus, Hans Hagen, Viktor Leonhardt, Heike Leitte, and Bernd Hamann. Interactive Quality Inspection of Measured Deviations in Sheet Metal Assemblies. In *2nd International Conference of the DFG International Research Training Group 2057 – Physical Modeling for Virtual Manufacturing (iPMVM 2020)*, volume 89 of *Open Access Series in Informatics (OASIcs)*, pages 6:1–6:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik

[5] Felix Claus, Bernd Hamann, Heike Leitte, and Hans Hagen. Decomposing deviations of scanned surfaces of sheet metal assemblies. *Journal of Manufacturing Systems*, 61:125–138, 2021

[6] Felix Claus, Bernd Hamann, and Hans Hagen. A finite-element based mesh morphing approach for surface meshes. *Computer-Aided Design*, 146:103232, 2022

[7] Viktor Leonhardt, Felix Claus, and Christoph Garth. Pen: Process estimator neural network for root cause analysis using graph convolution. *Journal of Manufacturing Systems*, 2021

[8] Patrick Ruediger, Felix Claus, Bernd Hamann, Hans Hagen, and Heike Leitte. Combining visual analytics and machine learning for reverse engineering in assembly quality control. *Electronic Imaging*, 2021(1):60405–1–60405–13, 2021

[9] Patrick Ruediger, Felix Claus, Viktor Leonhardt, Hans Hagen, Jan C. Aurich, and Christoph Garth. Previs – a combined machine learning and visual interpolation approach for interactive reverse engineering in assembly quality control, 2022

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

AR	Augmented Reality
AVGs	Automated Guided Vehicles
B-Spline	Bayesian Spline
CMM	Coordinate Measurement Machine
DOF	Degree of Freedom
DT	Digital Twin
FEM	Finite Element Method
FMEA	Failure Mode and Effects Analysis
FPP	Fringe Projection Profilometry
GF	Gravity Free
GUI	Graphical User Interface
IET	Impulse Excitation Technique
iFEM	inverse Finite Element Method
KPI	Key Performance Indicator
L-BFGS	Limited-Broyden-Flechter-Goldfarb-Shanno
LHS	Latin Hypercube Sampling
LIDAR	Light Detection and Ranging

LSQF	Least Squares Fit
MDS	Multidimensional Scaling
ML	Machine Learning
PC	Principle Component
PCA	Principle Component Analysis
POD	Proper Orthogonal Decomposition
RCA	Root Cause Analysis
RE	Reverse Engineering
RMS	Root-Mean-Squares
RP	Reference Point
RPS	Reference Point System
SAM	Spring Analogy Model
SE	Super Element
STL	Steroelithography
VC	Virtual Clamping

Chapter 1

Introduction

This dissertation presents and discusses a DT-based approach for self-compensating assembly processes for the purpose of sheet metal assemblies. To make this topic more accessible for the reader, the following section will give an introduction on a high level.

The process which is being addressed is the assembly of large thin components. These kinds of components often occur in automotive and aerospace, i.e., fuselage, exterior car body parts, etc. These parts can be from sheet metals, composites, or plastic materials. A common challenge when assembling such components is this: the mentioned parts must be considered to be non-rigid. This means they deform under their own weight which makes handling, measuring, and assembly of these components challenging. To improve assembly quality and decrease the amount of cost for rework and rejects, self-compensating production processes are employed which are explained in the following.

Figure 1.1 shows an abstracted representation of a production system. For demonstration, this figure is focused on the process chain and does neglect diverse aspects of a production system outside this scope. The shown production system does compose of three levels: (a) process level, (b) process control level, and (c) monitoring. On the process level, four stages are employed. Each stage can be interpreted as a defined single working step or a union of steps that transform the input towards a defined output. Each stage receives process parameters from its related process controller. A process controller is used to adjust process parameters to achieve the desired process output. At the end of each stage, the achieved output has to pass a quality gate. This gate defines requirements that the interim or final product has to fulfill to be used as input for the next stage. To assess quality goals, different pre-defined KPIs[1] are measured. The acquired measurement data is fed into the

[1] Key Performance Indicator

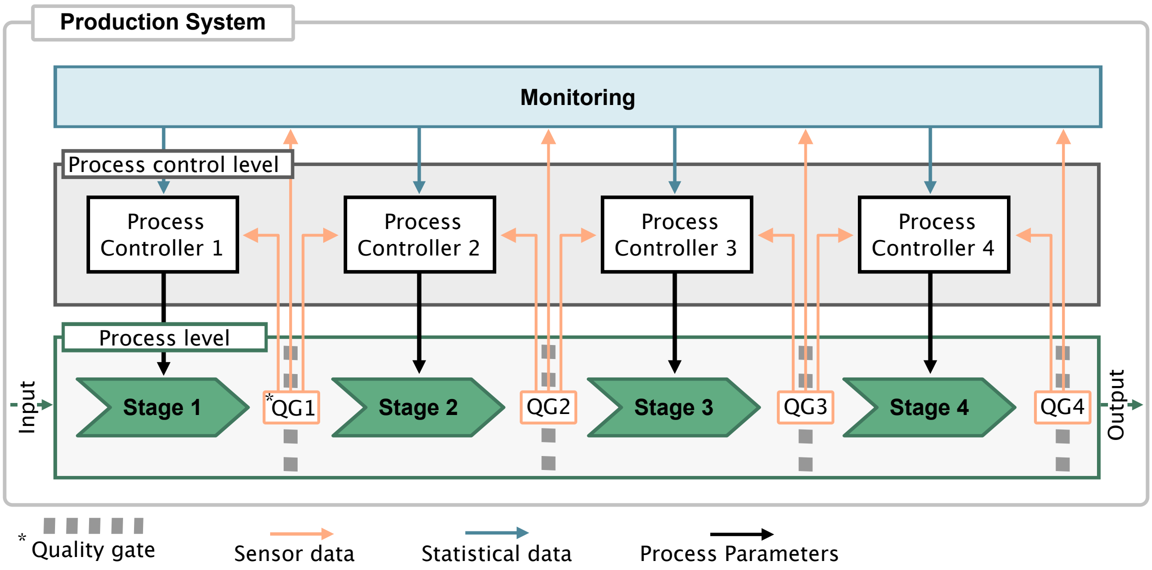


Fig. 1.1 High level description of an production system, divided into three levels: (a) process level, (b) process control level, and (c) monitoring. Each output of a process stage has to pass a quality gate where measures are fed back to level (a) and/or (b).

monitoring level. The monitoring level uses this data and calculates statistical quantities for measuring the performance of the whole production system. If KPIs are trending towards warning limits, adjustments are made to prevent rejects. These adjustments can be made manually or automatically by informing the responsible process controller which performs adequate adjustments to the process parameters. To improve the reaction time of such a system, modern approaches directly feed back sensor data from a quality gate to the previous process controller. The controller then can use this information to adaptively adjust process parameters online. This principle is called closed-loop process or self-compensating process. To improve production systems even further, predictive process controlling can be employed. This is achieved by feeding sensor data from a quality gate to the process controller of the following stage. This process controller uses a prediction kernel to estimate a best-fitting set of process parameters based on the measured input. The prediction kernel can be realized by using a DT that is capable of providing predictions in real time. A DT commonly describes a simulation model of the process which can be enriched with sensor data from the field to make a precise prediction for the physical production system. To set up such DT for process controlling, several challenges have to be overcome. This dissertation will discuss such challenges and provide solutions for the use case of assembly processes of exterior sheet metal car components. A general pipeline for generating such DT is shown in Figure 1.2.

1.1 Structure of this Dissertation

The structure and contributions of this dissertation are explained in the following using Figure 1.2. The first published paper [1] proposes a concept for applying a simulation model capable of making real-time predictions for setting up a self-compensating assembly system. The work [1] outlines this dissertation and identifies several challenges that must be tackled to implement a DT-based prediction model. These challenges are depicted in green and orange and complement to a workflow that is necessary to apply such a concept. Along the workflow, references to published papers are indicating at which step each contribution is situated. Topics that are discussed additionally within this dissertation are marked with an asterisk. The publications [2] and [3] contribute at the beginning of the workflow (data acquisition). They propose and validate methods for correctly post-process 3D-scanned point clouds to compensate for occurring deflections during measurement caused by gravity and fixture. The approaches use FE-based iterative inverse form-finding algorithms to calculate the unloaded shape of a 3D-scanned component. Next, the modeling of DTs is addressed in [6] by proposing an automated mesh morphing algorithm for surface meshes. The papers [4] and [5] are supporting the task "troubleshooting" by providing interactive visualization methods to the inspection engineer. These visualizations are giving valuable insights to optimize the DT model. The topics "application to manufacturing environment" and "verification/validation" are covered as discussion and motivation for adjacent topics.

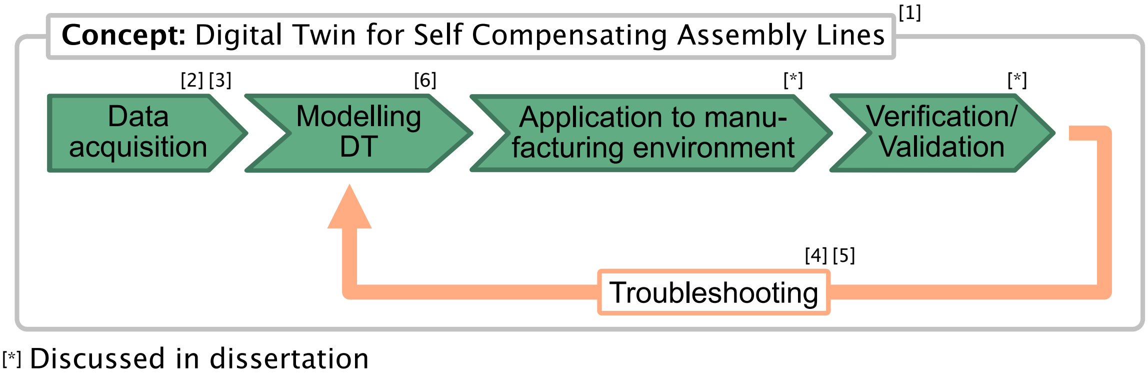

Concept: Digital Twin for Self Compensating Assembly Lines
[1]
[2] [3]
Data acquisition
[6]
Modelling DT
[*]
Application to manufacturing environment
[*]
Verification/ Validation
[4] [5]
Troubleshooting
[*] Discussed in dissertation

Fig. 1.2 Workflow of implementing a digital twin as predictive process controller. Key references are categorized using this flow chart.

1.2 Guidance for the Reader

To prevent misunderstandings or confusion, this section provides instructions for the reader to different style-related structures of this dissertation. Also, clarification for ambiguous wording is provided.

1.2.1 Structure of Background Sections

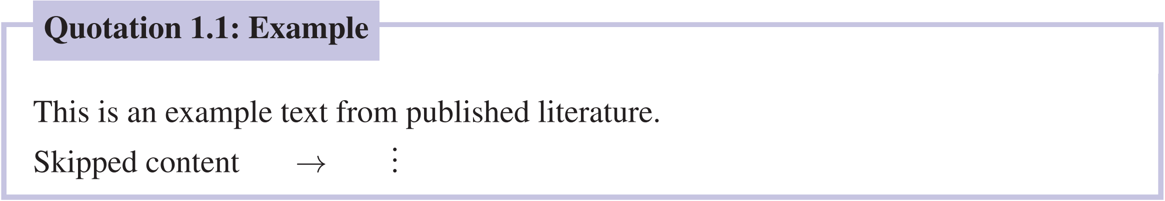
To make the reader familiar with related work, each chapter will provide a background section. Each background section covers topics that are necessary to understand presented methods. Also, literature that is closely related in terms of approach and/or goal are highlighted. In addition, this first chapter provides a high-level background section (Section 1.3) that introduces fundamental topics that are prerequisites for all following chapters.

1.2.2 Verifications and Validations

Most of the following chapters are providing case studies with different types of data. Depending on whether data used for experiments is artificially created or obtained from real measurements, the naming of the experiment is different. In the following, all case studies performed with artificial test data are called "verifications" while experiments with measurement data are called "validations".

1.2.3 Quotations

In this dissertation, excerpts of already published content can be found. For clear separation of previously published content and entirely new written text, long excerpts will be within a quotation box. Three vertical dots symbolize skipped content. Figure 1.3 shows an example for quotation layout. Within quotation boxes, the numbering of figures, tables, etc. is not consistent with the numbering in the main document. Reused figures within newly written text and quotations of definitions are cited in their caption.



Quotation 1.1: Example

This is an example text from published literature.

Skipped content → ⋮

Fig. 1.3 Example Quotation

Abbreviations

Abbreviations used in this dissertation were listed before. At the first time of occurrence, each abbreviation is introduced as a footnote.

Example: i.e.[1]

1.2.4 Definitions

This section defines phrases and expressions that are relevant for this work. Where possible, ISO standards are quoted. For non standardized phrases, the definition provided in the following is used throughout the whole dissertation.

ISO Definitions

Non-rigid Part - translated from DIN EN ISO 10579:2013-11 [10]
"A part that deforms in the free state with an amount that exceeds the dimensional and/or geometrical tolerances defined in the technical drawing"

Free State - translated from DIN EN ISO 10579:2013-11 [10]
"State of a part, where nothing but gravity load caused by its own weight is applied."

Digital Twin - from ISO 23247-1:2021 [11]
"fit for purpose digital representation (3.2.2) of an observable manufacturing element with synchronization between the element and its digital representation"

Digital Representation (3.2.2) - from ISO 23247-1:2021 [11]
"data element representing a set of properties of an observable manufacturing element (3.2.5)"

Observable Manufacturing Element OME (3.2.5) - from ISO 23247-1:2021 [11]
"item that has an observable physical presence or operation in manufacturing.

[1] Latin: "id est"

Note 1 to entry: Observable manufacturing elements include personnel, equipment, material, process, facility, environment, product, and supporting document."

Non Standardized Definitions

GF[1] State - introduced in [2]
"State of a part, where no external load affects its shape."

1.2.5 Wording

As this dissertation is interdisciplinary and related to various fields of research, some words and phrases can be misunderstood as they have different meanings in different fields. In the following, wording for overlapping formulations is clarified.

- Mesh[2] - an object that approximates a geometry by multiple connected elements. Each element consists of nodes, edges and faces.
- Node[2] - a vertex that belongs to a mesh.
- Edge (mesh)[2] - connection between two mesh nodes.
- Edge (sheet metal)[3] - beaded edge of exterior sheet metal component. Gap- and flushness measures are taken at this geometrical feature.
- Perimeter[2] - a closed curve (quantity of edges) along topological boundary.
- Boundary[3] - a constrain for defining an FE-problem.
- Interface[3] - a mechanical connection between two components.
- Nominal Geometry[4] - The ideal, desired geometry, often represented by CAD-definition.
- Actual Geometry[4] - Geometry of an actual part which differs from the nominal geometry. Usually represented as measured point cloud.

[1] Gravity Free
[2] Geometric Modelling
[3] Mechanical Engineering
[4] Metrology

1.3 Fundamentals

In this section the reader is introduced to topics that are fundamental to understand the concepts and contributions that are presented in this dissertation. In particular, the highlighted topics are:

1.3.1 Finite Element Simulation

1.3.2 Alignment Systems

1.3.3 Challenges in Sheet Metal Assembly

1.3.4 Inspection of Free-Form Surfaces

1.3.5 Distance Computations

1.3.6 Digital Twins

1.3.1 Finite Element Simulation

In the following, a brief introduction to the topic Finite Element Simulation is given. For detailed explanations, the book series [12–14] is advised.

Basics

FE-simulation calculates an approximated solution for continuous problems. While the deflection of a loaded beam could be calculated analytically resulting in a function for the occurring deflection based on probing position, mechanical problems for more complex shapes cannot be solved analytically anymore. Therefore, the continuous domain is discretized by a finite number of elements (mesh). Each element can be considered as a standard discrete problem. As elements are connected to neighbor elements, residual forces and strains must be harmonized. This is achieved by assembling individual element stiffness matrices and information of connection to a global stiffness matrix K. This matrix together with boundary conditions (forces f and displacements u), form the mathematical description of the simulation problem, see Equation 1.1. The equilibrium problem is then being computed

$$0 = K\vec{u} + \vec{f} \tag{1.1}$$

Equation 1.1: Formulation of FE-problem. With the stiffness matrix K given, force and displacements are solved.

by a solver. For nonlinear behaviors i.e. orthotropic material properties, stress stiffening, contact interactions, or plasticity, additional terms are added to the stiffness matrix K_{nonlin}. These terms are usually dependent on output quantities x of the simulation problem like element stress, nodal strains, temperature, or velocity, and must be solved iteratively, see Equation 1.2. The size of the simulation problem and the computational cost is mainly driven

$$K = K_{structure} + K_{nonlin}(x) \tag{1.2}$$

Equation 1.2: Nonlinear terms are added to the stiffness matrix K. These terms depend on simulation outputs x.

by the number and type of elements used for modeling the continuous domain. The number and type of elements must be chosen carefully and are usually determined during a mesh sensitivity study, see [15]. With this study, an acceptable trade-off between simulation time and output accuracy is achieved.

Types of Elements

For different purposes, different kinds of elements have been developed. In the following, explanations are related to the commercial FE-software tool ABAQUS which is used in this dissertation as FE-solver. ABAQUS provides 31 different element types. These types of elements can be clustered by: (a) family, (b) DOFs[1], (c) number of nodes and order of interpolation, (d) formulation, and (e) Integration, see [16]. Families of elements are for distinct applications. The most crucial difference between families is geometrical. For instance, volumetric elements are commonly used for representing 3D-solid structures while 1D elements are commonly used for establishing connections between solid components, i.e. springs or rods. Figure 1.4 gives an overview of different element families provided by ABAQUS. The DOF of an element defines how many fundamental variables are calculated for this element during simulation. For instance, a simple 1D element with two nodes in a 2D simulation scenario would have 4 DOFs as for each node, displacements in x and y direction must be calculated. The number of DOFs has a great impact on the simulation complexity. The number of nodes for one type of element can vary. For example, a simple cube can be represented with 8 nodes, ABAQUS also supports 20 node cubes, where an additional node is placed on every edge, see Figure 1.5. These kinds of elements are called second-order elements, as the order of interpolation between nodal values is quadric instead of linear. The formulation of an element can be either Lagrangian or Eulerian, where the Lagrangian formulation describes material properties for the element and calculates actual deformations. The Eulerian description uses elements that are calculating quantities along edges like flows

[1] Degree of Freedom

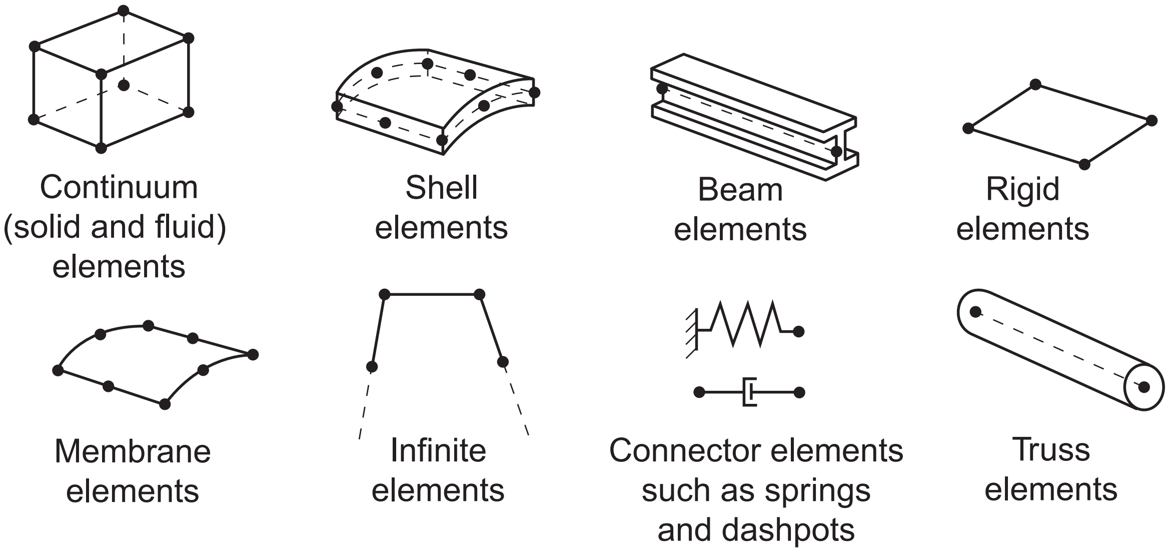


Fig. 1.4 Different families of elements for different purposes that are provided by ABAQUS. (This figure has been published in [16])

of gas, fluids, or electricity. In this case, the mesh nodes remain fixed in position. Last, the integration of an element defines the used technique for integrating nodal values across the volume of the element. For most elements, ABAQUS uses Gaussian quadrature interpolation. Other available integration techniques can be looked up in [16].

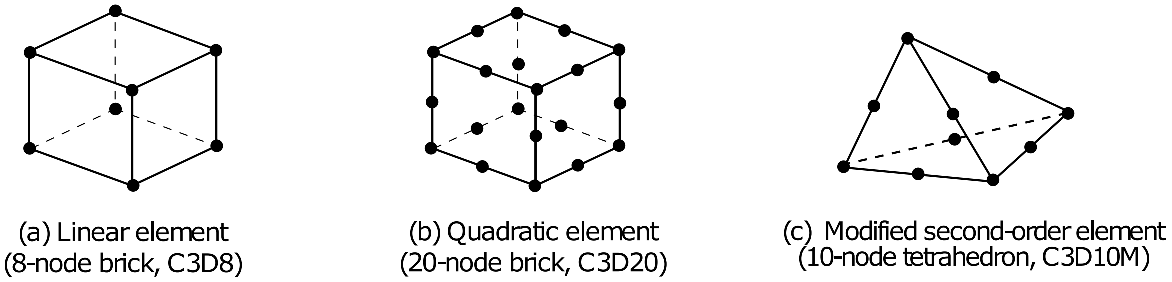


Fig. 1.5 Difference between linear- and quadratic-elements. (This figure has been published in [16])

Applications

Nowadays, FE-Method is the most commonly used simulation technique for numerically solving various mechanical-, electrical-, fluid-dynamics-, problems. The simulation problems described in this dissertation can all be categorized as “quasistatic structural analysis”. So, the following will focus on applications in the mechanical domain. Even within the domain of mechanical analysis a wide variety of applications can be found. For instance, see review papers:

- Manufacturing [17]
- Medical [18],[19]
- Aerospace [20],[21]
- Energy [22]
- Geo-technical [23]
- Automotive [24],[25],[26]

The great flexibility and modeling capabilities of FEM make this method applicable to almost any desired application. In this work, FEM is used as a simulation principle for setting up the DT model of the assembly process.

1.3.2 Alignment Systems

The alignment of an ideally rigid part is carried out with six locators that restrict movement in different directions. By that, the parts' rotational and translational DOFs are defined, and the part is positioned uniquely in a certain coordinate system. However, when working with non-rigid parts, additional DOFs must be taken into account as the part bends under its own weight [27]. This leads to different shapes with different positioning methods. In the following, different alignment methods, for rigid- and non-rigid parts are discussed.

3-2-1 Alignment (Rigid)

A rigid body has six DOFs that need to be constrained. Equation 1.3 [28] lists the equilibrium conditions that must be fulfilled for positioning any part. These conditions say that there must be no resulting force or moment along any parts' axis when summing up all forces and moments applied to the part, compare left picture of Figure 1.6.

To vanish resulting forces and moments by using most simple mechanical boundaries, the positioning scheme "3-2-1 alignment" is commonly used, see right picture of Figure 1.6. The name of this scheme is derived from the number of boundaries being placed in each axis direction: three boundaries applied in one direction (x-direction) are constraining the movement of the part to a plane (yz-plane). Two more boundaries applied in a second direction (z-direction) are constraining the movement to a line. A last single boundary is

$$\sum F_{ix} = 0, \sum F_{iy} = 0, \sum F_{iz} = 0$$
$$\sum M_{ix} = 0, \sum M_{iy} = 0, \sum M_{iz} = 0 \tag{1.3}$$

Equation 1.3: Sum of forces and moments that must be eliminated to position an object in space.

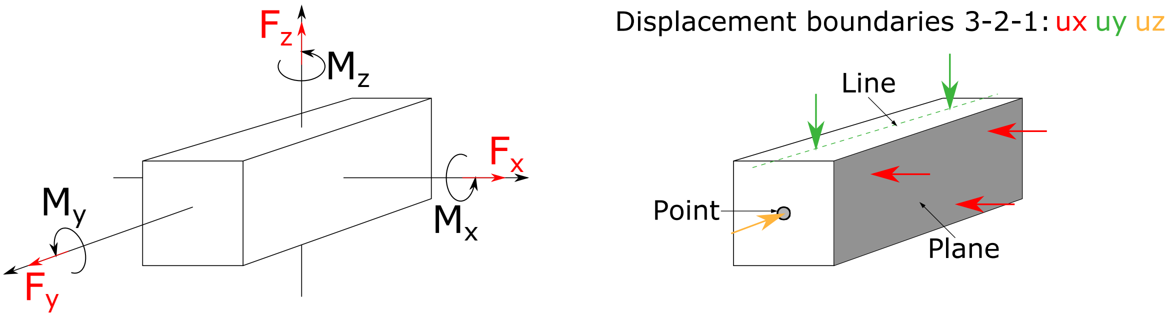


Fig. 1.6 “3-2-1-alignment”. Left: 6 DOFs of a rigid body. Right: placement of 6 boundaries to constrain a rigid body in space.

applied in the third direction (y-direction). By that, the parts’ position in space is defined and no further movement is possible. Remark: To make the alignment system as stable and reproducible as possible, the placement of boundaries per axis should be spread as far apart as possible.

Alignment of Non-Rigid Parts

To align non-rigid parts, more points for positioning are employed. This holds for assembly- but also for inspection- processes. This over-constraining is necessary to achieve a robust and reproducible process. The downside of over-constraining a sheet metal part is that it becomes tensioned during the process, resulting in errors. To deal with this challenge, Volkswagen introduced the RPS[1].

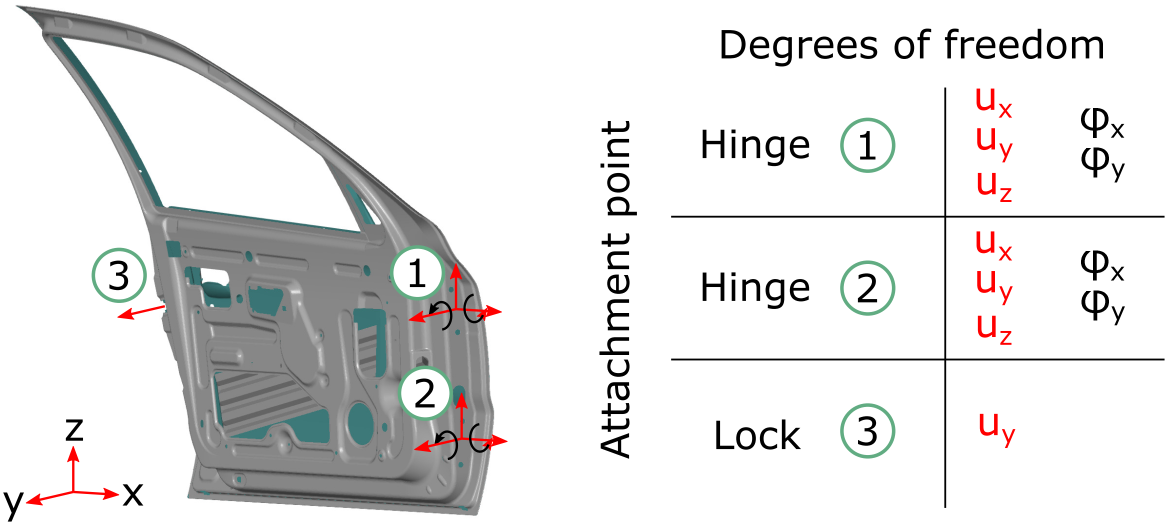
Derived from the concept of 3-2-1 alignment, Volkswagen created an internal norm specific for aligning sheet metal parts, see [29]. This alignment system uses measures of geometrical features, called RPs[2], for positioning. These features can be:

[1] Reference Point System
[2] Reference Points

- Location holes
- Surfaces
- Edges
- Long holes

While holes are used for constraining in two dimensions at once, all other features are only used to constrain one direction at once. This norm is designed to be used for dimensioning, manufacturing, and inspecting of single parts as well as assemblies.

As modern parts become softer and weaker, the RPS system has been extended to also cover the spectrum of non-rigid parts. Therefore, additional RPs are introduced that over-constrain the part. These so-called “secondary RPs” have a lower relevance compared to the “main RPs” from the 3-2-1 alignment system, and are only brought into position if possible. The necessity of such an alignment system is illustrated with an example part from the automotive industry. Figure 1.7 shows a door from a car. The door is attached to its chassis by two hinges and one lock. Both hinges can be adjusted in x-y-z-direction. Additionally, moments in x- and y-direction can be induced. Along the hinge axis, no moments can occur. The lock is designed with a catch hook that can only pull in one direction. This simple attachment layout already results in 11 DOFs. If the part would be ideally rigid, the part could be uniquely assembled by defining 6 of these DOFs. In contrast, if the part has to be considered as non-rigid in all directions, all 11 DOFs must be controlled during assembly



Degrees of freedom

Attachment point		
Hinge 1	u_x u_y u_z	φ_x φ_y
Hinge 2	u_x u_y u_z	φ_x φ_y
Lock 3	u_y	

Fig. 1.7 example for DOFs of a non-rigid part. Example shows a front car door that is attached with two hinges and one lock, resulting in 11 DOFs (listed on the right).

to define a unique assembly state. Two alignment strategies for the example part are shown in Figure 1.8. The left picture shows a classic 3-2-1 alignment using six reference points in different directions. The probing positions are chosen such that a CMM[1] can reach it in assembled state. The choice of RPs is similar to actual alignment strategies used in the automotive industry. In [30] fault patterns for a very similar door assembly are discussed. The picture on the right shows an extended alignment system. Here two additional probing locations (orange) were added. The assumption was made, that the angular displacements of the hinges can not be precisely adjusted and can be neglected. Z_3 was added to control the distance between hinges. By adding Z_3, all displacement boundaries are controlled. Additionally, Y_4 was added. The reason for adding Y_4 is, that this corner of the door is far away from all RPs. Because of that, the whole area near Y_4 already reacts very sensitive to small misplacements of Y_1 and Y_2, see [30]. By adding a secondary RP to this location a more reproducible alignment is achieved without being inconsistent with the main RPS from the left picture.

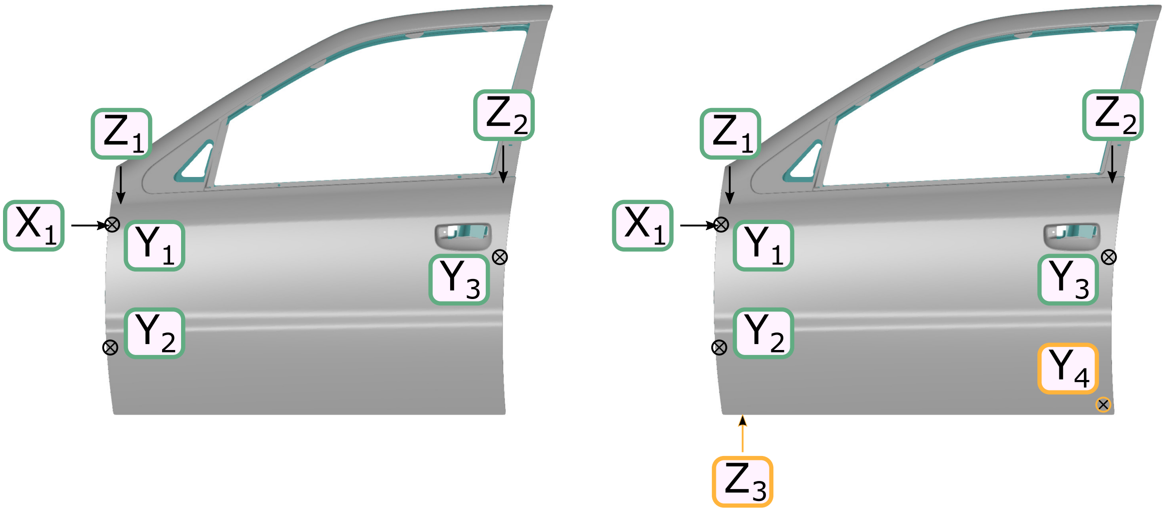


Fig. 1.8 Left: 3-2-1 alignment system for the car door. Shown probing positions relate to real established main-RPs. Right: extended RPS for non-rigid parts, employing Z_3 and Y_4 for additional restrictions.

1.3.3 Challenges in Sheet Metal Assembly

The assembly of sheet metal components is typically carried out as a multistage assembly process. Single part components and/or sub-assemblies are placed on fixtures and are joined together. A broad literature review of processes for sheet metal components can be found in

[1] Coordinate Measurement Machine

[31]. During manufacturing, dimensional variances occur. Identifying which process step causes errors in the final product is a challenging task. As a solution, different methods were developed to tackle this problem. This branch of methods is called RCA[1].

Root Cause Analysis

RCA analyzes possible influence onto the assembly process on a high level. In [32] a guideline and general description for RCA is given. Here, four crucial steps are introduced:

1 Data collection

2 Causal factor charting

3 Root cause identification

4 Recommendation generation and implementation

The concept of RCA is closely related to FMEA[2], see [33], a tool commonly applied in quality management. The principle of RCA or FMEA can be applied to various use cases. In the following, applications in assembly processes are highlighted as these are most relevant for this dissertation. In [34] an RCA-approach is presented that is designed to identify reasons for dimensional variation in assembled components. Different modeling approaches try to estimate variation propagation in multi-stage assembly processes, see [35, 36], to improve anomaly detection. More recently, approaches employ state of the art ML[3] methods and statistical models to identify fault patterns in multi-stage sheet metal assembly processes, see [37, 38].

RCA is mainly intended to identify weak spots of systems or processes. Optimizing identified weak points is partially also covered by the literature. Especially in sheet metal assembly, the placement of locators and clamping points is known as a sensible factor.

Optimization of Locators, Clamping points and Joints

Making sheet metal assembly processes robust and reliable is challenging. A crucial factor is the positioning and clamping of components during the assembly. As sheet metal parts are considered non-rigid, the position and number of locators and clamps can be optimized for minimizing deflection. To analyze and tackle this problem a whole branch of research has been established. A selection of publications regarding sheet metal assemblies: [39–42, 41, 43].

[1] Root Cause Analysis

[2] Failure Mode and Effects Analysis

[3] Machine Learning

1.3.4 Inspection of Free-Form Surfaces

Coordinate Measurement Machines

Exterior car body components or fuselage parts of airplanes are designed with large free-form surfaces. Assessing the geometrical correctness of such parts is a challenging task, and has been carried out by using CMMs. In [44] a review of CMMs and their use is provided. As CMMs can only acquire a restricted amount of points within an available time-frame [45] describes the usage of portable measuring arms, especially for an in-line 100% inspection of production steps in Body-In-White. In general, using tactile measurement systems is appropriate for probing a small number of points, but recognizing a free-form surface with a high resolution is complicated and time-consuming with CMMs. To overcome this shortcoming adaptive sampling methods were developed to deal with free-form surfaces, see [46]. However, recognizing a free-form surface with a CMM remains a time-consuming process and is also hard to implement for in-line purposes. Also, point-wise measurements only provide limited insight into a surface's characteristics.

Optical Measurement Machines

In modern manufacturing systems, the inspection of free-form surfaces is carried out with optical metrology systems, often referred to as 3D-scanners. 3D-scanners can capture several millions of vertices within seconds. Reviews of 3D-scanning systems and their applications are given in [47–49]. 3D-scan systems are very flexible and can be used to capture parts in various sizes. For instance, a review of large size applications (i.e. fuselage assembly of airplanes) can be found in [50] while also various small sized applications like producing tooth implants in medical applications can be supported with 3D-scanning, see [51]. In nowadays sheet metal assembly processes, quality inspection is carried out more and more by employing 3D-scanning technology. However, the implementation of 3D-scanning to inspection processes of sheet metals entails different challenges. The reflective surface of the metal disturbs the measurement process and makes it difficult to capture the whole part. To address this challenge, several works were published to optimize the viewpoints for capturing, while minimizing reflections, see [52, 53].

Further, challenging aspects occur when integrating 3D-scanning for in-line purposes. Therefore, requirements in terms of time and data processing must be considered. To speed up measurement process and data acquisition, [54] investigates the application of LIDAR[1] systems for in-line usage. Another challenge is to ensure coverage, especially for complex parts.

[1] Light Detection and Ranging

Not accessible regions or sharp edges are hard to capture and might take a long time and good preparations of the scan object. In [55] a method is proposed that drastically reduces the necessary amount of scan windows, by a least-squares reconstruction of not measured areas.

Fringe Projection Profilometry

The principle used by the scanning system that is employed in the following, is called structured light scan or FPP[1]. FFP is briefly explained using Figure 1.9 for details the book [56] is advised. FPP uses different patterns (commonly stripes with different spacing) that are projected onto the specimen while the cameras capture the object. The stripes are distorted by the object. This distortion is captured by the camera. During the measurement, the projected stripes must be distinguishable to be identified in the picture. To make single stripes distinguishable different approaches are available, see [57]. Within the picture, points can be identified and their position is calculated by using triangulation. Therefore, the distance between projector and camera ($\overline{AB}$) must be constant. Also, orientation between camera and projector must be fixed during scanning. Depending on the pixel location of a registered point, a ray from the focal point of the camera through the pixel is calculated. Together with intrinsic and extrinsic calibration results, the position in space is obtained. By projecting different patterns in horizontal and vertical direction, a cloud of points can be generated from the captured pictures. The accuracy of this scan principle can be improved by using a second camera that employs a third point of reference to the system.

Measurement Fixtures

Similar to the positioning problem during assembly, the component being assessed must be positioned in a reproducible way during the scan. Rigs equipped with locators and clamps are used to ensure such a robust measurement process. Analogous optimization methods (compare Section 1.3.3) are used to minimize fixture influence and parts deflection caused by weight, during measurement. Industrial measurement systems in sheet metal applications are capable of achieving measurement errors below ± 0.05 mm. This value sets the target for verifications and validations of methods that are presented in this dissertation.

1.3.5 Distance Computations

For evaluating a 3D-scanned surface, the acquired point cloud is usually compared to the desired shape. Therefore, the point cloud is post-processed by filtering steps like removing

[1] Fringe Projection Profilometry

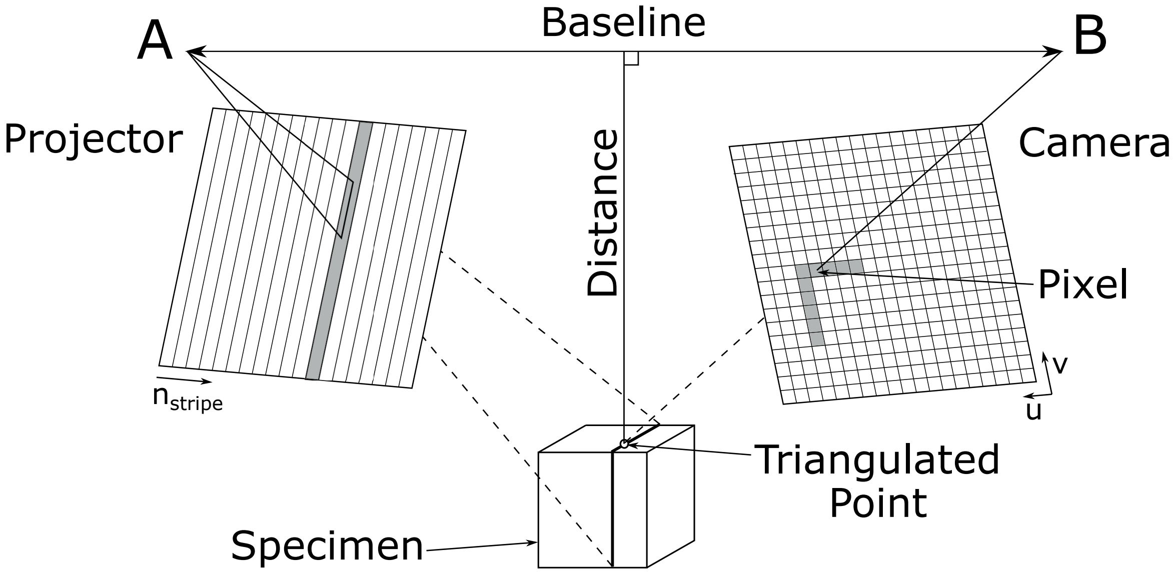


Fig. 1.9 Principle of FPP: projector at position A is projecting a pattern onto the specimen. The pattern is distorted by the specimen which is captured by the camera from point B. By matching stripe ID and position in camera pixel layout, the point on the stripe can be triangulated.

outliers and smoothing. Then, a mesh representation is obtained by tessellating the clean point cloud. The resulting mesh is used to calculate a mesh-to-mesh distance between measurement and desired geometry. The calculation of such distance is not straightforward and can be carried out with different metrics which are discussed in the following using Figure 1.10.

Cloud to Cloud Distance

The cloud to cloud distance is a nearest neighbor search that finds the closest point of the target point cloud for each vertex of the source point cloud. The found distance vectors can not be used further as directions might be chaotic and highly dependent on mesh resolutions. However, the length of each vector gives a first local approximation of the distance between the source- and target point cloud. Unfortunately, this method is not very accurate as mesh resolution and choice of source- and target mesh has a high impact on the results.

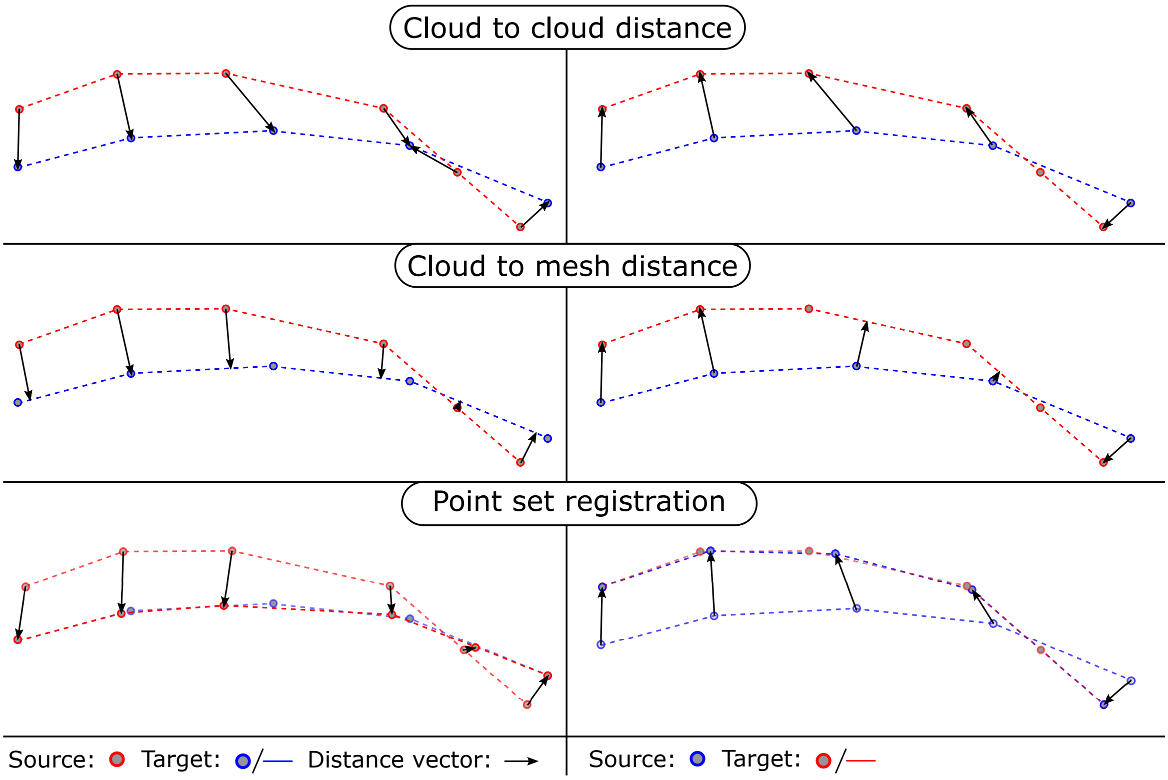


Fig. 1.10 Comparison of different metrics for distance computations. Left/right: switched source and target point cloud.

Cloud to Mesh Distance

This distance metric calculates the shortest distance of each source mesh node to the surface of the target mesh. The distance vectors are perpendicular to the target surface. This metric is commonly used by inspection software as it can be implemented very efficiently and delivers fast, robust, and smooth distance computations. However, this metric also shows inconsistencies when switching target- and source mesh. Also, the alignment of point clouds highly influences the results. The computed vectors are commonly not used for assessment as they are hard to visualize and only point in normal direction.

Point Set Registration

A point set registration method is an algorithm that computes a transformation - rigid or non-rigid - to map one point cloud to another. This branch of methods is commonly used for aligning and merging multiple point clouds in 3D-scanning. A collection of implemented methods can be found in [58]. A huge benefit of point set registration is, that distorted point clouds can still be matched very reliably. Because of that, modern inspection techniques for sheet metal are using point set registration as distance metric, see [59, 60]. Distance metrics like cloud to cloud or cloud to mesh would result in wrong results for areas with high deflection or distortion. Found distance vectors computed with point set registration can be used for further calculations as they point onto the correct geometric position on the target mesh. The downside of using a point set registration method for distance calculations is that the computational cost is tremendously higher compared with cloud to mesh metrics. Especially, for high-resolution point clouds with millions of vertices, the computing time is not acceptable for everyday usage.

For this dissertation, a custom implementation of cloud to mesh distance metric is used which will be described after this background section.

1.3.6 Digital Twin

A DT describes a virtual representation of a physical object, process, or system. A DT can be for instance CAD models, simulation models, or even mathematical models. What makes these models a DT is a synchronization between the physical counterpart and the model. Due to this synchronization, the DT is updated with real conditions. These conditions are sensor data or conditions measured at the physical twin. Based on the DT enriched with field data, precise predictions for the physical counterpart can be made. The concept of DT is an upcoming field of research and also gains interest in industrial applications. A widely spread example is predictive maintenance, where a DT is employed to define urgency for maintenance based on conditions of the physical system, see [61]. Although the concept of a DT is intuitive, the phrase "Digital Twin" had no standardized definition for a long time. Recently, the ISO standard 23247-1:2021 [11], introduced a definition for a DT for the context of manufacturing processes. In this section, no comprehensive literature review for DTs is provided, because the concept of using sensor data to improve simulation models goes much deeper. Different work between 1980 and 2000 can be found that would nowadays be classified as DT. For instance (not exhaustive), the following keywords apply:

- Parameter Identification
- Online-/Inline-Simulation
- Reverse Engineering
- Predictive Maintenance
- Closed Loop Simulation

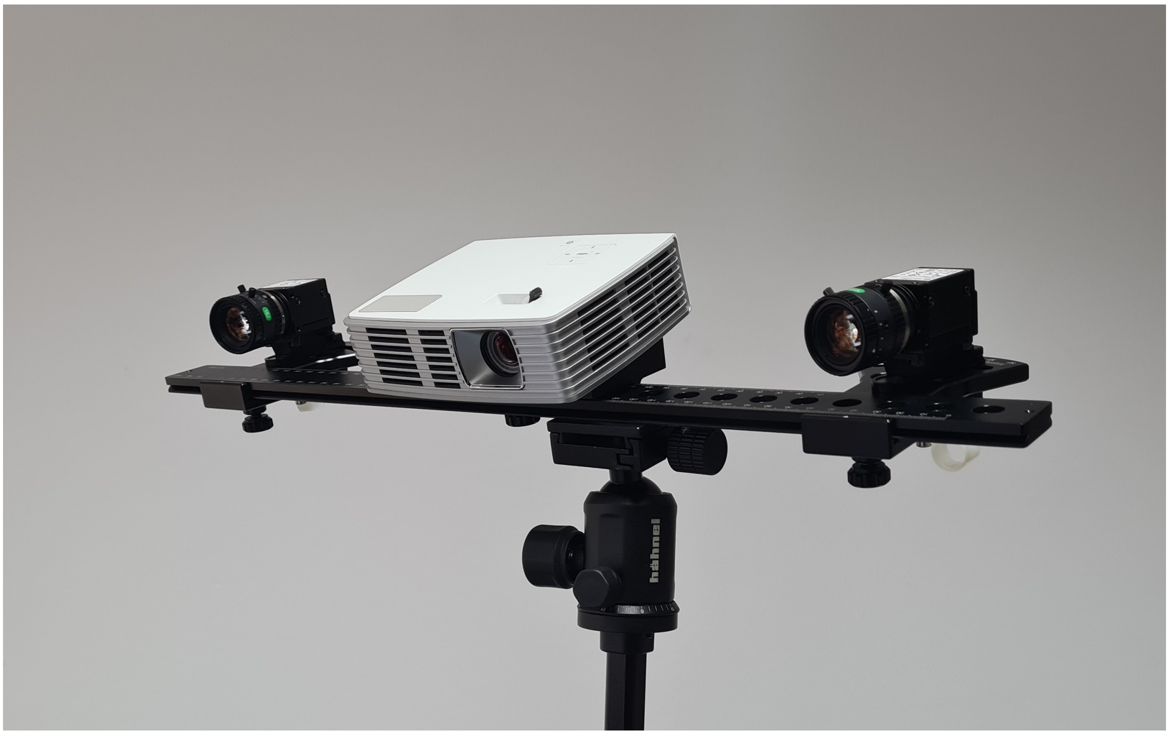


Fig. 1.11 3D-scanner: “HP 3D Structured Light Scanner PRO S3” with dual camera upgrade kit.

1.4 Introduction to Employed 3D-Scanning System

For performing experiments with real measurement data, a 3D-scanner is employed. The 3D-scanning system used in this dissertation is the “3D Structured Light Scanner PRO S3” from the manufacturer “HP”, see Figure 1.11. The scanning system is composed of two industrial-grade cameras and a 1080p projector with a special lens, that enables focusing on extremely close distances. As the name of the product already reveals, the system uses structured light for 3D-scanning. The same principle is also used in industrial applications for measuring sheet metals, see [62].

In the following, setup and calibration of the scan system, custom implementation of distance metric, custom colormapping, and assessment of the measurement noise and precision of the scan system, are presented.

1.4.1 Setup and Calibration

FPP systems can be calibrated with various methods, see [63]. In the following used setup procedure and calibration method are described. Based on the distance to the specimen

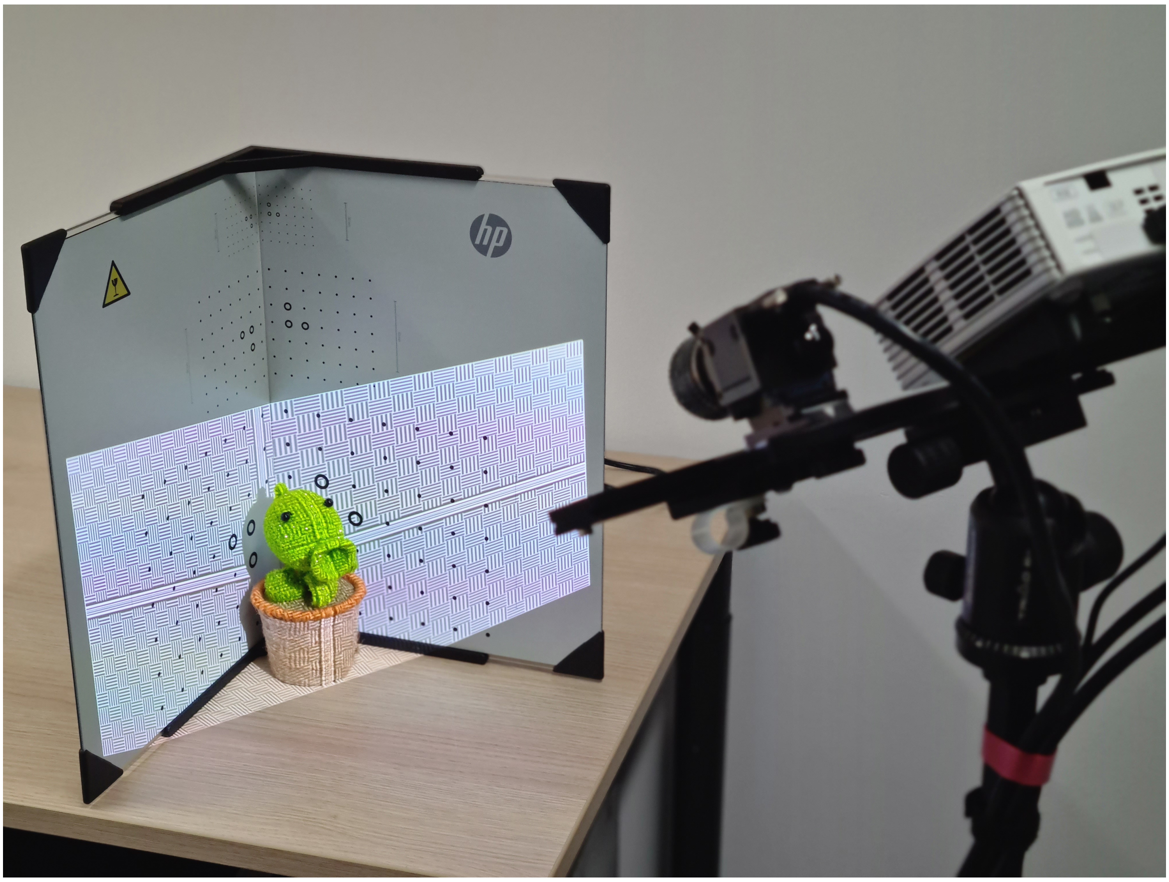


Fig. 1.12 The size of the calibration pattern is chosen according to the size of the object being scanned and the used scan window.

and the size of the specimen, the scan setup has to be adjusted. Accordingly, the distance between cameras and projector as well as camera angle and focus of optics is tuned in. Also, exposure time, aperture of the cameras, and brightness of the projector must be adjusted depending on lighting conditions. The intrinsic and extrinsic calibration is carried out using reference planes. These planes are realized with glass panels that have a printed surface with a grid of markers. With glass, an almost perfectly flat surface can be achieved with less effort compared to expensive machining processes. The two glass panels are set up with a 90-degree angle and build the coordinate system, the scanner is calibrated to. This coordinate system is used to define the positions of triangulated points during the scan process. The size of the grid and reference planes are chosen according to the size of the scan window, see Figure 1.12. With everything set, the calibration can be performed. Unfortunately, there are no mathematical details provided by the manufacturer of how, and what kind of calibration is implemented in the HP scan software. Quotation 1.2 gives the available information from HP Customer Support regarding the calibration process.

> **Quotation 1.2: Calibration Description of Manufacturer [64]**
>
> During calibration, the software first measures the position, orientation, focal length, and distortion characteristics of the camera. Then a pattern sequence is projected to measure the same optical characteristics of the projector. For color texturing, a white balance is also performed.

