

# **Machine learning algorithms for efficient process optimisation of variable geometries at the example of fabric forming**

Zur Erlangung des akademischen Grades eines

**Doktors der Ingenieurwissenschaften**

von der KIT-Fakultät für Maschinenbau  
des Karlsruher Instituts für Technologie (KIT)

genehmigte

**Dissertation**

von

**Dipl.-Ing. Clemens Rainer Zimmerling**

geb. in Jena

Tag der mündlichen Prüfung:

24. November 2022

Hauptreferentin:

Prof. Dr.-Ing. Luise Kärger

Korreferent 1:

Prof. Dr.-Ing. Benedikt Kriegesmann

Korreferent 2:

Prof. Dr.-Ing. Frank Henning



# Kurzfassung

Für einen optimalen Betrieb erfordern moderne Produktionssysteme eine sorgfältige Einstellung der eingesetzten Fertigungsprozesse. Physikbasierte Simulationen können die Prozessoptimierung wirksam unterstützen, jedoch sind deren Rechenzeiten oft eine erhebliche Hürde. Eine Möglichkeit, Rechenzeit einzusparen sind surrogate-gestützte Optimierungsverfahren (SBO<sup>1</sup>). Surrogates sind recheneffiziente, datengetriebene Ersatzmodelle, die den Optimierer im Suchraum leiten. Sie verbessern in der Regel die Konvergenz, erweisen sich aber bei veränderlichen Optimierungsaufgaben, etwa häufigen Bauteilanpassungen nach Kundenwunsch, als unhandlich.

Um auch solche variablen Optimierungsaufgaben effizient zu lösen, untersucht die vorliegende Arbeit, wie jüngste Fortschritte im Maschinernen (ML) – im Speziellen bei neuronalen Netzen – bestehende SBO-Techniken ergänzen können. Dabei werden drei Hauptaspekte betrachtet: erstens, ihr Potential als klassisches Surrogate für SBO, zweitens, ihre Eignung zur effiziente Bewertung der Herstellbarkeit neuer Bauteilentwürfe und drittens, ihre Möglichkeiten zur effizienten Prozessoptimierung für variable Bauteilgeometrien. Diese Fragestellungen sind grundsätzlich technologieübergreifend anwendbar und werden in dieser Arbeit am Beispiel der Textilumformung untersucht.

Der erste Teil dieser Arbeit (Kapitel 3) diskutiert die Eignung tiefer neuronaler Netze als Surrogates für SBO. Hierzu werden verschiedene Netzarchitekturen untersucht und mehrere Möglichkeiten verglichen, sie in

---

<sup>1</sup>Surrogate-based Optimisation, engl. surrogate-gestützte Optimierung

ein SBO-Framework einzubinden. Die Ergebnisse weisen ihre Eignung für SBO nach: Für eine feste Beispielgeometrie minimieren alle Varianten erfolgreich und schneller als ein Referenzalgorithmus (genetischer Algorithmus) die Zielfunktion.

Um die Herstellbarkeit variabler Bauteilgeometrien zu bewerten, untersucht Kapitel 4 anschließend, wie Geometrieinformationen in ein Prozess-Surrogate eingebracht werden können. Hierzu werden zwei ML-Ansätze verglichen, ein merkmals- und ein rasterbasierter Ansatz. Der merkmalsbasierte Ansatz scannt ein Bauteil nach einzelnen, prozessrelevanten Geometriemerkmalen, der rasterbasierte Ansatz hingegen interpretiert die Geometrie als Ganzes. Beide Ansätze können das Prozessverhalten grundsätzlich erlernen, allerdings erweist sich der rasterbasierte Ansatz als einfacher übertragbar auf neue Geometrievarianten. Die Ergebnisse zeigen zudem, dass hauptsächlich die Vielfalt und weniger die Menge der Trainingsdaten diese Übertragbarkeit bestimmt.

Abschließend verbindet Kapitel 5 die Surrogate-Techniken für flexible Geometrien mit variablen Prozessparametern, um eine effiziente Prozessoptimierung für variable Bauteile zu erreichen. Hierzu interagiert ein ML-Algorithmus in einer Simulationsumgebung mit generischen Geometriebeispielen und lernt, welche Geometrie, welche Umformparameter erfordert. Nach dem Training ist der Algorithmus in der Lage, auch für nicht-generische Bauteilgeometrien brauchbare Empfehlungen auszugeben. Weiter zeigt sich, dass die Empfehlungen mit ähnlicher Geschwindigkeit wie die klassische SBO zum tatsächlichen Prozessoptimum konvergieren, jedoch kein bauteilspezifisches A-priori-Sampling nötig ist. Einmal trainiert, ist der entwickelte Ansatz damit effizienter.

Insgesamt zeigt diese Arbeit, wie ML-Techniken gegenwärtige SBO-Methoden erweitern und so die Prozess- und Produktoptimierung zu frühen Entwicklungszeitpunkten effizient unterstützen können. Die Ergebnisse der Untersuchungen münden in Folgefragen zur Weiterentwicklung der Methoden, etwa die Integration physikalischer Bilanzgleichungen, um die Modellprognosen physikalisch konsistenter zu machen.

# Abstract

For optimum operation, modern production systems require a careful adjustment of the employed manufacturing processes. Physics-based process simulations can effectively support process optimisation, however, their considerable computation times are often a significant barrier. One option to reduce the computational load is surrogate-based optimisation (SBO). Surrogates are time-efficient, data-driven models which guide the optimisation procedure. They generally help improve convergence, but prove unwieldy when the optimisation task varies, e.g. due to frequent component adaptations for customisation.

In order to also solve such variable optimisation tasks, this work studies how recent advances in machine learning (ML) – especially in neural networks – can enhance and extend current surrogate capabilities. To this end, three main aspects are considered: first, their potential for classical SBO, second, their suitability for efficient manufacturability assessment of new component designs and, third, their options for efficient process optimisation for variable geometries. These aspects apply principally to any manufacturing process and are exemplarily studied at the example of textile forming.

The first part of this work (Chapter 3) examines the use of deep neural networks as surrogate models for SBO. Different network architectures are studied and options for integration into an SBO-framework are compared. Overall, the results show that deep neural networks are a viable option for SBO: For a fix component geometry, all variants successfully minimise the objective function faster than a reference algorithm (genetic algorithm).

In order to assess the manufacturability of variable geometries, Chapter 4 investigates how geometry information can be integrated into a process surrogate. To this end, two ML-approaches are compared: a feature-based and a grid-based approach. The feature-based approach scans a geometry for individual, process-relevant geometry features, whereas the grid-based approach interprets the geometry as a whole. The results show that both approaches can learn the process behaviour, however, the grid-based approach proves more transferable to new geometries. The results further underline that the variety rather than the amount of training data determines this transferability.

Eventually, Chapter 5 combines the surrogate techniques for flexible geometries with variable process parameters to enable an efficient process optimisation for variable geometries. To this end, an ML-algorithm interacts with generic geometry samples in a simulation environment and learns, which geometry requires which forming parameters. After training, it is able to give useful recommendations even for non-generic cases. The results further show that its recommendations converges to the process optimum at comparable speed as classical SBO, yet it does not need an a-priori sampling. Thus, once (pre-)trained, it is more efficient.

Overall, this work shows how ML-techniques can enhance current SBO-methods as the presented methods efficiently support process and product optimisation, especially at early development stages. The work concludes with follow-up questions for further method development, e.g. the integration of physics-based balance equations for improved physical consistency.

# Danksagung

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institutsteil Leichtbau des Instituts für Fahrzeugsystemtechnik (FAST) am KIT. Ohne die Unterstützung zahlreicher Personen und Institutionen hätte sie in dieser Form nicht realisiert werden können. Für die vielfältig erfahrene Hilfe möchte ich mich herzlich bedanken.

Besonderer Dank gilt dabei meiner Betreuerin Prof. Dr.-Ing. Luise Kärger für die intensive Betreuung über die Jahre und die Möglichkeit, das Thema Maschinenlernen in ihre Arbeitsgruppe neu einzubringen. Zahlreiche Anregungen, ihre unermüdliche Ansprechbarkeit bei kleineren und größeren Fragen und ihr hilfreiches Feedback haben mich fachlich und persönlich sehr weitergebracht. Für die Unterstützung, das entgegengebrachte Vertrauen und die thematische Aufgeschlossenheit bin ich sehr dankbar.

Darüber hinaus danke ich Herrn Prof. Dr.-Ing. Benedikt Kriegesmann für die Übernahme des ersten Korreferats und die hilfreichen Anregungen zur Arbeit. Gleichsam danke ich Herrn Prof. Dr.-Ing. Frank Henning für die kontinuierliche Förderung und die Übernahme des zweiten Korreferats. Sehr verbunden bin ich auch Frau Prof. Dr.-Ing. Gisela Lanza für die Übernahme des Prüfungsvorsitzes.

Insgesamt möchte ich mich bei allen KollegInnen am FAST und den Partnerinstituten für die vielfältige Unterstützung, allzeitige Hilfsbereitschaft und die herausragende Zusammenarbeit bedanken. Besonders erwähnen möchte ich dabei Christian Poppe, Benedikt Fengler, Dominik Dörr, Constantin Krauß und Julian Seuffert. Mit ihrer Expertise, ihren Modellen und tiefen Fachdiskussionen haben sie den Kern der Arbeit

substantiell geprägt. Zudem danke ich Sarah Müller und Kevin Daiß, die durch unentwegte Arbeit im Hintergrund zahllose administrative oder IT-technische Hürden aus dem Weg geräumt haben. Ein großes Dankeschön gilt auch allen Studierenden, die meine Forschung mit ihren Abschlussarbeiten unterstützt haben, besonders Daniel Trippe, Patrick Schindler und Christoph Geisendörfer.

Viele der Ergebnisse wurden in Projekten erarbeitet, die mit Mitteln des Ministeriums für Wissenschaft, Forschung und Kunst (MWK) des Landes Baden-Württemberg, der Vector Stiftung, des KIT-Zentrums MATHSEE und des Bundesministeriums für Wirtschaft und Klimaschutz (BMWK) über die Arbeitsgemeinschaft für industrielle Forschung (AiF) und das Forschungskuratorium Textil e.V. gefördert wurden. Für die damit verbundene finanzielle Unterstützung bedanke ich mich herzlich.

Von ganzem Herzen möchte ich mich auch bei meiner Familie und Freunden bedanken. Ihr habt mir die Kraft gegeben, die schwierigen Momente durchzustehen, und die Möglichkeit, die schönen Momente mit Euch zu teilen. Ich bin dankbar und glücklich, dass es Euch gibt. Dies gilt besonders meiner Freundin Fiona. Ich danke Dir für den liebevollen Rückhalt, Deine Geduld und die allseitige Unterstützung in den letzten Jahren.

Karlsruhe im Januar 2023

*Clemens Zimmerling*



# Contents

<b>Kurzfassung</b> . . . . .	<b>i</b>
<b>Abstract</b> . . . . .	<b>iii</b>
<b>Danksagung</b> . . . . .	<b>v</b>
<b>Acronyms and symbols</b> . . . . .	<b>xi</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
<b>2 State of the art</b> . . . . .	<b>5</b>
2.1 Lightweight engineering . . . . .	5
2.1.1 Materials . . . . .	7
2.1.2 Geometry . . . . .	8
2.1.3 Manufacturing . . . . .	10
2.2 Engineering textile forming processes . . . . .	13
2.2.1 Material-process relation . . . . .	13
2.2.2 Manipulation of textile forming processes . . . . .	20
2.2.3 Simulation of textile forming processes . . . . .	23
2.3 Optimisation in process engineering . . . . .	29
2.3.1 Experience-based optimisation . . . . .	30
2.3.2 Algorithmic optimisation . . . . .	31
2.3.3 Surrogates for efficient optimisation . . . . .	36
2.3.4 Machine Learning in Engineering . . . . .	41
2.3.5 Neural networks . . . . .	46
2.4 Literature summary and research gap . . . . .	52

<b>3</b>	<b>Reference optimisation method</b>	<b>55</b>
3.1	Optimisation approach	56
3.1.1	Objective function	56
3.1.2	Optimisation on the surrogate	57
3.1.3	Minimal SBO-example	62
3.1.4	Scalar and field surrogate	63
3.2	Validation of reference SBO-method	67
3.2.1	Validation use-case: Gripper-assisted fabric forming	68
3.2.2	Comparison of scalar and field surrogate	71
3.2.3	Optimisation performance assessment	75
3.3	Intermediate summary	80
<b>4</b>	<b>Formability estimation</b>	<b>83</b>
4.1	Approaches for formability assessment	83
4.1.1	Feature-based formability assessment	84
4.1.2	Grid-based formability assessment	91
4.2	Results of the formability assessment	102
4.2.1	Results of the feature-based approach	103
4.2.2	Results of the grid-based approach	113
4.3	Intermediate summary	123
<b>5</b>	<b>Process optimisation of variable geometries</b>	<b>127</b>
5.1	Reinforcement learning for process optimisation	127
5.2	Validation of RL for process optimisation	136
5.2.1	Validation use case: Pressure-pad assisted fabric forming	136
5.2.2	General training behaviour	143
5.2.3	Hyperparameter study for efficient training	147
5.2.4	Application to the fabric forming use-case	153
5.3	Comparison of optimisation methods	164
5.4	Intermediate summary	172
<b>6</b>	<b>Summary and outlook</b>	<b>175</b>
6.1	Summary	175
6.2	Answers to research hypotheses	179
6.3	Outlook	180

**A Appendix** . . . . . 185

    A.1 Definition of the CNN error metrics . . . . . 185

**List of Figures** . . . . . 187

**List of Tables** . . . . . 191

**Bibliography** . . . . . 193

**List of publications** . . . . . 217

    Journal articles . . . . . 217

    Conference contributions . . . . . 217

    Journal articles (co-authored) . . . . . 219

    Conference contributions (co-authored) . . . . . 219

    Other . . . . . 220



# Acronyms and symbols

## Acronyms

<b>ANN</b>	<u>A</u> r <u>tificial</u> <u>N</u> eural <u>N</u> etwork
<b>BC</b>	<u>B</u> oundary <u>C</u> ondition
<b>CAE</b>	<u>C</u> omputer <u>A</u> ided <u>E</u> ngineering
<b>CAX</b>	<u>C</u> omputer <u>A</u> ided <u>x</u> (arbitrary engineering task)
<b>CNN</b>	<u>C</u> onvolutional <u>N</u> eural <u>N</u> etwork
<b>CoFRP</b>	<u>C</u> ontinuous <u>F</u> ibre <u>r</u> einforced <u>p</u> lastics
<b>DL</b>	<u>D</u> eep <u>L</u> earning
<b>DOE</b>	<u>D</u> esign <u>O</u> f <u>E</u> xperiment
<b>EGO</b>	<u>E</u> fficient <u>G</u> lobal <u>O</u> ptimisation
<b>FE,FEM</b>	<u>F</u> inite <u>E</u> lement, <u>F</u> inite <u>E</u> lement <u>M</u> ethod
<b>FRP</b>	<u>F</u> ibre <u>r</u> einforced <u>p</u> lastics
<b>GA</b>	<u>G</u> enetic <u>A</u> lgorithm
<b>GC</b>	<u>G</u> eometry <u>C</u> lass
<b>GNN</b>	<u>G</u> raph <u>N</u> eural <u>N</u> etwork
<b>GPR</b>	<u>G</u> aussian <u>P</u> rocess <u>R</u> egression
<b>GPU</b>	<u>G</u> raphics <u>P</u> rocessing <u>U</u> nit

<b>LHS</b>	<u>L</u> atin <u>H</u> ypercube <u>S</u> ampling
<b>MAE</b>	<u>M</u> ean <u>A</u> bsolute <u>E</u> rror
<b>MGS</b>	<u>M</u> aterial <u>G</u> uiding <u>S</u> ystem
<b>ML</b>	<u>M</u> achine <u>L</u> earning
<b>MSE</b>	<u>M</u> ean <u>S</u> quared <u>E</u> rror
<b>NN</b>	<u>N</u> eural <u>N</u> etwork
<b>PDE</b>	<u>P</u> artial <u>D</u> ifferential <u>E</u> quation
<b>PINN</b>	<u>P</u> hysics <u>I</u> nformed <u>N</u> eural <u>N</u> etwork
<b>POD</b>	<u>P</u> roper <u>O</u> rthogonal <u>D</u> ecomposition
<b>ReLU</b>	<u>R</u> ectified <u>L</u> inear <u>U</u> nit
<b>RL</b>	<u>R</u> einforcement <u>L</u> earning
<b>RMSE</b>	<u>R</u> oot <u>M</u> ean <u>S</u> quared <u>E</u> rror
<b>RTM</b>	<u>R</u> esin <u>T</u> ransfer <u>M</u> oulding
<b>SBO</b>	<u>S</u> urrogate-based <u>O</u> ptimisation
<b>SGD</b>	<u>S</u> tochastic <u>G</u> radient <u>D</u> escent
<b>UD-NCF</b>	<u>U</u> nidirectional <u>N</u> on <u>C</u> rimp <u>F</u> abric
<b>VPO</b>	<u>V</u> irtual <u>P</u> rocess <u>O</u> ptimisation

**Latin symbols and variables (upper case)**

<i>A</i>	General part quality space (vector-valued)
<i>C</i>	Convolution matrix
<i>C</i>	Metric to assess optimiser convergence

$D$	Data base with input-output observations
$D_v$	Validation data set
$D_t$	Training dataset
$\mathcal{F}$	Surrogate Model Class
$G$	Geometry design space
$G_{12}$	In-plane shear modulus
$I$	Input grid for a convolution operation
$K$	Filter matrix ('Kernel') for a convolution operation
$O$	Output grid of a convolution operation
$\hat{O}$	Output grid after pooling of a convolution operation
$\mathcal{O}$	Landau symbol ('Of the order of...')
$P$	Process parameter space
$Q$	Performance or part quality metric
$S$	Symmetry line
$\mathcal{U}$	Uniform probability distribution

**Latin symbols and variables (lower case)**

$a$	Part quality vector
$c, \tilde{c}$	Optimisation (in-)equality constraint
$c_d$	Coefficient of Determination
$d_{\text{sph}}$	Diameter of the example sphere
$f$	Optimisation objective function

$g_p$	Geometry parameter (width, height, radius etc.)
$g_{dd}$	Double-dome geometry
$\mathbf{g}_p$	Geometry parameter vector
$h$	Height of the corner geometry
$i_{srt}$	Start iteration for actor training
$i_p$	Pixel index (input) during convolution operation
$j_p$	Pixel index (input) during convolution operation
$i_p^*$	Pixel index (output) during convolution operation
$j_p^*$	Pixel index (output) during convolution operation
$l$	layer number in a neural network
$l_a$	Distance of a geometry feature apex to the fabric edge
$l_2$	Geometry width
$l_f$	Fabric length for the considered use-case
$m_p$	Pixel index (kernel) during convolution operation
$m_d$	Dimension of the output space
$m_g$	Number of geometry parameters of a parametric geometry
$n_p$	Pixel index (kernel) during convolution operation
$n_k$	Number of observations
$n_g$	Number of geometries in geometry database
$n_d$	Dimension of the design/input space
$n_l$	Number of neurons in layer $l$ in a neural network
$n_L$	Number layers in a neural network



$n_{\text{ep}}$	Number of epochs during training
$n_{\text{rffn}}$	Number of refinement epochs for neural network refinement
$n_{\text{el}}$	Number of elements in a process simulation
$\mathbf{p}^*$	(Process) Parameter optimum
$\mathbf{p}_{\text{srg}}^*$	(Process) Parameter optimum according to a surrogate
$\mathbf{p}$	(Process) Parameter
$q$	Scalar part quality indicator
$q_{\text{srg}}$	Surrogate approximation of the scalar part quality indicator
$q_{\text{xmp}}$	Generic part quality indicator
$q$	Scalar part quality indicator
$r_{\text{p}}$	PEARSON correlation coefficient
$r_1, r_2$	Top and bottom radius of the corner geometry
$s, s_{\text{min}}$	(Minimum) distance of two geometry features for independence
$u$	Vector norm exponent
$w_1$	Geometry width
$w_{\text{f}}$	Fabric width for the considered use-case
$\mathbf{x}_{\text{nr}}$	Input vector of a neuron
$\mathbf{y}_l$	Output vector of layer $l$ in a neural network
$y_{\text{nr}}^l$	Output scalar of a neuron in layer $l$ in a neural network
$\tilde{y}_{\text{nr}}$	Input to a neuron's activation function in a neural network
$\mathbf{y}_l$	Output vector of layer $l$ in a neural network

### **Greek symbols and variables (upper case)**

$\Delta$	Absolute or relative difference between two magnitudes
$\Xi$	Region around an optimum

### **Greek symbols and variables (lower case)**

$\alpha$	Draft angle of the corner geometry
$\gamma_{12}$	In-plane shear angle
$\gamma_{12}$	Set of shear angles
$\gamma_{12}^{\max}$	Maximum shear angle
$\epsilon$	Exploration probability
$\epsilon_{0,1}$	Exploration probability at training begin and end, respectively
$\epsilon_d$	Exploration probability decay factor
$\epsilon$	Error metric of a surrogate or ML-model
$\epsilon_{\text{trn}}$	Error metric/Loss of a surrogate or ML-model on training data
$\epsilon_{\text{val}}$	Error metric/Loss of a surrogate or ML-model on validation data
$\zeta$	Material orientation angle
$\eta$	Learning rate
$\theta$	Surrogate or Machine Learning model parameters
$\theta$	Geometry template for feature-based geometry recognition
$\kappa$	Fabric curvature

$\mu_{\text{aux}}$	Auxiliary model for actor-critic-training
$\mu_f$	Model to estimate part quality field for a given geometry
$\mu_p$	Model to estimate optimal parameters for variable geometries
$\varphi$	Process simulation model
$\psi$	Neural activation function
$\psi_{\text{relu}}$	ReLU activation function
$\omega_g$	Geometry definition exponent

### **Operators and math symbols**

$a$	scalar quantity (lower case, non-bold typeface)
$\mathbf{a}$	vector quantity (lower case, bold typeface)
$A$	space quantity, e.g. parameter space (upper case, non-bold typeface)
$\mathbf{A}$	matrix quantity (upper case, bold typeface)



# 1 Introduction

Everywhere an early foreboding precedes the eventual knowledge.<sup>1</sup>

ALEXANDER VON HUMBOLDT [1, p 493]

Most modern production systems are complex systems and require a careful optimisation during production ramp-up. In current practice, this often involves resource-intensive trial-error campaigns combined with expert-judgment and experience from prior parts. However, such empirical approaches often entail significant rework for error correction. This holds in particular for complex processes with intricate behaviour like composite materials for lightweight applications.

At the same time, the continuing trend to smaller lot sizes and increasing product diversity ('mass customisation') calls for ever shorter development cycles and flexible manufacturing technologies. Accordingly, adaptive and customisable production principles have emerged, e.g. additive manufacturing. However, besides their sheer availability, such flexible technologies also require efficient approaches for process optimisation to minimise production downtime.

---

<sup>1</sup>Original in German: Überall geht ein frühes Ahnen dem späteren Wissen voraus.

For instance, SHAMSAEI et al. state that recurring optimisation tasks for ever-changing geometries or materials, respectively, are a significant economical barrier [2]. They call for a comprehensive framework to ‘leverage information from prior similar studies and . . . systematically characterize the relation between process parameters and part features so that the . . . process can be optimized in a more efficient manner.’

This holds all the more when processing delicate materials, such as textiles used for continuous-fibre reinforced plastics (CoFRP). They are usually processed in elaborate, multi-step processes and most often comprise a forming process of a textile. The wide range of adjustable process parameters and the complex, non-linear material behaviour require place high demands on a suitable process configuration and pose a challenging development task.

To reduce the cost of an experimental process development, numerical simulations have gained attention over the last decades. They allow for detailed analyses of complex processes and help concentrate costly experiments on the most promising variants. Also, their inherently digital nature allows a combination with optimisation algorithms which *principally* enables an automatic identification of process optima. However, they usually involve significant computational efforts and especially repetitive simulations, e.g. iterative optimisation, quickly renders them impracticable in practice.

One option to reduce the numerical effort in such cases is surrogate-based optimisation (SBO). Surrogates are numerically efficient, data-driven approximations of expensive simulations based on input-output-observations. Once sufficiently trained, optimisation can be done on the surrogate in short time. Overall, SBO results in significant optimisation speed-ups. However, current SBO-approaches are mostly application-specific and fall short on reusability in new scenarios. Even subtle problem variations, e.g. geometry variations in manufacturing, instantly invalidate the surrogate and require resampling of data and reconstructing the surrogate. Thus, demand for generalised models arose.

At the same time, developments in Machine Learning (ML) have achieved remarkable results in complex tasks and may open up new avenues for advanced surrogates. This work aims to study how ML-techniques can be harnessed to enhance the capabilities of classical SBO-strategies in order to solve flexible optimisation tasks more efficiently. Or, loosely speaking, how to equip surrogate models with an early foreboding, where the process optimum for a new component may hide.

The overarching concept is to sample process observations for a range of generic geometries and analyse it with ML-techniques. Recurring patterns in the data may then guide a process optimisation of a new component. If possible, such a model could provide an early estimation of process optima for new components. Owing to their reconfigurability and ease of evaluation, physics-based numerical process simulations are used for data sampling.

Specifically, this work studies two main hypotheses:

**Hypothesis 1.** It is possible with ML-methods to extract process knowledge from generic part-process-samples and apply it to a new geometry.

**Hypothesis 2.** Once trained, such an ML-model speeds up the optimisation similar to a classical surrogate.

While these hypotheses are principally applicable to any manufacturing process, they are exemplarily studied at the example of textile forming.

## Thesis outline

This work is organised as follows: At first Chapter 2 presents the state of the art in two main areas: It reviews *technological* aspects regarding lightweight engineering and especially textile forming (Sections 2.1–2.2) and then Section 2.3 presents relevant *methodological* aspects on optimisation, surrogate modeling and ML in engineering.

The subsequent chapters then present and discuss the methods of this work. Specifically, Chapter 3 outlines a state-of-the-art SBO-method for process optimisation. It is meant to serve as a benchmark for the new, ML-based optimisation method and discusses the optimisation results. Then, Chapters 4–5 present approaches to a more flexible ML-based surrogate. Specifically, Chapter 4 presents and compares two approaches for an ML-assisted formability assessment of variable component geometries. In Chapter 5, one approach is further developed towards efficient process optimisation of variable geometries. The results are presented, discussed and compared to the state-of-the-art method from Chapter 3. The thesis concludes with a summary and an outlook on possible next steps in Chapter 6.



## 2 State of the art

Based on the Introduction in Chapter 1, this work sees engineering design as an optimal match of three technological areas: Material, process and geometry. Achieving this match requires background information on *technical* and *methodological* aspects.

To this end, Section 2.1 at first outlines *technical* characteristics of relevant engineering areas and motivates the studied use-case of textile forming. Then, Section 2.2 provides an overview over textile forming processes, i.e. forming mechanisms and defect formation, technologies for defect control and numerical models for efficient (virtual) process development.

As the process shall be optimised, Section 2.3 reviews existing *methods* for (process) optimisation with particular focus on surrogate-based strategies. Shortcomings of current approaches are identified and discussed. The section concludes with concepts, how machine learning techniques can enhance current surrogates. They are the basis of this work. Eventually, Section 2.4 summarises the literature review and distils the research gaps.

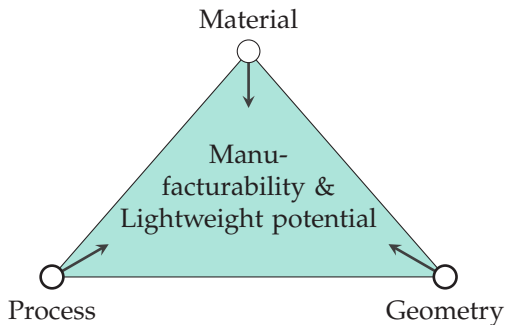
### 2.1 Lightweight engineering

Within mechanical engineering, lightweight design is a special design philosophy. It goes beyond a mere, retroactive parameter tuning of a given structure but embodies a holistic design strategy right from the beginning: It means the materialisation of technical functions at minimal mass utilisation while fulfilling a set of constraints. These constraints

can be of various nature in general but often originate from technical, economic, environmental and societal requirements. [3]

In general, five overarching areas govern the lightweight potential of structures [4, p 62-63], namely its 1) specifications and requirements, 2) concept, 3) material, 4) shape and 5) manufacturing process technology. In many cases, area 1 and 2 cannot be further exploited as requirements have already been reduced to the minimum and the component must conform to a certain higher-order concept. This leaves *material*, *shape* and *process* as the remaining options.

ISO/TR 581 finds the same areas to govern *manufacturability* [5]. Figure 2.1 visualises their interrelations by a triangle: The farther a structure lies from one of the corners, the less the associated factor is satisfied. Consequently, a well-manufacturable and lightweight structure lies in the centre where the three fields match in mutual regard.



**Figure 2.1:** Component geometry, material and manufacturing process technology govern part quality (manufacturability) and lightweight potential [5, 4].

Bringing a structure to the centre is a challenging engineering task, though, and requires an understanding of *material*, *shape* and *process*. Thus, their core aspects are briefly outlined in the following subsections.

### 2.1.1 Materials

In many modern applications materials must satisfy an ever more demanding requirement spectrum and often conventional monolithic materials cannot meet each demand to a satisfactory degree. Thus, materials have been deliberately combined to fuse their properties and improve them as a whole – the core concept of composite materials.

Composite materials consist of at least two mutually insoluble, solid phases, the constituents. Their individual properties, shape, volumetric proportions and interface characteristics determine new, quasi-homogeneous properties on macroscopic scale: the composite properties [4, p 341]. Structural composites often have a load-transferring reinforcement phase, which is embedded in a surrounding matrix phase. Depending on the reinforcement geometry, they can be grouped into three categories: *Particle-*, *fibre-* and *layer-*reinforcements [6, p 303].

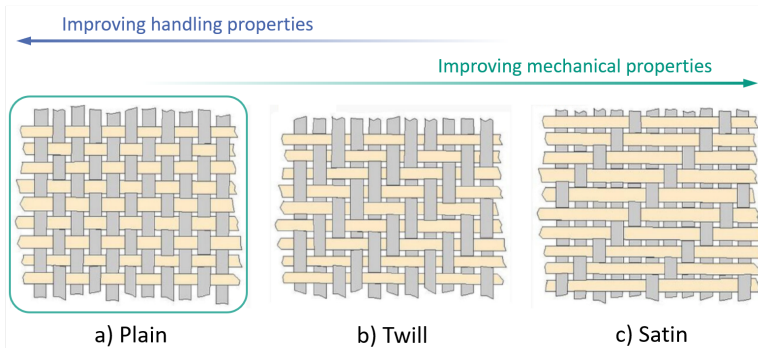
For high-performance applications, *fibre-*reinforcements have drawn most attention as they offer excellent weight-specific mechanical properties. Also, their properties can be adjusted to a wide extent, for instance by choice of fibre and matrix material. Generally, different materials for fibre and matrix are at disposal, however, carbon, glass and aramid fibres combined with polymer-based matrices have established in practice, i.e. ‘fibre-reinforced plastics’ (FRP). [4, p 342]

FRPs can use *continuous* and *discontinuous* fibres. While discontinuous fibres are shorter than the component dimensions and show random or mild orientation only, continuous fibres are generally highly oriented and extend over the whole component. Thus, continuous-fibre reinforced plastics (CoFRP) show the highest mechanical performance and are also the scope of this work.

Continuous fibres are not handled individually but typically come in the form of textiles. Textiles consist of interlaced ‘rovings’, i.e. bundles of individual fibres, and form highly-compliant, macroscopically continuous and homogeneous entities. This work concentrates on one of the most

common engineering textiles, *woven fabrics*. See [7] for a comprehensive review of technical textiles for lightweight engineering.

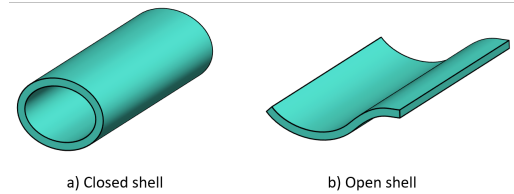
While different sub-variants exist (cf. [7, p 201-213]), conventional woven fabrics consist of at least two yarn systems that interlace at right angles: Warp yarns run in the direction of production, weft yarns orthogonal. Depending on the interlace-sequence, three main types of weaves are distinguished: *plain* weave, *twill* weave and *satin* weave. Figure 2.2 visualises them. They mainly differ in their handleability during manufacture and their resulting mechanical and visual properties. As it is most widespread [7, p 175], the plain weave is studied in this work.



**Figure 2.2:** Common fabric weave patterns used in CoFRP-components [8]. The weave pattern affects the handling and the mechanical properties of the fabric.

### 2.1.2 Geometry

Most structural CoFRP-components are 2.5D shell structures. They are the best-suited for distributed loads, e.g. static or dynamic fluid pressures, or when functional or aesthetic requirements call for a closed component [9, p 35-36]. According to POTTER [10], shell structures can be divided into *closed* and *open* shells as Figure 2.3 illustrates.



**Figure 2.3:** Visualisation of a) closed and b) open shell structures as used in this work.

Closed CoFRP-shells dominate in tanks, vessels and pipes for fluid transportation or storage and machine elements like shafts. Most of the closed surface areas are rather simple and well understood geometries, e.g. cylinders. They are manufactured by dedicated processes like winding or pultrusion and as such they are often of limited design variability [9, p 68-70].

Open shell structures, however, are often of tray-like nature and typically manufactured in press-processes. Compared to closed shells, they are generally more complex and variable in shape as they must often satisfy challenging functional, structural and aesthetic requirements [10]. Thus, even components with comparable specifications come in a wide range of designs and are additionally often subject of change during product revisions, e.g. for aesthetic reasons ('face-lift'). Figure 2.4 visualises this at example of the VOLKSWAGEN GOLF front mudguard (Mk4 to Mk6).

The designs clearly vary in details, e.g. position and width of attachment points, but the overall L-shaped characteristic stays the same across all design revisions. Since they would have been discarded otherwise, they must bear features with useful properties, e.g. good trade-off between structural behaviour and manufacturability.

Harnessing such favourable part-process-properties should provide useful information for efficient design and optimisation of new components



**Figure 2.4:** Recurring component characteristics during design revisions at the example of the VOLKSWAGEN GOLF's front mudguard (Mk4-6). While the mudguard retains its overall engineering design it varies slightly in shape. Images from [11, 12].

[2]. Unfortunately, such information is currently at best implicitly respected in form of human-subjective 'engineering experience', but currently not integrated in formal optimisation routines. Consequently, algorithmic optimisation requires many iterations until a satisfactory design is reached – a substantial bottleneck during development.

### 2.1.3 Manufacturing

This work studies open-shell structures of variable shapes and aims to extract knowledge from prior part-process examples so that a new part can be optimised more efficiently. This requires not just information on typical geometries but also on process specificities. They will be presented in the following.

**Process chain.** Unlike metals, composite manufacturing processes typically comprise a sequence of process steps, the 'process chain', with different dominating physics. Figure 2.5 visualises this at the example of resin transfer moulding (RTM), an automatised and widely applied process across many industries.

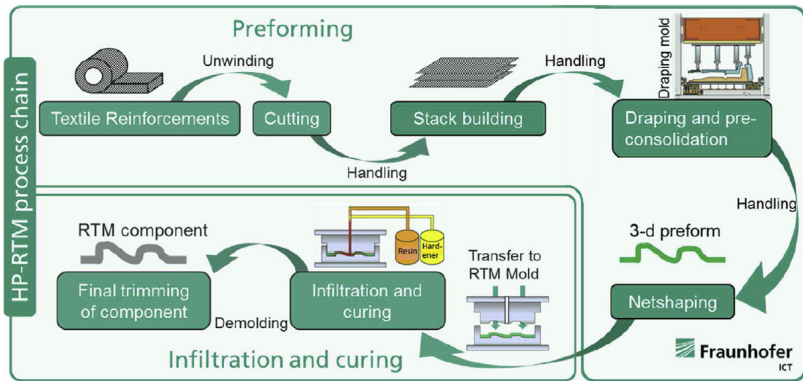


Figure 2.5: Schematic illustration of the Resin Transfer Moulding (RTM) process chain [3].

Initially, individual 2D-plyes are cut and stacked, which determines the later layup and fibre orientations. Then, the stack is transferred to a press tool. During tool closure the whole stack is formed to a 3D preform. Typically, an adhesive binder material activates to stabilise the preform for subsequent demoulding and handling to an infiltration tool. Then, a mix of resin, hardener and release agent is injected into the preform and the curing reaction starts typically at elevated temperatures. Reaction kinetics, injection location and conditions (pressure or flow rate) as well as temperature of resin and tool must be engineered so as to ensure both a complete mould filling before gelling and sufficient degree of cure for demoulding. Eventually, the part is demoulded for possible finishing operations. [3]

Due to the chain characteristic, results from upstream steps affect downstream steps. For instance, forming defects may lead to non-infiltratable regions during mould filling. Besides an adverse visual appearance, these spots can only partially take up loads and are thus weak points of the component [3, 13]. An improved forming process may remove these defects and enable satisfactory filling.

While all steps along the process chain require careful engineering and optimisation, this work focusses on the *forming* step for two reasons.

First, it is the first step after cutting and does thus not depend on upstream information. Second, the process dynamics depends significantly on the component geometry and thus, the forming step is a prime demonstrator for extraction of knowledge from prior process examples and application to a new component.

**Material forming.** Forming processes are typically grouped according to the dominant process-inherent material loads [14]. Within these, forming under *tensile and compressive* conditions, most notably *deep drawing*, is most widespread for tray-like 2.5D-structures.

Some niche-variants aside [15], deep drawing processes are generally characterised by an initially flat material sheet, the ‘blank’, two rigid tools – a male punch and a female die – and in most cases a circumferential blank holders. Figure 2.6 visualises the process principle.

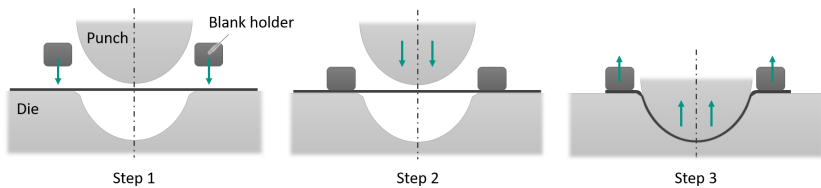


Figure 2.6: Schematic illustration of the main steps during deep drawing.

At first, the blank holder clamps the blank onto the forming die with a predefined force (Step 1). As the punch travels into the die (Step 2) it at first tensions the blank between punch and blank holder which evokes in-plane membrane stresses. They rise until the resulting membrane forces eventually exceed the clamping forces. At this point the material starts to flow into the mould from the outer flange and forms the component. Unlike stretch drawing, in deep drawing the sheet thickness is not meant to reduce but stays approximately constant. Depending on the material and geometry, the process needs to be adjusted by variation of



the tool speed and the blank holder forces and possibly their local variation. After the stroke, punch and blank holders retract for demoulding (Step 3). [16, p 263]

DIN 8580 notes that process descriptions are written for metals but can be transferred to other materials [17] as long as the main forming-conditions are comparable. As this work studies deep drawing of dry *woven fabrics*, some textile-specific remarks are necessary and will be outlined in the following.

## **2.2 Engineering textile forming processes**

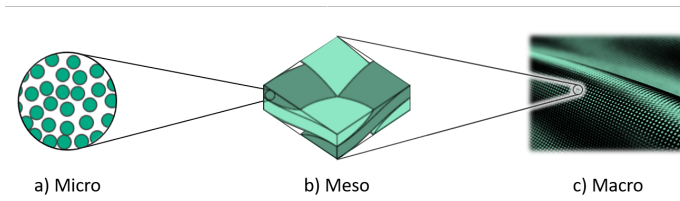
Section 2.1 shows that engineering of a well-manufacturable component reconciles material, geometry and process requirements, and outlines textile forming as the use-case of this work. Since it differs substantially from metal forming, this section presents at first material-process-interrelations, then briefly reviews technologies for process improvement and concludes with a short introduction to numerical methods for virtual process analysis as used in this work.

### **2.2.1 Material-process relation**

On a microscopic scale, textiles and metals have different forming mechanisms which lead to a disparate forming behaviour. In general, metals consist of grains with a crystal lattice of constant orientation. Under sufficient shear stress, the atoms glide from one lattice location to the next resulting in a permanent, plastic deformation. As this occurs almost exclusively in distinct planes and orientations, so-called 'gliding systems', each grain on its own is highly anisotropic. On a macroscopic scale

however, their sheer number provides such a large multitude of gliding systems that they become virtually isotropic<sup>1</sup>. [18, p 401-412]

**Textile forming behaviour.** Compared to metals, engineering textiles show a substantially more complex macroscopic behaviour which largely depends on the textile architecture. Engineering textiles for structural applications typically employ fibres of high stiffness. As a result, their deformation mechanisms are governed by relative movement of quasi-inextensible fibres constrained by the textile architecture [19, p 20]. Owing to the heterogeneous, multi-scale nature of textiles, deformation can take place on three different scales, namely *micro*-, *meso*- and *macro*-scale [20]. Figure 2.7 illustrates the scales schematically.



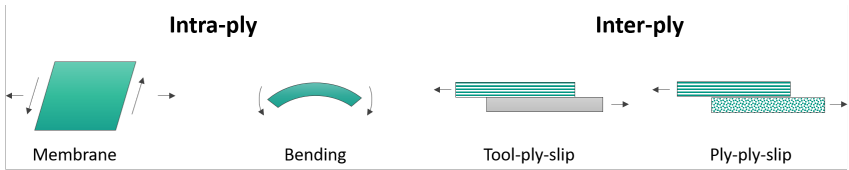
**Figure 2.7:** Schematic illustration of *micro*-, *meso*- and *macro*-scale for textiles. Image c) from [21] (modified).

*Micro*-scale deformations typically refer to individual fibre filaments within one roving. Except for (mostly academic) characterisation purposes, e.g. fibre-fibre-interactions [22, 23], *micro*-scale deformations are not explicitly considered when designing forming processes. In contrast, *meso*-scale deformations abstract from individual fibres but consider textile rovings as continuous entities which deform and move relative to each other. This scale can be of practical relevance and addresses effects like yarn gapping, slippage, loop formation, yarn pull-out [24]. Although the deformation is *meso*-scale, it affects also the *macro*-scale

<sup>1</sup>Upstream rolling may orient grains and introduces some anisotropy, though [16, p 66].

properties: Adverse visual appearance aside, these effects may lead to locally inhomogeneous composite properties in the final part [25, 26]. For instance, a local absence of fibres from gapping may lead to resin-rich spots and reduced stiffness and strength. Similarly, a misalignment of reinforcing fibres reduces composite strength and stiffness and has thus been studied early on [27].

Ultimately, *macro*-scale deformations abstract even from individual rovings and assume the whole textile as a homogenous 2D-continuum. They are usually categorised as *intra-ply* and *inter-ply* mechanisms [28, 29] as Figure 2.8 visualises.



**Figure 2.8:** Schematic illustration of the main macroscopic deformation modes of engineering textiles: Intra-ply and inter-ply deformation [28, 29].

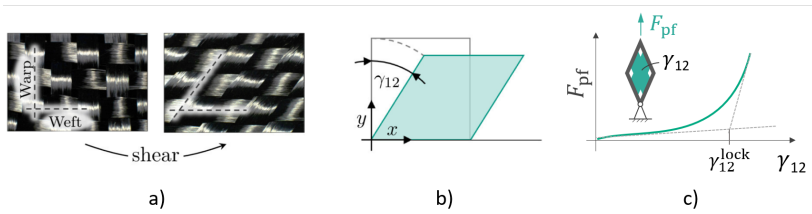
The intra-ply mechanisms comprise all compatible in-plane (tensile and shear deformation) and out-of-plane deformation modes (bending). Textiles also show some compressibility in thickness direction (‘compaction’). Apart from special cases like spacer fabrics [7, p 215-217], this deformation role is negligible for achieving a component shape and thus excluded from Figure 2.8. However, it substantially affects secondary process aspects, e.g. tightness in tool sealing in wet compression moulding through circumferential fibre clamping [30]. The inter-ply mechanisms comprise the tangential and normal contact behaviour between tool and ply as well as between two plies, respectively [29].

The yarns and the textile architecture govern the admissible deformation modes. A comprehensive review goes beyond the scope of this work but can be found in e.g. [7]. Thus, only the macro-scale deformation mechanisms of *woven fabrics* are outlined in the following.

**Forming mechanisms of woven fabrics.** The dominant deformation mechanisms of woven fabrics are out-of-plane *bending* and in-plane *shear*.

Components with single curvature can be achieved by pure bending, while a combination of bending and shear is required for double curvature components [19, p 20]. Since forming defects from pure bending, e.g. fibre breakage, requires extremely small bending radii [31], it typically does not limit the forming process in practical applications. Yet, unlike pure bending, shear deformation does introduce a limit to textile formability and thus requires deeper introduction.

Shear deformation of woven fabrics is characterised by a relative rotation of warp and weft yarn (pure shear) and is typically quantified by the shear angle  $\gamma_{12}$ , cf. Figure 2.9 a) and b). Since warp and weft yarn interlace, they cannot pass each other but increasingly press against each other. This leads to progressive in-plane yarn compaction and an accumulation of compressive stresses.



**Figure 2.9:** a) Photograph of a woven fabric undergoing shear deformation, b) definition of the pure shear kinematic and c) qualitative plot of the non-linear shear force  $F_{pf}$  against the shear angle  $\gamma_{12}$  for a woven fabric along in a picture frame test.

These stresses oppose further shear deformation and result in a non-linear increase of the shear modulus  $G_{12}$  – often referred to as ‘shear locking’. The shear angle  $\gamma_{12}^{lock}$  at which locking occurs depends on multiple textile parameters such as the roving size and the weave. It is usually measured in mechanical characterisation tests, e.g. a picture frame test [32], and manifests as a non-linear rise of the shear force, cf. Figure 2.9 c). Shear near  $\gamma_{12}^{lock}$  in combination with low bending stiffness

increases the likelihood of defects like wrinkling and textile folding [33]. However, the more  $\gamma_{12}$  exceeds the locking angle  $\gamma_{12}^{\text{lock}}$ , the more it loses expressiveness, since forming defects such as wrinkling may continue to grow although  $\gamma_{12}^{\text{max}}$  remains practically constant. Hence, a direct measure for the extent of wrinkling is desirable.

While the shear angle is a well-defined quantity in continuum mechanics, different approaches to quantify wrinkling exist in literature. For instance, SHEN et al. [34] propose to evaluate the ‘wrinkling amplitude’, i.e. out-of-plane deviations of the textile with respect to an ideally formed reference surface. Later VIISAINEN [35, p 47] suggests to also consider the ‘wrinkle spread’, i.e. the ratio of area where these deviations exceed a threshold-amplitude to the total textile area. As it does not require defining a reference surface, this work applies the modified mean Gaussian curvature  $\kappa$  proposed by HAANAPPEL [28] besides the shear angle.