1.4.2 Custom Implemented Distance Metric

To unify all visualized data, the implementation of the distance metric used in this dissertation is described. In the following, all mesh-to-mesh comparisons are based on this distance metric. As input for the computations serves two triangulated surface meshes. The principle of the comparison is explained using Figure 1.13. The metric composes of two steps, (a) finding the closest point on the target mesh for every vertex of the source mesh and (b) filtering found matches with angle criteria to remove miss-matches. For the first step, the library "Trimesh" [65] is used. This library has a built-in function that takes two triangulated meshes as input and outputs nearest point on target mesh, triangle ID, and a scalar value for the distance, for each vertex of the source mesh. Unfortunately, the closest point between two meshes is not in general the desired distance. In areas with high curvature or near intersections miss-matches might occur. To prevent such errors, the output of the "Trimesh"

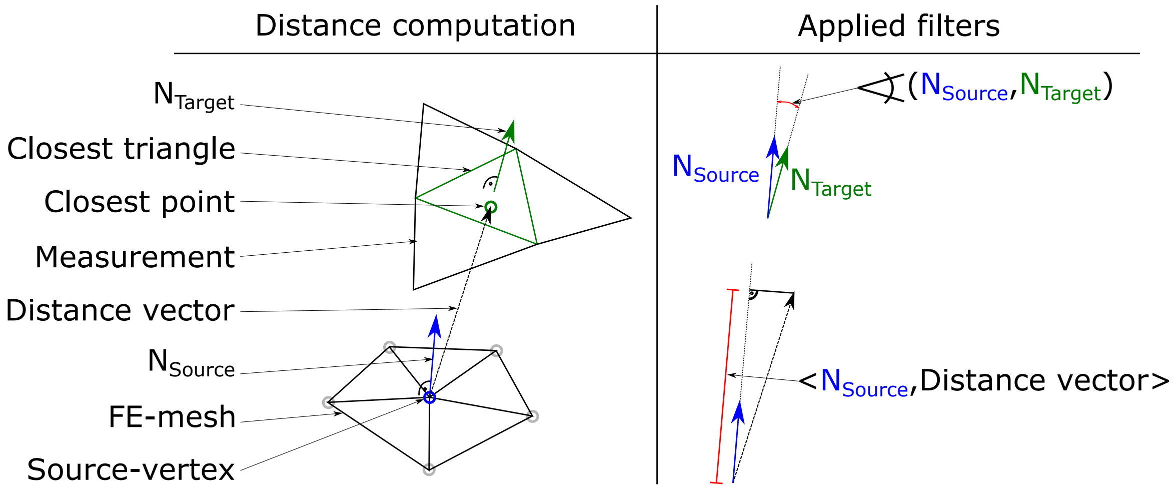


Fig. 1.13 Left: distance computation implemented in "Trimesh" and output quantities. Right: applied normal filter and calculating distance along source vertex normal. (This figure has been published in [6])

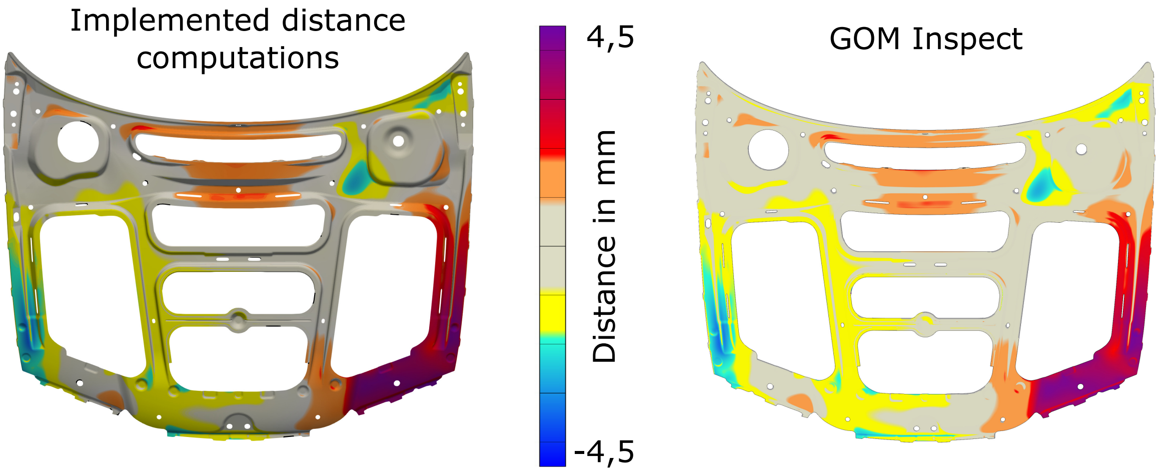


Fig. 1.14 Comparison between self implemented distance computation (left) with state of the art commercial software solution “GOM Inspect” (right). The shown part is a single sheet metal part which belongs to the substructure of an engine hood.

distance computation is filtered additionally. The implementation uses a maximum distance threshold combined with a simple normal filter that compares source vertex normal and matched triangle normal. If the distance or angle between normals exceeds user-defined values, the found closest node is classified A second filtering step is applied to all remaining matches. In this step, the dot product between the source mesh vertex normal and the distance vector is calculated. By that, only the portion of distance along source vertex normal is considered. This is important to ensure comparability as commonly used 3D-scanners are only able to observe differences in surface normal direction. The resulting distance computation is a vertex-based scalar field defined on the source mesh grid. To verify the implemented distance computations we compare the results with the state-of-the-art analysis software “GOM Inspect”. An example is shown in Figure 1.14. Both pictures show the same mesh comparison. On the left, the result of the self-implemented distance computation is rendered with “Paraview”[66]. On the right, the computed distance from “GOM Inspect” is shown. The user-defined filter parameters were set to 10 mm maximum search distance and 10° maximum allowed difference in normal angle. These standard values are used for all mesh comparisons in this dissertation. The results of both methods are displayed on the same range and colormap. Besides differences in lighting and shading, no deviations can be observed. The reason for choosing a self-implemented distance metric is to have access to nodal values and full control of the performed filtering.

1.4.3 Colormapping

Methods in this dissertation are validated with simulated and experimental data. For visualization two colormaps for different purposes are introduced - see Figure 1.15. The continuous colormap (top of figure) is chosen for symmetrical value ranges in the positive and negative direction. For values in between ± 20%, a grey color together with an opacity of 50% is chosen to weaken the perception of the areas with values near zero. An intensive color is chosen for the value ranges of above ± 20%. At ± 40% value range a second distinct step in color with a smooth transition towards ± 100% is used. The purpose of this colormap is to visualize and highlight differences in scalar fields where the areas of interest are non-zero values.

The second colormap (bottom of Figure 1.15) is a discrete 10 step color mapping. This map is designed to visualize percentage values on a 3D-domain. The colors are chosen such that value ranges can be identified easily. Especially, colors for values above 50% are chosen extra distinguishable. Values below 50% are weakened by the opacity mapping to highlight values above 50% even more.

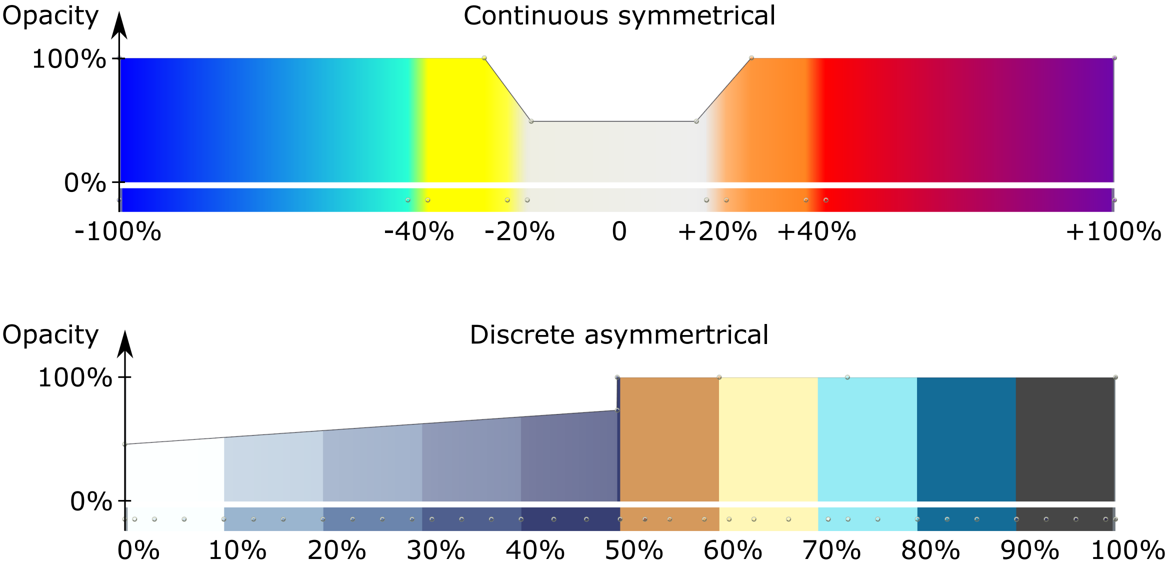


Fig. 1.15 Colormapping for data visualization in this dissertation. Top: continuous colormap for symmetrical value ranges; highlighting for nonzero values. Bottom: discrete colormapping in 10 steps for visualizing percentage values.

1.4.4 Measurement Noise and Precision of Scan System

The HP scan system is rated to errors below 0.05 % of the scan size and below ± 0.05 mm in the best case. To verify whether the scan system meets precision requirements for experimental validation, different characteristics of the scanner were investigated. This verification was carried out by scanning the calibration panels and assessing a local snippet (60mm x 60mm) of the scan. The scan software provides functions for automatically post-processing the acquired point cloud before saving. To avoid obfuscation due to excessive filtering, only outlier removal and quality checks are used, but no additional smoothing of the point cloud was performed. From the resulting point cloud, the local snipped is cut out and investigated. As a reference for comparison, a plane is fitted to the snipped. The required precision for assessing sheet metal are errors below ± 0.05 mm. The value of 0.05mm was chosen because scan systems commonly used for assessing sheet metals are also specified for the same precision.

The results of the verification are shown in Figure 1.16. The color scale is chosen such that everything below ± 0.05 mm is grey colored. The results show the noise pattern of the scan system. It can be observed that the noise pattern has a high frequency that can be filtered without losing relevant details. To analyze the differences in detail, next to the colormap a distribution of error values and statistical quantities are shown. The distribution of values is symmetrical and similar to a Gauss distribution. This suspects, that errors are randomly distributed and systematic errors are not relevant for this inspection. To get a better impression of this distribution, min, max, span, and sigma (σ) are calculated and shown. The span has a value of 0.52mm which is very large compared to the target span of 0.1 mm (± 0.05mm). However, in the colored picture, it can be seen that only single peaks of the noise are reaching that high. These peaks can be filtered out by smoothing the point cloud. The sigma value is 0.037mm, so roundabout 68% of all error values are within ± sigma. By applying strong smoothing to the mesh snipped, statistical values can be improved, resulting in span=0.06mm and sigma=0.005mm. These values sound very promising but must be looked at with caution. For ideal capturing conditions, the calibration panels are printed with a reflective color which is meant to be projected on. When using the scan system under non-ideal conditions, like capturing dark or shiny objectives in bright environments, the scan process is disturbed and precision is affected as well.

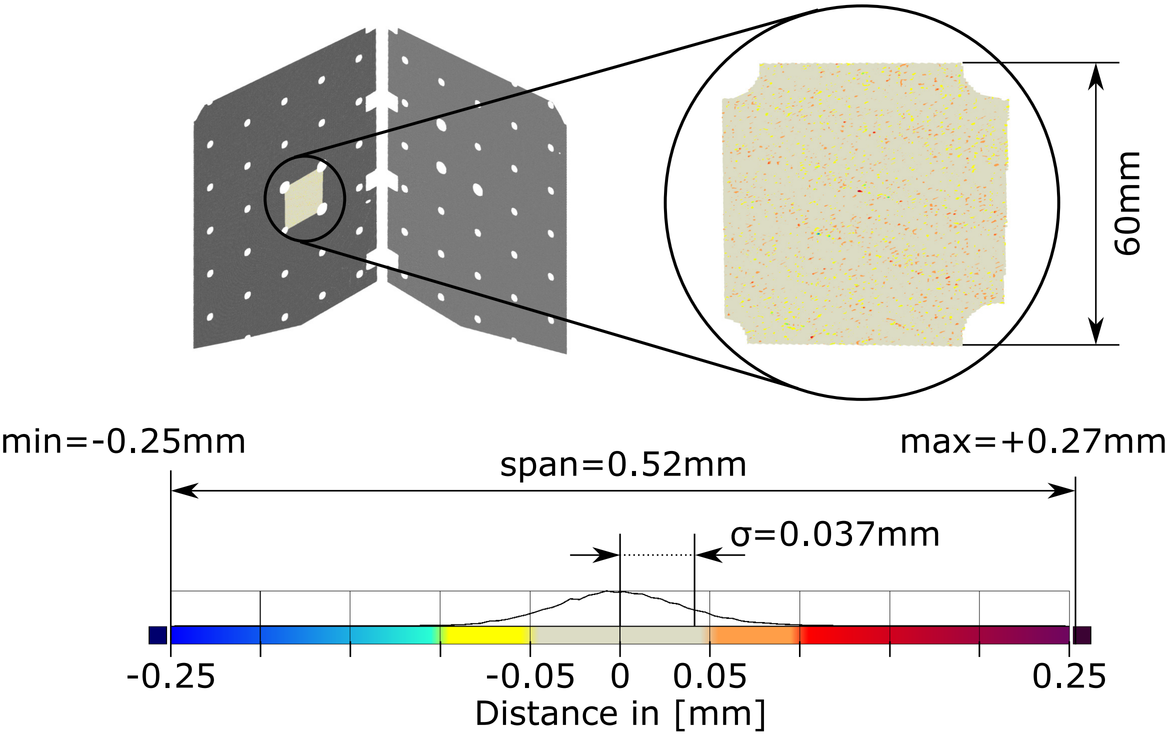


Fig. 1.16 Investigation of measurement noise: calibration panels are scanned and local scan snipped is evaluated. Statistical properties are shown next to colormap.

1.5 Summary of Research Focus

In the following, a DT-based approach for self-compensating assembly processes for thin, non-rigid parts is described. This research focuses on generating DTs from optical measurement data of free-form surfaces. The so-called “geometrical Digital Twin” is employed to make precise predictions for the best individual assembly parameters. Throughout the following chapters, different challenges for the realization of such concepts are identified and tackled. These challenges are spread across different fields of research:

- Manufacturing
- Finite Element Simulation
- Metrology
- Geometric Modeling
- Automotive / Aerospace Industry

Besides providing solutions for identified problems, the applicability to real production environments is discussed. The structure of chapters is closely related to the workflow described in Section 1.1: Chapter 2 does discuss a framework for generating DTs for smart assembly purposes. From this framework challenges and requirements for realizing a DT-based smart assembly system are derived. Chapter 3 does provide a method to correctly post-process 3D-scanned non-rigid components. Next, Chapter 4 shows a method to automatically generate geometrical DTs from scan data. In Chapter 5 a software tool is described that supports troubleshooting processes for DT-based smart assembly systems. The whole dissertation is discussed from the perspectives of different fields of research in Chapter 6 and is concluded in Chapter 7.

Chapter 2

Framework for Predicting Optimal Assembly Parameters

2.1 Introduction

This chapter is based on the core reference [1] which describes a framework for setting up a predictive simulation model for assembly processes. A concept description and preliminary studies that underline applicability for use cases from the automotive industry are provided. In particular, the assembly of exterior car components made from sheet metal is discussed. This process has strict quality goals regarding gap and flushness measures. To meet requirements from different stakeholders, an automotive manufacturer needs constantly research and develop new approaches to produce vehicles. Regarding chassis components, legal requirements for human- and crash-safety must be met. At the same time, requirements in terms of weight and cost must be fulfilled while maintaining aesthetic requirements. To improve crash properties and reduce weight, car components are made from thinner and thus softer sheet metals or polymers. By using thinner materials, the stiffness of the components is affected. This leads to non-rigid behavior which has a great impact on the robustness of manufacturing-, assembly-, and inspection-processes. For the final assembly of exterior car components, important quality measures are gap and flushness. These measures have tight tolerances that are commonly below manufacturing variabilities of single components. To tune in gap and flushness all exterior parts have adjustable mechanical interfaces. To achieve reproducible quality with an automated process, the placement of the interfaces must be chosen individually before fastening. To deal with the new challenges that occur during the assembly of non-rigid parts, several approaches have been developed. One branch of methods is called “smart assembly” which are computer-supported assembly processes.

Smart assembly processes could either be predictive (estimation-based) or compensating (deriving countermeasures from already produced assemblies). One predictive kind of smart assembly employs a surrogate model that is capable of predicting best fitting assembly parameters based on measured component characteristics, in real time. In this chapter, a surrogate model is presented for the application of assembling exterior car body parts. Benefits but also challenges that arise when implementing a predictive surrogate model to manufacturing processes are discussed.

This chapter frames the dissertation by providing a problem description and partial solutions on a higher level. Shown use cases and results are from preliminary studies and serve as input for a discussion to identify general challenges. These challenges are grouped into topics that are tackled in the subsequent chapters and topics that need to be solved outside the scope of this dissertation. This chapter is structured as follows: Section 2.2 provides an overview of related literature and state-of-the-art approaches. Section 2.3 is an excerpt of the Method from [1], from which requirements for the smart assembly system are derived in Section 2.4. Section 2.5 concludes this chapter and structures the subsequent chapters based on the identified requirements.

2.2 Background

2.2.1 Gap and Flushness Quality

Gap and flushness are critical quality measures in the assembly of exterior car components [67], fuselage panels in aerospace industry [68], household appliance [69], smartphones [70], etc. Especially, for exterior components, gap and flushness are crucial for the aesthetics of the final product. At the same time, these measures are hard to control due to tight tolerances which makes assembly challenging and costly. Figure 2.1 shows a cross-section of the border area of two parts illustrating gap and flushness measures. In car body assembly, gap and flushness are not unambiguously defined. Depending on viewing angle, gap and also flushness can be perceived differently by the customer. Therefore, the desired measures might vary depending on position and viewing angle. For this dissertation, gap and flushness are defined as follows:

- **Gap:** Shortest distance between two parts measured tangential to outer (visible) surface normal.
- **Flush:** Distance between points at the end of each edge radius, measured perpendicular to outer (visible) surface normal.

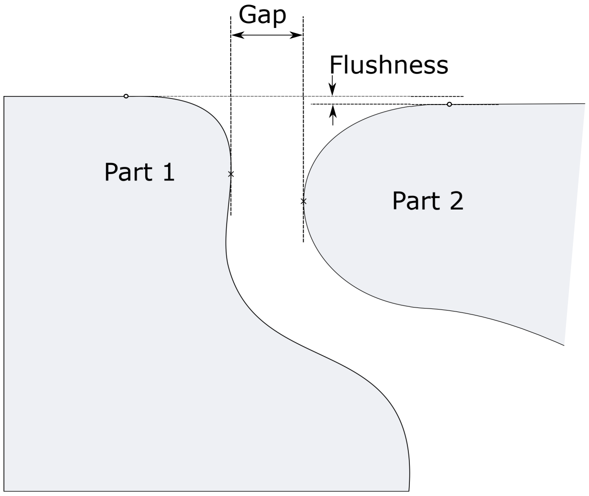


Fig. 2.1 The cross section illustrates the border area between two parts. Measures of gap and flushness between the parts are shown. (This figure has been published in [1])

The measurement of gap and flushness was traditionally carried out with gauges of different sizes that were stuck into the gap. Recent research focuses on developing non-contact measurement methods to improve precision and sampling rate while reducing the time needed for taking measures, see [67, 68, 71, 72].

2.2.2 Best Fit Assembly

Best fit assembly uses an adaptive positioning system that is controlled according to measurement data. This data could either be acquired before - or also during performing the actual assembly. In aerospace industry, this principle is used to assemble large components like fuselage parts, see [73, 74]. In the automotive industry, a common application is the assembly of doors to a chassis. In this kind of process, handling tools equipped with sensors measure gap and flushness in real time at multiple locations. The alignment of components is improved in real time as well, see [75]. However, best-fit assembly systems, in general, do not consider the behavior of the parts being assembled. By that, predictive capabilities are restricted to corrections in placement and orientation.

2.2.3 Smart Assembly

Smart assembly is an abstract phrase for different kinds of computer-supported assembly processes. This phrase is used in conjunction with multiple different assembly problems, which are highlighted briefly in the following. In [76] an AR[1] approach is presented that

[1] Augmented Reality

supports workers for manual assembly processes with a high product variety to reduce errors. In [77] a human-computer interaction approach is chosen to enable cooperation between workers and the equipment to reduce assembly time. In [78] cameras of different types are used for detecting objects and their orientation in the workspace for automated assembly. In [79] material handling with AVGs[1] is optimized to balance robotic assembly lines. Regarding sheet metal assemblies, [80] introduced an approach for individualizing locator positions in combination with selective assembly to improve overall assembly quality.

This wide variety of applications shows the broadness of the topic smart assembly. To narrow down the topic and bring it in relation to this chapter, important aspects of smart assembly are highlighted. To make assembly processes smart, different subsystems are implied: (a) recognition system, (b) prediction system, (c) action system, see Figure 2.2. The recognition system is needed to capture the current state of the environment and components. This system is related to data acquisition and contains all measurement and capture devices. The prediction system uses the results from the recognition system as input for estimating optimal actions. These actions are performed by the action system, which can be manual or automated.

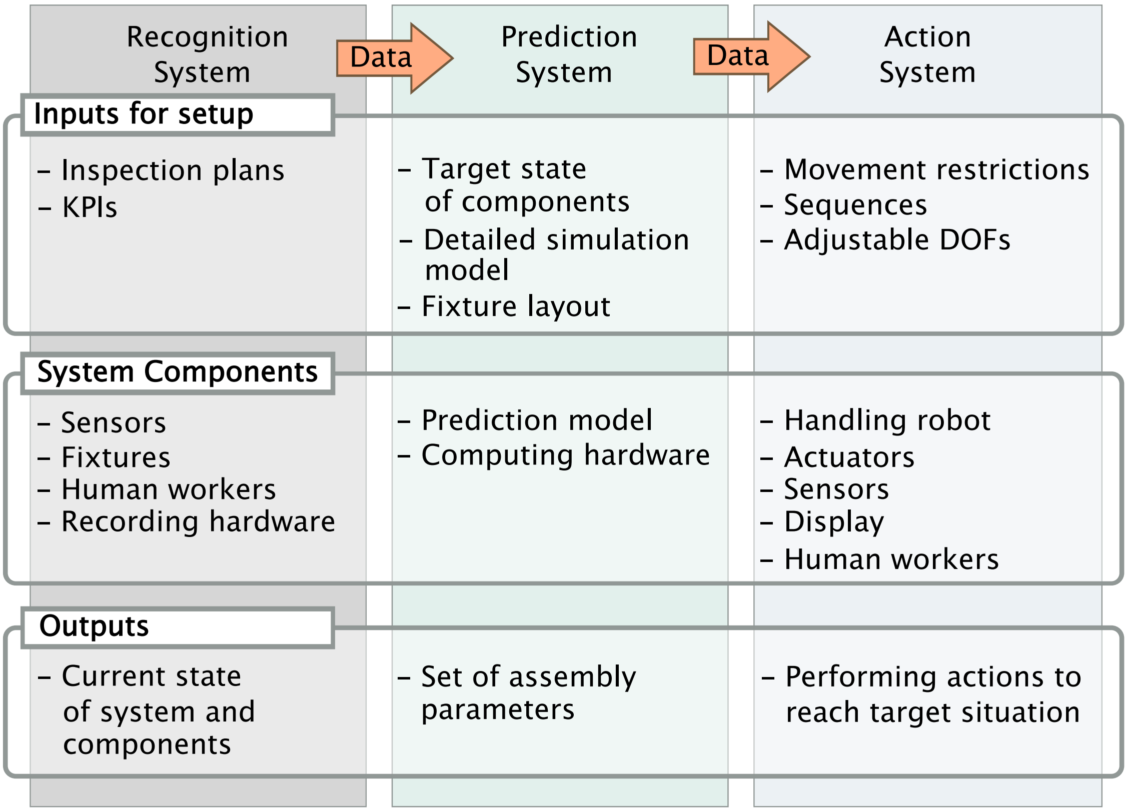
2.2.4 Online Simulation

Online simulation is any kind of simulation that is performed during a physical process. The simulation results are needed to control the physical process. For instance, any kind of prediction system for smart assembly needs to make decisions based on current sensor states. Therefore, commonly simulation models with real-time capabilities are used, see [81–86]. Recently, the phrase “online simulation” is being replaced and/or mixed up with “digital twin”. For clarification, the differences between these two expressions are highlighted. As defined in Section 1.2.4, a DT is a simulation model of a physical twin that has been enriched with sensor data from the real object. In contrast, an online simulation could also be an abstract simulation model supporting the process instead of mapping the physical process to a virtual representation.

2.2.5 Surrogate models

High-fidelity simulation models are capable of providing accurate results. However, run times are often far away from being applicable to real-time applications. To make simulations capable of supporting processes in real time, surrogate models are employed which provide

[1] Automated Guided Vehicles



	Recognition System	Prediction System	Action System
Inputs for setup	– Inspection plans – KPIs	– Target state of components – Detailed simulation model – Fixture layout	– Movement restrictions – Sequences – Adjustable DOFs
System Components	– Sensors – Fixtures – Human workers – Recording hardware	– Prediction model – Computing hardware	– Handling robot – Actuators – Sensors – Display – Human workers
Outputs	– Current state of system and components	– Set of assembly parameters	– Performing actions to reach target situation

Fig. 2.2 Subsystems of a smart assembly process control

only an approximated solution but very fast. These models are often based on big data sets of already simulated data, and approximate solutions in-between. Common methods for modeling a surrogate model are:

- Response surface [87]
- Kringing [88]
- Radial Basis Functions [89]
- Support Vector Machines [90]
- Artificial Neural Networks [91]

A general review for surrogate models in robust design optimizations can be found in [92].

Superelement Method

The principle used in the following for surrogate modeling, is called SE[1]-method. SE is a model reduction technique that is related to substructuring and is applied to FE-simulation models. By condensing the global stiffness matrix only results for pre-defined mesh nodes are calculated. This reduces the computing time tremendously. SE method is widely used for various applications, for instance, see [93–96].

The principle behind SE modeling is explained using Figure 2.3. In the figure, a 1D-mesh is shown, represented by three mesh nodes, n_{1-3} and four mesh elements, represented by the springs c_{1-4}. Each mesh node has one DOF - u_{1-3}. Assuming that one is only interested in the movement of n_2, SE-modeling can be applied. To achieve the substructure shown in the lower picture, an equivalent spring stiffness can be computed for c_1/c_2 and also for c_3/c_4, resulting in two SEs - SE_1 and SE_2 - that are sufficient for describing the movement u_2. Compared to previously listed surrogate models, the SE method does not need a pre-computed ensemble of simulation results. However, substructuring an FE-mesh has also shortcomings. For instance, the equivalent stiffness matrix is pre-calculated once and remains unchanged when used in FE-simulations. So, the stiffness in between SE nodes is not updated during simulation and can only reflect linear behaviors. Any kind of geometric non-linearity that is dependent on interim simulation results, can not be considered with a SE model.

[1] Super Element

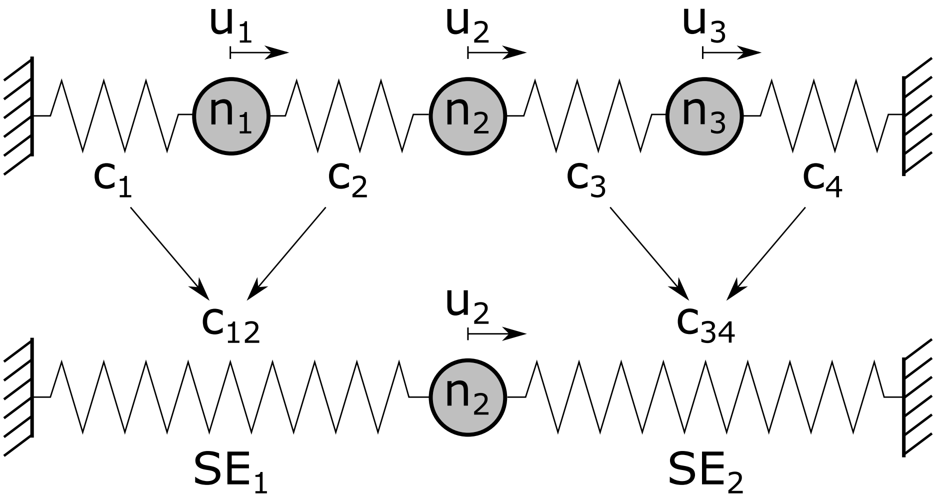


Fig. 2.3 Principle of super element modeling

2.2.6 Resume

To put the content of this chapter into perspective with the presented related work, a short resume follows. Gap and flushness are important for products' aesthetics and function. These measures are highly affected by assembly quality. To improve this quality, adaptive processes like best-fit assembly and smart assembly recently become popular. During this kind of computer-supported process, predictions based on sensor data from the physical process, are made. This concept is called online simulation. To support processes in real time, online simulation models must meet requirements in terms of computational time and output precision. To achieve real-time support, surrogate models are employed that provide an approximated solution of a high-fidelity simulation model with low computational effort.

In the following, a framework for smart assembly to improve the gap and flushness quality of exterior car components is described. An online simulation is achieved by using SE-method as a surrogate model to approximate solutions from an FE-simulation model.

2.3 Method and Use Case Scenario

The methodology and use case scenario are excerpted from the paper "Online Simulation Considering Production Uncertainties to Improve Assembly Quality" [1]. For consistency with introduced distance computations and color mapping (compare Sections 1.4.3 and 1.4.2), all figures that feature a color map are reevaluated and shown again after the excerpt.

Quotation 2.1: Chapter 3 & 4 excerpted from [1]

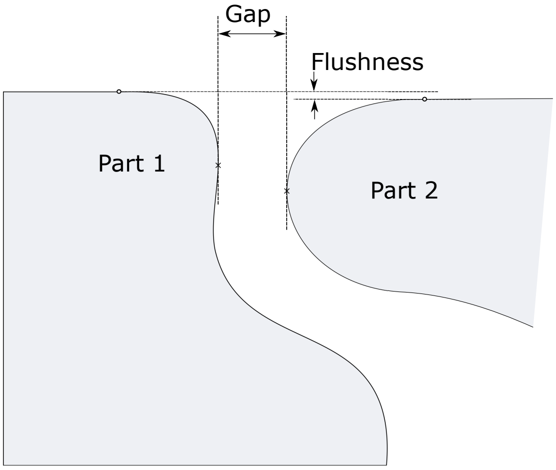


Figure 1: Crossection showing gap- and flushness-measurements

⋮

3. Method

In this section the optimization problem is described and the overall concept is sketched while clustering it into sub problems. For all sub problems a possible solution will be proposed.

Optimization Problem

To get a better understanding of how the quality of gaps and flushness can be improved by adjusting the assembly process for every part individually, the optimization problem that has to be solved with the simulation is explained.

Every ridged body has six Degrees Of Freedom (DOFs), three transitionally DOFs and three rotatory DOFs. In contrast, a deformable body has as many DOFs as the DOFs of its boundary conditions but at least six DOFs. For instance, we can consider a simple hood of a car as a deformable body. The hood is usually attached to the chassis by two hinges and one lock. Often additionally two or more adjustment puffers are attached to the hood. We consider now two hinges with five DOFs each, one lock with one DOF and two adjustment buffers with one DOF each. Overall, we have 13 DOFs which are partially coupled through the stiffness of the part. Beside the boundary conditions, loads are acting onto the part caused by peripheral elements like gas springs or sealings. These loads are depended of the geometries of the part itself and its assembly environment and they are also depended of the relative position between part and environment. These loads cause a deflection of the hood during assembly. All the boundaries listed before are adjustable within a certain tolerance. Changing the positions of the boundaries the shape, position, and orientation of the part can be adjusted. Every combination of positions and orientations of the 13 DOFs causes a unique shape of the part. To control these DOFs during an automated assembly process, the robot grabs the part with a tool that constrains the coupled DOFs with the clamps of the grabbing tool. Then the robot positions the part relative to the chassis before it gets fastened. After the tool releases the part, the final position and shape is achieved. The robot cannot directly grab the part at the actual fastening position. So the robot uses different areas to grab the part leading to an indirect positioning of the boundary.

With changing the position and orientation of the part accordingly to the measured geometries before fastening, a best-fit assembly can be performed. By predicting the behaviour of the part during assembly by simulations, the positioning of the part can be improved. Also the positions of the clamps of the grabbing tool and with that the pretension of the part before fastening, could be adjusted adaptively to archive a better fitting shape of the parts. This leads to the first optimization problem that has to be solved. The first optimization problem is to find the best combination of the clamping positions and the initial positioning of the part before fastening. This problem can be defined according to the preferences of the user by e.g. weighting critical areas higher than others.

By considering the geometries of the part and its assembly environment, a second optimization problem can be defined. The measured geometries can be arranged in different combinations. For every combination of part and chassis the first optimization problem (finding the best parameters for assembly) can be solved. For instance, we consider five measurements of chassis and five measurements of hoods that will be assembled to the chassis. So there are 5*5=25 different combinations of chassis and hoods. The idea is to choose the five combinations that lead to the best overall quality.

Overall Concept Steps

The basic idea of this approach is to use an FEM-Simulation to predict the behaviour of parts during assembly for production control. Figure 2 shows the detailed workflow of the production control concept.

The first step ”Preparation/Simulation Setup” has only to be performed once. In this step the FE-simulation-model for the online simulation is set up. To be able to predict the geometric properties on a scale beneath millimetres, the actual geometry of the assembling part as well as the assembly environment hast to be taken into account, thus, a measurement of both geometries is needed. These are analysed in the next step - ”Measurement/Simulation” - to get the relevant information for the FE-Simulation. After this post processing an individual FE-model for the measured part can be created. With this model an optimization problem for the assembly process can be set up and solved. In the last step ”Assembly” the simulation result from the previous step is analysed to derive individual assembly parameter sets for the automated assembly. Furthermore, the concept is extended to optimize different available combinations of parts and assembly environments to archive the best overall quality.

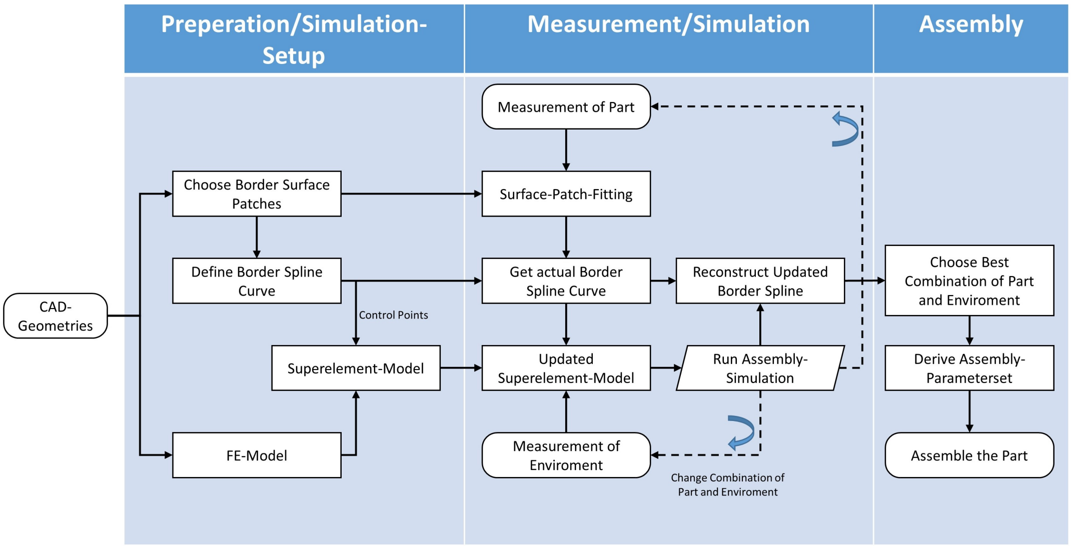

Preperation/Simulation-Setup
Measurement/Simulation
Assembly
Measurement of Part
Choose Border Surface Patches
Surface-Patch-Fitting
Define Border Spline Curve
Get actual Border Spline Curve
Reconstruct Updated Border Spline
Choose Best Combination of Part and Enviroment
CAD-Geometries
Control Points
Superelement-Model
Updated Superelement-Model
Run Assembly-Simulation
Derive Assembly-Parameterset
FE-Model
Measurement of Enviroment
Assemble the Part
Change Combination of Part and Enviroment

Figure 2: Overall workflow of the concept

The three steps are now explained in detail:

Preparation/Simulation Setup

The first step serves to prepare the simulation setup. This has to be done only once for different measurements. As input for this step, CAD-geometries of the part and its environment (surrounding parts) are necessary. From these geometries, a full detailed FE-model can be created considering the behaviour of the assembling part as well as the interactions with its environment. The FE-model has to be on a high level of detail to recognize deviations on a small scale beneath 0,1 millimetres. The CAD-geometry of the part is also used to define the areas of interest. These are surface patches along the gap to its neighbour part covering the whole border area. Figure 3 shows an example for the selection of surface patches in the border area (black patches covering the edge and surrounding area). Onto this surface patches a 3D-spline can be defined that follows the tip of the edge radius so that every point of the spline matches the measurement point of the gap (compare Figure 1). This spline is used to condense the representation of the geometry to a 3D-curve and is called hereafter ”contour spline”. The number of control points needed for the contour spline depends on the curvature of the geometry. In areas of high curvature more control points are needed to ensure a sufficient representation of the geometry. This kind of representation is also used for modal tolerance analysis as presented by (Adragna et al. 2007).

With the control points of the contour spline and the full detailed FE-model a SE-model can be created. A SE basically connects a set of boundary nodes chosen by the user, with a stiffness accordingly to the stiffness of the FE-model. The reduction is similar to the principle of finding an equivalent spring stiffness of a multiple spring system. In this case, the boundary nodes of the SE are chosen near to the control points of the contour spline. By linking the control points of the contour spline to the SE-model the whole information of the border area of the part can be reconstructed for the updated situation. The reconstruction of the whole border area presumes that the radius of the border area does not change during assembly. As mentioned before, all the tasks within this steps have only to be performed once. With these preparations done, the simulation based production control can be performed.

Measurement/Simulation

In the second step ”Measurement/Simulation” the actual online simulation is performed. Here, also the second optimization problem, finding the best combination of available chassis and parts, is solved. The idea is to use measurements of parts and chassis that will be available at the same time at the assembly line as input for this step. For now, we assume optical measurements of the border areas as well as optical measurements of the relevant environment (border areas of the neighbours, contact surfaces, etc.).

With the optical measurement of the part and the CAD-surface-patches of the border area a surface patch fitting can be performed. In this procedure the parametric surface is adjusted iteratively until it matches the point cloud of the measurement within a certain rage. This method is also called morphing. Note that the surface patches are not only become adjusted in term of position and orientation also the shape has to be adjusted to fit the point cloud. After fitting the surface patches to the measurement, a continuous representation of the actual border area is obtained. With the representation of the actual geometry the contour spline defined in section “Preparation” can be updated to an actual border spline. After the actual contour spline is obtained, the boundary nodes of the SE-model created in section “Preparation” can be updated with the control points from the updated contour spline. At this step the assumption is made, that the stiffness’s between the boundary nodes remain the same even if the actual geometry is slightly different compared to the CAD. With the measurement of the assembly environment, the boundary conditions of the SE-model can be updated. This is necessary because the assembly environment has influence on the behaviour of the part during assembly. For example, the force of a sealing between two parts depends on the relative position between the parts. With the SE-model containing the actual geometric properties the simulation can be performed.

After the simulation is done the result only contains the new position of the boundary nodes of the SE. To obtain a continuous representation of the border area an updated spline is reconstructed from the simulation result. To measure any point near the reconstructed border spline we assume that the edge radius of the part does not change significantly during assembly. With that assumption a crossection at any spline position can be calculated from the fitted surface patches.

Accordingly, the tasks presented in this step can be performed onto different combinations of parts and chassis. The idea is to use measurements of parts and chassis that will be available at the same time at the assembly line. The output of this section is a collection of simulation results that contains different possible combinations of parts and chassis.

Assembly

The last step is to derive assembly parameter sets for the automated assembly from the simulation results. Therefore, the positioning of the part and the initial deformation caused by the clamping tool before fastening the screws is needed. These can be obtained from the simulation input file of the last iteration of the optimization problem (see section ”Measurement/Simulation”).

4. Use case

To get a better understanding of the concept idea, this section presents an example for an arbitrary use case. For this purposes an open source CAD-model of a car front door is used.

Preparation / Simulation Setup

As initial situation we presume the scenario that we are responsible for the assembly quality of a car door for a new vehicle. We already have the CAD-model of the door and know the geometry of the tool that will be used for the automated assembly process. From the CAD, we can define surface patches at the areas of interest along the gap to the neighbour parts. In the case of the front door, the neighbours are: the back door, the wing, and the side part. In Figure 3 an example for the selection of the surfaces patches is shown for the front door. We also need to define surface patches for the neighbour parts to be able to obtain a continues representation of the area along the gap of both partners. Both representations are needed for performing gap and flushness measures. Next, we can define 3D-contour-splines on the selected surface patches of the door along the different gaps (see Figure 4).

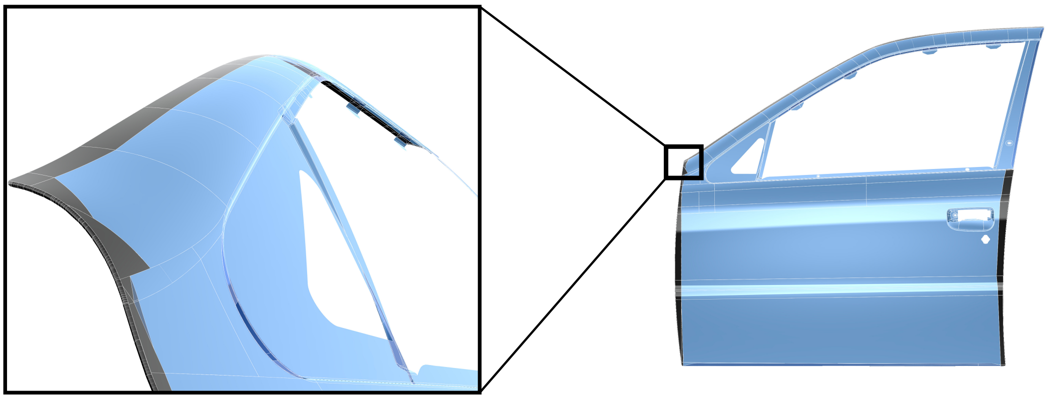


Figure 3: Definition of CAD-surface-patches in the border areas

With the CAD we can prepare beside the geometrical aspects also the structural information. These are obtained by creating a full detailed FE-model of the door considering material behaviour, thicknesses, joints between components, sealings, etc. From the detailed FE- model, we can derive a SE-model. This step presumes that the detailed FE-model is already validated. For the SE, we need to define a set of boundary nodes, that remain in the stiffness matrix. These boundary nodes are close to the position of the control points of the contour spline and also containing the interface nodes for applying forces and boundaries. This step can be scripted in a way that the user merely defines the contour spline.

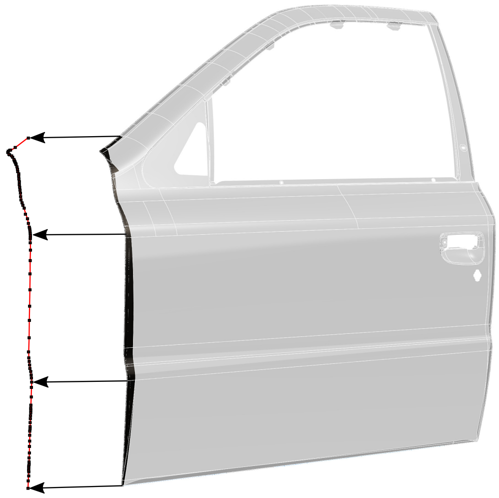
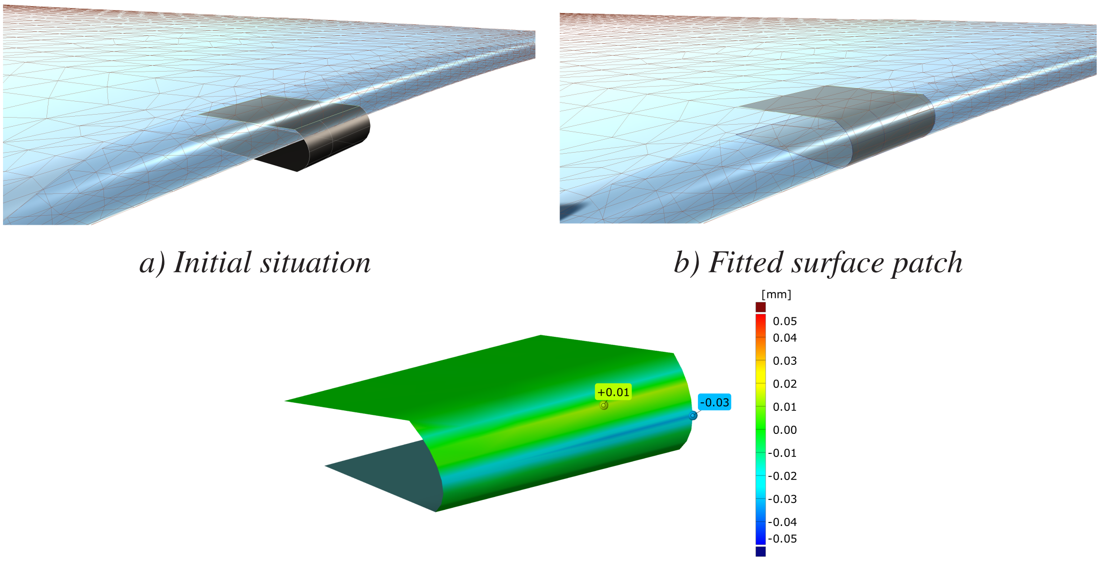


Figure 4: Extracted 3D-contour-spline from previous selected CAD surfaces

Measurement/Simulation

After the preparations are done, the SE, surface patches, and contour splines can be used for the actual process control. We assume, that we have five measurements of already produced doors as well as the measurements of five chassis. First, we can now fit the surface patches of the door and of the environment onto the measurements. By that we get the continues representations of the actual border areas of the door and its neighbours. Figure 5 shows an example for the fitting of a surface patch (software used for surface fitting: Autodesk Alias). Here the initial position a), the fitted result b), and the deviation between surface fitting and measurement c), is shown. Following, we can update the contour splines of the door to five sets of the actual contour splines.

Subsequent, we can update the SE-model created from CAD by adjusting the position of the boundary nodes accordingly to the five actual contour splines. By that we derive five different SE-models for the measured doors. Together with the continuous representations of the five chassis 25 simulations can be performed, each optimizing the gap and flushness measurements by adjusting the boundary conditions (compare section "Optimization Problem"). With the simulation results, the combination of chassis and doors are chosen by optimizing the overall product quality.



a) Initial situation

b) Fitted surface patch

c) Surface deviation of fitted patch

Figure 5: Surface patch fitting

Assembly

In the last concept step the actual assembly is performed. As input we get the collection of simulation results from the step above, for this example the results of the 25 combinations that have been simulated. Out of this collection, a combination is selected that fulfils best the overall quality criteria defined by the user. In the case of automotive exterior assembly, the quality criteria are gap- and flushness-measurements as well as weighted criteria for design important gaps or design lines. These criteria are already considered as target for the first optimization problem but have to be evaluated in the simulation results. After a combination is chosen, a parameter set for the automated assembly can be derived. Therefore, the SE-model contains boundary nodes at the clamping points of the handling tool. With the information where these points must be relative to the chassis before the part becomes fastened, an assembly parameter set can be created. Succeeding, the assembly process can be performed.

⋮

8. References

Adragna, Pierre-Antoine; Samper, Serge; Formosa, Fabien; Pillet, Maurice (2007): Modal Tolerancing — Application to Gap and Flush Analyses. In Serge Tichkiewitch, M. Tollenaere, P. Ray (Eds.): Advances in integrated design and manufacturing in mechanical engineering II. Dordrecht: Springer, pp. 417–430.

Re-Evaluation of Color-Coded Figures

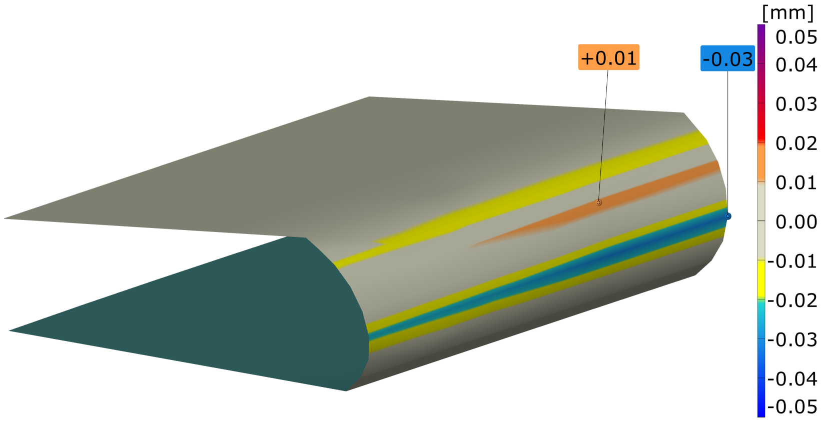


Fig. 2.4 Reevaluation of Figure 5c shown in Quotation 2.1.

For the presented approach, preliminary studies were performed that can be looked up in [1]. The results are discussed in detail in the original paper and underline the usability of the proposed method for the presented application.

2.4 Requirements for Applying DTs to Smart Assembly

The approach presented in Section 2.3 was developed for setting up a smart assembly process for exterior sheet metal components from the automotive industry. As prediction kernel, a SE-model coupled with a linear optimization algorithm is used to predict optimal positioning before fastening the mechanical interfaces of an assembly component. Based on the presented approach, requirements and challenges for applying a DT for smart assembly are identified and discussed. The requirements are grouped by the subsystems of a smart assembly system, which were introduced in Section 2.2.3.

2.4.1 Requirements for the Recognition System

The recognition system was only superficially addressed in [1] and is now being discussed more in-depth. As shown in Figure 2.2, the recognition system is composed of different kinds of sensors. For completeness, also human workers are listed as part of system components if measures are taken manually. The outputs of the recognition system are measurement data from the sensors which describe the current state of assembly- and/or part components. This measurement data serves as input for the prediction system and needs to fulfill certain requirements. To set up a recognition system, different requirements must be addressed.

Inspection plan and KPIs are defining what must be inspected during the current state of production. Also, requirements of the prediction system in terms of quality and completeness of data must be considered. Further, restrictions in terms of timings must be met to ensure target throughput times. These requirements have a great impact on the choice of sensors and peripheral measurement equipment.

Besides measurement hardware, the recognition system also must be capable of handling the accruing measurement data. For instance, high-density point clouds acquired with 3D-scanners, can easily exceed gigabytes within seconds. To record and post-process acquired data, on-site computational hardware is needed. As raw measurement data can not be processed by the prediction system, acquired data must be prepared. This kind of pre-/post-processing computations could for example be: filtering, coarsening, smoothing, calculation of indirectly measured quantities, tessellation, triangulation, etc. The processed output of the recognition system is transferred to the prediction system.

Special Demands for Sheet Metals

A 3D-scanned geometry of a sheet metal part can be used for generating a DT to make a more precise prediction of the following assembly steps. When setting up a recognition system for smart assembly of sheet metal components involving a DT, special requirements must be addressed. As discussed in Section 1.3 sheet metal components and assemblies are considered as non-rigid and consist of large free-form surfaces. Capturing geometries (current state) of these kinds of components is challenging in terms of fixturing and data acquisition, see Section 1.3.2 and Section 1.3.4. Although solutions for data acquisition and fixturing have been discussed, the acquired geometry is subjected to systematic and statistical errors. Statistical errors for instance are positioning errors or measurement noise. These kinds of errors can be addressed by optimizing fixtures and measurement systems. A crucial systematic error with a high impact on the DT, is the deflection of components due to gravity, during measurement. This deflection can be different for every individual part and must be compensated before the measurement can be used by the prediction system.

2.4.2 Requirements for the Prediction System

The prediction system does use a prediction kernel to estimate optimal action that must be performed. This optimal action is calculated based on the current state (provided by the recognition system) of the assembly system and a pre-defined target state. Regarding components, the prediction system is only composed of computing hardware to run the prediction model and the model itself. The computations performed by the prediction system could be either performed on on-site or off-site hardware. For off-site calculations the network needs to meet requirements to be able of handling data flows in an acceptable time. To set up the prediction system different inputs are necessary. First, the target state of the system must be defined. The target state, for instance, can be derived from the nominal positions of KPIs. Also, adaptive targets can be defined like gap and flushness measures based on actual geometries of assembly partners. Besides the target definitions, the prediction model must be set up. Commonly this is done by deriving the prediction model from a well-known and validated simulation model. To complement the prediction model, the fixture layout of the action system is used to define boundary conditions.

The prediction system presented in Section 2.3 is based on a prediction model that uses SE for performing computations in real time. For this model, only single points from the measurement are considered by applying offsets to the optimization problem. The properties of the actual geometry of the measured part are not considered. This causes systematic discrepancies in terms of parts' stiffness. If one has the goal to implement a DT as prediction kernel, actual geometric properties must be taken into account. To achieve this, the DT has to predict correct part behavior considering the actual geometric stiffness of each individual component. At the same time, the prediction system must maintain real-time support.

2.4.3 Requirements for the Action System

The action system composes of all hardware components that are performing the actual assembly, with parameters that are provided by the prediction system. To set up an action system, movement restrictions, adjustable DOFs, and assembly sequences must be defined. For smart assembly, the hardware needs to be adaptively adjustable as every component gets individual assembly instructions from the prediction system. With a handling robot, adaptive positioning of parts relative to the assembly partner can be easily realized. Unfortunately, by adaptive positioning, only 6 DOFs can be adjusted. For non-rigid parts additional DOFs must be controlled during assembly, compare Section "Optimization Problem" of Quotation 2.1. To make more DOFs adjustable, flexible handling tools are necessary that can pre-tension the part in the desired manner by changing positions of clamping points. Figure 2.5 shows a possible layout for clamping points of the front door. Designing flexible positioning devices that can be adjusted during the process has been a topic of research for the past decades and is still an ongoing research topic, see [97–99]. Realizing reliable and precisely functional, flexible assembly tooling is still challenging and also costly. In this dissertation, no aspects regarding the action system are addressed.

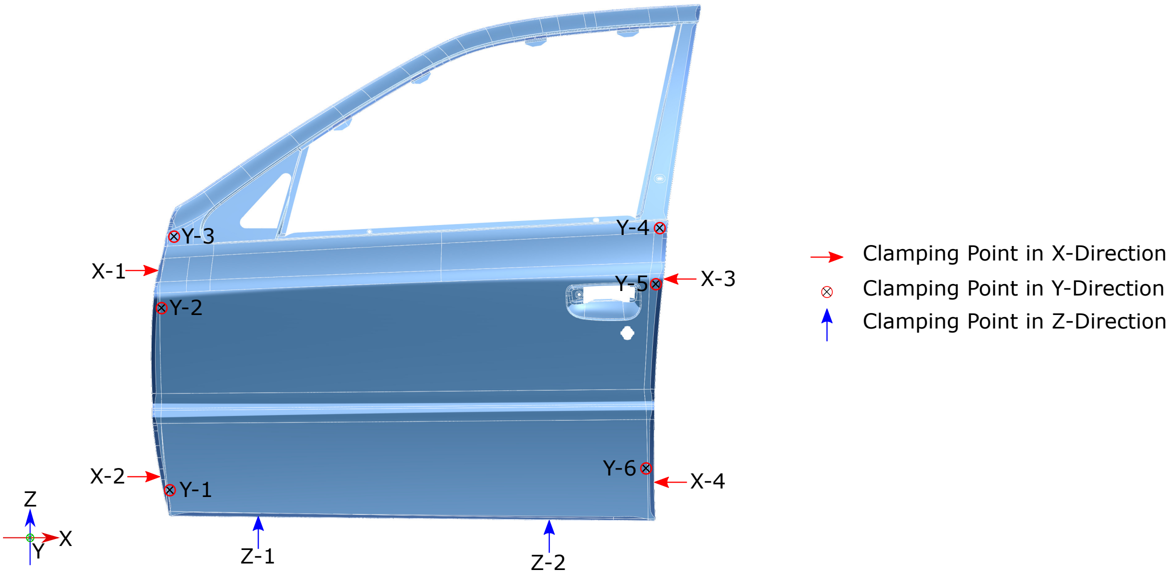

Y-3
Y-4
X-1
Y-5
X-3
Y-2
Z
X
Y
X-2
Y-1
Y-6
X-4
Z-1
Z-2
→ Clamping Point in X-Direction
⊗ Clamping Point in Y-Direction
↑ Clamping Point in Z-Direction

Fig. 2.5 Possible placement of locators for assembly process. For smart assembly purposes each locator must be individually adjustable. (This figure has been published in [1])

2.4.4 Verification, Validation, and Troubleshooting

During implementation but also during operation, the proper function of the smart assembly system must be ensured. To do so, verifications and validations are performed. Verifications are applied to every subsystem separately to ensure each system works properly. These verification steps are performed before the system is applied to a productive environment. During validation, the whole smart assembly system is tested under realistic circumstances. To check whether everything works properly, the predictions made with the prediction system can be compared to a post-assembly measurement. Sources of discrepancies are identified and corrections to the responsible system are made. This step is crucial for applying a smart assembly system to real environments. Especially, when applying a DT as prediction kernel, the DT itself can be adjusted and developed further to improve process robustness and quality. The process of verification, validation, and troubleshooting requires an expert or group of experts. To support experts during these processes, computer-aided methods can be employed.

2.4.5 Requirements on Factory Level

Outside the scope of the smart assembly system itself, different requirements on the factory level must be addressed. These requirements mainly concern the digital infrastructure.

Especially, if decentralizing all three subsystems of smart assembly process control, data management becomes very important and challenging. Depending on the actual setup, latency, bandwidth, databases, reliability, redundant connections, and server architectures must be considered. Smart assembly could also be decentralized further by involving suppliers for components being assembled. For instance, the manufacturing and measuring of the car door could be sourced out to a supplier. In this case, further requirements regarding supply management and data transfer must be considered. When using a DT for smart assembly, the kind of requirements on the factory level remains the same. However, the amount of data being transferred and processed is significantly larger compared to traditional methods.

2.5 Conclusions and Scope of Work

In this chapter, a smart assembly approach that is based on the SE method was presented. From this method, requirements for applying a DT to smart assembly were derived. The found requirements can be divided into requirements for each subsystem of smart assembly and requirements on the factory level. In the following, challenges that are tackled in the subsequent chapters are highlighted and tasks that are outside the scope of this work are pointed out.

Figure 2.6 summarizes the requirements/challenges that need to be addressed when implementing a DT-based smart assembly system. The requirements/challenges are divided by the introduced system levels. In red, challenges where this work contributes are highlighted. The system-levels “Factory level” and “Action system” are out of scope for this work and are not addressed in this dissertation.

The following chapters are providing approaches to overcome challenges arising in: (a) recognition system, (b) prediction system, and (c) verification, validation, troubleshooting. To improve data acquisition, the acquired measurement data needs to be post-processed by the recognition system to compensate for undesired deflections. A method solving this problem, as well as verifications and validations are presented in Chapter 3. The prediction system could be carried out in many different ways. Chapter 4 provides an automated workflow to create a high-fidelity DT for prediction purposes. Last, this dissertation contributes in terms of verification, validation, and troubleshooting aspects for creating DTs based on 3D-scan data. Therefore, two approaches that are designed to support an inspection engineer during decision-making processes are combined into a software tool which is presented and validated in Chapter 5.

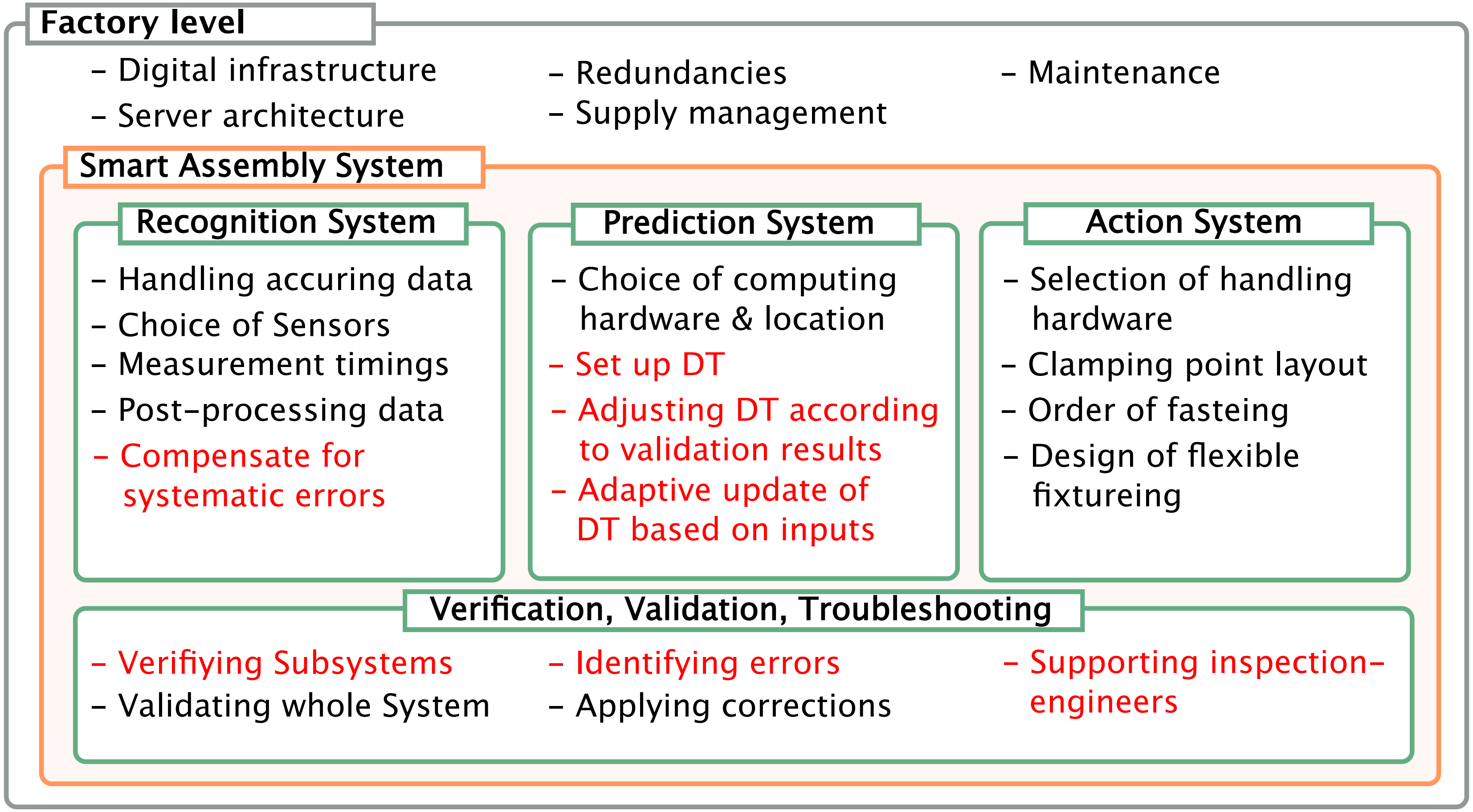


Fig. 2.6 Challenges to overcome during implementation of smart assembly system. The challenges are clustered by: (a) factory level and (b) sub-systems of the smart assembly system itself.

Core References of this Chapter

[1] Felix Claus, Franca-Alexandra Rupprecht, and Hans Hagen. Online simulation considering production uncertainties to improve assembly quality. NAFEMS, 2019

Chapter 3

Gravity Compensation of Scanned Non-Rigid Parts

3.1 Introduction

This Chapter is based on core references [2] and [3] and presents a gravity compensation method for post-processing 3D-scanned non-rigid parts. Optical measurement devices often referred to as 3D-scanners, became popular over the last years, also for industrial applications. With these scanners, several millions of points can be acquired within seconds resulting in a high-resolution point cloud representation of the scanned object. This enables the generation of high-fidelity simulation models preserving the actual geometrical information provided by the 3D-scan. These geometrical DTs can for instance be used to make precise predictions for assembly quality. However, the acquired point cloud is subjected to systematical errors caused by gravitational forces. Especially, for parts that are considered non-rigid, the error magnitude cannot be neglected. To compensate for occurring deflections during measurement, special fixtures are used that add additional support to the part. By adding additional supports to the measurement fixture, the part is positioned in an over-constrained way. To make measurements in the over-constraining state comparable to post assembly measurements, the fixture layout is chosen such orientation and allowed clamp forces are similar to conditions in assembled state. These kinds of fixtures require time-consuming planning and are also costly in terms of hardware. Although deflections caused by gravity can be minimized due to over-constraining the part, unknown tensions are induced. This makes it inevitable to compensate for tensions and deflections that occur during measurement to be able to create a correct DT from acquired measurement data. This compensation step can be carried out with different methods which are discussed in this chapter. Unfortunately, most methods are restricted to measurements performed with non-over-constrained fixtures. For large components from the aerospace industry, over-constrained fixtures are necessary to prevent damage to the components. To tackle this issue, this chapter provides an FE-based shape optimization algorithm that can be applied to non-over-constrained fixture setups and that can be extended to over-constrained setups as well. The precision of the approach is benchmarked with different kinds of verifications and validations involving real sensor data. Correct post-processing of scan data is essential to obtain a correct geometrical DT from measured point clouds. Respectively to the smart assembly system introduced in Figure 2.2 this chapter contributes to the recognition system.

This chapter is structured as follows: Section 3.2 provides an overview about related literature and state of the art approaches. Section 3.3 is an extended problem description which emphasizes the need for a gravity compensation method. In Section 3.4 the methods from [2] and [3] are excerpted. These approaches are verified in Section 3.5 and validated with experimental data in Section 3.6. Last, section 3.7 discusses method and achieved results considering the use case of smart assembly.

3.2 Background

The problem of finding an unloaded shape based on the known loaded shape and the also known load case is called "inverse form-finding problem." To solve for the unloaded shape, approaches from the literature can be clustered into two groups: (a) approaches based on inverse finite element simulations and (b) approaches based on mapping pre-calculated deformation fields. In addition, concepts that make use of compensation methods to avoid costly fixtures are discussed in this section.

3.2.1 Inverse Finite Element Method

If the loaded shape of a part is given as FE-mesh, the most intuitive way of finding the unloaded shape is to invert the FE-calculations. This kind of approach is called iFEM[1]. In [100] an inverse formulation is compared to a L-BFGS[2] algorithm. The results show that the inverse formulation is superior in terms of computational times while no compromises are made regarding output precision. In [101] the same authors present a mixed inverse form-finding approach that uses both, direct FEM and iFEM to address non-linearity prob-

[1] inverse Finite Element Method

[2] Limited-Broyden-Flechter-Goldfarb-Shanno

lems where the inverse formulation is not applicable. This approach is closely related to the method presented in this chapter. In [102] the approach is extended even further to orthotropic plasticity problems. For the use case of electric elasticity [103] proposed an inverse formulation as well as different case studies for verification.

iFEM is the most promising approach in terms of precision and computational cost for solving inverse form-finding problems. The main drawback is the need for a special solver that can handle inverse FE-problems. This affects the usability for industrial applications where commercial FE-software is used. Also, iFEM is currently restricted to available validated formulations which makes it only applicable to a restricted spectrum of simulation models.

3.2.2 Mapping of Pre-Calculated Deformation Field

A simple approach to compensate for occurring deflections is to apply corrections to an acquired point cloud based on a pre-calculated displacement field. This displacement field is commonly obtained from an FE-model that is based on the nominal geometry. This model is used to solve for the displacement field that results from gravity in fixtured state. The obtained field must be registered to the acquired measurement data. In [104] a voxel-based filter is employed to match point cloud with simulation data. This approach presumes that both data are already registered to the same reference frame. Another commonly used approach for registering two sets of points is called MDS[1]. MDS can be used for different kinds of applications, see [105–107]. A detailed description of this method can be found in [108]. Surveys are provided in [108] and [109]. To improve the process of point cloud registration, an approach was proposed in [59] which uses the properties of geodesics. To match two different sets of points, a grid of geodesics is calculated on both meshed point clouds. The assumption is made that the geodesics of two similar surfaces remain the same such positions of points can be expressed relative to the grid of geodesics. In [110] experimental validations for a use case from the aerospace industry were presented using a mapping approach. Although these results are also within pre-defined quality goals, the principle of mapping pre-calculated deformation fields has a significant disadvantage: the assumption of similarity between nominal CAD geometry and measured geometry. This assumption induces a systematic error by not considering the actual geometric stiffness of the measured part, resulting in an unknown error. Depending on the kind of part and kind of deviations this error can differ and must be taken into account for every part individually. In addition, mapping approaches are only applicable to non-over-constrained fixture setups.

[1] Multidimensional Scaling

3.2.3 Fixtureless Inspection and Virtual Clamping

To reduce the cost and time spent using special fixtures to compensate for deflections during measurement, recent approaches make use of virtual compensating approaches as presented in Section 3.2.1 or Section 3.2.2. The idea of fixtureless inspection is to use a simple kind of fixture or none fixture at all and virtually compensate for occurring deviations by calculating the GF state. In a second simulation step, the part can be virtually clamped in any desired fixture setup. This workflow is called VC[1]. In [111] VC is used to inspect sub-frame components of an automotive vehicle. In [60] an FE-based fixtureless inspection method is proposed that does not compensate for gravitational effects. In this work, only surface defects are of interest, and the stress FE-output is used to identify dents and buckles. This work was extended in [112] by providing verifications and validations. A survey of fixtureless inspection methods is provided in [113]. The main disadvantage of fixtureless inspection where no compensation of gravitational forces is used is this: observed deviation are not subdivided into deviations caused by load and actual geometrical deviations. This makes these approaches only applicable for assessing surface defects or parts that can be considered rigid.

3.2.4 Resume

The correct post-processing of measured non-rigid parts is crucial for generating DTs from scan data. This processing step can be currently solved in different ways. Most popular is the approach of mapping a pre-calculated displacement field to the acquired scan data and applying a vertex offset in the inverse direction. Although this method works fast, it is subjected to systematical errors caused by differences in geometrical stiffnesses between actual and nominal part. This problem can be solved by employing iFEM methods which precisely can solve for the desired unloaded shape. Unfortunately, iFEM methods require an FE-mesh representation of the loaded shape which is costly to obtain from every individual part. Also, a special implementation is necessary to solve the inverse formulation problem. This chapter addresses these issues by presenting an iterative inverse form-finding algorithm that only uses direct FEM.

[1] Virtual Clamping

3.3 Extended Problem Description

This section discusses in detail the problem of 3D-scanning non-rigid parts on different kinds of fixtures. Also, simulated examples are used to demonstrate possible effects onto the outcome of available compensation methods.

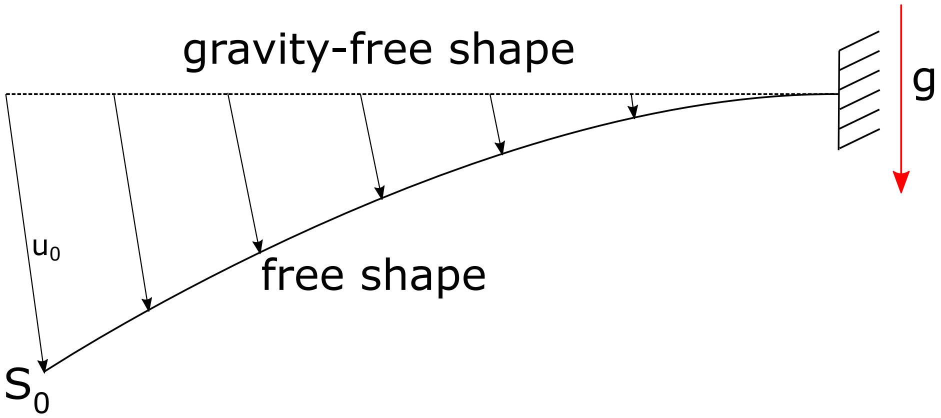


Fig. 3.1 2D example of a GF-shape and free shape. The gravity load does deform the GF-shape by u_0 resulting in the free shape S_0. (This figure has been published in [2], licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/; changes: none.)

3.3.1 Non-Over-Constrained Fixture Setups

When a non-rigid part is placed on a fixture, deflections caused by its own weight, occur. In Figure 3.1 an easy to understand 2D example is shown. The figure shows a beam that is only constrained at the right end which is a non-over-constrained fixture setup. Due to gravity g, the beam deflects by the displacement field u_0 towards the free shape S_0. The free shape is the situation that is measured by the 3D-scanner. The displacement field u_0 results from the gravity and must be calculated to obtain the GF-state from the scanned geometry. As mentioned before, there are different ways of calculating this displacement field:

Direct FEM - The GF-shape is assumed to be close to the nominal shape defined by CAD definition. An FE-simulation mesh based on the nominal geometry can be created and the known material properties, mechanical boundary conditions, and direction of gravity can be added to complement the simulation model. Next, an FE-simulation is performed to solve for u_0. The inverted displacement field u_0^{-1} can be applied to the shape S_0 to obtain the GF-shape.

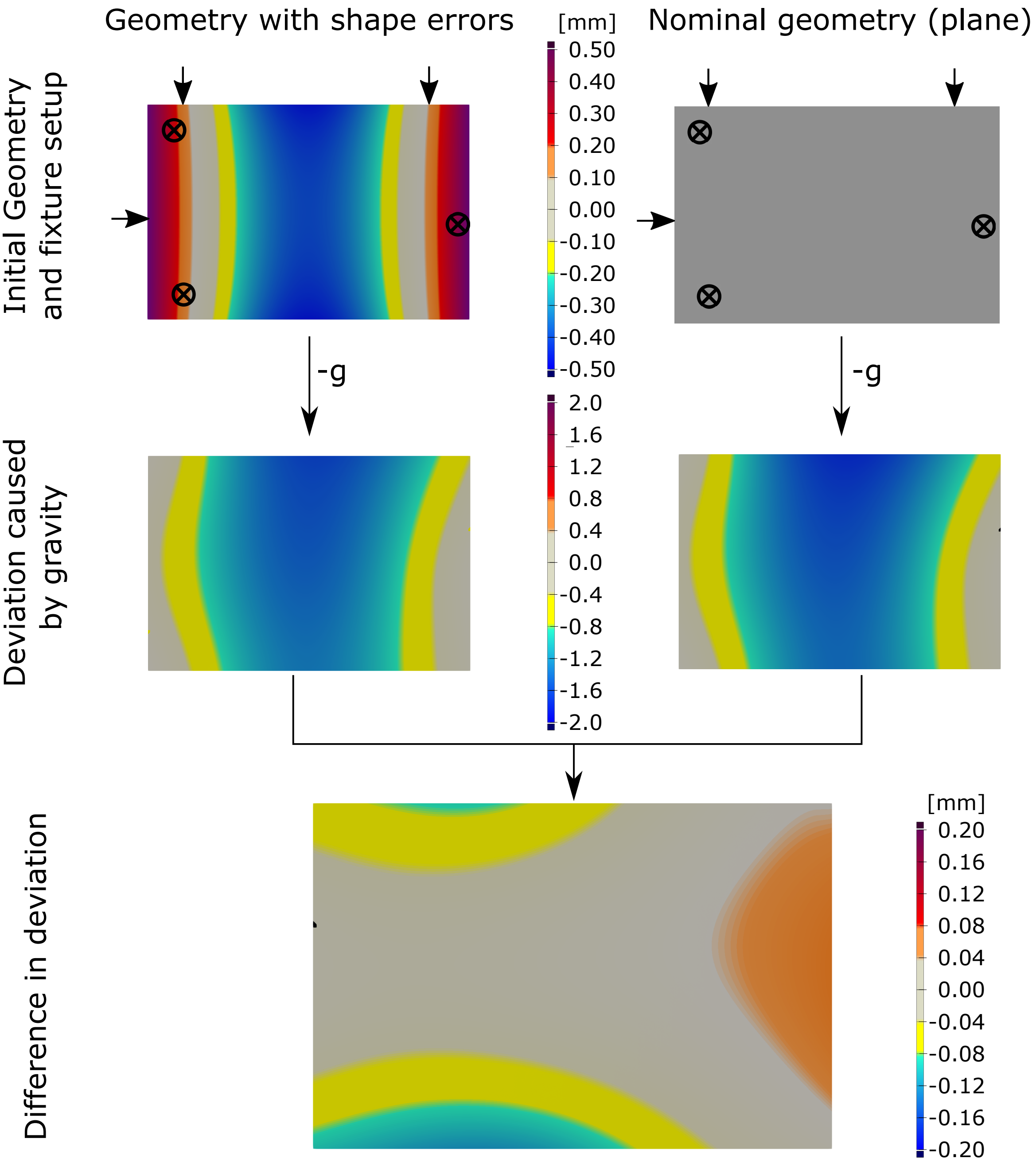


Fig. 3.2 Comparison of similar shapes placed on the same fixture. Top: Initial geometry compared with nominal geometry. Middle: deflection caused by gravity for each geometry. Bottom: direct comparison between deflections; differences indicate errors when assuming similarity.

Inverse FEM - For applying iFEM a simulation mesh must be created from the measurement which represents the loaded shape (free shape). Also, material properties, boundary conditions, and load case must be defined to complement the simulation model. By solving the inverse FE-problem, the inverse displacement field u_0^{-1} is obtained directly without similarity assumptions.

Figure 3.2 demonstrates the issue when assuming similarity between nominal and actual geometry. The figure shows a virtual experiment where a simple piece of sheet metal with- and without shape errors is simulated on the same fixture (3-point non over-constrained). The size of the simulated part is 200 mm by 300 mm by 0.5 mm. The geometry is meshed with 15000 first order shell elements. The material model is assumed to be linear with the properties for aluminum:

- Young's Modulus: 70 GPa
- Poisson's Ratio: 0.3
- Density: 2.7 $\frac{g}{cm^3}$

The second row of the figure shows the occurring deviations compared to initial geometry when gravity is applied perpendicular to the image plane. In mapping approaches, this displacement field is assumed to be equal. The direct comparison (bottom picture) shows the error that is neglected when mapping a pre-calculated displacement field based on nominal geometry. Already in this simple example errors up to 0.12 mm can be observed for areas far away from supports. As this error depends on individual differences in shape but also on the part size and structure, this error must be handled with care. If one does use such an approach that is based on similarity, the magnitude of the error is always unknown which makes it not applicable to high precision applications.

3.3.2 Over-Constrained Fixture Setups

To compensate for large deflections caused by gravitational effects, over-constrained fixture setups are used. To minimize the tensions that are induced to the part by the fixture, different methods can be found in the literature, see [40, 41, 114]. When adding additional supports to the fixture setup to compensate for part deflections, the fixture gets over-constrained. This results in an unambiguous geometry during measurement. The problem is described using Figure 3.3. In the 2D-figure two similar geometries - nominal (CAD) and actual geometry - are compared in different states and fixtures. a) shows both parts in the GF-state, where differences can be observed. In b) gravity is applied and the parts deflect because of their own weight. In c) an additional support (blue) is placed in the middle to compensate for deflections caused by gravity. This step makes the fixture setup over-constrained and both geometries cannot be differentiated anymore. By using only geometrical information of the parts' geometry and locator positions, the GF-shape can not be obtained due to the over-constraining. To resolve the over-constrained situation, additional information is required. One possible way is measuring the force, pressing down on over-constraining supports. By that, all required information is given to resolve for the GF-shape.

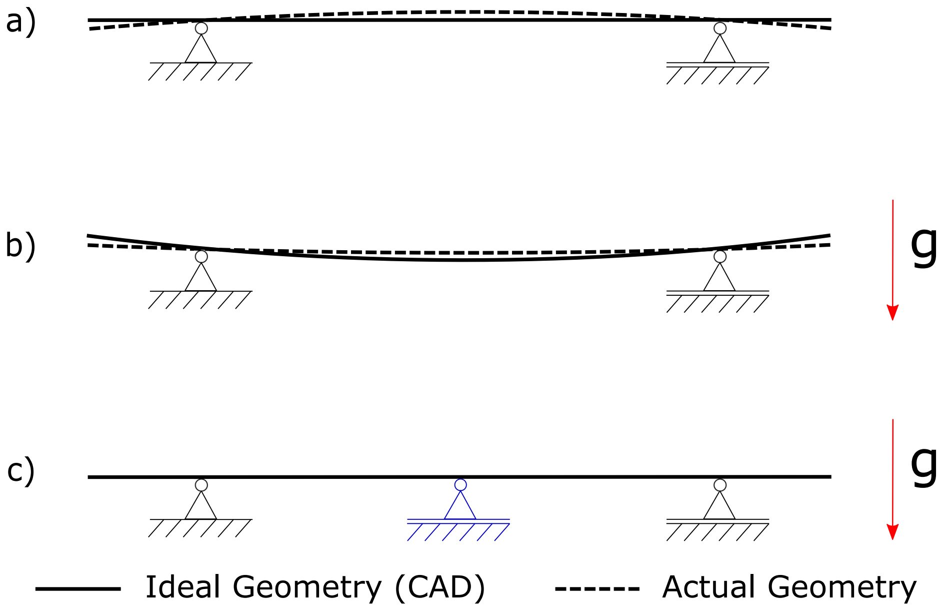


Fig. 3.3 Unambiguous behavior of flexible parts when using over-constrained fixture setups: a) GF-shape of CAD- and actual- geometry; (b) free shapes based on loading GF-shapes with gravity; (c) resulting shapes when adding additional support to compensate for occurring deflections caused by gravity. (This figure has been published in [2], licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/; changes: none.)

Not considering deflections of the part caused by over-constrained fixtures can lead to significant errors on the scale of millimeters. One can think of this problem like this: there exists a shape that barely touches the over-constraining support when it's loaded with gravity. In this loaded state, the part could have the exact nominal shape. On the other hand, there exists a shape where the force onto the over-constraining support is maximal while the part also has the exact same nominal shape when placed on the fixture. Comparing these two shapes in the GF-state will reveal significant differences in shape that must be considered for the generation of geometrical DTs. In Figure 3.4 this problem is illustrated. The first row shows the shape of a geometry in GF-state and gravity loaded shape, where the over-constraining support carries the maximum of the weight of the part. The loaded shape, however, does match with the nominal geometry. The second row shows the inverse extreme, where the load on the third support is minimal. The last row shows the direct comparison of the two geometries in the GF-state resulting in large deviations.

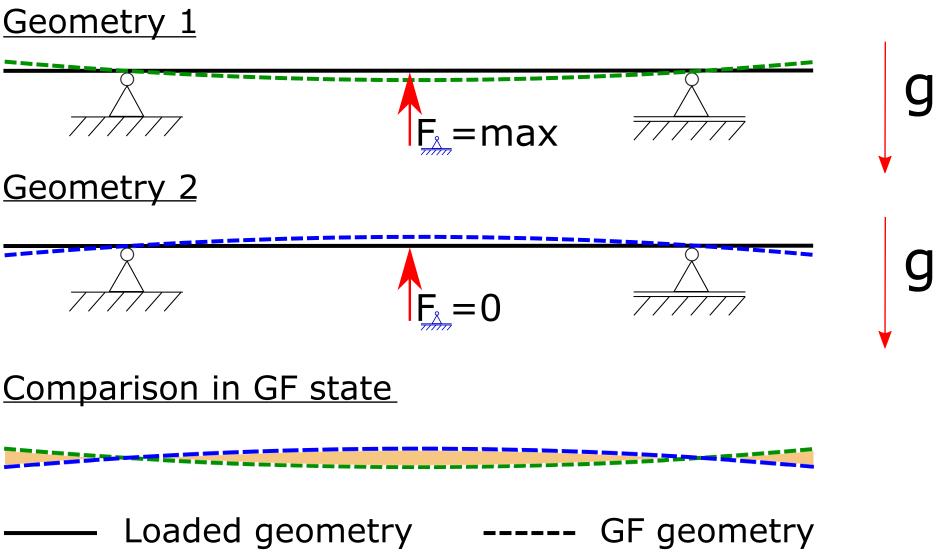


Fig. 3.4 Geometry 1: maximum shape deviation in direction of gravity, force on over-constraining support is maximal; Geometry 2 maximum shape deviation in the opposite direction of gravity, force on over-constraining support is just zero. Bottom: spectrum of possible GF-geometries with almost identical shapes in the free state, when placed on over-constrained fixture.

3.3.3 Resume

In summary, the post-processing of scanned non-rigid parts is a crucial aspect for the generation of DTs preserving real parts geometry. Especially, when working with over-constrained fixture setups this post-processing step cannot be carried out anymore with simple approximating mapping approaches. Therefore, a precise solution is required that ensures correct post-processing of acquired 3D-scans. In the following of this chapter, such a solution is presented, verified, and validated. This solution is capable of post-processing 3D-scans performed with over-and non-over-constrained fixture setups.

3.4 Method

The method presented in the following is excerpted from core reference [2] and [3]. Fundamentals of the approach are described in Quotation 3.1, followed by an extension of the approach to handle over-constrained fixtures in Quotation 3.2.

Quotation 3.1: Chapter 5 excerpted from [2][1]

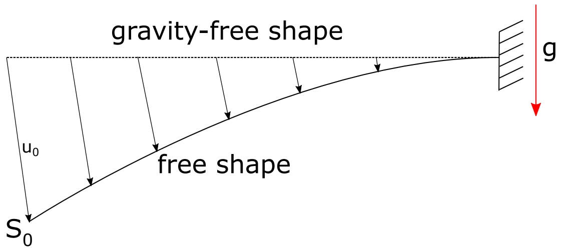


Fig. 1 Problem definition. Example of a simple 2D beam constrained at the right end; S_0 represents the free shape resulting from the influence of gravity (u_0).

⋮

5 Method

This section is split into two parts. First, the method is explained by discussing mathematical aspects of calculating the GF-shape. Second, the used algorithm resulting from the mathematical description is presented. To calculate the GF-shape, the method needs the following inputs: (1) a mesh representation of the measured object, (2) the locator positions and directions, (3) material information, and (4) the gravity direction. The output of the computation will be a mesh representation of the GF-state of the measured object.

5.1 Generation of the Simulation Model

The first step after measuring a part is the generation of a simulation model from the measured geometry. There are many different methods available for the generation of a mesh from a measured geometry. I. Gentilini and K. Shimada [20] for example, are applying a bubble packing method to the point cloud to reduce its density.

[1] This content has been published in [2], licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/; changes: none

After triangulating the coarsened point cloud they define the elements as FE shell elements to run simulations. Another method that automatically creates volume FE-meshes from STL files is proposed by Y. Liu et al. [25]. More recently, also a simulation method called "Finite Cell Method" became more popular. This method enables, to run simulations completely without the use of meshing the measured object. It can handle three-dimensional point clouds from different kinds of scans. A detailed review and description of the method are given by D. Schillinger and M. Ruess [26]. Another possible way to obtain an FE-mesh from a measurement is to use commercially available software tools for reverse engineering to get a CAD representation of the measured object and then again meshing the new CAD model. Thus, the generation of the simulation mesh is treated as a problem on its own that can already be solved in a satisfying way for different kinds of applications. To finalize the simulation model material properties, boundaries and loads must be added. The boundaries and loads can be derived from the measurement fixture. As mentioned, the fixture must be unambiguous to calculate the GF-shape only with geometrical information as input (compare Section 4.2). For further explanations, the load case shown in Figure 1 is used.

5.2 Iterative Approach (Loop 1)

Starting from the initial situation, the most intuitive way of finding the GF-shape would be to invert the direction of the load. This might work for really small magnitudes of u_0 and in some special cases but is wrong in general. As such, it is not applicable reasoned by the difference of geometrical stiffnesses. The pre-studies in Section 4.2 already discussed this problem. Motivated by the problem of different magnitudes by flipping the gravity direction, an iterative approach is used to approximate the displacement field u_0. The basic idea of this approach is to subdivide the total gravity load into smaller increments Δg and applying the load incremental wise to the structure. Per iteration, two simlation runs are performed, one with the load of $+\Delta g$ and in a separate simulation, the same incremental load is applied in the inverse direction. Figure 5 shows the resulting displacement fields $u_0^{+\Delta g}$ and $u_0^{-\Delta g}$. Now, the magnitudes of the displacement field $u_0^{+\Delta g}$ can be used to apply it in the direction of the field $u_0^{-\Delta g}$ for every node. The resulting shape S_1 is the updated mesh configuration after the first iteration, being the initial configuration for the next iteration. The shape S_1 can be expressed mathematically as:

$$S_1 = S_0 + (|S_0^{+\Delta g} - S_0| * \frac{S_0^{-\Delta g} - S_0}{|S_0^{-\Delta g} - S_0|}) \tag{2}$$

The displacements $u_0^{\pm\Delta g}$ can be written with:

$$S_0^{\pm\Delta g} - S_0 = u_0^{\pm\Delta g} \tag{3}$$

(3) in (2) results in:

$$S_1 = S_0 + (|u_0^{+\Delta g}| * \frac{u_0^{-\Delta g}}{|u_0^{-\Delta g}|}) \tag{4}$$

To approximate the target displacement field u_0 all incremental displacements need to be summed up:

$$u_0 \approx \sum_{i=0}^{n} |u_i^{+\Delta g}| * \frac{u_i^{-\Delta g}}{|u_i^{-\Delta g}|} \tag{5}$$

In every iteration, only node positions of the mesh are updated. Tensions and forces are not considered for the next iteration. When updating the position of a node, the geometry of the simulation model is changed. Thus, the displacement field calculated in the next iteration changes as well. This iterative approach solves the problem of different geometrical stiffnesses in different directions of gravity.To obtain the GF-shape approximation, iterations are performed until the sum of all Δg values equals the value of g, captured mathematically in Equation 5. Note that one could think, performing only the simulation run with $+\Delta g$ and flipping the displacement vectors might be sufficient, as the direction of displacement is very similar to the simulation run with $-\Delta g$, for small Δgs. Unfortunately, it turned out that this slight difference in direction is summed up while performing the iterative steps leading to recognizable errors.

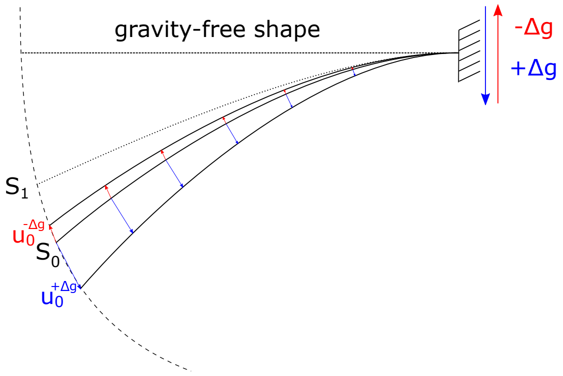


Fig. 5 Calculating one increment of "Loop 1" from shape S_0 to S_1 by displacing the mesh nodes using the magnitude of the displacement vectors calculated with the load $+\Delta g$ (blue) an applying them into the direction of the vectors calculated with $-\Delta g$ (red)

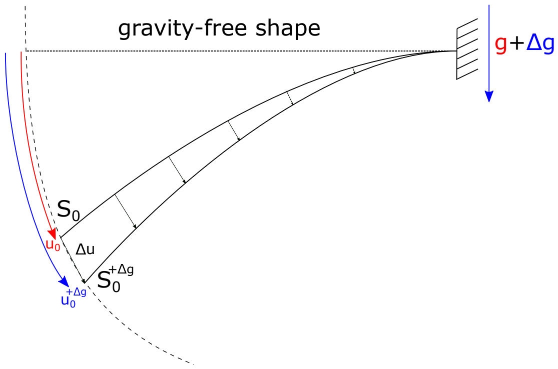


Fig. 6 Comparing one increment of Loop 2 by calculating $S_0^{+\Delta g}$in two different ways.

The user controls increments for this "Loop 1." By defining how many iterations are to be performed, the value of Δg is given as $\Delta g = g/\#iterations$. Similar to other iterative approaches, it is expected that the quality of the approximation improves with an increasing number of iterations, at the expense of computational cost. The ideal number of iterations is a trade-off between precision and computational cost and must be determined for every application individually via a sensitivity study. Once the ideal number of iterations is determined, this number can be used for the calculation of the GF-shape for further measurements of the same part.

Iterative approaches can be used to approximate a solution. By reducing the increment size resulting in a higher number of iterations the solution should become more precise. It turned out that the presented approach has a weak spot that causes a convergence towards a shape that slightly differs from the desired GF-shape.

5.3 Stress Stiffening (Loop 2)

To find the reason for the error in the approach of Section 5.2, first, the solution of one iteration is verified. To do so, two simulations are compared. The first simulation is applying $g + \Delta g$ to the known GF-shape to obtain the configuration $S_0^{+\Delta g}$ from Figure 6. The second simulation is applying the load $+\Delta g$ on the free shape S_0. Note that the shape S_0 was calculated before by applying the load $+g$ to the GF-shape and exporting the mesh. The initial meshes of both simulations are considered as stress- / strain-free. It turned out that these two simulations do not result in the same geometry. As a reason, an effect called "Stress Stiffening" is identified, which is not considered in the second simulation which causes the differences in the simulation results.

Stress stiffening sometimes also named geometric stiffening, incremental stiffening, initial stress stiffening, or differential stiffening is the effect that a loaded structure behaves differently from a stress-free structure. This effect usually occurs in simulation with thin structures like shells or beams. Depending on the geometry, the structure can be stiffened or weakened by this effect. The mathematical aspects are not discussed in detail, they can be looked up for example in the Ansys Theory Guide [27]. For further improvements, it is only important to know that the stress stiffening effect influences the behavior of a loaded structure.

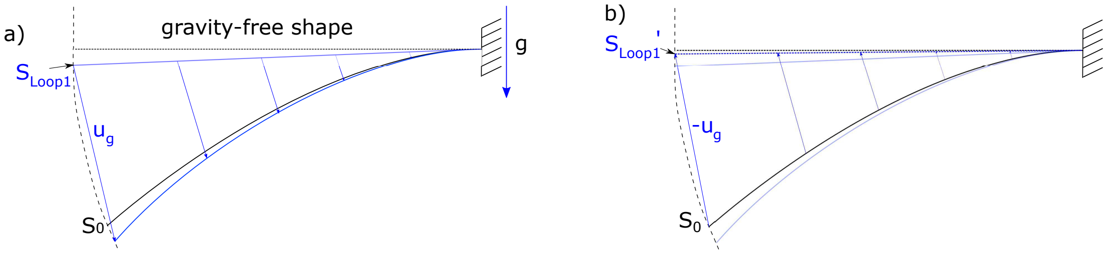


Fig. 7 Two-step optimization of "Loop 2." a) First, full gravity load g is applied to the shape S_{Loop1}; b) second, a displacement field u_g is applied in inverse direction to the shape S_0, resulting in $S^{\prime}_{Loop1}$.

As the stresses are lost after the mesh update in each iteration of "Loop 1", the stress stiffening effect cannot be considered directly. To overcome this problem, a second iterative approach is used, based on the calculations made before as described in Section 5.2. As mentioned before, a shape that does not exactly match the desired GF-shape can be calculated, but this shape is already very close to the GF-shape. The shape obtained from Section 5.2 is further used as initial geometry for the following calculations. The basic idea for the next iterative simulations is to use the similarity of the GF-shape and the already calculated shape. Both shapes result in similar displacement fields when applying the full gravity load. As the stress stiffening effect is considered in the forward simulation, the resulting displacement field will differ from the approximation calculated before in Section 5.2. This scenario is depicted in Figure 7 a), where S_{Loop1} is the shape initially used from the previous iterative calculations and S_0 the free shape. Now, the displacement field u_g that has been obtained from loading S_{Loop1} with the full gravity, can be used to displace the free shape S_0 in the opposite direction. The resulting shape $S^{\prime}_{Loop1}$ is closer to the desired GF-shape (see Figure 7 b)). This behavior can be expressed as:

$$|(S_n - S_n^g) - u_0| < |(\sum_{i=0}^{n} |u_i^{+\Delta g}| * \frac{u_i^{-\Delta g}}{|u_i^{-\Delta g}|}) - u_0| \tag{6}$$

As both geometries, S_0 and S_{Loop1}, are represented by the same mesh the displacements can be applied per node. So a registration method for matching the geometries is not necessary. By running the forward simulation from Figure 7 and updating the shape S_{Loop1} multiple times, the resulting geometry comes every step closer to the GF-shape. So the following sequence can be established:

$$S_i = S_0 - (S_{i-1}^{+g} - S_{i-1}) \tag{7}$$

The limit of S_i is the desired GF-shape:

$$S_{grav-free} = \lim_{i \to \infty} S_i \tag{8}$$

As the change of the shape from S_i to S_{i+1} can be calculated, a termination criterion ε can be defined as:

$$\varepsilon > max(|S_i - S_{i+1}|) \tag{9}$$

The maximal change of the shape converges to zero for an infinite number of iterations. The threshold ε can be viewed as a measure for the precision of an intermediate result. This approach allows a user to control output precision for a specific application by specifying the value of ε.

Note that the whole calculation of the GF-shape might work by only applying "Loop 2" multiple times directly to the measured shape S_0, but this would result in a large number of iterations that can be drastically reduced by using the geometry of "Loop 1" as input for the second loop.

5.4 Algorithm

This section will describe an algorithmic implementation of the method described before. This algorithm highly relates to the method proposed by S. Germain et al. [6]. The difference in the method which is proposed in this paper is that it does not use any inverse FE-formulation. This might result in way higher run times (see 6.3) but makes the method easier to implement with almost any simulation software and thus easier and capable to use in real applications.

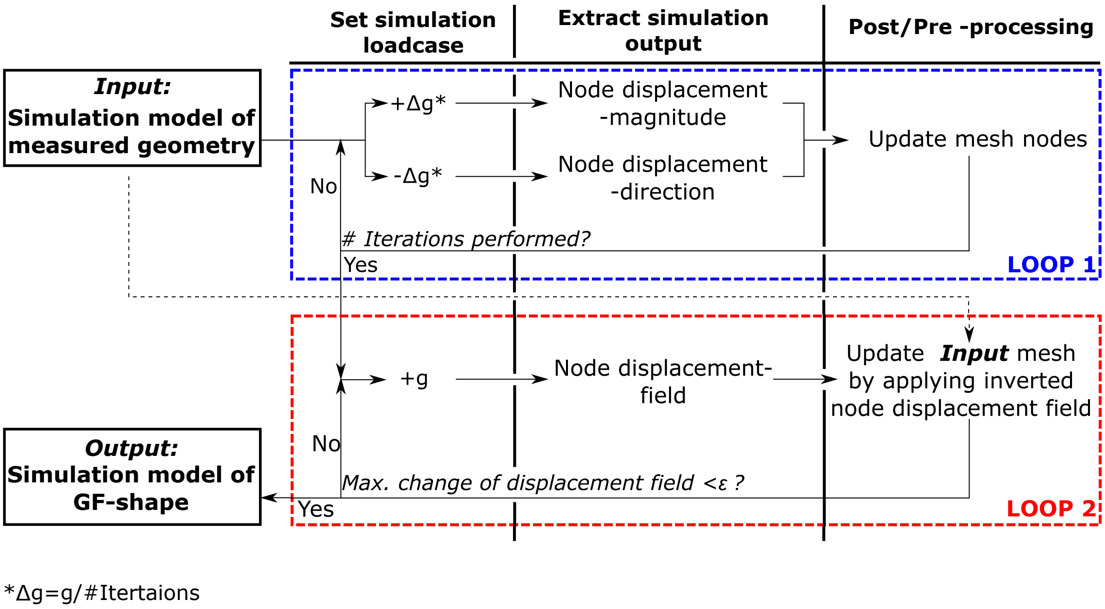

Set simulation loadcase
Extract simulation output
Post/Pre -processing
Input: Simulation model of measured geometry
+Δg*
-Δg*
Node displacement -magnitude
Node displacement -direction
Update mesh nodes
No
# Iterations performed?
Yes
LOOP 1
+g
Node displacement-field
Update Input mesh by applying inverted node displacement field
Output: Simulation model of GF-shape
No
Max. change of displacement field <ε ?
Yes
LOOP 2
*Δg=g/#Itertaions

Fig. 8 Flow chart of algorithm to calculate the GF-shape. The algorithm is divided into two loops; columns represent steps that are performed during one iteration. Loop 1, shown in blue, iterates a fixed number defined by the user. Loop 2, shown in red, terminates based on a threshold condition.

To explain the algorithm in detail, Figure 8 is used. As a prerequisite a simulation model of the measured geometry, containing boundaries, material information, and direction of gravity is necessary. The algorithm is split up into two loops. These loops refer to the iterations explained before in Section 5.2 and 5.3. Before starting the computation, the user has to define the number of iterations for the first loop and the abort criteria ε for the second loop. At the beginning of "Loop 1", a Δg is defined by dividing the full gravity load by the defined number of iterations. Then, two simulations can be started in parallel, one with the load vector $+\Delta g$ and one with $-\Delta g$. From these two results, the magnitude and direction of the displacement for the node update can be obtained. This step refers to the principle explained in Section 5.2. After the node positions of the mesh nodes are updated, the next iteration of "Loop 1" can be performed with the updated mesh. When all iterations are done, the mesh geometry already is close to the GF-shape. The updated mesh geometry is then used as input for "Loop 2". This loop refers to the iterations explained in Section 5.3.

```
Data:
    - Simulation model of free shape / nodes and loads accessible
    - number of iterations (user-defined variable see 5.2 for further explanation)
    - $\varepsilon$ (user-defined variable see 5.3 for further explanation)
Result: Simulation model of gravity free shape
LOOP 1: for number of iterations do
    set gravity= 9.81/number of iterations
    run $sim_1$
    set gravity= -(9.81/number of iterations)
    run $sim_2$
    get node_displacement_magnitudes from $sim_1$
    get node_displacement_vectors from $sim_2$
    node_displacement_direction=normalize (node_displacement_vectors)
    update node_positions(node_displacement_direction*node_displacement
    _magnitudes)
end
set gravity= 9.81
LOOP 2: while $delta > \varepsilon$ do
    run sim
    get node_displacements from $sim$
    overwrite node_positions with (initial_node_position-node_displacements)
    delta=max(abs((node_displacements($sim_n$) - node_displacements($sim_{n-1}$))))
end
```

Algorithm 1 Pseudocode of the Algorithm, splitted into two loops that refer to sections 5.2 and 5.3.

Here, the following steps are performed. First, the full load of $+g$ is applied to the resulting mesh from "Loop 1". Then, the direction of the displacement vectors become flipped and applied to the original input mesh that represents the measurement of the free state. The resulting mesh is closer to the GF-shape then the result from "Loop 1" and is used for the next iteration of "Loop 2". This loop is running until the maximum change of the displacement field from one iteration to the next one is below the user-defined value ε. The result of the algorithm is a simulation model of the part in its GF-shape.

From here, the user can run any forward simulation to virtually clamp or assemble the part in any desired situation to perform quality measures in other situations if needed.

As stated in the abstract this algorithm does not need a CAD model of the part and also does not use any inverse formulations. The user merely needs access to the node position of the mesh and the displacement vectors of the results to implement the proposed algorithm. Commonly, these are fundamental functions provided by scripting interfaces of commercial simulation software. An alternative visualization of the method is given in Algorithm 1 expressed as pseudocode.

⋮

References

6. Germain S, Landkammer P, Steinmann P (2014) Adv Model Simul Eng Sci 1(1):10. https://doi.org/10.1186/2213-7467-1-10

20. Gentilini I, Shimada K (2011) Comput Aided Des 43(3):316. https://doi.org/10.1016/j.cad.2010.11.004

25. Liu Y, Saputra AA, Wang J, Tin-Loi F, Song C (2017) Comput Methods Appl Mech Eng 313:106. https://doi.org/10.1016/j.cma.

26. Schillinger D, Ruess M (2015) Arch Comput Methods Eng 22(3):391. https://doi.org/10.1007/s11831-014-9115-y2016.09.038

27. ANSYS. Theory guide (2017). http://www.mm.bme.hu/~gyebro/files/ans_help v182/ans_thry/

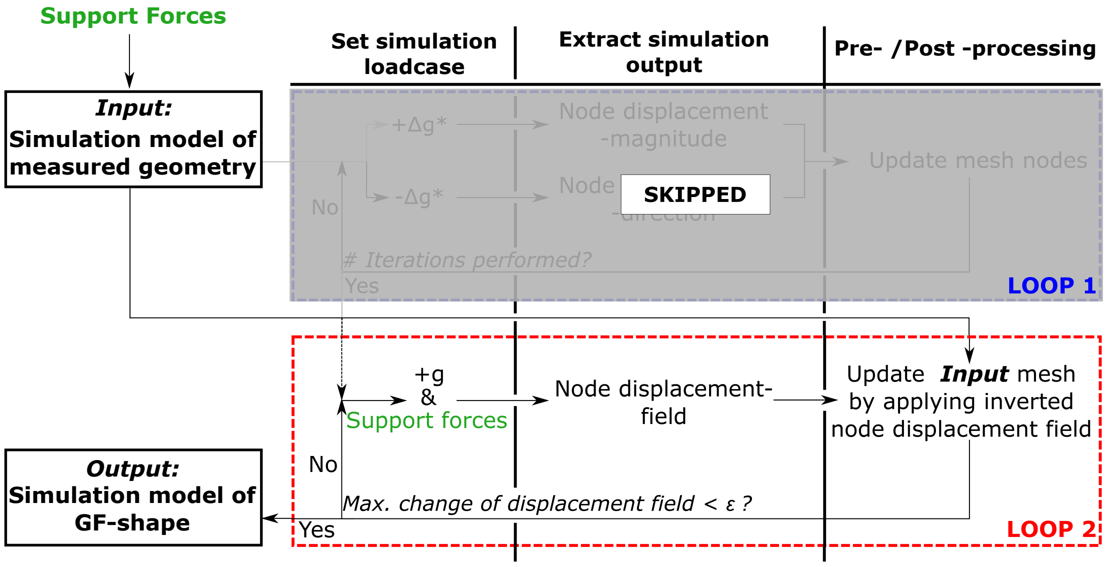
The so far presented method is only capable of post-processing non-over-constrained fixture setups. To enable post-processing for the over-constrained case, the algorithm is extended in the following by Quotation 3.2.

Quotation 3.2: Chapter 2 excerpted from [3]

Extensions to the Earlier Method

To apply the proposed algorithm to measurement data acquired on over-constrained fixture setups, different changes are made. Figure 2 shows an overview of the changes made to the existing algorithm.

It is most important to add the support forces to the input-simulation model. By adding this(these) force(s), the information necessary for resolving the over-constrained situation is provided to the model. Not all support boundaries need to be substituted by forces in the simulation model, only the supports that cause the over-constrained situation must be considered for change. Only for the supports that over-constrain the part, the forces occurring during measurement must be obtained. The second modification to the original method is to skip the first loop of the algorithm. This loop is intended to provide a robust way of finding a first coarse approximation of the GF-shape. “Loop 2” improves the result of “Loop 1” resulting in a better approximation of the GF-Shape. “Loop 1” is now intentionally skipped as we assume that the part is already in a shape similar to the GF-Shape due to the over-constrained fixture. Although skipping “Loop 1” is optional, the computational cost can be reduced drastically as this loop is the most expensive part of the overall algorithm.


Support Forces
Set simulation loadcase
Extract simulation output
Pre- /Post -processing
Input:
Simulation model of measured geometry
+Δg*
-Δg*
Node displacement -magnitude
Node -direction
SKIPPED
Update mesh nodes
No
# Iterations performed?
Yes
LOOP 1
+g & Support forces
Node displacement-field
Update Input mesh by applying inverted node displacement field
No
Max. change of displacement field < ε ?
Yes
LOOP 2
Output:
Simulation model of GF-shape

*Δg=g/#Itertaions

Figure 2: Modifications made to the original method: (1) Adding specific support forces, (2) skipping "Loop 1".

In the second loop, the support forces are considered for the load case. All loads – gravity and support forces – are applied in one simulation step. This loop performs iteratively a simple forward simulation and applies the resulting node displacements in inverse direction always to the original input simulation model. The resulting shape

> converges towards the desired GF-shape. The change of the displacement output is monitored and serves as adaptive termination criterion for this loop. This method does only need access to the node positions of the simulation model and the displacement output of the simulation result. It can be easily implemented with almost any commercial FE-simulation software.

The presented approach can be categorized as an inverse form-finding algorithm that employs shape optimization procedures. The advantage of this approach is, that only direct FEM is used to solve the problem iteratively which makes it applicable to commercial simulation software. A prototype implementation was realized using python together with the simulation software “ABAQUS” from Dassault Systèmes.

3.5 Verification

To verify whether the described method does perform as expected, different verifications are performed that use simulated data as input. The verifications are carried out by simulating different geometries and materials.

3.5.1 Simulation Models

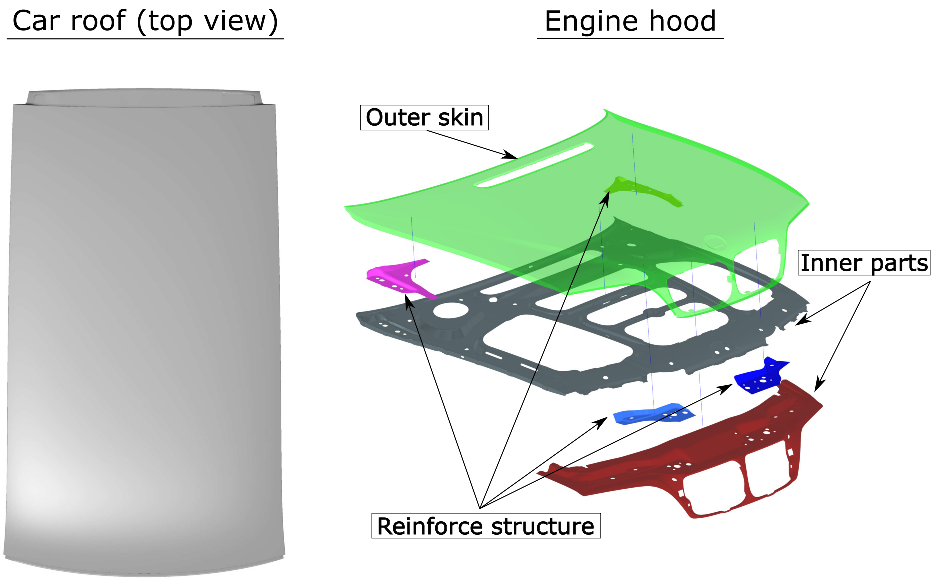


Fig. 3.5 Left: Geometry of a car roof. Right: Assembly structure of an engine hood (BMW E46 facelift).

To ensure that results from this verification are not lucky strikes, three different geometries are used for testing. Besides the simple piece of sheet metal already introduced in Section 3.3.1, two real-world use cases from the automotive industry are presented. These geometries are shown in Figure 3.5.

Car Roof - The geometry shown on the left side of the figure is a car roof from an automotive application. The part is made out of a single piece of sheet metal and has the size of about 2000 mm x 1000 mm. The material is steel with a thickness of 0.7 mm. The geometry was meshed with 169,378 single-order shell elements. An adaptive meshing was applied to achieve the best trade-off between the number of mesh elements and geometrical approximation. The used meshing parameters were elaborated within industrial projects and were proven to be the best practice for this application.

Engine hood - The second geometry is an engine hood, also from an automotive application. This part has outer dimensions of about 1300 mm x 1500 mm x 400 mm and composes of seven individual components that are joined together. The simulation model was built up by first meshing all individual components with a total of 365,152 mesh elements. The same adaptive meshing strategy was used. Next, all interconnections between components like spot-welds and adhesives are modeled. The simulation model is finalized by defining material properties.

The properties of all used simulation models for verification purposes are summarized in Table 3.1. For all simulation models in this dissertation, a linear elastic material model is chosen. The reason for choosing such simplification is this: for all load conditions that are considered (i.e. fixturing/clamping, handling, assembly) an elastic deformation is presumed. If the mechanical stress in any condition exceeds the elastic capabilities of the material the part would be permanently deformed. Such plastic deformation is not acceptable for any process that is considered in this dissertation. Badly shaped components that would permanently deform during assembly are expected to be sorted out by prior quality inspections.

3.5.2 Verification with Non-Over-Constrained Fixtures

In addition to verifications that already were proposed in [2], this section shows a verification that is performed with non-over-constrained fixtures but using the extended method, presented in Quotation 3.2. In [2] it was stated, that the first loop of the algorithm is necessary for parts with large deflections to reduce run time and undesired behavior. Unfortunately, every iteration performed by Loop 1 does trigger two simulation runs, which makes this

Table 3.1 Properties of different simulation models used for verification purposes.

Geometry	Sheet metal	Sheet metal	Car roof	Engine hood
Material	Aluminum	Steel	Steel	Steel
Density	2.7 $\frac{g}{cm^3}$	7.85 $\frac{g}{cm^3}$		
Young's Modulus	70 GPA	210 GPA		
Poisson's Ratio	0.3			
Material Model	linear elastic			
Thickness	0.5 mm	0.75 mm	0.7 mm	0.7 mm/2 mm
Mesh Elements	15,000	15,000	169,378	365,152

loop very costly in terms of computational times. Therefore, this verification investigates the behavior of the extended algorithm for large deflected parts.

To simulate a worst-case scenario, the two weakest parts from Table 3.1 are chosen for this verification. These are the piece of sheet metal made from aluminum and the car roof. The verification is executed as follows. First, the simulation model with nominal geometry is used to calculate displacements when the part is placed on a certain fixture. The fixture layout and orientation of the part are chosen such that the influence of gravity is maximal but without deforming the part permanently. The resulting geometry is exported and serves as input for the gravity compensation algorithm. Forces and tensions from the simulation result are neglected - only resulting geometry is used for the compensation. The compensation algorithm then calculates the GF-shape which can be directly compared to the nominal shape.

The results of this verification are shown in Figure 3.6. The initial setup shows the nominal geometry for each part together with the used boundary restrictions of the fixture. In the middle column, gravity is applied perpendicular to the image plane. The resulting shape compared to nominal geometry is shown in colors. The sheet metal part shows deflections up to 5 mm while the car roof deflects up to 15 mm. The third column shows the direct comparison between the nominal geometry (ground truth) and the output of the compensation algorithm. Only minor differences below 0.02 mm can be observed after 16 iterations for the piece of sheet metal and 15 iterations for the car roof.

3.5.3 Verification with Over-Constrained Fixtures

To verify the performance of the extended gravity compensation method for over-constrained fixture setups, a verification analog to Section 3.5.2 is performed. The used parts are the

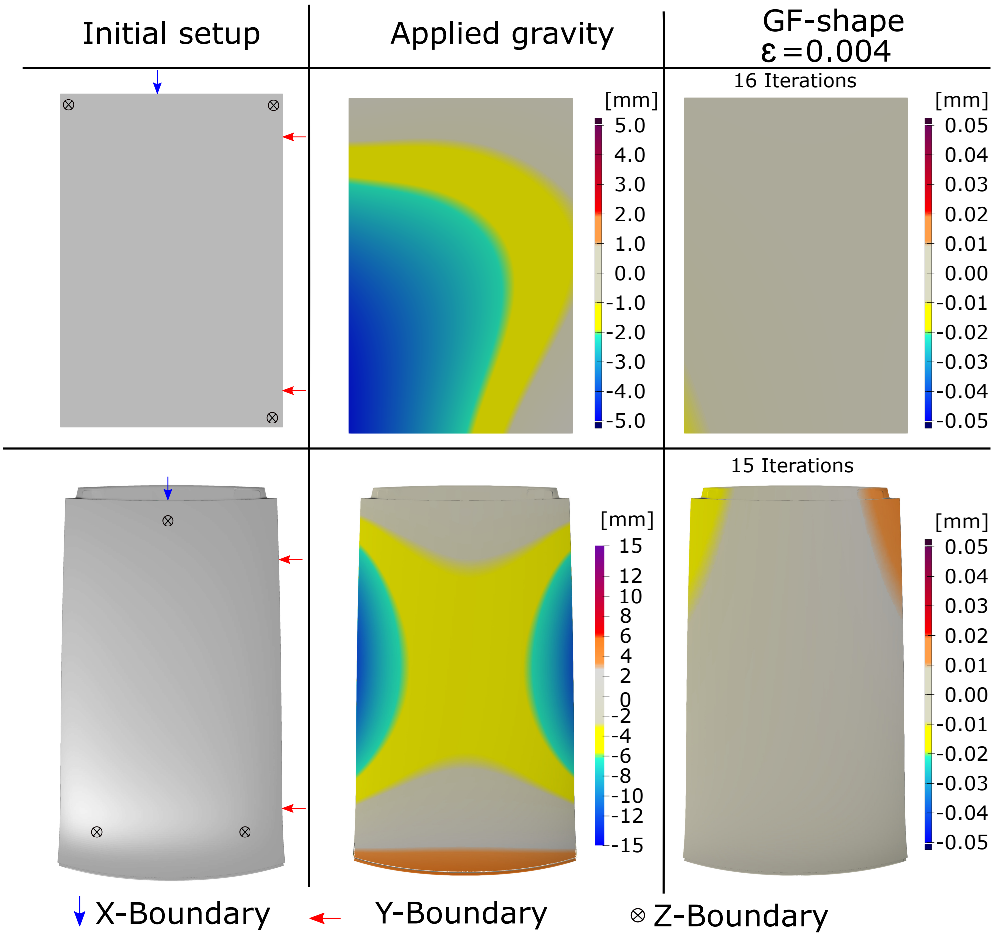


Fig. 3.6 Verification results for non-over-constrained fixture setups. Top: simple sheet metal made from aluminum; Bottom: car roof. Left Column: nominal CAD-geometry and locator layout. Middle column: parts deflection when applying gravity, compared to nominal geometry. Right column: calculated GF-shape based on geometry from middle column compared to nominal geometry.

simple piece of sheet metal made from steel and the engine hood, see Table 3.1. Workflow and figures are analog to the verification with non-over-constrained fixtures. For the engine hood, this fixture layout is similar to real measurement fixtures used in industrial applications. The verification results are shown in Figure 3.7. What changes with over-constrained fixtures, is that additional information must be acquired during measuring (second column). For these verifications, the force that acts on the over-constraining support F_{z4} is used, which is obtained from the first simulation output. The fourth support is modeled as an external force in the simulation model that serves as input for the gravity compensation method. To make the verification more challenging two aspects were set to not ideal conditions: (a) locator positions in z-direction for the sheet metal part are not in-plane, and (b) input GF-shape of the engine hood does deviate from the nominal geometry where locator positions are defined on. The last column again shows the result of the algorithm compared to the initial shape (ground truth). No relevant errors can be observed for both parts.