In the present work shear and wrinkling serve to quantify the part quality and thus additional details on the underlying defect mechanisms are presented in the subsequent sections.

**Defect formation in fabric forming.** Irrespective of the material, wrinkling is a well-known phenomenon in sheet forming, yet still difficult to predict. It refers to sudden and unintended out-of-plane deformations of the sheet during forming. Figure 2.10 shows severe wavy wrinkling-effects around a tetrahedral pyramid. Such instabilities (buckling) can occur under compressive stresses in combination with low bending resistance. Owing to their low thickness, sheets offer little bending resistance and are thus prone to such buckling effects.

Textiles are especially at risk of buckling because they have a dwindling bending stiffness compared to metal sheets of the same thickness [37]. This is because the textile consists of numerous individual filaments which slip relative to each other and cannot take up transverse shear strain. Consequently, the KIRCHHOFF-LOVE-hypothesis – plane and normal cross sections remain straight and normal during deformation – no



**Figure 2.10:** Photograph of large scale textile wrinkling after forming [36]

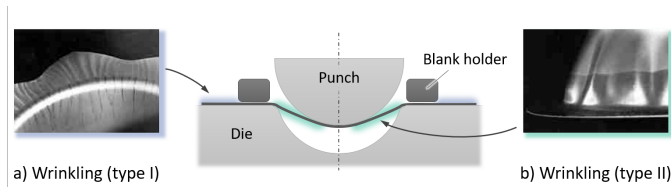
longer holds. For instance, the bending stiffness reported in [38] is only  $\approx 4\%$  compared to a shear-stiff KIRCHHOFF-LOVE beam of the same tensile stiffness.

The onset and growth of wrinkles in fabrics is caused by a complex interaction of in-plane stresses, in-plane stiffness, out-of-plane bending stiffness [39, 40] and other process conditions, e.g. stacking sequence [41]. Unfortunately, many of these factors are unknown at early stages of process design and may even change during component development.

Thus, design engineers pragmatically simplify the onset of defects to criteria which are easy to evaluate and straightforward to interpret. Although strictly speaking many times debunked as insufficient [36], one such criterion is comparing the present shear angle  $\gamma_{12}$  in the component to the locking angle  $\gamma_{12}^{\text{lock}}$ . If  $\gamma_{12}$  exceeds  $\gamma_{12}^{\text{lock}}$ , in-plane compressive stresses increase drastically which is assumed to provoke wrinkles and needs to be avoided.

Besides, excessive shear also impedes the resin permeability during infiltration. For instance, ENDRUWEIT et al. [42] report a permeability reduction of up to 60% solely from shear deformation. Such a reduction at best only prolongs infiltration time, at worst prevents infiltration at all and results in dry spots. In any case, the formation of according defects during the process needs to be analysed and ideally controlled by engineering measures.

**Process engineering for defect control.** Despite being one of the most prominent forming defects in both metal and textile forming [16, p 292], wrinkling still proves difficult to control. According to its location, two types of wrinkling are distinguished, type I and II [16, p 292], as Figure 2.11 illustrates.



**Figure 2.11:** Location of wrinkles in deep drawing [16, p 291]. Type I wrinkling occurs at the outer flange (blue halo), type II in the free forming zone (green).

Type I wrinkles form at the blank flange due to tangential compressive stresses. If adjusted properly, the blank holder prevents type I wrinkling. In contrast, type II forms in the free forming zones between the bottom of the punch and the die edge. These zones are 'free' since the sheet is (temporarily) not in contact with tool surfaces during the punch stroke. Thus, it is not supported by the tool walls and can deform freely in the drawing gap. Wrinkling type II is particularly difficult to control because the free zone is inaccessible during the tool stroke.

Large radii of punch and die [16, p 292] as well as tapered geometries [43, p 383] promote type II wrinkling since they enlarge the free zone. Thus, geometry adaptations can effectively prevent wrinkling. Yet, in some cases the geometry follows functional or structural requirements (cf. Section 2.1.2) and adaptations are undesirable. In such cases, external process manipulations are applied to mitigate wrinkling. To this end different approaches have emerged as outlined in the following.

## 2.2.2 Manipulation of textile forming processes

As in any other engineering design problem, design variables in textile forming processes can be grouped into two types, *categorical* and *continuous* variables [44]. Categorical parameters often refer to choice of technology or material, while continuous parameters mostly refer to a scalable, numerical representation of a process configuration, e.g. blank holder position. During process development, at first categorical parameters are selected from domain knowledge, then according continuous variables are tuned towards maximal quality ('process optimisation').

The following passage provides a short survey on different technological concepts (categorical variables) to improve the forming behaviour. In *principle*, geometry, material or process variations can be considered (Figure 2.1). However, besides minor adjustments of draft angles and fillet radii, geometry adaptations are often undesirable and seen as a last resort (cf. Section 2.2.1). Thus, the presented variations concentrate on *material* and *technology* aspects.

**Material.** One approach to improve the forming process is to alter the local formability of the material. The most fundamental option is to select a textile with better formability, e.g. a knitted instead of a woven fabric [45]. A less fundamental change is to improve the formability by *cut-ins*, *splicing* or *darting* of the fabric [46, 47]. Such local cuts enable the fabric to move freely or glide one above the other to relieve compressive stresses without folding. On the downside such cuts are a local interruption of the reinforcement [3].

Similarly, additional 'strain reserves' have been introduced by stitching roving slings onto a highly deformable carrier textile [48]. During forming, the slings first unfold before they begin to stretch, which artificially enhances the strain limits. The opposite mechanism, i.e. locally restraining the textile formability, can also be used to control forming, e.g. by stitching [49] or gluing sheets together [50].

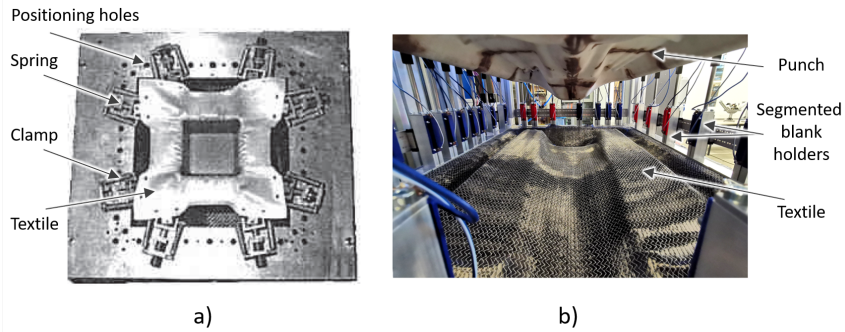


**Technology.** The second and arguably more versatile approach to control the forming process aims at the technological concept. Reverse pre-shearing of the fabric has been used [51] to introduce an additional ‘shear strain reserve’ before locking. High shear angles can also be mitigated by variation of the draping origin [52, 53] which is often done in manual hand layup [54] and has recently also been realised in a mechanised tool [55]. Leading pins in the tool can also avoid excessive sag of the textile and reduce uncontrolled forming behaviour in unsupported areas (type II wrinkling) [56]. Especially for complex geometries, forming the component sequentially with a segmented tool [57] or robot-assisted draping along optimal trajectories [55] can improve the result effectively.

Presumably the largest portion of process variation technologies, however, concentrate on control of the material flow into the mould, so-called material guiding systems (MGS). NEZAMI [19, p 45] divides MGS into *direct* and *indirect* principles. Direct principles act directly in the textile plane, while indirect principles require an intermediate physical effect to actuate an in-plane deformation mechanism.

Most direct principles involve a mechanism to clamp the textile around its perimeter. An early example of such an MGS are pivoting, spring-guided clamps mounted on a rigid frame which surrounds the blank [58], cf. Figure 2.12 a). The process can be varied by choice of spring stiffness, spring pre-tensioning, number and position of the clamps and a mechanical stop to limit the clamp travel. A similar clamping-frame is presented by LUEBBERING and LENGSELD [59]. It does, however, involve a whole array of clamps in favour of isolated clamps. LIEBAU et al. [60] vary the principle and use rope-guided weights instead of springs to restrain the clamps which ensures a constant restraining force during forming .

Indirect principles require an intermediate physical effect to manipulate the textile deformation mechanism. While the textile is ultimately actuated by friction, the technological principles to produce the normal forces vary to a large extent. Most similar to draw beads from sheet metal forming, clamping frames in diverse variations have been proposed; for



**Figure 2.12:** Examples for direct and indirect manipulation approaches for textile forming; a) discrete clamping [58] (direct), b) segmented blank holders (indirect) [61]

instance, by shape-variation of longitudinal grooves [62] or adjustable spring-suspensions [63, 64] or segmentation [65]. Recently, segmented blank holders with adjustable forces have been introduced for precise process manipulation, cf. Figure 2.12 b).

CHEN et al. [66] present a new type of bead (‘risers’) in double diaphragm forming. In double diaphragm forming the textile is sandwiched between two highly deformable diaphragms and maintains contact to them during forming. Unlike before, the beads do not directly act on the textile but impose tensile strain on the diaphragm which in turn strains the textile. Their position around the textile perimeter, length, height and contour can be adjusted to optimise the forming result.

Besides beads, drapery wings can be applied [67]. Contrasting classical beads, they not just clamp the textile but actively pretension it. Ultimately, vacuum grippers [68], pneumatic expansion tubes [69] or magnets [70, 71] have been suggested as textile grippers. More recently, SCHÖFER [72] proposed a detachable textile joint, a tuft seam, which exerts a defined retention force when detaching.

Overall a large variety of technologies exists to manipulate forming processes. However, each technology introduces adjustable parameters

which need to be optimised for each manufactured geometry and material. Experimental optimisation campaigns are resource-intensive and must be as purposeful as possible. Virtual experiments using simulations can help reduce the number of experimental trials and will thus be presented in the following.

### 2.2.3 Simulation of textile forming processes

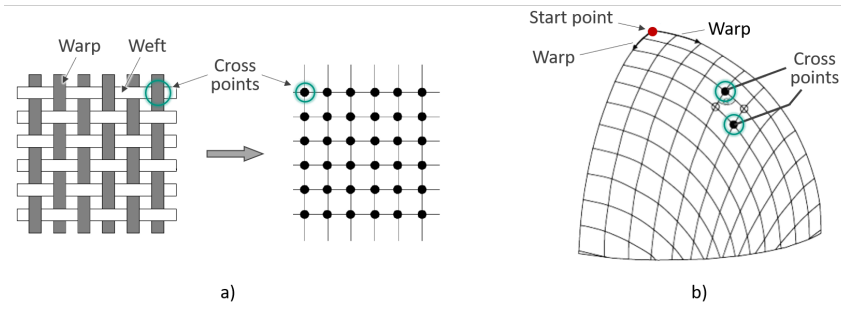
With the advent of ever more powerful computational resources and advances in software, virtual tools have established in engineering development. The collective abbreviation 'CAE' encompasses all computer-aided engineering technologies that support ideation, conceptualisation and analysis of engineering systems [73].

CAE-tools become especially handy in lack of suitable design guidelines or when only little experience from prior products is available. In particular the complex, anisotropic and non-linear behaviour of CoFRP-components often leads to unexpected and counterintuitive effects which in turn impedes an empirical assessment. Such cases require more rigorous computational means, i.e. simulations.

Simulating textile forming poses several challenges to accurately capture the intra-ply and inter-ply mechanisms. In following, a brief and schematic introduction to the simulation approaches for woven fabrics is given. Details on the models, implementation and experimental characterisation can be found in the according literature as recently reviewed in [20, 74].

According to LIM and RAMAKRISHNA [75], two types of forming simulation exist, *kinematic* and *mechanical* approaches. As both are used in this work, they are shortly introduced in the following.

**Kinematic draping.** ‘Kinematic’ draping approaches<sup>1</sup> are entirely geometrically motivated and require only the component shape, the sheet dimensions, the draping start point, the initial fibre orientation and the type of fabric (woven or unidirectional) as input. Figure 2.13 illustrates the concept. The fabric is modelled as a regular grid of rigid trusses connected by ideal spherical joints (crosspoints). From the start point, this grid (fabric) is mapped onto the tool surface in a step-wise manner.



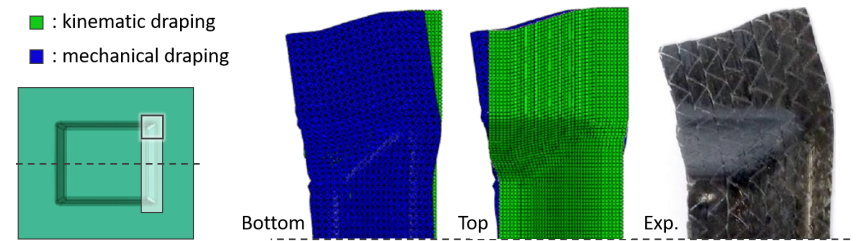
**Figure 2.13:** Visualisation of kinematic draping: a) Modelling of a woven fabric as a regular grid [47] and b) Mapping result onto a doubly-curved surface [7, p 597].

The underlying, fundamental assumption is that the fabric can only deform by bending and pure shear [76] with inextensible warp and weft yarns. In essence, the approach yields the forming result of an idealised textile. For some shapes even closed-form expressions for the maximum shear angle can be derived [77, 78, 79]. Further assumptions are: 1) The fabric crosspoints act as friction-free hinges. 2) No slippage occurs at crosspoints. 3) The fabric is in full contact with the tool. Also, the distance between two crosspoints must be chosen shorter than the smallest curvature radius of the component. Models for unidirectional non-crimp fabrics (UD-NCF) also exist and assume a simple shear kinematic [75].

<sup>1</sup>Other denominations are ‘pin-joint’, ‘mapping’ and ‘fishnet’ approach

As such methods disregard the actual constitutive behaviour of the textile, they only approximate the actual physics. For instance, wrinkling can only be indirectly assessed by comparison of the apparent shear deformation and the locking angle [80]. Also, the results depend on the user-defined starting point and the approach can only predict single plies. Ply-ply interactions cannot be modelled. Nevertheless, they are computationally extremely efficient and require no material characterisation. Being continuously refined over decades [81, 82], they are since widely applied in early stages of component design and provide the designer with critical pre-production information.

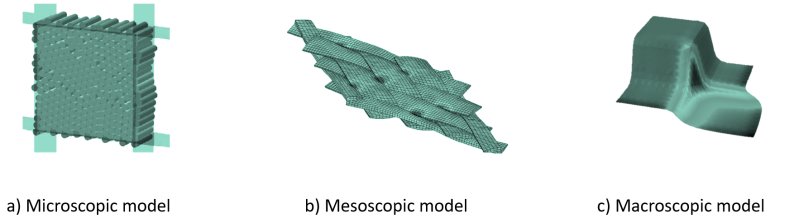
**Mechanical draping.** Unlike kinematic approaches, mechanical approaches seek to formulate and solve constitutive equations for the physical behaviour during forming. Figure 2.14 gives an illustrative comparison of a kinematic and a mechanical simulation along with experimental validation by FENGLER [83, p 38].



**Figure 2.14:** Comparison of mechanical and kinematic draping of a rectangular patch over an edge of a box geometry along with a real example from experiments [83, p 38].

It shows a rectangular patch being draped over the edge of a box-geometry. The kinematic approach reproduces the results over a wide area acceptably well but struggles around the corner area. In contrast, the mechanical simulation captures the results in these critical regions better. Consequently, kinematic approaches are adequate for initial process layout and mechanical simulations for detailed analyses.

As with the deformation mechanisms (cf. Section 2.2.1) mechanical approaches can address multiple scales (micro, meso, macro) to account for the multi-scale nature of textiles.



**Figure 2.15:** Exemplary visualisation of a) a *micro*-scale [22], b) a *meso*-scale [84] and c) a *macro*-scale FE-models of fabrics.

Micro-scale models seek to resolve individual fibres and mainly serve to determine material properties and calibrate material laws in larger scales [22, 85, 86]. An application on component scale is computation-wise practically impossible. Meso-scale models homogenise the fibres but still resolve individual rovings in the textile. This significantly reduces the computation times to a several hours or a few days on high-performance clusters and allows forming simulations on component scale [87, 88]. They provide detailed insights into process dynamics and meso-scale forming effects (cf. Section 2.2.1), but are still so computation-intensive that they thwart a reasonably efficient process engineering with multiple design iterations, let alone iterative optimisation.

For a reasonably efficient part and process design, only macro-scale simulations are eligible. They assume each ply of the fabric as one homogeneous continuum with effective mechanical properties and constitutive models of the related deformation modes. This reduces the computational load significantly but places higher demands on mechanical modelling and numerical implementation [20].

Unlike sheet metal forming, textile forming poses several additional challenges, whose most decisive are shortly outlined in the following. At first, membrane and bending behaviour must be *decoupled*. Textiles

show a low bending stiffness compared to their tensile (membrane) stiffness and imposing classical shell theory would gravely overestimate the bending stiffness, cf. Section 2.2.1. This decoupling can be achieved by decomposing the virtual work or the internal virtual energy in the material formulation to allow the usage of single shell elements, see e.g. [89, 39, 29]. It can also be achieved by superposing conventional membrane and shell elements [90, 91]. Other advanced approaches are still subject of research. A key challenge remains the accurate prediction of the bending and compaction behaviour of thick shells, e.g. by enhanced approaches like generalised continua or solid-shell elements [92, 93, 94, 95, 96].

A second major issue is anisotropy and the non-orthogonal reorientation of the anisotropy-axes (fibre orientation) during shear deformation of fabrics. Fabrics show a low in-plane shear stiffness compared to their tensile stiffness in fibre direction. Accurately capturing the non-linear moduli and their current, non-orthogonal orientation during forming is crucial. Next, the material law must be able to cope with non-linear elastic behaviour at large deformations. To this end typically, but not exclusively, *hypo-* or *hyperelastic* approaches are used. Enhanced constitutive formulations can also account for time-dependent material non-linearities like viscoelasticity, e.g. by VOIGT-KELVIN or generalised MAXWELL material models. See [97, p 38-44] and [29, p 22-27] for details.

Eventually, the large differences of the directional stiffnesses also requires numerical measures to avoid spurious stiffening or relaxation effects like numerical 'locking' or 'hourglassing'. They are pure numerical, physically implausible phenomena and as such they have to be avoided by suitable numerical measures, e.g. choice of element shape (triangular vs. quadrilateral), integration scheme (reduced vs. full), shape function and alignment of mesh and material directions [98, 99].

**Models in this work.** The Finite Element (FE) models used in this work satisfy the above mentioned issues. They are based on the macroscopic

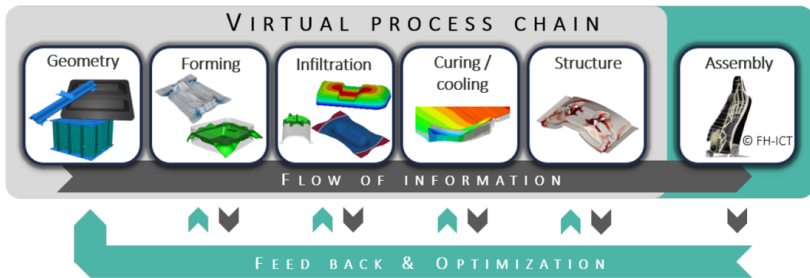
models by POPPE et al. [100, 38, 96, 97] and DÖRR et al. [101, 102, 29]. They comprise parameterised models for membrane, bending and contact behaviour and have been validated experimentally. Decoupling of membrane and bending behaviour is achieved by superposition of membrane and shell elements. The membrane behaviour employs a hyper-viscoelastic constitutive model, bending behaviour is implemented using a hypo-viscoelastic formulation coupled with a VOIGT-KELVIN approach. The models enable a covariant description of material stiffnesses at large strains and describe the fabric in a non-orthogonal, fibre-parallel frame. Kinematic draping is based on an implementation by FENGLER [103, 83] and has partly been advanced by the author.

An in-house developed, PYTHON-based framework [91, 102] has been adopted and further developed for fully automatic model generation, solution and result evaluation. It further enables concurrent handling of multiple jobs to keep the overall computation time within reasonable bounds [104].

**Process simulation chain.** In any multi-step process, and in particular in CoFRP-processes, upstream process steps may affect downstream process steps (cf. Section 2.1.3). Thus, besides accurately modelling the physics of individual process steps, any reliable simulation approach must be able to ingest and pass on information along the chain as illustrated in Figure 2.16. In general, this amounts to transferring scalar-, vector- and tensor-valued information for e.g. temperatures, material orientations and stresses. Mapping such information is a profound task as it requires adaptations between different computational setups, e.g. non-conforming meshes or element types, and calls for appropriate interpolation techniques [105].

A first realisation of such a process chain is presented in [107] for metal processes. For composite materials, KÄRGER et al. [106] developed and validated a virtual process chain. It was later embedded in a process optimisation environment [13] to improve the structural performance of





**Figure 2.16:** Continuous CAE-chain for virtual analysis of manufacturing effects and their impact on the structural behaviour of CoFRP-components. Image concept originally from [106] and modified in [97, p3].

a component by mitigation of process defects. The data interfaces of the process chain have recently been automated and formalised [105] and applied to different composite processes [108, 109, 110, 111]. Similar approaches to consider manufacturing effects have emerged in both industry [112] and academia [113, 114].

Overall, validated simulation methods with different levels of fidelity are available to analyse and assess the formability of a component. Such methods can effectively reduce the need for expensive trials during part- and process-development. However, besides cost for hard- and software, they require considerable computational efforts, which even multiplies for iterative design optimisation.

## 2.3 Optimisation in process engineering

Section 2.2.2 shows that a large number of technology variants in textile forming exists. Additionally, each technology introduces a set of process parameters, which need to be optimised for each component. Simulations help narrow costly experiments to the most promising variants, but their long computation times still call for efficiency during iterative optimisation.

In general, efficient process optimisation employs either 1) available process experience in combination with expert judgment or 2) elaborate mathematical algorithms. This work seeks to combine the advantages of both and accordingly, their core aspects are introduced in the following.

### **2.3.1 Experience-based optimisation**

Some processes appear in a manageably low number of different variants and show such recurring characteristics that they have been standardised, e.g. welding processes [115, p 623-638]. In such situations, process optimisation simply amounts to identifying and adhering to the relevant standards.

When such direct process instructions are not available, sometimes insights can be drawn from experimental trials of application-near reference structures. For instance, OEHLER and KAISER [116, p 284] present diagrams of resulting strains during deep drawing for a range of typical component features. A great advantage of such approaches is that they stem from experiments and do not bear idealisations from modelling. Although certainly providing valuable information to a process engineer, they still give only loose guidance and are restricted to the investigated cases (material, process conditions and geometry).

Consequently, process engineers often initialise their processes based on knowledge and experience from prior parts. Starting from an initial parameter set, the process is iteratively refined based on expert-examination production trials. This can be fast and may yield quick results but has two key disadvantages: First, it commits significant resources to production ramp-up; most notably for downtime of expensive infrastructure, cost for production and analysis of non-marketable output as well as rework for error correction [117, p 3-8]. Second, as with any human-subjective approach, the results vary with experience and knowledge of the expert and further depends on subjective factors like

concentration and available time. On top, unforeseen absence of the expert puts development at risk. Consequently, processes often only arrive at a *feasible* rather than an *optimal* configuration [118].

Trends towards shrinking lot sizes and individualised products further compound the situation. In the wake of ‘mass customisation’, adaptive and flexible manufacturing technologies have emerged, e.g. additive manufacturing [119], modular production cells [120], adaptive moulds [121] or mould-free technologies [122]. Although such technologies *principally* allow for efficient manufacture of individualised products, the recurring need for process optimisation for ever-changing geometries or materials poses a considerable economic barrier [2].

The presented shortcomings call for a more methodological approach towards process optimisation in variable situations. Physics-based process simulation, e.g. FE-simulations, enable a rigorous, virtual analysis of complex process dynamics (cf. Section 2.2.3). Thus, they can serve as a close-to-reality proxy of actual experiments and help ensure manufacturability at comparably low cost. Often termed ‘virtual process optimisation’ (VPO), their inherently digital nature allows to combine them with *optimisation algorithms*, which principally allows for automatic process optimisation [123]. This work studies methods for VPO and outlines its fundamentals in the following sections.

## 2.3.2 Algorithmic optimisation

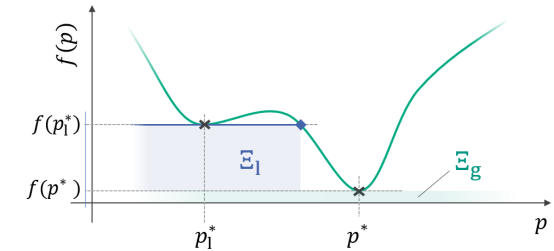
Formally, tailoring technical systems to maximum performance amounts to identifying a set of optimal design variables  $\mathbf{p}^* \in P$  within a feasible design space  $P \in \mathbb{R}^{n_d}$ , often formally cast as

$$\mathbf{p}^* = \arg \min_{\mathbf{p} \in P} f(\mathbf{p}) \quad \text{s.t.} \quad c_{1,2,\dots}(\mathbf{p}) \geq 0 \quad \tilde{c}_{1,2,\dots}(\mathbf{p}) = 0 \quad . \quad (2.1)$$

Therein,  $f : P \mapsto Q$  denotes the objective function, while  $c_{1,2,\dots}$  and  $\tilde{c}_{1,2,\dots}$  are (in-)equality constraints, respectively. The objective function yields

a scalar performance or quality metric  $Q \in \mathbb{R}$  and usually quantifies unwanted behaviour, e.g. deflection under load or defects in manufacturing. Hence, it is sought to be minimised. Note that a maximisation can be cast as a minimisation by  $\arg \max f = \arg \min -f$ .

An optimum  $\mathbf{p}^*$  is present when no other point  $\mathbf{p} \in P$  in a certain region  $\Xi$  around  $\mathbf{p}^*$  can reduce  $f$  below  $f(\mathbf{p}^*)$ . If  $\Xi$  expands over the whole domain, then  $\mathbf{p}^*$  is a *global* optimum, otherwise it is a *local* optimum  $\mathbf{p}_1^*$  as Figure 2.17 visualises with  $\Xi_g$  and  $\Xi_l$ .



**Figure 2.17:** Visualisation of a generic, one-dimensional objective function  $f$  with a local and a global optimum  $\mathbf{p}_1^*$  and  $\mathbf{p}^*$ , respectively.

The constraints  $c_i$  and  $\tilde{c}_j$  impose conditions on the solutions, e.g. regarding the feasible input design space  $P$  due to machine limits. This work considers (box-)constraints only, which require no special treatment. For further information on constrained optimisation the reader is referred to the literature, e.g. [124].

The achievable optimisation efficiency is mainly governed by the objective function  $f$ . More specifically, it is governed by the a-priori available knowledge on  $f$  and the possibility to choose a suitable solution technique.

**Types of objective functions.** From a system engineering vantage point, two characteristics of the objective function  $f$  prevail: *white-box* and *black-box* behaviour [125, p 77]. White-box behaviour describes a

system whose inner functions are known and can be advantageously exploited during optimisation. In a black-box situation they are unknown so that only the phenomenological input-output-behaviour can be observed.

Both types occur in engineering. A typical example for a white-box problem is FE-based topology optimisation [126, p 179-201]. Knowledge about the involved physics can be used to provide additional information, e.g. on second-order gradient-information [127], which are prime conditions for some gradient-based techniques. Another solution strategy are optimality criteria: They use existing engineering knowledge on the optimal system state, e.g. homogeneous stress, to guide the optimisation [128, 129, 130, 131].

As shown above, white-box problems are solved efficiently by exploiting known properties of the objective function  $f$ . However,  $f$ 's inner structure can be a-priori unknown, impossible to detect or for other reasons non-exploitable [132]. Such 'black-box' situations impede assumptions about the problem structure and for generalisability, they are the basis of this work.

In such situations only purposeful parameter exploration and iterative parameter variation are eligible. The iterations continue until triggering of a stopping criterion. The set of instructions how to explore and vary the parameters is commonly referred to as 'optimisation algorithm' or 'optimiser'. Iterative optimisation algorithms are usually divided into *gradient-based* and *gradient-free* methods [133, p 1-4].

**Optimisation algorithms.** Gradient-based methods employ first- and/or second-order derivative-information to generate a sequence of candidate solutions which eventually accumulates at least at a local minimum [134, p 47-92]. If  $f$  is convex and gradient-information is readily available, gradient-based methods show unparalleled efficiency and are guaranteed to convergence even for high-dimensional problems. They have

demonstrated their applicability in multiple engineering tasks yet seem to be mainly used in structural applications, see e.g. [135, 136].

However, black-box objective functions often show unsuitable characteristics for gradient-based methods [137, p 36]. For instance, gradient-information may be expensive to obtain or the objective may be non-convex and multi-modal so that the gradient-based optimiser converges to an inferior local optimum only. This favours gradient-free techniques as benchmarks in practical engineering settings show [138, 139]. Overall, gradient-free methods comprise three broad categories: *Model-based*, *direct search* and *heuristic* methods [137, p 37-50].

Model-based methods tacitly assume that – despite being a-priori unknown – the objective function bears some structure, at least in a certain ‘trust region’. They further assume that an analytical model, e.g. a low-order-polynomial, can *locally*<sup>1</sup> approximate this structure from a collection of points. This local model can then be used to find an (locally) improved point, e.g. by gradient-based methods, and the procedure starts anew. Although long studied [140], extended [141] and implemented in optimisation libraries [142], they seem rarely applied in engineering scenarios as only few recent applications are reported, e.g. [143, 144].

Direct search methods, most notably the ‘downhill simplex’-algorithm [145], evaluate a starting point and some pre-defined points in its vicinity. By sensible comparison and a clever, rule-based combination scheme within this point cloud, a new, potentially improved candidate solution is determined and evaluated and compared to the initial point cloud. Depending on the outcome, it replaces a point in the point cloud and the procedure starts anew. Thereby the point cloud gradually moves ‘downhill’ towards the next optimum. Although requiring no gradient-information, they are still likely to converge to a local optimum and have thus been superseded by heuristics over the last decades [146].

---

<sup>1</sup>Unlike surrogate models (cf. Section 2.3.3), which strive for a *global* approximation.

Heuristic methods also evaluate and combine a collection of points, the ‘population’ but differ in two major regards: First, their point cloud stretches over a much wider parameter space instead of the narrow vicinity of a starting point. Second, they introduce an element of randomness into the combination process: The new candidate solutions are not only combined by deterministic rules but additionally randomly mutated to a certain degree. The two mechanisms facilitate design space exploration and discovery of new promising solutions. Thus, these algorithms can escape local minima and tend to find the global optimum.

The combination and mutation principles are often nature-inspired<sup>1</sup>. Owing to the elements of serendipity, heuristics are mathematically not guaranteed to converge, let alone at a certain speed, but in practice they prove robust and tend to find optima even under adverse conditions [147]. Thus, they have become a standard tool in engineering optimisation over the last decades, see e.g. [148, 149] for composite examples, and have as such been successfully applied in fabric forming optimisation [66, 108, 150, 151].

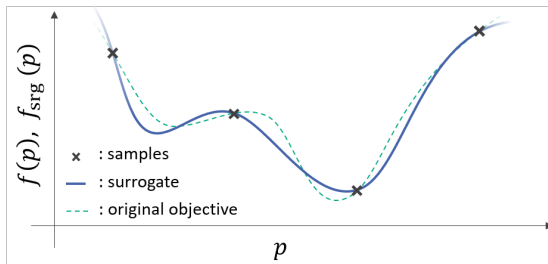
The presented gradient-free approaches can cope with black-box conditions and especially heuristics tend to find a global minimum of the objective function. However, they generally require many iterations and evaluations of  $f$  for convergence. This renders their direct application impracticable for VPO: Since process simulations are generally computation-intensive and may take up to a few hours, an iterative optimisation quickly amounts to days or weeks and often exceeds available resources [152]. Consequently, measures are required to increase optimisation efficiency.

---

<sup>1</sup>Hence their biological names, e.g. ‘genetic’, ‘particle swarm’ or ‘ant colony’.

### 2.3.3 Surrogates for efficient optimisation

One option to reduce the computational load when optimising expensive functions is surrogate-based optimisation (SBO) [153, p V<sup>1</sup>]. The key idea of SBO is to approximate the expensive objective function  $f$  by an easy-to-evaluate function  $f_{\text{srsg}}$  – the ‘surrogate’. Figure 2.18 visualises schematically an approximation based on four samples for a generic, one-dimensional example.



**Figure 2.18:** Schematic comparison of the original objective  $f$  and the surrogate  $f_{\text{srsg}}$  (approximation). Although  $f_{\text{srsg}}$  does not exactly match  $f$ , it still reproduces  $f$ 's structure and optima from the sampled observations.

Similar to model-based optimisers (cf. Section 2.3.2), SBO assumes that  $f$  has a certain structure which can be reproduced by another model function. Unlike them however, surrogates usually aim for a *global* instead of a local approximation. Once available, optimisation takes place on the surrogate in short time, which allows to concentrate costly simulations on the most promising parameter regions.

Besides optimisation, surrogates can also help determine sensitivities and are useful for repetitive evaluations, e.g. for uncertainty quantification [154] and reliability assessment [155]. In the context of composites, applications range from material scale, e.g. assessment of stiffness and

---

<sup>1</sup>Roman numeral



strength [156, 157], to component scale, e.g. vibrational [158] characteristics. It may be noted however, that surrogate techniques are not the only approach towards uncertainty and robust design, see [159] for an example of non-surrogate structural optimisation under uncertain input parameters.

**Surrogate construction.** Formally, a surrogate model<sup>1</sup> is a numerically efficient approximation of the relation of input parameters  $\mathbf{p}$  and observed responses  $f(\mathbf{p})$ , i.e.

$$f_{\text{srg}}(\mathbf{p}) \approx f(\mathbf{p}) \quad \forall \quad \mathbf{p} \in P \quad (2.2)$$

In many situation  $f$  is a black-box function which can be evaluated but is otherwise a-priori unknown. This makes classical analytical approximation techniques, e.g. Taylor Series, basically inapplicable as they typically require additional information, e.g. higher-order derivatives. Outside dedicated niche-frameworks [160, 161], numerically attaining such information quickly becomes prohibitively computation-intensive – especially for higher derivatives and multiple dimensions. Consequently, only data-driven techniques based on input-output-observations remain eligible.

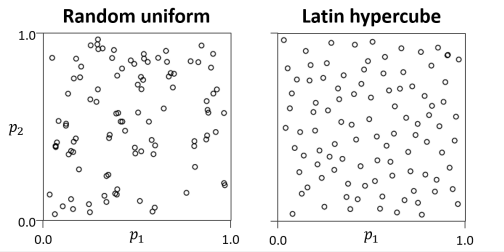
Obtaining a data-driven model – often termed ‘training’ – can be seen as tuning a highly-flexible model function  $f_{\text{srg}}$  from the model class  $\mathcal{F}$  in order to match  $k$  sampled observations in a data set  $D = (\mathbf{p}_i, f(\mathbf{p}_i))$ ,  $i = 1, \dots, k$ .

The samples are obtained according to dedicated sampling plans [162, p 153], which can be done efficiently in parallel. For deterministic computer experiments so-called ‘space-filling’ plans are typically used. In simple terms, they seek to distribute samples uniformly in the design

---

<sup>1</sup>Sometimes also called ‘meta-model’ or ‘response surface models’

space  $P$ , with as few and as small gaps as possible. Different approaches exist to ensure uniform coverage of  $P$  [163, 164]. However, Latin-Hypercube Sampling (LHS) and its sub-variants have established as the de-facto standard in practice as it proves sampling-efficient [165] and convenient to use. Figure 2.19 compares random sampling strategy and LHS-sampling. Clearly, LHS-sampling provides a more even distribution of samples with fewer holes.



**Figure 2.19:** Comparison of random-uniform sampling and Latin Hypercube Sampling (LHS). Clearly, LHS yields a more even coverage of the design space.

The model function  $f_{\text{srg}}$  offers adjustable parameters  $\theta$  which are tuned to match the supplied data. Formally this is often expressed as finding a set of optimal parameters  $\theta^*$  which minimise an error metric  $\varepsilon$ , the ‘loss’:

$$\theta^* = \arg \min_{\theta} \varepsilon(f_{\text{srg}}, D) \quad (2.3)$$

The loss  $\varepsilon$  is a distance measure between model predictions  $f_{\text{srg}}(\mathbf{p})$  and ground truth values  $f(\mathbf{p}_i)$  in the database  $D$ . In principle, different error metrics can be used, while for regression tasks the mean absolute error (MAE) and the mean squared error (MSE) have established in practice [162].

**Surrogate function types.** Similar to objective functions (cf. Section 2.3.2), surrogate models can be of white-box and black-box type. White-box

surrogates have a well-known structure and allow for certain interpretation. They often involve algebraic expressions whose predictable behaviour is used to capture a-priori known system dynamics. In this way, saturation effects or periodic behaviour can be directly built into the model equations, see [166, 167, 168] for examples. Non-algebraic white-box models exist as well, most notably decision trees. They process data by a sequence of comparably simple conditional statements and allow for engineering interpretation as exemplarily shown in [169].

White-box surrogates come with several advantages regarding model introspection, engineering interpretation and ease of use without elaborate software toolboxes. However, they typically show limited modelling capacity, i.e.  $f_{\text{srg}}$  may not fully reproduce  $f$ 's structure in complex cases.

Black-box surrogates seek to overcome this limitation and are often so-called *universal approximators* [170, 171, 172]. That is, with appropriate configuration they can reproduce any continuous function regardless of its complexity. However, they do so at the expense of human interpretability: They typically consist of complex and often nested functions with hundreds to millions of model parameters and even when their structure and parameters are fully known, their sheer number refuses human comprehension.

Many different black-box model functions have been developed, e.g. support vector machines, Gaussian Processes or neural networks, and are readily available in both commercial and academic software packages [173, 174, 175]. However, despite their diversity, WOLPERT [176] has expounded early on that no technique consistently outperforms all others in all cases. It has since been confirmed empirically in multiple disciplines, e.g. [177, 178], and is sometimes casually referred to as 'No free lunch theorem'. Consequently, the choice of the model function is always case-dependent.

**Surrogate-based optimisation.** Once trained, the surrogate model  $f_{\text{srg}}$  provides an easy-to-evaluate approximation of the expensive function  $f$  and thus, even an iteration-intensive black-box optimiser will find the optimum of the surrogate  $\mathbf{p}_{\text{srg}}^*$  in short time.

WANG et al. [179] call methods, which assume this surrogate-optimum  $\mathbf{p}_{\text{srg}}^*$  as the true optimum  $\mathbf{p}^*$ , ‘offline’ methods. However, as  $f_{\text{srg}}$  is just an approximation of  $f$ , the surrogate optimum  $\mathbf{p}_{\text{srg}}^*$  most likely deviates from the true optimum  $\mathbf{p}^*$ . Thus, ‘online’ methods are recommended in general, which try to eliminate deviations by iteratively improving the surrogate with new (simulated) evidence of the candidate optimum solutions. Thereby, the model gradually concentrates costly simulations on the most promising regions and refines near potential optima. Figure 2.20 compares a classical black-box optimiser to SBO and visualises the speed-up.



**Figure 2.20:** History of candidate solutions during a) direct optimisation with a genetic algorithm (GA) and b) SBO. Sub-figure c) shows a faster reduction of the objective  $f$  (convergence) over the iterations for SBO [180, 181].

The comparison of Figure 2.20 a) and b) shows that, unlike direct optimisation, SBO places the candidate solutions always near the optimum. Despite the initial sampling offset (blue shade), SBO yields a faster reduction of the objective  $f$ , i.e. higher optimisation efficiency, cf. sub-figure c).

Different variations of SBO exist: For instance, some SBO-methods propose space-reduction in promising regions in combination with re-sampling and re-optimisation only in these reduced regions. Although

such approaches can give good results, they are prone to premature convergence to local optima [179]. Also ‘space-mapping’ is an alternative approach, which seeks to localise the optimum of the objective function through transformation of the surrogate optimum [182, 183]. Some surrogate models, namely BAYESIAN models, provide not just a point estimation  $f_{\text{sr}}(\mathbf{p})$  but a complete probability distribution of the output quantity. Such additional information can additionally help explore the design space and purposefully reduce deviations between surrogate and ground truth. An example is the ‘EGO’<sup>1</sup>-algorithm [184]. It has proven efficient and reliable in both academic and real-world problems [185], although appears to struggle in high-dimensional spaces [186].

In any case, SBO reduces the overall computation time by two principles: parallelisation of initial data sampling and purposeful guidance of the optimiser in the parameter space. However, current SBO-strategies provide mostly application-specific, one-off models and struggle with unforeseen task variations. This impacts on the re-usability in new scenarios: Even a subtle problem variation, e.g. a geometry change, instantly invalidates the surrogate and requires resampling of data and reconstructing the surrogate. Thus, demand for generalised models has been identified early on, e.g. in [187], and continues to be prevalent. Recent advances in Machine Learning (ML) may provide the means to address this issue.

### 2.3.4 Machine Learning in Engineering

Classical computer programmes employ sequences of carefully crafted and well-defined instructions (‘rules’) to solve a problem, e.g. the finite difference method for a PDE. However, such rule-based concepts struggle when the governing rules are too complex to be cast into explicit instructions.

---

<sup>1</sup>EGO: Efficient Global Optimisation.

HAMOUCHE et al. [188] exemplify this by selecting a suitable forming process (deep drawing, spinning, rolling etc.) for a given geometry: Although a simple task for a seasoned practitioner, it proves difficult to define explicit selection-instructions for CAE-software and according attempts are often error-prone, inflexible or over-specialised [188, 189]. Consequently, such tasks are still manual and at best qualitatively guided [190, p 14-20].

ML provides an alternative approach to handle such difficult-to-define tasks, mainly by two means: 1) *Learning* instead of hand-crafting programme rules and 2) using *all relevant*, possibly high-dimensional data instead of a manually preprocessed, low-dimensional representation. Both the learning and data concepts will be presented in the following.

**Learning concepts.** MITCHELL [191, p 1] defines ML as designing ‘computer programs that *automatically* improve with experience’. That is, ML techniques, especially *deep learning* (DL), give up on hand-crafting programme instructions but seek to identify, reproduce and extract solution strategies solely from observations. And they do so with great success: In 2012 KRIZHEVSKY et al. [192] used DL for the first time in the ‘ImageNet’-competition, an annual computer vision benchmark. Their approach outperformed all manually written programmes by a large margin and thereby proved the practicability of self-improving, learning algorithms for complex tasks.

ML-algorithms split up into three main groups, 1) *unsupervised*, 2) *supervised* and 3) *reinforcement learning*, as the following paragraphs summarise.

Unsupervised learning aims at detecting similarities in data. For instance, they help understand systems by automatic grouping or clustering of similar system states or can inform about the significance of input parameters. Countless applications are reported across disciplines and also in product manufacturing: For instance, MCLEAY et al. [193] use it to live-detect anomalies in machining processes from sensor data and

SCHWARZ et al. [194] to tell relevant and irrelevant process parameters apart in deep drawing.

Supervised learning aims at establishing a phenomenological relation between input and output of a system. The output values can be discontinuous ('classification') or continuous ('regression'). Once the relation is available, it can be used for further analysis and optimisation. Classical surrogate models from Section 2.3.3 fall into this category.

Ultimately, reinforcement learning (RL) aims at estimating optimum actions in variable situations. Unlike the previous approaches, the algorithm is not presented a self-contained set of data but interacts with a responsive environment which issues action-dependent feedback signals. RL originates from control theory [195, p 16-17] but has recently drawn attention in engineering optimisation: For instance, DORNHEIM et al. [196] use it to estimate suitable processing paths to achieve desirable metallographic properties, while GÜNTHER et al. [197] apply RL to online-optimize laser welding processes. ZHOU et al. [198] optimize chemical reactions by RL: Unlike other work, they specifically seek to reduce the need for experimental data by inclusion of simulations to pre-train RL-algorithm before deployment.

With respect to manufacturing processes and their recurring need for component-specific optimisation (mass customisation, cf. Section 2.2.2), SHAMSAEI et al. [2] call for a comprehensive framework to 'leverage information from prior similar studies and ... systematically characterize the relation between process parameters and part features so that the ... process can be optimized in a more efficient manner'. Owing to the complexity of these relations, such a framework does not exist yet, although desirable. However, the presented learning concepts, especially RL, hold substantial promise in that regard.

**Data concepts.** As outlined above, besides a suitable learning concept, a generalised surrogate model also requires sufficiently informative

training data. To this end, LATANIOTIS et al. [199] distinguish *unstructured* and *structured* variables.

*Unstructured* input variables lack an intrinsic ordering, i.e. are self-contained and can be arbitrarily rearranged without loss of information. In engineering modelling they are typically associated with interpretable quantities such as temperature or loads at a specific location. In contrast, *structured* data does have a natural ordering and shows strong correlations along a physically meaningful set of coordinates, often time or space. Typically, the individual variables have little meaning on their own and only their well-ordered entirety provides information and a careless rearrangement leads to information loss.

Figure 2.21 illustrates the difference in a generic thought experiment: Suppose, the pressure distribution of a central source flow in a quadratic domain needs to be quantified. Sub-image a) quantifies the situation by three unstructured variables (pressure sensors  $p_{1...3}$ ); sub-image b) by structured variables, namely a full-field image of the pressure distribution. For completeness, the sensor's locations are given in sub-image b) (colour-coded). The sub-images come in two variants: original data (top) and shuffled columns (bottom).

Although the columns have been rearranged, the unstructured variables remain interpretable due to their inherent, technical meaning (inlet/outlet). The structured variables however, have completely lost their meaning. Sub-image c) gives an even more intuitive example of the information loss: Clearly, the original image shows the picture of the MONA-LISA but it is hard to identify it in the shuffled image.

However, structured variables allow to encode information which is hard to express through unstructured parameters: For instance they can excellently represent field quantities. Figure 2.22 compares unstructured and structured variables to encode thermal field as an initial condition (IC) for the heat diffusion problem from [200]. Both can efficiently describe the IC in simple cases (sub-image a) ) but in more complex situations (sub-image b) ), an unstructured, algebraic representation quickly



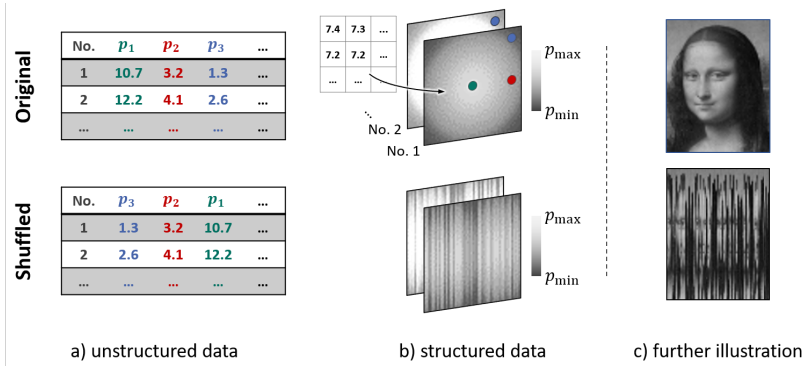


Figure 2.21: Comparison of a) unstructured and b) structured data at the example of a generic pressure distribution. Structured data show strong internal correlations. Thus, they lose their interpretability during shuffling operation (bottom), while unstructured variables retain it.

becomes convoluted. The structured input however, can efficiently reproduce both.