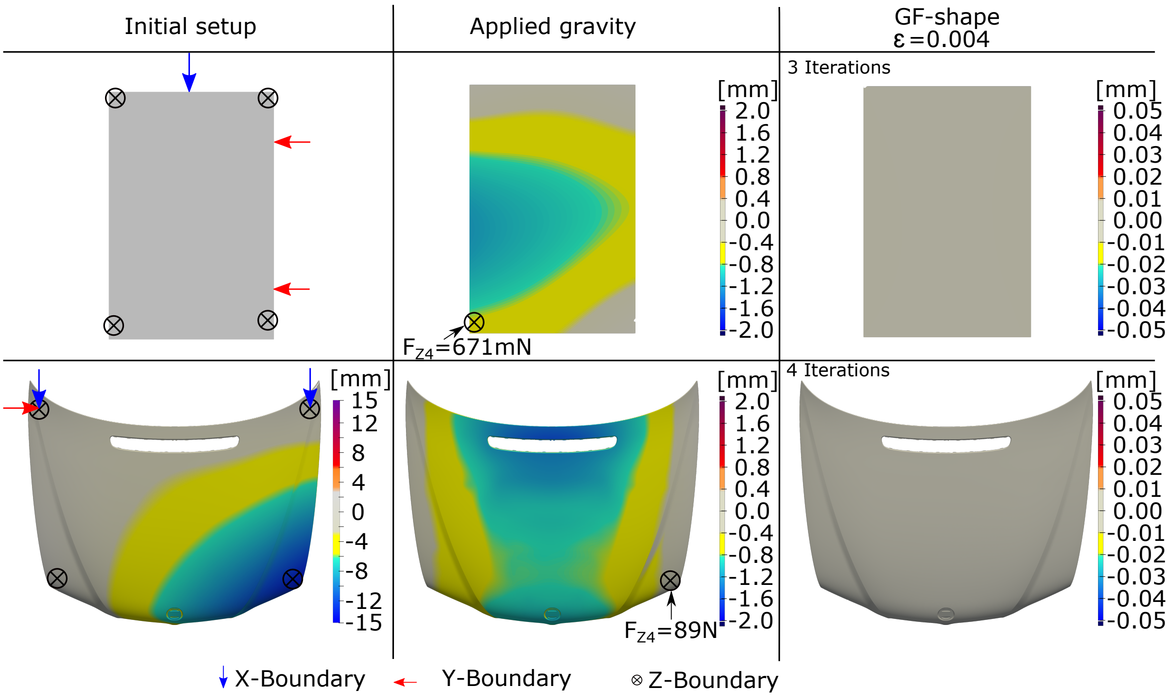


Fig. 3.7 Verification results for over-constrained fixture setups. Top row: simple sheet metal made from steel; Bottom row: engine hood. Columns are analog to Figure 3.6. In the middle column, the force on the over-constraining support is shown.

3.5.4 Run Times and Convergence Behavior

Run Times

All computational times were measured on the same computing hardware listed in Table 3.2. Run times were measured for each iteration, averages and total times are listed in Table 3.3.

Convergence Behavior

To evaluate the convergence behavior of the gravity compensation algorithm, the output of every iteration is compared to the initial GF-geometry. Differences are measured with the RMS[1]-error, see Equation 3.1. The RMS-error is evaluated every iteration and plotted for all four verification experiments in Figure 3.8. It can be seen that the calculations with

[1] Root-Mean-Squares

Table 3.2 Computing Hardware for measuring run times.

Component	Specification
CPU	AMD Ryzen 5 1600 6C/12T
GPU	NVIDIA GeForce GTX 1050 Ti (4GB)
RAM	16 GB DDR4
HDD	WD Black 2TB Read: 140Mb/s Write: 142MB/s
Solver	Dassault Systèmes ABAQUS 2019
OS	Windows 10 LTSB

Table 3.3 Measured run times for the performed verification.

Geometry	Sheet metal	Sheet metal	Car roof	Engine hood
Type of fixtures	N-OC[1]	OC[2]	N-OC	OC
Number of iterations	16	3	15	4
Time per iteration	55.8 s	44.3 s	139 s	387.25 s
Total time	893 s	133 s	2081 s	1549 s
Mesh Elements	15,000	15,000	169,378	365,152

non-over-constrained fixtures took about four times more iterations to reach the required goal of $\varepsilon < 0.004$ mm. In contrast, the curves of experiments with over-constrained fixtures converge much faster towards a small RMS value.

$$RMS = \sqrt{\frac{\sum_{i=1}^{n} {d_i}^2}{n}} \tag{3.1}$$

Equation 3.1: Calculating RMS error. Distance d_i is the distance error per node and n the number of nodes.

3.5.5 Benchmark with Mapping Approaches

To benchmark existing mapping approaches with the gravity compensation algorithm presented in this chapter, an additional experiment with simulated data was performed. Figure 3.9 shows the experiment setup and results. The part that was used is the same engine hood from the verification in Section 3.5.3. The upper left picture shows the nominal geometry and the applied fixture setup. When applying gravity, the deflections shown in the upper right picture, occur. This simulated displacement field is used in mapping approaches for

[1] non-over-constrained

[2] over-constrained

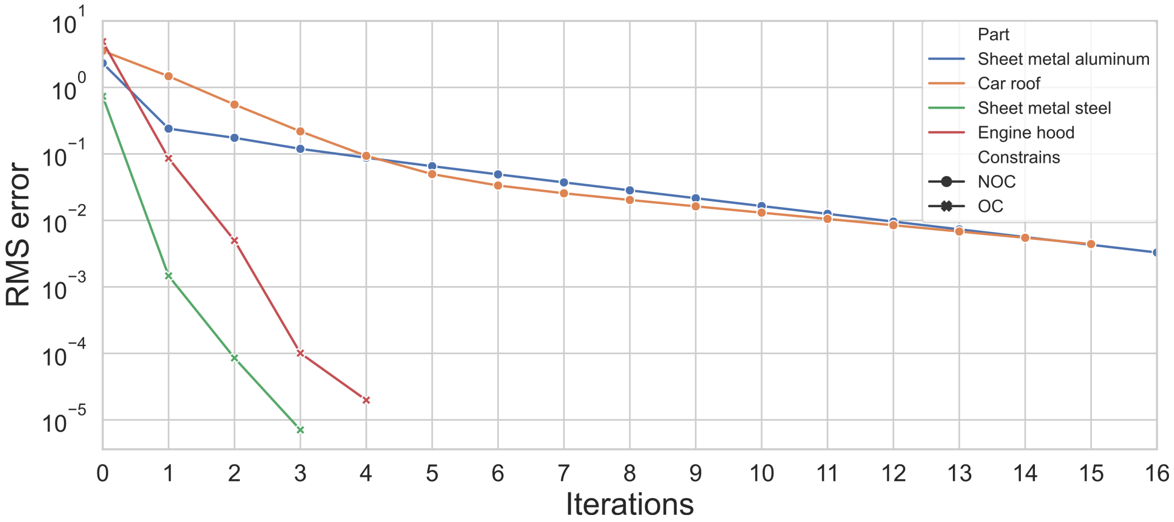


Fig. 3.8 Investigation of convergence behavior. RMS-error value is plotted over iterations for performed verifications. Markers distinguish between experiments with over-constrained and non-over-constrained fixture setups.

applying corrections to measured point clouds. A pseudo input for such measured point cloud is created in the second row of Figure 3.9. The “actual shape” in GF-state was created by applying the gravity compensation algorithm to the nominal shape in GF-state. This creates a worst-case scenario where the actual loaded shape (right picture in the second row) does exactly match with the nominal shape.

When now mapping the pre-calculated displacement field from the top and applying inverse displacements to the loaded actual shape, the actual GF-shape should result. As it can be seen in the lower picture of Figure 3.9 the error resulting from this approach gets high as ± 0.2 mm, where the error of the gravity compensation method can directly be evaluated from the actual shape in gravity loaded state, compared to the nominal shape (right side middle row). This error is below ± 0.01 mm.

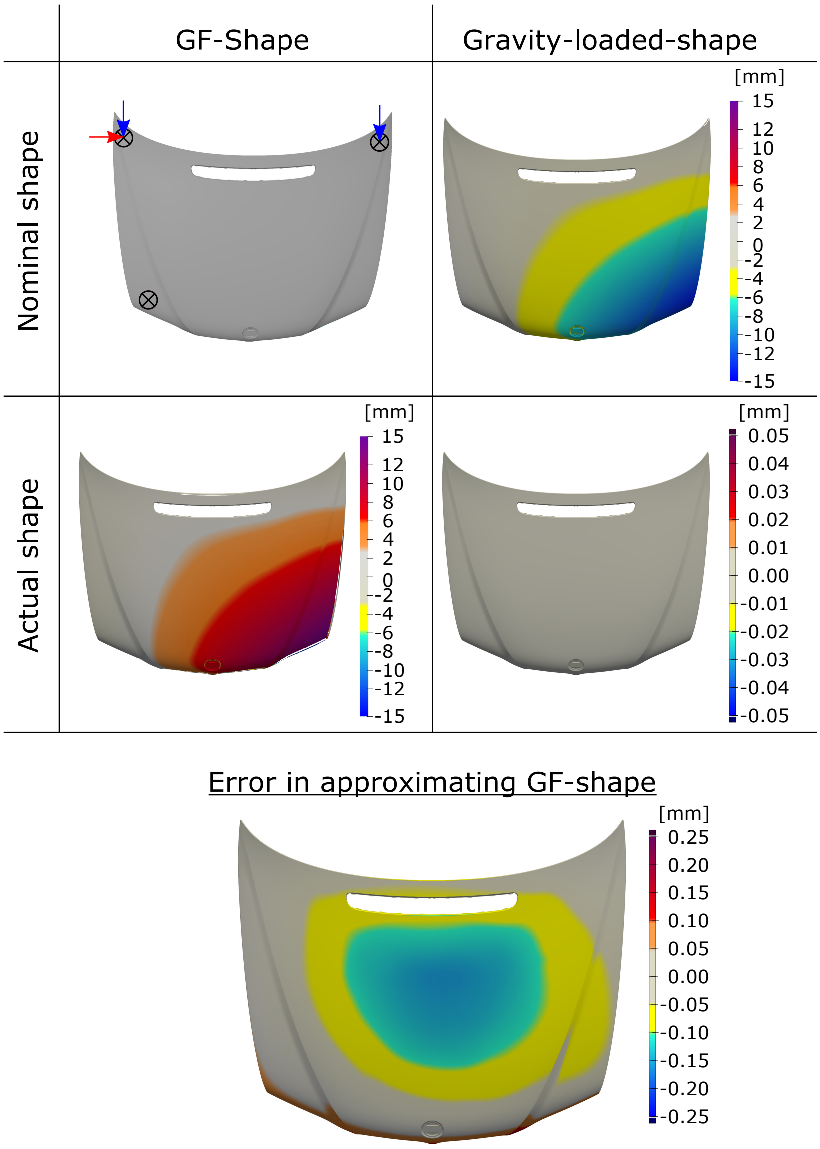


Fig. 3.9 Results of benchmarking a mapping approach to compensate for deflections caused by gravity. Top row: nominal shape in GF-state and gravity loaded shape (free state). Bottom row:“actual shape” created by applying the presented gravity compensation approach. When applying gravity the resulting shape is identical to the nominal GF-shape. Bottom: Comparison between actual shape and inverse displacement field from top right applied to nominal geometry.

3.6 Validation

To show the performance of the gravity compensation algorithm under real-world conditions, a validation based on experimental data was performed. This step is crucial, as many approaches from the literature do only perform verifications based on artificial data, but challenges that occur when applying concepts to real-world problems often are not addressed.

3.6.1 Experimental Setup

For validation, the scan setup described in Section 1.4 is used. The hardware setup also contains a specimen, fixture setup, precision scale, and measurement computer, see Figure 3.10. The used specimen is a simple piece of sheet metal, made from steel with the dimensions of 200 mm x 300 mm x 0.75 mm. The geometry has the same dimensions as the simple part, simulated in Section 3.5.3. The base of the fixture setup is made of aluminum profiles, where different locators are mounted. For restricting the part in direction of gravity, three fixed supports are realized by M8 x 60 mm DIN 912 screws. To achieve a single contact point with the specimen, each support is equipped with an 8 mm precision steel ball from ball bearings, which is placed on the in-bus socket of each screw. The whole fixture is leveled by adjusting each support in height using the threads of the screws. In-plane movement is restricted by three precision rods that are also attached to the aluminum profiles. These three supports and three precision rods are used as reference frame for aligning measurements with nominal data. The hardware setup also contains a fourth support (F_{Z4}), placed on a precision scale, which is removable. By this, the fixture can easily be adjusted to over- / and non-over-constrained setup. The scale is used to determine the force that pushes onto the support.

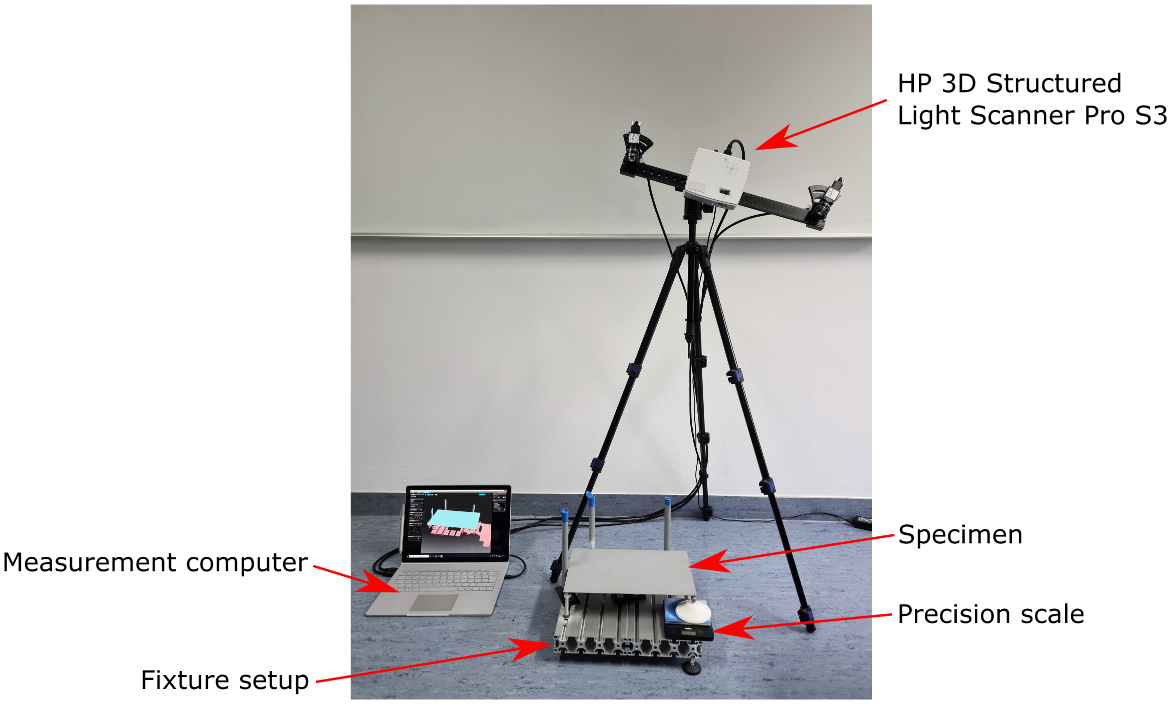


Fig. 3.10 Experimental setup that was used to perform validations. (This figure has been published in [3])

3.6.2 Workflow Validation

The problem with validating any GF-shape is that it cannot be validated directly. This is because a gravity-free environment during measurement cannot be achieved on earth. Thus, an indirect validation is necessary. The workflow of this validation is illustrated in Figure 3.11. The validation is based on the assumption, that there can only be one valid GF-shape. When acquiring and post-processing the same specimen with different fixture conditions, the same GF-shape must result from the gravity compensation algorithm. In Figure 3.11 two post-processing pipelines for the same specimen are shown: (a) post-processing of the specimen measured on non-over-constrained fixture in blue and (b) post-processing of the specimen measured on over-constrained fixturing in green. Both pipelines are analog. Starting from the specimen placed on the fixture, a 3D-scan is performed to acquire the part's geometry as a point cloud. In the over-constrained case, also the force acting on the over-constraining support is measured. From each point cloud an FE-simulation model is derived by fitting a rectangular B-spline[1] surface with the dimensions 200 mm x 300 mm to the point cloud. This surface representation then is meshed and material properties and boundary conditions are added to complement the FE-model. The two FE-models serve

[1] Bayesian Spline

as input for the gravity compensation algorithm which calculates one GF-shape per model. These GF-shapes can be compared directly, any differences above measurement uncertainty indicate errors.

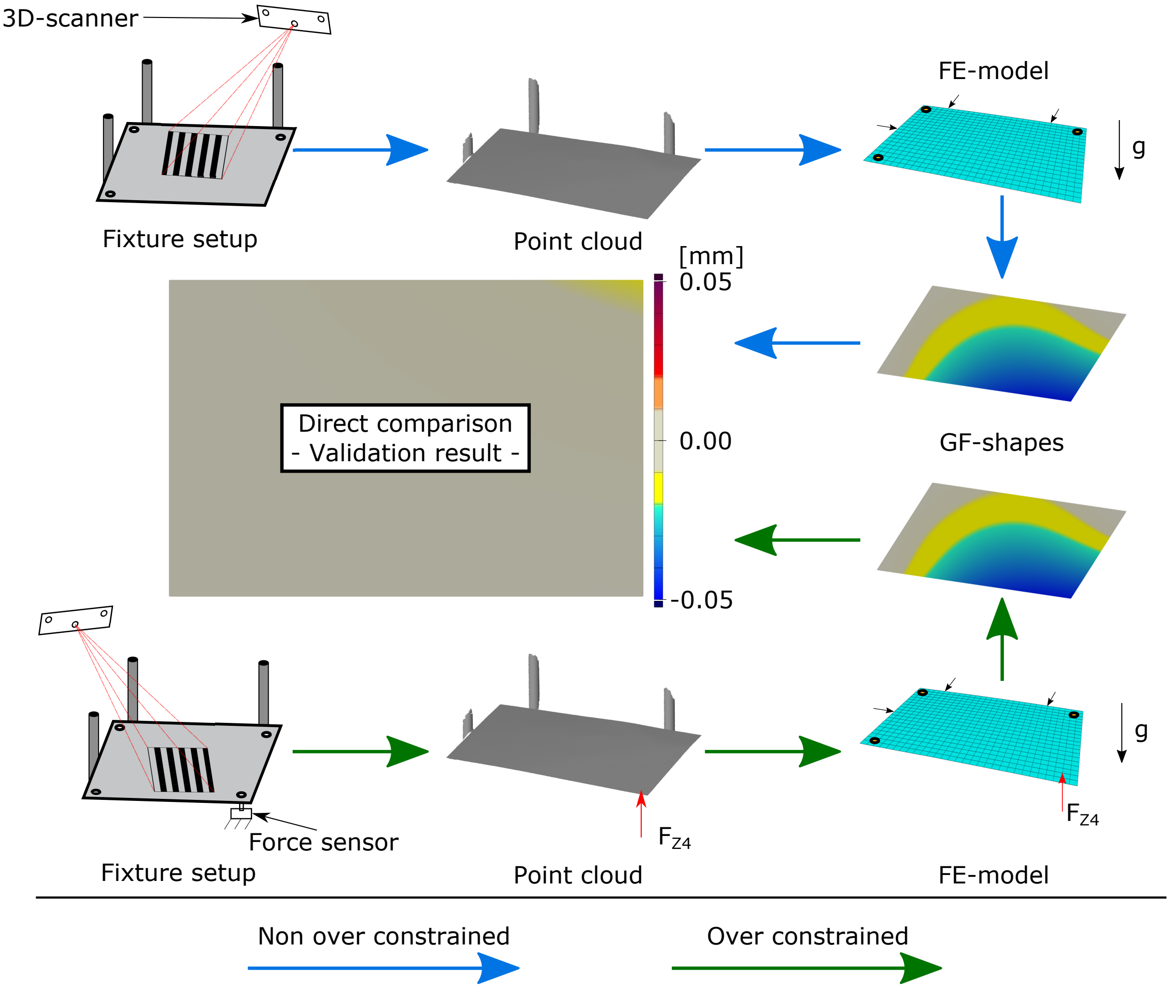


Fig. 3.11 Validation Workflows. Blue: specimen is placed on a non-over-constrained fixture and scanned. From the resulting point cloud, an FE-model is derived which serves as input for the gravity compensation algorithm. Green: Analog workflow for the same specimen on over-constrained fixture setup. At the end of the pipelines, both resulting GF-shapes are directly compared.

3.6.3 Preparations

Before performing the actual validation, several preparation steps are made. These steps are: (a) preparation of specimen and fixture and (b) tuning of material parameters.

Preparation of Specimen and Fixture - To prevent disturbances during 3D-scanning, reflective surfaces must be covered with mat finishes. This is achieved by applying a light grey mat spray paint to the specimen and reflective surfaces of the fixture. To ensure that the spray paint does not influence parts' weight or thickness both quantities are measured. The measured weight is with 353.8 g only 0.55 g above 353.25 g which are estimated by the simulation software. The error of 0.55 g is negligible for this validation. When assuming an even distribution of the weight error, and converting it to a surface load, this would be 8.99e-6 $\frac{N}{cm^2}$.
To ensure an even thickness distribution, the thickness of the specimen was probed at 20 points evenly distributed across the surface. All measured points did not vary above the range of ± 0.01 mm which matches with the measurement uncertainty of the used gauge.

Tuning of Material Parameters - Although a simple piece of sheet metal is used, a tuning of material parameters is necessary to fit the stiffness of the simulation model with the actual parts stiffness. This is necessary because of the manufacturing process of sheet metal. Typically sheet metals are produced by cold rolling. Due to the rolling process, Young's Modulus is influenced depending on evaluation direction relative to the rolling direction. Investigations regarding this phenomenon can be found in [115] and [116]. When having different Young's Moduli dependent on direction, an orthotropic material model must be presumed. The in-plane stress-strain relations for thin shell structures are given by Equation 3.2. To model transfer shear deformation in shells, ABAQUS also requires G_{13} and G_{23}, see [117]. To define all engineering constants (E_1,E_2,υ_{12},G_{12},G_{13},G_{23}) for the present specimen, only few non destructive methods are available. One possibility is using IET[1] which makes use of the parts vibration behavior to estimate material properties. Although there is a commercial product Resonalyser® from the company "BYTEC BV" that is designed for determining material properties for thin sheet metals, different specimens that fulfill certain requirements are necessary to perform the estimation. Due to the lack of specimens and measurement equipment, the estimation of material parameters is carried out with a parameter study. To make the study comparable with measurement values, the stiffness of the part is evaluated with a load-displacement curve for the over-constraining support (see Section 3.6.1). To

$$\left\{\begin{array}{c}\varepsilon_1\\ \varepsilon_2\\ \gamma_1\end{array}\right\} = \begin{bmatrix}\frac{1}{E_1} & \frac{-\upsilon_{12}}{E_1} & 0\\ \frac{-\upsilon_{12}}{E_1} & \frac{1}{E_2} & 0\\ 0 & 0 & \frac{1}{G_{12}}\end{bmatrix}\left\{\begin{array}{c}\sigma_{11}\\ \sigma_{22}\\ \tau_{12}\end{array}\right\} \tag{3.2}$$

Equation 3.2: Stress-strain relations for in-plane orthotropic material behavior. [117]

[1] Impulse Excitation Technique

measure this force-displacement curve, different equally spaced heights of F_{Z4} are dialed in. For each height, the force pressing down on the scale is measured. The same experiment is carried out virtually by simulating the force pressing on F_{Z4} for different heights. Figure 3.12 shows the comparison between simulated and measured forces. For the simulated curve, the isotropic material model was used. Both curves show the same linear relationship between force and displacement. However, the pitch of both curves which is related to the stiffness of the part is different. To find a suitable set of material parameters, a study was performed by running 2000 simulations with different combinations of parameter values. For each simulation result, a force-displacement curve analog to Figure 3.12 is calculated and compared with the measured results. The parameter set which creates the best fitting pitch is selected as the best match. The varied material parameters and ranges are listed in Table 3.4. All 2000 simulations are equally distributed in the parameter space by using LHS[1] [118]. The result of this study - the best fitting set of parameters - is also listed in this table. This set of best-fitting parameters is used further for the validation.

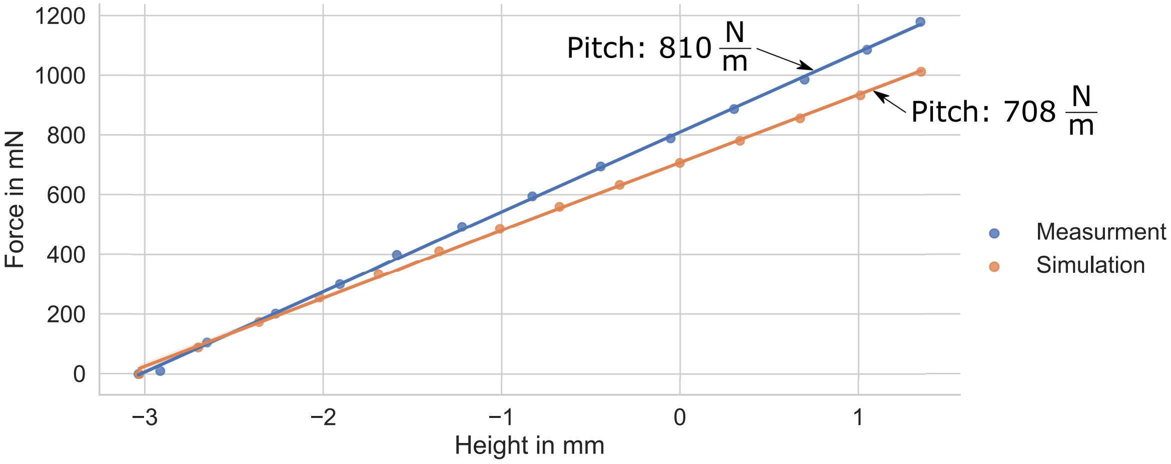


Fig. 3.12 Investigation of parts stiffness compared to nominal simulation model. Graph shows measures of force pressing on support F_{Z4} for different support heights. Height is equal to zero when F_{Z4} is in-plane with remaining supports.

3.6.4 Results

An experiment according to the workflow presented in Section 3.6.2 was performed. As material parameters for the simulation model, the best matching set of parameters listed in Table 3.4 was used. The result of this experimental validation is shown in Figure 3.13. The top row shows the scanned over- / and non-over-constrained situation for the same specimen

[1] Latin-Hypercube Sampling

Table 3.4 Parameter Study for finding best fitting set of material parameters.

Material Parameter	Tested Range	Best Match
E_1	189-245 GPa	211 GPa
E_2	189-245 GPa	215 GPa
υ_{12}	0.1-0.4	0.29
G_{12}	72-100 GPa	93 GPa
G_{13}	72-100 GPa	95 GPa
G_{23}	72-100 GPa	93 GPa

compared to the nominal geometry. As nominal geometry, the plane, spanned by the supports is assumed. The position of supports is marked in the pictures. In the non-over-constrained case, differences of up to 2 mm can be observed, while differences in the over-constrained case only are high as 0.3 mm. The second row of the figure shows a comparison between measurement data and the smooth fitted B-spline surface that is used to create the FE-model. Only minor approximation errors can be observed except towards the edges of the part. In these areas, higher approximation errors of about ± 0.05 mm occur. The third row shows the result of the gravity compensation algorithm compared to the nominal plane. These two resulting geometries should match exactly under ideal conditions. A direct comparison is given in the bottom picture of Figure 3.4. Only differences in the range between -0.079 mm and +0.034 mm occur. Differences with these extreme values, are only observable at single points that are located on the edge of the part. Expressed in percent, 99.3 % of all values shown in the distance map are within the margin of ± 0.05 mm, while only 0.066 % are outside. 0.70 % of the nodal values are classified as mismatch and have no assigned distance value.

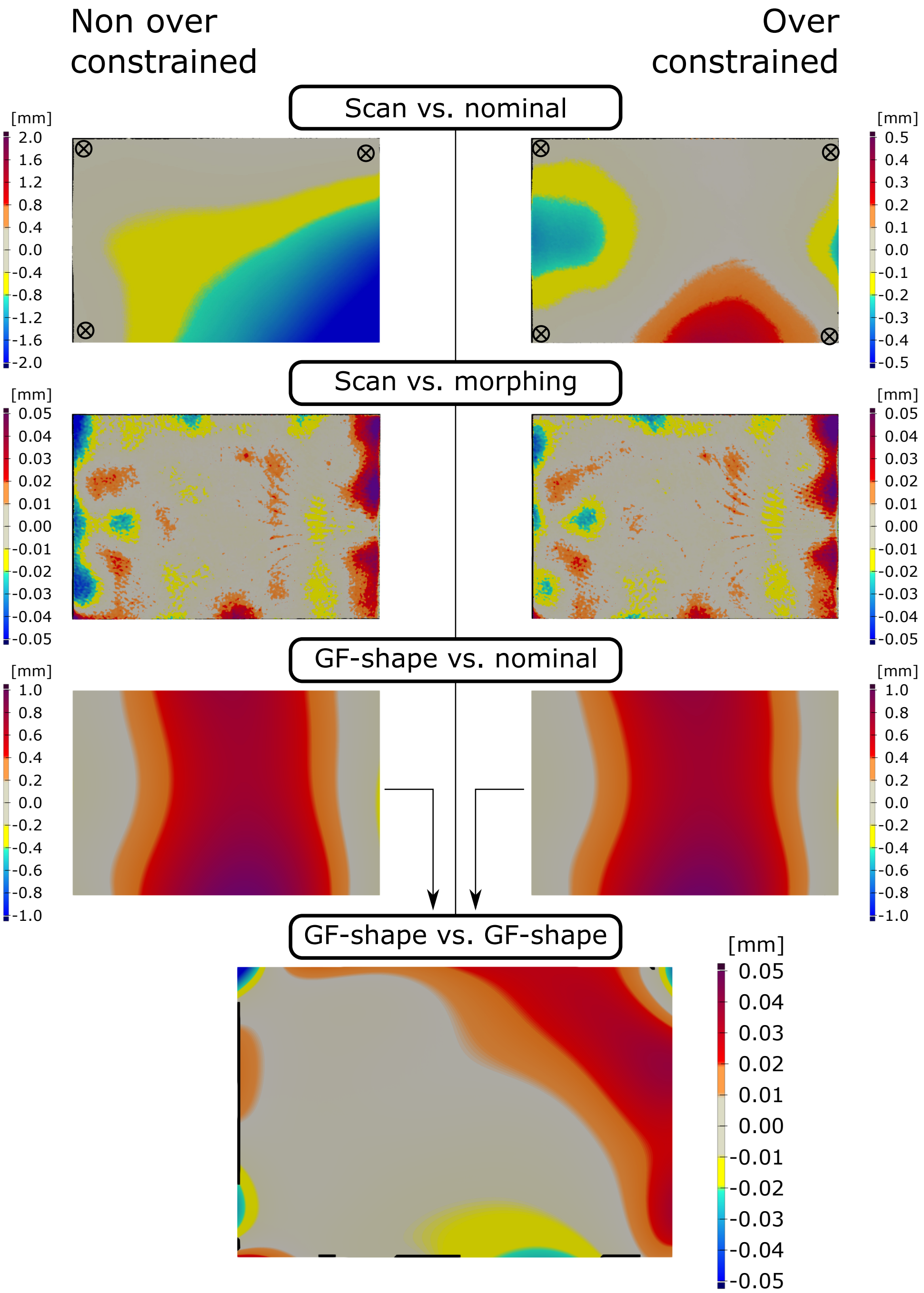


Fig. 3.13 Results of validation with scan data. Top row: same specimen measured with two fixture layouts. Acquired point clouds are compared to nominal geometry (ideal plane). Middle row: fit of the smooth surface compared to measurement. Bottom row: GF-shape calculated by the compensation algorithm. Bottom: direct comparison between both GF-shapes, differences indicate errors.

3.7 Discussion

This chapter presented a method that precisely compensates gravitational influence of measured sheet metal parts. Different experiments were performed to investigate the performance of the algorithm in terms of precision, flexibility, and computational cost. The achieved results are discussed in the following with respect to applications in smart assembly processes.

Verification with Simulation Data - The results of the verifications have shown that under ideal conditions, the gravity compensation algorithm does work as desired for different geometries, materials, and fixture setups. In Figure 3.6 extreme examples are shown, where very weak parts are supported unfavorably by only 3 supports. Using this orientation and fixture layout, gravity causes large deflections. Nevertheless, the algorithm is capable of compensating the occurring deflections with errors below ± 0.015 mm for the car roof and below ± 0.011 mm for the simple sheet metal. The result can always be improved further by performing additional iterations. However, the results are well within the pre-defined error margin of ± 0.05 mm and are satisfying for industrial applications. The results of the experiment performed with over-constrained fixtures are shown in Figure 3.7. This experiment shows the same simple sheet metal but different material, as well as a whole assembly of an engine hood being processed. To prevent lucky strikes, the fixture layout of the sheet metal was chosen to be not in-plane and the input geometry for the engine hood is deformed such that the supports of the fixture are loaded unbalanced. Still, the algorithm does perform great and the results show even fewer errors of below ± 0.01 mm for both cases. This can be explained by the information gained due to the additional force of the fourth support that was added to the simulation model.

When looking into run times (Table 3.3) and convergence behavior (Figure 3.8), it can be observed, that the total computation time of the experiments with over-constrained fixture layouts is significantly smaller. This is explained by the much faster convergence behavior, needing fewer iterations to achieve better approximation. Concerning applications where real-time support is needed, the achieved run times are still too slow. However, using over-constrained setups with additional force information can speed up the process significantly. Also, these verifications have used non-optimized high-fidelity models to eliminate any error caused by mesh resolution. For real applications, coarser models or surrogate models can be employed to speed up the calculation of GF-shape.

Within the verifications, also a comparison to an existing approach was made. The results of comparing the presented approach to the commercially available mapping approach are shown in Figure 3.9. State-of-the-art mapping approaches presume similarity between the behavior of nominal- and actual geometry. The presented example shows a real-world scenario where this assumption does cause errors up to ± 0.2 mm which is 2000 times higher than the approximation errors of the presented approach. The error of the mapping approach might vary as it is highly dependent on the part's geometry. Also, individual differences between actual part and nominal geometry do affect the results. In contrast, the gravity compensation method presented in this dissertation is unaffected by these variables.

Validation with Measurement Data - To validate that the presented workflow and algorithm can be applied to real measurement data, an experimental validation was performed. The most challenging part was to tune in the properties of the simulation model such that the stiffness of the virtual part does exactly match with the behavior of the real part. This step is crucial because a wrong stiffness of the simulation model will inevitably result in wrong results. However, if the manufacturing process of the part is well controlled and the parts stiffness does not vary between individual parts, this step needs only be performed once.

The results of the validation are shown in Figure 3.13. The figure shows initial measurement data, interim results, and final comparison between the results of the workflows described in Figure 3.11. The first critical step after scanning is the conversion to an FE-model. To ensure that errors made by the used morphing software (Autodesk Alias) are minimal, the resulting surface is compared to the raw point cloud. In both reconstructions, higher errors mainly occur towards the edges of the part. This can be explained by the used structured light scanning which becomes noisy near sharp edges. As deviation patterns of both scans are very similar, the approximation error should not affect the final results. The final validation result (bottom of Figure 3.13) does show only minor errors below ± 0.05 mm for most of the part. Only 0.066 % of the nodal values exceed this margin with a maximum error of 0.08 mm. When having in mind that this comparison is based on two different scans and processing pipelines, this result is astonishingly good. As the 3D-scanning system already has an uncertainty of ± 0.05 mm, errors on this scale cannot be analyzed further.

DT and Self Compensating Assembly Line - The experiments in this chapter have pointed out that the influence of gravity must be considered to be able to derive a geometrical DT from acquired measurement data. Also, a method and implementation for solving this problem were developed, verified, and validated. Applying the presented method for real processes, computation times must be lowered significantly. This could for instance be achieved by building a lightweight simulation model. Unfortunately, the most time-consuming processing step is the conversion of the measured point cloud to an FE-simulation model. This step was carried out by manually morphing the CAD representation of the nominal geometry to fit the actual scan data. For in-line usage, this model generation step must be automated.

Core References of this Chapter

[2] Felix Claus, Hans Hagen, and Bernd Hamann. Calculating the gravity-free shape of sheet metal parts. *The International Journal of Advanced Manufacturing Technology*, 113(11-12):3401–3417, 2021

[3] Felix Claus, Bernd Hamann, and Hans Hagen. Calculating stress-free shapes of sheet metal parts measured with over-constrained fixtures. NAFEMS, 2021

Chapter 4

Automated FE-Model Generation from 3D-Scan Data

4.1 Introduction

Motivated by the outcome of Chapter 3, this chapter presents an automated mesh morphing approach for generating geometrical DTs from 3D-scan data. This chapter is based on core-reference [6]. To derive a DT from 3D-scanned point clouds, a process is required that generates, or converts an FE-simulation mesh such way that the geometry of the mesh matches the measured geometry. A geometrical DT can be used to perform precise predictions by preserving the correct stiffness that is affected by geometrical properties. For instance, dents or buckles can stiffen or weaken the structure of the part in an undesired way resulting in different behavior compared to nominal geometry. Due to manufacturing variability, surface defects can be of different types, sizes, locations, and occurrences. By considering actual geometric properties, the output precision of the DT can be increased significantly. Unfortunately, the step of generating an FE-mesh that preserves measured geometry is not straightforward. Often, manual work is required to ensure that mesh quality criteria are fulfilled. Also, 3D-scans can not cover areas that are inaccessible for the scanning system. This results in incomplete measurements which is very challenging for automated approaches. As the step of generating a geometrical DT is extremely important for the concept of this dissertation, mentioned challenges must be addressed.

Solving the problem of generating a geometrical DT from scan data in an automated manner is challenging on different levels. Besides handling incomplete measurement data and pre-

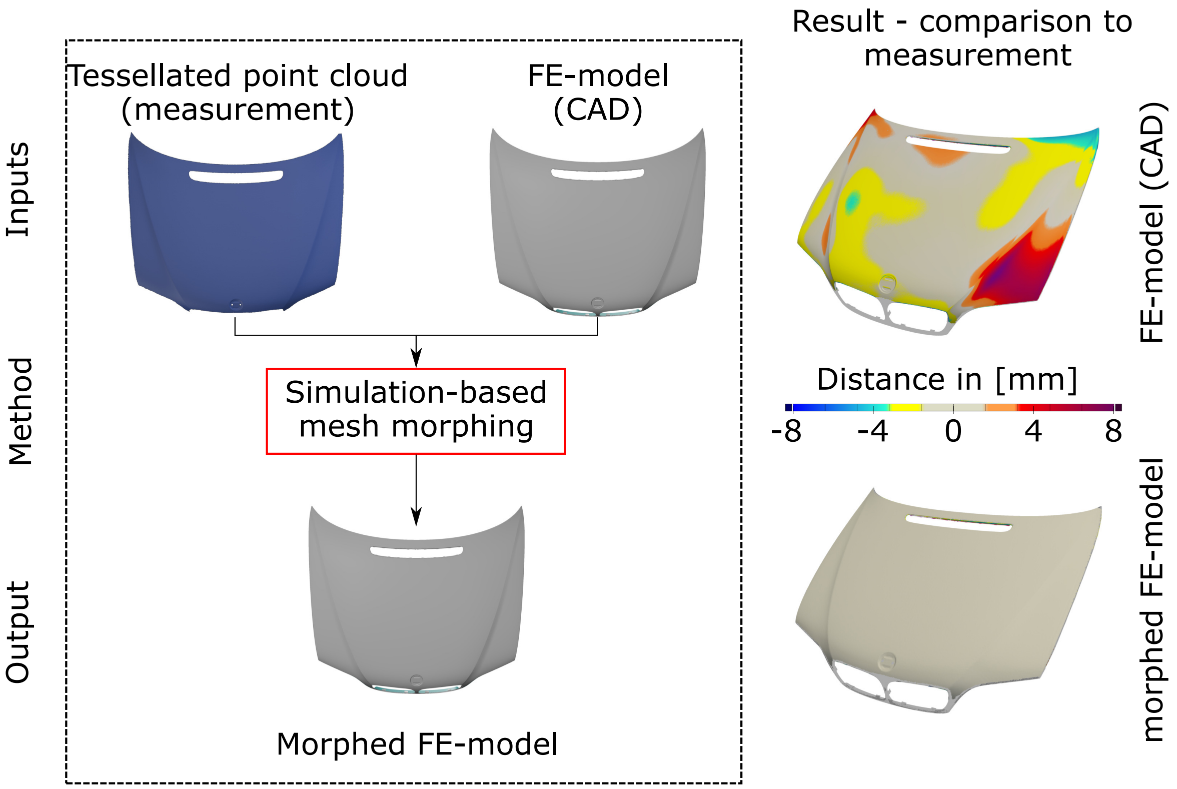

Inputs
Method
Output
Tessellated point cloud (measurement)
FE-model (CAD)
Simulation-based mesh morphing
Morphed FE-model
Result - comparison to measurement
FE-model (CAD)
Distance in [mm]
-8
-4
0
4
8
morphed FE-model

Fig. 4.1 High level description of the mesh morphing approach. Inputs: FE-model based on nominal geometry and tessellated point cloud. Output: morphed FE-model, preserving geometry of measured point cloud. Right: preview of input and output FE-mesh compared to measurement. (This figure has been published in [6], changes: color map)

serving element quality for correct simulation results, the size of data being handled has a high impact on the processing times and also on the needed computational resources. For inline usage, this step is very critical because of the potential time cost as the size of one scan of a single part easily exceeds gigabytes. On the other hand, 3D-scans provide a high-resolution 3D-representation of the scanned object, which contains detailed information about the actual geometry. Extracting relevant information from scans while preserving acceptable processing times is highly challenging. In the following, a first approach for automatically morphing FE-meshes to match scanned data is described. A high-level description of this approach is given in Figure 4.1 using the engine hood that was introduced in the previous chapter. As inputs, a tessellated point cloud from the 3D-scanner and an FE-model based on nominal geometry are used. The mesh morphing approach does deform the FE-mesh to match with the measured geometry. On the right side of the figure, a preview of the output is shown. The approach is verified using artificial data and also validated with actual measurement data.

This chapter is structured as follows: Section 4.2 provides a literature review of the relevant topics and state-of-the-art approaches. In Section 4.3 the used methods are presented as excerpt from [6]. In Section 4.4 the method is verified, followed by a validation with experimental data in Section 4.5. In Section 4.6 the run times and convergence of the used algorithm are assessed. Last, Section 4.7 discusses the method and results in the context of smart assembly systems.

4.2 Background

To generate a simulation mesh for geometrical DTs, different approaches are available in the literature. These approaches can be clustered by three principles: (a) RE[1], (b) direct mesh generation, and (c) morphing-based approaches. Each principle is briefly explained, and advantages and shortcomings are highlighted. Also, the approach presented in this chapter is put into perspective to approaches from the literature.

4.2.1 Reverse Engineering

An intuitive way of creating a simulation model from measured point clouds, is to use RE. An introduction to RE methods for geometric models is given in [119]. By using RE, a CAD representation is obtained which can be processed further towards an FE-model. To create such CAD representation, surface patches are fitted to the point cloud. This step was performed in the validations presented in Section 3.6. RE is used in many different applications, for instance:

• Manufacturing	[120],[121]
• Medical	[122],[123]
• Civil Engineering	[124],[125]
• Design Processes	[126]

Working with RE is very time-consuming and requires immense amount of manual work. Especially, for complex geometries, a well-suited division of surface patches cannot be carried out automatically nor unsupervised. To speed up RE processes, an approach that dynamically performs a reconstruction of B-spline surfaces during data acquisition to guide the further acquisition process, is presented in [127]. In [128] a multi-sensor fusion approach

[1] Reverse Engineering

for RE applications is presented that aims to increase speed and precision at the same time.

RE methods are a great choice for creating or adjusting CAD models based on 3D-scan data. However, for automated and unsupervised processes, RE methods are still not flexible enough. In addition, a modeling step has to be performed to obtain an FE-model from the RE result. This modeling step also does not work without manual interventions. Because of these drawbacks, RE is not applicable for use in self-compensating assembly lines.

4.2.2 Direct Mesh Generation

To skip the costly process of RE in the modeling pipeline, direct mesh generation methods are attractive. These methods are capable of creating an FE-mesh directly from the acquired point clouds. Unfortunately, creating a high-quality mesh from actual scan data is challenging due to various influences, see [129]. In [130] this problem was tackled by applying bubble packing (see [131]) to a coarsened point cloud. The achieved mesh was used to predict the post assembly shape of thin components. More recently, a voxel-based approach was published in [132] that is capable of creating high-quality quadrilateral elements from scan data. What all direct mesh generation methods have in common, is the fact that full coverage of the scanned object is presumed. This is not always possible in real-world applications, which restricts the applicability of these kinds of approaches.

4.2.3 Morphing

Another way of creating surfaces or meshes that reflect real measurement data is morphing. Morphing can be applied to a surface- but also to mesh-representations of a geometry. Typically, a model that is based on nominal geometry is used as the initial model. This initial model is deformed smoothly by the morphing algorithm such that the resulting geometry matches with the scan data, see [133]. During this process, the topology of the initial model remains untouched. The branch of morphing approaches can be divided into two groups: (a) surface-based morphing approaches, and (b) mesh-based morphing approaches.

Surface Based Morphing Approaches - Surface-based morphing approaches use parametric CAD representations of the nominal geometry which are fitted to scan data. CAD definition can be coupled with FE pre-processing tools such that an FE-model can be automatically regenerated when CAD definition is changed. This makes surface-based morphing approaches suitable for generating DTs based on scan data. This principle is used in various applications for instance in the aerospace industry. In [134], unloaded compressor blades are morphed such the resulting geometry matches a temperature-loaded part. In [135] a parametric model of compressor blades is designed, that is used to create geometric DTs to perform the impact on the overall performance of the turbine, see [136].

Mesh Based Morphing Approaches - In mesh-based morphing approaches, a deformation field is directly applied to a meshed geometry. In particular, node positions are displaced by a smooth mathematical displacement field. This field can be calculated in different ways, for instance, based on handle positions of a convex hull which are blended to the morphing volume. A survey of mesh morphing methods can be found in [137] and [138]. Mesh-based morphing is commonly applied in early prototyping stages where one must test changes without updating the whole CAD representation, see [139]. Mesh-based morphing can also be used to create geometrical DTs. For example, in [140] a mesh-based morphing approach is used to generate simulation models based on deviations measured with CMMs. An issue with mesh-based morphing approaches is the impact on mesh element quality. A commonly used approach to maintain element quality during morphing is SAM[1]. This approach does introduce a spring stiffness, along each element edge. Similar to the SAM approach are FEM based mesh morphing approaches, where the elasticity problem is solved, resulting in a self arrangement of mesh elements, see [141].

Although both groups of morphing approaches are applicable for generating geometrical DTs, both principles have certain drawbacks. Surface-based morphing approaches are commonly not usable robustly such that manual intervention is inevitable. In mesh-based morphing approaches, element quality in large distorted areas must be handled. Available approaches like SAM or FEM-based are computationally expensive.

4.2.4 Resume

The review of methods from the literature has shown that multiple approaches can be used to create a geometrical DT from acquired point clouds. Unfortunately, almost all approaches are typically carried out such that manual interventions are necessary. Setting up a robust fully automated processing pipeline is not possible with any approach presented by the literature by now. Also handling incomplete measurement data becomes an issue for some approaches. To solve both problems, an FE-based mesh morphing approach [6] was developed which is presented in this chapter. The approach does classify as a mesh-based morphing approach and does use an FE-mesh based on nominal geometry as input. Iterative FE-simulation is used as

[1] Spring Analogy Model

kernel for calculating displacement fields that are applied to nodal positions. The resulting geometry matches with the target geometry, represented by a tessellated measurement.

4.3 Method

The method used in this chapter is excerpted from core reference [6], which describes the FE-based mesh morphing approach for surface meshes.

Quotation 4.1: Chapter 3 excerpted from [6]

3. Method

Our approach falls in the class of FE-based mesh morphing methods. As kernel for the displacement field calculation, we use an FE simulation. The definition of the simulation problem is crucial for describing the method. First, a high-level description of the approach is given. An overview of the processing steps is presented by discussing a pseudo code, see Algorithm 1. We also discuss the mathematical details of the various functions involved.

3.1 High Level Description

Figure 2 provides a high-level overview of the computational scheme. The basic idea of the FE-based morphing approach is to calculate distances between FE mesh nodes and a measured target geometry; these distances define an FE-load case. The displacement of the simulation output is used to adjust node positions of the FE-model. This process is repeated until the geometries of FE-model and measurement match. Most crucial is the calculation of the distance field and the placement of FE-boundary conditions. Figure 3 shows how the placement of FE-boundaries is carried out, based on a calculated distance field. Following common meshing terminology, the phrases “source” and “target” are introduced to differentiate between meshes, finite elements, and vertices belonging to different inputs. For the use case, the source mesh is related to the FE mesh representation of the “nominal geometry,” while the target mesh is a tessellation of the point cloud of the measured, actual geometry.

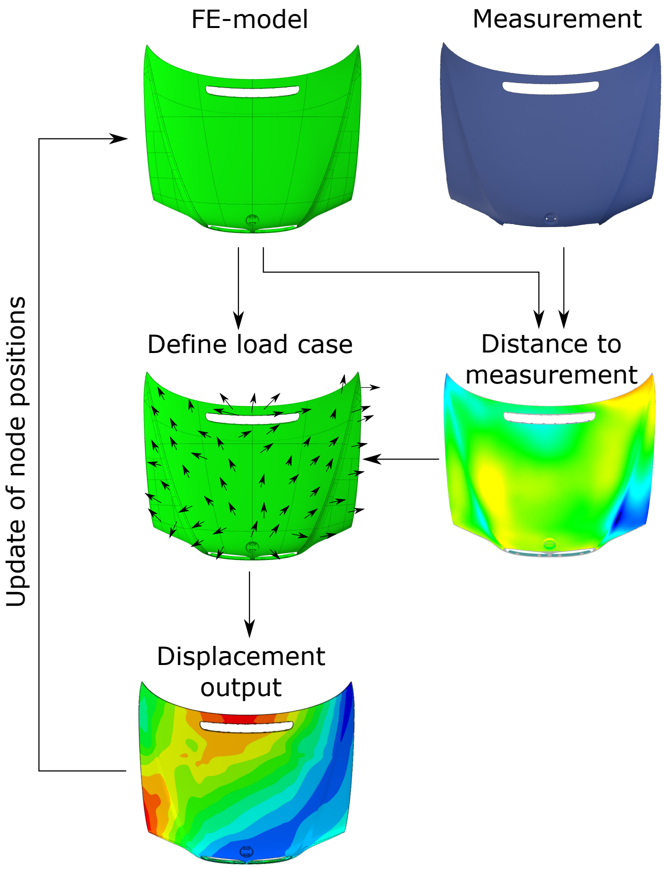


Figure 2: Computational progression of the proposed algorithm. The input of the algorithm consists of an FE-mesh based on the nominal geometry and tessellated measurement data of the actual part. A distance field is computed establishing a load case, subsequently applied to the FE model. The output updates the FE-model. The algorithm terminates when FE-model and measurement geometries match.

Based on the calculated distances, FE boundary conditions (displacements) are associated with the FE mesh. We divide the boundary conditions into two groups: (a) restrictive boundaries with prescribed movement conditions in all three dimensions, and (b) boundaries considered in node normal direction only, allowing in-plane movement. In Figure 3, restrictive boundaries (blue) are used for nodes belonging to mesh perimeters. The zone marked in grey is a row of elements next to the mesh perimeter. To prevent collision of restrictive perimeter boundaries and boundaries applied only in normal direction, this zone is introduced. For mesh nodes belonging to this zone, the placement of boundary conditions is restricted to perimeter boundaries only.

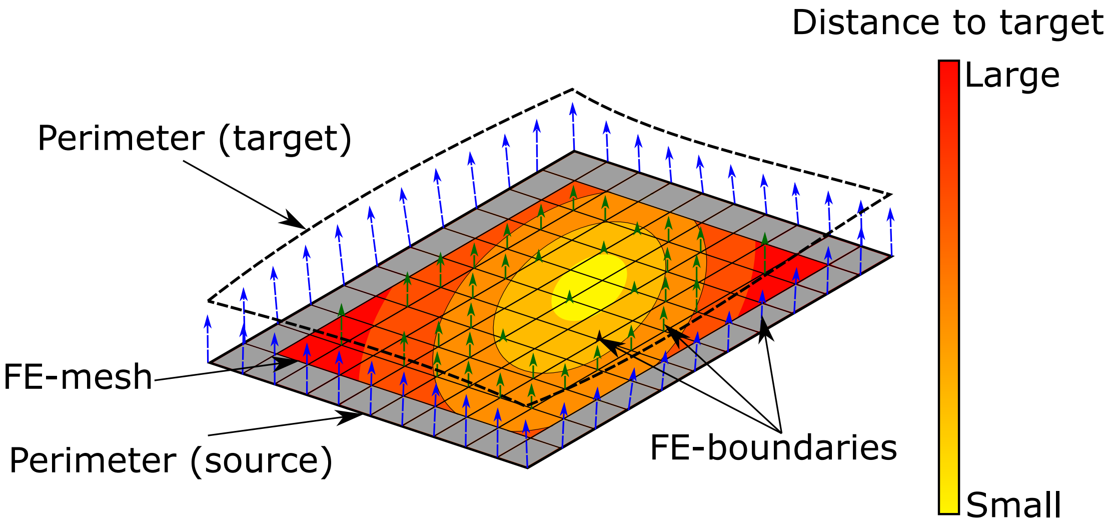


Figure 3: High-level illustration showing different types of boundaries applied to the FE-mesh which is used for morphing. Types of boundaries differentiated by (blue) perimeter vector boundaries and (green) displacement boundaries applied only in surface normal direction. The magnitude of displacements is calculated based on local distance to target geometry.

3.2 Algorithm

In the following, the algorithm implementation is described using the pseudocode Algorithm 1.

As input, two Trimesh [30] objects must be created - lines (2) and (3). While 3D scans usually are handled as tessellated data and can be directly loaded via Trimesh, the simulation mesh of the "nominal geometry," needs to be converted. Therefore, nodes and elements are parsed from the solver deck from which a Trimesh object can be created. In lines (5)-(8), the parameters that can be defined by the user are listed. These are: *(5) termination criterion (ε)* - floating-point value that is compared each iteration with the Root-Mean-Square(RMS) error value - see Equation 1, *(6) max-angle* - floating-point value that represents the maximum acceptable angle between normal of source mesh node and matched triangle of target mesh during distance computation, *(7) weight function* - a continuous function that returns a weighted value for a given displacement magnitude, *(8) coarsening parameters* - a set of parameters for coarsening a list of displacements. The first calculation steps are performed during pre-processing. In line (10), the perimeters of both input meshes are extracted and matched. A perimeter is considered a topological boundary of the geometry. Next, the extracted perimeters are parameterized and up-sampled, if necessary, and the point set registration problem is solved in line (11), resulting in a target displacement vector for each perimeter node on the source mesh. The target displacement vectors are scaled by the *weight_function* defined in line (7). Matching perimeters and calculating target displacements are optional and can only be applied if the perimeter curves are represented accurately in the target mesh.

Algorithm 1 Outline of the morphing algorithm. Input: measured point cloud and simulation mesh; Output: morphed simulation mesh.

```
 1: Data loading
 2: $X$ ← Target mesh                          ▷ Tessellated point Cloud as trimesh-obj
 3: $Y$ ← Source mesh                            ▷ Tessellated FE mesh as trimesh-obj
 4: Parameter
 5: $\epsilon$ ← Termination criterion
 6: max-angle ← Maximal normal angle difference
 7: weight_function(a) ← Magnitude weight function
 8: c_params ← Coarsening parameters
 9: Pre-processing
10: perims=extract_and_match_perimeters($X$ , $Y$)
11: disp_perims=register(perims)                   ▷ perimeter displacement vectors
12: disp=disp*weight_function(mag(disp)) for disp in disp_perims
13: Main loop
14: while $\epsilon < error$ do :
15:     magnitudes=[ ]
16:     for all $vert$ in $Y$.vertices do:
17:         $X_{tri}$, distance_vector = find closest triangle(vert , X)
18:         angle = angle(vert.normal , $X_{tri}$.normal)
19:         if angle > max-angle then skip                ▷ Filter out bad matches
20:         $b$ = ⟨ distance_vector , vert.normal ⟩
21:         magnitudes.append(weight_function($b$))
22:     end for
23:     magnitudes = coarsening(magnitudes , c_params)
24:     write_solver_deck($Y$ , magnitudes , disp_perims)
25:     run_FE_solver()
26:     update_verticies($Y$)
27:     error=RMS($X$ , $Y$)
28: end while
29: Output: write_out_morphed_model($Y$)
```

$$RMS = \sqrt{\frac{\sum_{i=1}^{n} {d_i}^2}{n}} \tag{1}$$

Equation 1: Calculating RMS error. Distance d_i is the distance error per node and n the number of nodes.