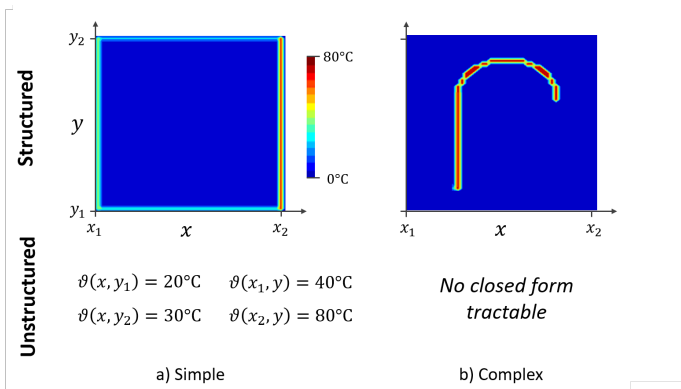


Figure 2.22: Comparison of structured and unstructured variables for a) simple and b) complex boundary conditions for a heat diffusion problem [200]. Structured variables can encode even complex situations, which unstructured struggle with.

However, this higher expressivity comes at the cost of considerably increased dimensionality: While unstructured variables typically range in the order of  $\mathcal{O}(10^{0\dots 2})$ , for structured data it quickly reaches  $\mathcal{O}(10^{3\dots 6})$  [199], clearly exceeding the capabilities of current surrogate techniques.

In some cases, *dimensionality reduction* techniques can be applied which seek to reproduce high-dimensional information in a lower-dimensional space [201]. However, loss of information often limits data compressibility and the need for high-dimensional surrogates persists. The advent of artificial neural networks (ANN) over the last decades offers means for both *handling of* and *learning from* large amounts of high-dimensional structured data, as outlined in the following.

### 2.3.5 Neural networks

ML is generally not restricted to a certain set of model functions; however, *neural networks* have taken centre stage in the field. This is mainly for four reasons: 1) They are universal approximators (cf. Section 2.3.3) [172] and show thus a remarkable modelling capacity. 2) A rich corpus of efficient and tested training algorithms is at disposal. 3) Specialised sub-types of neural networks exist, which take advantage of structured data [202, p 326] and 4) comprehensive open-source libraries allow for convenient and efficient implementation [203].

**Artificial Neurons.** ANNs are loosely inspired by the neural activity in vertebrates' brains which process information by electrical impulses. More specifically, if incoming impulses exceed a certain threshold, the neuron's synapse activates and transmits a signal – it 'fires'. Otherwise it remains inactive. Interestingly though, the synapses' thresholds are not fix but can adapt upon repeated activation and weigh incoming signals, which manifests as 'learning' in the macro-sense. [204, p 266-268]

McCULLOCH and PITTS [205] proposed as early as 1943, a mathematical model of a neuron with a number of input-values  $\mathbf{x} = (x_1, \dots, x_{n_d})^T$

and one output-signal  $\hat{y}$ . It processes data by weighting with the weight vector  $\boldsymbol{\theta}_w = (\theta_{w_1}, \dots, \theta_{w_{n_d}})^T$  and summation of the input signals plus a bias  $\theta_b$ :

$$z(\boldsymbol{x}) = \boldsymbol{\theta}_w^T \boldsymbol{x} + \theta_b. \quad (2.4)$$

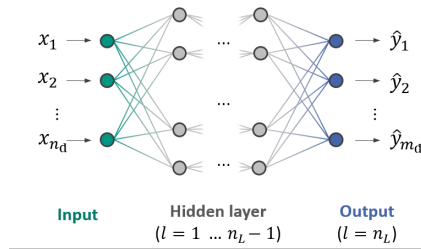
The weights  $\boldsymbol{\theta}_w$  and the bias  $\theta_b$  are adjustable parameters which can be tuned to match supplied data. At this stage, it is a mere linear-affine operation which cannot reproduce non-linear relations. Thus, an activation-function  $\psi$  introduces a non-linearity:

$$\begin{aligned} \hat{y}(\boldsymbol{x}) &= \hat{y}(\psi(z)) \\ &= \hat{y}(\psi(\boldsymbol{\theta}_w^T \boldsymbol{x} + \theta_b)) \end{aligned} \quad (2.5)$$

Different activation functions are applicable in general, while the *sigmoid* function  $\psi(z) = 1/(1 + e^{-z})$  and Rectifying linear unit (ReLU)  $\psi(z) = \max(0, z)$  with several sub-variants have proven their practicality in benchmarks [206] and are widespread in practice [207].

**Neural networks.** In their classical ‘feedforward’ form, ANNs are organised in a layer-structure with each layer  $l \in \{1, \dots, n_L\}$ , carrying  $n_{d,l}$  neurons (cf. Figure 2.23). The neurons collectively yield a layer-wise output vector  $\hat{\boldsymbol{y}}_l = (\hat{y}_1^l, \dots, \hat{y}_{n_{d,l}}^l)^T$ . The neurons of neighbouring layers are fully or partially interconnected so that the output of a previous layer is input to the following layer. See [202] for a comprehensive discussion of ANN-theory.

Information propagates through the network by neuron (de-)activation (Equation (2.5)) in subsequent layers. During this so-called ‘forward pass’, complex activation patterns form within the network. Ultimately, the last layer  $l = n_L$  converts them to an output quantity  $\hat{\boldsymbol{y}} = \hat{\boldsymbol{y}}_{n_L}$  for engineering interpretation. The intermediate layers are an internal representation of information with next to no interpretability, hence their common name ‘hidden layers’. The number of layers and neurons is arbitrary so that neural networks can in principle perform any mapping  $\mathbb{R}^{n_d} \mapsto \mathbb{R}^{m_d}$ .

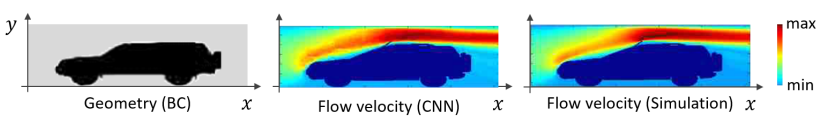


**Figure 2.23:** Schematic illustration of a fully connected feedforward neural network.

The above descriptions hold for *unstructured* input, e.g. distinct process parameters (cf. Section 2.3.4), and so far give little novelty compared to classical surrogates. However, ANN variants exist, which are specifically designed to cope with high-dimensional *structured* input, e.g. images. For instance, pioneered by FUKUSHIMA [208], put into practice by LECUN et al. [209] and substantially streamlined by KRIZHEVSKY et al. [192], *convolutional neural networks* (CNN) have established for analysis of spatially structured data like images. As they are used in this work, Section 4.1.2 provides background information on their functioning.

Though already introduced in the 1980s, it was not until a few years ago that CNNs appeared in engineering applications: For instance, FARI-MANI et al. [200] and GUO et al. [210] use them for full-field estimation of thermodynamic and aerodynamic problems. Unlike classical surrogates, they encode boundary conditions (BC) and geometries by structured input variables (images) instead of unstructured parameters and report a considerably enhanced generalisability. Figure 2.24 visualises the geometry encoding scheme along with a CNN-estimation of the velocity field and an according simulation for comparison.

This idea has been transferred to manufacturing scenarios by the author, e.g. in [211, 104], and other authors: HAMOUCHE et al. [188], GÜNTHER et al. [197], ATTAR et al. [212] and ZHOU et al. [213] use them in process engineering for efficient process selection, control and manufacturability estimation, respectively. While individual tasks differ, all results underpin that – given sufficient data – CNNs are capable of learning complex



**Figure 2.24:** Schematic illustration of flow field estimation of a convolutional neural network (CNN) based on structured input (geometry) and output variables (flow velocity) along with an actual simulation for reference [210].

dynamics from structured data and outperform classical surrogates regarding generalisability and re-usability in new scenarios.

**Neural network training.** The network parameters  $\theta$  are initialised randomly and consequently the initial forward pass will yield a random result  $\hat{y}$  and a large error  $\varepsilon$ . In order to match the data, RUMELHART et al. [214] proposed to improve the network by gradually tuning  $\theta$  in direction of decreasing error  $\varepsilon$ , i.e. solve Equation (2.3)

$$\theta^* = \arg \min_{\theta} \varepsilon(f_{\text{srg}}, D) \quad (2.3 \text{ revisited})$$

iteratively by gradient-descent.

More specifically, they suggest to ‘back-propagate’ the error  $\varepsilon$  by computing  $\nabla_{\theta} \varepsilon$ , i.e. the gradient of  $\varepsilon$  with respect to  $\theta$ , and then applying

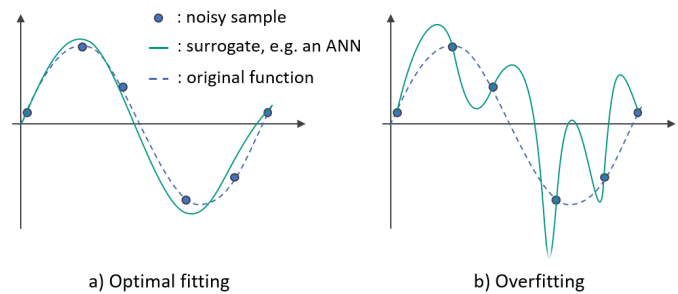
$$\theta^{i+1} = \theta^i - \eta \nabla_{\theta} \varepsilon \quad (2.6)$$

to each observation in the dataset  $D$ . The variable  $\eta$  denotes the ‘learning rate’<sup>1</sup>, a critical hyperparameter [202, 149-150]. The procedure repeats for a pre-defined number of epochs  $i = 1 \dots n_{\text{ep}}$ , while one epoch denotes passing through all samples in  $D$ . See [202, p 200-220] for details.

Owing to their large modelling capacity, neural networks – and especially deep neural networks – tend to overfit to training data. Figure 2.25

<sup>1</sup>Literature on gradient-based optimisation typically refers to  $\eta$  as ‘step size’.

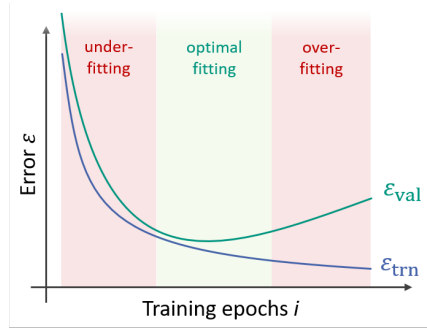
visualises the phenomenon schematically: While an optimally trained ANN captures the data-inherent dynamics (left), an overfit ANN (right) only reproduces the training data, but does not generalise to new data. Such a model is of little predictive use.



**Figure 2.25:** Schematic illustration of overfitting. An optimally fit surrogate reproduces the training data and can generalise to new data, while an overfit model reproduces training data only.

To avoid overfitting, it is common practice to utilise two separate sets of data: a training set  $D_t$  and a validation set  $D_v$ . The network fits to  $D_t$  only, i.e. minimises the training loss  $\varepsilon_{\text{trn}}$ , while constantly monitoring the validation loss  $\varepsilon_{\text{val}}$  on  $D_v$  as a measure of generalisation capability. Figure 2.26 illustrates the typical evolution of  $\varepsilon_{\text{trn}}$  and  $\varepsilon_{\text{val}}$  during training: Initially, both decrease, yet at some point  $\varepsilon_{\text{val}}$  grows again. At this point, training should stop as the model starts to overfit and loses its predictive accuracy on new, non-training data.

Several measures have established to counter overfitting. First, instead of applying Equation (2.6) to individual observations, the average gradient of multiple observations (‘minibatches’) is used. This increases the robustness of the training process and substantially reduces overfitting as well as the training time [215]. Other measures include *early-stopping* of the training [202, p 241-249], additional regularisation terms in loss function [202, p 224-226], random *dropout* of neurons during training [216].



**Figure 2.26:** Typical evolution of the training error  $\epsilon_{\text{trn}}$  and validation error  $\epsilon_{\text{val}}$  during surrogate training. The training error decreases continuously during training, while the validation error increases again at some point (begin of overfitting).

Also the number of layers  $n_L$  and neurons per layer  $n_l$  impact on the final network performance. These ‘hyperparameters’ need to be tuned in an upstream study. To this end, automated approaches have been proposed, e.g. [217], but with a little knowledge they can be set and efficiently refined manually [202, p 423-426].

## 2.4 Literature summary and research gap

**Summary.** Many lightweight engineering tasks amount to making material, process and geometry requirements match in mutual regard. This overall goal poses a great challenge, especially for materials with a complex behaviour like textiles used in CoFRP-components.

Manufacturing CoFRP-shells often involves a deep-drawing-like forming process of a textile. A common manufacturing defect in textile forming is wrinkling, an instability effect triggered by a complex interaction of compressive in-plane stresses and a low bending stiffness. Predicting its occurrence is a profound question and requires extensive experience from previous parts or elaborate process simulation techniques.

In general, forming processes must be adjusted so as to avoid wrinkling. To this end, most technological approaches manipulate the material draw-in. Although the principles differ, each brings a set of continuous variables that must be optimised for the envisaged component geometry.

Finding this parameter optimum is a challenging and resource-intensive endeavour, though: In industrial practice, experience-guided trial-error-experiments are applied but incur significant cost. Process simulations can reduce experimental work but prove computation-intensive, especially for iterative optimisations. Ever changing products and shorter production cycles ('mass customisation') further compound the situation and make optimisation-efficiency a key factor.

Surrogate-based optimisation seeks to reduce according efforts by concentrating costly simulations on the most promising regions. Although significant speed-ups are reported, current surrogate models are mostly low-dimensional and rather inflexible. As such they struggle with ever-changing optimisation tasks, e.g. variable geometries. Machine Learning (ML) techniques, especially deep neural networks, are designed to deal



with complex, high-dimensional data. The nascent availability of ML-techniques, especially deep learning, has been shown to open up new avenues for efficient optimisation in variable situations.

**Research Gap.** Unlike classical surrogates, ML-techniques have been shown to learn complex system dynamics from data and cope well with structured data like images. At the same time physics-based process simulations provide close-to-reality samples of manufacturing processes. Owing to their easy re-configurability regarding process conditions or component geometry, they allow to sample a highly diverse set of process observations.

Consequently, automated numerical simulations give access to comprehensive part-process-observations while recent ML-algorithms promise being able to extract inherent patterns from complex and high-dimensional data. Thus, their combination sounds appealing.

However, at the moment it is unclear, a) whether modern ML-techniques can indeed extract generalisable information from generic process-samples in order to efficiently optimise the process of a new, non-generic geometry and b) if so, whether this speeds up the optimisation similar to a classical surrogate. This research gap principally applies to any manufacturing process, but will be studied at the example of textile forming.



## 3 Reference optimisation method

High-fidelity process models, e.g. FE-simulations, in combination with general-purpose optimisation algorithms ('virtual process optimisation') help reduce costly process trials during production ramp-up, see e.g. [13, 150], but typically require considerable computational resources.

Surrogate-based optimisation (SBO) generally reduces the computational load when optimising expensive-to-evaluate functions like simulations [218]. However, the involved surrogates are mostly problem-specific, one-off models and struggle with problem variations (cf. Section 2.3.4). This work evaluates the potential of ML-techniques to enhance current surrogates. In order to benchmark the developed ML-methods (Chapters 4–5), this chapter presents the reference optimisation approach.

More specifically, Section 3.1.1 describes how to form the objective function from a simulation model. Section 3.1.2 then outlines how objective function, surrogate and optimiser generally work together along with a minimal application example (Section 3.1.3) for graphicness. Then, Section 3.1.4 presents three different surrogate-approaches using neural networks. For validation of the reference method, Section 3.2.1 presents at first the considered use case of gripper-assisted fabric forming. Then Section 3.2 evaluates the surrogate-approaches accuracy-wise, selects one for SBO and compares the optimisation performance to a state-of-the-art genetic algorithm. The chapter concludes with an intermediate summary in Section 3.3.

## 3.1 Optimisation approach<sup>1</sup>

### 3.1.1 Objective function

From an optimisation perspective, a process simulation can be seen as a function  $\varphi : P \mapsto A$  which maps process parameters  $\mathbf{p} \in P \in \mathbb{R}^{n_d}$ , i.e. an  $n_d$ -dimensional input-vector, to a part quality attribute  $\mathbf{a} \in A$ . In the context of process optimisation,  $A$  usually quantifies the extent of defects, e.g. wrinkles. Please note that  $\mathbf{a} \in \mathbb{R}^{n_{\text{el}}}$  is a field quantity and thus the overall product quality is also a vector  $\mathbf{a} = (a_1, \dots, a_{n_{\text{el}}})^T$  with  $n_{\text{el}}$  being the element count of the field.

For optimisation, this quality field needs to be converted to a scalar performance indicator using a *quality metric*  $q : A \mapsto Q$  with  $Q \in \mathbb{R}$ . To this end, often the vector norm

$$q(\mathbf{a}) = \|\mathbf{a}\|_u = \left( \sum_{i=1}^{n_{\text{el}}} |a_i|^u \right)^{1/u} \quad (3.1)$$

is used [179]. It contains the maximum-norm ( $u = \infty$ ), the sum-norm ( $u = 1$ ) and the EUCLIDEAN norm ( $u = 2$ ) as special cases. With such a performance indicator, the objective function  $f : P \mapsto Q$  can be expressed as a composition of the simulation  $\varphi$  and the performance metric  $q$ :

$$f(\mathbf{p}) = q(\mathbf{a}(\mathbf{p})) = q(\varphi(\mathbf{p})) \quad . \quad (3.2)$$

Consequently, optimisation amounts to

$$\begin{aligned} \mathbf{p}^* &= \arg \min_{\mathbf{p} \in P} f(\mathbf{p}) = \arg \min_{\mathbf{p} \in P} q(\mathbf{a}(\mathbf{p})) \\ &= \arg \min_{\mathbf{p} \in P} q(\varphi(\mathbf{p})) \quad . \end{aligned} \quad (3.3)$$

---

<sup>1</sup>The presented approaches are based on the prior works [152, 181, 219]

As Section 2.3.2 outlines in detail, Equation (3.3) could be solved using black-box optimisation algorithms like heuristics but takes prohibitively long. In such cases, surrogate-based optimisation suggests to devise an easy-to-evaluate approximation

$$\begin{aligned} f_{\text{srg}}(\mathbf{p}) &\approx f(\mathbf{p}) \\ \xrightarrow{(3.2)} \quad q_{\text{srg}}(\mathbf{p}) &\approx q(\varphi(\mathbf{p})) \quad . \end{aligned} \quad (3.4)$$

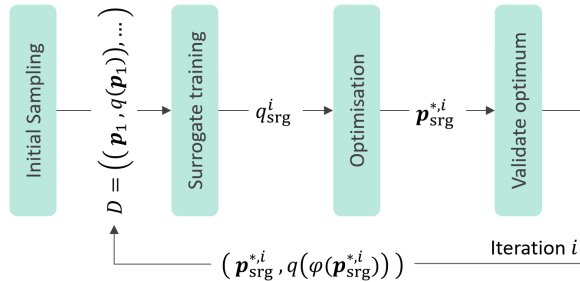
Optimisation then takes place on  $q_{\text{srg}}$  instead of  $q$ .

### 3.1.2 Optimisation on the surrogate

Once  $q_{\text{srg}}$  is available, it can be used for SBO. Since different SBO-variants exist in literature [179], Figure 3.1 on the following page visualises the concept adopted in this work: After a first sampling to initialise the database  $D = \{(\mathbf{p}_1, q(\mathbf{p}_1)), \dots\}$ , a surrogate is trained, in this work an ANN. Then, a genetic algorithm (GA) determines a candidate solution  $\mathbf{p}_{\text{srg}}^{*,i}$  on the surrogate by solving

$$\mathbf{p}_{\text{srg}}^{*,i} = \arg \min_{\mathbf{p} \in P} q_{\text{srg}}^i(\mathbf{p}) \quad . \quad (3.5)$$

However, the surrogate  $q_{\text{srg}}^i$  is only an approximation of  $q$  and similarly  $\mathbf{p}_{\text{srg}}^{*,i}$  is only an approximation of the actual optimum  $\mathbf{p}^*$  of  $q$ . To assess the candidate's validity, a simulation run evaluates  $q\left(\varphi\left(\mathbf{p}_{\text{srg}}^{*,i}\right)\right)$ . For brevity,  $q\left(\mathbf{p}_{\text{srg}}^{*,i}\right) = q\left(\varphi\left(\mathbf{p}_{\text{srg}}^{*,i}\right)\right)$  is used in the following. If the error between simulation and surrogate is sufficiently small, the optimum  $\mathbf{p}_{\text{srg}}^{*,i}$  is accepted and optimisation terminates, else the new observation is used for surrogate refinement, i.e.  $q_{\text{srg}}^i \rightarrow q_{\text{srg}}^{i+1}$ . Consequently,  $q_{\text{srg}}$  is not constant but adapts in each iteration  $i$  as indicated in the superscript. The procedure repeats until convergence.



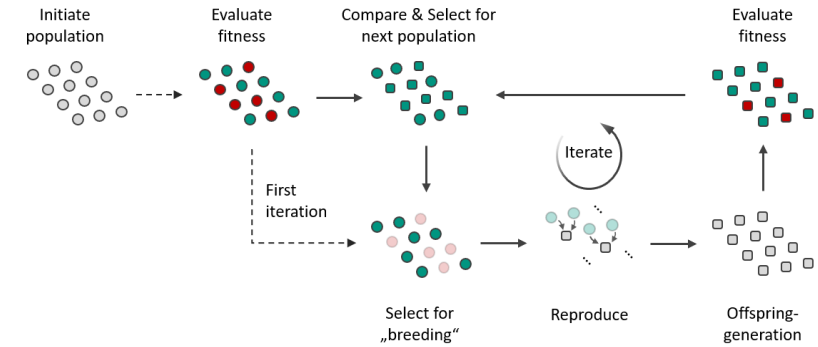
**Figure 3.1:** Schematic illustration of the employed workflow for surrogate-based optimisation (SBO) [152, 181]: After an initial sampling (database  $D$ ) and surrogate training (neural network), in each iteration  $i$  an optimiser identifies the optimum of the surrogate  $\mathbf{p}_{\text{srg}}^{*,i}$ , a simulation  $\varphi$  evaluates it and the new observation  $q(\varphi(\mathbf{p}_{\text{srg}}^{*,i}))$  is fed back to the database  $D$  for surrogate refinement. The procedure iterates until convergence or a stopping criterion.

The SBO-approach shows two key-features which require additional description: the *GA* and the *feedback loop* for sequential surrogate refinement. The section concludes with a minimal example to illustrate the optimisation procedure.

**Genetic Algorithm.** Since their introduction in the mid-70s [220], GAs and their derivatives have been continuously improved [221] and have proven robust and reliable in practice. Overall, GAs mimic evolution in nature and the according principle ‘survival of the fittest’ [222]. Figure 3.2 illustrates the underlying idea.

At start, a set of candidate solutions in the parameter space  $P$ , the ‘population’, is randomly initiated and each candidate (‘individual’) is evaluated for its objective function (‘fitness’). From this ‘parent’-population a reproduction step forms an ‘offspring’-generation of ‘children’. The exact implementations vary, but generally it comprises the following steps:

A set of individuals is drawn randomly from the parent-population. ‘Fitter’ individuals have a higher probability of being drawn. In a



**Figure 3.2:** Schematic workflow of a genetic algorithm (GA). The algorithm searches the design space for the most suitable solution by purposeful selection, combination and mutation of a pool (population) of candidate solutions (individuals).

‘crossover’-step, the selected individuals are randomly paired and combined to ‘children’. Ultimately, a mutation step randomly varies the children to increase solution diversity. Often crossover and mutation are done by bitwise logical operations on a binary representation of the input space [223, p215-216]. The number and place of these binary operations control the amount of crossover and mutation. The children-population is in turn evaluated and compared to the parent-population. Depending on their relative fitness (‘dominance’) some children replace their parents. The selection-reproduction-replacement-steps repeat until convergence or a triggering of a stopping criterion.

GAs offer a range of hyperparameters to control the optimisation process: For instance, larger population sizes and stronger mutation tend to increase solution diversity, i.e. facilitate parameter space *exploration*. Conversely, higher crossover rates and more assertive replacement-strategies promote *exploitation* of known solutions.

The optimal hyperparameter configuration is problem-specific and depends on a multitude of factors such as search space dimension  $n_d$  and problem properties. However, some configurations have empirically

proven useful [224] and established as default values [173]. For comparability with literature, the default values of the used software-package DAKOTA are used as summarised in Table 3.1. The values are assumed to facilitate convergence to the global optimum, even in higher-dimensional cases.

**Table 3.1:** Configuration of the genetic algorithm (GA) from the DAKOTA-toolkit used for optimisation of the surrogate.

Hyperparameter	Value	Hyperparameter	Value
Replacement type	Elitist	Crossover type	Multi-point binary
Convergence type	Fitness	Crossover rate	0.8
Population size	$10 \cdot n_d$	Mutation type	uniform
		Mutation rate	0.08

**Feedback loop for surrogate refinement.** In each iteration  $i$ , the GA optimises on the data-driven surrogate  $q_{\text{srg}}^i$ . However, being an approximation only, the according optimum  $\mathbf{p}_{\text{srg}}^{*,i}$  most likely deviates from  $q$ 's 'true' optimum  $\mathbf{p}^*$ . For surrogate refinement, an according simulation  $q(\mathbf{p}_{\text{srg}}^{*,i})$  yields the observation-tuple  $\{\mathbf{p}_{\text{srg}}^{*,i}, q(\mathbf{p}_{\text{srg}}^{*,i})\}$  to update the database  $D^i \rightarrow D^{i+1}$  and refine the surrogate. As this work employs artificial neural networks (ANN) the following passage concentrates on ANN-refinement.

During ANN-training, batch gradient-descents iteratively minimise the prediction error  $\varepsilon$  (cf. Equation (2.6)). Owing to its repetitive nature, the training can be easily resumed for a number of refinement-epochs  $n_{\text{rftn}}$  on the updated database  $D^{i+1}$  in order to evolve the surrogate  $q_{\text{srg}}^i \rightarrow q_{\text{srg}}^{i+1}$ .



Re-optimisation on  $q_{\text{srg}}^{i+1}$  then ought to give a refined optimum  $\mathbf{p}_{\text{srg}}^{*,i+1}$  with reduced prediction error  $\varepsilon(\mathbf{p}_{\text{srg}}^{*,i+1})$ . The procedure iterates until the difference  $\varepsilon(\mathbf{p}_{\text{srg}}^{*,i})$  between surrogate estimation  $q_{\text{srg}}^i(\mathbf{p}_{\text{srg}}^{*,i})$  and simulation  $q(\mathbf{p}_{\text{srg}}^{*,i})$  drops below a predefined threshold. The underlying assumption of this approach is that iteratively applying

$$\mathbf{p}_{\text{srg}}^{*,i} = \arg \min_{\mathbf{p} \in P} q_{\text{srg}}^i(\mathbf{p}) \quad (3.5 \text{ revisited})$$

converges  $q_{\text{srg}}^i(\mathbf{p}_{\text{srg}}^{*,i}) \rightarrow q(\mathbf{p}^*)$  for  $i \rightarrow \infty$ .

The ANNs of this work are configured so as to facilitate this assumption, mainly because of their ReLU activation function  $\psi_{\text{relu}}$ . The ReLU-activation is defined by  $\psi_{\text{relu}}(z) = \max(0, z)$ . Unlike other common activation functions, the ReLU is able to return an *exact* zero for a whole range of different input values, namely  $z < 0$ . This promotes a *sparse* instead of a dense propagation of activations through the network [225]. That is, only few, dedicated neurons activate and information travels along distinct paths through the network. Among others, sparse activations increase robustness to small input changes and foster output separability [225]. Conversely however, data which differs substantially from the training data may lead to under- or non-activations.

From a surrogate-perspective, this means that the ANN tends to *underestimate* the ‘true’ output in unknown parameter regions. Thus,  $|q_{\text{srg}}| \lesssim |q|$  can hold in these regions and the ANN may predict a deceptive ‘false minimum’. Note that this ‘false minimum’ does not reflect actual minimum of the objective  $q$  but stems from network non-activation due to local absence of observations, i.e. surrogate inaccuracy. Depending on the problem structure and dimensionality, the ANN can be riddled with such ‘false’ minima after the initial training. During the first optimisation iterations on the ANN, the GA can run into these false minima and large discrepancies to the simulation will be observed. Thus – loosely speaking – the difference  $\varepsilon(\mathbf{p}_{\text{srg}}^{*,i})$  can be thought of as an indirect and

qualitative indicator of surrogate uncertainty: The larger  $\varepsilon$ , the larger the surrogate uncertainty.

ANN-retraining on new observations sequentially eliminates these false minima and the according deviations reduce. Similar to optimisation strategies based on an expected improvement, e.g. the EGO-algorithm [184], the network at first *explores* unknown parameter spaces, i.e. eliminates the false optima, before closing in on the ‘true’ optimum (*exploitation*).

### 3.1.3 Minimal SBO-example

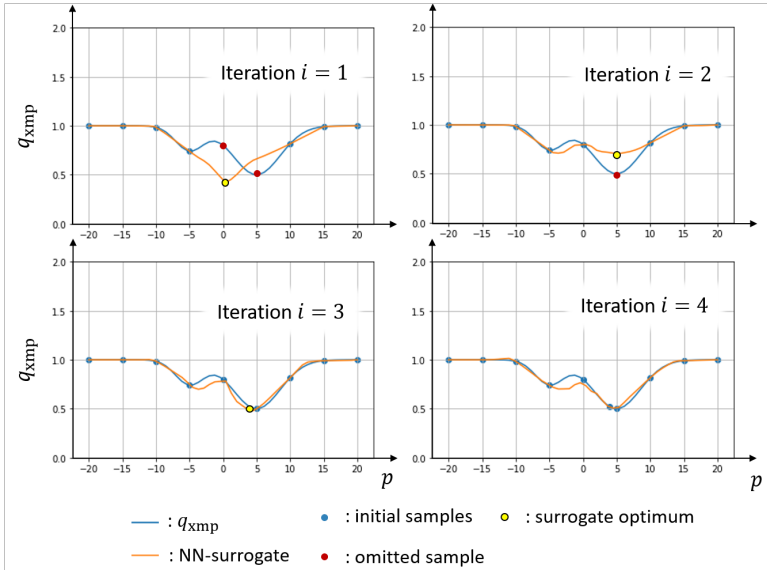
Figure 3.3 exemplifies the adopted SBO-principle for a generic, one-dimensional example objective function  $q_{\text{xmp}}$  (blue line). It is defined at haphazard by two shifted exponential bell-curves

$$q_{\text{xmp}}(\mathbf{p}) = q_{\text{xmp}}(p) = 1 - \frac{1}{4}e^{\left(\frac{p+5}{3}\right)^2} - \frac{1}{2}e^{\left(\frac{p-5}{5}\right)^2}. \quad (3.6)$$

and features a local and a global minimum at  $p = -5$  and  $p = 5$ , respectively. For ANN-training, it has been equidistantly sampled (blue markers) between  $-20 \leq p \leq 20$ . However, two samples near the global optimum ( $p = 0$  and  $p = 5$ , red markers) are deliberately withheld to generate a ‘non-sampled’ region.

After first training ( $i = 1$ , top left plot),  $q_{\text{srg}}^0$  (orange line) tends to zero in the non-sampled region and has a minimum at  $\mathbf{p}_{\text{srg}}^{*,i=1} \approx 0$  (yellow marker). The prediction  $q_{\text{srg}}^{i=1}$  deviates by  $\approx 50\%$  from  $q_{\text{xmp}}$ . Thus, it is a ‘false’ minimum from surrogate uncertainty rather than an actual optimum.

Retraining  $q_{\text{srg}}$  on the updated database shifts the surrogate optimum to  $\mathbf{p}_{\text{srg}}^{*,i=1} \approx 5$ . Again, it lies in the non-sampled region and the according evaluation reveals a substantial, yet smaller difference of  $\approx 30\%$ . The



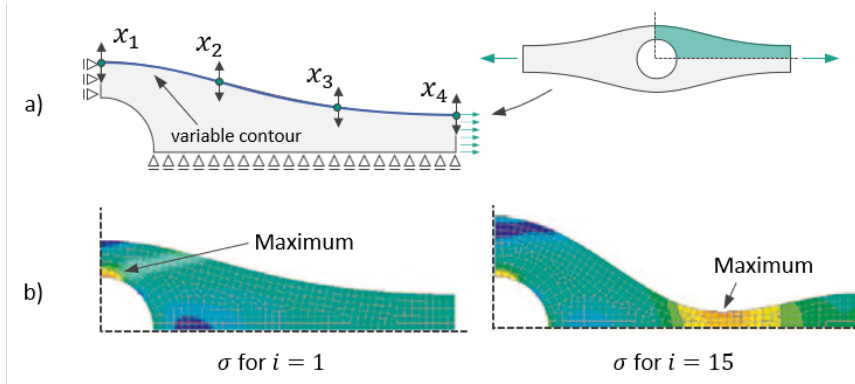
**Figure 3.3:** Generic example of the sequential refinement of an ANN with ReLU activation functions during SBO. At first, optimisations on the ANN yield ‘false minima’ in parameter regions with little data (red markers are withheld observations). However, according new observations sequentially eliminate them so that the optimisations finally close in on the true optimum.

following optimisation ( $i = 2$ ) yields again  $p_{\text{srg}}^{*,i=2} \approx 5$ . This time it differs by only  $\approx 4\%$ , which is deemed acceptable for termination.

### 3.1.4 Scalar and field surrogate

Any optimisation requires a scalar objective function to be minimised or maximised, respectively, and from a conceptual perspective, it is fully sufficient to approximate just this scalar objective. Consequently, classical surrogate techniques concentrate on constructing functions  $\mathbb{R}^{n_d} \mapsto \mathbb{R}$  only (cf. Section 2.3.3).

**Difficulties with scalar surrogates.** However, in some cases scalar surrogates can become problematic as SCHUMACHER [226, p 175-177] points out with an example. He considers a shape optimisation of a doubly-symmetric rod with a hole under tensile load as shown in Figure 3.4 a).

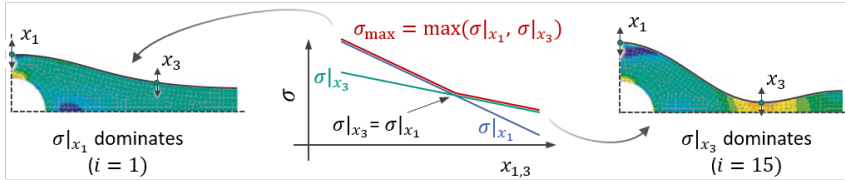


**Figure 3.4:** Surrogate-based shape optimisation of a rod with an abrupt stress maximum relocation from the central hole (iteration  $i = 1$ ) to the neck in iteration  $i = 15$  [226, p 175-177]. The relocation leads to a sudden change in parameter sensitivity which scalar surrogates usually struggle to capture (see also Figure 3.5).

The objective is to optimise the contour line parameters  $x_{1...4}$  for minimal mass without exceeding a maximum admissible Mises-stress  $\sigma$ . A surrogate models the according stress response. The initial design ( $i = 1$ ) on the left of sub-image b) locates the maximum stress near the hole, yet in iteration  $i = 15$  the stress maximum abruptly relocates to the neck of the rod (right).

SCHUMACHER reports difficulties during SBO. For simplicity, suppose the nominal stress  $\sigma|_{x_1}$  at position of  $x_1$  (blue) depended only linearly on  $x_1$  and likewise  $\sigma|_{x_3}$  (green) on  $x_3$  (Figure 3.5). Consequently, the *local* stress sensitivities with respect to  $x_{1,3}$  are constant. However, this does not hold for the global stress maximum (red): When it relocates ( $\sigma|_{x_1} = \sigma|_{x_3}$ ), its (global) sensitivity changes abruptly. Capturing this change with a scalar

(global) surrogate is difficult and requires many iterations for refinement and a sufficiently adaptive surrogate.



**Figure 3.5:** Visualisation of the sudden change of parameter sensitivity during stress relation for a scalar surrogate (red). A spatial resolution of the problem leads to constant sensitivities (green and blue) which are easier to learn.

As a remedy, SCHUMACHER suggests to split the part-domain into multiple sub-domains and train a separate surrogate for each sub-domain. Thereby, the (less complex) local sensitivity-information in each sub-domain must be captured only.

However, such an approach requires training of multiple surrogates and – more critical – an a-priori knowledge where the stress relocations occur. If such knowledge is unavailable, then a very fine partitioning could be used – ideally, one surrogate per element. However, this comes at the cost of training numerous separate surrogates and raises the more fundamental question, whether surrogates can be configured so as to predict multi-dimensional quantities, ideally a whole field, instead of just a scalar.

**Full-field surrogates.** Full-field estimations introduce additional information to the surrogate. For example, consider the rod in Figure 3.4. Clearly, the design variable  $x_1$  mainly affects the stress at the hole and  $x_3$  mostly the neck. Such local influence cannot be resolved in a global scalar metric, which consequently leads to a loss of information in the database, let alone engineering interpretability. However, training the

surrogate on field-data retains this information and may thus even increase accuracy and sample efficiency.

Different approaches exist to realise a multi-dimensional surrogate of the type  $\mathbb{R}^{n_d} \mapsto \mathbb{R}^{m_d}$  with  $m_d > 1$ . For instance, Gaussian Process Regression can be expanded to multiple dimensions ('Co-Kriging') [227]. Yet, they quickly become prohibitively computation-intensive even for only few output dimensions. Model reduction techniques can efficiently deal with higher dimensions and are often subsumed under the umbrella term 'proper orthogonal decomposition' (POD) [228]. In essence, POD-techniques decompose a dynamic field into a set of inherent 'modes' and seek to reconstruct the dynamic behaviour by superposition of these modes. The approach allows to give full-field estimations but is bound to unstructured input data, i.e. scalar input parameters. It cannot deal with structured input like images. As outlined in Section 2.3.4, this limits their expressiveness regarding information-encoding and thus, POD is taken out of consideration for this work.

ANNs allow for arbitrary input and output dimensions  $n_d$  and  $m_d$ , respectively, and can deal well with structured and unstructured input data. Surprisingly, ANNs for full-field estimations start only now to be applied in engineering: For instance, [229, 230] and the prior works [152, 181] use them to approximate costly FE-simulations.

This work compares both, scalar and full-field ANNs as Figure 3.6 illustrates. More precisely, the ANN  $q_{\text{srg}}$  is trained as a classical surrogate and mimics the *scalar* objective function

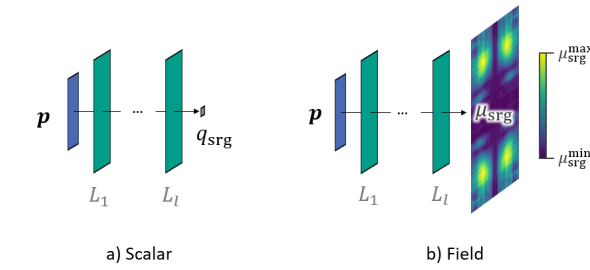
$$q_{\text{srg}}(\mathbf{p}) \approx q(\varphi(\mathbf{p})) \quad (3.4 \text{ revisited})$$

(cf. Section 3.1.1) and the ANN  $\mu_{\text{srg}}$  is trained to estimate the *field* quantity

$$\mu_{\text{srg}}(\mathbf{p}) = \mathbf{a}_{\text{srg}} \approx \mathbf{a} = \varphi(\mathbf{p}) \quad . \quad (3.7)$$

The scalar performance metric  $q$  can then directly evaluate the estimated field the same way it would evaluate an actual simulation:

$$q_{\text{srg}} = q(\mu_{\text{srg}}(\mathbf{p})) \quad . \quad (3.8)$$



**Figure 3.6:** Schematic illustration of a) a scalar ANN and b) an ANN for prediction of a full strain field.

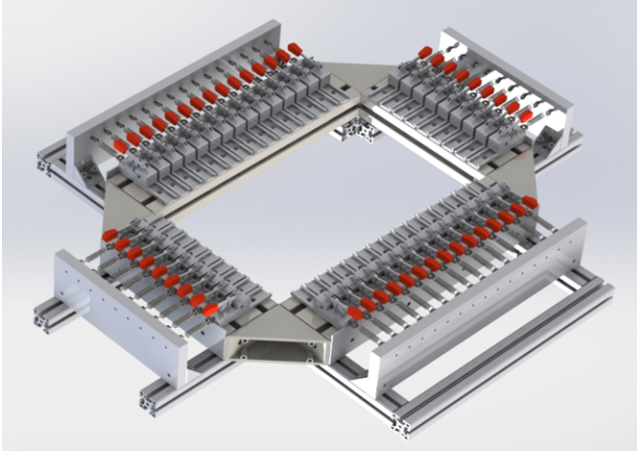
## 3.2 Validation of reference SBO-method<sup>1</sup>

In order to validate the applicability of the ANN-based SBO-approach, two key aspects are considered: After an introduction to the considered use case (Section 3.2.1), Section 3.2.2 compares the accuracy of two ANN-based surrogate strategies, namely for scalar and field-estimation. Section 3.2.3 then evaluates the effect of different update-strategies on the optimisation performance and compares the method to a classical GA.

<sup>1</sup>The presented results are published in the prior works [181, 219, 152]

### 3.2.1 Validation use-case: Gripper-assisted fabric forming

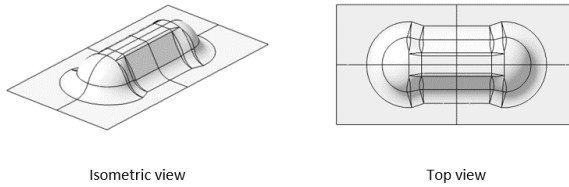
In order to validate the proposed ANN-assisted SBO approach, this work studies a high-dimensional optimisation forming example. To this end, a *gripper-assisted* forming process is considered which is inspired by [151] and the prior work [13, 231]. Figure 3.7 illustrates the process setup: Similar to the concepts presented in Section 2.2.2, membrane forces actuate the textile. More specifically, spring-guided clamps restrain the material flow into the mould.



**Figure 3.7:** Clamping frame (CAD drawing) developed by ALBRECHT et al. [231] which inspired the presented forming use-case.

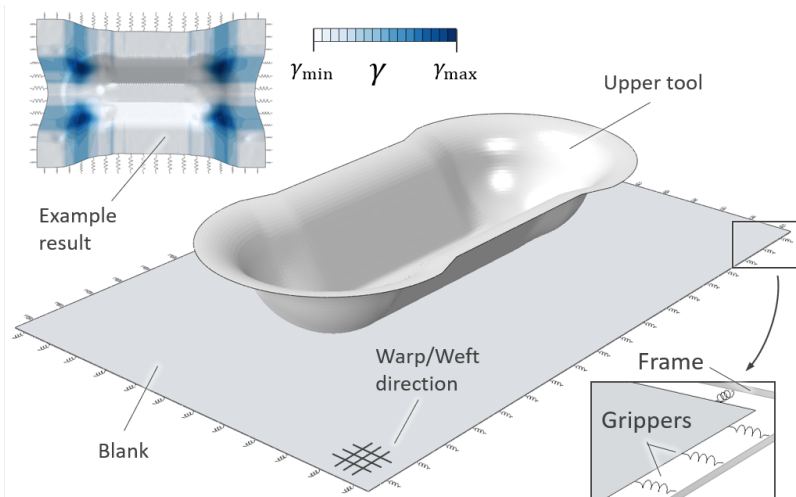
The considered geometry is the double-dome  $g_{dd}$ , a common benchmark geometry in textile forming. As Figure 3.8 shows, it features several forming-relevant characteristics, e.g. non-, single- and double-curved areas as well as concave and convex regions. Figure 3.9 visualises the according process simulation setup along with an exemplary forming result.





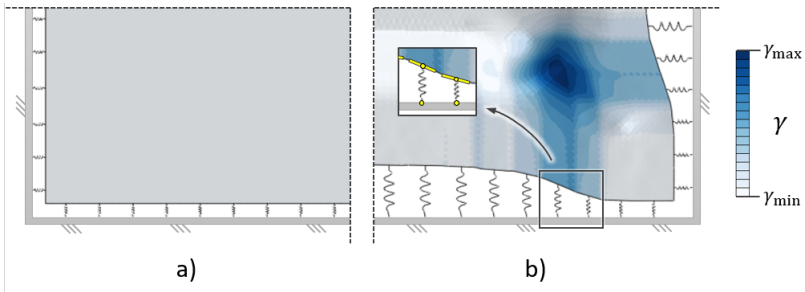
**Figure 3.8:** Double dome geometry  $g_{dd}$  as used in the presented use-case.

The blank measures  $560 \times 300$  mm; warp and weft are being oriented along the edges. The blank is discretised into  $n_{el} = 22080$  elements. In total, 60 clamps of 20 mm width each are evenly distributed along a surrounding frame. Hence, optimisation input space is 60-dimensional. The stiffness  $c_s$  of each spring can be varied between  $0.01 \frac{\text{N}}{\text{mm}} \leq c_s \leq 1.0 \frac{\text{N}}{\text{mm}}$ .



**Figure 3.9:** Illustration of the forming simulation setup with the clamping frame for validation of the SBO reference method. 60 circumferential grippers can actuate the textile during forming. The top left of the figure features an example forming result (shear distribution) for visualisation.

One end of the springs is attached to the textile, the other end to the frame (cf. Figure 3.10). Before the tool stroke, the springs are oriented perpendicular to the textile perimeter (sub-image a) ). The clamps are idealised to a universal joint, i.e. only the translational degrees of freedom are coupled. This allows the textile to rotate freely about the attachment points (yellow) as Figure 3.10 b) visualises.



**Figure 3.10:** Detail view on the clamping frame model in a) the initial configuration and b) after forming. The clamps are idealised as universal joints, i.e. only translational degrees of freedom are coupled.

The forming simulation uses a macroscopic FE-based approach, which captures the relevant forming defects of the studied fabric (cf. Section 2.2.3). Membrane and bending models have been parametrised by POPPE et al. [100, 38] for the balanced plain weave carbon fabric T700SC-12K-50C by ZOLTEK based on experimental characterisation. The material models are embedded within the commercially available FE-solver ABAQUS/EXPLICIT using VUMAT and VUGENS user-subroutines to account for textile-specific characteristics outlined in Section 2.2.3. ABAQUS' built-in general contact algorithm models the tool-ply interface. The tools are modelled as rigid surfaces and close in 2 s at constant speed.