The main loop of the algorithm iterates until the RMS error is below ε - line (14) - which is calculated at the end of each iteration - line (27). In the body of the main loop the following steps are performed: For each vertex of the source mesh, the nearest triangle of the target mesh is searched - line (16). This function in native, provided by the library Trimesh, returns the closest point, distance as a scalar, and closest triangle-ID. Next, the vertex normal of the source mesh is compared to the identified triangle normal. If the angle between those normals exceeds the user-defined value *max-angle* (6) the pair will be considered a mismatch and will not be considered further in the current loop - line (18). If the difference in normal direction is within the allowable tolerance a, the displacement magnitude for FE computations will be calculated in two steps. First, the scalar product between the distance vector and normal vector of the source mesh vertex is computed in line (20). Computing the scalar product is done to obtain only the portion of displacement in normal direction. Second, the obtained scalar value b is weighted by the weight function defined in line (7). The weighted scalar value is stored in a list - line (21). Before performing the FE computation, the list of magnitudes is coarsened - line (23) - to ensure that one does not apply boundary conditions to every single vertex. The coarsening function thresholds the list and selects every n^{th} element of remaining entries. The coarsened list of displacements, perimeter displacements, and current mesh node positions are used to generate a solver deck. We note that the values held by the list *magnitudes* are scalar values while *disp_perims* are displacement vectors. Both are used for generating displacement boundary conditions. The difference is that the scalar values from *magnitudes* are used to displace the corresponding mesh node in normal direction while the in-plane degrees of freedom (DOFs) remain unrestricted so the mesh can move in-plane. In contrast, the vectors of *disp_perims* are used to move corresponding nodes along a pre-defined vector by defining all three translational DOFs. When boundaries and vertex positions are written to the new solver deck, the FE computation can be performed. The FE problem is defined as a mechanical linear elastic problem, see Equation 2, which is solved by the simulation software. The resulting displacement field is used to update the vertex positions of Y, used to evaluate an RMS error by comparing with X. At this point, one iteration has been performed. The steps line (15)-(27) are repeated until the RMS error is below ε. The final mesh is written to an output file.

$$0 = K\vec{u} + \vec{f} \tag{2}$$

Equation 2: Equilibrium formulation of linear elastic problem. The matrix K is the stiffness matrix of the mesh, $\vec{u}$ are nodal displacements, and $\vec{f}$ are nodal forces.

3.3 Mathematical Description of Calculation Steps

Important mathematical functions used in the pseudo code Algorithm 1 are explained in more depth. In particular, these are: (a) perimeter matching and point set registration - line (10)/(11); (b) distance computations and filtering - line (17)-(21); (c) user-defined weight function - line (7); (d) coarsening the calculated distances with user defined parameters - line (8) and (23); and (e) handling edge cases - not mentioned in the pseudo code.

Matching Perimeters and Point Set Registration - In analogy to mass-spring models, the morphing approach matches the mesh perimeters and applies strict displacement conditions. The nodes on perimeters can be identified easily and matched robustly. This property is used to prevent overlapping of the mesh perimeters in the final result. First, all target and source mesh element edges that belong only to one element are extracted from both meshes. Next, resulting edges are grouped by connectivity. Each group of connected edges is a mesh representation of a perimeter. To eliminate mesh influence each perimeter is re-sampled. Re-sampling is carried out by parametrizing each edge-list of perimeters by scaling accumulated edge lengths to one. Equidistant sampling is performed, generating new nodes by linear interpolation of corresponding edges. For each re-sampled perimeter, the characteristics length (l) and center of mass (c) are calculated, see Figure 4. Based on these calculated characteristics, perimeters of source and target mesh are matched. After the matching of perimeters is done, a point set registration is performed to find target displacement vectors for each vertex on the perimeters of the source mesh. The used registration method is called “coherent point drift” (CPD) [31], implemented in the library “Probreg” [32]. CPD is a state-of-the-art non-rigid point set registration method. CPD solves the point set registration problem by viewing it as a probability density estimation problem to which a Gaussian mixture model (GMM) is applied. The GMM centroids of the source point cloud are fitted to the target point cloud while the movement of GMM centroids is restricted to be coherent to maintain topological structure.

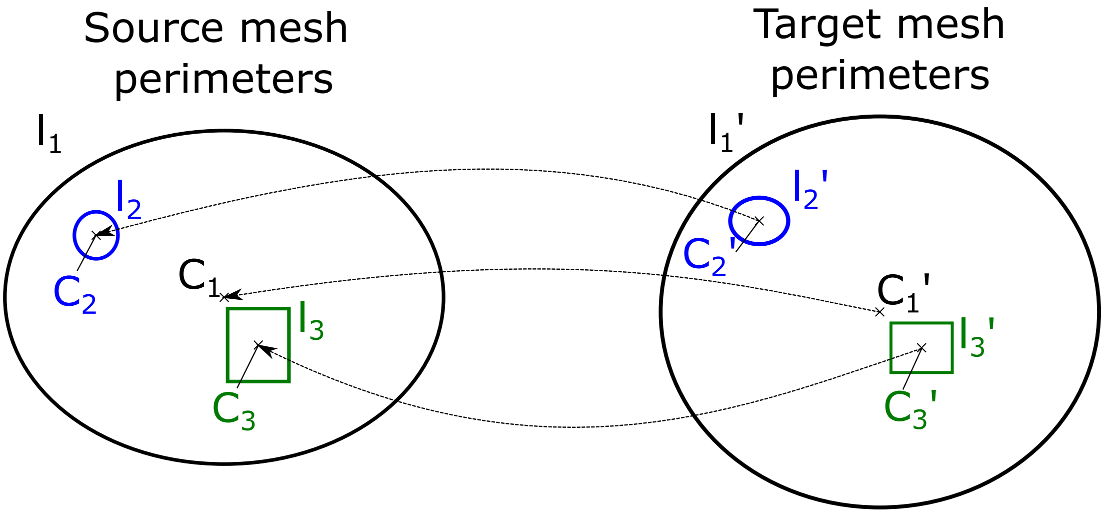


Figure 4: Matching perimeters (1-3) of source and target mesh. Perimeters with different shapes, lengths, and locations are matched using as characteristics center of mass (C) and length (l) for comparison.

The original paper provides more details. CPD can handle outliers and noisy data very well, which is important for our problem. Figure 5 shows a simple example for registering two matched perimeters that are discretized by vertices. By applying the transformation calculated with CPD to the source point cloud, the transformed source vertices are obtained. While the source point cloud preserves the original vertices from its mesh, the target point cloud is a dense equidistant re-sampling of the parametrized perimeter curve. The re-sampling of the target perimeter is done to achieve the best possible representation of the target curve, as the information of edges connecting the points or order of points is not used by the CPD algorithm. In contrast, the source point cloud is not re-sampled since we are interested in a target displacement vector for exact these vertices. If necessary, additional nodes on the perimeter edges of the source mesh could be interpolated for the registration step. However, this was not necessary for the use cases covered in this paper, and we did not implement this step. The advantage of using a point cloud representation for registering the nodes of matched perimeters is that orientation, the starting point of the curve, or distribution of vertices along the curve do not matter. The output of CPD provides the desired transformation of the source points. Based on this transformation, the desired displacement vectors are computed and are used to define nodal displacement boundary conditions on the perimeter vertices of the FE mesh (source).

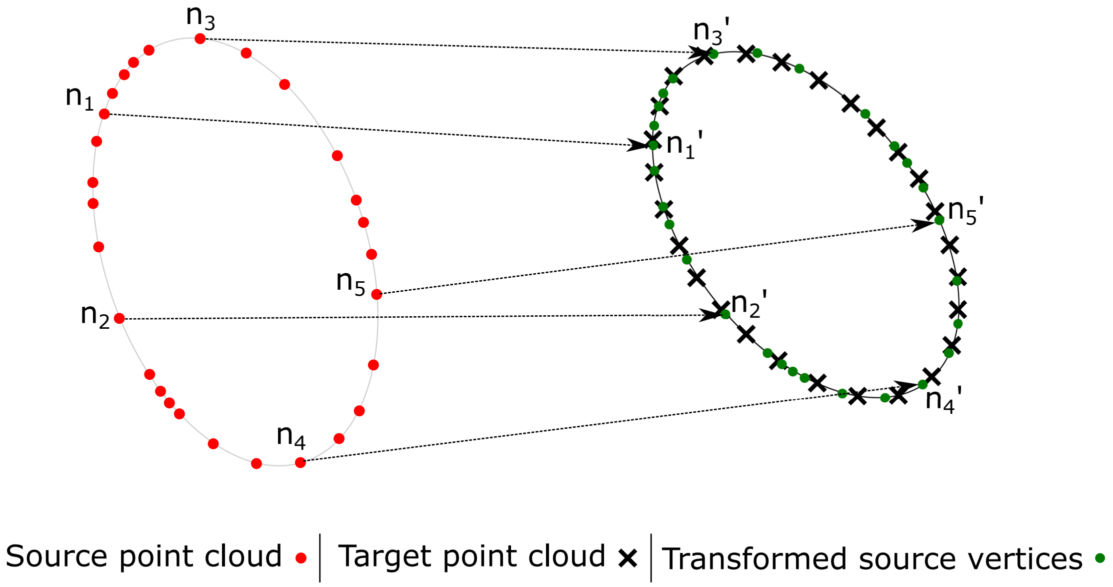


Figure 5: Point set registration using non-rigid CPD algorithm. This algorithm does not consider the order of vertices or orientation of the perimeter curve.

Distance Computation and Filtering - The distance computation for the vertices not contained in the perimeters is carried out by using a proximity function implemented in Trimesh. This implementation also does resolve ambiguous distances internally. As input, this function needs a mesh object (target) and a list of vertices (source). For each source vertex, the closest triangle of target mesh, the point on this triangle, and distance as scalar are returned. As we are interested in distances between the FE mesh vertices and the measurement, we use the tessellated measurement as target input mesh and the vertices of the FE mesh as source input points for distance calculations. On the left side in Figure 6, the components of the distance computation are visualized. Just finding the triangle closest to a vertex might result in mismatches for various cases. For example, incomplete meshes or large in-plane shifts between meshes lead to wrong matches that cause errors when deriving boundaries for the subsequent FE computations. To handle mismatches, two filters are used, see right side of Figure 6. First, the normal vector of the source vertex is compared to the normal of the matched triangle. If the angle between the two normal vectors is above the user-defined threshold

- line (6) - the vertex will not be considered for applying displacement boundary. To deal with large in-plane shifts, a second filter is applied. This second filter calculates the scalar product between the distance vector and source vertex normal vector. As every normal vector is automatically normalized by Trimesh, this scalar product can be geometrically interpreted as the component of the distance vector in vertex normal direction. The reason for only considering distances in vertex normal direction is this: The displacement boundaries for the FE problem are only applied perpendicularly to the mesh surface. All other DOFs are unrestricted. Thus, the mesh can slide during FE analysis, automatically resolving in-plane shifts.

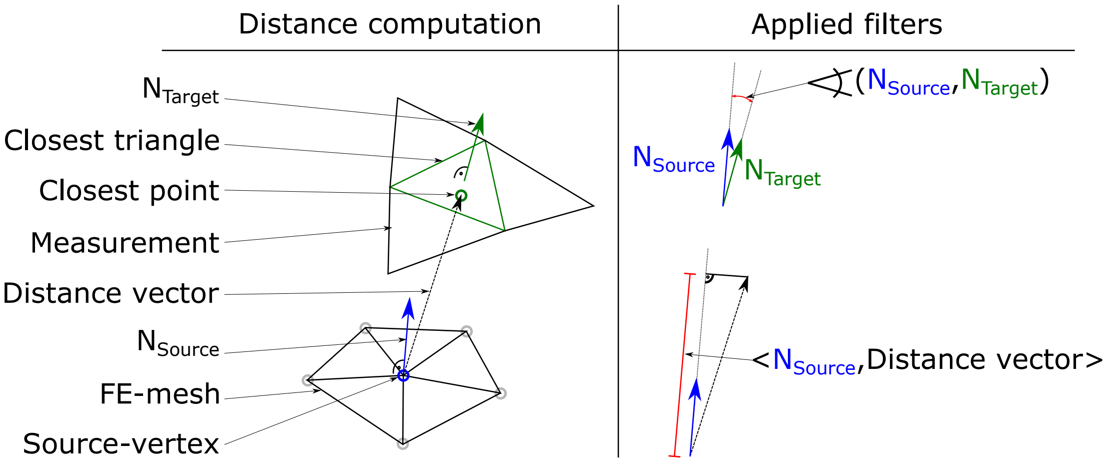


Figure 6: Left: distance computation used for finding distances for each source mesh node. Right: filters used to remove mismatches (normal - filter), and handling in-plane shifts (scalar product).

<u>**User-defined Weight Function**</u> - To prevent mesh distortions caused by largely displaced mesh nodes during the FE calculations, a weight function is applied to the magnitude of calculated distances. This approach generates a smooth displacement field with smaller displacement magnitudes. Between iterations, the distance to target geometry is re-evaluated and updated. The function used in our method is plotted in Figure 7. For small distance magnitudes, the function weights magnitude of the corresponding displacement boundary with a factor near 1. For large distances, the magnitude gets weighted down, but never towards zero as this would pin down single mesh nodes resulting in slow convergence of shape optimization problems. For different applications, the weight function needs to be scaled according to element size. The weight function could also be used to accelerate convergence as this function has a direct

impact on the choice of either more or less aggressive displacement magnitudes. The effect of applying such a weight function to the calculated displacement magnitudes is depicted in Figure 8. In the figure, the source mesh is considered as FE simulation mesh, to which boundary conditions are applied, while the target mesh corresponds to a tesselated representation of the actual measured geometry. The length of each displacement boundary is scaled by the weighting function. The boundaries placed on the perimeters are shown in blue. These boundaries have a pre-described vector calculated by the point set registration. The red boundaries are displacements for the source mesh nodes that do not belong to perimeters. These boundaries only displace the corresponding mesh node in normal direction during FE calculations. This is achieved via the second applied filter, see Figure 6, and is done so the node can slide perpendicular to the surface normal. This sliding has the effect, that the mesh can relax to avoid distortions. After each iteration, the mesh node positions are updated, shown via reduced opacity. The new shape is closer to the target mesh. Due to the mesh update, vertex normal vectors change as well. This leads to fewer mismatches when filtering distances in the next iteration. As matches and normal directions do change between iterations, the weighting function does define how aggressively the FE computation changes the mesh.

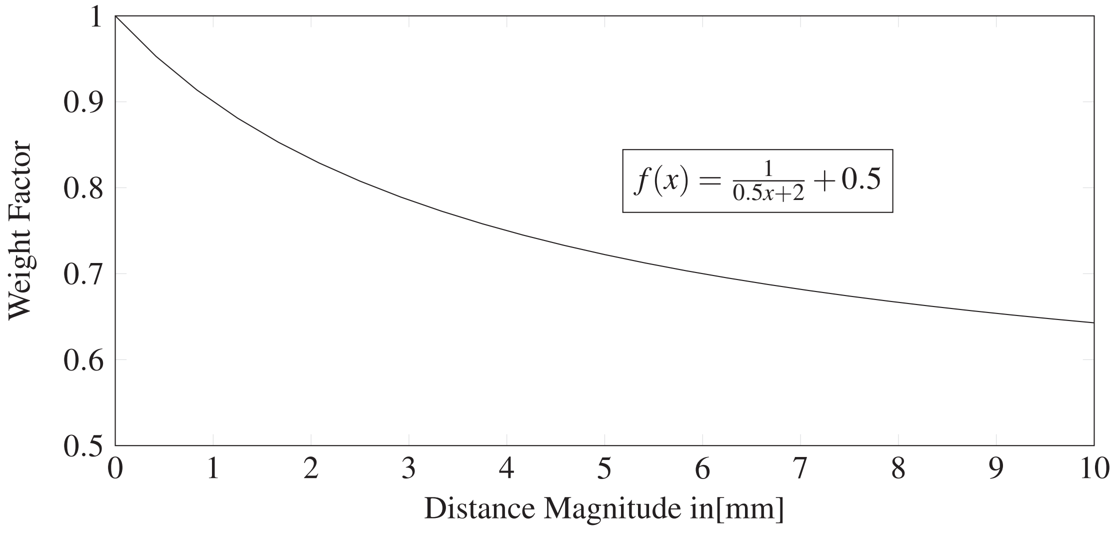


Figure 7: Weight function $f(x)$ for scaling matched distance magnitudes. Small distances are weighted near 1, while large distances are weighted by approximately 0.5.

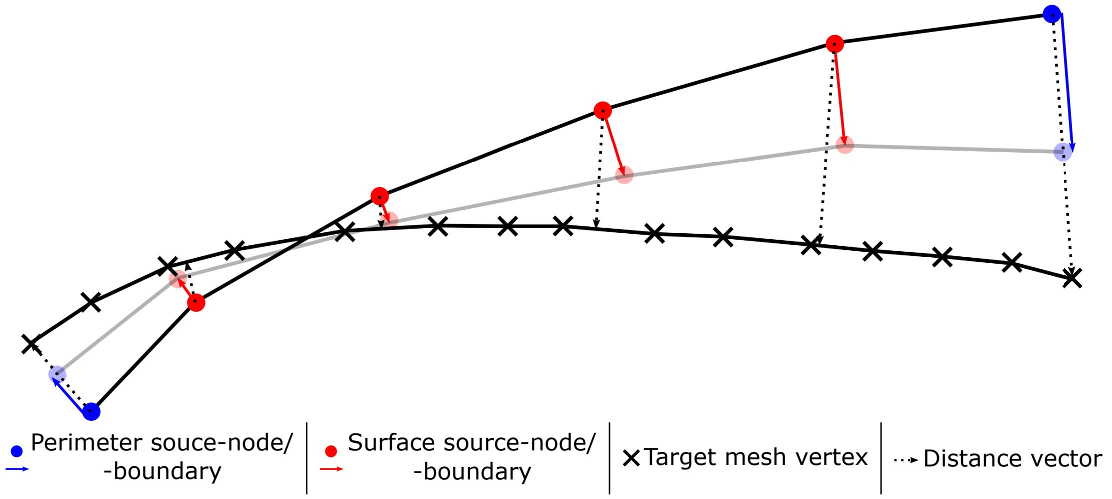


Figure 8: One iteration step with weighted boundaries. Blue: displacement vectors of perimeter mesh vertices. Red: displacement applied to surface mesh nodes in normal direction. Transparent: updated mesh node positions after FE simulation.

Coarsening Matched Distances - Before defining boundary conditions from the calculated and filtered distances, the identified matches are thresholded and coarsened. For thresholding and coarsening, different user-defined values can be set, considered as a set of coarsening parameters in line (8) of Algorithm 1. First, all distance values are thresholded by a certain value. This is done to only consider mesh nodes that are in a defined range of the target surface. Thus, mesh nodes far away from the target mesh are not considered until they move close enough due to previous iterations. Next, a margin can be defined by a number of rows. This number defines the number of element rows next to perimeters that must not have any boundary conditions, compare Figure 3. As the perimeters have restrictive displacement conditions, additional boundaries near perimeters can cause mesh distortions, especially during the first iterations. This value can be set to 0 towards the last iterations. All remaining values that are not classified as a mismatch by the distance computations or removed due to proximity to perimeter mesh nodes are clustered using three classes (a,b,c). For each class, different density parameters for boundary placements are defined. The purpose of this classification and varying densities for boundary placement is illustrated in Figure 9: Class (a), nodes that are already inside the target range. These nodes are kept in position (only normal direction restricted) - displacement value is set to zero; Class (b), nodes that are used mainly to pull the source mesh towards the measurement during the current iteration; Class (c) remaining nodes below the threshold, but outside class (a) and (b). For each class, distance value ranges and a coarsening number n must be defined by the user. Coarsening for each class is performed by selecting every nth node for placing a boundary condition. For class (a) and (c), a higher value, e.g. n=5, can be selected. The rationale is that in class (a) almost no deformations occur. Thus, only a few boundary conditions must hold for the source nodes near the target surface. In class (b), most of the deformations take place. Therefore, dense sampling of boundaries is chosen, i.e., n=2. Class (c) is used to guide the area adjacent to the deformation zone. As distance computations might be less robust and discontinuous in this class, only a few boundaries are placed. The reason for selecting only every n-th node for placing a boundary is this: The simulation result becomes “smoothed” as the part’s stiffness acts similarly to a cubic spline between restricted nodes. With each iteration, the classes shift. Class (a) permanently grows until the whole part is covered.

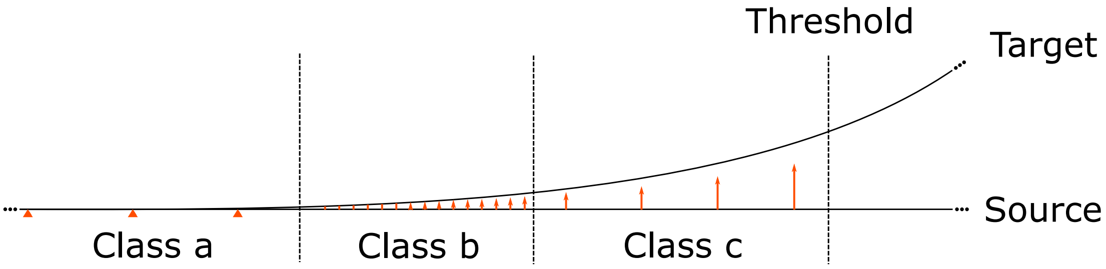


Figure 9: Placement of weighted displacement boundaries, clustered by distances into three classes: (a) close to target and held in position; (b) deformation zone and dense placement of boundaries; (c) coarse placement to guide source mesh towards target.

Edge Cases - Although the proposed algorithm is reliable and robust, some edge cases should be considered to improve results. The phrase “edge case” is meant in the context of computer programming. One edge case occurs in areas with high surface curvature. In these areas, the comparison between node normal and mesh element normal leads to errors. These errors are caused by the node normal not matching an adjacent element normal. This leads to a “mismatch” classification when the difference in normal angle exceeds the specified threshold for filtering. Specifically, when calculating the scalar product, see Figure 6, the resulting displacement vector is generally quite small. To compensate for such errors the calculation of the scalar product is skipped when the computed distance value is below local mesh resolution (maximum edge length of surrounding elements). The second edge case occurs when the target mesh does not cover the whole surface of the source mesh (incomplete measurements). In this case, perimeters might not be captured. Further, measurement values are often noisier near perimeters. To handle such problems, we remove rows near perimeters from the candidates for boundary placement. This applies to elements that are between class (c) and thresholded nodes, see Figure 9. As all not-measured areas have large distances to the next triangle on the measurement, there exists a high degree of change of distance near the outline of the measurement. This property can be used to shrink the nodes where boundaries can be placed. The user can define how many rows of elements are discarded to adjust the algorithm to measurement quality.

3.4 Parameters used to Adjust Algorithm

The proposed algorithm is based on various parameters that can be used to improve morphing quality for individual use cases. This section introduces parameters that need to be defined by the user. Their impact on the morphing algorithm is discussed as well.

Search Distance - This parameter defines the maximally allowed distance magnitude, calculated by trimesh, see, “Distance Computation and Filtering.” All distance values above the search distance are not considered for placing boundary conditions. This value only thresholds distances for nodes that do not belong to a perimeter of the geometry. Varying this value has a significant impact on the number of boundary conditions being placed.

Normal Angle - This parameter is used for the comparison of node normal (source) and triangle normal (target), determined with the distance computation. If the angle between node normal and triangle normal exceeds the parameter value, the match is classified as a mismatch and discarded, see Figure 6. This parameter impacts the number of node candidates that are kept for boundary placement greatly.

Class (a-c) - This set of parameters controls the density of boundary placement for each class introduced in “Edge Cases.” For each class, a range of distances as well as a number n must be defined. The distance range is used to generate a list that is a subset of all mesh nodes, where the calculated distance values are within the defined range. The number n is used to coarsen the list by selecting every n^{th} element. By selecting different ranges and sampling densities, the main deformation zone can be controlled, compare Figure 9.

Remove Rows - This parameter is an integer that defines how many rows of elements next to perimeters must not contain boundary conditions, except perimeter boundaries. In Figure 3, elements colored in grey show where this parameter is set to one. By increasing the number of element rows being excluded from boundary placement, the distance between the different kinds of boundary conditions is increased. In this zone (compare Figure 3) no boundary conditions except perimeter boundaries are placed. Subsequently, nodal displacements only result from deformation calculated by the FE simulation.

Relaxing Parameters - This condition determines when an intermediate morphing step is close enough to the target geometry in order to relax parameters, which were listed before. For example, this condition could be an RMS error value, maximum error value, or fixed iteration steps. The idea is to use moderate parameter sets in the beginning and switch to more aggressive parameters when getting close to the final step. For instance, by increasing the "normal angle" criterion, lowering "remove rows" and increasing sampling density *n* for all classes, faster convergence, and higher geometric precision result. When applying such conditions too early, the FE mesh can become distorted, resulting in low-quality results. The choice of this condition is problem-dependent. However, to achieve the desired behavior we apply them as early as possible, i.e., when perimeter boundaries have reached their target position. In general, it would be also doable to introduce multiple relaxing criteria that allow a successive change from moderate parameter sets towards restrictive ones.

⋮

References

[30] Dawson-Haggerty et al., trimesh.
URL https://trimsh.org/

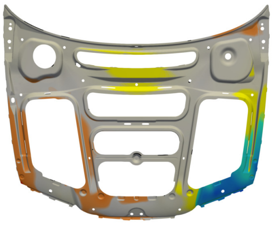
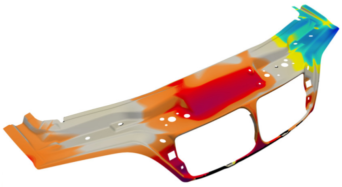
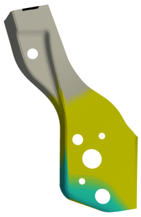
[31] A. Myronenko, X. Song, Point set registration: Coherent point drift, IEEE transactions on pattern analysis and machine intelligence 32 (12) (2010) 2262-2275.

[32] Kenta-Tanaka et al., probreg.
URL https://probreg.readthedocs.io/en/latest/

4.4 Verification

To verify the performance and usability of the presented method, verifications with different geometries are performed. The verifications are using artificially created pseudo measurements as ground truth. In the following, real-world parts with different sizes, complexities, and numbers of components are simulated. Also, special test geometries are created to investigate how the algorithm handles dents and buckles of different sizes and large lateral shifts.

4.4.1 Verifications with Single Parts

Part Feature	Part 1	Part 2	Part 3
# Mesh Elements	120381	40851	4039
Material	Steel	Steel	Steel
Thickness [mm]	0.7	0.7	2
# of Perimeters	51	27	7
Geometry & distance to measurement			

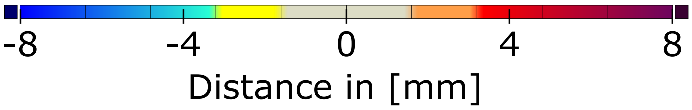


Fig. 4.2 Three different parts for verification with different characteristics. Colored pictures show the difference in geometry between nominal and artificially created pseudo measurement. (This figure has been published in [6], changes: color map)

In Figure 4.2 three different, single sheet metal parts are shown. All parts are from the substructure of the engine hood that was introduced in Section 3.5.1. In the figure characteristics of each geometry are listed, also the distance between nominal- and actual- geometry is shown. The actual geometry was created by deforming the nominal geometry via FE-simulation. The maximal difference between nominal and actual geometry gets as large as ± 8 mm which is significant for sheet metal assemblies in automotive applications. By generating the actual geometry via FE-simulation, the "actual" part is represented by the same smooth mesh structure like the FE-mesh of the nominal geometry. In contrast to actual real-world measurement data that is subjected to noise and measurement errors, these conditions are considered ideal. However, this verification under ideal conditions must succeed to ensure a correct behavior of the method.

The mesh morphing algorithm is applied to the geometries shown in Figure 4.2 with the user-defined parameters from Table 4.1. A tessellated point cloud was exported from a preceded FE-simulation and serves as pseudo measurement input for the morphing algorithm. The nominal geometry is given as FE-model. As the perimeters of the geometries are represented continuously in both meshes, perimeters are used for morphing.

Table 4.1 Parameter sets for verification with single parts.

Parameter	Part 1	Part 2	Part 3
n Class a [0mm,0.05mm]	3	3	3
n Class b]0.05mm,1mm]	2	2	2
n Class c 1mm<	2	2	2
Search Distance	10 mm	10 mm	10 mm
Remove Rows	2	2	1
Normal Angle	10°	30°	30°
		Relaxing Parameter	
Relax if *max dist* <	0.5 mm	2 mm	2 mm
Remove Rows	0	0	0
Normal Angle	30°	40°	40°

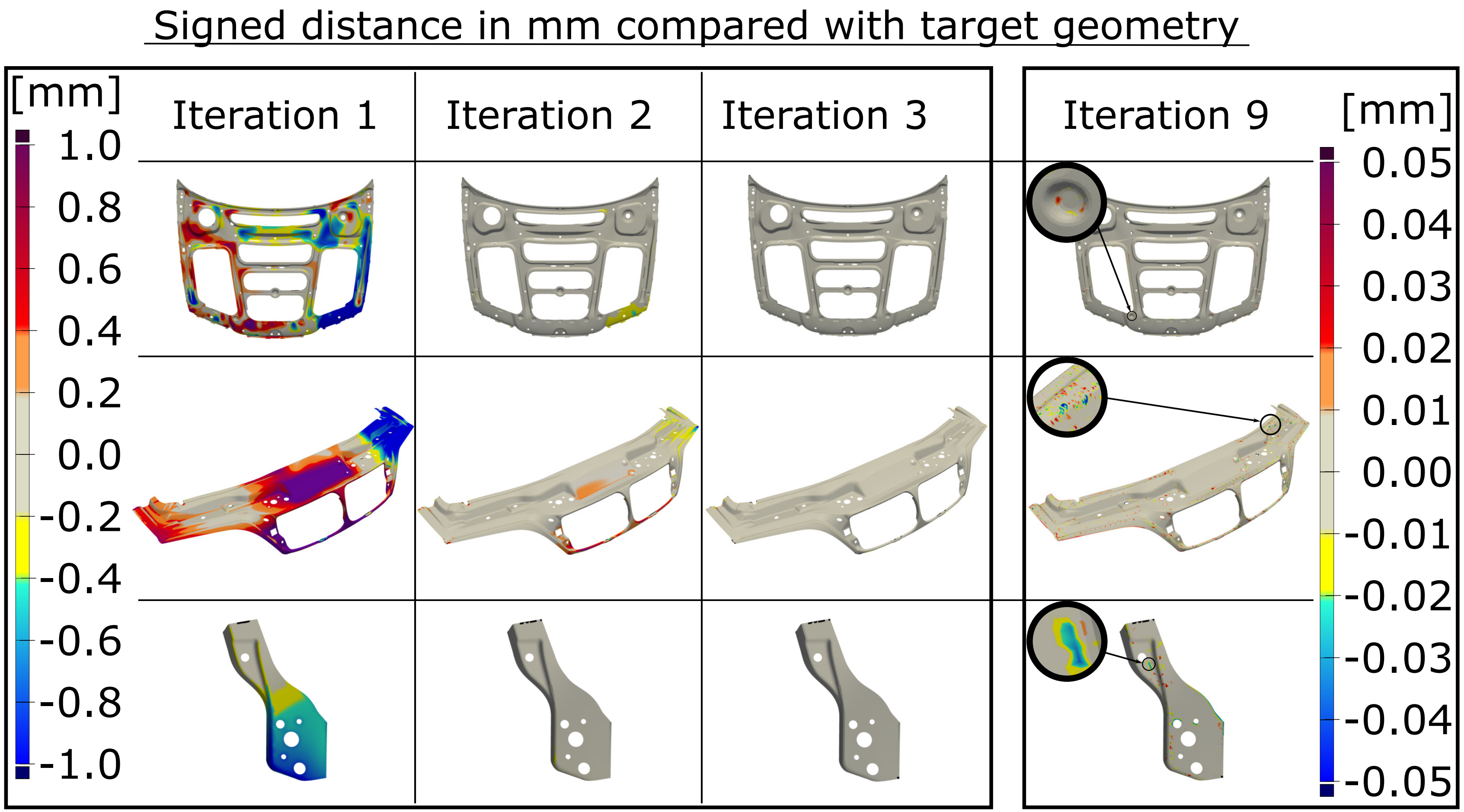


Fig. 4.3 Verification results for three different parts. Interim iteration steps are compared with target geometry on a scale of ± 1 mm. Final results are achieved after 9 iterations and are compared on a scale of ± 0.05 mm. (This figure has been published in [6], changes: color map, minor adjustments.)

Figure 4.3 shows the output of the mesh morphing algorithm. Besides the final results which are achieved after 9 iterations, the interim results of the first three iterations are shown on a scale of ± 1 mm. During these three iterations, most of the approximation takes place. After iteration two, the differences between morphed- and target geometry is already below ± 1 mm, after iteration three even below ± 0.2 mm. The final iteration shows no differences above ± 0.05 mm which is on the same scale as measurement uncertainties of industrial-grade 3D-scanners. Smaller differences are magnified in the figure. However, differences on the small scale are not relevant for the present application.

4.4.2 Verification with Assemblies

To investigate the behavior of the morphing algorithm when applied to assemblies, an additional verification is performed. In this verification, the whole assembly of the engine hood is morphed by only using geometrical information of the outer skin. This setup is intended to replicate real use cases where areas of the geometry and inner structures are not accessible for scanning. The assumption is made that the FE-simulation does also deform the underlying structure in such a way that the final geometry of the substructure results in a plausible shape. As the ground truth of this experiment is artificially created, the morphing quality of not scanned areas can be assessed. The initial shape and final morphed shape for all components of the engine hood are shown as an exploded view in Figure 4.4. On the left side of the figure, the initial shape deviation is shown which is obtained by comparing nominal geometry with the artificially created “actual” geometry. The right side of the figure shows the morphing result compared to “actual” geometry on the scale of ± 0.5 mm. As the outer skin of the hood is used for morphing, the approximation of this component does show no differences on this scale. In contrast, the underlying structure shows recognizable differences. However, the resulting geometry is significantly closer to the target than before morphing.

4.4.3 Verification with Dented Test Geometry

Common differences between nominal- and actual geometries are surface profile defects. These defects could for instance be dents, buckles, scratches, etc. Especially, dents and buckles can cause a local change in geometrical stiffness compared to an ideally smooth geometry. So, the profile defects must be considered for the generation of DTs to obtain a correct mechanical behavior. To investigate how the presented approach does handle dents and buckles of different sizes and magnitudes, a test geometry is created which is shown in Figure 4.5. The nominal geometry is a flat rectangular piece of sheet metal with the

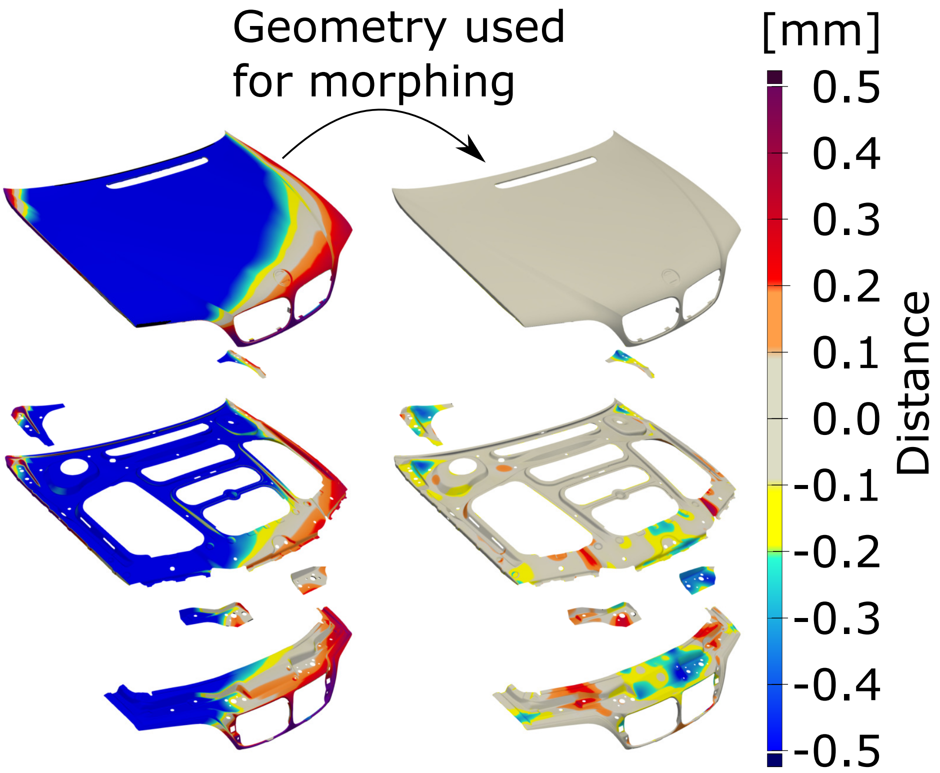


Fig. 4.4 Explosion view of the engine hood. Only the outer skin of the hood is considered to be accessible for measure. Left: comparison of nominal geometry with target geometry per component. Right: comparison of morphing results with target geometry per component. (This figure has been published in [6], changes: color map)

dimensions of 25 mm x 122 mm. The nominal geometry has meshed with 3025 first-order shell elements. At the top of Figure 4.5, the differences to the target geometry are shown by evaluating the profile deviation at the cross-section shown in the lower picture. The target geometry was created by using the mesh editing software “Meshmixer” from Autodesk. Five dents with different sizes and magnitudes were created by deforming the nominal geometry. The result was exported as tessellated mesh and serves as input for this verification.

The result of the morphing compared to target geometry is also shown in Figure 4.5. A Node-to-surface comparison is shown in the lower picture. This comparison does show no differences above ± 0.05 mm. The lower graph shows an evaluation of the same middle cross-section but this time comparing the morphed geometry to target. It is striking, that differences above ± 0.05 mm only occur at the position of the smallest dent. A closeup of this position is given by the view A-A.

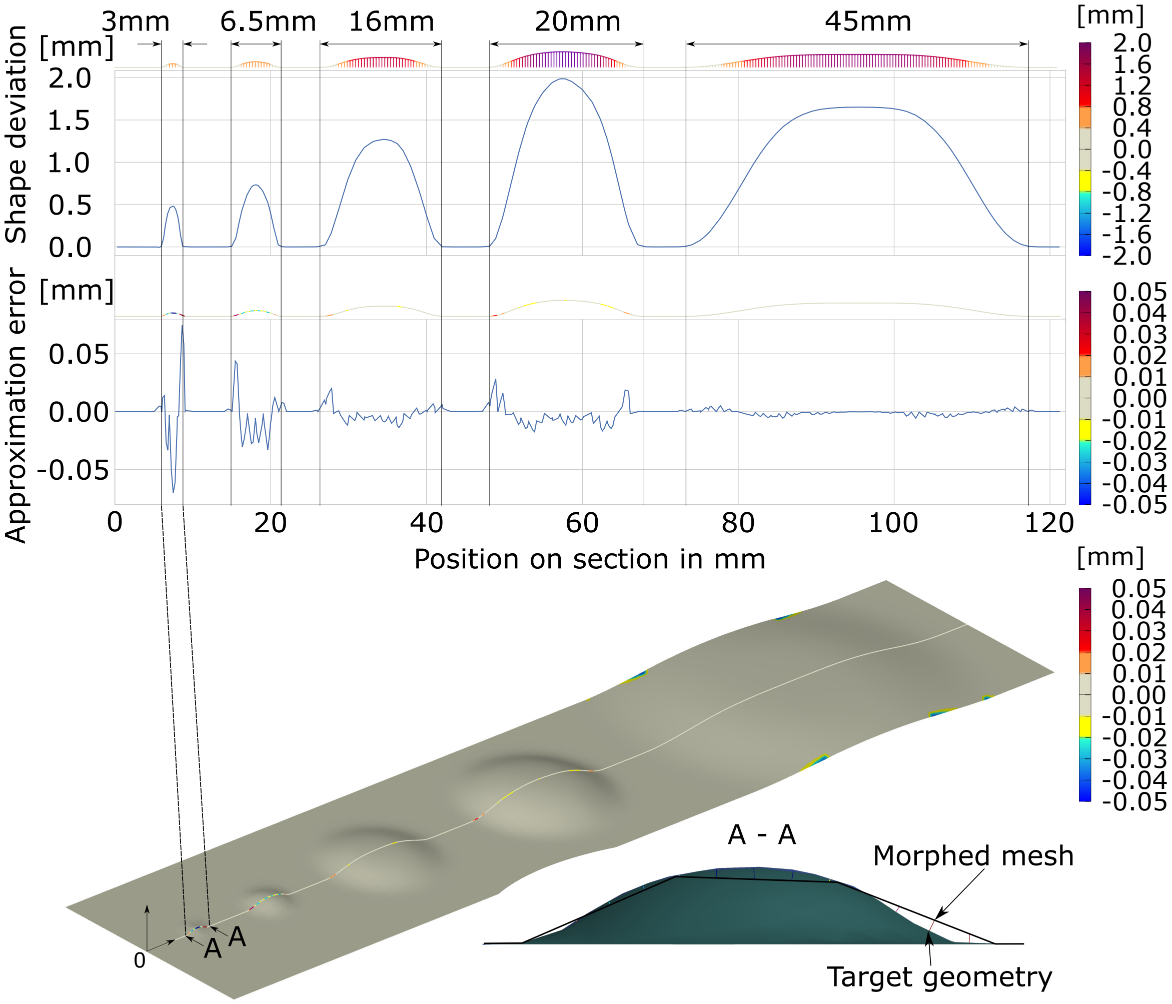


Fig. 4.5 Morphing of a test geometry with surface profile defects of different diameters and magnitudes. Top graph: profile comparison between nominal and dented geometry. Middle graph: profile comparison between morphed and dented geometry. Bottom: 3D-view of a node to surface comparison between morphed and target geometry. A-A: closeup of measured differences at the marked position.

4.4.4 Verification with Large Lateral Displacements

The third verification addresses cases where large lateral displacements occur. Therefore, an unfortunate scenario is created where an angled sheet metal is morphed towards a target geometry that has significantly different measures. To morph the initial FE-mesh to match with target geometry, nodes must be shifted up to 10 mm in-plane. Nominal and target geometries are shown in Figure 4.6 (Iteration 0). Also, interim results of the morphing process are showing how the utilized FE-simulation iteratively approximates the target geometry. The final result (Iteration 9) does not show any differences above ± 0.05 mm in the node-to-surface comparison. As the number of finite elements remains the same, the elements are stretched during the morphing process to approximate the larger surface of the target geometry. However, the quality of the element's shape remains good during morphing. Also, the solution converges towards a smooth mesh while interim results appear to be wrinkled. For this verification, also perimeters of source and target mesh are matched and used for morphing.

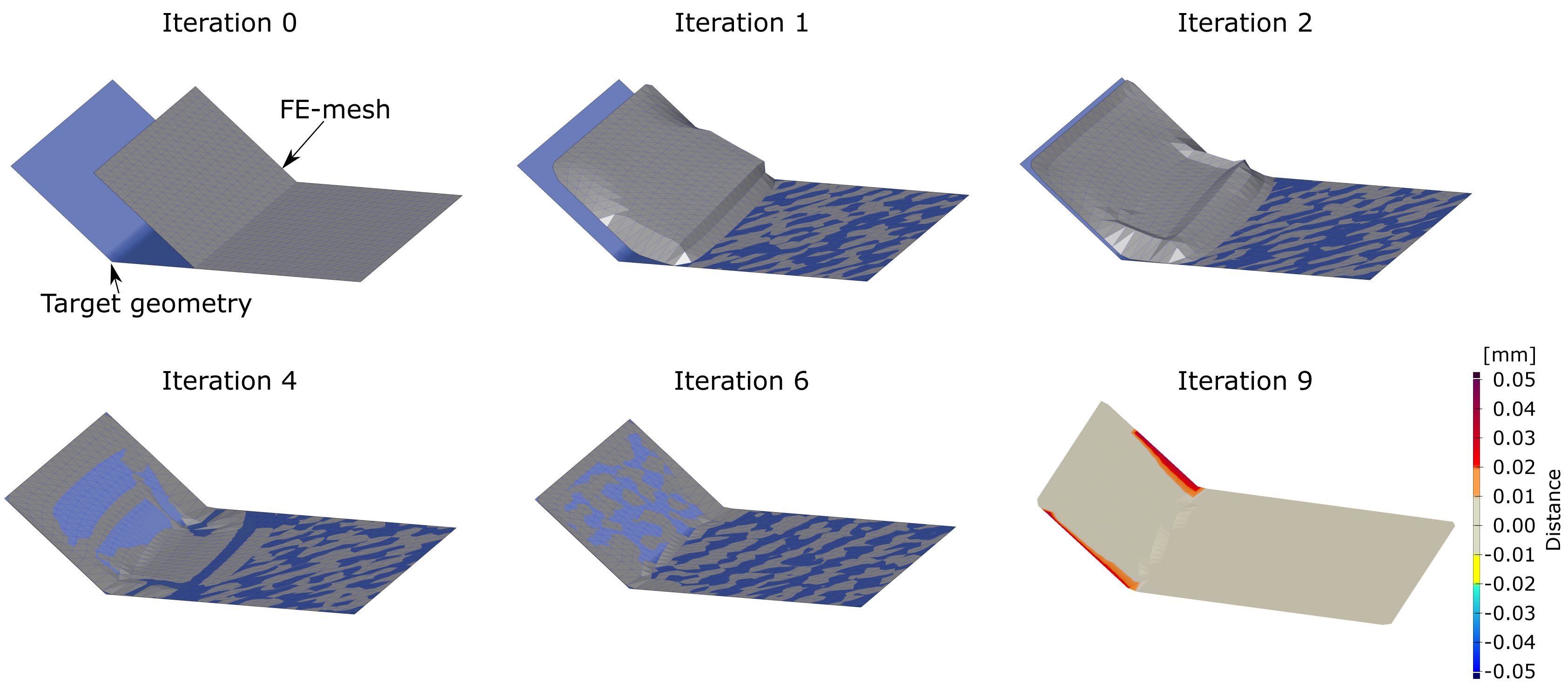


Fig. 4.6 Morphing of a test geometry with large lateral displacements. Initial situation (Iteration 0), interim results (Iteration 1-6) and final morphing result (Iteration 9) are shown. Final result is compared with target geometry on a scale of ± 0.05 mm. (This figure has been published in [6], changes: color map)

4.5 Validation

As all verifications with artificial test data were successful, validations with real measurement data can be carried out. This step is crucial to ensure the mesh morphing algorithm is applicable to real-world use cases. Often, methods in the literature are only verified with artificial data but the challenges that might occur when switching to real-world measurements, are not addressed. To address these kinds of challenges, this section presents a validation of the morphing approach involving real 3D-scan hardware and actual physical car components.

4.5.1 Experiment Setup

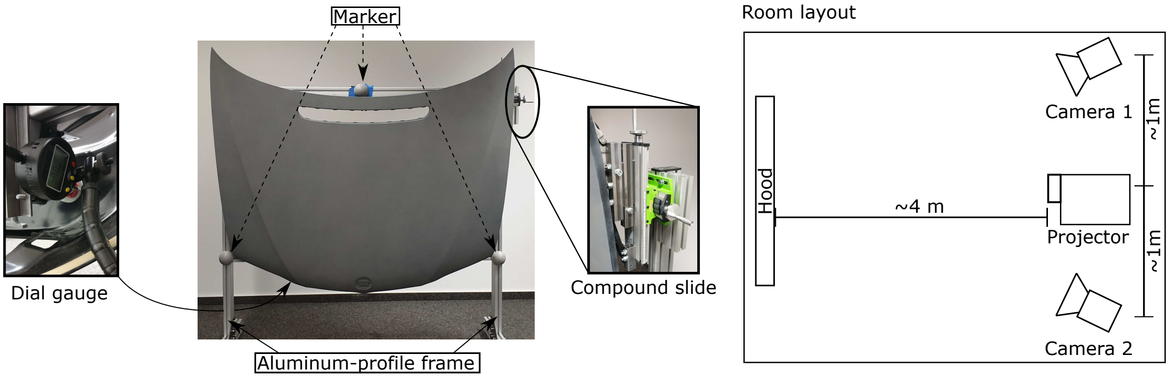


Fig. 4.7 Experimental setup. Left: engine hood mounted on an aluminum frame with adjustable mechanical boundaries. Right: room layout for setting up 3D-scanner for the large scan window. (This figure has been published in [4], licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/; changes: none)

The used hardware is introduced in Figure 4.7. The part being scanned is an actual physical counterpart of the simulation model that was used before for verifications. To prevent measurement errors caused by reflections, a light grey spray paint with a matte finish is applied to the outer skin of the part. The engine hood is mounted to a frame that is made of aluminum profiles. For connecting the engine hood with the frame, the original mechanical interfaces (hinges, locks, buffers, and gas springs) were used. The locations of mechanical interfaces are shown in Figure 4.8. While the hinges and locks hold the part in place, the gas springs push with about 580N each against the hinges to support lifting when the hood is being opened. Buffers can only push against the hood and are adjusted in height to tune in gap and flushness. The second purpose of buffers is to tension the hood in closed state to prevent vibration during high speeds. Figure 4.8 also introduces the orientation (left and right) which is defined relative to the direction of driving. To make the interfaces adjustable one hinge is mounted on a compound slide while the other hinge is fixed. The locks and buffers have proprietary adjustment possibilities. To make adjustments in a controlled manner, a dial gauge is used to observe changes. The aluminum frame is equipped with three 3D-printed markers, which are spherical shaped. The position of these markers can be precisely reconstructed from an acquired point cloud and so being used for aligning multiple scans. The setup of the 3D-scanner is shown on the right side of Figure 4.7. To avoid registration errors, the scan window of the measurement device is enlarged to capture the whole geometry in one picture. Therefore, the distance between projector and scan object, as well as distances between projector and cameras are greatly enlarged. To calibrate the scaled scan setup, also larger calibration panels are needed. In Figure 4.9 a size comparison of custom-made calibration panels and the largest calibration pattern of the glass panels that come with the scanner is shown. These custom panels are made from two mirrors to which an up-scaled print of the calibration pattern is applied. This enlarged setup is not expected to provide the same precision or level of detail as industrial-grade equipment does. However, a huge advantage of this setup is this: no registration between scans is needed which eliminates a critical error source.

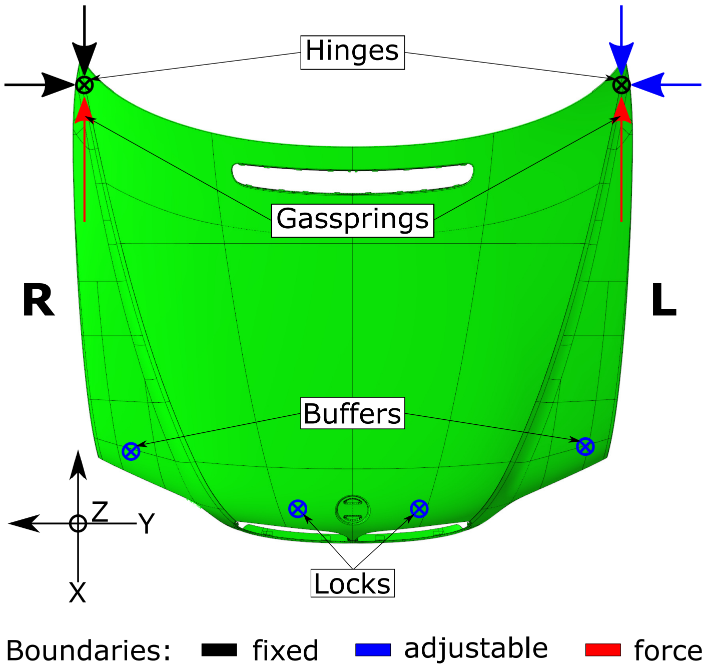


Fig. 4.8 Positions of mechanical interfaces. The hood is connected by hinges, buffers, locks with the chassis. Two additional gas springs are pressing with 580N each against the hinges in closed state. Also, orientation left (L) and right (R) are marked. (This figure has been published in [4], licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/; changes: only one subfigure is depicted.)

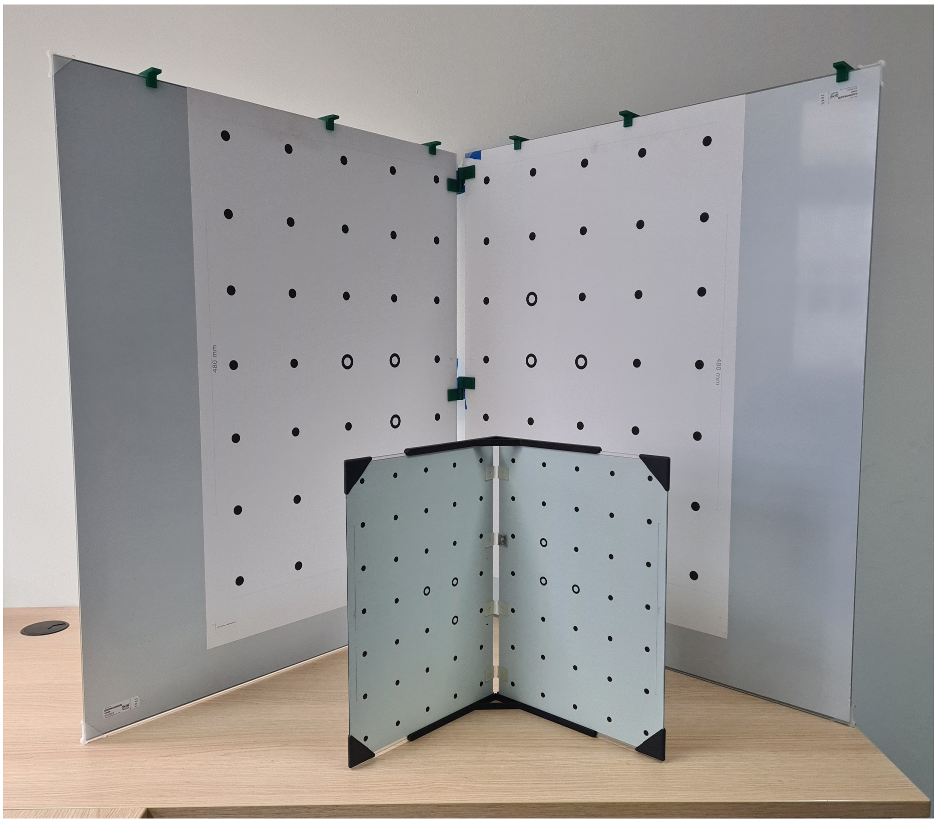
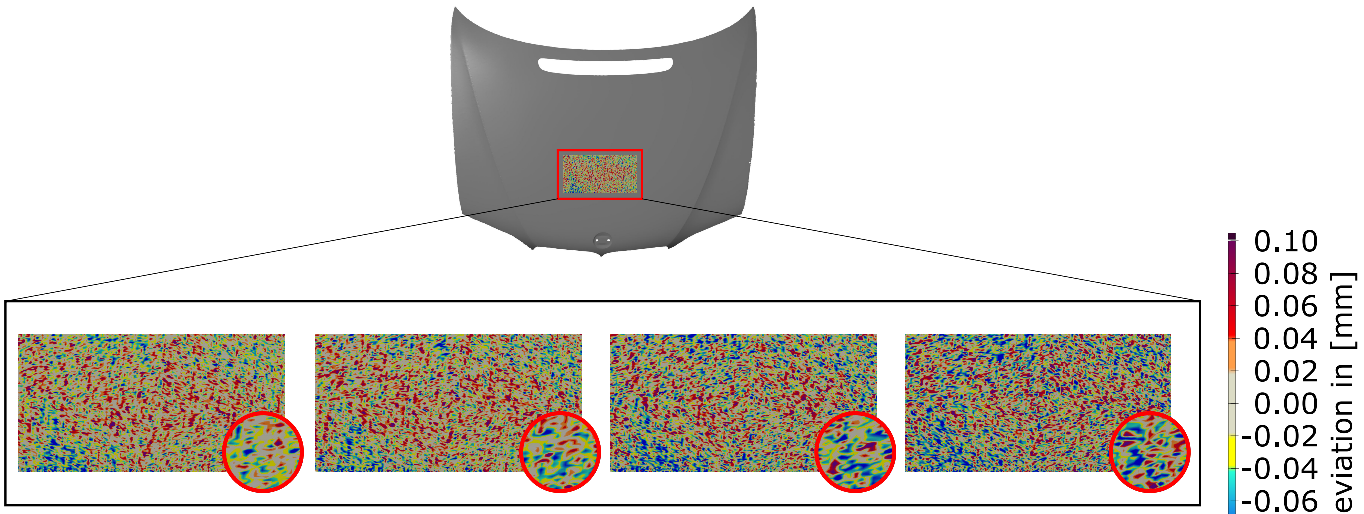


Fig. 4.9 Comparison of calibration panels. Front: small calibration panel that is delivered with the 3D-scanner. Back: custom calibration panel made from mirrors with printed calibration pattern.

4.5.2 Measurement uncertainty of the Scan System

To ensure the scan system does fulfill the requirements for the validations, preliminary tests are performed to investigate measurement noise and reproducibility. Also, the impact of different scan parameters on the output quality is measured to find the best suitable set of parameters. The phrase “scan system” does not only include the 3D-scanner but also measurement fixture, specimen, and scan environment. All factors that can impact the scan results must be considered and assessed to make a full comprehensive judgment. To achieve the best possible results, different values for adjustable parameters are tested. Also, the whole scan setup is validated by performing reproducibility tests.

One adjustable parameter that has a high impact on the measurement results is the exposure time of the cameras. To find the best matching value, a series of scans are performed. In between scans, nothing was changed, except for the exposure time in the scanning software. The results are shown in Figure 4.10. The figure shows snippets of the scans that are compared to a smooth surface that was fitted to the average point cloud. The distance to this smooth surface representation does reflect the noise of each scan. It can be observed that although measurement noise is very different regarding maximum and minimum deviation, the average deviation is almost the same for all exposure times. This is an important finding because the real surface of the part is expected to be smooth. Differences caused by measurement noise can be smoothed without losing important geometrical information. Based on this finding, the exposure time is set to $\frac{1}{10}s$. The reason for not choosing the setting with the lowest standard deviation is this: the scans performed with an exposure time of $\frac{1}{10}s$ provide the best compromise of good scan coverage while maintaining low measurement noise. Smoothing can be performed to remove occurring noise peaks.



Exposure time	$\frac{1}{7.5}$	$\frac{1}{10}$	$\frac{1}{15}$	$\frac{1}{20}$
Maximum positive deviation	+0.66 mm	+0.60 mm	+0.61 mm	+1.10 mm
Maximum negative deviation	-0.54 mm	-0.66 mm	-0.81 mm	-0.66 mm
Standart deviation	+0.07 mm	+0.08 mm	+0.09 mm	+0.10 mm
Average deviation	+0.01 mm	+0.01 mm	+0.01 mm	+0.02 mm

Fig. 4.10 Investigation of measurement noise using different exposure times of the scanner's cameras. A scan snippet is compared to a smooth surface fitted to a high-density point cloud that was created by combining all scans. (This figure has been published in [5], licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/; changes: color map)

To ensure the designed fixture can be used for performing reproducible experiments, additional testing is performed. For this experiment, the engine hood is scanned arbitrarily mounted to the frame. This means positions of the mechanical boundaries are not tuned in specifically. This scan is used as a baseline measurement. Next, locks and buffers are changed in a controlled way. Each offset has a magnitude of about 1mm. The change in position is captured with the dial gauge. After changing buffers and locks, their initial position is dialed in again. A second scan is performed which can be compared to the baseline measurement. This comparison is shown in Figure 4.11. Besides measurement noise, no significant deviations can be observed.

In conclusion, the used scan system which composes of a customized structured light scanner and a simple measurement fixture does not provide industrial-grade scan quality. However, having imperfect scan results for validation purposes is even more challenging for the method that is being tested. As the scans are reproducible, the whole system serves its purpose.

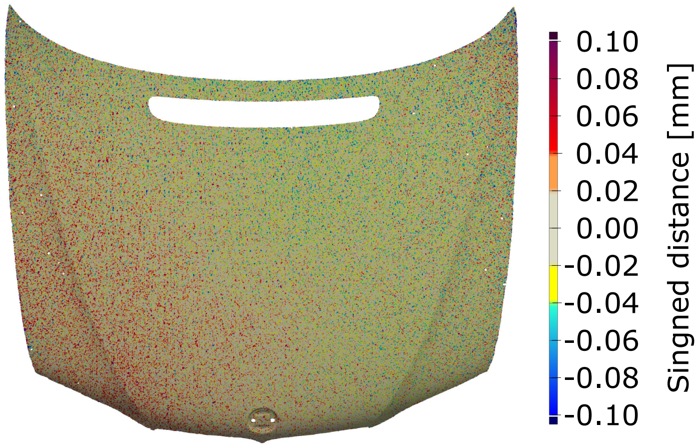


Fig. 4.11 Results of reproducibility test of the used measurement system. The comparison shows the difference between the baseline measurement and a second measurement that was acquired after changing positions of mechanical boundaries and resetting back to the initial state. (This figure has been published in [5], licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/; changes: color map)

4.5.3 Morphing results

The actual validation was carried out by scanning the engine hood which is mounted to the aluminum frame. The acquired point cloud covers only part of the outer visible skin. As for the tests from Figure 4.10 the same scan window was used, the coverage of the measurement

Table 4.2 Parameter set used for morphing scan data.

Parameter	Iteration 1-3	Iteration 4-6	Iteration 7	Iteration 8-9
n Class a [0mm,0.05mm]	5	3	3	1
n Class b]0.05mm,1mm]	3	3	3	3
n Class c 1mm<	3	3	3	3
Search Distance	5 mm	5 mm	5 mm	5 mm
Remove Rows	3	1	1	1
Normal Angle	15°	30°	50°	60°

can be seen in this figure. The acquired point cloud was smoothed by employing the state of the art software “GOM Inspect” which is commonly used for post-processing 3D-scan data. The point cloud of a single scan contains about 1 Mio. vertices. In contrast, the outer skin of the engine hood is represented by 188.559 FE-mesh nodes. So, the scan is about 5 times denser than the FE-mesh. The smoothed, tessellated point cloud serves as input for the validation. Table 4.2 lists the used parameter sets for the morphing algorithm. In contrast to the verifications, parameter sets depend on the current iteration number. By that, the sets of parameters become more restrictive towards higher numbers of iterations. For consistency with the verifications, also 9 iterations were performed.

The results of the mesh morphing are shown in Figure 4.12. The left picture shows the scanned geometry compared with the FE-mesh which is based on nominal geometry. Differences up to 8 mm can be observed. The right picture shows the scanned geometry compared to the morphing result after 9 iterations. No differences above ± 0.05 mm can be observed.

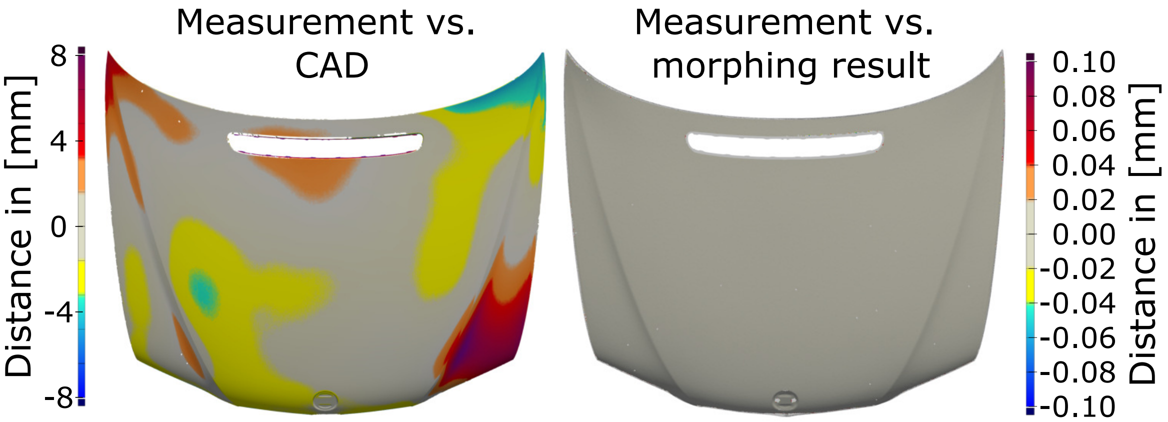


Fig. 4.12 Validation results of morphing FE-model towards real measurement data. Left: nominal geometry compared to actual geometry. Right: morphing result compared to actual geometry. (This figure has been published in [6], changes: color map)

4.6 Run Times and Convergence Behavior

To investigate run times and convergence behavior of the presented approach, different key quantities were evaluated to measure performance.

4.6.1 Run Times

Run times are critical for the use of any approach in real manufacturing systems. As the automated mesh morphing approach is intended to be used in-line, run times are very critical. Run times were measured for all verification and validation experiments performed in this chapter. The results are summarized in Figure 4.13. The computational times are divided into: (a) initial preparation time, (b) pre-processing time of FE-solver, (c) FE-solver time, and (d) interim calculations. Also, the total sum of computing time is shown for each experiment. Depending on the problem size, the total run time differs between 6 minutes and 120 minutes. It is striking that the biggest portion of time is used by the pre-processing of the FE-simulation software. Also, interim calculations take up the second-largest amount of time. “Interim calculations” are containing computing times of calculation steps performed by the algorithm presented in Quotation 4.1.

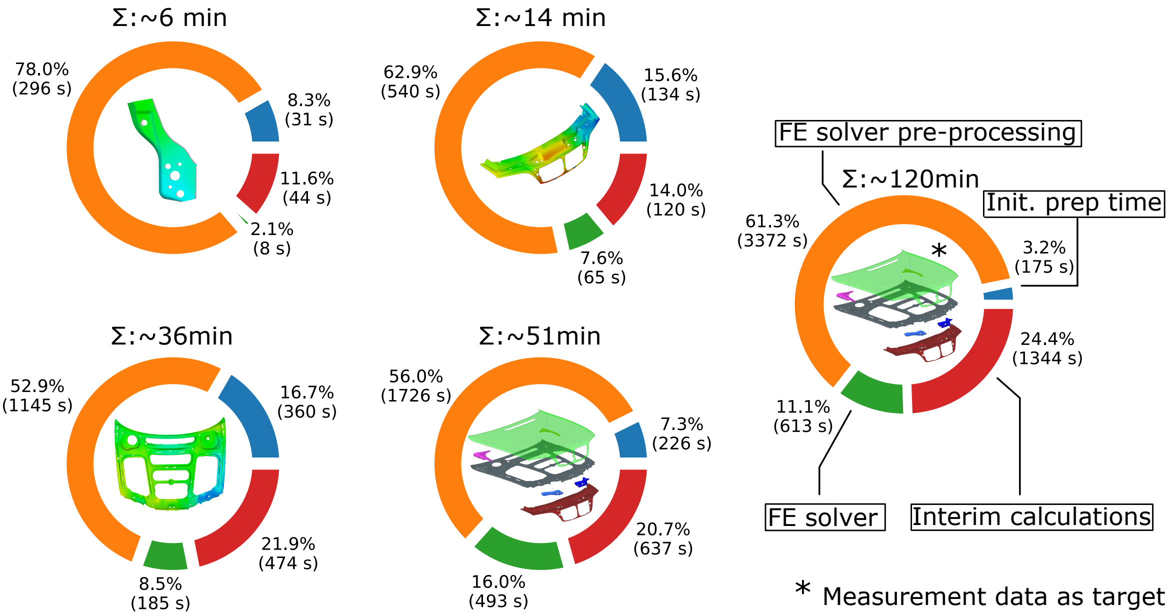


Fig. 4.13 Computational times clustered by: (a) initial preparation, (b) FE-solver pre-processing, (c) FE-solver time, and (d) interim calculations. (This figure has been published in [6])

Table 4.3 Abaqus standard element criteria used for evaluating mesh quality.

Quality Criterion	Value
Shape factor (triangles) less than	0.01
Triangle corner angle less than	5°
Triangle corner angle greater than	170°
Quadrilateral corner angle less than	10°
Quadrilateral corner angle greater than	160°
Aspect ratio greater than	10
Edge length shorter than	0.01 mm

4.6.2 Convergence Behavior

To assess convergence behavior, the RMS-error value is used to measure the distance between geometry of current morphing step and target geometry (measurement). Additionally, the change of element quality is measured. Table 4.3 lists the element quality criteria that were used to identify elements that are still acceptable but almost ill-shaped. The values of Table 4.3 are standard valued from the ABAQUS simulation software [16]. To generate a usable FE-mesh for a DT, both criteria - good geometrical approximation and good quality elements - must be fulfilled at the same time. The convergence behavior is assessed by evaluating and plotting RMS value for each iteration of the mesh morphing algorithm.

Figure 4.14 summarizes element quality and RMS-error value for each iteration of the performed verifications and validation. The upper graph shows the portion of badly shaped elements in %. It can be observed that there is almost no change during the morphing process. The lower graph shows the development of the RMS-error value on a logarithmic scale. All experiments show a converging behavior. The validation experiment with real measurement data shows a slightly higher remaining RMS value compared to the verifications where ideal meshes were used as morph targets.

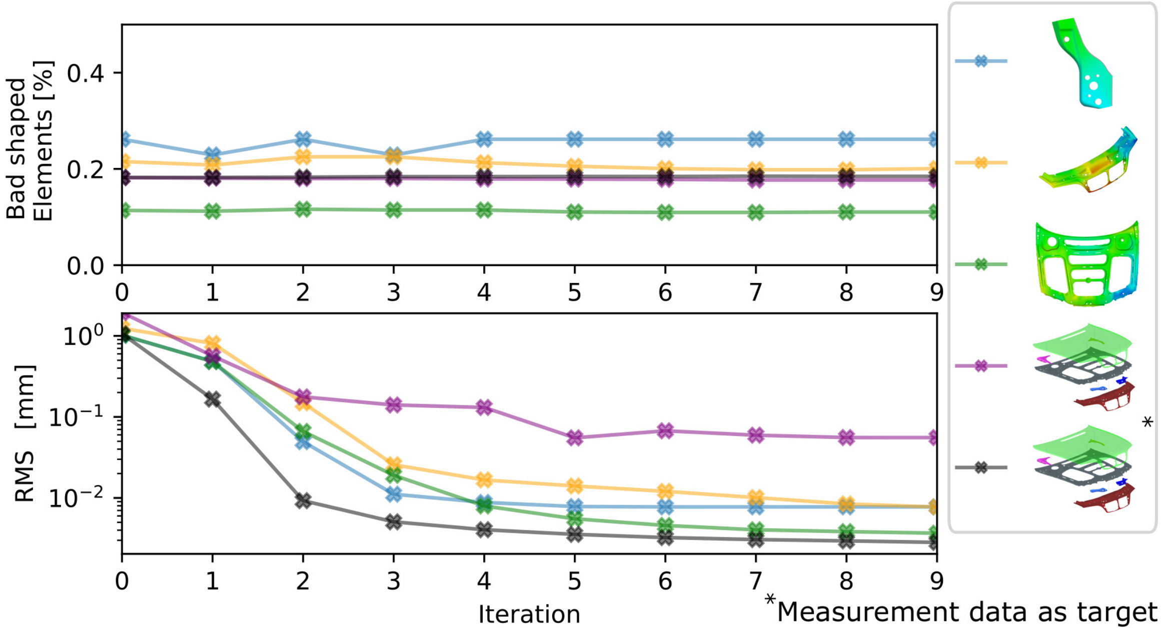


Fig. 4.14 Top: portion of bad shaped elements according to criteria listen in Table 4.3 per iteration. Bottom: RMS-error over iteration on logarithmic scale. (This figure has been published in [6])

4.7 Discussion

This section discusses the achieved results and applicability of the presented approach for smart assembly purposes.

4.7.1 Discussion of Verification Results

The different verification results show the performance of the presented approach under ideal conditions. This means all uncertainties occurring during measurement are neglected by using artificial data as input. So, the behavior and precision of the method can be assessed without any unknown influences. The verification results for single sheet metal parts are shown in Figure 4.3. These results do not show any maximum differences above ± 0.05 mm which does match the pre-defined precision goal for generating geometrical DTs from measured point clouds. The evaluation of this error measure is carried out by considering every single nodal distance value of the final result. This study shows that the presented approach can handle different geometries with various kinds of features like the number of topological perimeters, number of mesh elements, part size, etc. Differences below the scale of ± 0.05 mm can be observed. These minor differences are not relevant for applicability. However, as the conditions are ideal, also minor differences can be explained. Close examination revealed, that the user-defined parameter "n Class a" does influence the local approximation quality. By changing this value accordingly, the density of FE-boundary placements is increased and the local approximation can be improved further. Unfortunately, this parameter also has a critical impact on the computational times. Because of that, smaller approximation errors are accepted to speed up computing speed.

Figure 4.4 shows verification results for the morphing of a partially scanned assembly. In this verification, the target geometry is also generated via FE-simulation. This verification is performed to demonstrate a typical real-world scenario where the sub-structure of a part can not be entirely scanned. The results show that the component that is covered by the measurement can be well approximated. Underlying structure, however, is only morphed indirectly by interacting during the FE-simulation with the outer skin. This kind of morphing is of course not precise, but as the actual geometry of the underlying components is unknown, there is no possibility to achieve a precise DT. However, the verification results show that the approach of morphing these components indirectly via FE-simulation does produce plausible shapes for the sub-components with an error on the scale of ± 0.5 mm. This makes the presented approach superior to the state of the art morphing approaches, as no manual effort is needed to create a usable DT.

Figure 4.5 and Figure 4.6 are showing verifications that were designed to be challenging for the morphing algorithm. These challenging aspects are: (a) profile defects and (b) large lateral displacements. The most critical surface profile defects are dents and buckles. In Figure 4.5 a target geometry was set up with dents of different sizes and magnitudes. The chosen magnitude of almost 2 mm is very high for sheet metal assemblies. Parts with such significant profile defects should normally be sorted out in advance. Only for verification purposes, this test geometry is chosen such extreme. The profile evaluation in Figure 4.5 shows higher approximation errors for small diameter surface defects. The cross-section at the bottom of the figure reveals the reason for this phenomenon. The mesh resolution of the FE-mesh is too coarse to be able to approximate such a small local defect. As no mesh refinement is performed, this approximation error can not be avoided.

Figure 4.6 shows an extreme scenario for large lateral displacements. The geometry is a simple angled piece of sheet metal. One edge of the target geometry is chosen to be 50% longer compared to nominal geometry. The figure shows interim results of the morphing algorithm and the final result after 9 iterations. The node-to-surface comparison shows no deviations above ± 0.05 mm. The bending edge however is not well approximated and cut by element edges. Also, the matching of perimeters is not flawless. The sharp corners are cut off by element edges as well. This is either caused by the re-sampling of the target perimeter curve or by the CPD-registration. Nevertheless, the morphing algorithm provides a usable simulation mesh even under extreme conditions.

Summarized, the verifications show very promising results in terms of flexibility and precision of the presented approach. All achieved results are satisfying and even under extreme conditions, the presented mesh morphing method provides a usable output.

4.7.2 Discussion of Validation Results

To ensure that the presented approach is capable of handling real measurement data, a validation experiment was performed. For this validation characteristics of the customized scan, setup were determined in advance. In particular, measurement noise for different scan settings and repeatability were investigated. The results of these preliminary tests are shown in Figure 4.10 and Figure 4.11. Although measurement noise with a high magnitude is present in the raw scan data, the smoothed point cloud provides a very good approximation of the part's geometry. This noise might become problematic for geometries with plenty of small details, but for a smooth sheet metal part, this scan setup is sufficient. The investigations regarding repeatability are not relevant for the topic in this chapter as only one scan of one state is used for morphing. However, this scan setup is being used for further experiments where repeatability is critical. In terms of repeatability, only minor differences can be observed in Figure 4.11 that are mostly caused by measurement noise. These results are very good considering that the used fixture setup does not meet industry standard requirements.

The validation results of the morphing approach are shown in Figure 4.12. The left picture shows the initial differences between nominal- and actual geometry. These differences get as high as 8 mm, which is significant. The right picture does show the morphed FE-model compared to the measured point cloud. No differences above ± 0.05 mm can be observed. So, the validation results also fulfill the pre-defined quality goal and are considered very good. Even a low-frequency noise is approximated which can be observed when inspecting the results in a 3D-render view.

4.7.3 Discussion of Run Times and Convergence Behavior

Convergence behavior and run times were measured to reveal characteristics of the algorithm as well as potentials for improvement. In Figure 4.14 the convergence behavior is described by plotting change of element quality and RMS-error over iterations. While element quality remains almost untouched, the RMS-error is reduced. Especially, during the first three iterations, most of the approximation does take place. This behavior could be used for reducing the number of iterations resulting in faster run times.

Measured run times can be found in Figure 4.13. The total computing time does vary between 6 min and 120 min. This discrepancy is related to the FE-problem size. It is striking, that the pre-processing time of the solver takes up the most portion of the total computing time. This step is not parallelized by the software ABAQUS and can not be improved without changing the FE-solver. However, this step has a huge potential for time-saving and should be considered when using such an approach for time-critical applications. When comparing computing times of verification and validation a noticeable difference can be seen. This difference is caused by the higher density of the measured target point cloud which must be handled during the interim calculations. Also, the density of the FE-boundary placement impacts the pre-processing and solver times drastically.

4.7.4 Applicability for Smart Assembly Processes

The presented verifications and validations have shown, that the FE-based mesh morphing approach does provide precise geometrical DTs from scanned data. For applications where computing time is not critical, this mesh morphing approach is extremely helpful as it does not need manual work. However, in terms of run times, the approach is not practical especially for in-line usage. Based on the current state of the art, using a high-fidelity DT for predicting best fitting assembly parameters in real time seems to be impracticable due to handling and processing big amounts of data. However, this approach might become applicable in the future due to increasing computational power, and more efficient simulation software. Also, the timings that must be achieved are highly dependent on the sequence of process steps in the manufacturing system. For instance, if a component is produced by an external supplier, generating the DT and compensating gravitational influences could already be performed while the part is transported to the assembly line. As soon as the assembly partner is digitized, also the best fitting assembly parameters could be estimated in advance.

Core References of this Chapter

[6] Felix Claus, Bernd Hamann, and Hans Hagen. A finite-element based mesh morphing approach for surface meshes. *Computer-Aided Design*, 146:103232, 2022

Chapter 5

Validation and Troubleshooting of Smart Assembly Processes with Human in the Loop Approaches

5.1 Introduction

This chapter is based on core reference [4] and [5]. Also, relations with core references [7], [8], and [9] are briefly highlighted. Chapters 3 and 4 discussed methods for generating high-fidelity DTs from optical measurement data. These DTs can be used to predict assembled state of components and optimize assembly parameters for each individual measured geometry. Solving this optimization problem is not discussed in this dissertation as this kind of optimization is closely related to spring back optimization problems, see [142]. For the present use case, a simple linear response optimization is sufficient to solve the assembly spring back problem. In this chapter, the aspects of troubleshooting when implementing a DT for self-compensating assembly lines are addressed. Therefore, working recognition-, prediction- and action- systems are presumed, as well as a post-assembly quality gate where geometrical checks are performed. If deviations from the desired target geometry are observed, the smart assembly system must be analyzed to detect and correct errors within the system. This troubleshooting process is extremely challenging because the smart assembly system is composed of many subsystems which are clustered across many engineering disciplines, i.e. measurement systems, robotics, computer science, material science. Also, the interplay between these subsystems must be considered for the troubleshooting process. To analyze such a complex system based on a geometrical deviation between predicted and actual geometry, correlations between observed deviations and characteristics of the smart assembly system must be identified. This problem is tackled by making predictions of the positions of all mechanical interfaces based on a post-assembly scan. To isolate errors, the result of such prediction can be analyzed further by comparing with simulation results from the DT.

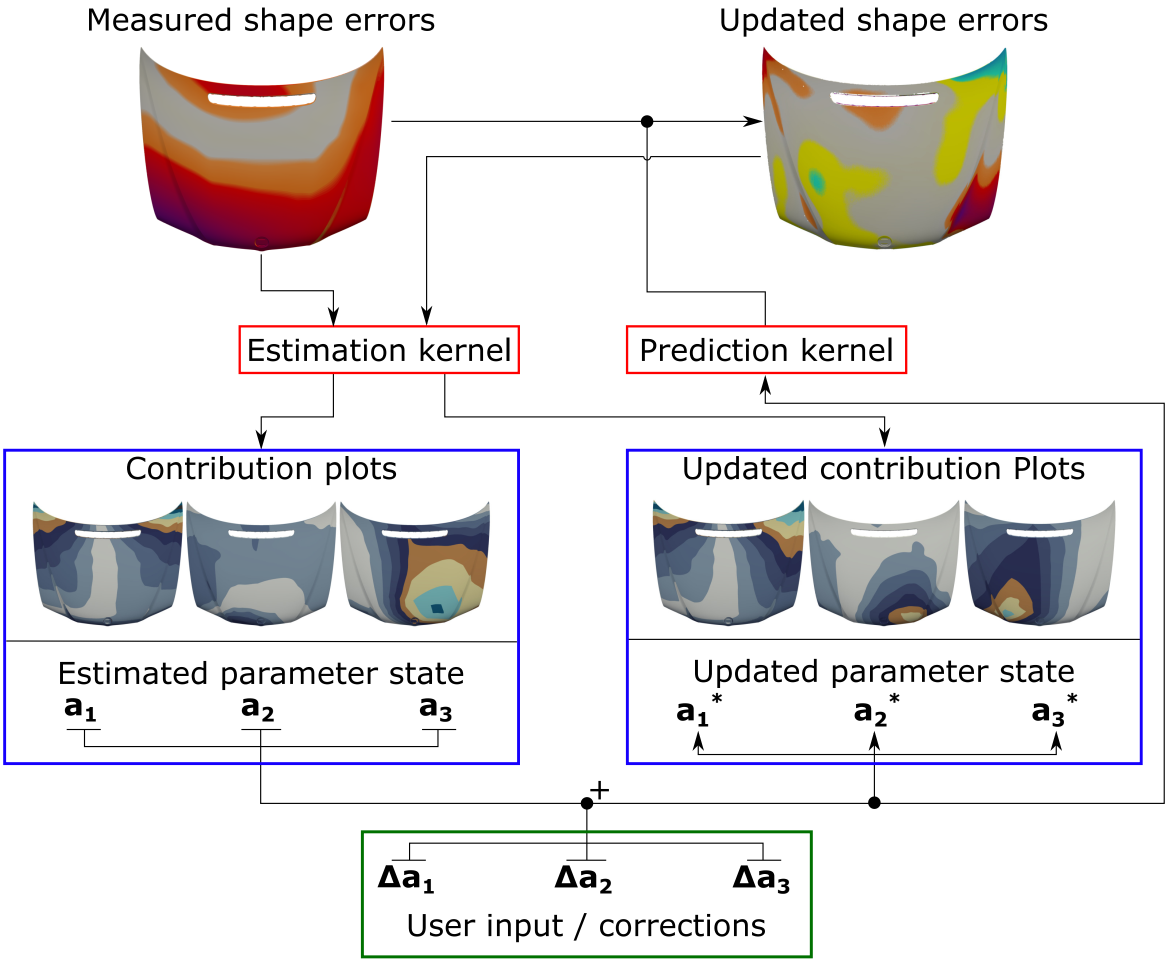


Fig. 5.1 Workflow of the interactive troubleshooting tool and integration of estimation- and prediction kernel. As input serves a scalar field of measured shape errors. The estimation kernel calculates contribution plots and estimated parameter states. The user can apply changes (green) which trigger an update of shape errors using the prediction kernel.

Besides identifying errors, deriving adequate corrections is also not straightforward due to complex part behaviors. To support the inspection engineer in finding the best possible countermeasures, this chapter provides an interactive exploration approach combined with special decomposition visualizations. A high-level illustration of this approach is given in Figure 5.1. The input to the troubleshooting method are the measured shape errors. For the measured errors, an estimation kernel calculates the parameter states a_1-a_3. Each parameter is related to a position of a mechanical interface. Also, the estimation kernel provides contribution plots. These plots decompose the contribution of each mechanical interface to the total observed error. The user can interact with the software by virtually applying corrections to each parameter. These corrections are fed to the prediction kernel which predicts the geometrical changes of the parts shape. An updated view of the shape errors is provided in real time. Also, the contribution plots are updated via the estimation kernel. The real-time update of the shape errors enables an explorative and interactive troubleshooting experience for the inspection engineer.

This chapter is structured as follows: Section 5.2 provides a literature review of relevant topics and state of the art approaches. Section 5.3 describes the method which composes of contributions of the publications [4] and [5]. The presented approach is verified in Section 5.4 and validated with experimental data in Section 5.5. Computational times are shown in Section 5.6. Section 5.7 composes chosen approaches into a interactive trouble shooting tool. Last, this chapter is discussed in Section 5.8.

5.2 Background

This background section summarizes literature that is related to either: (a) identifying sources of errors, (b) estimating parameter states, or (c) interactive exploration approaches.

5.2.1 Root Cause Analysis

A high-level introduction to RCA was already given in Section 1.3.3. This section briefly highlights the state of the art, especially for the use of RCA in sheet metal assemblies.

In [34] the RCA approach is presented to identify fixture-related errors in multi-stage assembly processes. When performing RCA, all possible factors that can influence the output of the process are gathered and analyzed. For sheet metal assembly processes, typical factors are: (a) assembly tool variation, (b) part variation, and (c) order of fastening screws or spot welds. To minimize variations of process chain output, different optimization methods are available in the literature for improving mentioned impact factors: [41, 39, 143, 144, 40].

5.2.2 Parameter Estimation

Parameter estimation is used for different kinds of applications and can be carried out with various methods, see [145–147]. A common application is the tuning of simulation models based on experimental data. Parameters, i.e. material properties, are derived from experiments to improve the precision of the corresponding simulation model, see [148] and [149]. In [150] parameter estimation is used as a prediction system for automotive applications. The predictions are used to detect and prevent critical motion states. A widely used method for predicting parameters based on experimental data is LSQF[1]. An overview is given in [151].

Proper Orthogonal Decomposition

For performing a parameter estimation via LSQF a suitable basis must be generated that condenses the whole solution space. The generation of such a basis can be carried out differently. The most common branch of methods is called POD[2]. POD is used to find a lower-dimensional base for a high-dimensional solution space. A review of POD methods and their applications can be found in [152]. In [153] POD is used in conjunction with LSQF to reconstruct airfoil pressure fields based on a single measurement point on the surface of the airfoil. While the decomposition of the solution space is typically carried out with large simulation ensembles, [55] uses multiple 3D-scans of sheet metals for the decomposition to reconstruct not measured areas. By that, the process of digitizing actual parts with 3D-scanning can be sped up significantly. The most commonly used method from the branch of POD is called PCA[3]. In [154] a PCA-based approach is used to predict statistical shape deviation errors in sheet metal assemblies. A suitable basis can also be generated by performing a modal decomposition which is a physics-based approach. The concept behind modal decomposition is, that the shape of a part can be expressed as a combination of its eigenmodes. This has the great advantage that no large simulation ensemble is needed for a decomposition. Especially, for improving sheet metal assemblies, this concept is often used, see [155–158].

Machine Learning Approaches

To predict parameter states from measurement data or perform classifications, ML approaches are often used in recent publications, see [159–161]. A great advantage of ML approaches is

[1] Least Squares Fit

[2] Proper Orthogonal Decomposition

[3] Principle Component Analysis

this: instead of finding a suitable basis for performing an LSQF, a ML model can be trained to perform the predictions, which is more generally applicable. For the application of sheet metal assemblies in conjunction with 3D-scan data, the usage of ML is not straightforward because of the large data sizes that must be handled. If the ML model is not chosen carefully, excessive coarsening of the point clouds and huge computational costs are necessary to train the ML model, see [162]. On the other hand are ML models appropriate for this use case if the ML approach is well chosen. For instance, a graph convolution neural network does perform extremely well for the use case of predicting assembly parameters from 3D-scans. This approach can handle big data sets while maintaining efficient training and high precision outputs, see [163, 7].

5.2.3 Interactive/Explorative Design

Interactive design exploration approaches are employed to find the best fitting design by user interaction. This approach can also be used for engineering tasks with a wide range of applications, see [164, 165], [166]. Explorative approaches can be coupled with physics-based simulations to ensure correct behavior of the model and provide additional information to the user, see [167, 168]. To provide complex simulation results in real time, data exploration techniques can be applied to pre-simulated data ensembles, see [169].

5.2.4 Resume

The methods that are presented in this chapter are designed to support the troubleshooting of errors in smart assembly processes by employing an interactive human-in-the-loop approach. The main branch of research related to these approaches is RCA. The highlighted approaches from the literature provide general approaches that gather all possible aspects that can impact the process being assessed. In the general case, this is the preferable way of troubleshooting. However, since sheet metal assemblies react extremely sensitive to changes of interface position the troubleshooting approaches being presented in the following focus on identifying misplaced interfaces. Therefore, a parameter estimation based on PCA and LSQF is used to reconstruct interface positions from a 3D-scan. Next, an interactive exploration tool is employed that supports the engineer in finding the best possible corrections. The interactive tool does extend by additional decomposition views that provide information about interfaces that have a major contribution to measured deviations. This kind of troubleshooting is novel for the use case of improving sheet metal assemblies.

5.3 Method

To realize the interactive troubleshooting tool described in Figure 5.1, the following challenges have to be tackled: (a) setting up an estimation kernel, (b) extending the estimation kernel to make it capable of providing contribution plots, and (c) setting up a prediction kernel. This method section provides solutions for these challenges and additionally discusses how the partial solutions are brought together.

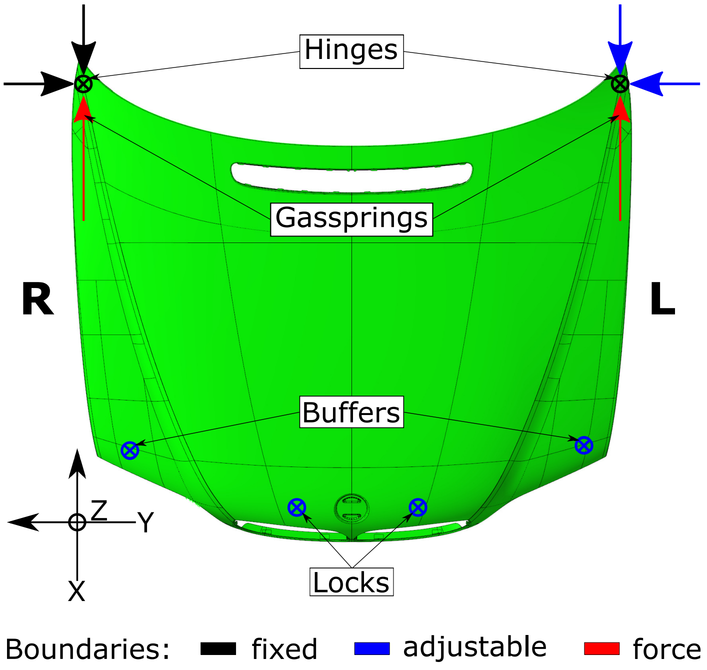


Fig. 5.2 Repetition of Figure 4.8 - Position of mechanical interfaces. The hood is connected by hinges, buffers, locks - two each - with the chassis. Two additional gas springs are pressing with 580N each against the hinges in closed state. Also, orientation left (L) and right (R) are marked in the figure. (This figure has been published in [4], licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/; changes: only one subfigure is depicted.)

5.3.1 Estimation of Parameter States

The estimation of parameter states is the foundation of the troubleshooting system. In the following, a parameter is related to a directional position of a mechanical interface (i.e. hinges, locks, buffers). Knowing which mechanical interface is misplaced in what direction is crucial for deriving countermeasures to improve assembly quality. To estimate the parameter states, a scalar field of measured shape errors serves as input for the estimation kernel. This scalar field is obtained by comparing a post-assembly 3D-scan to the predicted ideal assembly state. The estimation kernel can be set up with different methods. In the following, three different estimators are described and compared - one PCA-based and two ML approaches. All three approaches have in common that a large simulation ensemble is needed for training. This ensemble provides information about the behavior of the parts under different load cases.

Simulation Model and Parameters

For demonstration, the simulation model of the engine hood that was already introduced in Section 3.5 is used. Figure 5.2 shows the mechanical boundaries that are applied to the simulation model. In the figure, also the directions and positions of mechanical boundaries considered as adjustable are marked in blue. In the following, each adjustable boundary is considered as a simulation parameter that can be varied in a certain range. Parameters and value ranges are summarized in Table 5.1. To reduce the number of parameters, the right hinge is considered to be fixed. Also, the movement of the hinges in the Z-direction is restricted. This is done as only relative motions between the hinges induce stresses to the part. Pure translations or rotations could be removed by pre-processing steps like rigid registration. In the following, a simulation ensemble is created by sampling the parameter space that is spanned by a_1-a_6.

Table 5.1 Simulation parameters and corresponding value ranges.

Simulation parameter	Label	Description	Value range [mm]
a_1	HINGE_X	Hinge left, X-direction	[-1;1]
a_2	HINGE_Y	Hinge left, Y-direction	[-1;1]
a_3	BUFFER_L	Buffer left, Z-direction	[-1;1]
a_4	BUFFER_R	Buffer right, Z-direction	[-1;1]
a_5	LOCK_L	Lock left, Z-direction	[-1;1]
a_6	LOCK_R	Lock right, Z-direction	[-1;1]

Generation of Simulation Ensemble

To obtain comprehensive information about the behavior of the part, a simulation ensemble is created by running multiple simulations with systematically chosen sets of parameters. The simulation model has n adjustable parameters a which can be varied within a defined interval $[min, max]$. One simulation result $\vec{u}$ with a defined set of parameters $\vec{a}$ is a discrete sampling point of the solution space. So, the FE-simulation can be considered as a map from parameter space to solution space:

$$\vec{u} = \vec{u}(\vec{a}),\\ \vec{a} = (a_1, a_2, ..., a_n), a_i \in [min_i, max_i] \tag{5.1}$$

Equation 5.1: Displacement field $\vec{u}$ as function of parameter set $\vec{a}$. Each individual parameter a_i is restricted to the interval $[min_i, max_i]$.

To ensure an optimal sampling of the parameter space while maintaining a manageable number of simulation runs, a sampling strategy needs to be defined. For the present application, two simulation ensembles of different sizes are created. The first ensemble is created with a full factorial sampling strategy and serves for training purposes. The second ensemble uses LHS and serves as verification ensemble.

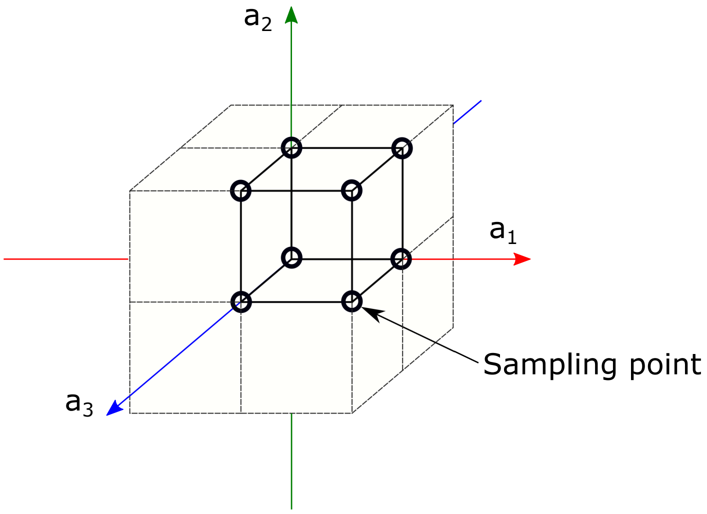


Fig. 5.3 Full factorial sampling strategy with three values per axis, demonstrated for the 3D-case. The corners of a resulting hypercube are highlighted by black circles. (This figure has been published in [4] licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/, changes: variable names)

Full Factorial Sampling

The training data set is sampled by selecting a fixed number of values for each axis. The principle of this sampling strategy is illustrated in Figure 5.3 for the three-dimensional case. Here, three values per parameter a_i are chosen, resulting in $3^3 = 27$ sampling points. This sampling scheme divides the parameter space into 8 cubes. Such structured sampling is fundamental for generating the contribution plots. More details follow in Section 5.3.2. When extending the sampling scheme to a higher number of parameters i or extending the number of sampling points per axis n, the total number of sampling points n_{total} can be calculated with $n_{total} = n^i$. For the example of the engine hood, 6 adjustable parameters are present. Each parameter axis is sampled with three values, so the total number of sampling points (possible combinations) is $3^6 = 729$. The sampling structure of this higher dimensional case can still be described as clustering into n-cubes, also called “hypercubes”. Table 5.2 lists characteristics of hypercubes of different dimensions. For the 6 dimensional case the sampling of the parameter space is clustered by 64 6D-hypercubes.

The scheme of full factorial sampling is used to generate the training data set. Each parameter axis is sampled with the values [-1,0,1]. Additionally, each center of mass of the 6D-hypercubes is sampled. This leads to a total of 793 simulation runs being performed for generating the training data set. The distribution of the sampling points in the parameter space is shown in Figure 5.4.

Latin Hypercube Sampling

n	Name	0-face (vertex)	1-face	2-face	3-face	4-face	5-face	6-face	...
0	Point	1							
1	Line	2	1						
2	Square	4	4	1					
3	Cube	8	12	6	1				
4	Tesseract	16	32	24	8	1			
5	Penteract	32	80	80	40	10	1		
6	Hexeract	64	192	240	160	60	12	1	
⋮									

Table 5.2 Elements of an n-dimensional hypercube. This table can be calculated by the sequence A038207 that can be found in "The On-Line Encyclopedia of Integer Sequences®" (OEIS®) [170]

As full factorial sampling does grow exponentially with the number of varied parameters, the number of samples gets impractical very early. For instance, when going from 6 parameters up to 7 the number of sampling points grows from 729 to 2187. Depending on the run time of each simulation run, a full factorial sampling becomes impractical because of excessive computational costs. For this kind of problem, LHS can be applied. LHS was introduced in [118] and provides a samplings strategy that distributes a pre-defined number of samples equally spaced in the high dimensional parameter space, for details see [171]. LHS is used to create a second simulation ensemble with a maximum sampling coverage of the parameter space for verification purposes.

The verification data set was created by sampling the same parameter space given in Table 5.1 by a fixed number of 1400 samples with LHS. The distribution of samples is also given in Figure 5.4. With the generated training and verification data set, the estimation kernel can be set up.

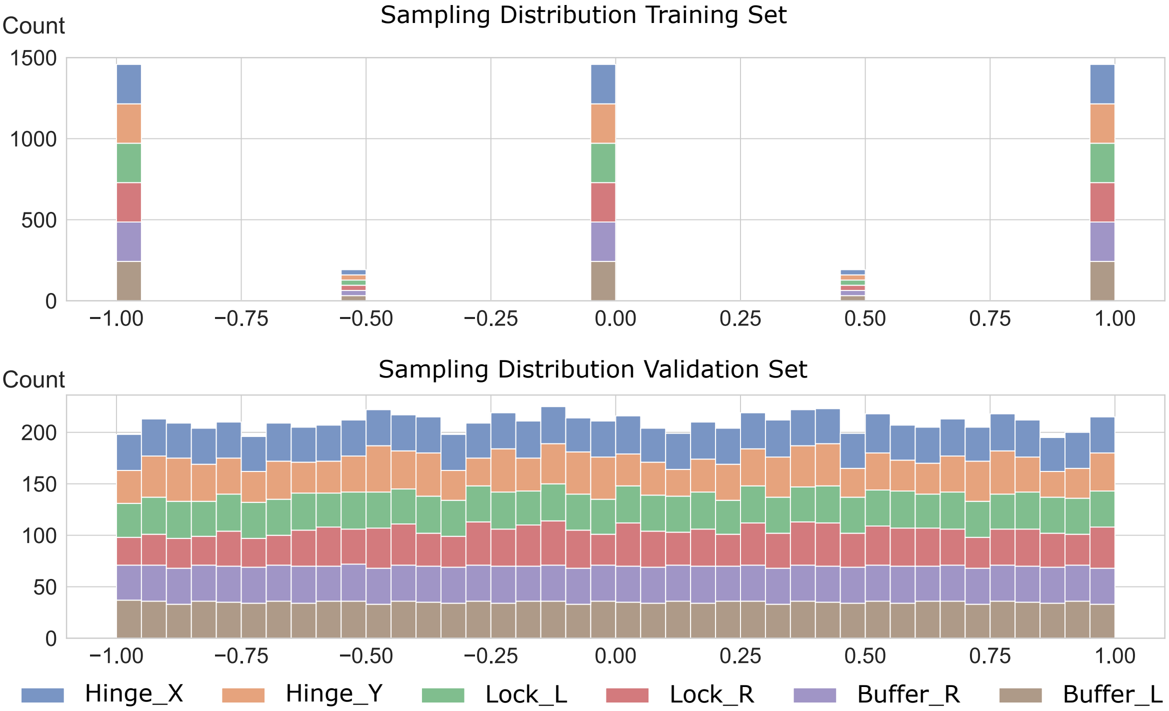


Fig. 5.4 Stacked histogram of parameter value distributions in the sampoling range of ± 1 mm. Top: training data set using systematic sampling strategy. Bottom: verification data set using LHS.

Estimation via least-squares fitting

A commonly used approach to estimate or reconstruct data is LSQF. When using LSQF, the data that is being reconstructed is expressed as linear combination of a chosen basis. LSQF calculates the best set of linear factors $\vec{c}$ for a given basis $\vec{B}$ and target data $\vec{u}$, see Equation 5.2. Depending on the information contained in the basis, not only reconstructions of given displacements but also estimations of parameters can be performed. Considering the engine hood, each basis field B_i has the same form as a simulated displacement field. So, the linear combination calculated by the LSQF can be viewed as a linear combination of displacement fields. When adding parameter space information to each basis, the LSQF result can also provide an estimation of parameter values, only based on inputs in the solution space. This

$$\vec{u} \approx (c_1, c_2, ..., c_m) \begin{pmatrix} \vec{B}_1 \\ \vec{B}_2 \\ ... \\ \vec{B}_m \end{pmatrix} = c_1\vec{B}_1 + c_2\vec{B}_2 + ... + c_m\vec{B}_m = \sum_{i=1}^{m} c_i\vec{B}_i, \quad (5.2)$$

Equation 5.2: Approximation of the displacement field $\vec{u}$ by a linear combination of the basis members $\vec{B}_i$.

way, the LSQF method can be used to perform parameter estimations. For clarification, Figure 5.5 is provided. In the figure a target displacement field $\vec{u}_{target}$ is approximated with a linear combination of $\vec{u}_{B1}$-$\vec{u}_{B3}$. The linear factors c_1-c_3 are obtained via least-square fit. The same factors are used to calculate the linear combination for the parameter space, resulting in the unknown estimated parameter values $\vec{a}_{est}$.

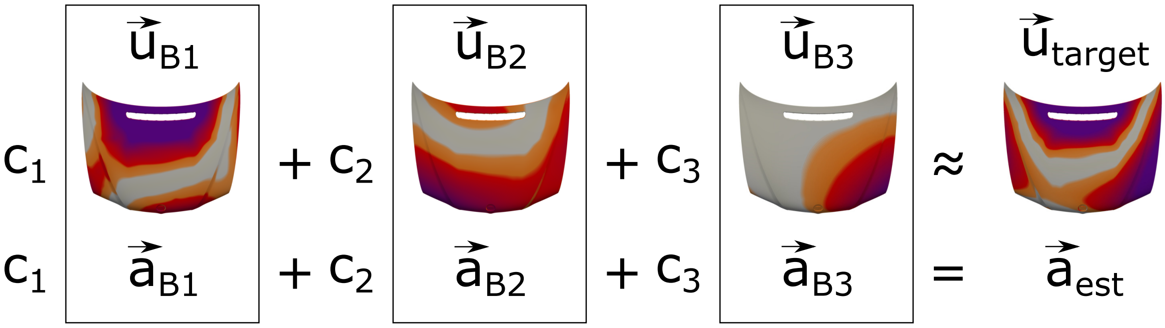


Fig. 5.5 Illustration of the parameter estimation via LSQF for a given displacement field $\vec{u}_{target}$. The pre factors c_i are calculated by fitting the basis fields $\vec{u}_{B_i}$ to the target field. The same linear combination can be set up for the parameter space resulting in the unknown parameter state $\vec{a}_{est}$.

The challenging part of approximating parameter states via LSQF is not finding the best set of linear factors but finding a suitable basis $\vec{B}$ that does condense the solution space into a small number of basis fields while maintaining optimal coverage and precision in terms of approximation.

Generation of Basis for LSQF

As mentioned in Section 5.2, different approaches of POD can be used for dimensionality reduction of a simulation ensemble. For the engine hood, either modal decomposition or PCA is preferable. Although modal decomposition is applicable due to the simulation of mechanical deformations, a PCA approach was chosen to ensure general applicability to simulation ensembles. To apply PCA to an ensemble of simulated displacement fields, the data set must be pre-processed. The reason for the pre-processing is this: the PCA method can only handle 3D-scalar fields but not 3D vector fields. Therefore, a displacement magnitude along the geometry surface normal is calculated for every mesh node. By using this signed normal metric, the resulting scalar fields are comparable with surface-to-surface comparisons obtained from scan data. After condensing the ensemble of vector fields to an ensemble of scalar fields, PCA can be applied.

PCA is performed on the pre-processed training data set with 793 ensemble members. The result of the PCA is plotted in Figure 5.6. The upper bar chart shows the explained variance in percent per PC[1]. It can be observed, that most of the variance is explained by the first 5 components. To ensure not to lose any relevant information, the first 10 PCs are used as a basis for further calculations. These 10 components accumulated, explain 99.99985% of the variance in the training data set. The lower half of Figure 5.6 does show all 10 PCs in the solution space on adequate scales. These displacement fields are obtained by calculating a linear combination of all ensemble members. Each set of linear factors is provided by the PCA output. These linear factors can be applied to the ensemble of 3D-vector fields and also to the corresponding parameter space. To match with measurement data, only the portions of displacements along surface normal are considered. Each PC is a set of 3D-vector fields and corresponding parameters. However, these PCs do not reflect actual simulation results or the physical behavior of the engine hood. They are artificially created by the PCA to reduce the solution space optimally. Nevertheless, the underlying characteristics are partially correlating with behaviors of the engine hood. The calculated PCs 1-10 can be used for least-squares parameter estimation.

Estimation via Machine Learning

In contrast to the deterministic parameter estimation via least-squares, ML approaches are becoming very popular for these kinds of applications. When carried out clever, ML has the great advantage that it can be trained to efficiently perform parameter estimations although the data set features non-linear behavior. Although ML approaches are not considered as a contribution of this dissertation, the co-authored papers [7–9] are worth referencing at this point, because they address the challenge of setting up ML models for parameter estimation using the exact same data set of the engine hood. In the referenced publications, two ML approaches - a one-layer feed-forward and graph convolution neural network - are selected and benchmarked with the linear least-squares parameter estimation. For technical details like training and selection of hyperparameters, the original papers are referred. All approaches were tested with the verification data set (1400 ensemble members). For each approach

[1] Principal Component

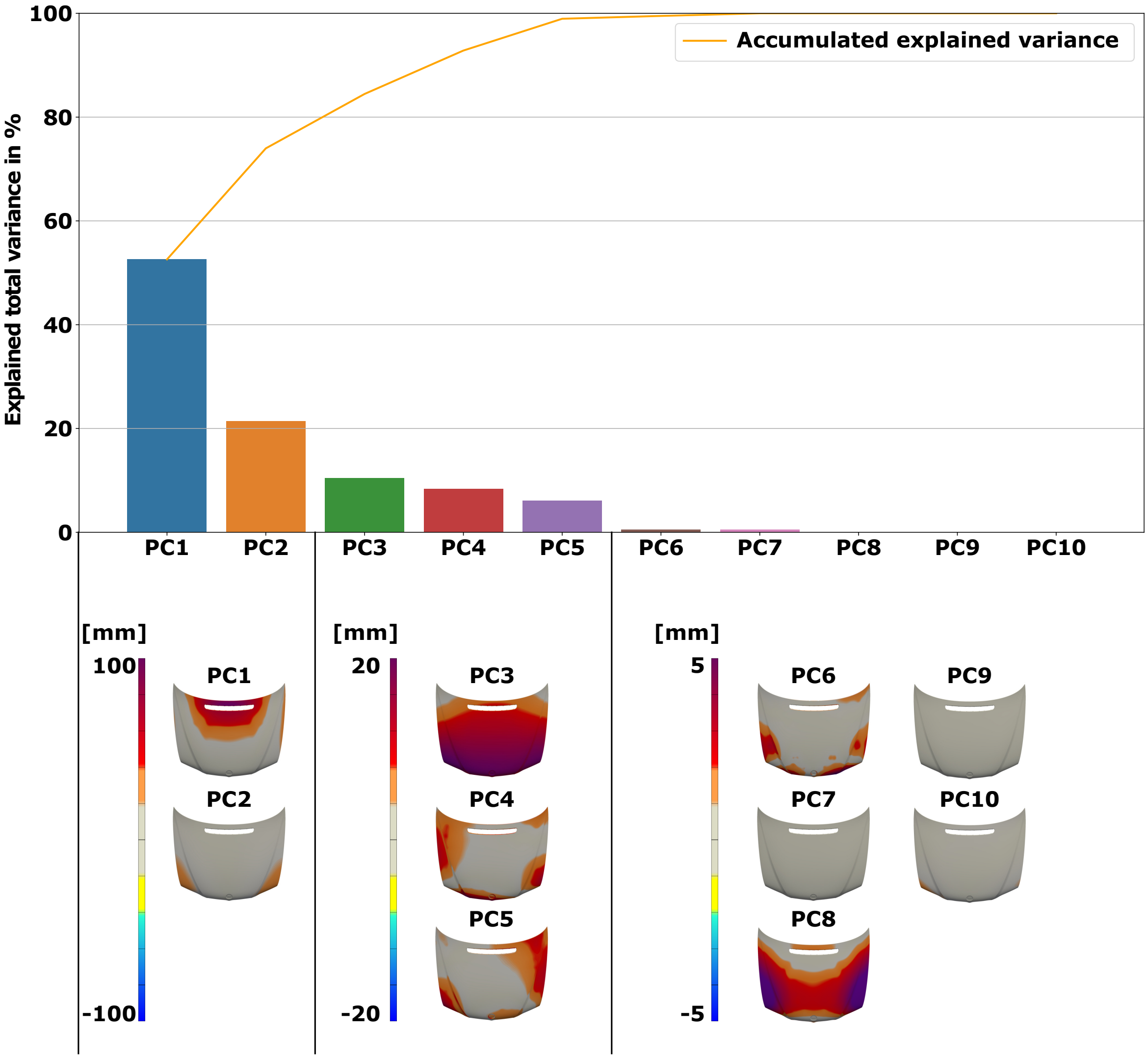


Fig. 5.6 Result of the principle component analyses performed on the training ensemble. The graph shows explained variance of each component in %. Bottom: Principle components in the 3D-domain on three different scales. By a linear combination of PC 1-10, 99.99985% of the training data set can be reconstructed.

parameter estimation was performed for every single ensemble member. The results are shown in Figure 5.7. It can be observed that both ML approaches do perform worse compared to the parameter estimation with LSQF (orange). Although these results do not emphasize the need for an ML approach for parameter estimation, ML approaches are more generally applicable and can handle more complex data, where PCA and LSQF become useless. A qualitative comparison between analytical and ML approaches is listed in Table 5.3. In the following, LSQF together with the basis calculated via PCA is used for performing estimations, as this approach does perform superior for the present data set.

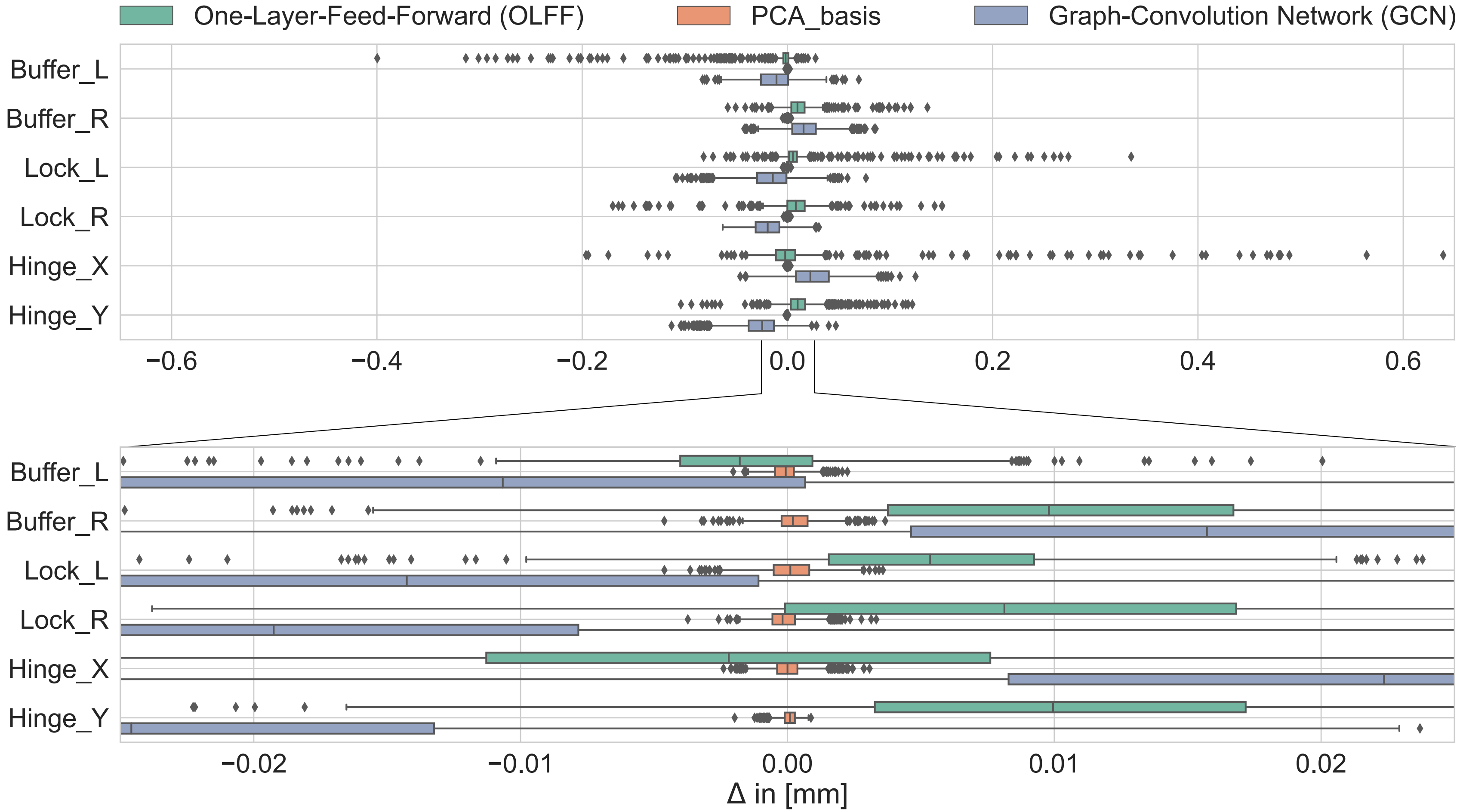


Fig. 5.7 Comparison of different parameter estimation approaches using the same data for training and testing. The PCA-based approach used in this dissertation is superior in terms of approximation erros for the present application.