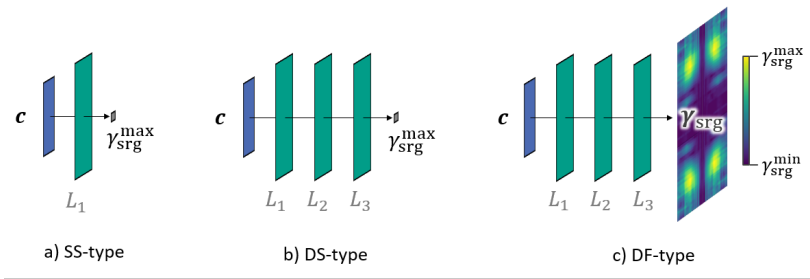
### 3.2.2 Comparison of scalar and field surrogate

Section 3.1.4 proposes that multi-dimensional surrogates can increase the accuracy and data-efficiency of surrogate models. This becomes especially useful, when multiple critical areas in a region of interest affect the objective function. In such cases a multi-dimensional output helps to resolve local effects of parameter changes.

Such spatial effects also occur in forming processes. In the considered validation use-case, the doubly-curved ends of the double dome  $g_{dd}$  evoke four separate, forming-critical shear zones as the example result in Figure 3.9 qualitatively shows. Grippers in vicinity of these zones and their opposite counter-parts along the yarn will affect these zones most [108], while remote grippers have almost no impact. For instance, the grippers on the top left corner will barely affect the shear zone in the bottom right corner and vice versa, i.e. each gripper actuates a certain region only. A scalar surrogate strategy cannot reproduce such local effects and requires more observations to reproduce the dynamics. Thus, changing from a scalar to a field surrogate should improve the prediction accuracy.

In order to validate this hypothesis, three different ANN-architectures are considered: 1) a shallow ANN with scalar output (SS-type), 2) a deep ANN with scalar output (DS-type), 3) a deep ANN with field output (DF-type). Figure 3.11 visualises them schematically. The SS- and DS-type are used to assess the effect of network depth; the DS- and DF-type then allow a separate assessment of scalar and full-field data.

The networks are regular feedforward ANNs with fully connected layers whose neurons use the ReLU-function for sparse activation. Under supervision of the author, SCHINDLER [232] performed extensive hyperparameter studies on these network types in his master thesis on a form filling problem and a forming problem. Among others, the network architectures were varied from 25 to 10 000 neurons per layer and 2 to 5 hidden layers.



**Figure 3.11:** Three different ANN-architectures considered as surrogates during this work: a) A shallow-scalar network (SS-type), b) a deep-scalar network (DS-type) and c) a deep-field network (DF-type) [181].

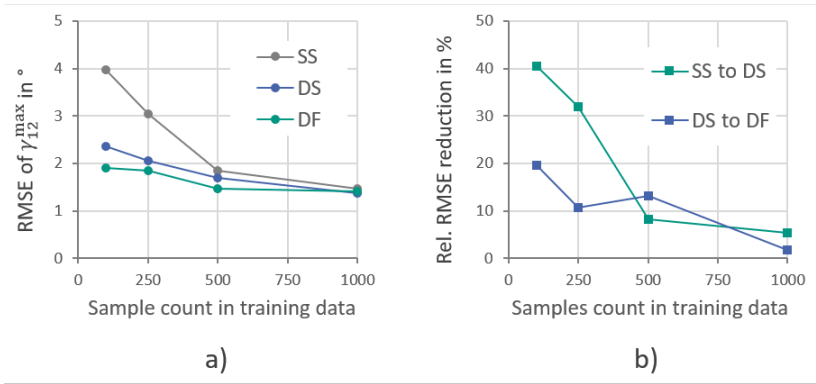
The final network architectures are summarised in Table 3.2. Note that the full-field predictions DF-type amount to predicting all  $n_{\text{el}} = 22080$  elements of the textile forming simulation. Since the number of network parameters is larger than the number of supplied observations, measures were required to prevent overfitting. To this end, SCHINDLER studied dropout, regularisation terms, mini-batches and batch normalisation (cf. Section 2.3.5). However, only the latter two turned out to improve the performance. See [232] for the comprehensive studies.

**Table 3.2:** Network architecture of the three ANN-types (SS, DS, DF) used for the reference SBO-method.

Type	Input	$n_{l=1}$	$n_{l=2}$	$n_{l=3}$	Output
SS	60	100	-	-	1
DS	60	2500	2500	2500	1
DF	60	500	500	500	22080

For evaluation of the networks' data efficiency, four different-sized training databases with  $n_k \in \{100, 250, 500, 1000\}$  samples are used. Each database has been sampled independently by a separate LHS, i.e. smaller

databases are not a subset of larger ones. The predictive accuracy is assessed on an additional, separate validation set with  $n_k = 100$  observations. The ANN-training continued for 500 epochs, while continuously measuring validation loss. After the training, the network parameters with the least validation loss were restored for evaluation. Figure 3.12 shows the results in terms of a) the absolute predictive error and b) its relative reduction when changing the network types.



**Figure 3.12:** Comparison of the absolute and relative accuracy of the three different network types SS, DS, DF. Sub-image a) shows the RMSE of  $\gamma_{12}^{\max}$  and b) relative error when changing the architecture.

More specifically, Figure 3.12 a) shows the evolution of root mean squared error (RMSE) of the maximum shear angle  $\gamma_{12}^{\max}$  for the validation set. That is, for the scalar ANNs (SS-, DS-type):

$$\varepsilon_{\text{rmse}} = \sqrt{\sum_{i=1}^{n_k} (\hat{\gamma}_{12,i}^{\max} - \gamma_{12,i}^{\max})^2} \quad . \quad (3.9)$$

Since the DF-type network predicts a full field, the maximum value is computed before analysis:

$$\varepsilon_{\text{rmse}} = \sqrt{\sum_{i=1}^{n_k} (\max(\hat{\gamma}_{12,i}) - \gamma_{12,i}^{\max})^2} \quad (3.10)$$

Sub-image b) relates the graphs by the relative error reduction of  $\varepsilon_{\text{rmse}}^{\text{rel}}$  when changing the ANN-type. For instance, when changing from SS to DS,  $\varepsilon_{\text{rmse}}^{\text{rel}}$  reads:

$$\varepsilon_{\text{rmse}}^{\text{rel}} = \frac{|\varepsilon_{\text{rmse}}^{\text{SS}} - \varepsilon_{\text{rmse}}^{\text{DS}}|}{\varepsilon_{\text{rmse}}^{\text{SS}}} \quad (3.11)$$

In accordance with the universal approximation theorem, for all network types the predictive error  $\varepsilon_{\text{rmse}}$  reduces as more observations become available until a final value of  $\varepsilon_{\text{rmse}} \approx 1.4^\circ$  is reached for  $n_k = 1000$  training observations. Interestingly, all networks reach approximately the same final value. This may indicate (numerical) noise in the validation set which cannot be learnt due to its aleatoric nature. However, large discrepancies become apparent when only few observations are used during training. Even though all networks are trained on the same data, the SS-type consistently shows a larger error  $\varepsilon_{\text{rmse}}$  (Figure 3.12 a) ) compared to DS-type. Consequently, the SS-type cannot extract the process inherent dynamics as efficiently from supplied data.

Figure 3.12 b) visualises the error reduction. Consider for instance the change SS  $\rightarrow$  DS: The introduction of additional layers reduces the error by  $\approx 40\%$  for  $n_k = 100$ . However, as more data becomes available, this advantage gradually diminishes – in loose terms: The SS-network catches up. These findings align with [233] and are often loosely summarised by the notion ‘wide models memorise, deep models generalise’. In sparse-data situations, the ANN must identify and generalise the data-inherent patterns for a accurate prediction. When more data becomes available, memorising and moderately adapting the most similar data-point appears to become sufficient. It may be noted that the SS-type

employs only  $n_l = 100$  neurons per layer, i.e. substantially fewer than the DS- and DF-type. This is because more neurons led to overfitting effects; presumably, since they mainly enhance the memorising-capacity but less the generalisation-capacity of shallow networks.

Shear relocations may explain the performance improvement of DF vis-a-vis DS. Analogous to the aforementioned stress relocations (cf. Figure 3.4) reported by SCHUMACHER [226, p 175-177], the shear maximum also jumps between corners depending on the current gripper configuration. Capturing these relocations in a mere scalar is difficult, a spatial resolution, however, helps to allocate the shear strain to specific corners.

Hence, changing the ANN-type from DS to DF further reduces the error by  $\approx 20\%$ . Resolving the results spatially (field estimation, DF-type) allows to attribute local effects to local grippers and indeed improves the accuracy. The observations corroborate the findings in [232] for RTM form filling and the prior work [152, 219] for textile forming.

Overall, the results hint that a deep network trained on full-field data (DF-type) yields the highest accuracy. The following section integrates the DF-type ANN into an SBO-routine and evaluates its applicability for optimisation.

### 3.2.3 Optimisation performance assessment

The DF-type network has proven most data-efficient to construct a surrogate. However, besides initial accuracy, for an effective SBO the surrogate needs to adapt and refine as new observations from the feedback-loops become available (cf. Figure 3.1). For this feedback loop two main strategies prevail: Samples can be either placed in parameter regions with sparse evidence in order to foster discovery of new, potentially better optima (exploration) or in most promising regions for perfection of the found parameters (exploitation).

To this end, SCHINDLER [232] has conducted extensive parametric studies in the context of RTM form filling under supervision of the author. He identifies two SBO-hyperparameters which dominate the convergence behaviour: The size of the initial training database  $n_k$  and the number of refinement-epochs  $n_{\text{rfn}}$ . As the previous Section 3.2.2 shows, larger values of  $n_k$ , i.e. more observations, yield a more accurate surrogate – presumably with fewer ‘false minima’ (Section 3.1.2) from non-activation of neurons. If so, then the optimiser does not have to eliminate them but can directly start exploiting the ‘true’ optimum, which should speed-up convergence. However, this allocates more computational effort to the initial sampling. Which effect dominates, is currently unclear.

The second hyperparameter, the number of refinement-epochs  $n_{\text{rfn}}$ , decides how fast the ANN adapts to a new observation, i.e. it promotes exploitation. The higher  $n_{\text{rfn}}$ , i.e. the more gradient-descents, the more emphasis lies on new observations and the stronger the attraction of an optimum. The effect of both parameters,  $n_k$  and  $n_{\text{rfn}}$ , is assessed in a full-factorial comparison on two levels:  $n_{\text{rfn}} \in \{2, 25\}$  and  $n_{\text{rfn}} \in \{100, 1000\}$ , respectively.

As introduced in Section 3.1.1, the vector norm Equation (3.1) constitutes the objective function:

$$q(\mathbf{a}) = \|\mathbf{a}\|_u = \left( \sum_{i=1}^{n_{\text{el}}} |a_i|^u \right)^{1/u} \quad (3.1 \text{ revisited})$$

and specifically for  $u = 4$  follows:

$$= \|\mathbf{a}\|_4 = \left( \sum_{i=1}^{n_{\text{el}}} |a_i|^4 \right)^{1/4} \quad (3.12)$$

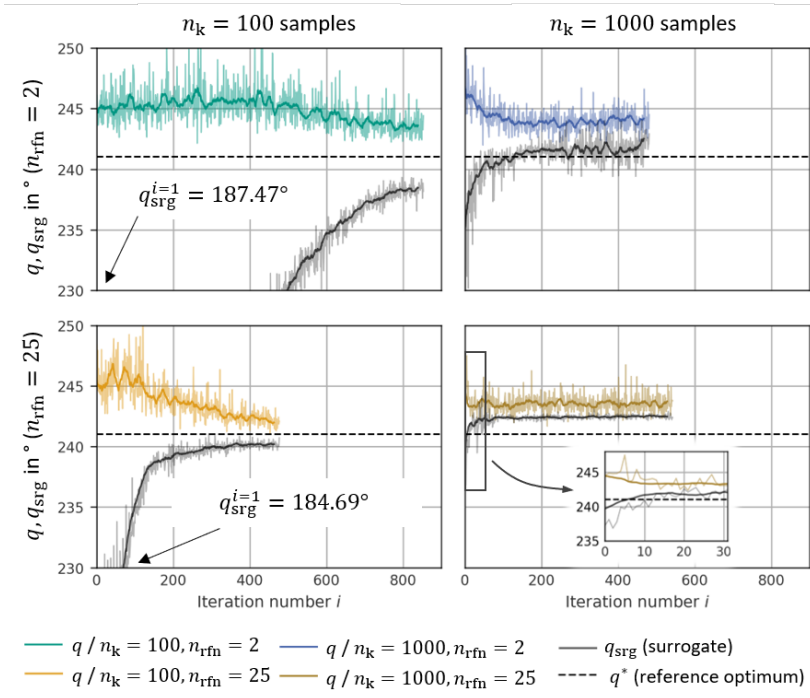
The exponent  $u = 4$  is used as it has empirically been found a useful compromise between penalisation of maximum shear and shear in general [152].



Figure 3.13 visualises the optimisation progress for each of the four combinations of  $n_{\text{rftn}}$  and  $n_k$ . Each plot features a coloured graph and a black graph. The coloured graphs quantify the objective function of the simulations, i.e.  $q(\mathbf{a})$ , the black graphs according the prediction of the surrogate  $q_{\text{srg}} = q_{\text{srg}}(\mathbf{a}_{\text{srg}})$ . Owing to their erratic appearance, a moving-average over 15 iterations (darker curve) smooths each graph. Both the original (bright) and the smoothed graph (dark) are plotted in diagrams. The dashed horizontal line is the best solution  $q^*$  across all simulations during this study. It is deemed the best approximation of the true, yet unknown optimum.

Abstracting from their volatility, the graphs share three common characteristics: 1) Overall, the objective function  $q(\mathbf{a})$  decreases at some point, i.e. the approach serves its optimisation purpose. 2) The surrogate predictions  $q_{\text{srg}}$  (black curves) eventually approach the simulation results  $q$  (coloured curves). That is, the surrogate indeed refines its accuracy and learns from new samples. 3) The surrogate  $q_{\text{srg}}$  approaches the simulation results  $q$  from below, i.e. it constantly *underestimates* the simulation results. This corroborates the false-minimum concept. 4) The approach occurs in a hyperbolic  $1/x$ -manner: At first it quickly improves its accuracy but the rate of improvement gradually reduces and eventually flattens out in an asymptote.

At the same time, the graphs also bear differences. Consider at first the plots for  $n_k = 100$  (left plots): The graphs  $q$  (simulation) show an initial wavering around  $q \approx 246^\circ$  before it starts to decrease. While the wavering continues until iteration  $i \approx 500$  for  $n_{\text{rftn}} = 2$  (top left), it only lasts until  $i \approx 130$  for  $n_{\text{rftn}} = 25$  (bottom left). Approximately at these iterations, the surrogate estimates the objective function to be  $\approx 233^\circ$ , i.e. has reached an accuracy of  $\approx 5\%$  deviation to  $q$ . From then on, the simulated results start to decrease, i.e. an actual optimisation progress can be observed. Both,  $n_k = 2$  (top) and  $n_k = 25$  (bottom) share the same shape, however, the graph for  $n_{\text{rftn}} = 25$  appears compressed to the left owing to faster adaption to new observations.



**Figure 3.13:** Optimisation progress for each SBO-configuration. Surrogate predictions  $q_{\text{srg}}$  are given in black (moving average) and grey, while the coloured graphs represent simulation results  $q$ . The initial  $q_{\text{srg}}$ -values of the left column are omitted in the plot for readability. Their respective values are directly printed for reference [181].

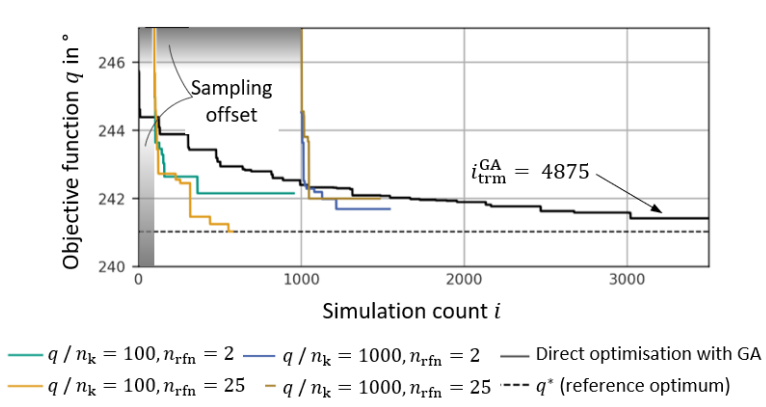
The behaviour is analogous to the minimal SBO-example in Section 3.1.3. At first, the optimiser on the ANN-surrogate runs into false minima and gradually eliminates them by retraining with according new observations (simulations). These false minima stem from surrogate inaccuracy and are generally far from the true optimum. Thus, the initial iterations may be interpreted as exploration. Once a sufficient accuracy is achieved ( $\approx 5\%$ ) the optimiser-results start converging to the true optimum.

The opposite behaviour can be observed for  $n_k = 1000$  (plots on the right). Unlike before, the objective function  $q$  (simulation) declines directly at the beginning before flattening out in an asymptote. For  $n_{\text{rfn}} = 25$  (bottom right), the decline phase is barely discernible so that graph looks almost horizontal throughout. Again the behaviour can be explained using false minima: Owing to the larger database, the ANN is already so accurate ( $< 5\%$ ) that only few false minima must be eliminated, i.e. little exploration required, and the optimiser directly converges to a true optimum. Yet it appears to be a local optimum, since in both cases (top and bottom right) the best found solution is still inferior to  $q^*$  (dashed horizontal line).

Figure 3.14 allows to compare the convergence behaviour. It shows the evolution of the objective functions  $q$  of each SBO-configurations and features additionally a graph of a non-surrogate, direct coupling of the simulation with a GA (black). The GA settings are identical to the surrogate-configurations reported in Table 3.1. The graphs quantify the ‘so-far-best’ value of objective function, i.e. the lower envelope of the erratic, coloured graphs in Figure 3.13. Unlike Figure 3.13, the plots account for the upstream sampling effort by an offset of  $n_k = 100$  or  $n_k = 1000$ , respectively.

The graphs summarise the results. Although the objective function  $q$  reduces faster for the large database ( $n_k = 1000$ ), their results are inferior. Not only objective-wise but also regarding efficiency. In total (sampling plus refinement), they take  $\approx 1100$  and  $\approx 1300$  simulations to reach their final  $q$ -value, while ‘ $n_k = 100, n_{\text{rfn}} = 25$ ’-configuration takes only  $\approx 600$  simulations for an even improved result. Its final value even slightly outperforms the GA-reference. Overall, this hints that online-sampling (during SBO) provides more informative samples than a-priori offline sampling.

Please note that ‘ $n_k = 100, n_{\text{rfn}} = 2$ ’-configuration could have achieved a similarly good result but was manually terminated due to excessive computation time ( $> 10$  weeks). At this time, the objective function was still descending, albeit at a low rate (Figure 3.13 top left). Also note



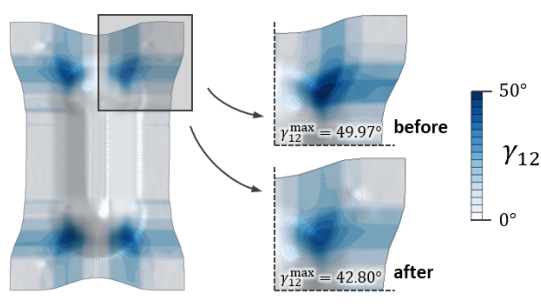
**Figure 3.14:** Convergence comparison of the studied SBO-configurations and a non-surrogate, direct optimisation using a genetic algorithm (GA). The GA terminated after 4875 iterations without further improvement (omitted for readability) [181].

that each SBO-variant outperformed the directly coupled GA in terms of efficiency. The GA took more than 3000 simulations until its final value and auto-terminated after 4875 simulations.

Ultimately, Figure 3.15 shows the shear angle distribution for engineering interpretation and clearly a substantial reduction of the maximum shear angle  $\gamma_{12}^{\max}$  is achieved. During SBO it dropped by 14.3% by restraining the local material take-in so that shear deformation stretches over a wider area instead of concentrating locally.

### 3.3 Intermediate summary

This chapter studies deep neural networks as surrogate models for high-dimensional, virtual process optimisation. The studied use-case is gripper-assisted forming. Three different surrogate-approaches are compared. The best performance was achieved with deep neural network which predicts the full strain field instead of a mere performance



**Figure 3.15:** The forming results before and after optimisation show a successful reduction of the maximum shear angle  $\gamma_{12}^{\max}$  [181].

scalar. The obtained network is integrated into an SBO-framework and four hyperparameter settings with different exploration-vs-exploitation strategies are studied. A state-of-the-art GA serves as an additional benchmark.

In each case, the SBO-framework outperformed the GA in terms of efficiency. One configuration even determined an improved optimum. The results also hint that online-samples obtained for surrogate-refinement are more informative than a-priori determined offline-samples. This leads to the insight that a smaller database – and thus an initially less accurate surrogate (!) – can indeed lead to better optimisation results. Note however, that the surrogate is only valid for this specific process-setup only. That is, a process modification, e.g. a geometry variation, instantly invalidates the surrogate and requires resampling of data and retraining the surrogate. Approaches for generalisation will be studied in the following chapters.

Overall, the results presented in this section show, that the available SBO-framework is able to successfully reduce an objective function, even in high-dimensional cases. It will be used as a reference for a generalised surrogate for flexible geometries.



## 4 Formability estimation

Section 3.2 shows that SBO helps reduce optimisation efforts compared to direct optimisation. However, the according surrogate models are application-specific and fall short on re-usability in new scenarios, e.g. after a geometry revision. This work proposes recent ML-techniques to enhance surrogate-capabilities towards variable manufacturing situations<sup>1</sup>. More precisely, the overall aim is to train a geometry-informed model which can efficiently guide process optimisations of variable component geometries.

To this end, this chapter studies concepts for ML-assisted *formability assessment* for variable geometries. Chapter 5 then presents an ML-approach for *optimisation* of according process parameters.

### 4.1 Approaches for formability assessment

An automatic evaluation of part manufacturability, as aspired in this work, requires at first a surrogate-readable geometry representation. For this purpose, two overarching concepts have been developed in the past, namely *local* and *global* descriptions, whose suitability mainly depends on the manufacturing process.

Local approaches often describe the geometry by a collection of independent points with fundamental properties such as position and principal

---

<sup>1</sup>For a term-wise distinction, the term ‘surrogate’ refers generally to classical surrogate models with unstructured input and ‘ML’ to models with structured input

curvatures. They are thus well-suited for manufacturing processes with a point-wise characteristic, e.g. metal driving or incremental forming [234, 235, 236].

However, local approaches remain unaware of interactions of neighbouring geometric features as observed in stamp forming. Thus, *global* geometry descriptions are required for this work which can describe the component as a whole – for instance by parameters like width, length or diameter. Additional process manipulations by e.g. grippers are excluded in this chapter (free forming) as only geometry variations are of interest. The methods split into *feature-based* and *grid-based* descriptions. As both are compared in this work they are outlined in the following Sections 4.1.1–4.1.2.

### 4.1.1 Feature-based formability assessment<sup>1</sup>

Feature-based geometry descriptions seek to retrieve and quantify typical geometry characteristics ('features') in the component such as bends, cutouts or corners. Once quantified, the features' formability can be assessed. Owing to the larger community, most of the prior work stems from sheet metal forming. However, despite long-standing analyses [239, 240, 241, 242, 243, 244], only general trends have been detected and overall an efficient predictive tool for a process designer still lacks [245]. Only recently and partly based on the prior work [238], ATTAR et al. [245] have developed a set of fitted empirical equations to predict forming responses during sheet metal forming of rectangular cups.

Similar holds for textile forming: For simple geometries like cylinders or spheres analytical formulae have been derived to estimate the maximum shear angle [77, 78]. Also, geometry analysis tools have been developed, which compare draping paths to geometry contour lines [246]. Alternatively, inverse methods have emerged which revert the design process:

---

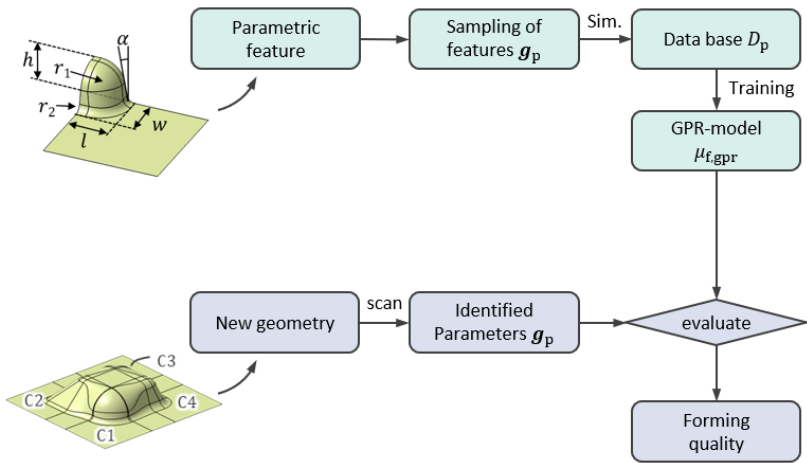
<sup>1</sup>The presented approaches are based on the prior works [237, 238]



Instead of assessing a given geometry, they generate a catalogue of well-manufacturable shapes, so-called ‘curve-glide’ geometries, which serve as inspiration during the design process [247].

However, for both sheet metal forming and textile forming, no predictive tool other than simulations is currently available to efficiently assess the formability of geometry features. Thus, the following sections present a surrogate approach to close this gap.

**General workflow and assumptions.** Figure 4.1 visualises the overarching workflow. It consists of two main phases: *training* (green shade) and *application* (blue).



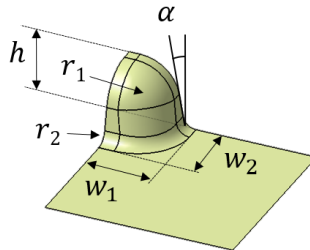
**Figure 4.1:** General workflow for geometry assessment [238]. The workflow comprises two phases: a training phase (green) and an application phase (blue). Training involves defining, sampling and simulating parametric geometry features  $\mathbf{g}_p$  for training a surrogate  $\mu_{f,\text{gpr}}$ . During application, a new component geometry is scanned for the predefined features. Identified geometry parameters  $\mathbf{g}_p$  can then be evaluated by the surrogate  $\mu_{f,\text{gpr}}$ .

During training, a set of geometry feature parameters  $\mathbf{g}_p$  is sampled, their forming quality metrics  $q$  determined (simulation) and collected in

a tuple  $\{g_p, q\}$ . A geometry-process-database  $D_p = \{\{g_p, q\}_1, \dots\}$  stores these tuples for surrogate training. During application, the component is scanned for relevant geometry features and each feature's parameters are retrieved. Then, the surrogate  $\mu_{f, \text{gpr}}$  evaluates them for the expected forming result.

The surrogate  $\mu_{f, \text{gpr}}$  itself is based on Gaussian Process Regression (GPR) as it has proven superior performance in a comparative study [237]. GPR is a non-parametric universal approximator based on Bayesian statistics theory. In essence, it aims at inferring the most likely distribution, that the observations in  $D_p$  have been sampled from. See [248] for details on GPR model theory and [238] for model and hyperparameter details.

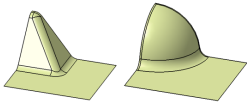
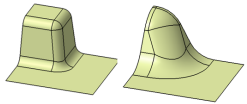
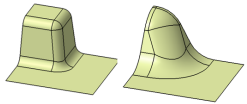
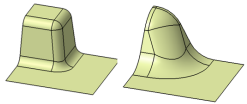
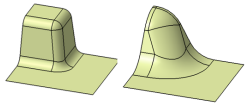
**Data curation.** To start with, this work assumes a *parametric* rectangular cup geometry with variable corners inspired by [249, 250] and the NUMISHEET93-benchmark [251]. Figure 4.2 visualises the corner parameterisation scheme  $g_p$  (quarter geometry).



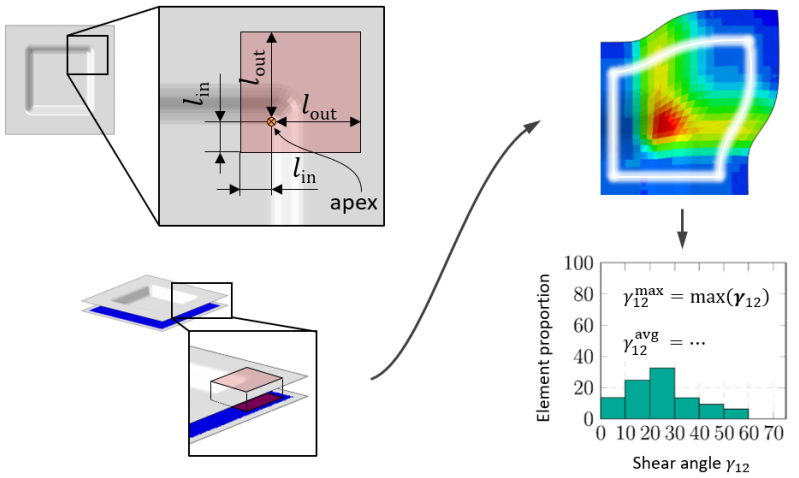
**Figure 4.2:** Visualisation of the parametric geometry (corner) considered in this work [237, 238]. The parameterisation scheme allows for a wide range of corner geometries, e.g. pyramids, cuboids or spherical corners (cf. Table 4.1).

It aims at a wide coverage of possible corner geometries ranging from a box-shaped geometry to a pyramid or spherical geometry in variable sizes as visualised in Table 4.1.

**Table 4.1:** Explanation of geometry parameter set  $\mathbf{g}_p$ , the considered parameter range and corner geometry examples for visualisation [238].

Symbol	Meaning	Range	Examples
$r_1$	Top radius	5...40 mm	
$r_2$	Bottom radius	5...40 mm	
$\alpha$	Draft angle	5...40 deg	
$w_{1,2}$	Width, length	50...300 mm	
$h$	Height	50 mm	

The geometry vector  $\mathbf{g}_p = (r_1, r_2, \alpha, w_1, w_2, h)^T$  collects the feature parameters and serves as surrogate input. Simulations of the resulting shear angle distribution then quantify the formability and are meant as output. Figure 4.3 illustrates the procedure.



**Figure 4.3:** Evaluation principle for the corner geometries [238]. An evaluation region is defined in such a way that it covers the main deformation region of the fabric. From this evaluation region the fabric forming quality (shear angle  $\gamma_{12}$ ) is extracted as a histogram and statistical metrics determined, e.g. maximum or mean.

Around the corner's apex an evaluation region is defined by  $l_{\text{in,out}}$  (highlighted in red) for each geometry. Its size is not fix but depends on the corner parameters  $g_p$  and grows with  $h$ ,  $r_1$  and  $r_2$  [238]. This ensures that the evaluation region always fully encloses the main deformations independent of the current corner geometry setting. Elements inside the projection are collected in an evaluation set. Their shear angles  $\gamma_{12}$  are sorted into a histogram and statistical metrics are computed, most notably the maximum shear angle  $\gamma_{12}^{\text{max}}$ . The procedure repeats for all geometries and all observations are stored in a database  $D_p = \{(g_p, \gamma_{12}^{\text{max}})_1, \dots\}$  for surrogate training.

**Assumption of independent strain fields.** The presented approach assumes that strain fields of geometric features form independent of each other. This assumption is inspired by SAINT-VENANT's Principle which essentially states that local stress- or strain-peaks attenuate with distance. More formally, it states that the order of magnitude  $m_s$  of a strain component in a loaded body decays by

$$m_s \propto \frac{\mathcal{S}}{d_s^{p_{sv}}} \quad , \quad (4.1)$$

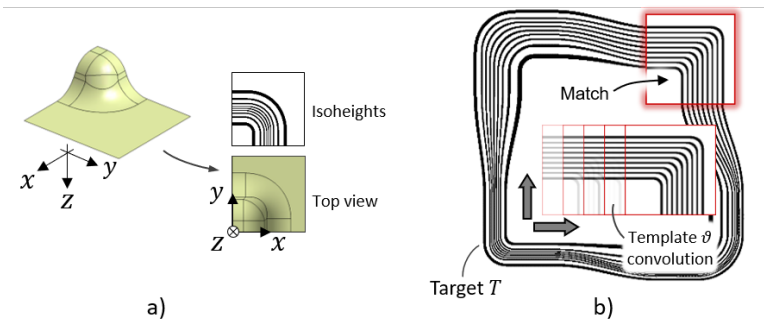
wherein  $\mathcal{S}$  is a conceptual surface enveloping the cause of the strain (corner) and  $d_s$  is the distance from a point of interest to the centre of  $\mathcal{S}$  [252, p 359-362].

The decay exponent  $p_{sv}$  is load case dependent, with  $p_{sv} \geq 0$  in general. Clearly, with increasing distance ( $d_s \rightarrow \infty$ ) or higher values of  $p_{sv}$ ,  $m_s$  decreases and thus, strain fields localise in vicinity of their cause. The value of  $p_{sv}$  depends on the resulting loads transferred across  $\mathcal{S}$ , while in general, vanishing resultant loads yield higher values of  $p_{sv}$ . In other words, the strain state at a point of interest should remain unaffected of other strain fields in the body given either 1) a sufficient distance or 2) vanishing resultant loads. That is, if two corners lie sufficiently far apart, they can be assessed individually. Otherwise, a more complex

surrogate for interacting strain fields is required. A numerical study is presented in Section 4.2.1 for validation of this assumption.

**Feature recognition.** Integral part of the algorithm is the ability to recognise features in a given geometry. To this end, an image-based approach is selected. From a manufacturing perspective, formed geometries must necessarily be undercut free to allow for collision-free tool closure. Therefore, without loss of information, a bijective projection of the geometry into the tool-plane is tractable, which can be encoded in an image. Besides an advantageous dimensional reduction to a 2D representation, an image-based representation enables access to numerous analysis-techniques, specifically object recognition, classification and tracking.

Figure 4.4 illustrates the concept. This work uses a topographic map with contour lines of constant elevation (isoheights). Beginning from the top, isoheights divide the geometry in user-specified steps in tool direction.



**Figure 4.4:** Visualisation of the image-based geometry recognition scheme for feature-based formability assessment: The geometry of interest is encoded in a target-image by contour lines of constant elevation (isoheights) [238]. Likewise, feature templates with known geometry parameter  $g_p$  mapped to images. These template-images are convolved across the target until a match is observed and the parameters of the matches are extracted from a lookup table.

The geometric features must be detected irrespective of their position in the component. Consequently, translation invariant feature recognition is required. Note that rotation invariance is not required in this work as forming results depend on the relative angle between fabric and geometry feature due to material anisotropy. Also scale invariance is not pursued since pixels encode distance information. Thus, images must not be scaled.

Inspired by image processing in neural networks, convolving templates  $\vartheta$  across a target image (investigated geometry) ensures translation invariance. The templates  $\vartheta$  stem from a large geometry repository with known geometry parameters  $g_p$ . Figure 4.4 b) visualises the convolution operation: It can be interpreted as sliding a template  $\vartheta$  over the target image  $T$  until a match (i.e. high correlation) is observed. If a template matches, the geometry parameters are extracted from a lookup table and parsed to the GPR-surrogate  $\mu_{f,gpr}$  for evaluation.

Since pixels are discretised values, a discrete convolution operation is performed yielding the normalised 2D cross-correlation [253]:

$$C(u, v) = \frac{\sum_x \sum_y [T(x, y) - \bar{T}_{xy}] [\vartheta(x - u, y - v) - \bar{\vartheta}]}{\sqrt{\sum_x \sum_y (T(x, y) - \bar{T}_{xy})^2 \cdot \sum_x \sum_y (\vartheta(x - u, y - v) - \bar{\vartheta})^2}} \quad (4.2)$$

wherein  $T(x, y)$  is the target image,  $\bar{T}_{xy}$  the average of  $T(x, y)$  under the template  $\vartheta$  positioned at  $(u, v)$ . Summation range for  $x$  and  $y$  is the region under  $\vartheta$ . For increased recognition robustness, in this work, the isoheights are additionally thickened by two pixels to allow for minor mismatches between template and target.

### 4.1.2 Grid-based formability assessment<sup>1</sup>

The presented feature-based approach conceptually allows to assess the formability of a component by analysis of individual, feature parameters (*unstructured* input). It offers some appealing advantages; most notably, it employs engineering-interpretable parameters like radii or draft angles. However, the need for predefined parameterisation scheme limits the scope of geometry variations: Any variation outside this parameter scheme, however small, cannot be modelled.

Also, in its current form, the approach requires geometry features to be sufficiently far apart so that they can be analysed individually (SAINT-VENANT'S Principle). Otherwise, a more advanced surrogate with additional parameters to capture interaction effects would be required. Yet, such an advanced model immediately introduces more complexity to the geometry parameter space.

Literature reports attempts for such advanced geometry schemes in the context of sheet metal forming [255, 256]; yet in practice, the implementations prove unstable and susceptible to modelling errors [188]. Overall, feature-based geometry-descriptions may be suitable for niche applications with little and well-defined variability but appear of limited universal applicability.

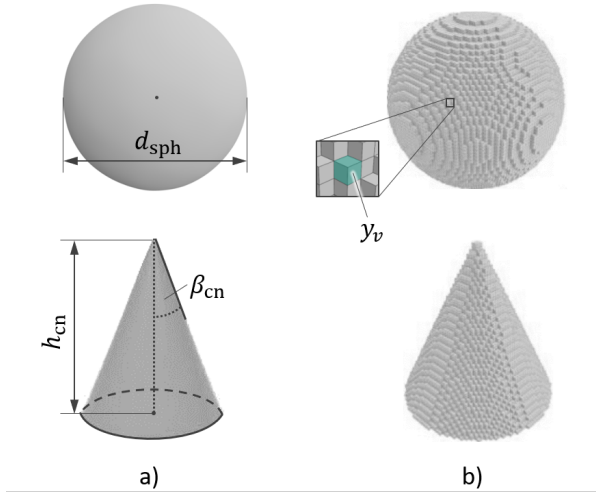
As a remedy, *structured* geometry encoding schemes have been proposed [210, 257], namely grid-based representations. They not just allow for a more flexible input space but also allow to estimate whole field-quantities instead of mere scalars. The following paragraphs present their underlying concept, the general workflow for integration into forming analysis and the according ML-techniques to process them.

**Data structure.** Structured encoding schemes suggest to give-up on distinct geometry parameters, i.e. unstructured input (cf. Section 2.3.4),

---

<sup>1</sup>The presented approaches are based on the prior works [211, 254]

in favour of a grid-based, i.e. *structured*, representations by voxels or pixels [210, 257]. Figure 4.5 juxtaposes both at the example of a sphere and a cone to outline their differences.

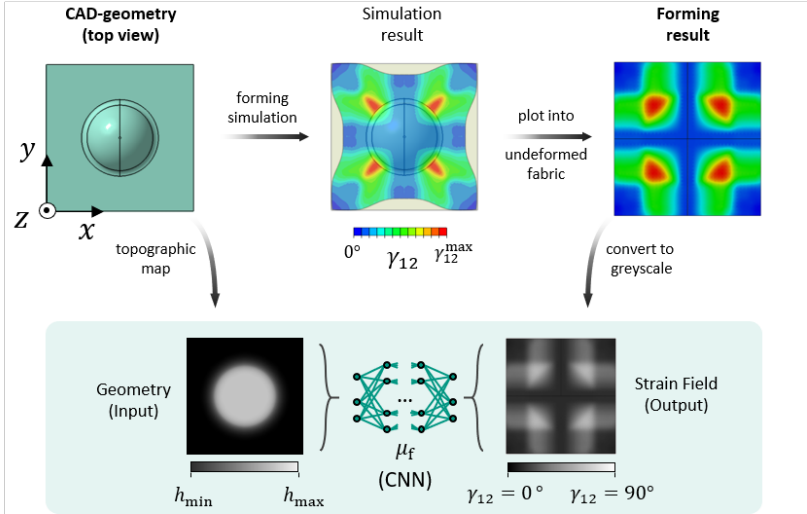


**Figure 4.5:** a) Unstructured (feature-based) and b) structured (grid-based) geometry representation by voxels at the example of a sphere and a cone [258, p 2].

The feature-based representation (sub-image a) ) requires only few parameters to define each geometry, namely the diameter (sphere) and height and opening angle (cone). In contrast, the grid-based representation requires  $n_v = 50^3 = 3\,125\,000$  (binary) voxel-parameters  $y_v$ ,  $v = 1 \dots n_v$  at the given discretisation level, i.e. substantially more parameters. However, the grid-representation can reproduce both geometries, while the feature-based representation needs to redefine the parameters. Thus, the grid-representation is more versatile but requires handling of a considerably higher number of parameters.



**Data representation and general workflow.** Figure 4.6 visualises the encoding scheme for geometry (input) and forming result (output) to enable supervised training of an grid-based ML-algorithm  $\mu_f$ .



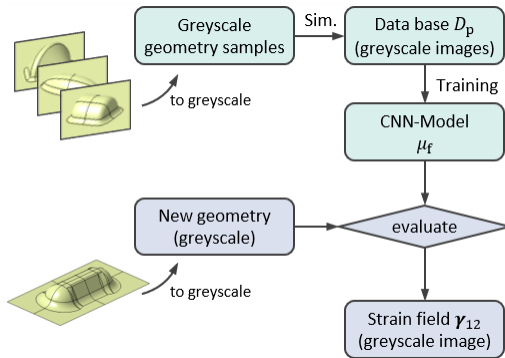
**Figure 4.6:** Image-based geometry and forming result encoding for supervised training of a convolutional neural network (CNN): In textile forming, a close spatial relation between geometry curvature and forming result is observed, which can be efficiently encoded in a greyscale-image. A CNN with an autoencoder structure can learn these relations by supervised training (image-to-image regression) [238].

In general, the geometry could be encoded using voxels as shown in Figure 4.5. However, analogous to the feature-recognition step in Figure 4.4, a convenient reduction to a two-dimensional image-representation is admissible. More specifically, greyscale values quantify the local elevation  $z(x,y)$  above the tool-plane similar to a topographic map. Likewise, grey-scale values represent the material strain field of a simulation with each pixel representing an element's shear angle  $\gamma_{12}$ . To allow for a representation in a rectangular image, the shear strain is mapped to the initial configuration, i.e. the *undeformed* (flat), rectangular fabric.

In case of textile forming, a close correlation between component curvature and material strain can be observed as shown at the top of Figure 4.6. Starting from the centre, the mild double-curvature of the hemisphere evokes only moderate shear strains, yet they increase abruptly at the comparably sharp transition to the plane outer area.

Such spatial relations often govern manufacturing dynamics: Not just in fabric forming but also in fully unrelated processes like moulding. For instance, clearly, a strong correlation between flow behaviour and geometry exists. While these spatial relations could get lost with an unstructured representation through geometry parameters, structured, grid-based representations allow to retain them efficiently.

Similar to the feature-based approach, the grid-based workflow comprises two steps as Figure 4.7 shows: At first, a set of geometry-process-observations is sampled (training database  $D$ ) and the ML-model  $\mu_f$  learns the input-output-dynamics (green shade). After training, the algorithm then assesses a new geometry (blue).



**Figure 4.7:** General workflow for grid-based formability assessment. The workflow comprises two phases: a training phase (green) and an application phase (blue). Training involves sampling and simulating geometries to build a database (greyscale-images) for supervised training of the CNN  $\mu_f$ . During application, a new geometry is converted to a greyscale-image and evaluated by  $\mu_f$ . Note that, unlike the feature-based approach, whole geometries instead of isolated features are considered.

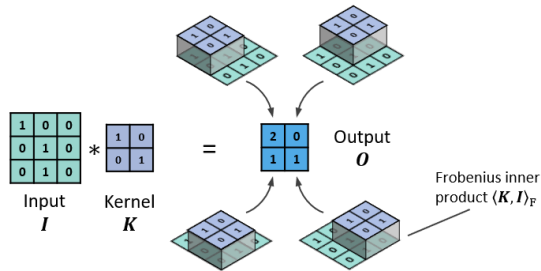
Note however, that – unlike the feature-based approach – the workflow thus does *not* require a recognition step to extract geometry parameters. The evaluated components use the same grid-based description as the training samples and can thus be directly analysed.

Overall, the aspired image-based representation offers two promising properties: It allows 1) for a potentially, universal geometry description and 2) to resolve local process phenomena instead of a abstract performance scalar. Some ML-techniques have specialised in analysing such high-dimensional, grid-structured data - namely convolutional neural networks (CNN). CNNs can be configured to provide such image-to-image mappings, for instance in so-called ‘autoencoders’ [259] or ‘U-nets’ for image segmentation [260]. The following paragraphs outline the underlying concept.

**Convolutional neural networks.** As the name hints, CNNs ‘slide’ (convolve) smaller-sized filter matrices (‘kernels’)  $\mathbf{K}$  across an input grid  $\mathbf{I}$  while continuously measuring the local degree of coincidence between  $\mathbf{I}$  and  $\mathbf{K}$ . The output  $\mathbf{O}$  is again a grid-structure and typically referred to as ‘feature map’. According to [202, p 328], a discrete convolution can be defined for images by

$$\mathbf{O}(i_p^*, j_p^*) = \mathbf{I} * \mathbf{K}(i_p^*, j_p^*) = \sum_{m_p} \sum_{n_p} \mathbf{I}(i_p + m_p, j_p + n_p) \mathbf{K}(m_p, n_p) \quad . \quad (4.3)$$

Therein,  $i_p, j_p, i_p^*, j_p^*$ , denote the pixel indices of input and output image, respectively, while  $m_p, n_p$  denote pixel indices of the kernel. Essentially, Equation (4.3) computes the FROBENIUS inner product  $\langle \mathbf{K}, \mathbf{I} \rangle_F$  of the kernel  $\mathbf{K}$  and the ‘covered’ part of image  $\mathbf{I}$  at position  $(i_p^*, j_p^*)$ . The summation range for  $m_p$  and  $n_p$  is the kernel size. Figure 4.8 visualises the convolution operation with a simple example.



**Figure 4.8:** Schematic visualisation of a convolution operation as an array-ordered sequence of FROBENIUS inner products.

Essentially, the kernel  $K$  slides pixel-wise across the input image, computes  $\langle K, I \rangle_F$  at each position and inserts the result into the output feature map  $O$ . Higher values of  $\langle K, I \rangle_F$  denote higher local correlation. In the given example, the kernel-position with the highest coincidence is the top left corner.

The procedure resembles the template-based feature recognition approach in Section 4.1.1 but differ in two major regards: First, the templates represent one specific geometry-configuration, while the CNN-kernels are more abstract greyscale-contours and only their purposeful combination may uniquely identify a geometric feature. Secondly, the templates are handcrafted but the CNN-kernels are learnt during CNN-training.