Table 5.3 Qualitative comparison between analytical and ML-approaches.

Benchmark Aspect	Analytical Approach	Machine Learning
Generating Training Data	-	- -
Training Time	++	- -
Expertise Required for Training	+	-
Computing Resources (Training)	++	- -
Computing Resources (Execution)	++	++
Transparency	+	- -
Flexibility (Adjustments)	+	-
General Applicability	- -	+ +

5.3.2 Decomposition of Scanned Deviations

Another way for creating a basis for performing an LSQF is described in [5]. This approach was designed to decompose the solution space into basis fields that can be interpreted and assigned to misplacement of single mechanical boundaries or combinations of boundaries. The resulting basis size is way larger compared to PCA decomposition. However, due to the interpretability contribution plots can be derived from an LSQF performed with this larger basis.

Quotation 5.1: Chapter 3 excerpted from [5][1]

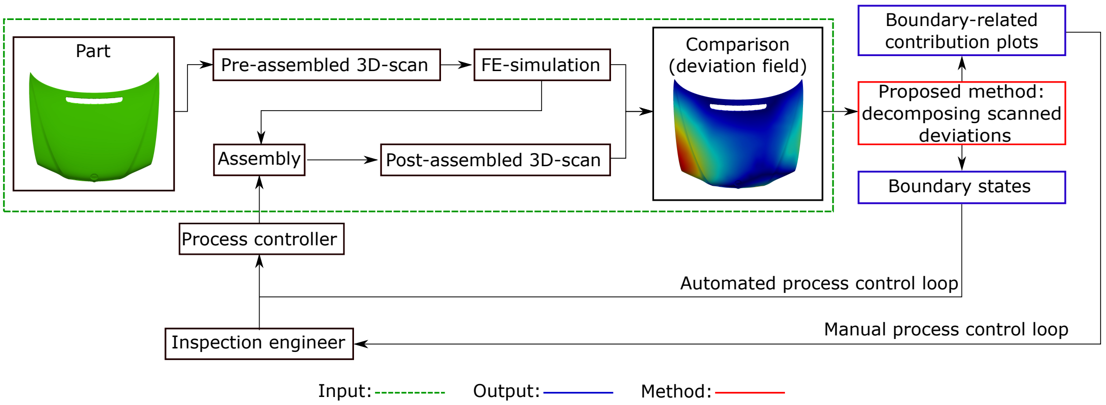

Part
Pre-assembled 3D-scan
FE-simulation
Comparison (deviation field)
Boundary-related contribution plots
Proposed method: decomposing scanned deviations
Boundary states
Assembly
Post-assembled 3D-scan
Process controller
Automated process control loop
Inspection engineer
Manual process control loop
Input:
Output:
Method:

Figure 1: Overview of example assembly process and two possible control loops. As input serves a comparison between simulated and measured assembly geometry (green). The proposed method (red) predicts from the deviation field, corresponding boundary states (blue) as well as boundary-related plots (blue). It can support automated process control as well as human-in-the-loop process control by providing additional insight into the measured deviation field.

⋮

[1] This content has been published in [5], licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/; changes: none

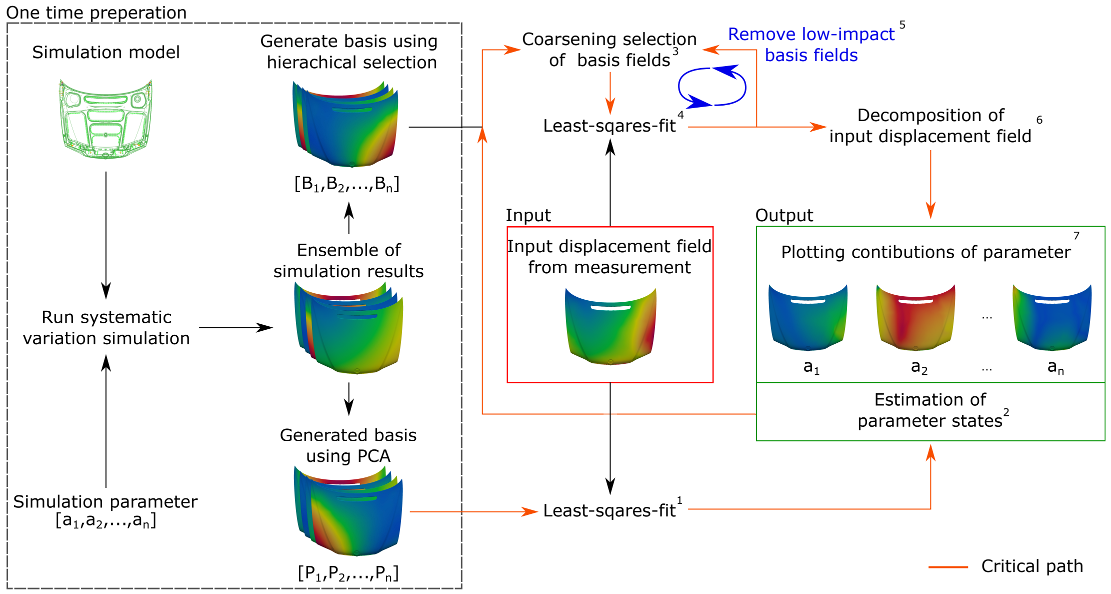

One time preperation
Simulation model
Generate basis using hierachical selection
[B1,B2,...,Bn]
Ensemble of simulation results
Run systematic variation simulation
Generated basis using PCA
Simulation parameter [a1,a2,...,an]
[P1,P2,...,Pn]
Coarsening selection of basis fields 3
Remove low-impact basis fields 5
Least-sqares-fit 4
Decomposition of input displacement field 6
Input
Input displacement field from measurement
Output
Plotting contibutions of parameter 7
a1
a2
...
an
Estimation of parameter states 2
Least-sqares-fit 1
Critical path

Figure 2: Workflow. One-time preparation, input and output of the method. The one-time preparation step concerns simulation set-up and performing simulations to create basis fields. The input displacement field is approximated via a least-squares step.

⋮

3. Method

Our goal is to decompose a displacement vector field generated by comparing a measurement of a post-assembly geometry with the predicted desired geometry. As an example, one can consider an FE-simulation performed with the pre-assembly shape to predict the optimal assembly. The described decomposition of the vector field is not a vector field decomposition in the mathematical sense. It should be viewed as a reconstruction based on multiple FE-simulations using varying parameter states.

We provide an overview of our approach with a flowchart and a detailed description of the mathematical aspects. Figure 2 shows the workflow that considers “One-time preparation," “Input" and “Output." The path marked in orange is the “critical path." Steps 1–7 along this path decompose the input field. The workflow is explained at a high level, but we also cover the mathematical details. As one-time preparatory steps, we set up an FE-simulation model of the part we must inspect. This model needs to be validated to ensure that it reflects the behavior of the real part. The geometry of the simulation model is generated via a CAD system. Although the geometric stiffness slightly differs between the simulation model and the real part, we assume that this difference in stiffness is negligible for our purposes. This aspect, with examples, is discussed in Claus, Hagen and Hamann (2021a). Nevertheless, the geometrical deviations of a real part relative to CAD model must be taken into account during the pre-processing step of the input to isolate measured displacements caused by misplaced boundaries, see Figure 1. Next, we need to identify the parameters that are considered adjustable during the assembly process. When these parameters and their ranges are known, the simulation is performed with a systematic variation of parameter values. The resulting ensemble of simulation results is used to generate two different bases for least-squares-fitting. Both are capable of approximating the whole solution space within a defined tolerance. One basis is set up by performing a PCA for the results of the whole simulation ensemble. The basis generated by PCA is an ensemble of computationally generated synthetic displacement fields with corresponding parameter states. This basis is ideal for estimating the parameter states of the input displacement field. The second basis is generated by a hierarchical selection of important simulation runs from the ensemble. This basis is not ideal for approximating the parameter states, but it has the significant advantage that each single basis field can be assigned to one parameter. The desired contribution plots can be generated from this basis.

We now consider the input to our method. The input is a displacement field calculated by comparing a measurement with the desired ideal. This comparison results in a scalar distance field. To obtain a displacement vector field all scalar values at the simulation's mesh nodes are multiplied with an associated normal vector. The calculation of the displacement magnitude is not unique, as distance values can vary slightly when changing reference and target geometries. This problem is depicted in Figure 3. For our purposes, we require results for every node of the simulation mesh. The distance metric used is the “Cloud to Mesh Distance" implemented in the software package “CloudCompare" Cloudcompare (2020). It calculates the distance from each vertex of the source mesh to the nearest triangle of the target mesh. The position of the vertex and the triangle's outward normal are used to (i) determine whether the vertex is inside or outside the triangle and (ii) compute a signed distance to the plane containing the triangle. One could consider using more accurate and complex distance measures like the Hausdorff distance Aspert, Santa-Cruz and Ebrahimi (2002). Point set registration could also be performed Kenta-Tanaka et al.. We use the simple CloudCompare metric as it provides sufficiently precise results for our application. For more complex geometries a different metric might be necessary.

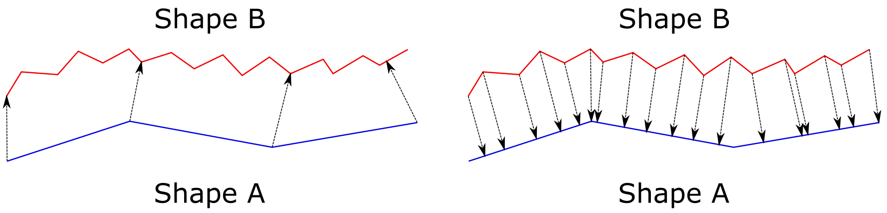


Figure 3: Calculating distances between two meshed shapes can lead to different results depending on the choice of distance metric.

We now explain the steps to process one input field. The 1-7 along the critical path are discussed. In (1), the input displacement field is used as a target for a least-squares-fit using the basis generated by PCA. This fit produces “pre-factors" for each basis from which we can estimate the parameter states (2) by multiplying the pre-factors with the corresponding basis parameters. These estimated parameters are one output that is communicated to the user and, at the same time, used for further calculations. Based on the estimated parameters a subset of basis fields (3) is selected from the hierarchically generated basis by excluding the basis fields that cannot be present for the current input. For instance, if the parameter estimation of (2) reveals that the parameter a_1 is zero, all basis fields of the hierarchical basis with a non-zero value for a_1 can be excluded. With this subset selected from the hierarchical basis, another least-squares-fit (4) is performed, also having the input displacement field as a target. The resulting pre-factors are analyzed and further basis fields with minor contributions are excluded iteratively (5). This step is necessary to reduce the effects needed to explain the input displacement field to a minimum. The actual decomposition is calculated per node (6), the contribution of each basis field to the total measured displacement. Finally, the contribution plots (7) are generated.

3.1. Preparation Steps

⋮

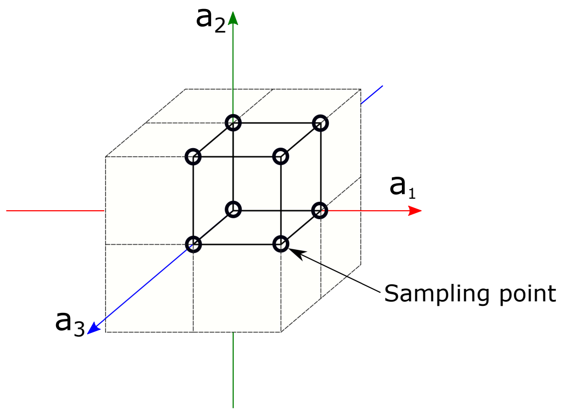


Figure 4: Locations of chosen sample points for basis fields. Points in parameter space are placed at positions defined by combining maximum, minimum and zero parameter values.(This figure has been published in Claus, Hagen, Leonhardt, Leitte and Hamann (2021b) licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/, changes: variable names)

⋮

3.1.2 Generating the Bases for Least-Square-Fitting

⋮

The second basis needed for the proposed method is generated by a hierarchical selection of important simulation results from the ensemble. The workflow of this selection is depicted in Figure 5. The motivation for a hierarchical selection is the goal to find simulations that can be used as bases for a linear least-squares-fit and are related to single effects at the same time. The objective of the hierarchical selection is to automatically discover the given ensemble of simulation results from simple to complex parameter states. The steps performed during this automatic discovery are explained next. First, an initial basis is chosen that consists only of simulation results that have parameter states, located on the axis of the parameter space (Rank 0 and Rank 1 - see Table 1). This initial basis is assumed to be almost orthogonal as only one parameter is varied per simulation. Second, the initial basis is used to approximate all simulation results of rank two via a least-squares-fit. The result of each approximation is compared to each actual simulation. Each comparison produces a “Difference Field." We select the difference field with the largest (absolute) difference. When the magnitude is larger than a tolerance ε a new effect must be considered for approximating the current rank within the defined tolerance. The initial basis is expanded by adding the difference field with the largest difference. This new effect can also be interpreted as a new axis in parameter space. As the sampling of the parameter space of the original data set is uniform (see

Figure 4), a second difference field with inverse parameter states exists, located on the same axis. This difference field with inverse parameter states is added to the basis. Two corresponding fields per effect are present, covering both axis directions. The added difference fields are considered essential for capturing the new effect that cannot be explained by the previous basis. This second step is repeated until all simulation results from Rank two are approximated within the tolerance ε. Third, all displacement fields from rank three are approximated with the new basis, and the iterative process is repeated until all ranks are approximated within the defined tolerance. Choosing the value for ε is highly dependent on the required precision of an application.

3.2. Parameter Estimation and Contribution Plots

$\vdots$

Rank	$\vec{a}$	a_1	a_2	a_3	a_4	a_5	a_6	a_7	...
0	$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$							
1	$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$	$\begin{pmatrix} max \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ max \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ max \end{pmatrix}$	$\begin{pmatrix} min \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ min \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ min \end{pmatrix}$		
2	$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$	$\begin{pmatrix} max \\ max \\ 0 \end{pmatrix}$	$\begin{pmatrix} max \\ 0 \\ max \end{pmatrix}$	$\begin{pmatrix} 0 \\ max \\ max \end{pmatrix}$	$\begin{pmatrix} min \\ max \\ 0 \end{pmatrix}$	$\begin{pmatrix} max \\ min \\ 0 \end{pmatrix}$	$\begin{pmatrix} min \\ 0 \\ max \end{pmatrix}$	$\begin{pmatrix} max \\ 0 \\ min \end{pmatrix}$	...
3	$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$	$\begin{pmatrix} max \\ max \\ max \end{pmatrix}$	$\begin{pmatrix} max \\ max \\ min \end{pmatrix}$	$\begin{pmatrix} max \\ min \\ min \end{pmatrix}$	$\begin{pmatrix} min \\ min \\ max \end{pmatrix}$	$\begin{pmatrix} min \\ max \\ min \end{pmatrix}$	$\begin{pmatrix} min \\ min \\ min \end{pmatrix}$		

Table 1: Sampling scheme for parameter space. The classification of ranks is done by grouping samples based on the numbers of non-zero parameter values.

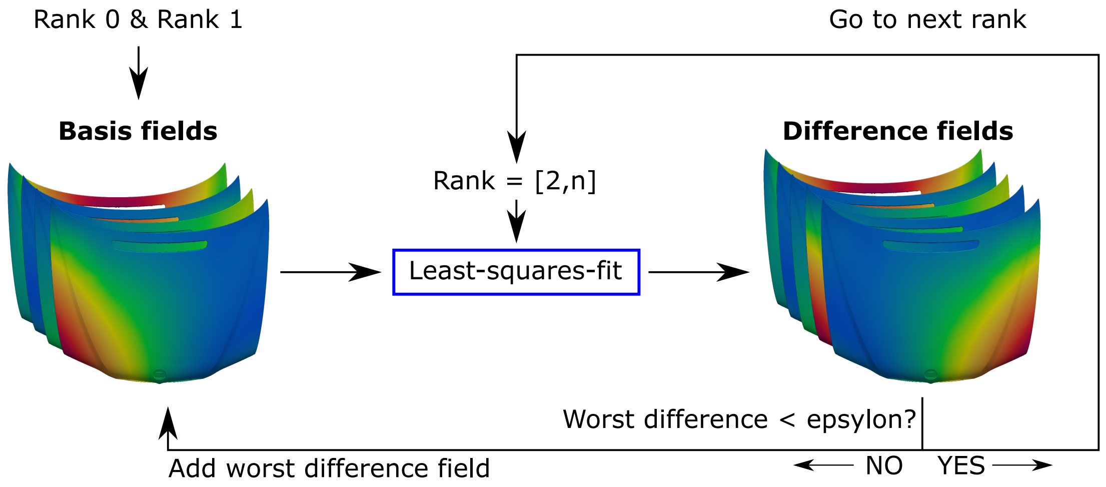


Figure 5: Hierarchical basis generation for least-squares-fit. The algorithm approximates every simulation rank-by-rank from the generated ensemble and adds an “effect" via a displacement field when approximation error is too large.

3.2.2 Contribution Plots

The basis fields generated via hierarchical selection, see Figure 5, are not ideal for approximating the solution space as none of the selected basis fields has the same properties as a PCA basis element. The hierarchical basis must contain more basis fields to represent the solution space at least within the specified tolerance. Although the selected displacement fields might not be an optimal basis for the approximation problem, they have a significant advantage:

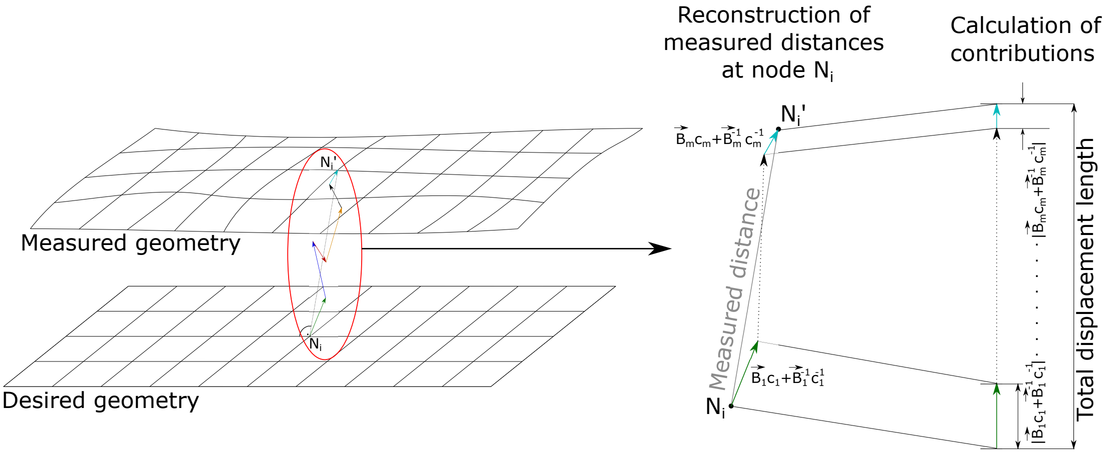

Measured geometry
Desired geometry
N_i'
N_i
Reconstruction of measured distances at node N_i
Calculation of contributions
$\vec{B}_m c_m + \vec{B}_m^{-1} c_m^{-1}$
$\vec{B}_1 c_1 + \vec{B}_1^{-1} c_1^{-1}$
Measured distance
$|\vec{B}_m c_m + \vec{B}_m^{-1} c_m^{-1}|$
$|\vec{B}_1 c_1 + \vec{B}_1^{-1} c_1^{-1}|$
Total displacement length

Figure 6: Left: Desired geometry and measured geometry represented by same mesh. Right: Reconstruction of measured distance for single node N_i and contribution for

every effect calculated from corresponding basis fields.

The chosen basis vector fields can be interpreted as contributors to a single effect (one parameter or a combination of parameters). The contribution per effect is calculated as follows: We use the hierarchical basis for a least-squares-fit with the measured displacement field as input. The obtained linear combination can be interpreted as a combination of single effects. To plot the contribution of one single effect to the total measured displacement additional calculations are necessary. Calculating the single contribution per effect is illustrated in Figure 6. The measured displacement for a node N_i is the distance between the desired geometry and the actual measurement. This distance is reconstructed by a combination of vectors from the basis field using the least-squares-fit. The used basis has two basis fields per effect, see Section 3.1.2. This property makes it necessary to add the corresponding terms of the linear combination to obtain the contribution of the underlying effect. The partial displacement field d_i is calculated by adding the terms $B_i c_i$ and $B_i^{-1} c_i^{-1}$. The partial displacement field is condensed to a scalar field by calculating the magnitude of the displacement vector for each node:

$$Partial\ displacement\ d_i\ of\ the\ effect\ e_i:\ d_i\ =\ |c_i\vec{B}_i + c_i^{-1}\vec{B}_i^{-1}|. \tag{6}$$

Here, B_i and B_i^{-1} are the basis fields that relate to a single effect, and c_i, c_i^{-1} are the pre-factors resulting from the least-squares-fit. Next, the total magnitudes of all partial displacements are computed for all nodes, by adding the partial contributions:

$$Total\ displacement\ D:\quad D\ =\ \sum d_i \quad (sum\ of\ all\ partial\ displacements\ of\ e_i). \tag{7}$$

Equations 6 and 7 are illustrated on the right-hand side of Figure 6. The calculated total displacement length is used for contribution normalization to the interval [0,1]:

$$Contribution\ of\ e_i:\ d_i/D\ (ratio\ of\ partial\ displacement\ of\ e_i\ and\ total\ displacement). \tag{8}$$

Once the contribution d_i/D has been calculated for every effect e_i, we can plot scalar fields in the 3D domain. These plots show the contributions for every effect to the measured displacement field on the scale of [0%,100%], providing detailed insight to an inspection engineer.

⋮

References

⋮

Aspert, N., Santa-Cruz, D., Ebrahimi, T., 2002. Mesh: measuring errors between surfaces using the hausdorff distance, in: Proceedings. IEEE International Conference on Multimedia and Expo, pp. 705–708 vol.1. doi:10.1109/ICME.2002.1035879.

⋮

Cloudcompare, 2020. URL: http://www.cloudcompare.org/.

⋮

Claus, F., Hagen, H., Hamann, B., 2021a. Calculating the gravity-free shape of sheetmetal parts. The International Journal of Advanced Manufacturing Technology 113,3401–3417. URL: https://doi.org/10.1007/s00170-021-06702-6, doi:10.1007/s00 170-021-06702-6.

Claus, F., Hagen, H., Leonhardt, V., Leitte, H., Hamann, B., 2021b. Interactive Quality Inspection of Measured Deviations in Sheet Metal Assemblies, in: Garth, C., Aurich, J.C., Linke, B., Müller, R., Ravani, B., Weber, G.H., Kirsch, B. (Eds.), 2nd International Conference of the DFG International Research Training Group 2057 – Physical Modeling for Virtual Manufacturing (iPMVM 2020), Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany. pp. 6:1–6:18. URL: https://drops.dagstuhl.de/opus/volltexte/2021/13755, doi:10.4230/OASIcs.iPMVM. 2020.6.

⋮

Kenta-Tanaka et al., . probreg. URL: https://probreg.readthedocs.io/en/latest/.

5.3.3 Prediction of Displacements

This section describes the setup of the prediction kernel, compare Figure 5.1. The prediction kernel uses a certain parameter state as input and predicts the corresponding displacement field.

Interpolation in High Dimensional Spaces

In [4] a high dimensional interpolation scheme was used to make these predictions in real time. This approach is excerpted in the following:

Quotation 5.2: Chapter 3.2.2 excerpted from [4][1]

3.2.2 Interpolation / Approximation Scheme

For generating information between sampled data points, we use interpolation. The interpolation scheme chosen in this paper for interpolating in a higher-dimensional space is called "inverse distance weighting". This scheme will now be explained in detail:

Let $\vec{a} = (a_1, a_2, ..., a_n)$ be a vector of parameters of one simulation and $\vec{u}(\vec{a})$ the corresponding simulation result represented by a displacement field, then we can express the pre-simulated displacement fields S as a set of simulation runs based on the parameter sets A:

$$A = \{\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_m\} \tag{3}$$

$$S(A) = \{\vec{u}(\vec{a}_1), \vec{u}(\vec{a}_2), \ldots, \vec{u}(\vec{a}_m)\} = \{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_m\} \tag{4}$$

Every parameter set contained in the quantity from equation 3 is considered as a known or sampled point in the parameter space. The corresponding displacement field can be interpreted as the "value" of this point.
Let $\vec{b} \in X$ be also a set of simulation parameters but $\vec{u}(\vec{b}) = \vec{u}_b \notin S(A)$, thus we consider $\vec{u}_b$ as an unknown value in the solution space $S(A)$ that needs to be interpolated. To do so, every known value is weighted by the inverse of the distance using the function $\lambda : X \times X \to \mathbb{R}$, calculated in the parameter space:

$$\vec{u}_b \approx \frac{1}{\sum_{l=1}^{m} \lambda(\vec{b}, \vec{a}_l)} \sum_{i=1}^{m} \lambda(\vec{b}, \vec{a}_i) \vec{u}(\vec{a}_i) \tag{5}$$

$$\lambda(\vec{b}, \vec{a}_i) = \frac{1}{D(\vec{b}, \vec{a}_i)} \tag{6}$$

$$D(\vec{b}, \vec{a}_i) = ||\vec{b} - \vec{a}_i||_2 \tag{7}$$

[1] (This content has been published in [4], licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/; changes: none)

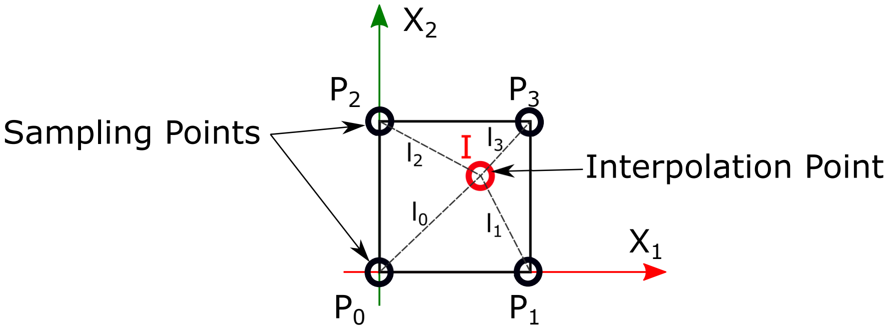


Figure 4: Calculation of the distances in the parameter space for an interpolation point (red) inside a sampled cell.

Equation 6 is a weighting factor by inverting the distance calculated between the point $\vec{b}$ and the sampled point $\vec{a}_i$, while

$$\frac{1}{\sum_{l=1}^{m} \lambda(\vec{b},\vec{a}_l)} \tag{8}$$

is a factor to scale the sum of weights to 1 so that we can introduce the scaled weight $\lambda^*(\vec{b},\vec{a}_i)$:

$$\lambda^*(\vec{b},\vec{a}_i) = \frac{\lambda(\vec{b},\vec{a}_i)}{\sum_{l=1}^{m} \lambda(\vec{b},\vec{a}_l)} \tag{9}$$

To locally increase the weight of a sampled point, the inverse lengths are raised to the power of k:

$$\lambda^{*k}(\vec{b},\vec{a}_i) = \frac{(\lambda(\vec{b},\vec{a}_i))^k}{\sum_{l=1}^{m}(\lambda(\vec{b},\vec{a}_l))^k}, where \sum_{i=1}^{m} \lambda^{*k}(\vec{b},\vec{a}_i) = 1 \tag{10}$$

With the introduced scaled weights and the notation $\lambda_i^{*k} = \lambda^{*k}(\vec{b},\vec{a}_i)$ we can express the interpolation as:

$$\vec{u}_b \approx \sum_{i=1}^{m} \lambda_i^{*k}(\vec{b},\vec{a}_i)\vec{u}(\vec{a}_i) = \lambda_1^{*k}\vec{u}(\vec{a}_1) + \lambda_2^{*k}\vec{u}(\vec{a}_2) + \ldots + \lambda_m^{*k}\vec{u}(\vec{a}_m) \tag{11}$$

Figrue 4 demonstrates a simple 2D example the interpolation scheme. The red point I can be approximated by calculating the distances l_{0-3} and applying equation 11.

Weak Spots of Inverse Distance Weighting

When using inverse distance weighting, some aspects that are affecting the interpolation results are discussed in the following.

First, the distribution of the sampled points in the parameter space plays a significant role in this interpolation scheme. In Figure 5 an example is given for an uneven distribution of sampling points. The red point again is the point that needs to be interpolated. When performing inverse distance weighting on this kind of data set the interpolation result will be shifted towards the dense cluster of sampling points. Thus, the result of the interpolation would not be plausible anymore. Another factor that influences the interpolation result is the threshold radius when filtering the distances so that the interpolation does not have to calculate weights and the linear combination of Equation 11 for the whole data set. Depending on the location of the interpolation point a slight change of the filter radius can result again in an even or uneven set of sampling points that are used for the interpolation. To prevent these shortcomings, the data set of pre-simulated displacement fields is distributed in an even manner like presented in Section 3.2.1. To prevent uneven distribution due to distance filtering, our method uses just the nodes of the hypercube where the interpolation node is located in. These sampling points are found by sorting the whole data set by distance and use the n closest points, where n is equal to the number of 0-faces of the corresponding hypercube, see Table 1. Although there are more general approaches available in the branch of multivariate interpolation commonly used like kriging, spline interpolation, or radial basis function interpolation, inverse distance weighting is preferred in this paper. Different methods for scattered data interpolation are discussed and tested in [10]. The reason for choosing inverse distance weighting for solving the interpolation problem in the higher dimensional parameter space is the simplicity, robustness, and easy to implement nature of the algorithm. Also, the mentioned shortcomings of this algorithm can be prevented as discussed.

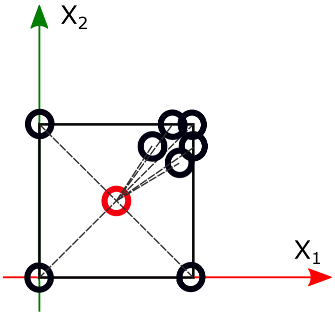


Figure 5: Influence of the distribution of sampling points. Local clusters lead to a higher weighting towards the cluster position.

⋮

References

⋮

10 Richard Franke. Scattered data interpolation: Tests of some method Mathematics of Computation, 38(157):181–200, 1982. URL: http://www.jstor.org/stable/2007474.

Least Squares Fitting

Besides a high-dimensional interpolation approach, also an LSQF can be utilized to predict displacement fields based on given parameter states. Like before, a PCA-generated basis is fitted to target values. The only difference is, that the fit is performed in the parameter space and the found linear factors are used to reconstruct a displacement field in the solution space. This approach has to be treated with caution because LSQF is only applicable for special cases. The reason why performing an LSQF on the parameter space does not work in general is this: the solution is not unique if the number of PCs needed to represent the solution space is higher than the number of parameter axis that span the solution space. As described before, 6 adjustable parameters are present for the engine hood but 10 PCs are selected from the dimensionality reduction of the solution space. This means that LSQF can not be applied to find a unique solution. However, testing has revealed that the LSQF-approach does converge to the preferred solution for the present data set. The whole verification data set was approximated by using LSQF in the parameter space with the same basis generated before via PCA. The maximum error at single mesh nodes was below ± 0.1 mm. This error is astonishingly low considering the simplicity of the approximation scheme. Also, computational times are very low making the LSQF approach attractive. So, the following conceptional troubleshooting tool uses LSQF together with the PCA-generated basis as a prediction kernel. To ensure general applicability, LSQF is not advised for this purpose. Therefore, high dimensional interpolation or neural ML approaches are advised.

5.4 Verification

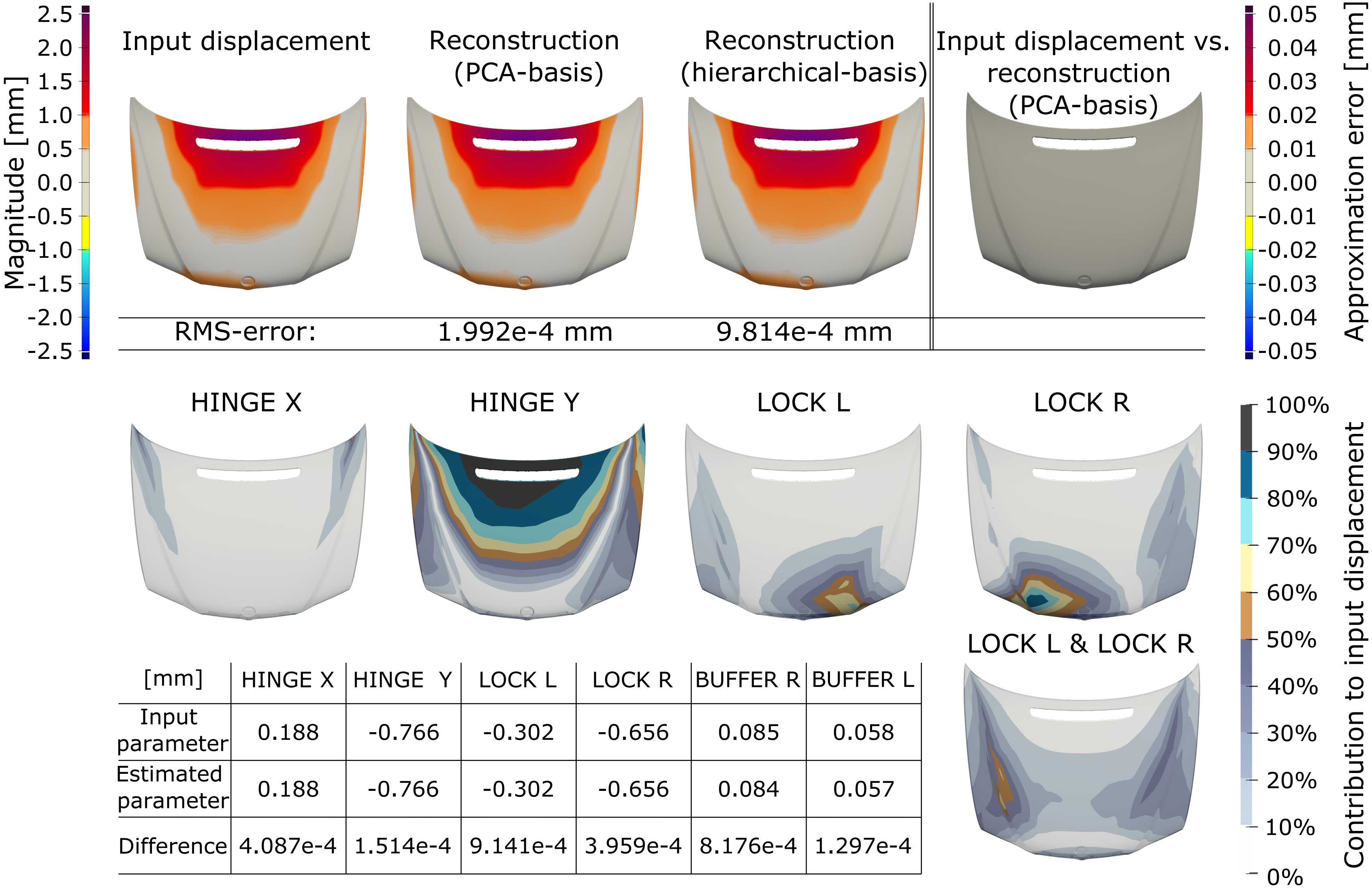
In this section, the approaches that were selected in the previous section are verified. The verification splits up into: (a) verification of the estimation kernel and (b) verification of the prediction kernel.

5.4.1 Verification of the Estimation Kernel

The verification of the estimation kernel is carried out by selecting a random ensemble member of the verification data set and performing the parameter estimation via LSQF together with the decomposition for creating contribution plots. The used simulation parameters are considered as hidden parameter states or ground truth. The estimation results are compared to these hidden parameter states and an estimation error can be calculated. The quality of contribution plots can not be measured directly. To quantify the quality of the plots, the reconstruction errors between hierarchical basis and PCA generated basis are compared.

Figure 5.8 shows the results of this verification. On the top left, the randomly selected displacement field from the verification ensemble is depicted. This displacement field is considered as pseudo measurement input. Based on the input displacement two reconstructions - one with PCA basis and one with the hierarchical approach - are performed. The reconstructions are shown next to the input displacement field. To measure the quality of reconstruction, RMS-error is evaluated for both results. A direct comparison between PCA-based LSQF and input displacement field is shown on the upper right - no observable differences are present. The parameter estimation results are listed in the table inside the figure. All 6 parameters were estimated with errors significant below ± 0.05 mm. The largest estimation error has the parameter “LOCK L” with 9.1e-4 mm. Regarding the contribution plots, 6 major influences were identified that are used to explain the input displacement field. Besides the two locks, and two directions of the hinge, a basis field that is related to the combination of left and right locks were used.

To ensure that the estimation kernel works for the whole solution space, all 1400 ensemble members were used for estimation. The results for the parameter estimations were already shown in Figure 5.7. Regarding the reconstruction of the displacement field using the PCA-basis and hierarchical approach, also the whole verification ensemble was tested with both reconstruction techniques. A distribution of RMS-error values across the verification ensemble is shown in Figure 5.9. It can be observed that RMS values for the reconstruction with the hierarchical approach are higher (max 0.08 mm) compared to the reconstructions with the PCA basis.



[mm]	HINGE X	HINGE Y	LOCK L	LOCK R	BUFFER R	BUFFER L
Input parameter	0.188	-0.766	-0.302	-0.656	0.085	0.058
Estimated parameter	0.188	-0.766	-0.302	-0.656	0.084	0.057
Difference	4.087e-4	1.514e-4	9.141e-4	3.959e-4	8.176e-4	1.297e-4

Fig. 5.8 Verification results. Top: input displacement field and reconstruction results via LSQF with both bases. Top right: reconstruction error of PCA-based LSQF. Bottom: resulting contribution plots for each relevant parameter (or combination). Table: parameter erstimation results compared to hidden parameter states of simulated input. This figure has been published in [5], licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/; changes: color maps)

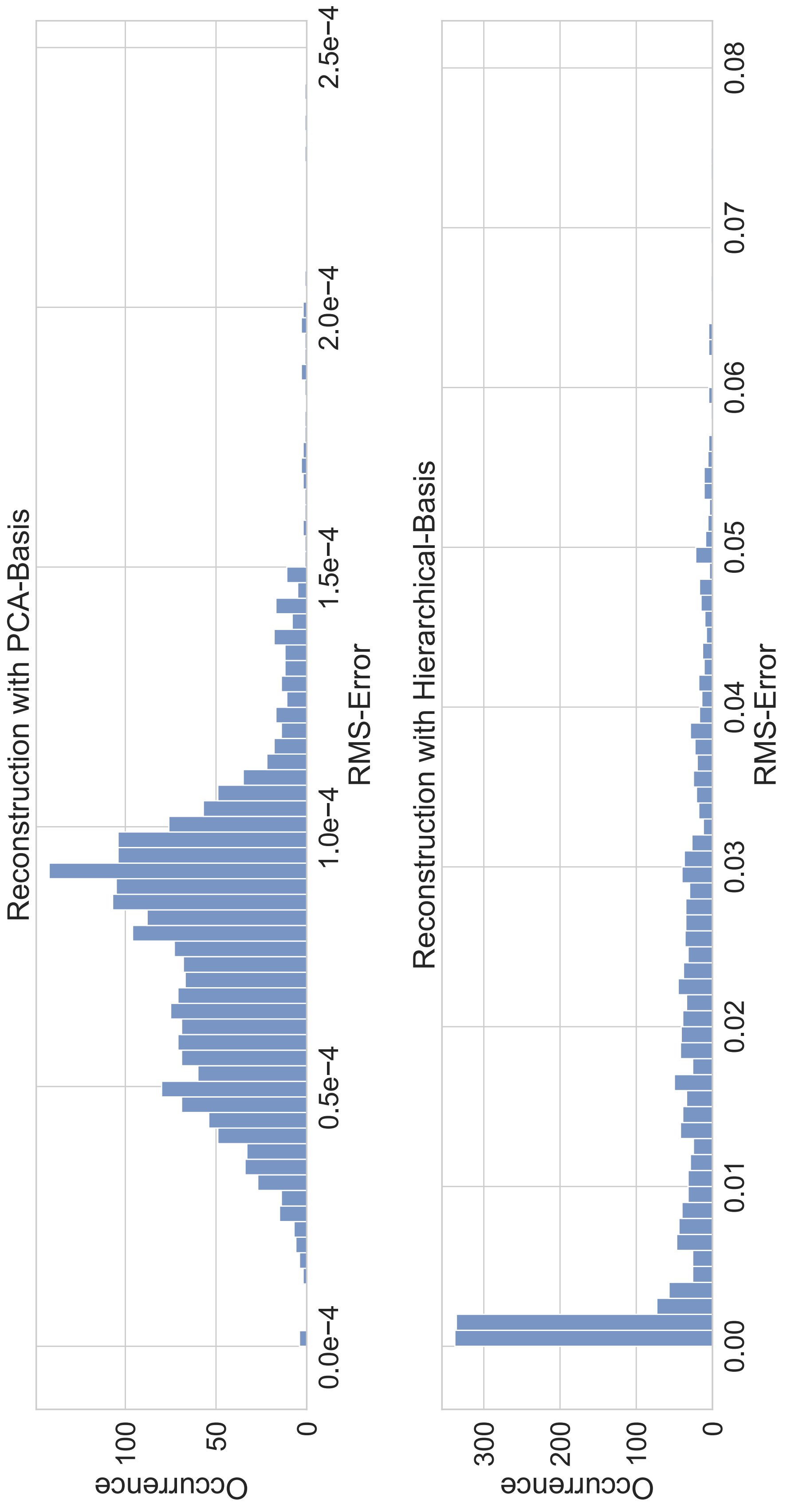


Fig. 5.9 Histogram of reconstruction quality across verification ensemble with PCA basis (top) and hierarchical approach (bottom). Reconstruction error is measured with the RMS metric.

5.4.2 Verification of the Prediction Kernel

To verify the performance of the prediction kernel, all 1400 simulation results from the verification ensemble are used. Only the parameter values are given to the prediction kernel. Based on each set of parameters, a displacement field is predicted. Then, each prediction result is compared with the actual simulation result and RMS-error is calculated. The results are summarized in Figure 5.10. The maximum RMS error for the whole data set is below 0.0145 mm.

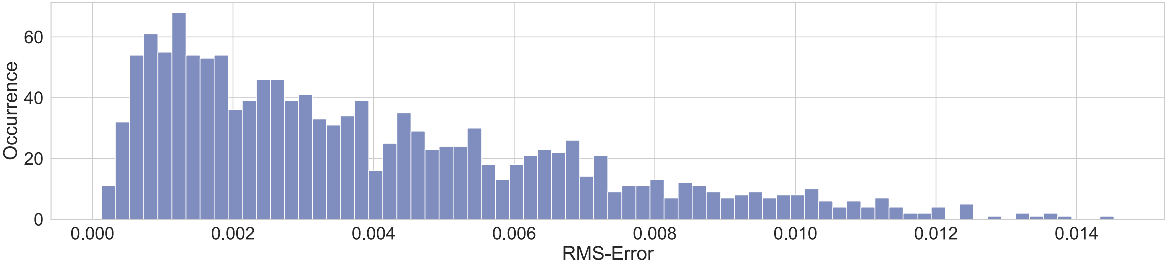
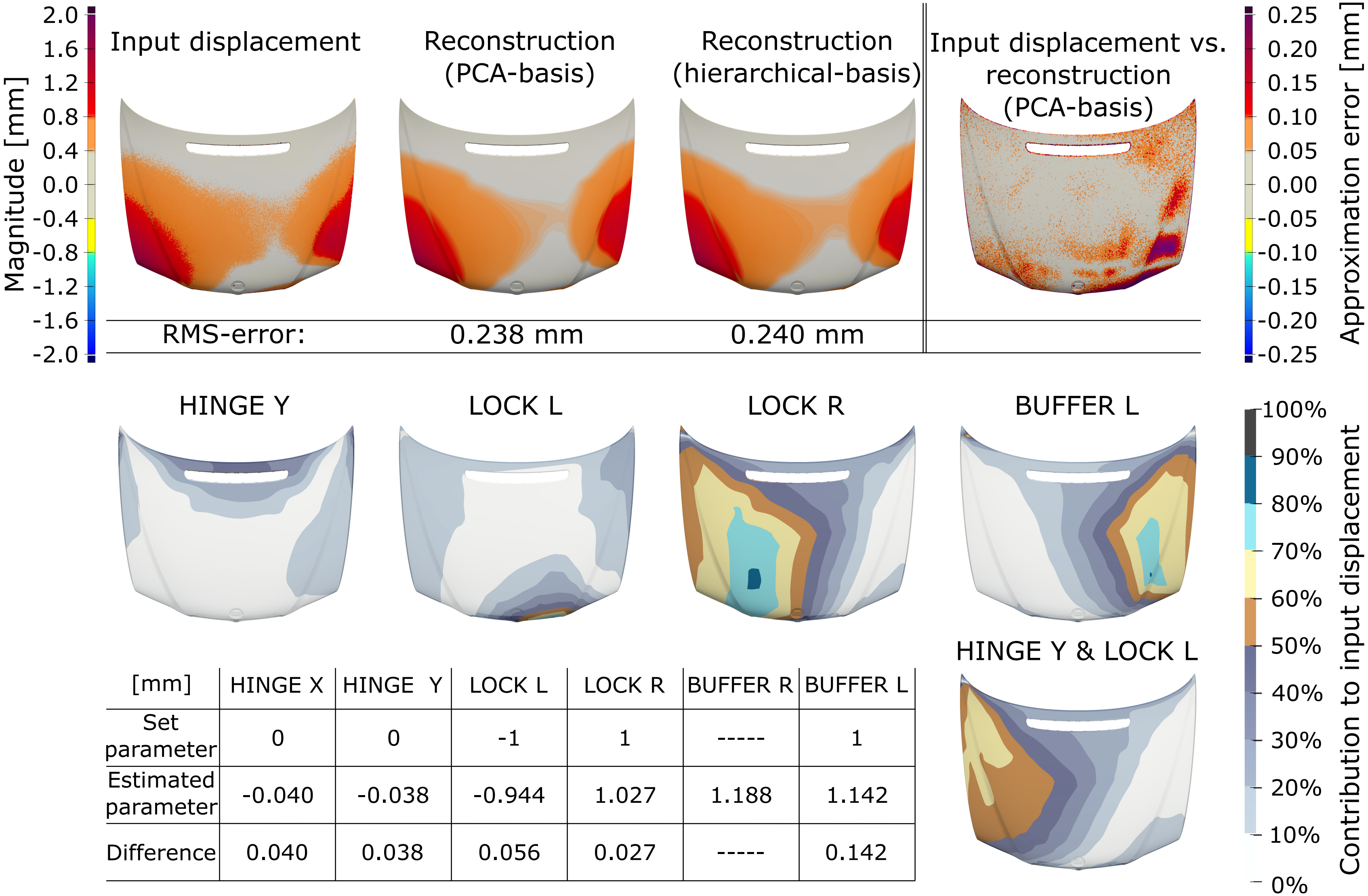


Fig. 5.10 Verification of the prediction kernel. All displacement fields of the verification ensemble were predicted based on the parameter states. The prediction error is measured with RMS metric.

5.5 Validation

Although the results of the verification with simulated data are very promising, validation experiments are carried out to ensure applicability to real-world problems. For the experiments, the same hardware setup is used that was introduced in Section 4.5. The validation is designed analogous to the verification presented in Figure 5.8. This time, the input displacement field is based on measurement data. To generate a displacement field, two measured geometries are compared. The first measurement is performed to capture the geometry of the engine hood before changing interface position. This 3D-scan is considered as a baseline. Next, interfaces are changed according to the values listed in the table of Figure 5.11. To ensure a precise change of position, multiple precision gauges are used. This is necessary to compensate for unwanted runaway of fixture positions. This runaway is caused by the lack of rigidity of the fixture hardware. After all fixture positions are dialed in, a second 3D-scan is performed. Due to the measurement layout, both scans are performed in the same reference frame such no alignment between the acquired point clouds is needed. However, the scans must be aligned once to match with the reference frame of the simulation mesh. When calculating the distance between both scans, a scalar displacement field is obtained. As measured distances are only in surface normal direction, the scalar field can be converted to a displacement field by using the surface normal as the direction of the vector. The resulting displacement vector field serves as input for the validation and is shown at the top left of Figure 5.11. Based on the differences in geometry, the estimation kernel is used to estimate the related parameter states. The results are shown in the table at the bottom of the figure. The highest estimation errors can be observed at the left buffer with 0.142 mm. Next to the input displacement field, two reconstructions with PCA generated- and hierarchical-basis are shown. Both reconstructions show a higher RMS value compared with the verification results. A direct comparison between PCA-based reconstruction and input displacement field is shown at the top right of the figure. On the scale of ± 0.25 mm significant differences can be observed at single spots on the geometry. The lower half shows the contribution plots that are derived from reconstruction with the hierarchical basis.



[mm]	HINGE X	HINGE Y	LOCK L	LOCK R	BUFFER R	BUFFER L
Set parameter	0	0	-1	1	-----	1
Estimated parameter	-0.040	-0.038	-0.944	1.027	1.188	1.142
Difference	0.040	0.038	0.056	0.027	-----	0.142

Fig. 5.11 Validation with real measurement data. This figure is set up analog to Figure 5.8. Note, the parameter “BUFFER R” was not measured during the experiment. This figure has been published in [5], licensed under Attribution 4.0 International (CC BY 4.0), see https://creativecommons.org/licenses/by/4.0/; changes: color maps)

5.6 Run Times

All relevant computing times were measured and listed in Table 5.4. The run times are divided into “preparation times” and “execution times”. The preparation times summarize all computational times that have to be performed once in advance. In contrast, the execution times consider the run times of all processes that must be performed when working interactively with the troubleshooting tool. The preparation times include all simulation times for generating the necessary ensembles. In the table the total computing time is listed, parallel computing was not considered for these measurements. The generation of the bases is also a one-time preparation step, as the computed bases can be used for any measurement of a related part. The execution times are composed of the run times for the estimation kernel and prediction kernel. These steps are time-critical as they are performed during user interaction.

Table 5.4 Computational run times. “Preparation times" and “Execution times" are listed.

Task	Time per Simulation	# of Simulations	Total Time
	Preparation Times		
Generating Training Ensemble	187s	793	41.19h
Generating Verification Ensemble	187s	1400	72.72h
PCA-Basis Generation	–	–	41s
Hierarchical-Basis Generation	–	–	3:21h
	Execution Times		
Estimate Parameter States	–	–	1.84s
Generate Contribution Plots	–	–	19.77s
Predict Displacement Field	–	–	0.0337s

5.7 Explorative Trouble Shooting

The presented methods are implemented and combined into a single software tool. This section presents a mock-up of a possible user interface. The software intends to support the inspection engineer during troubleshooting of measured geometrical errors. The tool is designed to be interactive and explorative. This means the engineer can try different countermeasures and the software provides predictions of the effects in real time.

Figure 5.12 shows a mockup of the user interface. The interface is a multiple coordinated visual exploration approach. The view is composed of 5 components:

1. Render View

2. Histogramm

3. Data Loading

4. Contribution Plots

5. Tune Parameters

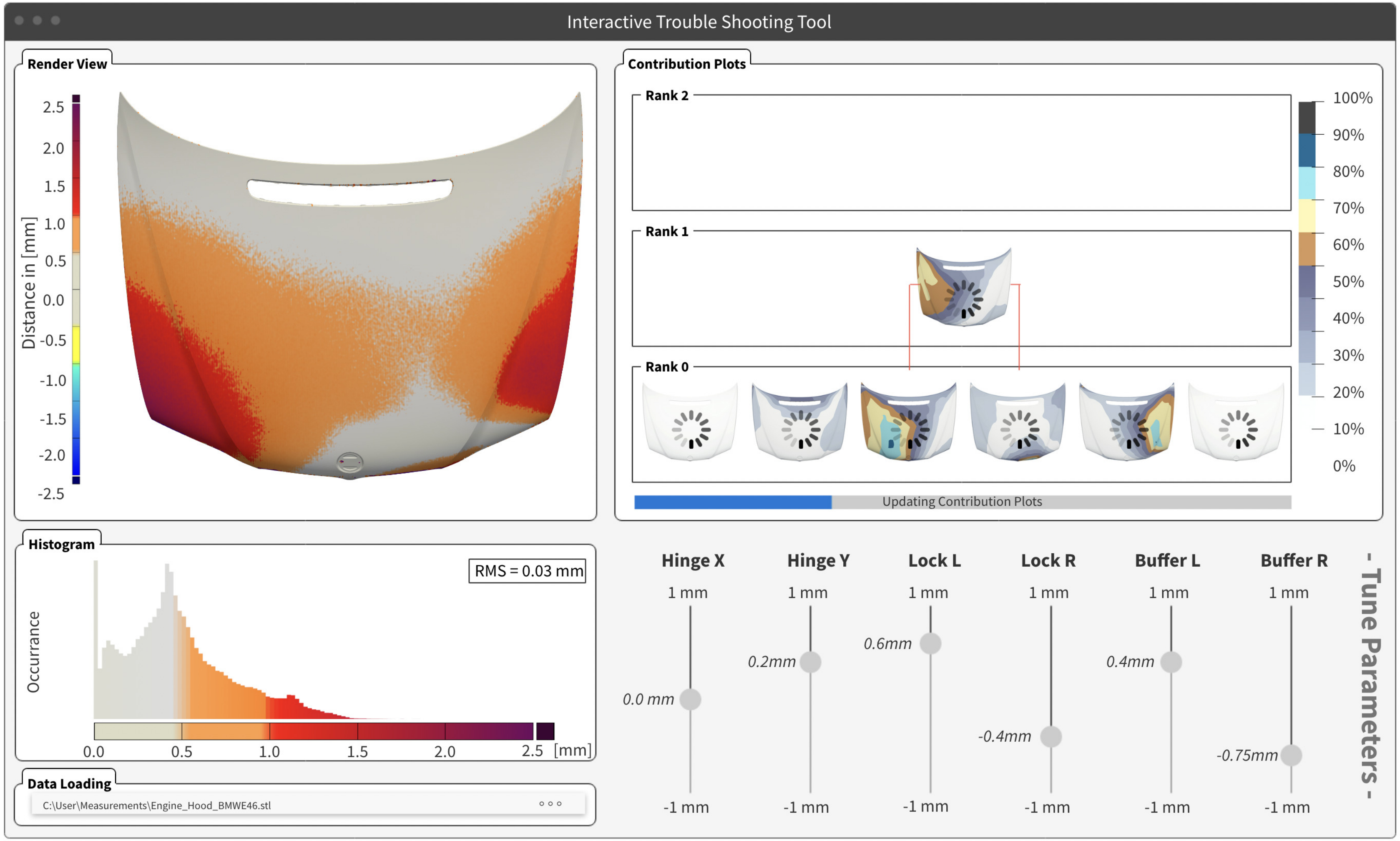
Render View - The render view shows initially the measured shape errors compared with the optimal assembled shape predicted by the smart assembly system. This view is updated live each time the parameters from the "tune parameters" panel are adjusted. This view could be realized as interactive, such that the user can rotate and zoom the inspection object.

Histogram - The histogram shows the distribution of occurring errors. Also, RMS-error over all visualized values from the render view is provided at the top of the graph. This diagram is linked to the render view and is also being updated in real time if parameter values are changed.

Data Loading - This field can be used to browse for the data that must be inspected.

Contribution Plots - This panel shows the hierarchical decomposition of the shape errors that are currently shown in the render view. The contribution plots are clustered by rank. Each plot of rank 0 is related to one parameter. For higher ranks, relations to parameters are drawn with red lines. This view can not be updated in real time because of run times up to 20s. Therefore, a loading bar with the current status of the background computation is provided. Also, each contribution plot has and loading indicator if the plot is not up to date.

Tune Parameters - This panel provides one slider per parameter where the user can try out different changes of interface positions. The initial value of each slider is set to the estimation output based on the measurement data. Any movement of any slider executes the computing pipeline described in Figure 5.1. For more parameters, this view as well as the contribution plot panel must be extended to the right.



RMS = 0.03 mm

C:\User\Measurements\Engine_Hood_BMWE46.stl

Hinge X 0.0 mm

Hinge Y 0.2mm

Lock L 0.6mm

Lock R -0.4mm

Buffer L 0.4mm

Buffer R -0.75mm

Fig. 5.12 GUI mock up for the interactive troubleshooting tool. The interface is designed as multiple coordinated visual exploration approach and composes of the panels: (a) render view, (b) histogram, (c) data loading, (d) contribution plots, and (e) tune parameters.

5.8 Discussion

This section discusses various aspects of the interactive troubleshooting tool. Following topics are covered in this discussion: (a) generation of training data, (b) verification results, (c) data acquisition, (d) validation results, (e) computational times, (c) user interface, and (d) general discussion in the context of smart assembly systems.

5.8.1 Generation of Training Data

The generation of the training data set is the most costly step in terms of computing resources and time. In the presented example of the engine hood, the parameter space has 6 dimensions which were sampled with a full factorial approach with 3 sampling points per parameter axis. Additional 64 sampling points were added to the center of each hypercube of the parameter space. The verification data set was sampled equally distributed with 1400 sampling points. A total of 2193 simulations was performed to generate both ensembles. As mentioned earlier, this number growth exponential and using such a sampling strategy might result quickly in a number of simulation runs that cannot be handled anymore. To overcome such limitations, adaptive sampling strategies could be employed that drastically lower the number of ensemble members while maintaining a good representation of the solution space. Also, the verification ensemble can be reduced to a smaller number of samples if a well-known, trustworthy prediction model can be established based on similar parts.

Regarding the contribution plots, the sampling strategy can not be changed drastically as full factorial sampling is required. However, not relevant parts of the solution space could be excluded. Unfortunately, identifying non-relevant parts of the solution space requires an experienced engineer and a well-understood part behavior.

5.8.2 Verification Results

The verification results shown in Figure 5.8 demonstrates the performance of the estimation kernel under ideal conditions. The estimation kernel composes of the parameter estimation method and the hierarchical decomposition. While the quality of parameter estimation can be measured directly by comparing the hidden parameter states with the prediction results, the quality of the contribution plots can only be measured indirectly. As a quality measure, the displacement field from the LSQF performed with the hierarchical basis is compared with the target (input) displacement field. For the shown verification RMS-error value of $9.81\text{e}{-4}$ mm was achieved which on the same magnitude as the PCA-based reconstruction.

Due to this high-quality reconstruction, also the contribution plots are trustworthy.

The verification results of the parameter estimation were already shown earlier in Figure 5.7. For this box plot, all parameters for all simulations of the whole verification ensemble were estimated. As the maximum error is below ± 0.01 mm, the parameter estimation does perform very well under ideal conditions and lies within the goal of ± 0.05 mm.

5.8.3 Data acquisition

To eliminate uncertainties, the acquisition of the input displacement field was carried out differently compared to the real production process. For the validation, two measurements are compared - one baseline measurement where the part is mounted arbitrarily to the fixture and a second measurement after changing interface positions in a controlled way. In a real production environment, the input displacement field is obtained by a direct comparison between the predicted ideal assembly situation and a post-assembly measurement of the actual geometry.

The quality and density of the acquired point cloud are sufficient for validation purposes. For the application in real-world production systems, even better resolution and coverage of the measurement data can be expected if industrial-grade hardware is employed.

5.8.4 Validation Results

The purpose of the validation is to investigate the performance of the presented method with real measurement data. The results are shown in Figure 5.11. When comparing the results with the analog verification results, higher errors can be observed. This is intuitive as real experimental data is subjected to known and unknown uncertainties, i.e. fixture variation, measurement uncertainty. When comparing the reconstruction via PCA basis with the measured displacement field, single spots on the geometry show significant errors up to ± 0.25 mm. As the PCA basis was proven to cover the whole solution space with great precision, differences in this comparison indicate that the behavior of the simulation model and actual part differs. Although the observed differences are relevant in terms of mathematical precision, they cannot be avoided in real circumstances. Even slight differences like geometrical deviations of substructure, application of adhesives, or residual stresses in the material can cause such errors. Therefore, the margin of ± 0.25 mm is still acceptable. It can be argued that errors at single spots do not affect the correct reconstruction of the underlying shape errors. The shape errors that are of interest are caused by misplaced interfaces and affect large areas of the part.

Summarized, the validation results are satisfying. Although the errors are way higher compared to the experiments with simulated data, the method does behave as expected and provides plausible results. Also, the noise of the scan data does not affect the performance of the approach.

5.8.5 Run Times

As the final software tool is intended to be interactive, computational steps need to be differentiated by time-critical- and not time-critical steps. In general, all time-critical steps are computations that are performed during the user interaction with the software. In Table 5.4 the critical run times are listed as “Execution Times”, while non critical run times are named “Preparation Times”. The preparation times take up most of the computational effort and contain the generation of the simulation ensembles as well as the generation of PCA-basis and hierarchical basis. These computing steps must be performed in advance before the interactive troubleshooting tool can be used. The listed times are accumulated and can be reduced drastically by parallelization. The only step that can not be accelerated by parallel computing is the generation of the hierarchical basis.

The execution times are composed of the run times of the estimation- and prediction -kernel. These times are critical for the user experience. Most critical is the prediction of displacement fields because this operation is performed every time the user does change parameter values. This computing step is very fast with an average of 0.0337 s, which enables responsive behavior. The contribution plots are also intended to be updated in real time, but with an average of 19.77 s run time this is not possible. This drawback is only minor as the user should tune in the parameter values by using the render view. The contribution plots only provide additional insights for troubleshooting and do not need to be updated in real time.

5.8.6 User Interface

The shown user interface in Figure 5.12 is only a first mockup of a possible GUI[1]. This mockup is intended to provide an impression of how such an interactive experience could look like. To realize this software tool, design studies, expert surveys, and design guidelines must be taken into account.

[1] Graphical User Interface

5.8.7 Discussion in the Context of Smart Assembly Systems

In the context of smart assembly systems, the presented software tool can be used to identify positioning errors and derive countermeasures quickly. As this troubleshooting approach only indicates errors in the action system and the DT, the recognition system can not be improved based on the outcome of this method.

The tool is designed to be interactive such that the inspection engineer can gain insights into complex behaviors and gain a better understanding of the whole assembly system. By this, complex relations are understood better and downtime can be reduced. One great benefit of the system is this: no fundamental knowledge about the behavior and interaction of the parts with the assembly partner is presumed. In high flexible production systems, this knowledge can not be acquired in a reasonable amount of time and testing. Regarding automotive components, legal requirements and cost-efficiency lead to the use of thinner sheet metals. By that, parts become weaker, and deformation behavior more complex. Considering these aspects, the need for such a troubleshooting system is constantly rising.

The interactive human-in-the-loop approach might be obsolete if the production system is set up as a closed-loop system. Nevertheless, to achieve such a self-compensating assembly line, the system needs to build up trust before it is employed. The interactive troubleshooting tool can be used for ramp-up but also for recurring errors during the production phase that can not yet be compensated automatically.

Core References of this Chapter

[4] Felix Claus, Hans Hagen, Viktor Leonhardt, Heike Leitte, and Bernd Hamann. Interactive Quality Inspection of Measured Deviations in Sheet Metal Assemblies. In *2nd International Conference of the DFG International Research Training Group 2057 – Physical Modeling for Virtual Manufacturing (iPMVM 2020)*, volume 89 of *Open Access Series in Informatics (OASIcs)*, pages 6:1–6:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik

[5] Felix Claus, Bernd Hamann, Heike Leitte, and Hans Hagen. Decomposing deviations of scanned surfaces of sheet metal assemblies. *Journal of Manufacturing Systems*, 61:125–138, 2021

[7] Viktor Leonhardt, Felix Claus, and Christoph Garth. Pen: Process estimator neural network for root cause analysis using graph convolution. *Journal of Manufacturing Systems*, 2021

[8] Patrick Ruediger, Felix Claus, Bernd Hamann, Hans Hagen, and Heike Leitte. Combining visual analytics and machine learning for reverse engineering in assembly quality control. *Electronic Imaging*, 2021(1):60405–1–60405–13, 2021

[9] Patrick Ruediger, Felix Claus, Viktor Leonhardt, Hans Hagen, Jan C. Aurich, and Christoph Garth. Previs – a combined machine learning and visual interpolation approach for interactive reverse engineering in assembly quality control, 2022

Chapter 6

Discussion of the Digital Twin based Smart Assembly System

In this dissertation, a DT-based smart assembly system was presented. Not all challenges could be tackled within these studies to realize and apply such a system to a real production environment. However, major sub-problems were tackled and significant contributions were made. To realize the concept of DTs for smart assembly, problems that require competencies from different fields must be tacked. For instance (not exhaustive at all), these fields are: production planning, metrology, FE-simulation, surrogate modeling, robotics, data management, machine learning, geometric modeling, etc. To make this chapter more comprehensive, the outcome of this dissertation is discussed from the perspectives of different fields of research. In particular, these fields are:

- Manufacturing
- Finite Element Simulation
- Metrology
- Geometric Modeling
- Automotive / Aerospace Industry

For each field of research, a dedicated section will follow, discussing contributions and aspects regarding this dissertation. The reader is advised to one discussion of interest. If there is no preference, the discussion from the perspective of manufacturing is advised. The discussions from different perspectives are followed by general conclusions in Chapter 7.

6.1 Discussion from the Perspective of Manufacturing

The concept of a DT-based smart assembly system impacts a manufacturing system on different levels. This section discusses the outcome of this dissertation regarding manufacturing clustered by three levels: (a) factory level, (b) machine level, and (c) process level.

6.1.1 Factory Level

The factory level does cover all high-level aspects of a manufacturing system i.e. material flow, factory layout planning, or sustainability aspects, see [172]. On the factory level, this dissertation has no major contributions. However, the concept of DT-based smart assembly does enable a more flexible production where set up time can be drastically reduced and so start of production achieved earlier. Also, implementing the presented concept does have great dependencies on the factory level. Some of these aspects were already mentioned in Section 2.4.5. From the point of view of manufacturing, the mentioned challenges can be addressed without further research. Although the organizational aspects are not trivial in terms of data management, infrastructure, sequence of execution, supplier management, etc., all occurring challenges are doable with the current state-of-the-art technology.

6.1.2 Machine Level

The machine-level covers the hardware equipment and control software of the related processes. In the case of smart assembly, most of the system components of recognition-, prediction-, and action-system are belonging to the machine level. This level is identified to contain critical aspects that impact the applicability of the closed-loop smart assembly approach. Especially, recognition- and action-system need advanced and costly hardware to ensure flexibility and timings that are usable for real production processes. This kind of hardware i.e. 3D-scanning system for in-line usage or flexible tooling is already available or can be realized as a custom solution. Although, this dissertation does not provide any relevant contributions to improve applicability on the machine level, realizing a DT-based smart assembly system is possible in theory with state-of-the-art hardware.

6.1.3 Process Level

The process level contains all aspects regarding the underlying method or principle of a single process. The contributions of this dissertation are mostly situated at the process level. However, the core contributions are not belonging directly to the field of manufacturing.

This section only mentions the major achievements for each chapter and highlights new possibilities regarding manufacturing processes.

Chapter 2: Framework for Predicting Best Fit Assembly Parameters - This chapter, provides a first approach for a surrogate model for a DT-based smart assembly system. From the approach being presented in Method 2.1, requirements for each subsystem of smart assembly were derived. In this chapter, the conceptual foundations are provided that enable the development of the DT-based smart assembly. A general concept for sheet metal assemblies was already introduced by [86]. The contribution of Chapter 2 extends this idea to exterior sheet metal components which have special requirements in terms of gap- and flushness quality.

Chapter 3: Gravity Compensation of Scanned Non-Rigid Parts - This chapter has an exemplary contribution on the process level. The main contribution is Method 3.1 which can correctly compensate for deflections caused by gravity that occur during measurement. This method enables the correct post-processing of scanned non-rigid parts. By that, an improved DT can be derived and precise predictions become possible. This step is crucial for the whole concept. With this approach, the process of measuring assembly components can be simplified by using non-over-constrained fixtures. At the same time, precision is improved and process time is reduced.

Chapter 4: Automated FE-Model Generation from 3D-Scan Data - Chapter 4 does also contribute on the process level. It provides an automated mesh morphing approach that is designed to be employed as a process step in the prediction system. The presented Method 4.1 is intended to generate high-fidelity DTs based on measurement data. This important process step was not possible before without manual interventions. Depending on timing requirements, this approach might be substituted by a surrogate model. Nevertheless, it is suitable for the early production phase or pre-production.

Chapter 5: Validation and Troubleshooting of Smart Assembly Processes - This chapter addresses the troubleshooting of a DT-based assembly system. This work is also highly relevant as the simulation model does feature complex part behavior and interactions. Identifying correct countermeasures requires a high level of experience and well-known assembly components. The contribution of Chapter 5 does not directly contribute on the process level, but can be seen as a supporting process for the manufacturing process. With the presented troubleshooting tool, an inspection engineer can interactively explore possible countermeasures with ease. Subsequently, downtime of the assembly process can be drastically reduced.

6.2 Finite Element Simulation

Every approach that was presented in this dissertation uses FE-simulation or simulation outputs. This section discusses the used simulation techniques and also contributions to the field of FEM.

6.2.1 Element Selection

All chapters in this dissertation are using simulations of sheet metal components. As discussed in Section 1.3, different elements can be used for FE-computations. For the generation of the FE-model, shell elements are used. This kind of element is designed to provide efficient simulation for thin structures. The surface meshing does reduce the number of DOFs that must be solved by the FE-software. At the same time, the amount of elements that are needed to represent a thin geometry is reduced drastically, as the thickness of the part is substituted. For different kinds of simulation problems (i.e.thermal, acoustic, mechanical) a variety of shell elements were developed. Choosing a suitable type of shell element for the application is crucial for the simulation output quality. As only elastic deformations are expected for the presented use cases, simple first-order shell elements were selected. Although approximation of geometry is critical and second-order shell elements are preferable, first-order shell elements were used to increase computational stability.

6.2.2 Convergence Behavior

The iterative approaches that are based on FE-simulation ([1–3, 6]) rely on the convergence of the FE-solver. To increase the robustness of the FE-computation certain assumptions are made. First, no plastic deformations are presumed. A linear material model serves well for the applications in this work and is computationally more robust than a plastic material model. Next, all use cases have in common that the FE-simulation calculates a quasi-static state. This means no behavior such as dynamic movements, thermal loads, vibrations, etc., need to be considered. These factors reduce the FE-problem to a simple, linear, quasi-static simulation which is most likely to converge. During all experiments performed in this dissertation, the convergence of the FE-solver became never a problem.

6.2.3 Mesh Resolution

To achieve a good geometric representation of the parts, with first-order shell elements, a high-resolution mesh with adaptive element size is used. Depending on local curvature, the element size is changed to achieve a pre-defined surface deviation threshold. This threshold was chosen depending on part size and geometry. A second restriction was the maximum number of elements. Type and number of elements define the number of DOFs being calculated by the solver. To ensure acceptable solver time and needed computational resources, a maximum number of $\leq$ 400.000 elements was set as a target. From previous experiences with simulations of sheet metal components, mesh resolution was never critical for the precision of the simulation output when using these meshing criteria.

6.2.4 Contributions Regarding FE-Simulation

The outcome of this dissertation contributes to the field of FE-simulation, in particular to the branch of mesh generation methods. Chapter 4 does provide a novel, FE-based mesh morphing approach that generates geometrical DTs based on 3D-scan data. This approach enables mesh morphing without the need for manual interventions. Due to the high degree of automation, this method can be used for FE-model generation within production and pre-production processes for generating high-fidelity simulation models. A great advantage compared to existing solutions is this: a full simulation model with all definitions of material, connections, and contacts, can be used as input. The output of the morphing approach preserves these definitions. By this, time and cost for model generation can be reduced drastically.

6.3 Metrology

The field of Metrology, especially optical measurement devices are highly relevant for this dissertation. In different chapters (Chapter 3-5) measurements from an optical scan device are used for validation.

6.3.1 Scan Hardware

The scanning hardware used for validation was introduced in Chapter 1. As already discussed, this hardware does not meet industry standards but is sufficient for validation purposes within this dissertation. For real applications, measurement equipment of better quality is preferable. By using better hardware, improved measurement quality, process robustness, and scan speeds are expected. On the downside, more powerful computational resources are needed to handle larger measurement data.

6.3.2 Data Management

Implementing 3D-scanning to a production system does raise the need for a powerful data infrastructure. Modern 3D-scanners can acquire millions of vertices within seconds. The raw data sizes of a single scan in stl[1] file format easily exceeds Gigabytes. To handle such an amount of data, the measurement computer needs to be very powerful to process the raw data in an acceptable time frame. The next challenge is to manage occurring data. Companies like Zeiss or Hexagon are currently developing data management software that is capable of handling large amounts of 3D-scan data. So, commercial solutions can be expected in the near future.

6.3.3 Alignment

Aligning measurement data with the nominal reference frame can have a great impact on the proposed methods. To ensure consistent alignment for an in-line metrology system, automatic positioning of the specimens is required, as well as fixed points of reference within the scan window. By that, each scan can be automatically registered. To improve precision further, the scanning device should be mounted on a robotic arm. The position and orientation of the robot arm can be used to compose different scan windows.

6.3.4 Contribution to the Field Metrology

The main contribution of this dissertation to the field of metrology is the gravity compensation method for scanned non-rigid parts (Chapter 2). This method enables correct post-processing of scanned point clouds and compensates for errors that are occurring during the measurement process. This method is advanced compared to existing approaches which assume similarity between nominal and actual geometry. By employing this method to measurement processes, costly fixtures can be avoided while increasing outcome precision.

6.4 Discussion from the Perspective of Geometric Modeling

This dissertation has intersections with the field of geometric modeling. Especially, the automated mesh morphing approach presented in Chapter 4 directly contributes to this field

[1] Sterolithography

of research. The presented approach enables an automated generation of geometrical DTs based on a 3D-scan of the actual part and an FE-mesh representing the nominal geometry. By employing FE-simulation for calculating smooth displacement fields required for morphing, a robust and reliable method was achieved. Chapter 4 already provides a detailed discussion of the morphing approach covering crucial aspects from the field of geometric modeling.

6.5 Automotive and Aerospace Industry - The Users Perspective

All presented methods were designed to be used with thin non-rigid parts. These kinds of parts are commonly used in the automotive and aerospace industry. This section discusses this dissertation from the users' perspective. To gain insights discussions with experts from the automotive and aerospace industry were held on a regular basis. The gained insights are briefly highlighted in the following.

6.5.1 Aerospace Industry

In general, all approaches presented in this dissertation can be applied to the aerospace industry for instance for the assembly of large non-rigid components like fuselage parts. All methods are designed to support assembly processes, which become problematic due to the parts' mechanical weakness. For the aerospace industry, especially, the contributions from Chapter 1 were highly relevant. A similar approach is already implemented and used by a big commercial aircraft manufacturer for fuselage assembly. A main concern of the aerospace industry was also to achieve a more integrated workflow for simulation model generation, as this process is very time-consuming and requires manual work. This concern is partially addressed by the automated mesh morphing approach in Chapter 4.

6.5.2 Automotive Industry

The use cases that were presented in this dissertation were exclusively from the automotive industry. So, all methods and contributions are directly applicable to the assembly of exterior car components. In the following, concerns from the experts regarding the DT-based self-compensating assembly approach are discussed.

Internal or External Competences? - To realize such self-compensating assembly, complex implementations are necessary that need multi-disciplinary competencies. A choice that has to be made, where these competencies are built up - internally or bought in by external services. This makes the company dependent on either external suppliers or single experts. This dependence makes the whole manufacturing process vulnerable to dropping outs of experts. This kind of problem could, for instance, be tackled by a tech spin-off for maintaining such digital solutions.

Attractive Workplace? - Another question that arose was whether a workplace for the manual worker would be attractive if every action is imposed by a computer system. This aspect must be addressed but does not fit the scope of this work.

Competitive with existing solutions? - Whether proposed solutions are competitive to existing solutions, was proven. However, if a contribution of this dissertation has a benefit for the user, highly depends on the individual application. For the presented use cases, the benefits compared to existing approaches are clearly highlighted. For different kinds of uses, the presented approaches might still be applicable but not necessarily preferable.

Chapter 7

Conclusions

This dissertation presented a DT-based approach for self-compensating assembly processes for the purpose of sheet metal assemblies. Challenges for realization were gathered, and each chapter features a solution for a sub-problem within the scope of this work. The major contributions of this work are:

(1) Framework for DT-based self-compensating assembly system for exterior car components.

(2) Method for calculating the gravity-free shape of 3D-scanned non-rigid parts. Verification and validation with experimental data of presented method.

(3) Automated mesh morphing approach for surface meshes. Verification and validation using experimental data of real car body parts.

(4) Interactive troubleshooting tool for deriving countermeasures based on measured shape deviations. Verification and validation using experimental data of real car body parts.

Each contribution either improves or extends existing approaches or provides benefits, e.g. higher precision, compared to existing solutions. In particular, (1) extends existing DT-approaches for simple sheet metal assemblies to exterior assembly components. These components have special requirements which were not addressed before. The method for calculating the GF-shape of measured sheet metal components (2) is a novel approach that provides more precise results compared to existing approaches. This method has been implemented as a software prototype and is already used for real applications in the automotive industry. The automated mesh morphing approach (3) is also a novel concept that substantially simplifies the generation of geometrical DTs. By using this approach, manual work is drastically reduced for generating simulation models from acquired point clouds.

The interactive troubleshooting tool (4) features an interactive approach which enables the user exploring the space of possible corrections. By providing real-time predictions for the effect of changes to the assembly, the user can quickly understand the behaviors of the part and find the best set of corrections before applying. This tool is entirely new and enhances troubleshooting processes in sheet metal assemblies.

References

[1] Felix Claus, Franca-Alexandra Rupprecht, and Hans Hagen. Online simulation considering production uncertainties to improve assembly quality. NAFEMS, 2019.

[2] Felix Claus, Hans Hagen, and Bernd Hamann. Calculating the gravity-free shape of sheet metal parts. *The International Journal of Advanced Manufacturing Technology*, 113(11-12):3401–3417, 2021.

[3] Felix Claus, Bernd Hamann, and Hans Hagen. Calculating stress-free shapes of sheet metal parts measured with over-constrained fixtures. NAFEMS, 2021.

[4] Felix Claus, Hans Hagen, Viktor Leonhardt, Heike Leitte, and Bernd Hamann. Interactive Quality Inspection of Measured Deviations in Sheet Metal Assemblies. In *2nd International Conference of the DFG International Research Training Group 2057 – Physical Modeling for Virtual Manufacturing (iPMVM 2020)*, volume 89 of *Open Access Series in Informatics (OASIcs)*, pages 6:1–6:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[5] Felix Claus, Bernd Hamann, Heike Leitte, and Hans Hagen. Decomposing deviations of scanned surfaces of sheet metal assemblies. *Journal of Manufacturing Systems*, 61:125–138, 2021.

[6] Felix Claus, Bernd Hamann, and Hans Hagen. A finite-element based mesh morphing approach for surface meshes. *Computer-Aided Design*, 146:103232, 2022.

[7] Viktor Leonhardt, Felix Claus, and Christoph Garth. Pen: Process estimator neural network for root cause analysis using graph convolution. *Journal of Manufacturing Systems*, 2021.

[8] Patrick Ruediger, Felix Claus, Bernd Hamann, Hans Hagen, and Heike Leitte. Combining visual analytics and machine learning for reverse engineering in assembly quality control. *Electronic Imaging*, 2021(1):60405–1–60405–13, 2021.

[9] Patrick Ruediger, Felix Claus, Viktor Leonhardt, Hans Hagen, Jan C. Aurich, and Christoph Garth. Previs – a combined machine learning and visual interpolation approach for interactive reverse engineering in assembly quality control, 2022.

[10] DIN EN ISO. Din en iso 10579:2013-11, geometrische produktspezifikation (gps)_-bemaßung und tolerierung_- nicht-formstabile teile (iso_10579:2010, einschließlich cor 1:2011); deutsche fassung en_iso_10579:2013, 2013.

[11] ISO. Iso 23247-1:2021: Automation systems and integration — digital twin framework for manufacturing — part 1: Overview and general principles, 2021.

[12] The finite element method: Its basis and fundamentals. In O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu, editors, *The Finite Element Method: its Basis and Fundamentals (Seventh Edition)*, page i. Butterworth-Heinemann, Oxford, seventh edition edition, 2013.

[13] The finite element method for solid and structural mechanics. In O.C. Zienkiewicz, R.L. Taylor, and David Fox, editors, *The Finite Element Method for Solid and Structural Mechanics (Seventh Edition)*, page i. Butterworth-Heinemann, Oxford, seventh edition edition, 2014.

[14] The finite element method for fluid dynamics. In O.C. Zienkiewicz, R.L. Taylor, and P. Nithiarasu, editors, *The Finite Element Method for Fluid Dynamics (Seventh Edition)*, page i. Butterworth-Heinemann, Oxford, seventh edition edition, 2014.

[15] Yucheng Liu and Gary Glass. Effects of mesh density on finite element analysis. Technical report, SAE Technical Paper, 2013.

[16] Dassault Systèmes. Abaqus 6.11: Analysis user's manual volume iv: Elements, 2011.

[17] Fritz Klocke, Thorsten Beck, Stefan Hoppe, Tilo Krieg, Norbert Müller, Tobias Nöthe, Hans-Willi Raedt, and Kevin Sweeney. Examples of fem application in manufacturing technology. *Journal of Materials Processing Technology*, 120(1):450–457, 2002.

[18] SD Mohammed and H Desai. Basic concepts of finite element analysis and its applications in dentistry: An overview. *Oral Hyg Health*, 2(156):2332–0702, 2014.

[19] Nitin Kumar Sahu and Ajay Kumar Kaviti. A review of use fem techniques in modeling of human knee joint. *Journal of Biomimetics, Biomaterials and Biomedical Engineering*, 28:14–25, 10 2016.

[20] Andrés García-Pérez, Félix Sorribes-Palmer, Gustavo Alonso, and Ali Ravanbakhsh. Overview and application of fem methods for shock analysis in space instruments. *Aerospace Science and Technology*, 80:572–586, 2018.

[21] Aniello Riccio, Salvatore Saputo, and Andrea Sellitto. Crashworthiness of a general aviation fuselage section: 3d fem numerical modelling and validation. *Materials Today: Proceedings*, 34:69–73, 2021. 12th International Conference on Composite Science and Technology.

[22] Piotr Iwicki and Jarosław Przewłócki. Short review and 3-d fem analysis of basic types of foundation for offshore wind turbines. *Polish Maritime Research*, 2020.

[23] Jun Wang, Derek B. Apel, Yuanyuan Pu, Robert Hall, Chong Wei, and Mohammadali Sepehri. Numerical modeling for rockbursts: A state-of-the-art review. *Journal of Rock Mechanics and Geotechnical Engineering*, 13(2):457–478, 2021.

[24] Huajiang Ouyang, Wayne Nack, Yongbin Yuan, and Frank Chen. Numerical analysis of automotive disc brake squeal: a review. *International Journal of Vehicle Noise and Vibration*, 1(3-4):207–231, 2005.

[25] Olaf Engler and Jürgen Hirsch. Texture control by thermomechanical processing of aa6xxx al–mg–si sheet alloys for automotive applications—a review. *Materials Science and Engineering: A*, 336(1):249–262, 2002.

[26] Philipp Spethmann, Cornelius Herstatt, and Stefan H. Thomke. Crash simulation evolution and its impact on r&d in the automotive applications. *International Journal of Product Development*, 8(3):291, 2009.

[27] SC Liu, SJ Hu, and TC Woo. Tolerance analysis for sheet metal assemblies. 1996.

[28] Karl-Heinrich Grote and Jörg Feldhusen. *Dubbel*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[29] Volkswagen AG. Reference point system - rps -, 1996.

[30] Jaime A Camelio and Benjamin Heichelbech. Comparison of diagnosis methodologies on sheet metal assembly. *Trans Namri/sme*, 34:135–142, 2006.

[31] Jibin J. Mathew, Chandrashekhar N. Sakhale, and Sagar D. Shelare. Latest trends in sheet metal components and its processes—a literature review. In Harish Sharma, Kannan Govindan, Ramesh C. Poonia, Sandeep Kumar, and Wael M. El-Medany, editors, *Advances in Computing and Intelligent Systems*, pages 565–574, Singapore, 2020. Springer Singapore.

[32] James J Rooney and Lee N Vanden Heuvel. Root cause analysis for beginners. *Quality progress*, 37(7):45–56, 2004.

[33] Robin E McDermott, Raymond J Mikulak, and Michael R Beauregard. *FMEA*. New York: Taylor & Francis Group, 2009.

[34] Johan S. Carlson and Rikard Söderberg. Assembly root cause analysis: A way to reduce dimensional variation in assembled products. *International Journal of Flexible Manufacturing Systems*, 15(2):113–150, 2003.

[35] Junkang Guo, Baotong Li, Zhigang Liu, Jun Hong, and Xiaopan Wu. Integration of geometric variation and part deformation into variation propagation of 3-d assemblies. *International Journal of Production Research*, 54(19):5708–5721, 2016.

[36] Jaime Camelio, S Jack Hu, and Dariusz Ceglarek. Modeling variation propagation of multi-station assembly systems with compliant parts. *J. Mech. Des.*, 125(4):673–681, 2003.

[37] Pasquale Franciosa, Arnab Palit, Ferdinando Vitolo, and Darek Ceglarek. Rapid response diagnosis of multi-stage assembly process with compliant non-ideal parts using self-evolving measurement system. *Procedia CIRP*, 60:38–43, 2017.

[38] Gerardo Beruvides, Alberto Villalonga, Pasquale Franciosa, Darek Ceglarek, and Rodolfo E Haber. Fault pattern identification in multi-stage assembly processes with non-ideal sheet-metal parts based on reinforcement learning architecture. *Procedia Cirp*, 67:601–606, 2018.

[39] Jaime A. Camelio, S. Jack Hu, and Dariusz Ceglarek. Impact of fixture design on sheet metal assembly variation. *Journal of Manufacturing Systems*, 23(3):182 – 193, 2004.

[40] Xiaoyun Liao and G. Gary Wang. Simultaneous optimization of fixture and joint positions for non-rigid sheet metal assembly. *The International Journal of Advanced Manufacturing Technology*, 36(3):386–394, 2008.

[41] Cong Lu and Hong-Wang Zhao. Fixture layout optimization for deformable sheet metal workpiece. *The International Journal of Advanced Manufacturing Technology*, 78(1):85–98, 2015.

[42] Bing Li, Hongjian Yu, Xiaojun Yang, and Ying Hu. Variation analysis and robust fixture design of a flexible fixturing system for sheet metal assembly. *Journal of manufacturing science and engineering*, 132(4), 2010.

[43] Kristina Wärmefjord, Johan S Carlson, and Rikard Söderberg. Controlling geometrical variation caused by assembly fixtures. *Journal of Computing and Information Science in Engineering*, 16(1):011007, 2016.

[44] A. Weckenmann, T. Estler, G. Peggs, and D. McMurtry. Probing systems in dimensional metrology. *CIRP Annals*, 53(2):657–684, 2004.

[45] I. Kovac. Flexible inspection systems in the body-in-white manufacturing. In *International Workshop on Robot Sensing, 2004. ROSE 2004.*, pages 41–48, 2004.

[46] Mingrang Yu, Yingjie Zhang, Yunlong Li, and Ding Zhang. Adaptive sampling method for inspection planning on cmm for free-form surfaces. *The International Journal of Advanced Manufacturing Technology*, 67(9-12):1967–1975, 2013.

[47] Andres G. Marrugo, Feng Gao, and Song Zhang. State-of-the-art active optical techniques for three-dimensional surface metrology: a review [invited]. *J. Opt. Soc. Am. A*, 37(9):B60–B77, Sep 2020.

[48] Qian Wang and Min-Koo Kim. Applications of 3d point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Advanced Engineering Informatics*, 39:306–319, 2019.

[49] Song Zhang. High-speed 3d shape measurement with structured light methods: A review. *Optics and Lasers in Engineering*, 106:119–131, 2018.

[50] R.H. Schmitt, M. Peterek, E. Morse, W. Knapp, M. Galetto, F. Härtig, G. Goch, B. Hughes, A. Forbes, and W.T. Estler. Advances in large-scale metrology – review and future trends. *CIRP Annals*, 65(2):643–665, 2016.

[51] Abid Haleem and Mohd. Javaid. 3d scanning applications in medical field: A literature-based review. *Clinical Epidemiology and Global Health*, 7(2):199–210, 2019.

[52] Quan Shi, Ning Xi, Weihua Sheng, and Yifan Chen. Development of dynamic inspection methods for dimensional measurement of automotive body parts. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 315–320, May 2006.

[53] Kwan H Lee and Hyun pung Park. Automated inspection planning of free-form shape parts by laser scanning. *Robotics and Computer-Integrated Manufacturing*, 16(4):201 – 210, 2000.

[54] Ercihan Kiraci, Pasquale Franciosa, Glen A. Turley, Alan Olifent, Alex Attridge, and Mark A. Williams. Moving towards in-line metrology: evaluation of a laser radar system for in-line dimensional inspection for automotive assembly systems. *The International Journal of Advanced Manufacturing Technology*, 91(1-4):69–78, 2017.

[55] Manoj Babu, Pasquale Franciosa, and Dariusz Ceglarek. Spatio-temporal adaptive sampling for effective coverage measurement planning during quality inspection of free form surfaces using robotic 3d optical scanner. *Journal of Manufacturing Systems*, 53:93 – 108, 2019.

[56] Rui Chen, Jing Xu, and Song Zhang. Digital fringe projection profilometry. In *Advances in Optical Form and Coordinate Metrology*, 2053-2563, pages 5–1 to 5–28. IOP Publishing, 2020.

[57] Shijie Feng, Liang Zhang, Chao Zuo, Tianyang Tao, Qian Chen, and Guohua Gu. High dynamic range 3d measurements with fringe projection profilometry: a review. *Measurement Science and Technology*, 29(12):122001, nov 2018.

[58] Kenta-Tanaka et al. probreg.

[59] Hassan Radvar-Esfahlan and Souheil-Antoine Tahan. Nonrigid geometric metrology using generalized numerical inspection fixtures. *Precision Engineering*, 36(1):1–9, 2012.

[60] V. Sabri, S. A. Tahan, X. T. Pham, D. Moreau, and S. Galibois. Fixtureless profile inspection of non-rigid parts using the numerical inspection fixture with improved definition of displacement boundary conditions. *The International Journal of Advanced Manufacturing Technology*, 82(5-8):1343–1352, 2016.

[61] P. Aivaliotis, K. Georgoulias, and G. Chryssolouris. The use of digital twin for predictive maintenance in manufacturing. *International Journal of Computer Integrated Manufacturing*, 32(11):1067–1080, 2019.

[62] Yujia Zhang and Alper Yilmaz. Structured light based 3d scanning for specular surface by the combination of gray code and phase shifting. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 41, 2016.

[63] Shijie Feng, Chao Zuo, Liang Zhang, Tianyang Tao, Yan Hu, Wei Yin, Jiaming Qian, and Qian Chen. Calibration of fringe projection profilometry: A comparative review. *Optics and Lasers in Engineering*, 143:106622, 2021.

[64] HP Customer Support - Knowledge Base. Hp 3d scan software pro v5 online instructions.

[65] Dawson-Haggerty et al. trimesh.

[66] Ahrens, James, Geveci, Berk, Law, Charles. *The visualization handbook: ParaView: An End-User Tool for Large Data Visualization*. Elsevier Butterworth-Heinemann, Burlington, MA, 2005.

[67] Long Hoang Pham, Duong Nguyen-Ngoc Tran, Jin Young Byun, Chul Hong Rhie, and Jae Wook Jeon. A smartphone-based laser measuring system for gap and flush assessment in car body. *IEEE Transactions on Industrial Electronics*, 68(7):6297–6307, 2021.

[68] Kun Long, Qian Xie, Dening Lu, Qiaoyun Wu, Yuanpeng Liu, and Jun Wang. Aircraft skin gap and flush measurement based on seam region extraction from 3d point cloud. *Measurement*, 176:109169, 2021.

[69] Jinsu Kim, Jae-Sung Kim, and Hyunjune Yim. Tolerance analysis and design of refrigerator door system for functional and aesthetic quality of gap and flush. *Journal of the Korean Society for Precision Engineering*, 31(1):59–66, 2014.

[70] Elisa Minnetti, Paolo Chiariotti, Nicola Paone, Gisela Garcia, Helder Vicente, Luca Violini, and Paolo Castellini. A smartphone integrated hand-held gap and flush measurement system for in line quality control of car body assembly. *Sensors*, 20(11), 2020.

[71] Thi-Trang Tran and CheolKeun Ha. Non-contact gap and flush measurement using monocular structured multi-line light vision for vehicle assembly. *International Journal of Control, Automation and Systems*, 16(5):2432–2445, 2018.

[72] Long Hoang Pham, Duong Nguyen-Ngoc Tran, Chul Hong Rhie, and Jae Wook Jeon. A mobile vision-based system for gap and flush measuring between planar surfaces using aruco markers. In *2021 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–4, 2021.

[73] Chevassus Nicolas, Falgarone Hugo, Thiebaut François, Bourdet Pierre, and Georges-Eric Moufle. A new approach for best fit assembly based on the behaviour of components. *SAE Technical Paper*, pages 01–3174, 2006.

[74] Zhehan Chen, Fuzhou Du, and Xiaoqing Tang. Position and orientation best-fitting based on deterministic theory during large scale assembly. *Journal of Intelligent Manufacturing*, 29(4):827–837, 2018.

[75] R Schmitt and F Moenning. Ensure success with inline-metrology. In *XVIII IMEKO world congress Metrology for a Sustainable Development*, pages 17–22, 2006.

[76] Oliver Sand, Sebastian Büttner, Volker Paelke, and Carsten Röcker. smart.assembly – projection-based augmented reality for supporting assembly workers. In Stephanie Lackey and Randall Shumaker, editors, *Virtual, Augmented and Mixed Reality*, pages 643–652, Cham, 2016. Springer International Publishing.

[77] Shiqing Wu, Zhonghou Wang, Bin Shen, Jia-Hai Wang, and Li Dongdong. Human-computer interaction based on machine vision of a smart assembly workbench. *Assembly Automation*, 2020.

[78] Adam Kleppe, Asgeir Bjørkedal, Kristoffer Larsen, and Olav Egeland. Automated assembly using 3d and 2d cameras. *Robotics*, 6(3):14, 2017.

[79] Humyun Fuad Rahman, Mukund Nilakantan Janardhanan, and Peter Nielsen. An integrated approach for line balancing and agv scheduling towards smart assembly systems. *Assembly Automation*, 2020.

[80] Abolfazl Rezaei Aderiani, Kristina Wärmefjord, and Rikard Söderberg. Combining selective assembly and individualized locator adjustments techniques in a smart assembly line. *Procedia CIRP*, 97:429–434, 2021. 8th CIRP Conference of Assembly Technology and Systems.

[81] J. Wahle, L. Neubert, J. Esser, and M. Schreckenberg. A cellular automaton traffic flow model for online simulation of traffic. *Parallel Computing*, 27(5):719–735, 2001.

[82] André Hanisch, Juri Tolujew, Klaus Richter, and Thomas Schulze. Modeling people flow: Online simulation of pedestrian flow in public buildings. In *Proceedings of the 35th Conference on Winter Simulation: Driving Innovation*, WSC '03, pages 1635–1641. Winter Simulation Conference, 2003.

[83] Anton Jaklič, Franci Vode, and Tomaž Kolenko. Online simulation model of the slab-reheating process in a pusher-type furnace. *Applied Thermal Engineering*, 27(5):1105–1114, 2007.

[84] O. Kabadurmus, O. Pathak, J. S. Smith, A. E. Smith, and H. Yapicioglu. A simulation methodology for online process control of hot mix asphalt (hma) production. In *Proceedings of the 2010 Winter Simulation Conference*, pages 1522–1533, 2010.

[85] Vicente Román-Ibáñez, Antonio Jimeno-Morenilla, Francisco Pujol-López, and Faustino Salas-Pérez. Online simulation as a collision prevention layer in automated shoe sole adhesive spraying. *The International Journal of Advanced Manufacturing Technology*, 95(1):1243–1253, 2018.

[86] Rikard Söderberg, Kristina Wärmefjord, Johan S. Carlson, and Lars Lindkvist. Toward a digital twin for real-time geometry assurance in individualized production. *CIRP Annals*, 66(1):137–140, 2017.

[87] André I Khuri and Siuli Mukhopadhyay. Response surface methodology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(2):128–149, 2010.

[88] Jack PC Kleijnen. Kriging metamodeling in simulation: A review. *European journal of operational research*, 192(3):707–716, 2009.

[89] PA Ramachandran and K Balakrishnan. Radial basis functions as approximate particular solutions: review of recent progress. *Engineering analysis with boundary elements*, 24(7-8):575–582, 2000.

[90] Giorgos Mountrakis, Jungho Im, and Caesar Ogole. Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3):247–259, 2011.

[91] Magali RG Meireles, Paulo EM Almeida, and Marcelo Godoy Simões. A comprehensive review for industrial applicability of artificial neural networks. *IEEE transactions on industrial electronics*, 50(3):585–601, 2003.

[92] Tanmoy Chatterjee, Souvik Chakraborty, and Rajib Chowdhury. A critical review of surrogate assisted robust design optimization. *Archives of Computational Methods in Engineering*, 26(1):245–274, 2019.

[93] V.V. Tkachev. The use of superelement approach for the mathematical simulation of reactor structure dynamic behaviour. *Nuclear Engineering and Design*, 196(1):101–104, 2000.

[94] Zu-Qing Qu and R. Panneer Selvam. Dynamic superelement modeling method for compound dynamic systems. *AIAA journal*, 38(6):1078–1083, 2000.

[95] Shujuan Huang and Pengzhong Li. Superelement method based structure simulation of large pallet structure. In *Computer Engineering and Applications (ICCEA), 2010 Second International Conference on*, volume 1, pages 92–95, 2010.

[96] Markus Burchert and Mario Schwalbe. Neue berechnungsmethoden zur nvh-bewertung des powertrains. In Johannes Liebl and Gotthard Rainer, editors, *VPC.plus 2014*, pages 290–308, Wiesbaden, 2018. Springer Fachmedien Wiesbaden.

[97] S Arzanpour, J Fung, JK Mills, and WL Cleghorn. Flexible fixture design with applications to assembly of sheet metal automotive body parts. *Assembly Automation*, 2006.

[98] Ilker Erdem, Henrik Kihlman, and Alf Andersson. Development of affordable reconfigurable tooling in car manufacturing cells–a case study. In *23rd International Conference for Production Research, ICPR 2015, Manila, Philippines, 2-6 August 2015*, 2015.

[99] Feiyan Guo, Zhongqi Wang, Jianhua Liu, Fang Zou, and Xudong Zhao. Locating method and motion stroke design of flexible assembly tooling for multiple aircraft components. *The International Journal of Advanced Manufacturing Technology*, 107(1):549–571, 2020.

[100] Sandrine Germain and Paul Steinmann. A comparison between inverse form finding and shape optimization methods for anisotropic hyperelasticity in logarithmic strain space. *PAMM*, 11(1):367–368, 2011.

[101] Sandrine Germain, Philipp Landkammer, and Paul Steinmann. On a recursive formulation for solving inverse form finding problems in isotropic elastoplasticity. *Advanced Modeling and Simulation in Engineering Sciences*, 1(1):10, 2014.

[102] Philipp Landkammer, Sandrine Germain, and Paul Steinmann. On inverse form finding for orthotropic plasticity. *Computer Assisted Methods in Engineering and Science*, 20(4):337–348, 2017.

[103] Anna Ask, Ralf Denzer, Andreas Menzel, and Matti Ristinmaa. Inverse-motion-based form finding for quasi-incompressible finite electroelasticity. *International Journal for Numerical Methods in Engineering*, 94(6):554–572, 2013.

[104] Claire Lartigue, François Thiebaut, Pierre Bourdet, and Nabil Anwer. Dimensional metrology of flexible parts: identification of geometrical deviations from optical measurements. In *Advanced mathematical and computational tools in metrology VII*, pages 196–203. World Scientific, 2006.

[105] S. A. Meyer and P. D. Wolf. Application of sonomicrometry and multidimensional scaling to cardiac catheter tracking. *IEEE Transactions on Biomedical Engineering*, 44(11):1061–1067, Nov 1997.

[106] K.Y. Choi and S.S. Kim. Morphological analysis and classification of types of surface corrosion damage by digital image processing. *Corrosion Science*, 47(1):1 – 15, 2005.

[107] Yoshitada Mito, Mohd Ashraf Mohamad Ismail, and Takuji Yamamoto. Multidimensional scaling and inverse distance weighting transform for image processing of hydrogeological structure in rock mass. *Journal of Hydrology*, 411(1):25 – 36, 2011.

[108] Ingwer Borg and Patrick J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer Series in Statistics. Springer Science+Business Media Inc, New York, NY, second edition edition, 2005.

[109] Nasir Saeed, Haewoon Nam, Mian Imtiaz Ul Haq, and Dost Bhatti Muhammad Saqib. A survey on multidimensional scaling. *ACM Comput. Surv.*, 51(3):47:1–47:25, May 2018.

[110] François Thiébaut, Cyril Lacroix, Loïc Andolfatto, and Claire Lartigue. Evaluation of the shape deviation of non rigid parts from optical measurements. *The International Journal of Advanced Manufacturing Technology*, 88(5-8):1937–1944, 2017.

[111] Valtteri Tuominen. Virtual clamping in automotive production line measurement. *Expert Systems with Applications*, 38(12):15065–15074, 2011.

[112] Sasan Sattarpanah Karganroudi, Jean-Christophe Cuillière, Vincent François, and Souheil-Antoine Tahan. Assessment of the Robustness of a Fixtureless Inspection Method for Nonrigid Parts Based on a Verification and Validation Approach. *Journal of Verification, Validation and Uncertainty Quantification*, 2(4), 02 2018. 041002.

[113] Gad Noriel Abenhaim, Alain Desrochers, and Antoine Tahan. Nonrigid parts' specification and inspection methods: notions, challenges, and recent advancements. *The International Journal of Advanced Manufacturing Technology*, 63(5):741–752, Nov 2012.

[114] W. Cai, S. J. Hu, and J. X. Yuan. Deformable sheet metal fixturing: Principles, algorithms, and simulations. *Journal of Manufacturing Science and Engineering*, 118(3):318–324, 1996.

[115] D. Banabic. Anisotropy of sheet metal. In Dorel Banabic, H.-J Bunge, K. Pöhlandt, and A. E. Tekkaya, editors, *Formability of Metallic Materials*, volume 42 of *Engineering Materials*, pages 119–172. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[116] Taro Kizu, Kaneharu Okuda, Yasunobu Nagataki, Toshiaki Urabe, and Kazuhiro Seto. Influence of tensile strain on young's modulus in high-strength cold-rolled steel sheets. *ISIJ International*, 55(7):1502–1511, 2015.

[117] Dassault Systèmes. Simulia user assistance 2019, 2019.

[118] M. D. McKay, R. J. Beckman, and W. J. Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[119] Tamas Varady, Ralph R Martin, and Jordan Cox. Reverse engineering of geometric models—an introduction. *Computer-aided design*, 29(4):255–268, 1997.

[120] Eyup Bagci. Reverse engineering applications for recovery of broken or worn parts and re-manufacturing: Three case studies. *Advances in Engineering Software*, 40(6):407–418, 2009.

[121] Nabil Anwer and Luc Mathieu. From reverse engineering to shape engineering in mechanical design. *CIRP Annals*, 65(1):165–168, 2016.

[122] R. Racasan, D. Popescu, C. Neamtu, and M. Dragomir. Integrating the concept of reverse engineering in medical applications. In *2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, volume 2, pages 1–5, 2010.

[123] Hacene Ameddah and Mekki Assas. Bio-cad reverse engineering of free-form surfaces by planar contours. *Computer-aided design and applications*, 7(sup1):1–7, 2010.

[124] In Tae Yang, Tri Dev Acharya, Dong Ha Lee, and Moon Seung Shin. Reverse engineering of a bobsleigh structure using lidar. *International Journal of Applied Engineering Research*, 12(6):976–980, 2017.

[125] M. Cabaleiro, B. Riveiro, P. Arias, J.C. Caamaño, and J.A. Vilán. Automatic 3d modelling of metal frame connections from lidar data for structural engineering purposes. *ISPRS Journal of Photogrammetry and Remote Sensing*, 96:47–56, 2014.

[126] Karan Singh. Industrial motivation for interactive shape modeling: A case study in conceptual automotive design. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, page 3–9, New York, NY, USA, 2006. Association for Computing Machinery.

[127] Yunbao Huang and Xiaoping Qian. Dynamic b-spline surface reconstruction: Closing the sensing-and-modeling loop in 3d digitization. *Computer-Aided Design*, 39(11):987–1002, 2007.

[128] Zhiqiang Yu, Taiyong Wang, Peng Wang, Ying Tian, and Hongbin Li. Rapid and precise reverse engineering of complex geometry through multi-sensor data fusion. *IEEE Access*, 7:165793–165813, 2019.

[129] Fabio Remondino. From point cloud to surface: the modeling and visualization problem.

[130] Iacopo Gentilini and Kenji Shimada. Predicting and evaluating the post-assembly shape of thin-walled components via 3d laser digitization and fea simulation of the assembly process. *Computer-Aided Design*, 43(3):316–328, 2011.

[131] Kenji Shimada. *Physically-based mesh generation: automated triangulation of surfaces and volumes via bubble packing*. PhD thesis, Massachusetts Institute of Technology, 1993.

[132] Boliang Guan, Shujin Lin, Ruomei Wang, Fan Zhou, Xiaonan Luo, and Yongchuan Zheng. Voxel-based quadrilateral mesh generation from point cloud. *Multimedia Tools and Applications*, 79(29):20561–20578, 2020.

[133] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, 1986.

[134] *Hot-to-Cold CAD Geometry Transformation of Aero Engine Parts Based on B-Spline Morphing*, volume Volume 2B: Turbomachinery of *Turbo Expo: Power for Land, Sea, and Air*, 06 2014. V02BT45A020.

[135] *Introduction of a Parameter Based Compressor Blade Model for Considering Measured Geometry Uncertainties in Numerical Simulation*, volume Volume 6: Structures and Dynamics, Parts A and B of *Turbo Expo: Power for Land, Sea, and Air*, 06 2009.

[136] Alexander Lange, Matthias Voigt, Konrad Vogeler, Henner Schrapp, Erik Johann, and Volker Gümmer. Impact of Manufacturing Variability on Multistage High-Pressure Compressor Performance. *Journal of Engineering for Gas Turbines and Power*, 134(11), 09 2012. 112601.

[137] Marc Alexa. Recent advances in mesh morphing. *Computer Graphics Forum*, 21(2):173–198, 2002.

[138] M.M. Selim and R.P. Koomullil. Mesh Deformation Approaches – A Survey. *Journal of Physical Mathematics*, 7(2):1 – 9, 2016.

[139] H Van der Auweraer, T Van Langenhove, M Brughmans, I Bosmans, N Masri, and S Donders. Application of mesh morphing technology in the concept phase of vehicle development. *International journal of vehicle design*, 43(1-4):281–305, 2007.

[140] Pasquale Franciosa, Salvatore Gerbino, and Stanislao Patalano. Simulation of variational compliant assemblies with shape errors based on morphing mesh approach. *The International Journal of Advanced Manufacturing Technology*, 53(1):47–61, 2011.

[141] Jibum Kim, Brian J. Miller, and Suzanne M. Shontz. A hybrid mesh deformation algorithm using anisotropic pdes and multiobjective mesh optimization. *Computers & Mathematics with Applications*, 70(8):1830–1851, 2015.

[142] T. Meinders, I.A. Burchitz, M.H.A. Bonte, and R.A. Lingbeek. Numerical product design: Springback prediction, compensation and optimization. *International Journal of Machine Tools and Manufacture*, 48(5):499–514, 2008. Advances in Sheet Metal Forming Applications.

[143] S. C. Liu, S. J. Hu, and T. C. Woo. Tolerance Analysis for Sheet Metal Assemblies. *Journal of Mechanical Design*, 118(1):62–67, 03 1996.

[144] Jaime Camelio, S. Jack Hu, and Dariusz Ceglarek. Modeling Variation Propagation of Multi-Station Assembly Systems With Compliant Parts . *Journal of Mechanical Design*, 125(4):673–681, 01 2004.

[145] A. Zerwer, G. Cascante, and J. Hutchinson. Parameter estimation in finite element simulations of rayleigh waves. *Journal of Geotechnical and Geoenvironmental Engineering*, 128(3):250–261, 2002.

[146] Erin Santini Bell, Masoud Sanayei, Chitra N. Javdekar, and Eugene Slavsky. Multiresponse parameter estimation for finite-element model updating using nondestructive test data. *Journal of Structural Engineering*, 133(8):1067–1079, 2007.

[147] Mario Teixeira Parente, Steven Mattis, Shubhangi Gupta, Christian Deusner, and Barbara Wohlmuth. Efficient parameter estimation for a methane hydrate model with active subspaces. *Computational Geosciences*, 23(2):355–372, 2019.

[148] A. S. Gendy and A. F. Saleeb. Nonlinear material parameter estimation for characterizing hyper elastic large strain models. *Computational Mechanics*, 25(1):66–77, 2000.

[149] F. Yoshida, M. Urabe, and V.V. Toropov. Identification of material parameters in constitutive model for sheet metals from cyclic bending tests. *International Journal of Mechanical Sciences*, 40(2):237 – 249, 1998.

[150] Giulio Reina, Matilde Paiano, and Jose-Luis Blanco-Claraco. Vehicle parameter estimation using a model-based estimator. *Mechanical Systems and Signal Processing*, 87:227 – 241, 2017. Signal Processing and Control challenges for Smart Vehicles.

[151] Michael L. Johnson and Lindsay M. Faunt. Parameter estimation by least-squares methods. In *Numerical Computer Methods*, volume 210 of *Methods in Enzymology*, pages 1 – 37. Academic Press, 1992.

[152] Kuan Lu, Yulin Jin, Yushu Chen, Yongfeng Yang, Lei Hou, Zhiyong Zhang, Zhonggang Li, and Chao Fu. Review for order reduction based on proper orthogonal decomposition and outlooks of applications in mechanical systems. *Mechanical Systems and Signal Processing*, 123:264 – 297, 2019.

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

[154] Björn Lindau, Lars Lindkvist, Alf Andersson, and Rikard Söderberg. Statistical shape modeling in virtual assembly using pca-technique. *Journal of Manufacturing Systems*, 32(3):456–463, 2013. Assembly Technologies and Systems.

[155] Lazhar Homri, Edoh Goka, Guillaume Levasseur, and Jean-Yves Dantan. Tolerance analysis — form defects modeling and simulation by modal decomposition and optimization. *Computer-Aided Design*, 91:46–59, 2017.

[156] Pierre-Antoine Adragna, Serge Samper, Fabien Formosa, and Maurice Pillet. *Modal Tolerancing — Application to Gap and Flush Analyses*, pages 417–430. Springer Netherlands, Dordrecht, 2007.

[157] Edoh Goka, Lazhar Homri, Pierre Beaurepaire, and Jean-Yves Dantan. Geometrical variation simulation for assembly with form defects. 2017.

[158] Abhishek Das, Pasquale Franciosa, Achille Pesce, and Salvatore Gerbino. Parametric effect analysis of free-form shape error during sheet metal forming. *International Journal of Engineering Science and Technology*, 9(09S):117–124, 2017.

[159] Samuel Nolan, Augusto Smerzi, and Luca Pezzè. A machine learning approach to bayesian parameter estimation. *npj Quantum Information*, 7(1):169, 2021.

[160] Parameter estimation for optimal asteroid transfer trajectories using supervised machine learning. *Aerospace Science and Technology*, 79:570–579, 2018.

[161] C. Wang, X.P. Tan, S.B. Tor, and C.S. Lim. Machine learning in additive manufacturing: State-of-the-art and perspectives. *Additive Manufacturing*, 36:101538, 2020.

[162] Sumit Sinha, Emile Glorieux, Pasquale Franciosa, and Dariusz Ceglarek. 3D convolutional neural networks to estimate assembly process parameters using 3D point-clouds. In Ettore Stella, editor, *Multimodal Sensing: Technologies and Applications*, volume 11059, pages 89 – 101. International Society for Optics and Photonics, SPIE, 2019.

[163] Sumit Sinha, Pasquale Franciosa, and Dariusz Ceglarek. Object shape error response using bayesian 3-d convolutional neural networks for assembly systems with compliant parts. *IEEE Transactions on Industrial Informatics*, 17(10):6676–6686, 2021.

[164] Evan Strasnick, Maneesh Agrawala, and Sean Follmer. Scanalog. In Krzysztof Gajos, Jennifer Mankoff, and Chris Harrison, editors, *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 321–330, New York, NY, USA, 10202017. ACM.

[165] Yuki Mori and Takeo Igarashi. Plushie. *ACM Transactions on Graphics*, 26(99):45, 2007.

[166] J. Whyte, N. Bouchlaghem, A. Thorpe, and R. McCaffer. From cad to virtual reality: modelling approaches, data exchange and interactive 3d building design tools. *Automation in Construction*, 10(1):43–55, 2000.

[167] Gergely Kristóf and Bálint Papp. Application of gpu-based large eddy simulation in urban dispersion studies. *Atmosphere*, 9(11):442, 2018.

[168] Christopher Fleischmann, Irina Leher, Reinhold Hartwich, Marc Hainke, and Stefan Sesselmann. A new approach to quickly edit geometries and estimate stresses and displacements of implants in real-time. *Current Directions in Biomedical Engineering*, 5(1):553–556, 2019.

[169] Stratos Idreos, Olga Papaemmanouil, and Surajit Chaudhuri. Overview of data exploration techniques. In Timos Sellis, Susan B. Davidson, and Zack Ives, editors, *Compilation proceedings of the 2015 ACM Symposium on Principles of Database Systems, ACM SIGMOD International Conference on Management of Data, and SIGMOD/PODS 2015 PhD symposium, May 31 - June 4, 2015, Melbourne, VIC, Australia*, pages 277–281, New York, NY, 2015. ACM.

[170] OEIS Foundation Inc. (2020). *The On-Line Encyclopedia of Integer Sequences*, 2020.

[171] A. Donald Keedwell and József Dénes. Chapter 5 - the concept of orthogonality. In A. Donald Keedwell and József Dénes, editors, *Latin Squares and their Applications (Second Edition)*, pages 159–203. North-Holland, Boston, second edition edition, 2015.

[172] IRTG2057. Description of the core research idea and the irtg's resulting main focus. Available at http://www.irtg2057.de/research-program.html.