Each kernel scans the input space for one specific greyscale-characteristic. Consequently, using many kernels can encode and retrieve many and more complex greyscale (geometry) characteristics. Thus, convolution operations only unfold their potential when multiple kernels are used. Also, the output feature maps themselves can again be used as an input for a subsequent convolution. Thereby ‘higher-order features’ form and encode more complex features, for instance the relative position of two different features. Overall, the hierarchical CNN-architecture provides much more expressiveness compared to the earlier template-based feature-recognition approach.

Among others, the number of convolutional layers, the number of kernels per layer, their sizes, types as well as their step size ('stride') are user-defined hyperparameters. Note however that – unlike the example – the *entries* of the kernels are *not* user-defined but tunable model parameters. They are randomly initialised and adapt during the CNN-training to reproduce the training database, similar to the weights and biases in fully connected networks. See [202, p 326-366] for details on CNNs and their training.

In practice, inevitable 'side-activations' are filtered out by a subsequent pooling operation, often *max-pooling*. To this end, a kernel slides across  $O$ . Yet this time, it extracts the maximum value instead of  $\langle K, I \rangle_F$ . Additionally, the kernel-positions for pooling are usually non-overlapping. That is, two neighbouring kernel positions are not one pixel but a kernel-width apart. The distance between two kernel positions is usually referred to as 'stride'. Figure 4.9 visualises the operation for a  $2 \times 2$ -kernel on a  $4 \times 4$  convolution output  $O$ . At each position (colour-coded), the kernel extracts the maximum (bold numbers) and stores it in the corresponding entry of the output matrix  $\hat{O}$ . Thereby, it reduces the output dimensions and compresses information as only the largest activations enter the pooled feature map  $\hat{O}$ .

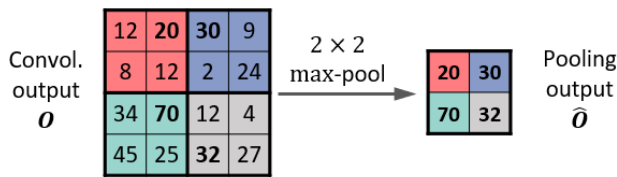
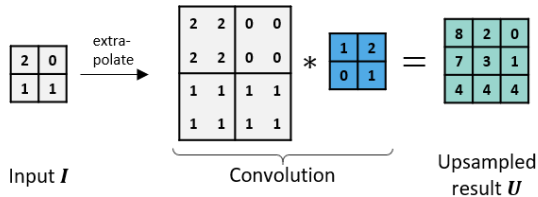


Figure 4.9: Schematic illustration of the max-pooling operation.

Setting aside special measures to artificially expand the image boundaries ('padding'), convolutions and pooling operations typically compress information and *reduce* ('downsample') the image size as shown in Figures 4.8–4.9: The output is smaller than the input. Thus, many CNN-based algorithms require at some point inverse *upsampling* steps to

restore the image-size again. This is also the case in this work as it seeks to estimate whole strain fields.

Two different upsampling-methods dominate: 1) transposed convolutions and 2) nearest-neighbour extrapolation followed by a regular convolution. In his master thesis under supervision of the author, TRIPPE [261] identified the latter as most promising. As it is used in this work it is briefly outlined. To this end, each pixel simply divides into a number of new pixels with the same values to obtain a larger image. Then, a regular convolution ‘adjusts’ these extrapolated pixels so that the output  $U$  matches better supplied data.

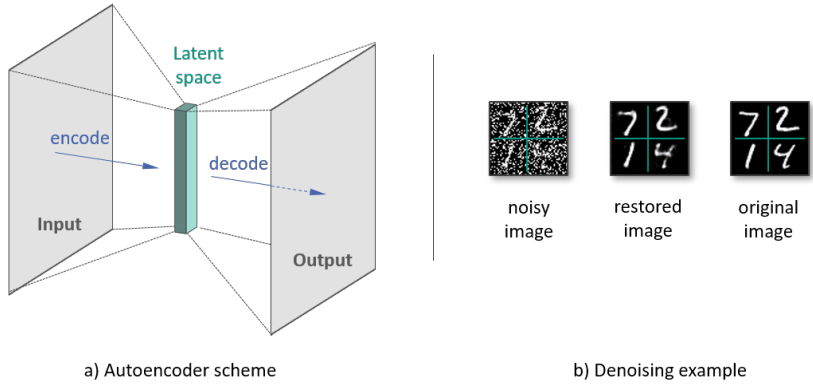


**Figure 4.10:** Visualisation of upsampling by convolution of a nearest neighbourhood extrapolation.

**Network architecture.** Formally, the envisaged image-to-image correlation is a function

$$\mu_f : G \mapsto A \quad (4.4)$$

mapping from a geometry-grid  $g \in G$  to a part quality field  $a \in A$  (shear strain). To this end, a U-net-shaped autoencoder architecture is chosen for  $\mu_f$ . Autoencoders are frequently used to find an information-equivalent, lower-dimensional representation (latent vector) of high-dimensional data, i.e. data compression. Unlike rule-based compression algorithms, they are meant to *learn* a suitable compression automatically from data – hence their name. Figure 4.11 a) shows their schematic structure.

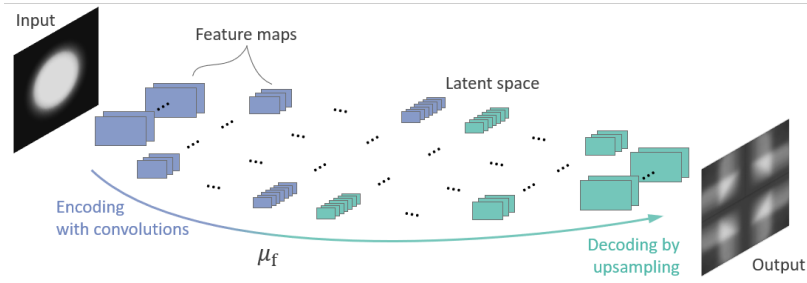


**Figure 4.11:** Autoencoder structure and denoising example. Image in b) from [203]

Overall, they have a bottleneck-structure and consist of three main parts: an *encoding* part, a *decoding* part and a ‘latent space’ in between. All input data is transformed and compressed to the latent space before the second, decoding part interprets it. The latent space essentially acts as an information-sieve. During training the algorithm is forced to adjust the latent space in such a way that it extracts and stores informative-rich data features only and discards irrelevant noise.

A typical application example for autoencoders is image denoising. Figure 4.11 b) shows noisy images of hand-written digits [203]. The autoencoder is trained to remove the noise, while retaining relevant information to read the digits. Comparing its result the original images shows that auto-encoder successfully removes the noise without impairing the readability of the images. That is, it tells relevant and irrelevant data features apart.

In this work, they are adapted to identify forming-relevant geometry features and estimate their forming result as Figure 4.12 shows. Geometry encoding is done using convolution operations, for decoding nearest-neighbour extrapolation with a subsequent convolution is used in accordance with the findings by TRIPPE [261].



**Figure 4.12:** Schematic visualisation of the autoencoder structure used for  $\mu_f$  within this work.

During geometry encoding, multiple convolution layers process the greyscale input image of the geometry to ensure that sufficiently complex geometry features can be encoded. In the second part of the network the encoded geometry information is interpreted and layer-wise upsampled.

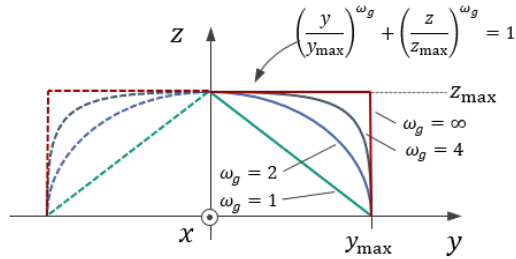
**Data curation.** The image-based approach analyses the component as a whole and as such, defining distinct features like corners for algorithm training is inexpedient. Complete specimen geometries are analysed instead. As before, a parametric CAD-model is used to auto-generate training geometries. To this end, the equation

$$\left(\frac{y}{y_{\max}}\right)^{\omega_g} + \left(\frac{z}{z_{\max}}\right)^{\omega_g} = 1 \quad (4.5)$$

defines the surface contour line. See Figure 4.13 for example contours.

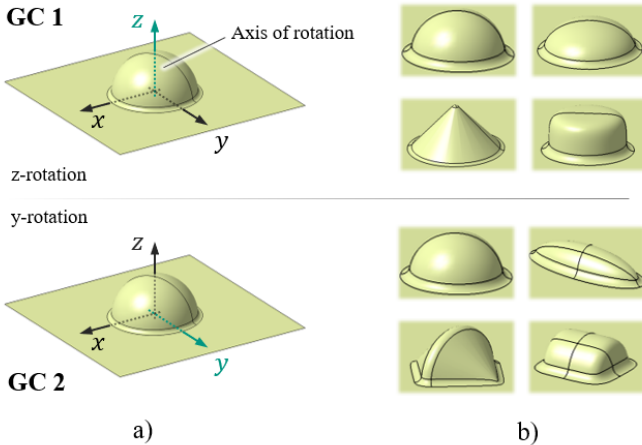
In Equation (4.5), the variables  $y_{\max}$  and  $z_{\max}$  control the contour aspect ratio, i.e. stretches or compacts it in  $y$ - and  $z$ -direction; the exponent  $\omega_g$  varies the contour's curvature. Rotating them about the  $y$ - and  $z$ -axis, respectively, then allows to generate two geometry classes (GC). GC 1 primarily varies the *out-of-plane* curvature, GC 2 the *in-plane* curvature, which allows to study the effect of different training geometries on the





**Figure 4.13:** Definition of the contour line to generate bodies of revolution during geometry sampling for supervised training of  $\mu_f$  [254].

CNN-performance. Overall, a wide range of different geometries ranging from cones ( $\omega_g = 1$ ) to ellipses ( $\omega_g = 2$ ) to cylinders ( $\omega_g \rightarrow \infty$ ) can be generated. Figure 4.14 a) illustrates the axes of rotation, b) provides geometry samples for visualisation.



**Figure 4.14:** Illustration of the a) axis of rotation and b) some example geometries obtained by contour-line-rotation about the  $y$  and  $z$ -axis, respectively [254]. Rotating about  $y$  primarily varies the in-plane curvature, rotating about  $z$  the out-of-plane curvature.

In this work, the parameters

$$\begin{aligned}50 \text{ mm} &\leq y_{\max} \leq 150 \text{ mm} \\0.05 &\leq \frac{z_{\max}}{y_{\max}} \leq 1.0 \\1.0 &\leq \omega_g \leq 6.0\end{aligned}$$

are used. The extent of the component in  $x$ -direction  $x$  results from the axis of rotation: Rotating about the  $y$ -axis results in  $x = z$  and about the  $z$ -axis in  $x = y$ . Any sharp edges are rounded with a radius of 10 mm.

From these geometries, different numbers of samples are drawn by LHS and their shear angle distribution is evaluated in textile forming simulations. To save computation time during sampling, a kinematic draping simulation approach is chosen over an FE-based approach. Both the geometries and the shear angle distribution are eventually converted to greyscale images and constitute the database for ML-training. Note that the geometry parameters  $y_{\max}$ ,  $z_{\max}$  and  $\omega_g$  are used for geometry generation only. Once all training geometries are available and converted to a greyscale-images, the parameters are discarded. Only the greyscale images enter the CNN.

## 4.2 Results of the formability assessment

In the following sections the two main approaches – *feature*-based and *grid*-based – are discussed for formability assessment. Specifically, Section 4.2.1 validates the assumptions and results of the feature-based approach and Section 4.2.2 for the grid-based approach.

### 4.2.1 Results of the feature-based approach<sup>1</sup>

Validating the feature-based approach amounts to providing a proof of concept for 1) the inherent assumption of strain field independence (SAINT-VENANT's principle) and 2) an application to a demonstrator part.

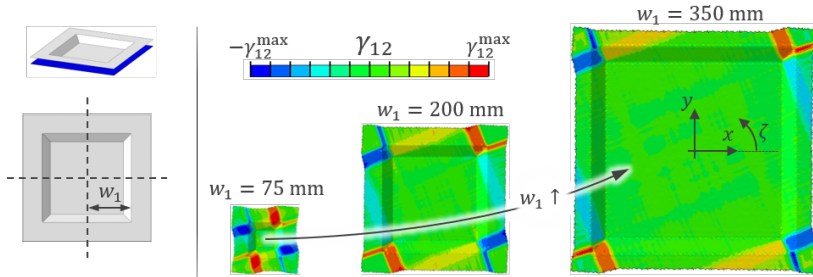
**SAINT-VENANT's principle in fabric forming.** The feature-based approach (Section 4.1.1) assumes that the strain fields of geometric features form independent of each other provided they are sufficiently far apart or the resultant loads vanish (SAINT-VENANT's principle, SVP).

In order to validate this assumption, a parametric study is conducted. More specifically, a fabric cut is draped onto the parametric box geometry from Figure 4.2. The box is quadratic ( $w_1 = w_2$ ) with constant corners and exemplarily configured to form a pyramid frustum of variable width  $w_1$ . Specifically, the parameters  $\mathbf{g}_p = (r_1, r_2, \alpha, w_1, w_2, h)^T = (5 \text{ mm}, 5 \text{ mm}, 40^\circ, w_1, w_1, 50 \text{ mm})^T$  generate the pyramid frustum. The width  $w_1$  varies between  $35 \text{ mm} \leq w_1 \leq 850 \text{ mm}$  during the study; the fibre orientation  $\zeta$  relative to the box-edges between  $0^\circ \leq \zeta \leq 45^\circ$ . Figure 4.15 visualises the setup along with three exemplary forming results for  $\zeta = 22.5^\circ$ .

Different metrics of the shear angle distribution have been recorded during variation of  $w_1$ . Yet, as it provides most insight, the maximum shear angle  $\gamma_{12}^{\max}$  is discussed. Figure 4.16 shows the evolution of  $\gamma_{12}^{\max}$  with  $w_1$  for each material orientation.

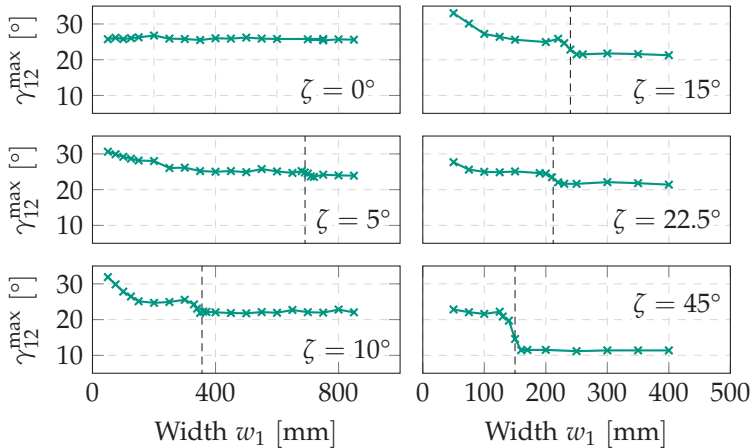
The graphs share three common characteristics. First, starting from a maximum value, they decline and approach a minimum value in an asymptotic manner. Second, the difference between maximum and the minimum value shrinks for  $\zeta \rightarrow 0^\circ$ . It is practically zero for  $\zeta = 0^\circ$ , i.e. when the material orientations align with the box-edges. Third, each curve except  $\zeta = 0^\circ$  shows a more or less pronounced drop highlighted

<sup>1</sup>The presented results are published in the prior works [237, 238]



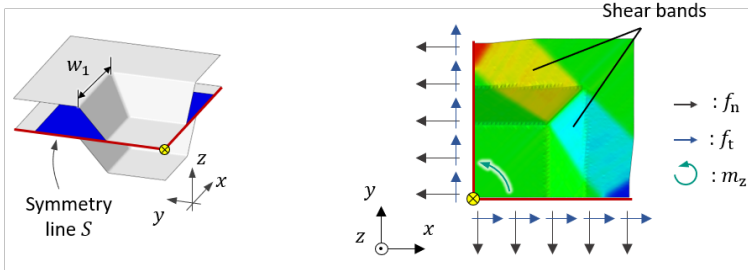
**Figure 4.15:** Study setup to validate the applicability of SAINT-VENANT'S principle (SVP) [238]. The distance between corner geometries  $g_p$  grows in a stepwise manner while measuring the forming quality (shear angle maximum  $\gamma_{12}^{\max}$ ).

by the dashed vertical line. Additional observations have been recorded around these drops for sufficient resolution. Beyond these drops, the graphs are approximately constant, which hints strain field independence.



**Figure 4.16:** Evolution of  $\gamma_{12}^{\max}$  with the box-width  $w_1$  between two pyramid corners.

The observation can be justified by SVP. To this end, the resulting forces across a reference surface are evaluated. Figure 4.17 exemplarily visualises the procedure for the case  $\zeta = 45^\circ$ .

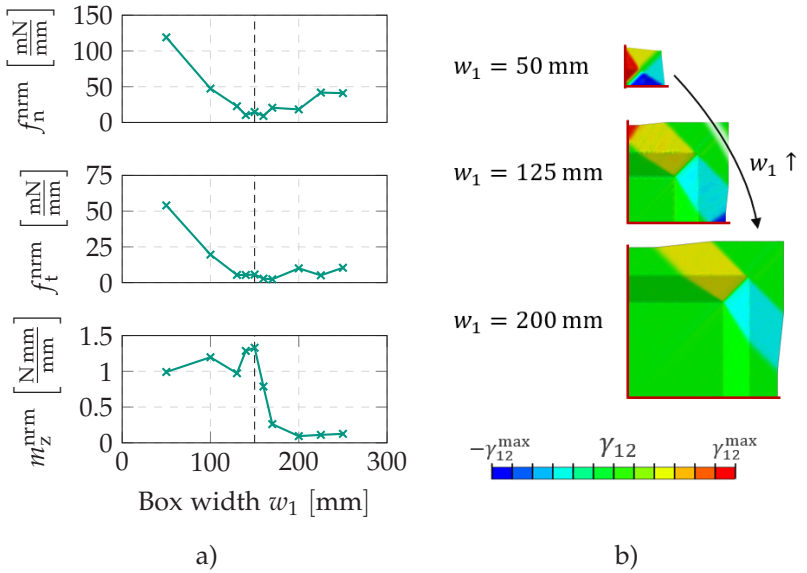


**Figure 4.17:** Extraction procedure of the internal line-loads across the fabric symmetry line during forming (normal and tangential forces  $f_{n,t}$  and moment  $m_z$  about the  $z$ -axis) [238].

As can be seen from the qualitative plots in Figure 4.17, distinct shear bands form along warp and weft direction. Since shear deformations inevitably cause shear stresses, reaction forces and moments across the symmetry line  $\mathcal{S}$  can be extracted. More specifically, the resulting tangential and normal forces  $f_{n,t}$  and the resulting moment  $m_z$  about the  $z$ -axis are extracted at the symmetry line  $\mathcal{S}$  for different values of  $w_1$ . The loads are normalised (superscript 'nrm') to account for the gradual growth of the symmetry line length with  $w_1$ .

Figure 4.18 a) quantifies the evolution of the loads transmitted across  $\mathcal{S}$ . The forces  $f_{n,t}^{\text{nrm}}$  reduce until they reach a minimum at  $w_1 \approx 150$  mm. Some fluctuations aside, the moment  $m_z^{\text{nrm}}$  initially stays constant but drops also at  $w_1 \approx 150$  mm. Beyond this width both the forces and the moment stay approximately constant; the minor growth of  $f_n^{\text{nrm}}$  stems from growing frictional forces due to increased tool-ply-contact area.

The absence of transmitted loads for  $w_1 \gtrsim 150$  mm satisfies the strict conditions for a so-called *astatic equilibrium*. That is, the loads are not just in equilibrium but remain in equilibrium during rotation about an

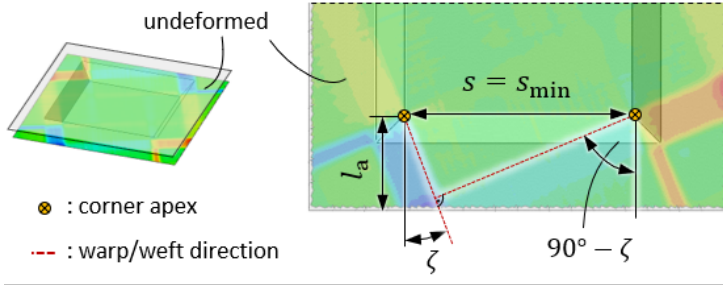


**Figure 4.18:** a) Plot of the evolution of line-loads in the fabric during forming; b) visualization of the shear band movement from the boundary (red) to the free edge when  $w_1$  grows.

arbitrary angle [252, p 359-362]. As the direction of vanishing loads is arbitrary, they are always in an astatic equilibrium. In this case the literature reports a decay exponent of  $p_{sv} \geq 4$  in Equation (4.1). Hence, SVP predicts a strong localisation effect. Note that the sudden drop of  $\gamma_{12}^{\text{max}}$  to a constant value in Figure 4.16 coincides exactly with the drop of transmitted loads. Both are marked by a dashed vertical line.

The qualitative plots in Figure 4.18 b) also offer a mechanical interpretation for the drop of the transmitted loads. As  $w_1$  grows, the shear bands gradually move outwards and at some point they leave the symmetry line  $\mathcal{S}$  at which the loads are extracted. At this point the associated shear stresses also leave the symmetry line and cannot evoke reaction forces and moments, respectively, i.e. they vanish.

This geometric consideration also allows to deduce an independence criterion for the minimum distance of two corners. Consider again the rectangular fabric cut as Figure 4.19 illustrates. Note that the shear strains are plotted into the initial, i.e. undeformed fabric.



**Figure 4.19:** Exemplary determination of the minimum required distance  $s_{\min}$  for independence of two doubly-curved regions aligned to the edges of a rectangular fabric cut. The plot shows the limit case of beginning independence (i.e.  $s = s_{\min}$ ). When  $w_1$  increases, the shear bands separate further, when it decreases, the shear bands move together and shear strains will interact [238].

Let  $l_a$  be the distance of the corner apices to the fabric edge and  $s$  the distance between two corner apices. Then the minimum distance  $s_{\min}$  can be determined to be

$$\begin{aligned} s_{\min} &= l_a (\tan(\zeta) + \cot(\zeta)) \\ \Leftrightarrow &= \frac{l_a}{2 \sin(2\zeta)} \quad . \end{aligned} \quad (4.6)$$

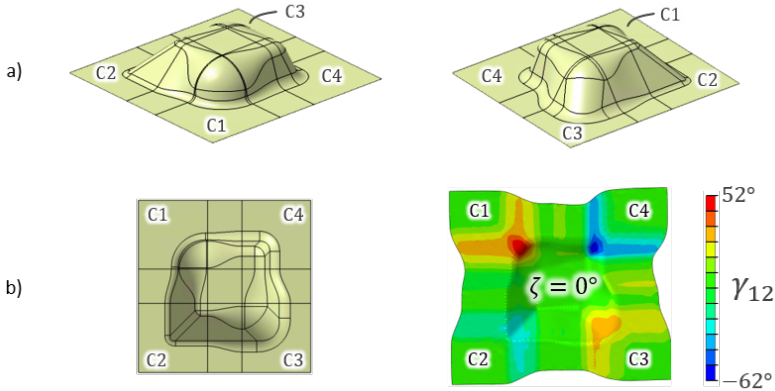
Figure 4.19 visualises the situation  $s = s_{\min}$ , i.e. the beginning of strain field independence. The strain fields will be independent, should  $s$  increase; and will interact, should it decrease.

The dashed, vertical lines in both Figures 4.16 and 4.18 support the devised relation. They are in fact determined by Equation (4.6). As they coincide with the drops in the graphs, they validate it as a sufficient criterion for strain field independence. Note however, that Equation (4.6)

only holds for rectangular fabric cuts and that other relations may be required for other cuts.

With this paragraph, a justification for the individual assessment of the forming results has been established. In the following, the approach is exemplarily applied to a demonstrator geometry.

**Application to a demonstrator geometry.** With the above explanation, the feature-based approach shall be used to assess and eventually improve the manufacturability of a box-demonstrator. The demonstrator is inspired by the NUMISHEET93-benchmark [251] but features higher complexity as Figure 4.20 a) shows: While the original is a double-symmetric square-cup, this demonstrator is non-symmetric and features four individual corners C1 . . . C4.



**Figure 4.20:** a) Isometric and b) top-view for the box-demonstrator used to validate the feature-based formability assessment along with a forming simulation result (fibre orientation  $\zeta = 0^\circ$ ) for visualisation. The box demonstrator features four corners of different forming difficulty [238].

The corners stem from the introduced parameterisation scheme (Figure 4.2). Table 4.2 summarises their parameter values. The parameters were set in such a way that they are not part of the training geometries.



That is, all corner geometries are new to the surrogate and must make inference on the basis of the training corners.

**Table 4.2:** Definition of geometry parameters for the box-demonstrator [238].

Corner	$\frac{r_1}{\text{mm}}$	$\frac{r_2}{\text{mm}}$	$\frac{\alpha}{^\circ}$	$\frac{w_1}{\text{mm}}$	$\frac{w_2}{\text{mm}}$
C1	40	10	10	35	35
C2	10	10	40	15	15
C3	40	20	35	25	25
C4	15	15	15	30	30

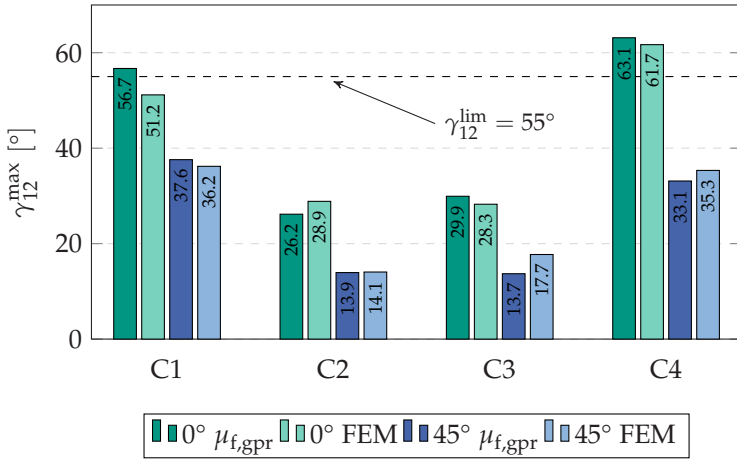
For approach validation, two fibre orientations are considered, namely  $\zeta = 0^\circ$  and  $\zeta = 45^\circ$ . Figure 4.20 b) shows an example FE-forming result of the demonstrator with  $\zeta = 0^\circ$ . For both fibre orientations, a separate surrogate is trained on a database with  $n_k = 71$  sample corners each. This number was found sufficient in an upstream study [237].

Figure 4.21 compares  $\gamma_{12}^{\max}$  predicted by the GPR<sup>1</sup>-surrogate  $\mu_{f,\text{gpr}}$  to the FE-simulation for each corner and material orientation. In general, the surrogate-estimations match the simulation results well. The maximal deviation amounts to  $\approx 5.5^\circ$  ( $\approx 9.7\%$ ), while all other differences remain below  $4^\circ$ .

The plot also features a shear strain limit  $\gamma_{12}^{\text{lim}} = 55^\circ$  (dashed horizontal line). It is taken as a shear limit to tell manufacturable and non-manufacturable designs apart [238]. Clearly for  $\zeta = 0^\circ$ , corner C1 and C4 exceed this limit according to the surrogate estimation. If  $\mu_{f,\text{gpr}}$  was used for component design, it would suggest to redesign C1 and C4. For  $\zeta = 45^\circ$  it deems all corners manufacturable, though.

Owing to its low evaluation time,  $\mu_{f,\text{gpr}}$  can be used for extensive re-design studies as Figure 4.22 exemplifies. It shows  $\mu_{f,\text{gpr}}$ 's estimation for

<sup>1</sup>Gaussian Process Regression

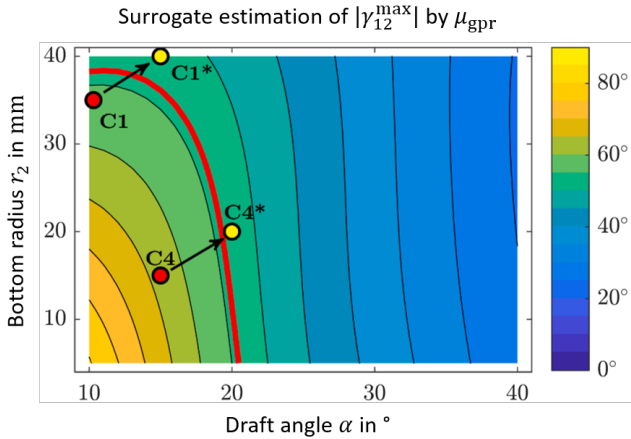


**Figure 4.21:** Comparison of  $\gamma_{12}^{\max}$  for each corner of the box-demonstrator as predicted by the ML-model (dark shade) and an according FE-Simulation (light shade) [238].

$\gamma_{12}^{\max}$  as a function of the wall draft angle  $\alpha$  and the radius  $r_2$  as they have been identified as most influential. A red line marks the locus of ‘limit-designs’ with  $\gamma_{12}^{\max} = \gamma_{12}^{\lim} = 55^\circ$  and separates manufacturable from non-manufacturable designs. Additionally, red markers indicate the critical corner designs C1 and C4. Clearly, they lie in the critical region.

This corner design map can be conveniently used to improve the forming behaviour: A potential designer can pick any design from the manufacturable zone and can concentrate on other design requirements, e.g. functional or packaging aspects. The yellow markers represent an exemplary corner redesign, denoted by an asterisk. Specifically, C1 changes according to  $\alpha : 10^\circ \rightarrow 15^\circ$  and  $r_2 : 35 \text{ mm} \rightarrow 40 \text{ mm}$ ; C2 according to  $\alpha : 15^\circ \rightarrow 20^\circ$  and  $r_2 : 15 \text{ mm} \rightarrow 20 \text{ mm}$ .

Figure 4.23 compares the estimated and the simulated strains before and after the redesign. Indeed, a successful reduction of  $\gamma_{12}^{\max}$  below  $\gamma_{12}^{\lim}$  can be observed. Also for the redesigned corners C1\* and C4\* the surrogate

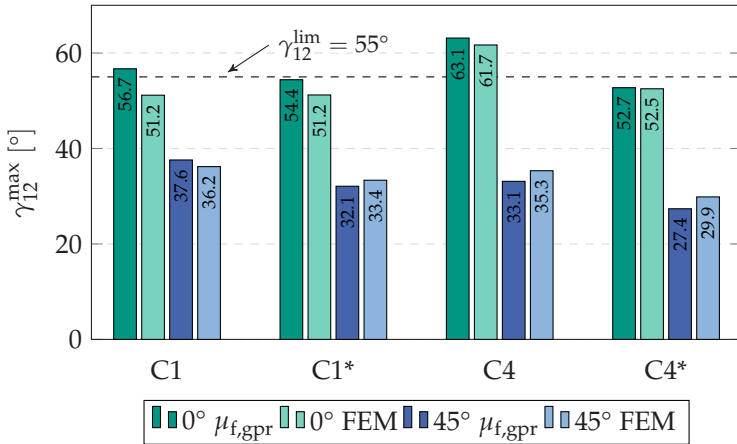


**Figure 4.22:** Corner design map for manufacturability assessment [238]. The red line marks the locus of  $\gamma_{12}^{\max} = \gamma_{12}^{\lim} = 55^\circ$  and separates manufacturable from non-manufacturable corner designs. Markers indicate the location of corner C1 and C4 before (red) and after redesign (yellow).

estimation and FE-simulation match well (maximal deviation of  $\approx 3^\circ$ ). For the sake of completeness, the results for  $\zeta = 45^\circ$  are also given and show that manufacturability is conserved ( $\gamma_{12}^{\max} < \gamma_{12}^{\lim}$ ).

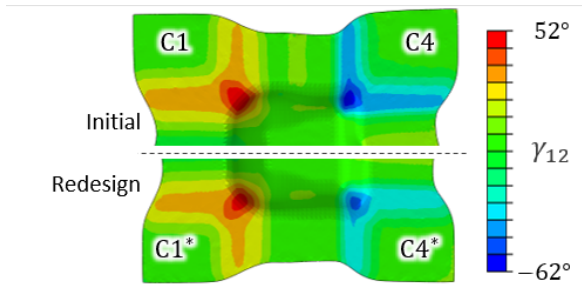
Eventually, Figure 4.24 gives a visual impression of the improved forming behaviour. It shows the shear angle distribution of the initial and the redesigned corner geometries. While C1 and C1\* differ only marginally, a substantial reduction of  $\gamma_{12}^{\max}$  can be observed when changing C4  $\rightarrow$  C4\*. Consequently, the trained surrogate was successfully applied to efficiently assess the manufacturability and facilitated redesigns of individual geometric features.

Overall, the results show that the feature-based approach supports an efficient formability assessment – provided a set of preconditions is satisfied (strain field independence). If so, further analyses can be performed to quantify and visualise the impact of local design variations. The usage of geometrical parameters, e.g. radii or angles, facilitates an understanding of geometry-process-interdependencies and allows



**Figure 4.23:** Comparison of  $\gamma_{12}^{\max}$  for the initial and the redesigned corners of the box-demonstrator as predicted by the ML-model (dark shade) and an according FE-Simulation (light shade) [238].

to create engineering-interpretable design maps. They can then support the designer in an intuitive manner without need for laborious and computation-intensive FE-simulations.



**Figure 4.24:** Visualisation of the shear angles  $\gamma_{12}$  after forming for the initial corners C1 and C4 (top) and their redesigned pendants C1\* and C4\* (bottom, mirrored view) [238].

Despite these results, the presented approach is bound to predefined and sufficiently distant geometry features which may or may not exist in the analysed part design. Features outside the predefined scheme cannot be analysed. As a remedy, grid-based methods promise higher generalisation capabilities (Section 4.1.2). Their results will be analysed in the following section.

## 4.2.2 Results of the grid-based approach<sup>1</sup>

While the above feature-based approach assesses the formability of a component by identification and evaluation of discrete geometry parameters, i.e. *unstructured* input. *Structured* data representations offer a more versatile description but have yet not been applied for formability analysis. Formally, the formability assessment sketched in Figure 4.6 amounts to image-to-image-regression, for which Section 4.1.2 proposes a U-net-shaped, autoencoder architecture (Figure 4.12).

Assessing the applicability of the grid-based approach with CNNs concerns three key aspects, namely 1) general proof that the CNN  $\mu_f$  indeed captures the inherent process dynamics, 2) that it can generalise to unknown geometries and 3) how different training databases affect the result.

**CNN-architecture and hyperparameters.** Under supervision of the author, TRIPPE [261] performed extensive hyperparameter studies on three different image-based autoencoder networks from literature [210, 200, 262] in his master thesis. Specifically he studied the effect of 1) different training hyperparameters (learning rate, weight initialisation, batch size), 2) different network architecture (number and size of layers and kernels, upsampling strategy, central dense layer) and 3) different

---

<sup>1</sup>The presented results are published in the prior works [211, 254]

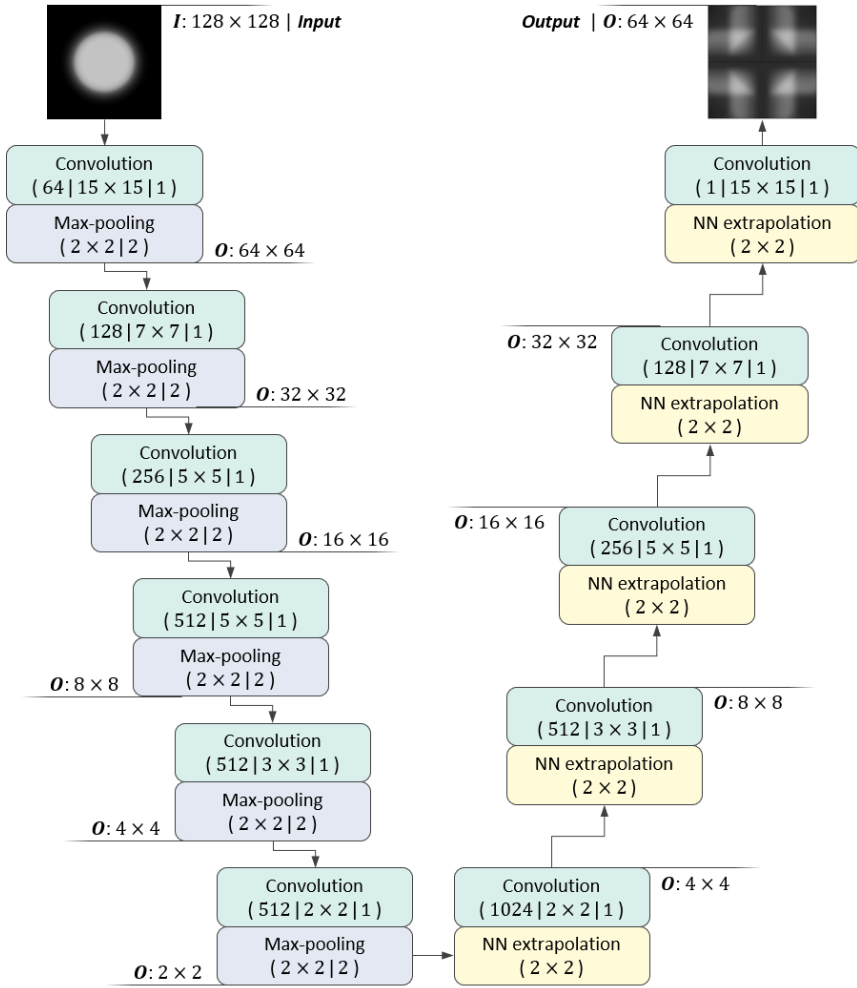
data representations (size of input- and output-image, greyscale encoding scheme). He summarised the findings in a best-practice-guideline and presents an optimised architecture which this work adopts. Figure 4.25 shows the final architecture.

In accord with TRIPPE's recommendations,  $\mu_f$  employs few, large kernels in the first layers to capture larger geometry characteristics and then reduces the kernel size in small steps while increasing their number. The inverse holds for the upsampling steps: Starting with many, small kernels, their size gradually increases while their number reduces. See [261, p 87-93] for further details. The optimised network of  $\mu_f$  reduces the prediction error by  $\approx 30\%$  compared to the original architecture by FARIMANI et al. [200].

The CNN  $\mu_f$  consists of 11 layers (6 encoding layers, 5 decoding layers) with  $\approx 19.65 \cdot 10^6$  tunable model parameters. The U-shaped flowchart style is a typical representation for 'U-nets' – hence their name. The network takes a  $128 \times 128$  greyscale-image of the geometry as input. Then convolution and pooling operations follow. Their configuration is set up so as to yield an output  $\mathcal{O}$  half the size of the input size, i.e.  $64 \times 64$  as annotations flags show. For instance, the  $128 \times 128$  input image results in  $64 \times 64$  feature maps.

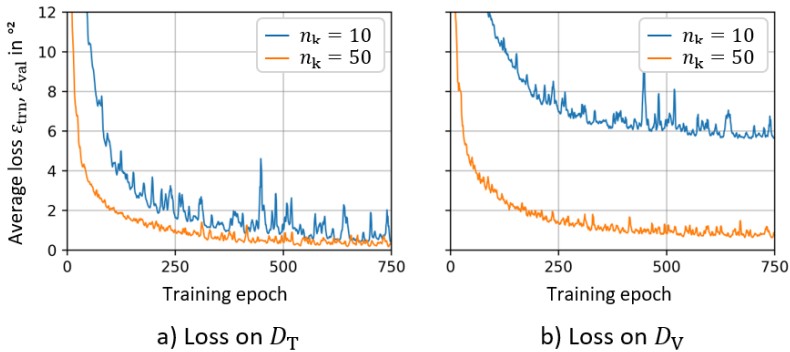
The convolution nomenclature is as follows: The first number denotes the number of kernels of this layer, the second denotes the kernel size and the last number denotes the stride. For instance, the first convolution  $64|15 \times 15|1$  implies convolving 64 different kernels, each of the size  $15 \times 15$  with a stride of 1. Similar holds for the pooling and upsampling operations: The first numbers denote the kernel size, the last the stride. Analogous to the convolutions, the upsampling parameters are configured in such a way that they double the output in each layer.

All neurons in the network use ReLU-activation. Some publications, e.g. [210, 262], suggest an additional, fully-connected layer after the convolutional layers, but TRIPPE's studies show no significant benefit for this work [261].



**Figure 4.25:** Employed architecture of the CNN  $\mu_f$  for the grid-based formability assessment. The architecture follows the hyperparameter study by TRIPPE [261].

**CNN-training behaviour.** Figure 4.26 shows the training progress of  $\mu_f$  for two different-sized *training* databases  $D_T$  with  $n_k = \{10, 50\}$  geometry samples from GC 1 (cf. Figure 4.14). Additionally, a separate *validation* set  $D_V$  with  $n_V = 20$  validation geometries from GC 1 allows to assess the performance on unseen geometry samples (Figure 4.26 b). As  $\mu_f$ 's network parameters  $\theta$  are initialised randomly, individual training runs show scatter. Thus, the graphs show the average of ten independent training runs with 750 epochs using the Adam-algorithm (learning rate  $\eta = 5 \cdot 10^{-5}$ , batch size of 1).



**Figure 4.26:** Evolution of the prediction error a) on the training data ( $\epsilon_{\text{trn}}$ ) and b) on the validation data ( $\epsilon_{\text{val}}$ ). Each plot features two graphs, one for a small database ( $n_k = 10$  samples) and one for a large database ( $n_k = 50$  samples) [254].

The graphs show the classical behaviour of supervised learning (cf. Figure 2.26): Some erratic volatility aside, the prediction error on training data  $\epsilon_{\text{trn}}$  (sub-image a) ) approaches zero in a monotonous manner for both database sizes. That is, the network learns to reproduce the training data.

However, the graphs of  $\epsilon_{\text{val}}$  (validation data) in sub-image b) differ: For the large database with  $n_k = 50$  samples the graph still almost reaches zero, but not for  $n_k = 10$  (small database). Here, an error of  $\approx 6^{e^2}$  remains. That is,  $n_k = 10$  samples do not provide sufficient information



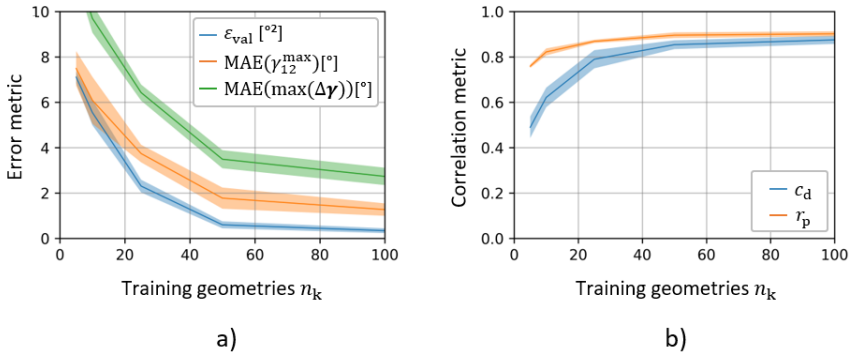
for an accurate generalisation to new geometry samples from GC 1, but  $n_k = 50$  do. Consequently, a threshold ought to exist which introduces just enough information for generalisation. The following paragraphs examine this threshold more closely.

**CNN-validation *inside* the class of training geometries.** Figure 4.27 shows the evolution of different accuracy metrics for different-sized databases from GC 1. Analogous to Figure 4.26 a), the prediction error on the training geometries approaches zero in all cases and is of little informative value. Thus, only the accuracy on the *validation* geometries is discussed here. The graphs show the average over ten independent training runs; the shaded area denotes the according 95 % percentiles.

Subplot a) visualises three *difference metrics*, namely 1) the validation loss  $\varepsilon_{\text{val}}$ , the prediction error of the shear maximum  $\Delta\gamma_{12}^{\text{max}}$  and the largest overall prediction error  $\Delta_{\text{max}}(\gamma_{12})$ . Subplot b) shows *correlation metrics* of the shear angles in the estimated and simulated strain field. Specifically,  $c_d$  and  $r_p$  denote the coefficient of determination and PEARSON correlation coefficient, respectively [263, p 585]. They measure how much the simulated and the estimated shear angle correlate per element. See Appendix A.1 for the formal definition of all metrics.

The metrics are evaluated and averaged across 20 validation geometries. Please note that the sense of the accuracy metrics is opposite between the plots. In subplot a), a perfect estimation implies zero error, in subplot b)  $c_d = r_p = 1$  is ideal.

In both plots the metrics improve monotonously as more data becomes available. Yet from  $n_k \approx 50$  on, the metrics improve only marginally in an asymptotical manner, which implies the aforementioned threshold. Interestingly, this threshold appears even earlier in subplot b) at  $n_k \approx 25$ . This hints that the network at first learns the qualitative distribution of the shear strain field before it eliminates absolute deviations. Similar results were found for GC 2 but are omitted here to avoid redundancy.



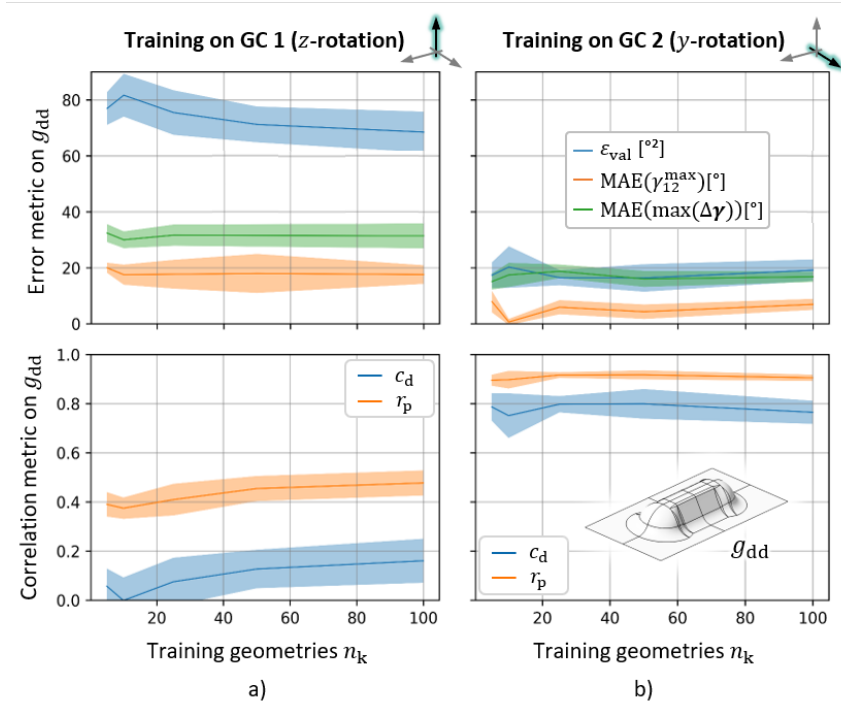
**Figure 4.27:** Effect of the number of training geometries on the achievable CNN-model accuracy on geometries inside the class of training geometries [254]. Similar results were found for GC 2 but are omitted here to avoid redundancy.

Overall, the CNN shows a classical surrogate behaviour: When evaluating geometries *inside* the class of training geometries, its accuracy on new geometries improves with available data.

**CNN-validation *outside* the class of training geometries.** Feature-based surrogates are only capable of assessing geometries from a pre-defined parameterisation scheme (Section 4.2.1) – a severe limitation for generalisation. The overarching motivation behind grid-based approaches is that they should also be capable of analysing geometries *outside* their class of training geometries. This capability is analysed in the following.

The previously trained models of  $\mu_f$  are again considered. Unlike before however, they are evaluated on the double-dome  $g_{\text{dd}}$  benchmark geometry. For reference,  $g_{\text{dd}}$  has been added to Figure 4.28 (bottom right). Please note that – despite some similarity – neither GC 1 nor GC 2 can exactly reproduce  $g_{\text{dd}}$  and consequently,  $\mu_f$  must make inference on a new geometry outside the class of training geometries.

Figure 4.28 visualises the results by *difference* and *correlation* metrics. Interestingly, in both cases the number of training geometries has only little effect on the accuracy. The metrics stay approximately constant. Only in sub-image a) a moderate improvement can be observed when training on GC 1.



**Figure 4.28:** Effect of the number of training geometries on the achievable CNN-model accuracy on geometries *outside* the class of training geometries [254].

However, a substantial difference prevails between sub-image a) and b). For instance, after training on GC 1 the loss  $\varepsilon_{\text{val}}$  amounts at best to  $\approx 68^{\circ 2}$ , while only  $\approx 20^{\circ 2}$  is observed for GC 2, i.e. a gain of  $\approx 70\%$  in accuracy. Similar differences occur for the other metrics. This hints that

GC 2 is more informative to  $\mu_f$  than GC 1 when evaluating the double-dome  $g_{dd}$ . For other geometries it might be different. This immediately raises the question, how different training datasets affect  $\mu_f$ 's accuracy and – in the long term – how to design a suitable sampling for a sufficiently general ML-model.

**Influence of the training database.** In order to analyse the effect of different training databases, Figure 4.29 shows plots of the estimated and the simulated strain fields for three different scenarios: Training of  $\mu_f$  on  $n_k = 100$  samples of a) GC 1 (top), b) GC 2 (centre) and c) a combination of GC 1 and GC 2 (bottom). The strain field difference  $\Delta\gamma_{12}$  allows to assess local under- and overestimations. Each plot also features its maximum value for ease of interpretation.

The plots can be used to explain the difference of  $\mu_f$ 's accuracy between training on GC 1 and GC 2. Consider at first sub-image a), i.e. after training on GC 1. The strain field estimations only loosely reproduce the simulation result. In fact, only the general formation of horizontal and vertical shear bands with a local maximum at the intersection is reproduced. Their position, width and absolute values differ considerably as the difference plot highlights.

Upon closer inspection, it may be seen that  $\mu_f$  erroneously predicts shear angles in the image centre (black ellipsis in the difference plot), although this region remains unsheared in the simulation. These false shear bands stem from inadequate training geometries: The geometries in GC 1 are rotationally symmetric with respect to the  $z$ -axis as such they feature no straight areas. As a consequence,  $\mu_f$  remains uninformed that straight areas evoke no shear during forming.

Interestingly though, the vertical shear bands are more pronounced than the horizontal bands in the estimation. This is surprising because all geometries in GC 1 are rotational-symmetric about  $z$ . Thus, all samples give equal horizontal and vertical shear bands. Although it has

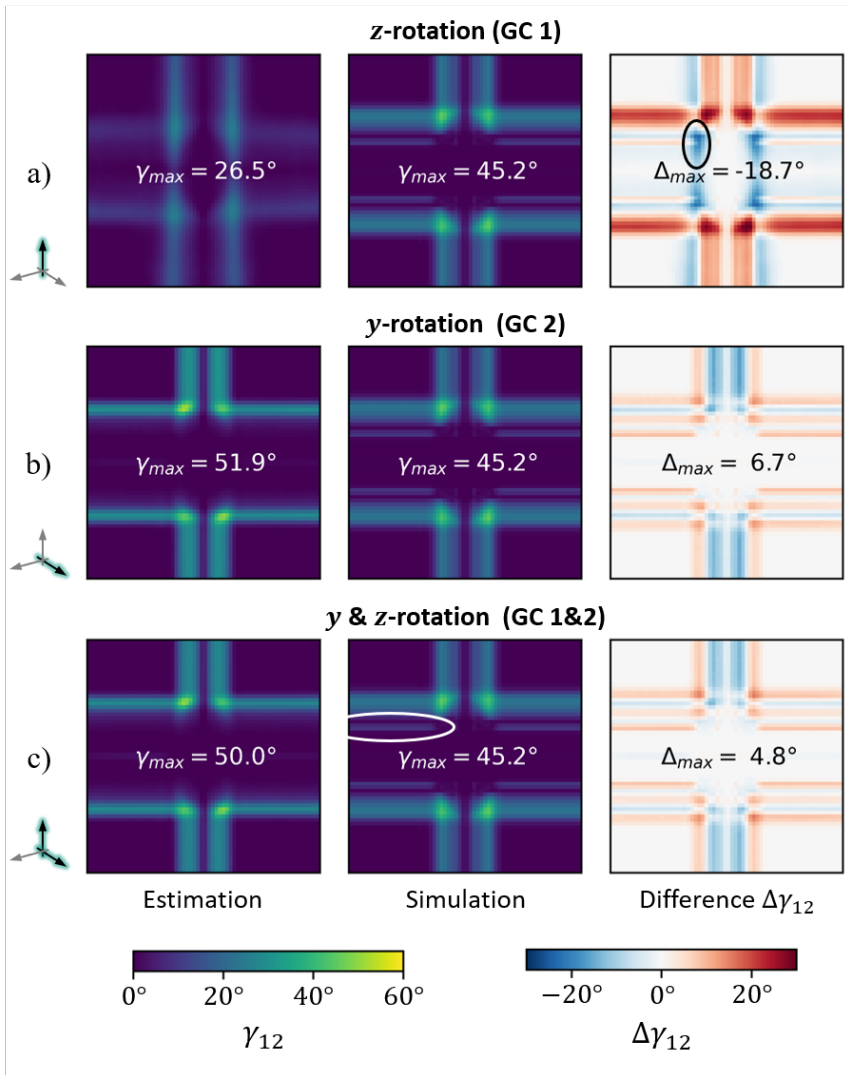


Figure 4.29: Comparison of the simulated and estimated shear strain field along with a plot of the difference [254].

never seen unequal shear bands  $\mu_f$  still predicts them for the double-dome. This indicates that it at least ‘tries’ to address the observed non-rotational-symmetry of the double-dome.

The situation considerably improves in sub-image b) after training on GC 2. Not just the position of the shear bands but also their absolute values match better the simulation results. Also the false shear bands in the centre do not occur since GC 2 does comprises samples with nearly straight areas, namely for  $\omega_g \gg 2$  (cylinders). Exactly these samples provide the information that straight component features evoke no shear. Consequently, the geometry characteristics in Figure 4.29 match better the features of the double-dome.

Ultimately, sub-image c) shows the results after a combined training on GC 1 and GC 2. However, it is to the widest extent identical to sub-image b) except for a marginally improved accuracy regarding the maximal shear. This result is more promising as it may seem at first glance. From a modelling perspective, this means that combining geometry classes does not automatically deteriorate the results. This holds even when the added geometry class contributes little information as GC 1 shows: Although the information-poor samples from GC 1 make up half of the total sample count, the remaining GC 2 samples provide still enough information for  $\mu_f$  to learn the process dynamics – provided that  $\mu_f$  shows a sufficiently large modelling capacity. This hints that adding geometries at best improves the ML-model and is at worst indifferent, i.e. is conservative.

Note however, that the prediction quality *outside* the training geometry class ( $\epsilon_{\text{val}}(n_k = 25) \approx 18^{\circ 2}$ , Figure 4.28) is considerably worse than *inside* ( $\epsilon_{\text{val}}(n_k = 25) \approx 2^{\circ 2}$ , Figure 4.27). In other words, a few samples may be enough to exhaust the informativeness of this geometry class with respect to a new geometry; however, estimating the forming result of a component *inside* the geometry class is still substantially more accurate.

It may further be noted that none of the estimations reproduces the minor shear bands on the inside of the main deformation zone (white ellipsis in sub-image c ). The shear bands stem from the concave indentations of the double-dome as Figure 4.30 highlights (green hue). All (training) geometries in GC 1 and GC 2 are fully convex and thus – similar to the straight geometry features of the double-dome –  $\mu_f$  remains unaware that concave also evoke shear deformations. This again emphasises the need to cover all forming-relevant geometry-characteristics during training such as convex and concave surfaces plus their transitions and different types of symmetries or curvature (non-, single- and double-curvature).

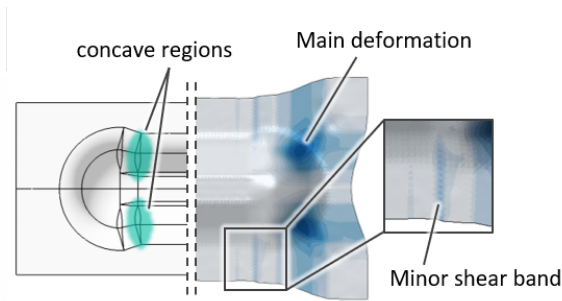


Figure 4.30: Minor shear bands and indentations in the double dome [254].

## 4.3 Intermediate summary

This chapter presents and compares two data-driven approaches for rapid formability estimation in textile forming, a *feature-based* and a *grid-based* approach. The feature-based approach (Section 4.1.1) seeks to identify forming-critical geometry elements in a component and evaluates them individually. The individual assessment of features is based on SAINT-VENANT'S principle and is validated in a numerical study. The method is applied to a modified benchmark geometry from literature

(Section 4.2.1). Overall, the method is found to give meaningful estimations of the local part quality of each feature. Additionally, the parametric nature of the approach is utilised to locally improve the manufacturability of a design.

In order to be analysed, the geometry features must conform to a predefined parameterisation scheme, though. Non-conforming features cannot be analysed. While this may be acceptable in certain scenarios, it generally limits the approach versatility. As an alternative, this work analyses a grid-based approach (Section 4.1.2) based on CNNs. Unlike the parametric surrogate, the CNN uses non-parametric, structured data. The CNN is trained on two different classes of training geometries and after the training it evaluates a geometry outside the training database, the double-dome.

The results show that the CNN indeed learns to estimate the strain field of new geometries, even when they are outside the class of training geometries. The estimation accuracy however depends on the evaluated geometry and the training database. For geometries *inside* the class of training geometries, the accuracy consistently improves with the number of training geometries. However, this does not hold for geometries *outside* the training geometry class. In this case, the prediction accuracy is nearly independent of the number of training samples but depends on the geometry class itself – more precisely, their covered geometry characteristics. The results also show, that geometry classes can be combined without impairing the prediction accuracy, even if the added geometry class provides little additional information. That is, adding geometries is conservative.

The findings suggest that carefully engineering and selecting the classes of generic training geometries is more important than the sheer number of samples. Loosely speaking, the findings support the notion of ‘smart data outperforms big data’. For a fully generalised ML-model, geometry characteristics are most important to be covered during training. This work already highlights the importance of single and double curvature



areas as well as convex and concave features. Other characteristics may be devised in future work.

From an application perspective, both methods hold promise, yet in different regards: The *feature*-based approach builds on well-established (parametric) surrogate-techniques for which convenient software-solutions are at hand. It also employs engineering-interpretable component parameters (draft angles, radii etc.) and thus support an intuitive inspection of part-process-relations, e.g. by design maps. Thus, for parameterisable design tasks with limited variability, the feature-based approach appears a practicable option.

However, design variations beyond the predefined feature-scheme are inapplicable. In this case, *grid*-based approaches have proven a viable remedy. They do not rely on specific geometry parameters but employ a more versatile grid-structure to represent the geometry. While this considerably improves versatility, it in turn raises follow-up research questions regarding geometry characterisation and sampling strategies for non-parametric spaces. Overall however, their versatility seems promising for the aspired generalisation. Thus, this works selects them for further development towards process optimisation of variable geometries.



# 5 Process optimisation of variable geometries

Chapter 3 shows that SBO helps reduce optimisation efforts. However, according surrogates are typically application-specific and fall short on re-usability in new scenarios, e.g. after an unforeseen geometry revision (Section 3.3). Thus, demand for generalised models has been identified early on [187]. Chapter 4 shows that grid-based ML-techniques, namely convolutional neural networks (CNNs), can learn process dynamics from generic data and apply it – to a certain degree – to a new geometries.

In this chapter, these geometry-informed techniques will be coupled with variable process parameters to allow process optimisation of variable geometries. More precisely, the overall aim is an approach for estimation of optimal manufacturing parameters for variable component geometries.

## 5.1 Reinforcement learning for process optimisation

This work proposes the combination of physics-based process simulations with ML-techniques to extend classical SBO towards variable instead of fixed manufacturing scenarios. The overall aim is to train an ML-model to estimate optimal manufacturing parameters for a new component geometry. The previous Chapter 4 has presented a feature-based surrogate approach and a grid-based CNN-methodology for formability assessment.

Theoretically, both could be extended towards process optimisation. However, the feature-based approach proves unsuitable as the following remarks expound, and thus, only the grid-based approach is further pursued.

**Non-suitability of a feature-based approach.** The feature-based approach (Section 4.1.1) scans a given geometry for predefined, potentially critical forming features, quantifies them by a set of geometry parameters and evaluates an according surrogate. Conceptually, its parametric nature allows to add additional process parameters in order to form a combined surrogate  $\mu_{\text{srg}} : G, P \mapsto Q$ . For a given geometry, i.e. fix geometry parameters  $g \in G$ , an optimiser could then tune the process parameters  $p \in P$  for maximal quality  $q \in Q$ .

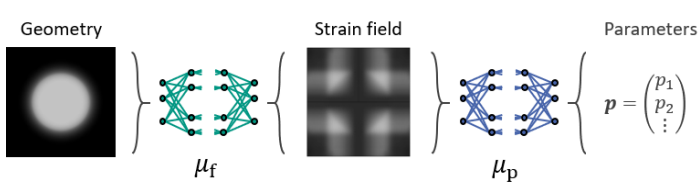
Yet, upon closer consideration this approach turns out naïve and hard to realise in practice for gripper-assisted fabric forming. As outlined in Section 4.1.1, the feature-based approach relies on the assumption that features can be analysed individually. This in turn assumes vanishing intersection loads in sufficient distance from the feature (SAINT-VENANT'S principle, cf. Section 2.2.2). Blank holder forces, however, inevitably introduce (non-vanishing) membrane loads across the textile, which instantly violates the prerequisites of SVP. This impedes the assessment of individual features. A hypothetical surrogate  $\mu_{\text{srg}}$  would have to parameterise *all* possible geometry features in the component plus process parameters. For the box-demonstrator in Section 4.2.1, it would amount to four corners with each five parameters, i.e. 20 parameters, plus blank holder forces and even more parameters for more complex features. This appears practically non-realizable.

**Algorithm architecture.** Owing to the overwhelming complexity, this work discards casting and optimising a combined surrogate  $\mu_{\text{srg}} : G, P \mapsto Q$  in favour of a more generalised, grid-based function

$$\mu : G \mapsto P \quad . \quad (5.1)$$

The function  $\mu$  accepts a geometry  $g \in G$  as input and directly estimates the optimal process parameters  $\mathbf{p}_{\text{srg}}^*$  as output. That is, it infers optimal actions (process parameters) in varying situations (geometries) – the idea of reinforcement learning (RL, cf. Section 2.3.4). Like a regular surrogate,  $\mu$  will be constructed using observations.

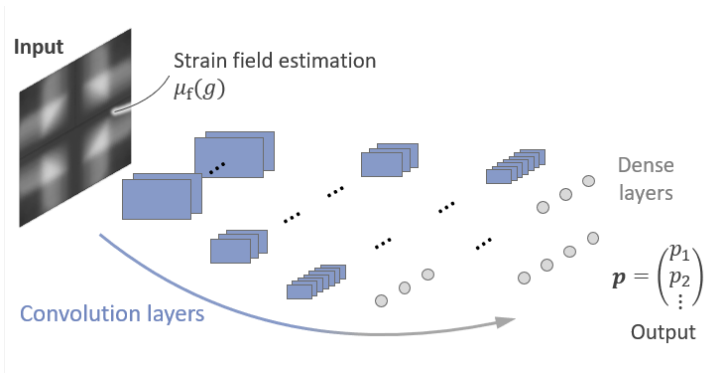
The envisaged function  $\mu$  consists of two composed sub-functions and extends the CNN for formability assessment  $\mu_f$  from Section 4.1.2. As Figure 5.1 illustrates, the output of  $\mu_f$  is directly input to another CNN for optimisation  $\mu_p$ . That is,  $\mu$  connects two CNNs in series which is formally cast as  $\mu = \mu_p(\mu_f(g))$ .



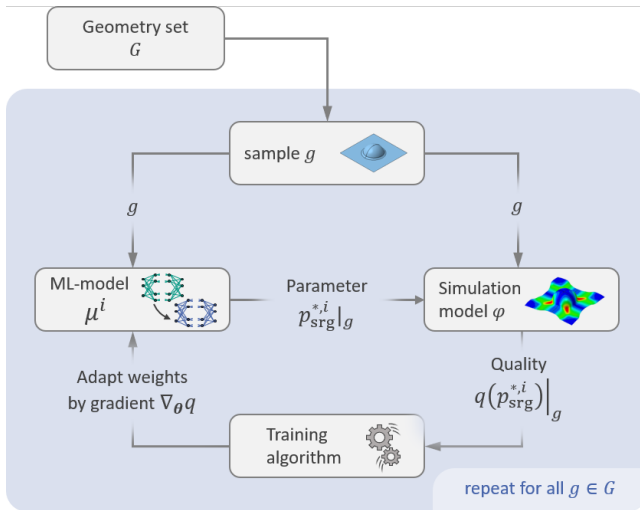
**Figure 5.1:** Architecture of the algorithm for estimation of optimal process parameters of variable components. The algorithm consists of two connected CNNs, the above introduced CNN  $\mu_f$  for formability assessment and a second CNN  $\mu_p$  for parameter estimation.

The sub-function  $\mu_f$  has already been discussed in Section 4.2.2 so that this chapter focusses on  $\mu_p$ . Again a CNN is used to enable a grid-based analysis. Figure 5.2 visualises the architecture. The first part resembles the aforementioned autoencoder structure, however the second half is different: It features fully connected layers instead of upsampling layers because it shall map to a process parameters  $\mathbf{p}$ , i.e. a vector, instead of an image.

**Reinforcement Learning concept.** Figure 5.3 illustrates the overall training concept. Therein, a vertical bar denotes ‘evaluated for’. For example  $\mathbf{p}_{\text{srg}}^{*,i}|_g$  implies the estimated parameter optimum in training iteration  $i$  when evaluating geometry  $g$ .



**Figure 5.2:** Architecture of the CNN  $\mu_p$  for process parameter estimation. The first part consists of convolutional layers for analysis for the image-input (shear strain field), the second part consists of fully-connected layers in order to yield a vector of process parameters.



**Figure 5.3:** Training and application scheme using reinforcement learning (RL): In each iteration  $i$ , a geometry is drawn (greyscale image) and evaluated by for a process parameter recommendation. An according simulation determines the resulting forming quality. If  $\mu$ 's process recommendations improve the forming quality, it is encouraged to give similar recommendations for similar geometries, and penalised otherwise [104].

In each training iteration  $i$ ,  $\mu^i$  is presented samples  $g$  from a predefined set  $G$  of  $n_g$  generic training geometries. For each sample it returns an according process recommendation  $\mathbf{p}_{\text{srg}}^{*,i}|_g$ . An according process simulation  $\varphi(\mathbf{p}_{\text{srg}}^{*,i}|_g)$  then evaluates this recommendation and the objective function  $q$  quantifies the resulting scalar part quality  $q(\varphi(\mathbf{p}_{\text{srg}}^{*,i}|_g))$ . As before, for brevity  $q(\mathbf{p}_{\text{srg}}^{*,i}|_g)$  tacitly implies  $q(\varphi(\mathbf{p}_{\text{srg}}^{*,i}|_g))$  ( $\varphi()$  is dropped for readability).

If the process recommendation improves the part quality  $q$ , then  $\mu^i$  is encouraged to give similar recommendations in similar situations, otherwise it is penalised. The ML-literature usually refers to  $q$  as *reward*. Over the iterations  $i = 1 \dots i_{\text{max}}$ ,  $\mu^i$  seeks to optimise  $q$ , so as to make useful process parameters more likely over time. After the training,  $\mu$  is meant to give recommendations even for new geometries which were not part of the training data. In ML, the overall approach is known as Reinforcement Learning.

This work assumes *constant* process parameters  $\mathbf{p}$  during forming, i.e.  $\mathbf{p}$  is set once and kept constant during the tool stroke. Consequently, only a single decision must be taken per geometry; state transitions do not occur. For completeness, please note that some process technologies may require parameter variations over time, though. For instance, consider in-situ adaptations of the tool closing speed to avoid defects from excessive or uneven cavity pressures, e.g. in Wet Compression Moulding. In such cases, the process passes through a whole sequence of states (pressure distributions) and according actions (increase/decrease speed). RL can also solve such time-dependent optimisation tasks but are deliberately excluded in this work in order to concentrate on variable geometries.

**Training implementation.** Within  $\mu$ , the function  $\mu_p$  shall learn to estimate optimal process parameters when given a strain field estimation  $\mu_f(g)$  for geometry  $g$ . From an RL-perspective this amounts to finding a function – a ‘policy’ in ML-terms – which yields the optimal action

(process parameter  $\mathbf{p}^*$ ) for a given state (strain field estimation  $\mu_f(g)$ ), i.e. solving

$$\mathbf{p}^* = \mu_p(\mu_f(g)) = \arg \min_{\mathbf{p} \in P} q(g, \mathbf{p}) \quad . \quad (5.2)$$

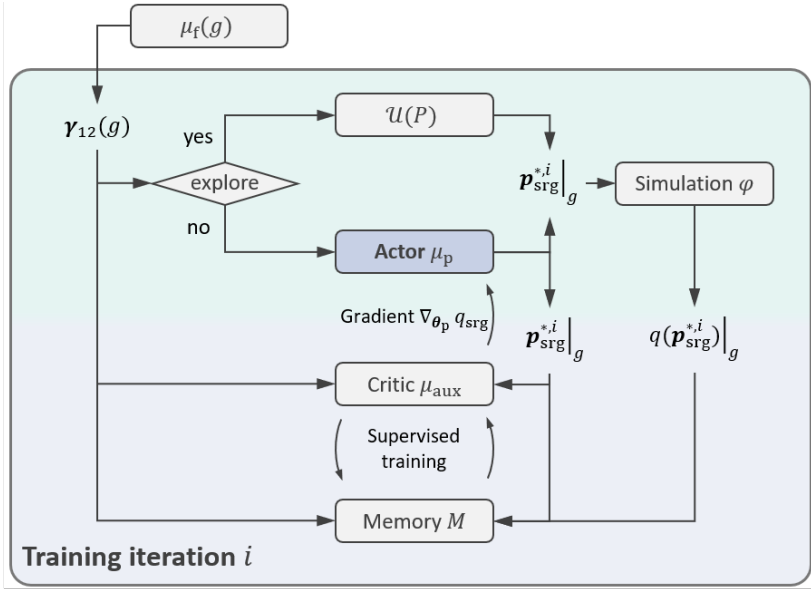
Classical, iterative optimisation techniques could solve Equation (5.2). As this is usually time-consuming, ‘Actor-Critic’-techniques eliminate the need for optimisation by parametrising  $\mu_p$  through a convolutional neural network with adjustable weights  $\theta_p$ . During training,  $\theta_p$  is then gradually tuned in direction of improving part quality  $q$  [264].

The training scheme is visualised in Figure 5.4 and involves two networks: The desired network  $\mu_p$ , the so-called ‘actor’-network, and an auxiliary ‘critic’-network  $\mu_{aux}$ . In loose terms, the overall idea is that the actor gives recommendations and the critic then informs the actor, how to improve. Upon closer inspection, training the actor  $\mu_p$  resembles the principle of classical backpropagation: At first,  $\mu_p$  makes a prediction (forward pass, green hue). Then these predictions are refined by an adjustment of model parameters  $\theta_p$  (backward pass, blue hue) with aid of the critic. This ‘prediction-correction’-scheme iterates until convergence of  $\mu_p$ .

More specifically, per geometry  $g$  and training iteration  $i$ , at first, new observations for each geometry  $g \in G$  are generated. To this end,  $\mu_p$  receives a shear-estimation  $\gamma_{12} = \mu_f(g)$  and infers a parameter recommendation  $\mathbf{p}_{srg}^{*,i}|_g$ . An according forming simulation  $\varphi(\mathbf{p}_{srg}^{*,i})|_g$  determines the resulting quality  $q(\mathbf{p}_{srg}^{*,i})|_g$ . Eventually, the observation tuple  $\{\gamma_{12}, \mathbf{p}_{srg}^{*,i}|_g, q(\mathbf{p}_{srg}^{*,i})|_g\}$  is appended to a process database  $M$ , the ‘memory’. The procedure repeats for the remaining geometries in  $G$ .

During the backward pass  $\mu_p$ ’s weights will be adjusted with the aid of the auxiliary critic  $\mu_{aux}$ . To this end, at first the critic parameters  $\theta_{aux}$  are updated by a classical batch-gradient descent step, cf. Equation (2.6). More specifically, the prediction error  $\varepsilon_M$  between the actual quality  $q$





**Figure 5.4:** Summary of the implemented Actor-Critic Training scheme: For each shear strain estimation  $\gamma_{12} = \mu_f(g)$  in each training iteration  $i$ , either random process parameters are drawn (exploration) or the actor  $\mu_p$  issues a process recommendation (exploitation). After simulation and result storage in a memory, the auxiliary critic-network then informs the actor-network by the gradient  $\nabla_{\theta_p} q_{srg}$  how to improve its recommendations for the next iteration  $i + 1$  [104].

stored in the memory  $M$  and the critic-estimation  $q_{srg}$  is stepwise reduced by

$$\theta_{aux}^{i+1} = \theta_{aux}^i - \eta \nabla_{\theta_{aux}} \varepsilon_M \quad . \quad (2.6 \text{ revisited})$$

Thereby, the critic gradually learns to estimate the resulting quality for a given geometry-process-combination.

A gradient-descent also updates the actor recommendations  $p_{srg}^{*,i}$  in direction of improving part quality estimation  $q_{srg}$  according to

$$\theta_p^{i+1} = \theta_p^i - \eta \nabla_{\theta_p} q_{srg} \quad . \quad (2.6 \text{ revisited})$$

That is, in the next iteration it should yield an improved quality.

The gradient  $\nabla_{\theta_p} q_{\text{srg}}$  is at first unknown, but can be obtained with the aid of the critic. Following the chain rule of calculus,  $\nabla_{\theta_p} q_{\text{srg}}$  decomposes into

$$\nabla_{\theta_p} q_{\text{srg}} = \nabla_{\mathbf{p}} q_{\text{srg}} \cdot \nabla_{\theta_p} \mathbf{p} \quad . \quad (5.3)$$

Therein, the first factor  $\nabla_{\mathbf{p}} q_{\text{srg}}$  is the gradient of the critic-network  $\mu_{\text{aux}}$  with respect to the process parameters  $\mathbf{p} = (p_1, p_2, \dots)^T$ , i.e.

$$\nabla_{\mathbf{p}} q_{\text{srg}} = \left( \frac{\partial q}{\partial p_1}, \frac{\partial q}{\partial p_2}, \dots \right)^T = \left( \frac{\partial \mu_{\text{aux}}}{\partial p_1}, \frac{\partial \mu_{\text{aux}}}{\partial p_2}, \dots \right)^T \quad . \quad (5.4)$$

The second factor  $\nabla_{\theta_p} \mathbf{p}$  is the Jacobian of the actor  $\mu_p$ . The  $jl$ -th entry is the partial derivative of the  $j$ -th process parameter  $p_j$  with respect to the  $l$ -th actor parameter  $\theta_{pl}$ :

$$\frac{\partial p_j}{\partial \theta_{pl}} = \frac{\partial \mu_{pj}}{\partial \theta_{pl}} \quad . \quad (5.5)$$

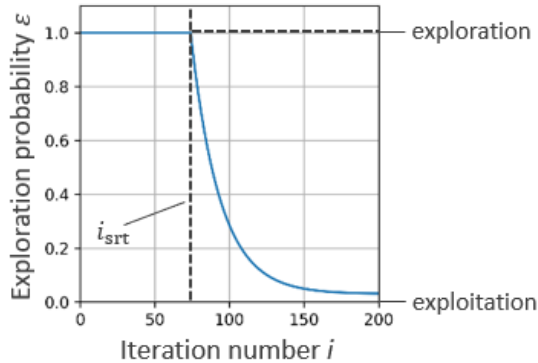
**Exploration-exploitation dilemma.** The parameters of both  $\mu_p$  and  $\mu_{\text{aux}}$  are initialised randomly. As a result, parameter recommendations  $\mathbf{p}_{\text{srg}}^{*i}$  and gradient-information  $\nabla_{\theta_p} q_{\text{srg}}$  are of little significance at training begin. Directly following the initial gradient-descents ('greedy' exploitation) most likely converges prematurely to a poor local optimum of the actor policy. Thus, additional iterations are necessary to explore alternative policies and increase the chance of a global policy optimum. However, since they add computational effort and prolong training time, they must be balanced with policy fine-tuning (exploitation).

How to optimally reconcile this exploration-vs-exploitation dilemma is still subject of ML-research. This work employs a variant of 'ε-greedy' exploration. Actor updates are delayed until the critic has reached a sufficient accuracy. Until then, parameters  $\mathbf{p}$  are drawn randomly from a uniform distribution  $\mathcal{U}(P)$  for policy exploration (cf. Figure 5.4). Once

the actor training starts ( $i \geq i_{\text{srt}}$ ), the probability  $\epsilon \in [0, 1]$  of probing random parameters decreases exponentially by

$$\epsilon = (\epsilon_{\text{srt}} - \epsilon_{\text{end}}) \epsilon_{\text{d}}^{i - i_{\text{srt}}} . \quad (5.6)$$

Therein,  $\epsilon_{\text{srt}}$  and  $\epsilon_{\text{end}}$  denote the start and end values of  $\epsilon$  and  $\epsilon_{\text{d}}$  is a decay factor with  $0.95 \leq \epsilon_{\text{d}} \leq 0.99$  in general. Figure 5.5 shows an example graph of  $\epsilon$  for  $\epsilon_{\text{srt}} = 1.0$ ,  $\epsilon_{\text{end}} = 0.03$ ,  $\epsilon_{\text{d}} = 0.95$  and  $i_{\text{srt}} = 75$ . Overall, Equation (5.6) gradually shifts from random parameter *exploration* ( $\epsilon \approx 1$ ) to deterministic *exploitation* ( $\epsilon \approx 0$ ). Note however, that  $\epsilon$  only affects actor recommendations; training of critic network  $\mu_{\text{aux}}$  on the process memory  $M$  is independent of  $\epsilon$  and takes place in each iteration.



**Figure 5.5:** Example evolution of the exploration probability  $\epsilon$  over the training iterations  $i$  [265].

## 5.2 Validation of Reinforcement Learning for process optimisation<sup>1</sup>

Validation of the RL-based optimisation approach splits up into four separate parts: After an introduction to the considered use case (Section 5.2.1), a general assessment of the training behaviour is presented (Section 5.2.2), followed by a study to assess the effects of different hyperparameters for optimal training (Section 5.2.3). Once suitable hyperparameters have been found, the method is applied to the textile forming use-case (Section 5.2.4). Eventually, Section 5.3 compares RL-based method to the SBO-method from Chapter 3 and a state-of-the-art GA and concludes with a intermediate summary (Section 5.4).

### 5.2.1 Validation use case: Pressure-pad assisted fabric forming

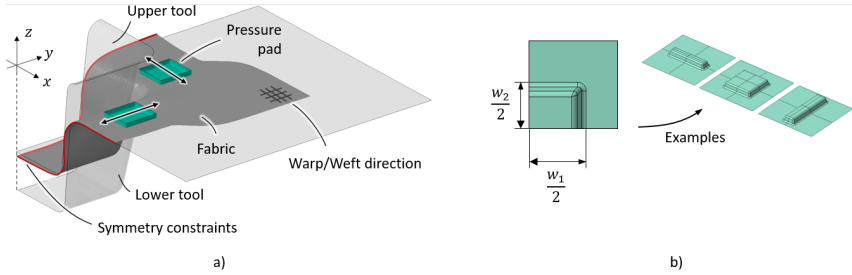
This section presents the use-case for validation of the RL-based optimisation approach. Specifically, it considers optimisation of pressure pad positions  $\mathbf{p} = (p_1, p_2)^T$  during draping of a woven fabric single layer into corners of variable cuboid geometries  $g$  similar to Figure 4.2. For comprehensibility during algorithm development in this work a simple, double-symmetric FE-model with symmetry constraints is considered. Additionally, an analytical substitute model is temporarily employed for efficient algorithm development and testing. Once this is completed, all techniques are applied to FE-models.

**FE-model.** As shown in Figure 5.6 for the double-symmetric variant, the quadratic pressure pads ( $25 \times 25$  mm) can be positioned along the component perimeter and show a 25 mm distance to the stamp opening line. The process aligns with recent studies by JAGPAL et al. [71, 266] and

---

<sup>1</sup>The presented results are based on the prior works [265, 104]

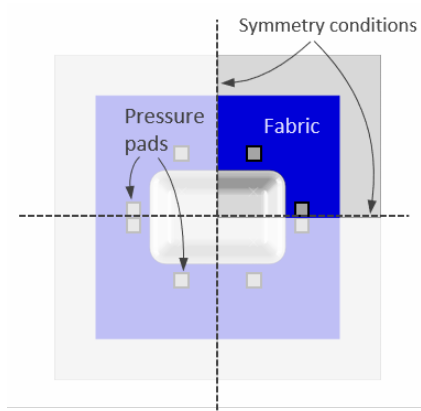
allows for highly flexible clamping and convenient reconfigurability. A Coulomb friction law with a constant, isotropic coefficient of friction of 0.25 and pad downforces of 7N impose tangential stresses which control the deformation of the fabric during draping.



**Figure 5.6:** a) Schematic illustration of an exemplary process simulation setup for pressure-pad assisted fabric forming with applied symmetry conditions, b) Parameter definition for the box-geometries along with geometry examples for visualisation [104].

The fabric is quadratic and measures  $w_f \times l_f = 600 \times 600$  mm in width and length with a constant thickness of 0.3 mm. Warp and weft direction align with the  $x$ - and  $y$ -axis, respectively. To facilitate engineering interpretation of the results and the RL-algorithm behaviour, this work focuses on cuboid geometries whose corners are deliberately severe in order to evoke strong defect tendencies (cf. Figure 5.6 b)). While the geometries' height is fixed (150 mm), their width and length  $w_{1,2}$  are bound relative to the fabric according to  $\frac{1}{6} \leq \frac{w_{1,2}}{w_f} \leq \frac{1}{2}$ . Within these bounds LHS (cf. Section 2.3.3) ensures a space-filling sampling of the geometry parameters.

The forming simulation uses a macroscopic FE-based approach, which captures the relevant forming defects of the studied fabric. Membrane and bending models have been implemented and parametrised by POPPE et al. [100, 38] for the balanced plain weave carbon fabric T700SC-12K-50C by ZOLTEK based on experimental characterisation. The material models are embedded within the commercially available FE-solver



**Figure 5.7:** Sketch of the forming simulation model for utilisation of symmetry conditions (top view). Two pressure pads per quarter model are used.

ABAQUS/explicit using VUMAT and VUGENS user-subroutines to account for textile-specific characteristics outlined in Section 2.2.3. The fabric mesh is a structured triangular mesh and aligned with the initial fibre orientations to prevent numerical locking effects. ABAQUS' built-in general contact algorithm models the tool-ply interface. The tools are modelled as rigid surfaces and close in 2 s at constant speed.

**Part quality quantification.** As outlined in Section 2.2.1 the shear angle  $\gamma_{12}$  is a frequently used proxy for wrinkling during forming process optimisation. However, the more  $\gamma_{12}$  exceeds the locking angle  $\gamma_{12}^{\text{lock}}$ , the more it loses expressiveness as wrinkling may intensify although  $\gamma_{12}^{\text{max}}$  remains practically constant. The corners of the considered boxes are set so severe that this effect occurs. Thus, this work applies a more direct measure for wrinkling as a part quality attribute  $a$ , namely the modified mean Gaussian curvature  $\kappa$  proposed by HAANAPPEL [28]. Again  $\kappa$  is a field quantity and requires a mapping to a scalar quality metric  $q \in Q \in \mathbb{R}$  for optimisation (cf. Section 3.1.1)

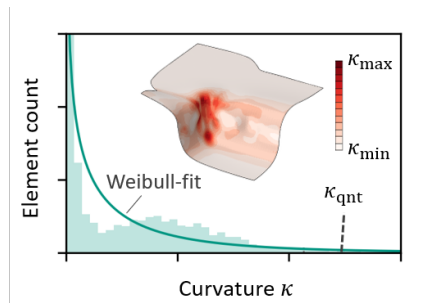
To this end, a common approach is the vector norm according to Equation (3.1)

$$q(\mathbf{a}) = \|\mathbf{a}\|_u = \left( \sum_{i=1}^{n_{\text{el}}} |a_i|^u \right)^{1/u} \quad (3.1 \text{ revisited})$$

where  $u = 1$  penalises the mean and  $u \rightarrow \infty$  progressively penalises the maximum. In prior work [152, 181],  $u = 4$  has been empirically found as a useful trade-off for shear deformation.

However, applying Equation (3.1) to the curvature  $\kappa$  yields erratic results in practice and proves susceptible to physically implausible outliers. In such cases literature recommends characterising the defects by a global statistical distribution [150] instead of a local criterion. Figure 5.8 shows a histogram of  $\kappa$  along with an exemplary forming result for visualisation. While the vast majority of the fabric experiences only mild curvature, few elements show severe curvature. Under such conditions, a sufficiently high quantile  $\kappa_{\text{qnt}}$  of a fitted Weibull distribution proves a robust proxy for the curvature maximum [150]. Specifically, this work evaluates the curvature distribution  $\kappa$  by the 99.5%-quantile, i.e.

$$q = \kappa_{\text{qnt}}(\kappa) = \kappa_{99.5} \quad (5.7)$$



**Figure 5.8:** Exemplary fabric forming result and histogram of the local fabric curvature. The histogram also shows a fitted Weibull distribution, whose 99.5%-quantile quantifies part quality during optimisation [104].

**Phenomenological substitute.** Developing and testing the outlined SBO-techniques depends on an appropriate algorithm configuration and hyperparameter tuning, which in turn requires extensive parameter studies with numerous simulations. Performing these studies with actual FE-simulations quickly becomes prohibitively expensive, since each simulation  $\varphi$  and quality evaluation  $q(\varphi)$  take in total about 20...60 min to solve.

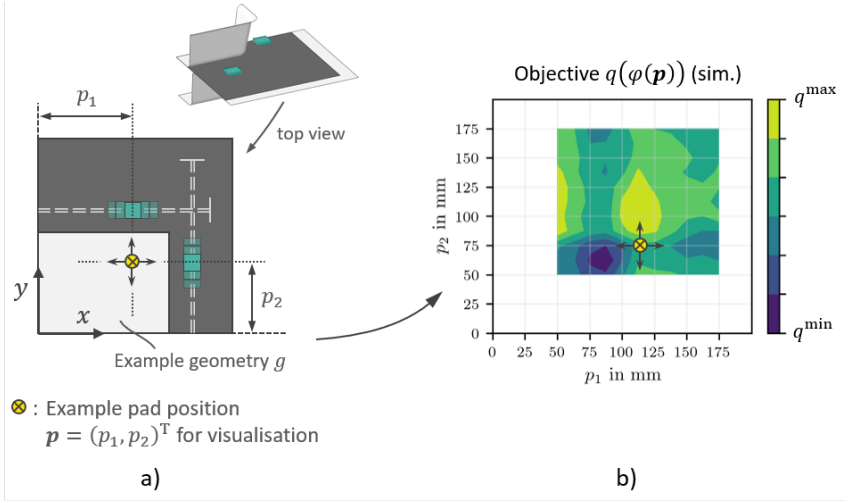
As a remedy, this work utilises the deliberate simplicity of the box-geometries to formulate an analytical substitute function  $q_{\text{sub}}$  of the quality. The substitute function *temporarily* replaces the actual simulations during algorithm development and hyperparameter tuning. Once appropriate hyperparameters are determined, the substitute will be discarded.

The underlying reasoning is outlined in the following. Figure 5.9 a) visualises the process variations schematically and sub-plot b) shows a contour plot of the according process responses  $q(\varphi(\mathbf{p}))$  from FE simulation. The yellow marker illustrates how a specific pad position relates to the process response. Clearly, the process responses show a minimum and diagonally above a maximum.

The location of maxima and minima directly correlates to the geometry width and length as Figure 5.10 shows. It features the process responses obtained by FE-simulations  $\varphi$  for two geometry examples  $g_1$  and  $g_2$ . In both cases, the process optimum (blue markers) lies below and left of the corner, while the maxima (worst pad positions) coincide with corners (grey markers). Consequently, the positions of the maxima and minima moves as the geometry changes. Closer examination further reveals that the difference  $\Delta_p$  between minimum and maximum (black arrow) is approximately constant. These observations can be used to construct the substitute function  $q_{\text{sub}}$ .

The bottom plots show the results of the substitute function  $q_{\text{sub}}$ . It is an analytical relation defined by





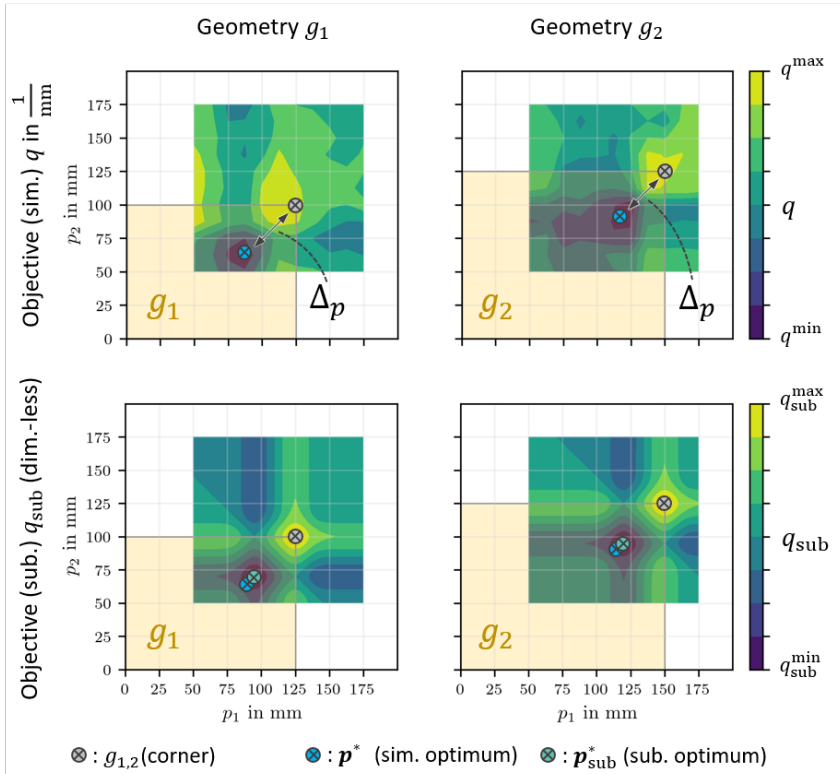
**Figure 5.9:** a) Example visualisation of the process variation through repositioning of pressure pads and b) contour plot of the objective function  $q(\varphi(\mathbf{p}))$  of an example geometry  $g$  obtained through full-factorial sampling of all possible pad positions. Depending on the pad position (yellow marker), the process response can be varied with a distinct clear optimum (minimum) at  $\mathbf{p} \approx (80, 65)$  mm [104].

$$q_{\text{sub}}(\mathbf{p})|_g = q_{\text{sub}}(p_1, p_2)|_g \quad (5.8)$$

$$= q_{\text{sub}1}(p_1)|_g + q_{\text{sub}2}(p_2)|_g \quad (5.9)$$

wherein

$$q_{\text{sub}1,2}(p_{1,2})|_g = \frac{1}{2} \left[ \exp\left(-a_{1,2}|_g^2\right) - \exp\left(-b_{1,2}|_g^2\right) \right] \quad (5.10)$$



**Figure 5.10:** Juxtaposed objective functions  $q(\varphi(\mathbf{p}))$  for two different geometries  $g_1$  and  $g_2$  obtained from FE process simulations (top) and the devised substitute function  $q_{\text{sub}}(\varphi(\mathbf{p}))$  (bottom) for temporary replacement of  $q(\varphi(\mathbf{p}))$  during hyperparameter studies [104].

with

$$a_{1,2}|_g = a(p_{1,2})|_g = \frac{p_{1,2} - p_{1,2}^*|_g}{5 \text{ mm}} \quad (5.11)$$

$$b_{1,2}|_g = b(p_{1,2})|_g = \frac{p_{1,2} - \tilde{p}_{1,2}|_g}{5 \text{ mm}} \quad (5.12)$$

Therein,  $\tilde{p}_{1,2} = w_{1,2}$  denotes the worst and  $p_{1,2}^* = w_{1,2} - \Delta_{p_{1,2}}$  with  $\Delta_{p_{1,2}} = 30 \text{ mm}$  the best (optimal) position.

Essentially, Equations (5.8)–(5.12) yield a sum of positive and negative bell-curves. The curves are shifted such that they generate a maximum at  $\tilde{\mathbf{p}} = (\tilde{p}_1, \tilde{p}_2)^T = (w_1, w_2)^T$  and an adjacent minimum at  $\mathbf{p}^* = (p_1^*, p_2^*)^T = (\tilde{p}_1 - \Delta_{p_1}, \tilde{p}_2 - \Delta_{p_2})^T$ . Note that  $q_{\text{sub}}$  is dimensionless and normalised to the interval  $[-1, 1]$  for a value-wise distinction from a actual simulation-based quality metric  $q(\varphi)$ .

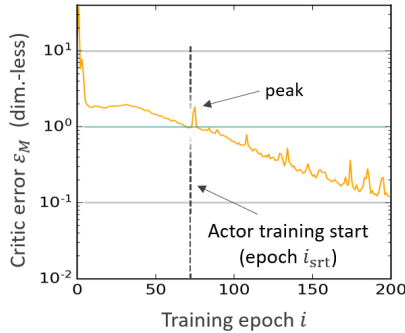
This analytical relation seeks to *phenomenologically* reproduce the relative position of minimum and maximum as a function of the geometry width and length  $w_{1,2}$ . Green markers locate the process optimum according to the substitute. Since they lie in close proximity to the simulation optimum (blue markers), the substitute can at least *qualitatively* reproduce the objective function. Owing to its negligible evaluation time, this substitute will be used for hyperparameter studies in this work. Once suitable hyperparameters are determined, the substitute will be discarded in favour of actual simulation.

## 5.2.2 General training behaviour

For discussion of the general training behaviour, in total 20 box-geometries (Figure 5.6 b) ) are generated – more specifically,  $n_g = 15$  training geometries and 5 validation geometries. They are converted to greyscale images (topographic maps) and in turn evaluated by  $\mu_f$  for 20 greyscale-images of shear angle distributions, cf. Figure 5.1. Since length and

width  $w_{1,2}$  are known for each geometry  $g$ , a substitute model  $q_{\text{sub}}(\mathbf{p})|_g$  is established (Equations (5.8)–(5.12)).

For assessment of the training behaviour on  $q_{\text{sub}}$ , at first the critic network is considered. To this end, Figure 5.11 shows the evolution of the critic-error  $\varepsilon_M$ . Within the first epochs ( $i \lesssim 10$ ) the error drops precipitously from  $\varepsilon_M \gg 30$  to  $\varepsilon_M \approx 2$ . Then the error stagnates for about 20 epochs ( $10 \lesssim i \lesssim 30$ ) before it reduces again in a roughly monotonous manner.



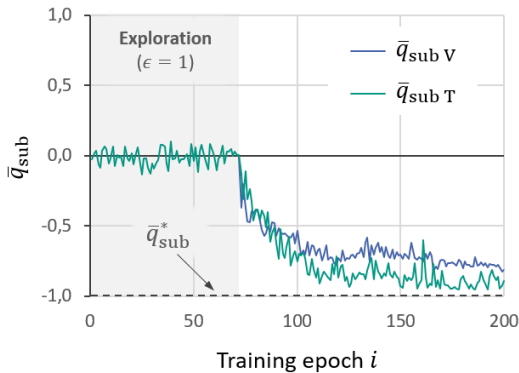
**Figure 5.11:** Evolution of the error of the auxiliary critic-network ( $\mu_{\text{aux}}$ ) during actor-critic-training of  $\mu_p$  [265]. The loss at first drops substantially, remains constant and ultimately declines in a roughly monotonous manner towards zero, which implies that the critic network learned the underlying process dynamics.

This behaviour can be explained as follows: When the first observations enter the memory  $M$ ,  $\mu_{\text{aux}}$  simply adjusts to  $q_{\text{sub}}$ 's range of values  $[-1, 1]$ . During the intermediate plateau ( $10 \lesssim i \lesssim 30$ ) a growing number of observations with new information on the system dynamics becomes available which  $\mu_{\text{aux}}$  ingests and adapts to. The continuous reduction of the error hints that from this moment on ( $i \gtrsim 30$ ), incoming observations are rather recurring and introduce only ever less new information on the system behaviour to  $\mu_{\text{aux}}$ . In loose terms,  $\mu_{\text{aux}}$  has sufficient information of the underlying process dynamics, captures already general trends and only refines its accuracy.

In iteration  $i = i_{\text{srt}} = 74$ ,  $\varepsilon_M$  falls below  $1.0^{\text{o}2}$  (green horizontal line) which has empirically proven a useful start signal for actor training. At this time also the exploration probability  $\varepsilon$  starts decaying according to Equation (5.6). That is, rather than probing random parameters,  $\mu_p$  progressively follows its own policy.

Interestingly, directly with the start of actor training ( $i = 74$ ), the critic-error  $\varepsilon_M$  doubles and peaks back to the plateau-level for few iterations. This may happen because the critic is – similar to Section 3.1.3 – riddled with false minima and erroneously guides the actor to these minima. When probing them, large differences between  $q_{\text{sub}}$  and the critic estimation become apparent which are – analogous to the initial deviations in Figure 3.13 – gradually eliminated in subsequent iterations.

The quality of the process recommendations of  $\mu_p$  is quantified by  $q_{\text{sub}}$ . Since multiple geometries are evaluated in each epoch, the process responses  $q_{\text{sub}}$  are geometry-averaged ( $\bar{q}_{\text{sub}}$ ). Figure 5.12 shows the evolution of  $\bar{q}_{\text{sub}}$  during training.



**Figure 5.12:** Evolution of the average process response  $\bar{q}_{\text{sub}}$  during actor-critic-training. After an initial exploration phase, the process responses on training and validation geometries rapidly improve and approach the optimal response ( $\bar{q}_{\text{sub}} = -1.0$ ) [265].

Figure 5.12 features two graphs, the averaged substitute process response for the training geometries  $\bar{q}_{\text{sub T}}$  and the validation geometries  $\bar{q}_{\text{sub V}}$ . A dashed horizontal line marks the best-possible response  $\bar{q}_{\text{sub}}^* = -1.0$ , i.e. when the actor predicts for each geometry the optimal parameters.

During the initial exploration phase with ( $\epsilon = 1.0$ , grey shade), process parameters are random and consequently, the according process responses randomly vary around  $\bar{q}_{\text{sub T}} = 0$ , hence the noisy appearance. These ‘exploration samples’ serve only to build an initial memory  $M$ . Note that although individual responses  $q_{\text{sub}}$  vary between  $[-1, 1]$ , taking their average  $\bar{q}_{\text{sub}}$  reverts the graph to zero. Further note that during the initial exploration, no training of the actor  $\mu_{\text{p}}$  takes place and any evaluation on the validation geometries would be meaningless. Consequently,  $\bar{q}_{\text{sub V}}$  (blue) is omitted.

Once actor training starts in epoch  $i = 74$ , its model parameters  $\theta_{\text{p}}$  are stepwise adjusted in direction of improved (reduced) process response. Both graphs,  $\bar{q}_{\text{sub T}}$  and  $\bar{q}_{\text{sub V}}$ , display a similar shape. Some volatility aside, at first both curves decline steeply ( $74 \leq i \leq 90$ ) and then they gradually level out in an asymptotic manner. Their final values are  $\bar{q}_{\text{sub T}} \approx -0.9$  and  $\bar{q}_{\text{sub V}} \approx -0.8$ .

The response for the training geometries  $\bar{q}_{\text{sub T}}$  is nearly constantly lower, i.e. of higher quality, than for the validation geometries ( $\bar{q}_{\text{sub V}}$ ). This behaviour agrees with classical surrogate modelling: While the surrogate fits to the training data, a transfer to unknown (validation) data usually introduces some error. Compared to the best-possible value ( $\bar{q}_{\text{sub}} = -1.0$ ) and the value range  $[q_{\text{sub}}^{\min}, q_{\text{sub}}^{\max}] = [-1.0, 1.0]$  this implies an exhaustion of the optimisation potential of 95% (training geometries) and 90% (validation geometries), respectively. Thus, the actor network  $\mu_{\text{p}}$  has successfully transferred ‘knowledge’ from training geometries and applied it to new scenarios. It is emphasised that  $\mu_{\text{p}}$  can give recommendations for a whole *range of geometries* – unlike a classical, application-specific surrogate.

### 5.2.3 Hyperparameter study for efficient training

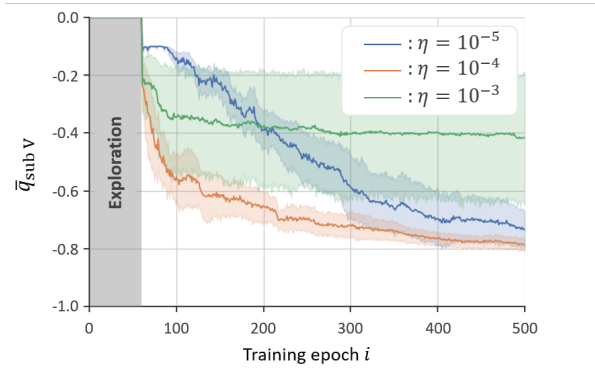
Analogous to surrogate models, ML-models generally require tuning of hyperparameters for optimal performance and also applies to this work. To this end, often extensive parametric studies are performed. Since this quickly becomes time-consuming with actual simulations, the substitute-function  $q_{\text{sub}}$  (Equations (5.8)–(5.12)) is used again.

In upstream studies, three hyperparameters have been identified as most decisive and will be discussed in the following: 1) the actor learning rate  $\eta$ , 2) the use of data augmentation techniques and 3) the number of training geometries  $n_g$ .

Owing to their random weight-initialisation, individual training runs show scatter and thus, analogous to Figures 4.27–4.28, ten independent training runs are performed for each hyperparameter configuration to allow for statistical performance evaluation. The following graphs visualise their averaged performance  $\bar{q}_{\text{sub V}}$  on the five validation geometries and the 95 %-percentile.

**Learning rate.** According to [267], one of the most decisive hyperparameters for ANNs is the learning rate  $\eta$  (Equation (2.6)) for which a typical value range of  $10^{-6} < \eta < 1$  is reported. In general, smaller values for  $\eta$  stabilise the training process and thus, complex problems typically require smaller learning rates [268]. Thus,  $\eta$  can serve as a rough, qualitative assessment of the problem complexity. However, this comes at the cost of prolonged training, which leads to the notion of ‘as large as possible, as small as necessary’.

The suggested range stretches over six decades thus will be varied on a logarithmic scale. Figure 5.13 shows the evolution of  $\bar{q}_{\text{sub V}}$  for  $\eta = 10^{-3} \dots 10^{-5}$ . Lower and higher values were also tested but did not converge at all. Therefore, they are excluded. In all cases,  $n_g = 16$  geometries were used for algorithm training.



**Figure 5.13:** Effect of different learning rates  $\eta$  during training of  $\mu_p$  on  $n_g = 16$  geometries [104].

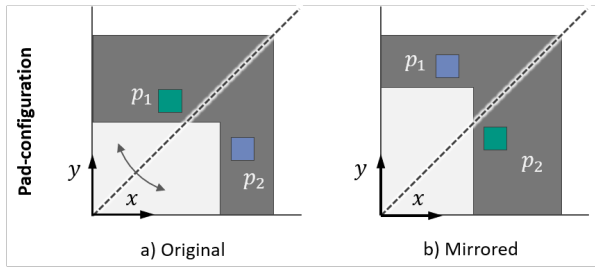
Overall, the graphs share the same characteristic as in Figure 5.12: After an initial exploration phase ( $i \leq i_{\text{srt}} = 75$ ) the graphs decrease and gradually approach a horizontal asymptote. The shape however varies and reveals performance differences: The graph of  $\eta = 10^{-4}$  performs best as it yields the lowest final value ( $\bar{q}_{\text{sub V}} \approx -0.8$ ), declines most rapidly and shows least scatter. Conversely,  $\eta = 10^{-3}$  performs worst. Although it converges rapidly, the final value ( $\bar{q}_{\text{sub V}} \approx -0.4$ ) is inferior. Note that  $\eta = 10^{-5}$  may have also reached a good final value, but converges much slower than  $\eta = 10^{-4}$ . Comparing  $\eta = 10^{-4}$  to the suggested value range  $10^{-6} < \eta < 1$  implies a medium-complex problem. A further fine-tuning of the learning rate resulted in a final value of  $\eta = 1.5 \cdot 10^{-4}$  for the rest of this work.

**Data augmentation.** Generally, ML-algorithms improve with available data. However, in many practical applications, datasets are limited and acquisition of further observations is expensive. A common strategy to improve the performance of ML-algorithms is to conjure additional data by careful variation of existing observations – typically referred to as ‘data augmentation’. For instance, in image recognition, transformations



like translation, rotation, mirroring, scaling, cropping or distortion can be applied, which alter the image but not its semantic content [269].

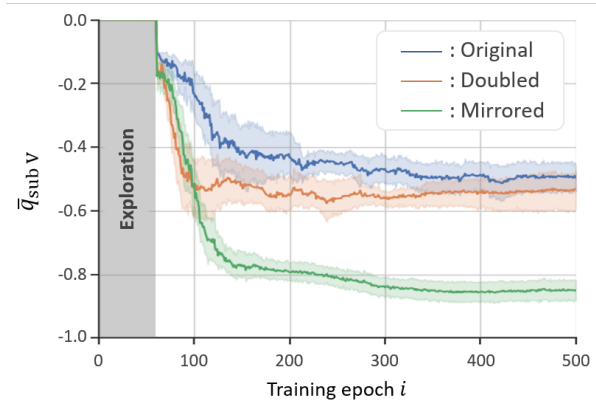
Since pixels carry spatial information, they must not be scaled or distorted and the eligible transformations for the considered use-case (fabric forming) reduce to rotating and mirroring. In the considered case, the applied symmetry conditions (Figure 5.6) additionally preclude rotation and leave mirroring about the  $x$ - $y$ -bisector as the remaining option. The considered fabric (Section 3.2.1) is a balanced, plain weave fabric. Hence, it shows identical behaviour in warp and weft direction, which introduces an additional symmetry plane at  $45^\circ$  as Figure 5.14 visualises. Note that both pad configurations yield the same (mirrored) forming result and give an identical quality metric.



**Figure 5.14:** The symmetry line at  $45^\circ$  during pressure pad assisted forming offers to conjure one additional observation per simulation by mirroring (data augmentation) [104].

Figure 5.15 illustrates the effect of data augmentation on the training behaviour. Specifically, it shows the training progresses for the learning rate  $\eta = 1.5 \cdot 10^{-4}$  and  $n_g = 12$  training geometries. It features three graphs with different data strategies, namely, 1) original data without any treatment as a ‘reference’, 2) original and mirrored data in the memory  $M$  and 3) doubled-data. The latter (doubled data) implies original data that is just copied and appended to the memory  $M$ . This introduces no additional information but ensures that the memory  $M$  has the same size as the augmented dataset. Otherwise, the training behaviour would

be deceptively distorted. A comparison of original data (blue) and the doubled data (orange) visualises this distortion-effect.



**Figure 5.15:** Effect of data augmentation by mirroring during training of  $\mu_p$  on  $n_g = 12$  geometries [104].

Even though they use exactly the same information content, the graph of the doubled dataset declines faster than the original dataset. However, it would be premature to conclude that doubling the data improves the learning behaviour. The number of gradient descents per epoch is directly proportional to the number of observations in a dataset. Thus, a larger dataset leads to more gradient-descents, i.e. weight adaptations, and an *seemingly* faster training per epoch. Mirroring the data thus inherently results in twice as many weight-adaptations per epoch. Thus, for a valid assessment of the effect of data augmentation, mirrored data must be compared to doubled data.

Overall, the graphs are of the same characteristic as before. Both graphs, doubled and mirrored, initially decline at the same rate; however, their final values differ: While the doubled data approaches  $\bar{q}_{\text{sub } V} \approx -0.5$ , the mirrored graph achieves  $\bar{q}_{\text{sub } V} \approx -0.85$ . Consequently, exploitation of known symmetries by data augmentation does indeed introduce useful information and improves the learning performance at negligible cost.

Similar speed-ups were also observed for RTM form filling optimisation in the master thesis of GEISENDÖRFER<sup>1</sup> [270]. Note that in the presented forming example only one symmetry plane remains for augmentation. The augmentation impact presumably grows when more symmetry are present.

**Number of training geometries.** The third important hyperparameter for the training performance is the number  $n_g$  of training geometries. Clearly, using only few geometries will not provide a sufficiently comprehensive process memory  $M$  to enable generalisation to new geometries. Conversely, when already using many geometries, adding even more geometries probably introduces only little additional information. Thus, analogous to Figure 4.27, a *threshold number* of training geometries must exist, which provides just sufficient information for generalisation.

In order to validate this hypothesis,  $\mu_p$  is trained with different numbers of geometries, namely  $n_g = 8, 10 \dots 20$ . Figure 5.16 shows four examples of the training progress. Again, the learning rate  $\eta = 1.5 \cdot 10^{-4}$  is used and data augmentation (mirroring) is employed. The graphs' shapes show little noteworthy novelty (initial decline and asymptote). However, the final values and the scatter differ.

In general, the more geometries are used, the better (lower) the final value. For instance, while  $n_g = 8$  only yield a final value of  $\bar{q}_{\text{sub V}} \approx -0.55$ ,  $\bar{q}_{\text{sub V}} \approx -0.94$  is achieved for  $n_g = 20$ . And not just the final value improves with more geometries, also graphs show less scatter, i.e. a more stable training is observed. However, the improvement is non-linear. While the final value improves from  $\bar{q}_{\text{sub V}} \approx -0.5$  ( $n_g = 8$ ) to  $\approx -0.83$  ( $n_g = 12$ ) with only four additional geometries, it takes another 8 geometries to improve it to  $\approx 0.94$  ( $n_g = 20$ ). This hints a saturation effect. Figure 5.17 further illustrates this issue in a bar plot.

---

<sup>1</sup>Master thesis at Institute for Operations Research of the Karlsruhe Institute of Technology (KIT-IOR) in cooperation with KIT-FAST; co-mentored by the author.

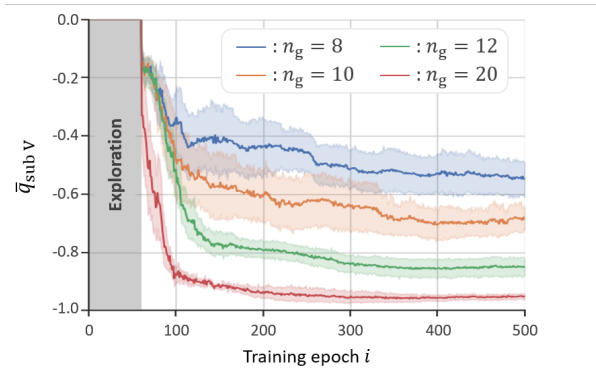


Figure 5.16: Effect of different numbers  $n_g$  of training geometries during training of  $\mu_p$  (data augmentation applied) [104].

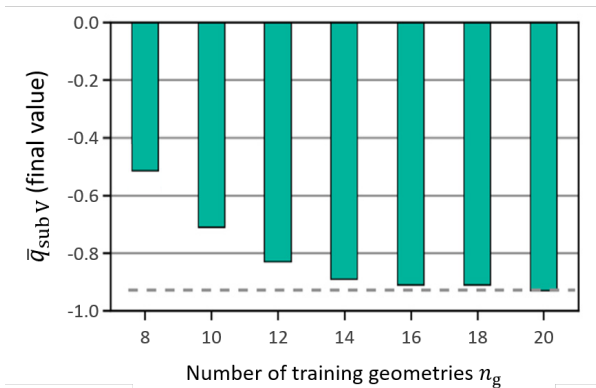


Figure 5.17: Bar plot of the best (lowest) value of  $\bar{q}_{\text{sub } v}$  for different numbers  $n_g$  of training geometries. Increasing  $n_g$  initially improves  $\bar{q}_{\text{sub } v}$  until it stays practically constant for  $n_g \geq 14$  [104].

The bars show the best (lowest) value of  $\bar{q}_{\text{sub} \vee}$  for each number of geometries. Overall, adding geometries constantly improves the training. However, the first additional geometries contribute most, which supports above saturation concept and the notion of a ‘marginal utility’ of additional geometries.

Furthermore, from approximately  $n_g = 14$  on, the final values of  $\bar{q}_{\text{sub} \vee}$  stay approximately constant. This hints a threshold of geometry samples, beyond which additional geometries contribute just little additional information. In general terms, the results suggest that a finite number of geometries from  $G$  holds sufficient information to analyse any new sample from  $G$ . This threshold will certainly grow when the considered geometries become more complex but it will presumably remain finite. This is a necessary precondition and an encouraging finding when striving for a fully geometry-independent forming-surrogate in the long run.

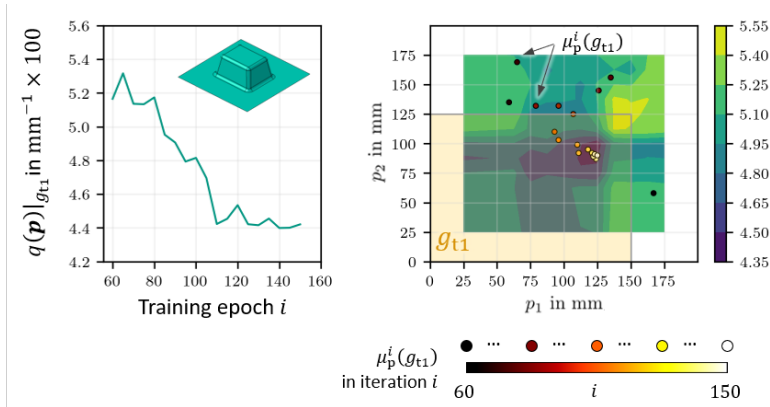
## 5.2.4 Application to the fabric forming use-case

The hyperparameter studies of the above Section 5.2.3 yield a promising configuration to train  $\mu_p$ . However, as these studies employ the substitute function  $q_{\text{sub}}$ , their actual proof of suitability is yet to prove on actual simulations  $\varphi$ . The following sections outline the results of an entirely FE-based training of  $\mu_p$ ; the substitute function  $q_{\text{sub}}$  is discarded from now on.

In accordance with Section 5.2.3,  $n_g = 14$  geometries in combination with data augmentation (mirroring) is applied. The training process is monitored by intermittent evaluations during algorithm training. More specifically, the training progress in three different situations is evaluated, namely on 1) training geometries and 2) validation geometries *inside* the class of box-geometries and additionally also on 3) new geometries *outside* this box-class of training geometries.

**Learning progress on *training* geometries.** Figure 5.18 exemplarily visualises the training progress for  $g_{t1}$ , i.e. one of the 14 training geometries. Analogous to Figures 5.13–5.16, the line plot on the left quantifies the part quality  $q(\mathbf{p})|_{g_{t1}}$  over the course of training along with an image of . To keep the computational load within reasonable bounds, it features just one training run instead of the average of ten runs.

The contour plot on the right allows to track the recommendations  $\mu_p$  issued during training. According markers locate  $\mu_p$ 's parameter recommendations during the course of training; their colour-code represents the order of appearance. In order to validate the optimality of the recommendations after 150 training epochs, the objective function  $q(\mathbf{p})|_{g_{t1}}$  of  $g_{t1}$  is separately sampled and plotted, analogous to Figure 5.9. Note that these samples are *by no means* involved in algorithm training but serve only for illustration of the training progress from an 'omniscient' perspective.

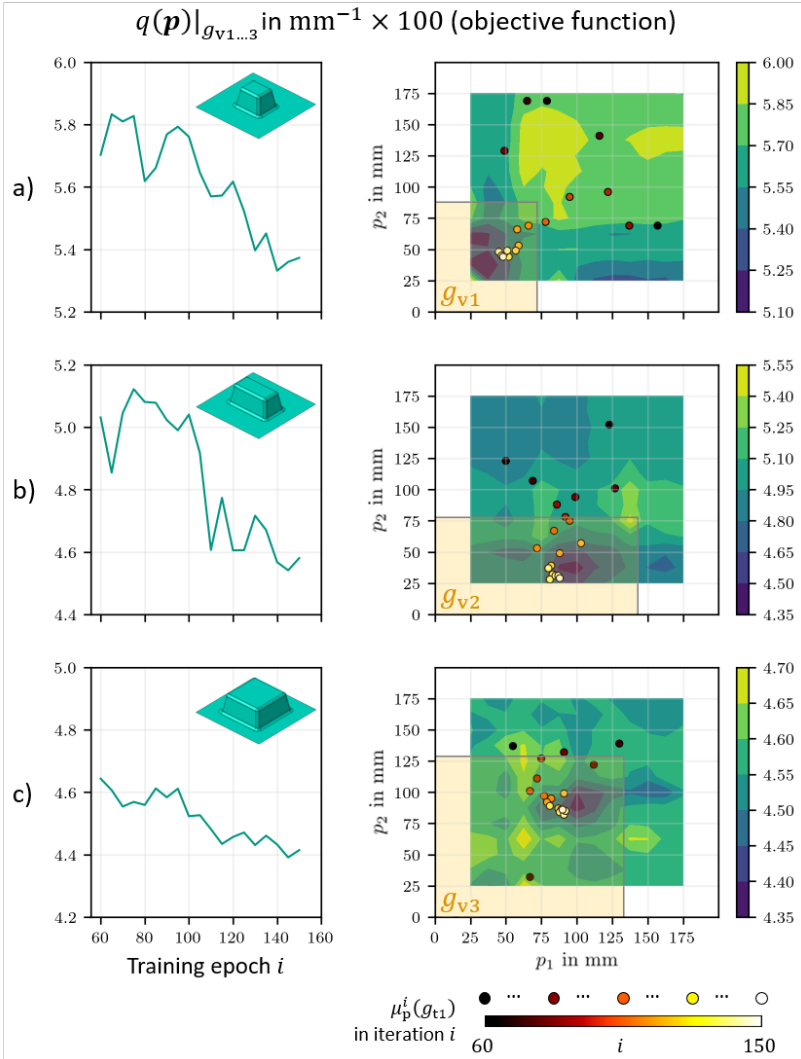


**Figure 5.18:** Contour plot of the objective function  $q(\mathbf{p})$  (contour plot) for an exemplary *training* geometry  $g_{t1}$ . Markers visualise  $\mu_p$ 's parameter recommendations during training; the line plot allows for convergence assessment of the process response  $q$ . After an initial scatter,  $\mu_p$ 's recommendations close in on the process optimum. Accordingly, the objective function decreases and implies successful optimisation behaviour [104].

The plots reveal a successful optimisation: The convergence plot (left) shows some fluctuation around  $q|_{g_{\text{tr}}} \approx 5.2 \times 10^{-2} \text{ mm}^{-1}$  at the beginning ( $60 \leq i \leq 80$ ). Then it decreases by  $\approx 13\%$  within 30 iterations ( $80 \leq i \leq 110$ ) to an approximately constant value of  $q|_{g_{\text{tr}}} \approx 4.5 \times 10^{-2} \text{ mm}^{-1}$ . In the contour plot (right), the initial markers (blackish) scatter significantly throughout the whole domain but begin to close in on the optimum (orange to white hue). Analogous results are obtained for the other training geometries. That is, during the course of training,  $\mu_{\text{p}}$  successfully learns to give near-optimal recommendations for its training geometries.

**Learning progress on *validation* geometries.** Having shown that  $\mu_{\text{p}}$ 's recommendations successfully converge to the optimum for training geometries (Figure 5.18), its behaviour on 'unknown' validation geometries needs analysis. Figure 5.19 presents analogous convergence diagrams and contour plots for three exemplary validation geometries  $g_{\text{v}1\dots 3}$ . Note that  $\mu_{\text{p}}$  is only *evaluated* for  $g_{\text{v}1\dots 3}$  but has never seen them during training. Simulations on  $g_{\text{v}1\dots 3}$  enter at no point the process memory  $M$ . Therefore the geometries are always new to  $\mu_{\text{p}}$  during evaluation.

The overall behaviour resembles the previous plots (Figure 5.18). The convergence plot displays an initial wavering before declining. Likewise the markers appear randomly distributed at the beginning (black hue) before contracting near the optima. However, two remarkable differences can be observed: First, though the markers approach the minimum, they do not exactly pinpoint it. More specifically, the estimated optimum lies top right of the true optimum in sub-image a) and bottom left in b) and c). Second, the initial fluctuations take a little longer ( $60 \leq i \lesssim 100$ ) compared to the training geometries ( $60 \leq i \lesssim 80$ , cf. Figure 5.18). This indicates that  $\mu_{\text{p}}$  takes more epochs to collect observations with sufficient information for generalisation to the (unknown) validation geometries than for the (known) training geometries.



**Figure 5.19:** Contour plots of the objective function  $q(\mathbf{p})$  (contour plot) for three exemplary validation geometries  $g_{v1\dots3}$  analogous to Figure 5.18 [104].



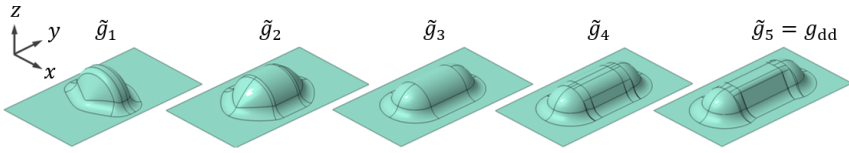
Bearing in mind the difference between training and validation geometries, the differences are well explicable: The geometry  $g_{t1}$  is part of the training geometries and during training  $\mu_p$  directly explores the according objective function  $q|_{g_{t1}}$ . Consequently, evaluating  $\mu_p$  on  $g_{t1}$  amounts just to ‘recalling’ information from previous process samples.

The opposite holds for the validation geometries  $g_{v1...3}$ , though: They are *not* part of the training geometries. Consequently, their objective functions  $q|_{g_{v1...3}}$  are unknown to  $\mu_p$  during evaluation and  $\mu_p$  must infer optimal parameters on the basis of prior training observations. Such inference of course introduces certain deviations, yet they appear acceptable as the process recommendations (markers) lie in close proximity of the actual optimum.

Overall, the results show that  $\mu_p$  is *principally* capable of extracting process-geometry-knowledge from the supplied training geometries and transferring it to unknown (validation) geometries.

**Transferability to new geometry classes.** The above results show that  $\mu_p$  can extract useful process-geometry-relations from training geometries and apply these relations to new, unknown samples from the same geometry class – in this work box-geometries. This result leads to the follow-up question, whether or not  $\mu_p$  can also give useful recommendations on geometries *outside* the class of training geometries.

To this end, a sequence of five different geometries is considered as visualised in the gallery in Figure 5.20. Starting off with a rather compact geometry  $\tilde{g}_1$ , the geometries gradually stretch in  $y$ -direction in steps of 20 mm while their ends morph from conical to spherical. The first three geometries also flatten in  $z$ -direction from  $z_{\max}(\tilde{g}_1) = 80$  mm to  $z_{\max}(\tilde{g}_3) = 60$  mm in two steps of 10 mm. Eventually, they yield the double-dome geometry  $\tilde{g}_5 = g_{dd}$ . Figure 5.21 b) sketches their contour lines for reference.



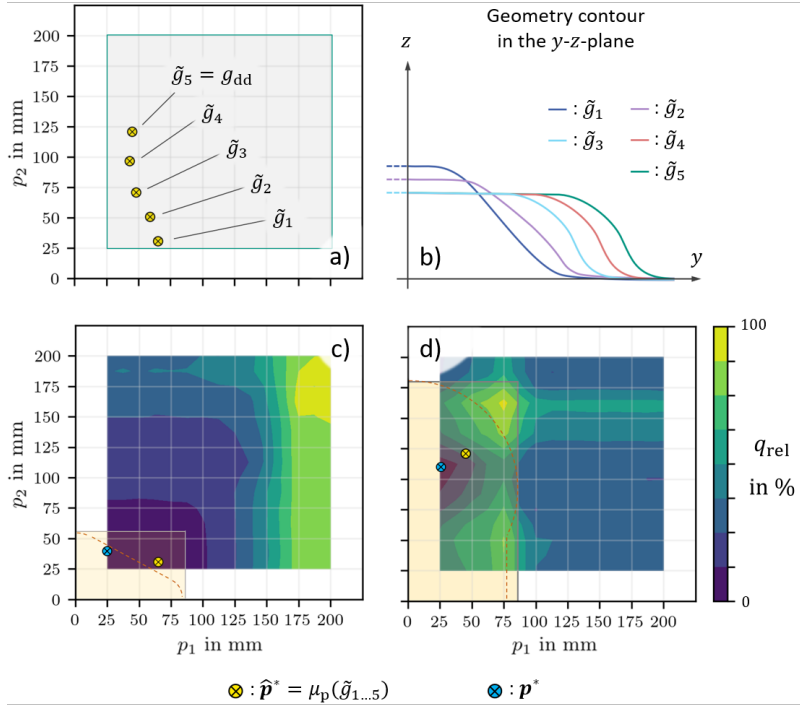
**Figure 5.20:** Visualisation of five test geometries  $\tilde{g}_{1\dots 5}$  outside the class of training geometries (boxes). Starting from a rather compact, rotational-symmetric geometry they gradually morph into the double-dome [104].

The geometries  $\tilde{g}_{1\dots 3}$  are rotational-symmetric about the  $y$ -axis, while  $\tilde{g}_{4,5}$  feature a trapezoidal cross-section in the middle. The trapezoid contracts from  $\tilde{g}_4$  to  $\tilde{g}_5$ , which creates the well-known concave indentation of the double-dome in the centre. Note that – despite some similarity – the box geometry-scheme (Figure 5.6) can reproduce neither of these geometries. Also note that  $\tilde{g}_{1\dots 5}$  are all near-convex and thus show a qualitatively similar forming behaviour as the training box-geometries.

Figure 5.21 summarises the results when evaluating  $\mu_p$  on  $\tilde{g}_{1\dots 5}$ . Specifically, sub-image a) displays the parameter recommendations of  $\mu_p$  for each geometry (yellow markers). For the two limit-cases  $\tilde{g}_{1,5}$  a full-factorial sampling has been performed and allows to compare  $\mu_p$ 's recommendations  $\hat{p}^*$  to the true optimum  $p^*$  (sub-plots c) and d)).

The yellow markers in sub-image a) show that  $\mu_p$ 's parameter recommendations follow the sense of the geometry change: Some wavering aside, the markers relocate in  $x$ - and  $y$ -direction as the geometries contract in  $x$  and stretch in  $y$ . These reactions of  $\mu_p$  to geometry changes appear plausible: When the geometry stretches in  $y$ -direction, the main deformation zone relocates and the pads follow this relocation.

On top of this brief sanity check, a more rigorous inspection of the recommendations is desirable. To this end, sub-images b) and c) show a contour plot of the objective functions ( $q|_{\tilde{g}_{1,5}}$  of  $\tilde{g}_1$  and  $\tilde{g}_5$ ) respectively, from a full-factorial sampling. For readability and comparability, both objective functions are normalised according to



**Figure 5.21:** a) Visualisation of  $\mu_p$ 's parameter recommendations  $\hat{\mathbf{p}}^*$  for the five test geometries  $\tilde{g}_{1..5}$ , b)  $\tilde{g}_{1..5}$ 's contour lines for comparison of height and length. Sub-plot c) and d) show  $\mu_p$ 's recommendations for the limit geometries  $\tilde{g}_{1..5}$ . A blue marker shows the true optimum  $\mathbf{p}^*$ , a yellow marker  $\mu_p$ 's parameter recommendation  $\hat{\mathbf{p}}^*$  after training on boxes. [104].

$$q_{\text{rel}} = \frac{q - q_{\text{min}}}{q_{\text{max}} - q_{\text{min}}} \cdot 100\%. \quad (5.13)$$

Consequently,  $q_{\text{rel}} = q_{\text{rel}}^* = 0\%$  implies the process optimum. For completeness, Table 5.1 summarises the values for  $q_{\text{min,max}}$ .

**Table 5.1:** Minimal and maximal values for  $q_{\text{min,max}}$  for  $\tilde{g}_1$  and  $\tilde{g}_5$ , respectively.

Geometry	$q_{\text{min}}$	$q_{\text{max}}$
	$\text{mm}^{-1} \times 100$	
$\tilde{g}_1$	2.81	3.42
$\tilde{g}_5$	2.39	2.80

The ochre-shaded areas mark the bounding box of the geometries; the dashed lines localise their plan view contour. The yellow markers locate  $\mu_p$ 's parameter recommendation; a blue marker the actual optimum of the objective function observed during the full-factorial sampling.

A perfect recommendation would mean that the blue and the yellow marker coincide, i.e. the recommendation matches perfectly well the true optimum. And indeed, an inspection of the contour plots shows that the markers lie in proximity to the actual optimum. An overestimation of  $\approx 18\%$  ( $\tilde{g}_1$ ) and  $\approx 13\%$  ( $\tilde{g}_5$ ) relative to the parameter range [25,200] mm is observed, though. They are still deemed acceptable bearing in mind that the geometries differ from the training geometries, e.g. regarding corner curvature. For  $\tilde{g}_1$  it must also be noted that the minimum and  $\mu_p$ 's recommendation both lie in a plateau-region with approximately equal quality  $q_{\text{rel}}$ .

Overall,  $\mu_p$ 's recommendations are deemed useful suggestions even for geometries *outside* the training geometry-class of boxes. Since it was only trained on box-geometries, it may be concluded that the employed RL-based approach has successfully extracted tacit process-knowledge from process-samples and is able to apply it to new situations. This

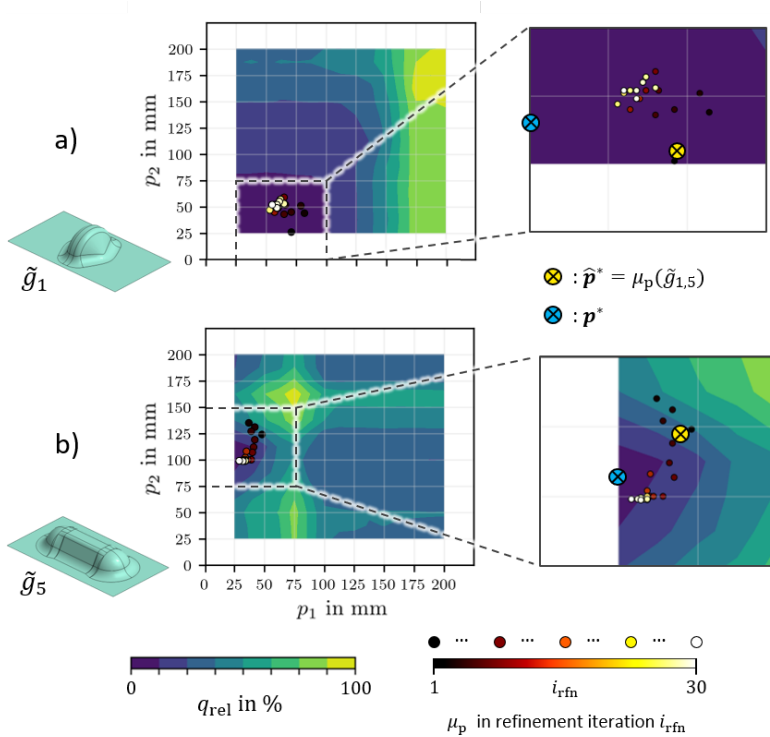
applicability across geometries significantly enhances current surrogate capabilities, which are typically geometry-specific ‘one-off’ models.

**Component-specific refinement.** Although the parameter estimations of  $\mu_p$  yield an acceptable approximation of the true optimum  $\mathbf{p}^*$  (Figure 5.21), some deviations remain. This resembles classical SBO, where the initial optimisation on the surrogate  $\mathbf{p}_{\text{srg}}^*$  also deviates from the true optimum  $\mathbf{p}^*$ . In SBO, additional refinement iterations sequentially eliminate these deviations in order to converge the surrogate optimum to the actual optimum, i.e.  $\mathbf{p}_{\text{srg}}^{*,i} \rightarrow \mathbf{p}^*$  for  $i \rightarrow \infty$ .

Likewise, this paragraph studies, whether or not  $\mu_p$ 's recommendations converge to the actual optimum  $\mathbf{p}^*$  upon component-specific refinement. That is, after the above presented, initial training on generic boxes,  $\mu_p$  interacts now only with geometry  $\tilde{g}_1$  or  $\tilde{g}_5$ , respectively. Thereby  $\mu_p$  can refine on these specific geometries.

Figure 5.22 shows the results in the same manner as before: A contour plot presents  $\mu_p$ 's recommendations by markers. Their hue represents their order of appearance. The contour plots are complemented by a detail view for closer inspection. Additionally, a blue and a yellow marker illustrate the actual optimum from full-factorial sampling and  $\mu_p$ 's original estimation. Again the two example geometries  $\tilde{g}_1$  (sub-plot a) ) and  $\tilde{g}_5$  (sub-plot b) ) from Figure 5.21 are considered. On each of these,  $\mu_p$  refines for thirty iterations in separate training runs.

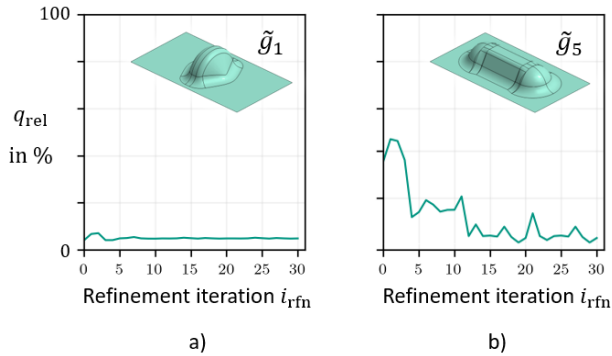
The plots reveal a disparate refinement behaviour. For geometry  $\tilde{g}_1$  (sub-plot a) ), the refined markers appear somewhat incoherently scattered around the initial recommendation with light tendency to the top left. In contrast, for  $\tilde{g}_5$  (double-dome, subplot b) ) the markers do accumulate and show a coherent evolution: At first, the markers move to the top left but, as this deteriorates the results, they reverse and gradually move down before concentrating near the minimum.



**Figure 5.22:** Visualisation of  $\mu_p$ 's parameter recommendations  $\hat{p}^*$  for the two limit geometries  $\tilde{g}_{1,5}$  during component-specific continuation of the training by colour-coded markers. Additionally, blue markers show the true optimum  $p^*$  and yellow markers denote  $\mu_p$ 's initial parameter recommendation  $\hat{p}^*$  immediately after training on boxes.

These observations can be explained when examining the objective function. Owing to its conical and rotational symmetric shape, the optimum pad position for  $\tilde{g}_1$  is not just a point but a plateau of approximately equal forming response. Thus, identifying a gradient for an improved pad position is difficult and the recommendations (markers) appear erratic. For  $\tilde{g}_5$  however, a distinct optimum is observable and – after few iterations for ‘orientation’ – the markers move in direction of improved part quality. For some reason however, they only approach but do not reach the observed optimum (blue marker). This may be due to numerical noise when evaluating the forming simulation.

Ultimately, Figure 5.23 quantifies the optimisation progress. For both geometries, it shows  $q_{\text{rel}}$  over the refinement epochs.



**Figure 5.23:** Plot of the evolution of the objective function  $q$  during component-specific training continuation on a)  $\tilde{g}_1$  and b)  $\tilde{g}_5$  for  $i_{\text{rfn}} = 30$  refinement iterations.

The diagrams support the above line of thought: Sub-image a) shows that the forming quality  $q_{\text{rel}}$  stays practically constant for  $\tilde{g}_1$  through all refinement iterations, while it improves (declines) for  $\tilde{g}_5$  after correction of an initial peak. Overall, the results indicate that – like a regular surrogate –  $\mu_{\text{p}}$  can indeed refine its initial recommendations with new evidence.

### 5.3 Comparison of optimisation methods

Ultimately, the three optimisation approaches – direct optimisation (GA), SBO and the RL – can be compared to each other. Each method is applied to the geometries  $\tilde{g}_1$  and  $\tilde{g}_5$  for inspection of the optimisation behaviour.

To this end, Figures 5.24 and 5.26 visualise the optimisation progress on geometry  $\tilde{g}_1$  and  $\tilde{g}_5$  for each optimisation approach. Specifically, sub-image a) visualises the results for the GA, b) for the SBO-reference method (Chapter 3) and c) for the RL-based approach. Their layout follows Figures 5.18–5.19 and shows a contour plot of the objective function and its evolution during optimisation. Note that the RL-results are identical to Figures 5.22–5.23. They are just added for ease of comparability. Also note that each optimisation approach employs different numbers of optimisation iterations: While the GA is set to terminate after 300 simulations (15 generations with 20 individuals each), SBO and RL employ 30 (refinement) simulations. However, SBO requires a component-specific sampling (LHS), which accounts for additional 20 a-priori simulations. Being pre-trained on generic boxes, RL needs no further sampling and can directly start iterating for refinement. Although this pre-training involves considerable numerical efforts, the direct applicability to multiple geometries may outweigh pre-training expenses in the long term.

**Optimisation performance on  $\tilde{g}_1$ .** All plots in Figure 5.24 show a successful optimisation behaviour and approach the minimum of the objective function ( $q_{\text{rel}} = 0\%$ ). However, they do so in different manners: The GA in sub-image a) approaches it in a step-wise manner, while SBO and RL appear continuous. These steps of the GA form every 20 simulations since the GA evaluates generations of individuals. The graph shows the best found solution in each generation. The 19 remaining, non-dominant solutions per generation are omitted.

Note that – unlike SBO and RL – the graph of the GA is overall decreasing with some parts of constant  $q_{\text{rel}}$ , e.g.  $100 \leq i \leq 160$ . This behaviour

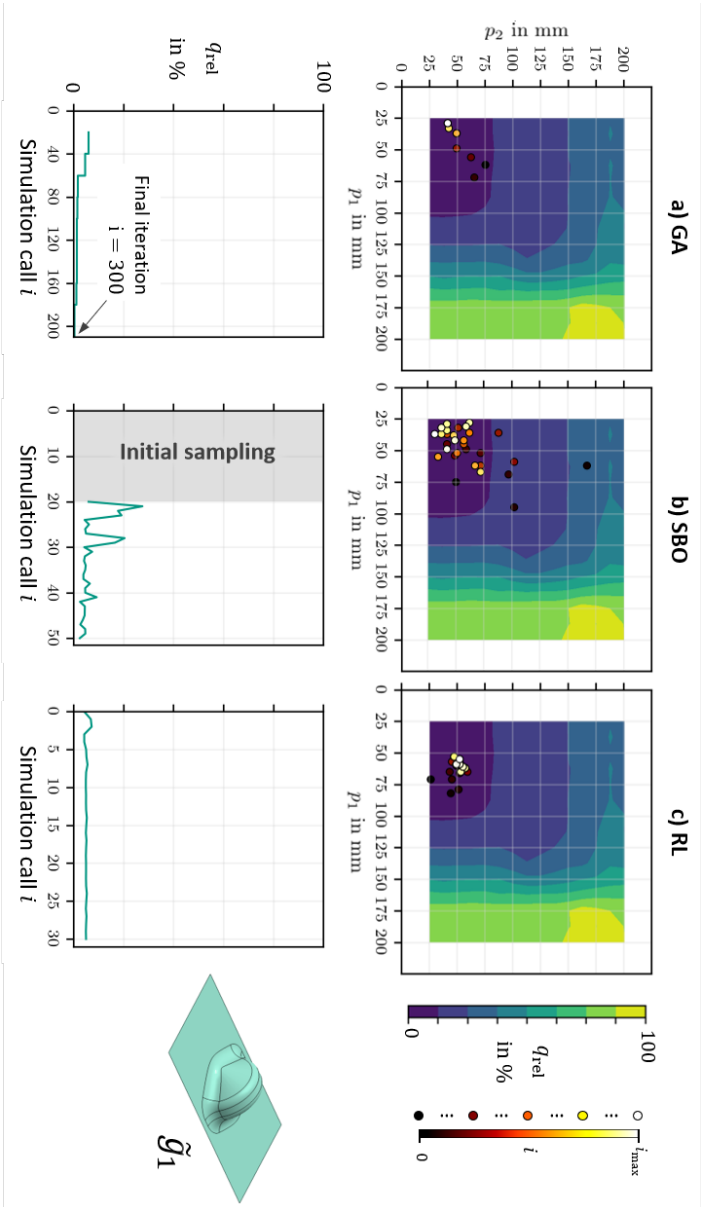


reflects the ‘elitist’-setting of the algorithm. That is, for prevention of a potential quality-loss, the best found solution from the previous generation is (unaltered) carried over to the next generation. If a generation does not offer an improved solution, the carry-over solution dominates the population. In this case,  $q_{\text{rel}}$  stays constant and the convergence plot shows just a plain horizontal line. Note that – for readability – in the contour plot only the best solution per generation is plotted, i.e. in total 15 markers for 15 generations. Also note that the carry-over solutions coincide and overlay each other in the plot, which makes only seven from fifteen markers visible.

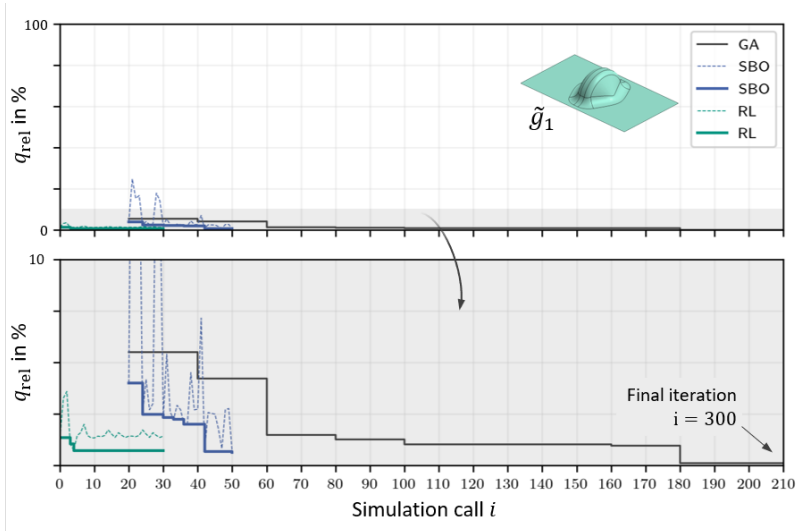
Although all optimisation approaches keep improving the solution over the iterations, the gain during additional optimisation iterations is limited as the first solutions are already near-optimal ( $q_{\text{rel}} \approx 1..5\%$ ). Thus, they offer little room for improvement with further iterations. This phenomenon can be explained by inspection of the contour plot and the markers: Since the optimum is not a sharp point but a comparably large plateau, the optimisers are likely to find and exploit it at the first attempt. Only SBO places few samples outside the optimal plateau, presumably due to false minima (local underestimations from missing samples). Interestingly, SBO additionally shows notable scatter around the plateau, while RL is able to concentrate its recommendations. Although this has little effect on the value of  $q_{\text{rel}}$ , this hints a more stable learning progress of the RL.

For a direct comparison, Figure 5.25 plots the evolution of  $q_{\text{rel}}$  for each optimiser in one diagram along with a detail view for closer inspection (grey shade). Besides the sheer sequence of  $q_{\text{rel}}$  (thin dashed line), the plots also show their lower envelope (bold solid line). Essentially, it gives for each iteration the ‘so-far-best’ solution and visualises, how fast each method reduces the objective function.

All graphs show a decline of the objective function, albeit with limited gain. The RL-approach finds its best solution in iteration  $i = 4$ , SBO in iteration  $i = 42$  followed by a marginal improvement in  $i = 50$ . The GA takes 180 simulations to find its best value. It also finds the overall-best



**Figure 5.24:** Convergence behaviour on  $g_1$  for a) a genetic algorithm (GA), b) SBO (Chapter 3) and c) the RL-approach. Markers represent the tried candidate solutions of each optimiser (top); the line plot visualises the evolution of the normalised objective function  $q_{rel}$  (bottom).



**Figure 5.25:** Juxtaposition of the objective function  $q_{\text{rel}}$  on  $\tilde{g}_5$  for the genetic algorithm (GA), SBO (Chapter 3) and the RL-approach. The plots of GA and SBO are offset and account for the number of simulations until a first iteration takes place.

value ( $q_{\text{rel}} = 0.2\%$ ), while both RL and SBO ( $q_{\text{rel}} \approx 0.8\%$ ) remain slightly inferior ( $q_{\text{rel}} \approx 0.8\%$ ). It may be noted however, that RL and SBO require far fewer iterations to reach their final value, i.e. converge faster.

In order to quantify the speed of convergence of optimisers, a measure is required. This usually involves measuring how many iterations are needed until a critical point with vanishing gradient is obtained. When a computation-intensive function and limited numerical resources are considered, the quality of the ultimate solution becomes less important and algorithms that early provide acceptable solutions become more relevant [271]. This shifts the focus to the initial behaviour and how long

an algorithm takes to find an acceptable solution. For this reason, the convergence metric  $\mathcal{C}$  is introduced:

$$\begin{aligned}\mathcal{C} &= \sum_{i=0}^{i_{\max}} (q_{\text{rel}_i} - q_{\text{rel}}^*) = \sum_{i=0}^{i_{\max}} (q_{\text{rel}_i} - 0) \\ &= \sum_{i=0}^{i_{\max}} q_{\text{rel}_i} \quad .\end{aligned}\tag{5.14}$$

Essentially, it measures the area under the envelope curve like an integral. Due to the integer-abscissa (iterations), the integral simplifies to a sum. A smaller value of  $\mathcal{C}$  implies better convergence. If an algorithm finds a good solution early on, the following summands and thus  $\mathcal{C}$  becomes smaller. This reflects the desired behaviour of an optimisation procedure of an expensive function, namely a rapid minimisation with as few function evaluations as possible. In order to apply Equation (5.14),  $i_{\max}$  needs to be defined. It is set such that all algorithms must have reached a certain quality  $q_{\text{rel}}^{\text{ref}}$ , i.e..  $q_{\text{rel}} \leq q_{\text{rel}}^{\text{ref}}$  must hold. For  $\tilde{g}_1$ ,  $q_{\text{rel}}^{\text{ref}} = 0.5\%$  is chosen. Table 5.2 summarises the values. For SBO two different scenarios are considered, one *with* consideration of the sampling effort (asterisked) and one *without*. The sampling effort itself is quantified by the product of sample size and the best quality observed during sampling.

For the remaining discussion, please note that all compared optimisation methods – GA, SBO and RL – are non-deterministic and involve elements of randomness. Thus, re-running them on the problem may lead to slightly different results of  $\mathcal{C}$ . Ideally, each method needs to run several times to allow for a stochastic analysis which was however computation wise impossible. Thus, the results may be interpreted as general trends rather than a quantitative performance discrimination to the last digit.

The results quantify what Figure 5.25 already graphically implied: The GA converges slowest, while SBO and RL range in the same order of

**Table 5.2:** Convergence comparison according to the convergence metric  $\mathcal{C}$  for  $\tilde{g}_1$ .

Optimiser	GA	SBO	SBO*	RL
$i_{\max}$	180	22	42	4
$\mathcal{C}$	4.39	0.79	1.83	0.31

magnitude when neglecting the sampling effort. When the sampling effort is considered, SBO positions itself between RL and GA. The true value presumably lies between the two values: Since sampling is done according to a sampling plan, it can be perfectly parallelised if sufficient computational resources are available, which in turn cuts the computation time.

**Optimisation performance on  $\tilde{g}_5$ .** In a similar fashion, Figure 5.26 shows plots for each optimiser on  $\tilde{g}_5$  (double-dome). Unlike before however, the problem structure proves much more complex: It is no longer convex but shows two connected, local maxima which divide the objective function into plateaus of approximately constant, mediocre quality ( $q_{\text{rel}} \approx 40\%$ ). These plateaus take up most of the parameter space. The optimum ( $q_{\text{rel}} = 0\%$ ) falls into a comparably small region and is surrounded by the maxima in a funnel-like shape. This poses two main challenges: First, the optimisers must not lock-in in one of these mediocre plateaus and, second, must successfully overcome the ridge between the maxima in order to descend into the funnel.

The overall behaviour of the algorithms stays the same as before: A successful optimisation can be observed. However, this time a notable effect of additional iterations is observed. Due to the majority of mediocre process responses (plateaus), all methods start with  $q_{\text{rel}} \approx 40 \dots 50\%$  in their first iteration, but then the progresses differ. The markers in sub-image a) show that in the first generation of the GA the best (elitist) individual lies in the plateau region. In the next generation however, the population has changed and the best individual is now located in the funnel. From

then on all markers gradually descend to the minimum, while a substantial improvement occurs in the fifth generation ( $80 < i \leq 100$ ) when  $q_{\text{rel}}$  drops from  $\approx 24\%$  to  $\approx 8\%$ .

The SBO approach in sub-image b) looks different. It shows significant initial scatter ( $20 \leq i \lesssim 30$ ) before eventually closing in on the optimum. The initial scatter presumably stem from elimination of the false minima (quasi exploration) and for  $i \gtrsim 30$  the optimiser reproducibly runs into the minimum (exploitation) with only minimal scatter. The RL-based approach has been discussed above, see Figure 5.22 and the neighbouring paragraphs for details.

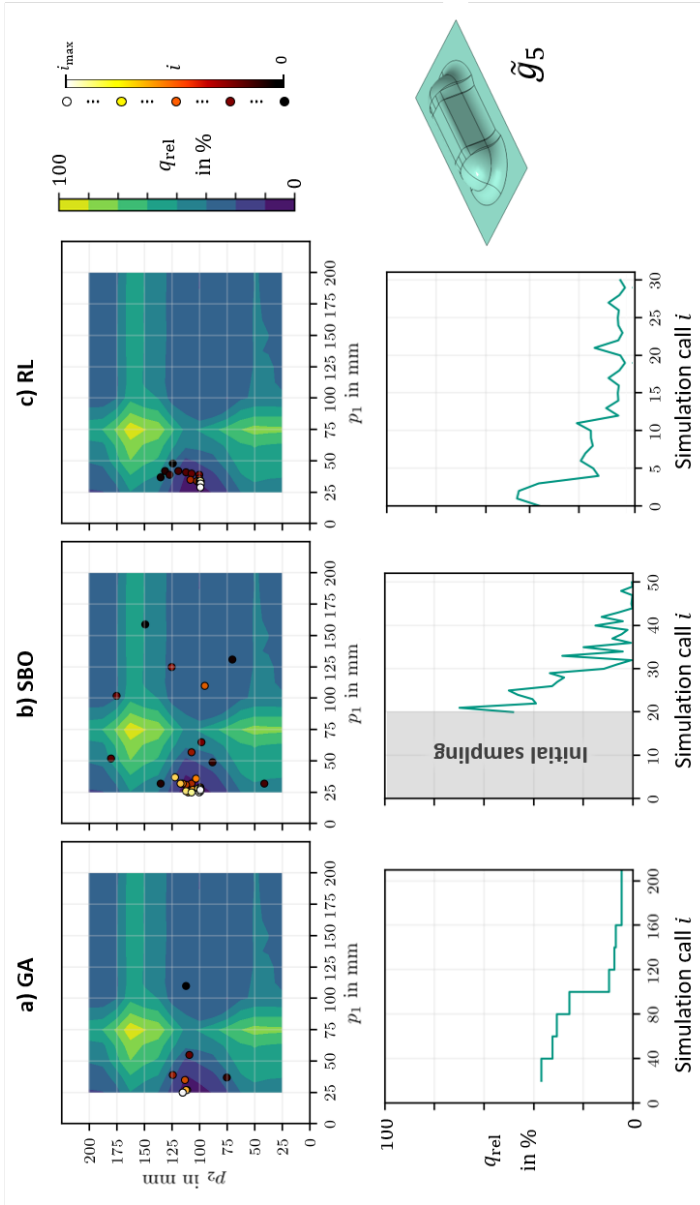
Ultimately, Figure 5.27 plots again the optimisation progress for each optimiser in a single plot for  $\tilde{g}_5$ . Overall, the plot resembles the previous: The graphs decline in a monotonous manner to the minimum ( $q_{\text{rel}} = 0\%$ ) and RL and SBO outperform the GA. However, this time all graphs show a substantial improvement of the objective function  $q_{\text{rel}}$  during optimisation.

All optimisers find a solution within a 5%-range around the optimum, while SBO approach finds the overall best-solution with ( $q_{\text{rel}} \approx 2\%$ ) within  $i = 32$  (with sampling) or  $i = 32 - 20 = 12$  iterations (without sampling), respectively. To determine the convergence factor  $\mathcal{C}$ , the algorithms must reach a minimal quality of  $q_{\text{rel}} = 5\%$ . The GA reaches this goal after 160 simulations, SBO after 32 and RL after 18 iterations. Table 5.3 summarises the resulting values for  $\mathcal{C}$ .

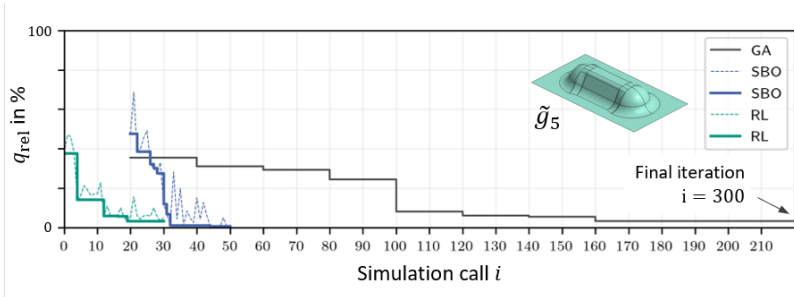
**Table 5.3:** Convergence comparison according to the convergence metric  $\mathcal{C}$  for  $\tilde{g}_5$ .

Optimiser	GA	SBO	SBO*	RL
$i_{\text{max}}$	160	12	32	19
$\mathcal{C}$	37.13	3.84	13.50	4.02

Again, Table 5.3 corroborates the findings of Figure 5.27. SBO and RL perform equally well, when neglecting the sampling effort. If sampling



**Figure 5.26:** Convergence behaviour on  $g_5$  for a) a genetic algorithm (GA), b) SBO (Chapter 3) and c) the RL-approach. Markers represent the tried candidate solutions of each optimiser (top); the line plot visualises the evolution of the normalised objective function  $q_{rel}$  (bottom).



**Figure 5.27:** Juxtaposition of the objective function  $q_{rel}$  on  $\tilde{g}_5$  for the genetic algorithm (GA), SBO (Chapter 3) and the RL-approach. The plots of GA and SBO are offset and account for the number of simulations until a first iteration takes place.

must be considered, e.g. due to limited computational resources, RL becomes superior and SBO positions itself between GA and RL.

The advantage of the pre-trained RL-model, becomes even more evident when optimising multiple geometries: Suppose, both geometries  $\tilde{g}_1$  and  $\tilde{g}_5$  need to be manufactured and their processes optimised. Optimising both geometries with SBO requires in total  $i_{max}^{\tilde{g}_1} + i_{max}^{\tilde{g}_5} = 42 + 32 = 74$  simulations, whereof 20 sampling simulations are necessary for each geometry, i.e. 40 simulations in total for the two geometries considered here. In contrast, the RL-model can do without component-specific sampling due to its pre-training on boxes and takes only  $i_{max}^{\tilde{g}_1} + i_{max}^{\tilde{g}_5} = 4 + 19 = 23$  simulations. Although the exact numbers will certainly vary in different applications, the non-necessity of the a-priori sampling substantially cuts the computational effort. Note however, that this deployment advantage comes at the cost of substantial pre-training effort.

## 5.4 Intermediate summary

This chapter presents an approach for estimation of optimal process parameters for textile forming of variable component geometries. It extends the grid-based approach for formability estimation and accepts



the strain field estimation (image) as input and yields an estimation of optimal parameters. Unlike the formability assessment, however, the algorithm is trained by Reinforcement Learning (RL) instead of supervised learning. During training, the algorithm interacts with a responsive forming-simulation and a set of training geometries. More specifically, for each geometry it infers a process parameter recommendation and starts an according forming simulation. If the recommendation improves the forming result, the algorithm is rewarded and will give similar recommendations in the future. Otherwise it is penalised. The process iterates for a predefined number of epochs.

Training of the algorithm is done using actor-critic-techniques and comprises two networks: The desired actor-network  $\mu_p$  and an auxiliary critic-network. During the training, the actor-network  $\mu_p$  gives process recommendations and the critic-network in turn informs the actor how to improve.

The algorithm is applied to an exemplary forming process of doubly-symmetric box-geometries and can be manipulated by freely positionable pressure pads. The part quality is measured by the local fabric curvature as a measure of wrinkling tendencies. At first, an extensive hyperparameter study is performed to maximise training efficiency. It revealed two aspects: First, data augmentation can be used to improve training performance. Second, a finite number of training geometries introduces sufficient information to  $\mu_p$  to give useful parameter estimations for geometries inside and – to a certain degree – outside the class of training geometries.

After a (pre-)training on box-geometries,  $\mu_p$  is deployed on five non-box geometries. Its recommendations are found to follow the geometry changes in a plausible way. Two limit cases are selected for further assessment of their optimality. To this end,  $\mu_p$ 's recommendations are compared to a full-factorial sampling. The results show that they are indeed near-optimal. Component-specific refinement iterations are additionally applied to converge  $\mu_p$ 's recommendations to the true optimum like in regular SBO.

Ultimately, this chapter concludes with a performance comparison of the newly developed RL approach, a classical genetic algorithm (GA) and surrogate-based optimisation (SBO). Again the previous two geometries are considered. The results hint that RL and SBO outperform the GA and show a similar convergence during refinement iterations. However, SBO needs a component-specific a-priori sampling while the RL-based method is pre-trained and can directly start refining. Thus,  $\mu_p$  speeds up the optimisation process similar to a classical surrogate but beneficially cuts the component-specific sampling effort. However, this comes at the cost of substantial numerical efforts for algorithm pre-training.

In summary, the overall results of this chapter show that it is possible to extract usable part-process-relations from generic samples. The obtained relations can then be applied to a new geometry and give a meaningful initial recommendation. The results further hint that  $\mu_p$  converges to the true optimum at roughly the same speed as a regular surrogate.

# 6 Summary and outlook

## 6.1 Summary

Modern manufacturing processes generally require a careful adjustment of process parameters to achieve maximum part quality. To cut the cost of experimental production trials, physics-based process simulations are increasingly applied as they allow for detailed analyses of the process dynamics at early development stages before procurement of equipment and material. Owing to their digital nature, they can beneficially be coupled with optimisation algorithms for performance maximisation. However, despite considerable process improvements being observed, the computational load for algorithmic optimisation still poses a substantial barrier in practice.

An option to reduce the computational load during process optimisation is surrogate-based optimisation (SBO). SBO employs time-efficient, phenomenological models (surrogates) to guide the optimisation procedure and concentrates costly simulations on the most promising regions. While this generally accelerates the optimisation, classical SBO still proves inflexible and cumbersome when the optimisation task varies, e.g. due to recurring design adaptations for customisation. Recent advances in Machine Learning (ML) – especially neural networks (NN) – show promise to enhance SBO-capabilities towards flexible optimisation tasks. This is mainly for two reasons: First, they are able to handle and learn from large amounts of high-dimensional data and, second, they can be tailored so as to take advantage from certain, potentially more informative data structures, e.g. images.

This work explores the potential of ML-techniques as a prospective tool for efficient process engineering and focusses on three main aspects: 1) Their applicability as classical surrogate for SBO (Chapter 3), 2) the potential of structured input data (images) for manufacturability assessment of variable geometries (Chapter 4) and 3) their suitability as an advisory tool for process optimisation of variable components (Chapter 5). These aspects apply in principle to any manufacturing process but are exemplarily studied at the example of textile forming.

**Applicability as classical surrogates.** Chapter 3 examines the use of deep neural networks as surrogate models for virtual process optimisation. The considered use-case is gripper-assisted fabric forming. Different NN-types are compared regarding their predictive quality. The best prediction performance is achieved using a deep NN which predicts the full strain field instead of just a single performance scalar. This hints that the full-field resolution facilitates learning, especially when multiple critical regions impact on the overall part quality: Compressing this information into a single performance scalar removes informativeness. A spatial resolution is found to help allocate the quality gains or losses to specific regions, retains sensitivity information and overall facilitates learning.

The network is further integrated into an SBO-framework to study the suitability and convergence behaviour during optimisation. Four SBO-configurations with different exploration-exploitation emphasis are investigated. In each case, the developed SBO-framework successfully reduces the objective function, while best performance is observed for an exploitation-emphasised variant. Also, each SBO-variant outperformed a state-of-the-art genetic algorithm (GA) in terms of efficiency. Overall, the results show that deep neural networks are a viable option for SBO. The developed NN-based SBO-framework is later on used as a reference for optimisation of variable geometries.

**Surrogate-based formability assessment.** Chapter 4 develops and assesses options for surrogate-assisted formability assessment of variable geometries. Two overarching approaches are compared: a *feature*-based and a *grid*-based approach.

The *feature*-based approach scans a component for (potentially critical) features from a predefined geometry-scheme. It then assigns each recognised feature a set of geometry parameters and a pre-trained surrogate model then evaluates them for the expected forming quality, e.g. maximum shear strain. The approach is exemplarily applied to a complex box-geometry with rounded corners and is found to well-reproduce the results of physics-based simulations while being numerically inexpensive. Thus, the surrogate may be used to explore many geometry alternatives in an intuitive manner without need for laborious and computation-intensive FE-simulations.

In contrast, the *grid*-based discards distinct geometry parameters but encodes the geometry – and also the forming result (strain field) – in an image. A convolutional neural network (CNN), a sub-type of NN for image-processing, then learns the part-process-relation from supplied simulation samples. As the geometry description is not confined to a preselected set of geometry parameters, the grid-approach proves more versatile and can principally process any undercut-free geometry. The results show that the trained CNN gives sound estimations not only within the class of training geometries but also for new geometries outside this class. However, at the same time the importance of an appropriate training database becomes apparent: It can only make valid predictions for geometry characteristics covered during training. If process-relevant aspects, e.g. concave indentations, are not covered during training, the algorithm remains unaware of their process effects.

**Process optimisation for variable geometries.** Chapter 5 uses reinforcement learning (RL) to train an NN that estimates optimal process parameters for a given geometry. The algorithm builds on the grid-based

CNN presented in Chapter 4: It accepts a strain field estimation as input (image) and yields a set of process parameters as output. During training, the algorithm is repeatedly presented (generic) geometries and issues for each geometry a parameter recommendation. If they prove useful in according process simulations, the algorithm is rewarded and penalised otherwise. It thereby learns which geometry features require which parameters. The approach is again applied to an example from fabric forming and optimises the position of restraining forces from pressure pads (blank holders).

Although being trained on generic geometries only, the algorithm is found to give useful recommendations also for non-generic geometries. This implies that the proposed RL-approach is capable of extracting process 'knowledge' from a finite set of process samples and can apply it in a new situation. Further investigations show that its initial (near-optimal) recommendations do converge to the true process optimum upon component-specific continuation of training. In the studied examples, the speed of convergence is approximately equal to a classical SBO. Unlike classical SBO however, the RL-algorithm needs no a-priori sampling for new geometries due to its pre-training on generic geometries. Thus, once (pre-)trained, it is more efficient, although it must be noted that the computational effort for the initial pre-training may be considerable. However, the benefits of an immediate process recommendation and rapid convergence to the optimum may outweigh initial expenses in the long term.

## 6.2 Answers to research hypotheses

The results can be used to answer the initial research hypotheses formulated in Chapter 1:

**Hypothesis 1.** It is possible with ML-methods to extract process knowledge from generic part-process-samples and apply it to a new geometry.

**Answer: Validated.**

The results show that a finite set of generic part-process-samples contains – within certain limits – sufficient information for generalisation to a new, non-generic use-case. With growing complexity of part and/or process, this number may however become significantly large. The results further show that (grid-)structured input variables like images provide a substantially larger expressiveness for a geometry encoding than unstructured input variables like (scalar) geometry parameters. The inevitable increase in data dimensionality can be compensated for by recent ML-techniques, e.g. convolutional neural networks.

**Hypothesis 2.** Once, trained, such an ML-model speeds up the optimisation similar to a classical surrogate.

**Answer: Validated.**

In the studied examples, the classical surrogate-based optimisation (SBO) and the developed RL-based method both show roughly the same convergence speed during optimisation. However, when taking the a-priori sampling effort into account, the RL-method gradually takes the lead: Being pre-trained on generic geometries it does not need an a-priori sampling and can directly start optimising. In contrast, SBO requires an initial sampling for each new geometry. This advantage comes however at the expense of (significantly) increased pre-training efforts.

## 6.3 Outlook

Further research is still envisaged and can be divided into three main areas of research, namely

- 1) application and evaluation in real-world scenarios,
- 2) inclusion of known physics into the surrogates,
- 3) usage of more versatile ML-architectures

and will be outlined one by one in the following.

**Application.** The results obtained in this work exemplify the use of ML-techniques in the context of manufacturing process optimisation at a simplified example from textile forming. Having outlined their principal potential, the techniques need yet to be advanced to more complex and application-centred use-cases for use in industrial practice. Regarding textile-forming, this means the integration of more complex process scenarios, i.e. additional process parameters and variable material properties or orientations. The results have also shown that the generalisation capabilities depend substantially on the underlying training catalogue of geometry characteristics, e.g. convex/concave, (a-)symmetry or non-, singly- or doubly-curved. However, at the moment it is unclear how to define a comprehensive sampling plan for them.

Besides additional process parameters also their variation over time may be addressed. This work assumes constant process parameters during tool stroke. However, some process technologies may require parameter variations over time. For instance, consider in-situ adaptations of the tool closing speed to avoid defects from excessive or uneven cavity pressures. In such cases, the process passes through a whole sequence of states (pressure distributions) and according actions (increase/decrease speed). Such adaptive processes can also be solved by RL and may be addressed in the future.



In the wider sense, the developed methods can also be transferred to other processes inside and outside composite manufacturing. The experiences from this work hints processes with recurrent characteristics most promising. For instance, consider additive manufacturing by fused filament fabrication: The distribution of surrounding material, its temperature field and heat transfer conditions will certainly affect the optimal deposition parameters for e.g. optimal filament adhesion. Encoding them in a grid-structure around the deposition point may enable an ML-algorithm to learn optimal deposition parameters given the local process situation. Since the governing process dynamics (melting, deposition, cooling) are mainly local effects, algorithm training may be easier and it would presumably be applicable to any printed geometry.

Ultimately, the results also show that (pre-)training and deployment can be decoupled to a certain extent. That is, the computation- and know-how-intensive (pre-)training could be centralised, while a potential user 'only' evaluates the algorithm at comparably low numerical cost – a possible business case in the long term for software manufacturers. Leaving aside issues of data security, such a centralised system could also use user requests for algorithm refinement similar to internet search engines refine their results based on user interaction.

**Inclusion of known physics.** Data-driven methods learn the dynamics of a system from (sufficiently many) observations. However, especially in engineering applications, comprehensive knowledge on the governing physics is at hand, such as conservation of energy, momentum and mass. Integrating this knowledge into the training progress, this could reduce the required amount of data, speed up the training and make ML-predictions more physically consistent.

So called physics-informed neural networks (PINNs) pursue exactly this concept. First introduced 2019 by RAISSI et al. [272], they do not just minimise the error between ML-predictions and observations but between ML-predictions and an underlying constitutive equation . In extreme,

PINNs can do without observations and just minimise the error with respect to a constitutive equation. They thereby essentially become a solver for PDEs. Unlike classical numerical methods however, they can cope with variable boundary conditions: If appropriately trained on a multitude of boundary conditions, they can be evaluated for a new situation and will provide a physically-consistent solution. From a user perspective, it feels like an analytical solution.

First applications to composite processing - specifically curing – show comparable accuracy compared to classical numerical methods [273]. However, for lack of efficient preprocessing tools, applying PINNs currently involves substantial amounts of manual work and restricts them to comparably simple scenarios. Especially, the appropriate configuration of the loss function such that it correctly accounts for the PDE, initial and boundary conditions and the solution domain is currently a significant barrier. Once these issues are resolved, PINNs provide substantial promise for process engineering.

**Usage of more versatile ML-architectures.** Although the presented CNNs are a new ML-tool in engineering, they are not the latest state of the art in ML-research. The CNNs in this work use an equidistant, rectangular grid-domain (image) as input. While this is sufficient for this work, it appears somewhat restrictive in the long term. So called graph neural networks may offer a more versatile alternative.

In general, a graph is defined as a set of nodes and connecting edges. Both, nodes and edges can have (engineering) attributes, e.g. material properties, current temperature or stress/strain state. Overall, such a graph can be thought of as an information-enriched and more flexible grid compared to an image. Dedicated graph neural networks (GNN) can then learn information that resides in the nodes and their connectivity (edges).

In a recent publication SANCHEZ-GONZALEZ et al. [274] present particle-based fluid simulation based on GNNs that reproduces remarkably well

simulation results in real-time. First adaptations to process engineering (injection moulding) have been done [275]<sup>1</sup>. Although the techniques are still in their infancy, the approach's principal versatility presages great potential for generalisation to arbitrary geometries – especially when combined with PINNs.

---

<sup>1</sup>Bachelor Thesis at the Institute for Program Structures and Data Organization (KIT-IPD) at Department of Informatics in cooperation with Nils Meyer and Louis Schreyer from KIT-FAST.



# A Appendix

## A.1 Definition of the CNN error metrics

The accuracy and correlation metrics for the *grid*-based estimation of a shear strain field (cf. graphs in Figure 4.27) are defined as follows:

### Accuracy metrics

$$\begin{aligned}\varepsilon_{\text{val}} &= \text{MSE}(\hat{\gamma}_{12} - \gamma_{12}) = \text{mean}(\hat{\gamma}_{12j} - \gamma_{12j})^2 \\ \Delta\gamma_{12}^{\text{max}} &= |\hat{\gamma}_{12}^{\text{max}} - \gamma_{12}^{\text{max}}| = |\max(\hat{\gamma}_{12}) - \max(\gamma_{12})| \\ \Delta_{\text{max}}(\gamma_{12}) &= \max(|\hat{\gamma}_{12} - \gamma_{12}|)\end{aligned}$$

### Correlation metrics [263, p 585]

$$c_d = 1 - \frac{\sum_j (\gamma_{12j} - \hat{\gamma}_{12j})^2}{\sum_j (\gamma_{12j} - \text{mean}(\gamma_{12j}))} \quad (\text{A.1})$$

$$r_p = \frac{\sum_j [(\hat{\gamma}_{12j} - \text{mean}(\hat{\gamma}_{12j}))(\gamma_{12j} - \text{mean}(\gamma_{12j}))]}{\sqrt{\sum_j [(\hat{\gamma}_{12j} - \text{mean}(\hat{\gamma}_{12j}))^2]} \sqrt{\sum_j [(\gamma_{12j} - \text{mean}(\gamma_{12j}))^2]}} \quad (\text{A.2})$$

Therein,  $j$  denotes the  $j$ -th *grid*-pixel in the strain field (greyscale image). A hat superscript ( $\hat{\gamma}$ ) denotes an estimation by the CNN  $\mu_f$ ; otherwise a simulation result is implied (ground truth). In Figure 4.27 a) and b), the metrics are separately computed for  $n_V$  validation geometries their average and 95 %-percentiles are plotted.



# List of Figures

2.1	Influences on manufacturability and lightweight potential . . .	6
2.2	Common fabric weave patterns . . . . .	8
2.3	Open and closed shell structures . . . . .	9
2.4	Automotive example of component design revisions . . . . .	10
2.5	Resin Transfer Moulding (RTM) process principle . . . . .	11
2.6	Deep drawing process principle . . . . .	12
2.7	Visualisation of textile scales . . . . .	14
2.8	Main deformation modes of textiles . . . . .	15
2.9	Fabric shear deformation . . . . .	16
2.10	Example of textile wrinkling . . . . .	18
2.11	Wrinkling type I and II . . . . .	19
2.12	Examples of textile forming manipulation . . . . .	22
2.13	Visualisation of kinematic draping . . . . .	24
2.14	Kinematic, mechanical and experimental draping . . . . .	25
2.15	Model scales for textiles . . . . .	26
2.16	Virtual process chain (principle) . . . . .	29
2.17	Local and global optima (visualisation) . . . . .	32
2.18	Function approximation by a surrogate . . . . .	36
2.19	Random vs. Latin Hypercube Sampling . . . . .	38
2.20	Direct vs. surrogate-based optimisation . . . . .	40
2.21	Unstructured vs. structured variables . . . . .	45
2.22	(Un-)structured boundary conditions . . . . .	45
2.23	Fully connected neural network (principle) . . . . .	48
2.24	Example application of convolutional neural networks . . . . .	49
2.25	Surrogate overfitting (illustration) . . . . .	50
2.26	Neural network training progress (principle) . . . . .	51
3.1	Surrogate-based optimisation workflow in this work . . . . .	58

3.2	Principle of a genetic algorithm . . . . .	59
3.3	Minimal example of surrogate-based optimisation . . . . .	63
3.4	Shape optimisation with sudden stress relocations . . . . .	64
3.5	Parameter sensitivity change during shape optimisation . . . . .	65
3.6	Visualisation of scalar and field ANNs . . . . .	67
3.7	Clamping frame (CAD-drawing) . . . . .	68
3.8	Double dome geometry (CAD drawing) . . . . .	69
3.9	Forming simulation setup with clamping frame . . . . .	69
3.10	Forming simulation setup with clamping frame (detail) . . . . .	70
3.11	Three ANN-architectures for surrogate modelling . . . . .	72
3.12	Accuracy comparison of three ANN surrogates . . . . .	73
3.13	Optimisation progress for different SBO-configurations . . . . .	78
3.14	Convergence comparison for each SBO-configuration . . . . .	80
3.15	Optimisation result comparison for the reference method . . . . .	81
4.1	Workflow for the feature-based formability assessment . . . . .	85
4.2	Parametric corner geometry for this work . . . . .	86
4.3	Evaluation of feature formability (principle) . . . . .	87
4.4	Geometry recognition for feature-based formability estimation . . . . .	89
4.5	(Un-)Structured geometry representation in voxels . . . . .	92
4.6	Image-based geometry and forming result encoding . . . . .	93
4.7	Workflow of the grid-based formability assessment . . . . .	94
4.8	Convolution operation (principle) . . . . .	96
4.9	Max-pooling operation (principle) . . . . .	97
4.10	Upsampling by nearest neighbourhood extrapolation (principle) . . . . .	98
4.11	Autoencoder principle . . . . .	99
4.12	Autoencoder-CNN used in this work (schematic visualisation) . . . . .	100
4.13	Contour line definition for geometry samples . . . . .	101
4.14	Example bodies of revolution for CNN-training . . . . .	101
4.15	Study setup to validate SAINT-VENANT's principle . . . . .	104
4.16	Evolution of the shear angle with box-size . . . . .	104
4.17	Extraction of resulting fabric internal loads during forming . . . . .	105
4.18	Line-loads in the fabric during forming for different-sized boxes . . . . .	106
4.19	Criterion for strain field independence . . . . .	107
4.20	Demonstrator geometry for the feature-based approach . . . . .	108
4.21	Validation of the feature-based formability approach . . . . .	110



4.22	Corner design map for efficient formability assessment . . . .	111
4.23	Validation of the surrogate-based component redesign . . . .	112
4.24	Plot of the shear angles before and after component redesign .	112
4.25	CNN-architecture for grid-based formability assessment . . .	115
4.26	CNN training progress for two databases . . . . .	116
4.27	CNN-accuracy inside the class of training data . . . . .	118
4.28	CNN-accuracy outside the class of training data . . . . .	119
4.29	Comparison of simulation and CNN-estimation . . . . .	121
4.30	Short caption . . . . .	123
5.1	Algorithm architecture for process parameter estimation . . .	129
5.2	Architecture of the CNN $\mu_p$ . . . . .	130
5.3	Training and application scheme of $\mu$ . . . . .	130
5.4	Actor-Critic Training scheme . . . . .	133
5.5	Evolution of the exploration probability . . . . .	135
5.6	Pressure-pad assisted fabric forming . . . . .	137
5.7	Sketch of the employed forming simulation model . . . . .	138
5.8	Curvature histogram with quantile (quality metric) . . . . .	139
5.9	Example objective function variable pressure pads . . . . .	141
5.10	Comparison of simulation and substitute function . . . . .	142
5.11	Evolution of the critic loss . . . . .	144
5.12	Evolution of the process response (substitute function) . . . .	145
5.13	Effect of different learning rates . . . . .	148
5.14	Visualisation of the data augmentation by mirroring . . . . .	149
5.15	Effect of data augmentation . . . . .	150
5.16	Effect of different numbers of training geometries . . . . .	152
5.17	Effect of different numbers of training geometries (bar plot) .	152
5.18	Training behaviour of $\mu_p$ - training geometries . . . . .	154
5.19	Training behaviour of $\mu_p$ - validation geometries . . . . .	156
5.20	Five test geometries outside the range of box-geometries . . .	158
5.21	Process recommendations after training on boxes . . . . .	159
5.22	Component-specific training . . . . .	162
5.23	Component-specific training – progress of the objective function	163
5.24	Comparison of GA, SBO and RL on geometry $\tilde{g}_1$ . . . . .	166
5.25	Convergence comparison of GA, SBO and RL on $\tilde{g}_5$ . . . . .	167
5.26	Comparison of GA, SBO and RL on geometry $\tilde{g}_5$ . . . . .	171

5.27 Convergence comparison of GA, SBO and RL on  $\tilde{g}_5$  . . . . . 172

# List of Tables

3.1	Employed configuration of the GA . . . . .	60
3.2	Network architecture for the ANN-surrogates . . . . .	72
4.1	Corner parameter explanation, range and examples . . . . .	87
4.2	Parameters for the box-demonstrator . . . . .	109
5.1	Normalisation parameters of the objective function . . . . .	160
5.2	Convergence comparison for $\tilde{g}_1$ . . . . .	169
5.3	Convergence comparison for $\tilde{g}_5$ . . . . .	170



# Bibliography

- [1] *Zitate: Sprichwörter und Redensarten von A-Z*. Otus Verlag, St. Gallen/Schweiz, 2003.
- [2] Nima Shamsaei, Aref Yadollahi, Linkan Bian, and Scott M. Thompson. An overview of direct laser deposition for additive manufacturing - part 2. *Additive Manufacturing*, 8:p 12–35, 2015.
- [3] Frank Henning, Luise Kärger, Dominik Dörr, Fabian J. Schirmaier, Julian Seuffert, and Alexander Bernath. Fast processing and continuous simulation of automotive structural composite components. *Composites Science and Technology*, 171:p 261–279, 2019.
- [4] Frank Henning and Elvira Moeller, editors. *Handbuch Leichtbau: Methoden, Werkstoffe, Fertigung*. Hanser Verlag, München, 2011.
- [5] ISO Technical Report 581. *Weldability of metallic materials - General principles*, 2005.
- [6] Eberhard Roos and Karl Maile. *Werkstoffkunde für Ingenieure*. Springer Verlag, Berlin/Heidelberg, 2015.
- [7] Chokri Cherif. *Textile Werkstoffe für den Leichtbau*. Springer Verlag, Berlin/Heidelberg, 2011.
- [8] Fibermax Composites Ltd. Weaving styles, [www.fibermax.eu/index\\_files/weavings\\_tylesandpatterns.html](http://www.fibermax.eu/index_files/weavings_tylesandpatterns.html), 2007, last accessed: 31.08.2022.
- [9] Werner Hufenbach and Olaf Helms. *Leitfaden zum methodischen Konstruieren mit Faser-Kunststoff-Verbunden*. Leichtbau-Zentrum-Sachsen GmbH (Eigenverlag), Dresden, 2010.
- [10] Kevin Potter. *An introduction to composite products: Design, development and manufacture*. Chapman & Hall, London/United Kingdom, 1 edition, 1997.
- [11] Adrian Padeanu. VW Golf 8: Erste offizielle Skizzen zeigen den neuen Golf, [www.motorsport-total.com/auto/news/vw-golf-8-2020-erste-offizielle-skizzen-zeigen-den-neuen-golf-19101301](http://www.motorsport-total.com/auto/news/vw-golf-8-2020-erste-offizielle-skizzen-zeigen-den-neuen-golf-19101301), 2019, last accessed: 31.08.2022.
- [12] ATP Autoteile GmbH. Kotflügel und Radhaus für VW Golf, [www.atp-auto-teile.de/de/search/ma-9992/n-324/vw-kotfluegel-radhaus](http://www.atp-auto-teile.de/de/search/ma-9992/n-324/vw-kotfluegel-radhaus), 2022, last accessed: 31.08.2022.

- [13] Luise Kärger, Siegfried Galkin, Clemens Zimmerling, Dominik Dörr, Johannes Linden, André Oeckerath, and Klaus Wolf. Forming optimisation embedded in a cae chain to assess and enhance the structural performance of composite components. *Composite Structures*, 192:p 143–152, 2018.
- [14] DIN 8582:2003-09. *Fertigungsverfahren Umformen - Einordnung, Unterteilung, Begriffe, Alphabetische Übersicht*, 2003.
- [15] DIN 8584-1:2003-09. *Fertigungsverfahren Zugdruckumformen*, 2003.
- [16] Eckart Doege and Bernd-Arno Behrens. *Handbuch Umformtechnik*. Springer Verlag, Berlin/Heidelberg, 2010.
- [17] DIN 8580:2003-09. *Fertigungsverfahren*, 2003.
- [18] Werner Schatt, editor. *Werkstoffwissenschaft*. Wiley-VCH, Weinheim, 9 edition, 2003.
- [19] Farbod Nosrat Nezami. *Automatisiertes Preforming von Kohlefaserhalbzeugen mit aktiven Materialführungssystemen zur Herstellung komplexer Faserverbundstrukturen*. Dissertation, TU Dresden, Dresden, 2015.
- [20] Philippe Bussetta and Nuno Correia. Numerical forming of continuous fibre reinforced composite material: A review. *Composites Part A*, 113:p 12–31, 2018.
- [21] Jeff Sloan. Materials and processes: Composites fibers and resins, [www.compositesworld.com/articles/composites-101-fibers-and-resins](http://www.compositesworld.com/articles/composites-101-fibers-and-resins), 2016, last accessed: 31.08.2022.
- [22] S. E. Stapleton, L. Appel, J.-W. Simon, and S. Reese. Representative volume element for parallel fiber bundles: Model and size convergence. *Composites Part A*, 87:p 170–185, 2016.
- [23] Michel Tournalonias, Marie-Ange Bueno, and Dominique Poquillon. Friction of carbon tows and fine single fibres. *Composites Part A*, 98:116–123, 2017.
- [24] Rym Azzouz, Samir Allaoui, and Raphael Moulart. Composite preforming defects: a review and a classification. *International Journal of Material Forming*, 14(6):p 1259–1278, 2021.
- [25] Siegfried Galkin, Eckart Kunze, Luise Kärger, Robert Böhm, and Maik Gude. Experimental and numerical determination of the local fiber volume content of unidirectional non-crimp fabrics with forming effects. *Journal of Composites Science*, 3(1):Art. 19, 2019.
- [26] Eckart Kunze, Siegfried Galkin, Robert Böhm, Maik Gude, and Luise Kärger. The impact of draping effects on the stiffness and failure behavior of unidirectional non-crimp fabric fiber reinforced composites. *Materials*, 13(13):Art. 2959, 2020.

- 
- [27] H. M. Hsiao and I. M. Daniel. Elastic properties of composites with fiber waviness. *Composites Part A*, 27(10):p 931–941, 1996.
- [28] Sebastiaan Haanappel. *Forming of UD fibre reinforced thermoplastics: A critical evaluation of intra-ply shear*. Dissertation, Universiteit Twente, Twente, 2013.
- [29] Dominik Dörr. *Simulation of the thermoforming process of UD fiber-reinforced thermoplastic tape laminates*. Dissertation, Karlsruher Institut für Technologie, Karlsruhe, 2021.
- [30] Christian Poppe, Fabian Albrecht, Constantin Krauß, and Luise Kärger. A 3d modelling approach for fluid progression during process simulation of wet compression moulding – motivation & approach. *Procedia Manufacturing*, 47:p 85–92, 2020.
- [31] H. Fukuda and Yakushiji, M., Wada, A. A loop test to measure the strength of monofilaments used for advanced composites. In Golam M. Newaz and Ronald F. Gibson, editors, *Proceedings of the Eighth Japan-U.S. Conference on Composite Materials*, pages 886–893. CRC Press, Boca Raton/USA, 2019.
- [32] Farbod Nosrat-Nezami, Thomas Gereke, Christian Eberdt, and Chokri Cherif. Characterisation of the shear–tension coupling of carbon-fibre fabric under controlled membrane tensions for precise simulative predictions of industrial preforming processes. *Composites Part A*, 67:p 131–139, 2014.
- [33] P. Boisse, J. Colmars, N. Hamila, N. Naouar, and Q. Steer. Bending and wrinkling of composite fiber preforms and prepregs. a review and new developments in the draping simulations. *Composites Part B*, 141:p 234–249, 2018.
- [34] Hao Shen, Peng Wang, Xavier Legrand, and Lingshan Liu. Characterisation and optimisation of wrinkling during the forming of tufted three-dimensional composite preforms. *Composites Part A*, 127:Art. 105651, 2019.
- [35] Johan Viisainen. *Wrinkling behaviour of biaxial non-crimp fabrics during preforming*. Dissertation, University of Cambridge, Cambridge/UK, 2022.
- [36] Philippe Boisse, Jin Huang, and Eduardo Guzman-Maldonado. Analysis and modeling of wrinkling in composite forming. *Journal of Composites Science*, 5(3):Art. 81, 2021.
- [37] P. Boisse, N. Hamila, and A. Madeo. The difficulties in modeling the mechanical behavior of textile composite reinforcements with standard continuum mechanics of cauchy - some possible remedies. *International Journal of Solids and Structures*, 154:p 55–65, 2018.
- [38] Christian Poppe, Tobias Rosenkranz, Dominik Dörr, and Luise Kärger. Comparative experimental and numerical analysis of bending behaviour of dry and low viscous infiltrated woven fabrics. *Composites Part A*, 124:Art. 105466, 2019.

- [39] P. Boisse, N. Hamila, E. Vidal-Sallé, and F. Dumont. Simulation of wrinkling during textile composite reinforcement forming - influence of tensile, in-plane shear and bending stiffnesses. *Composites Science and Technology*, 71(5):p 683–692, 2011.
- [40] P. Boisse, N. Hamila, and A. Madeo. Analysis of defect developments in composite forming. In Peter W. R. Beaumont, Constantinos Soutis, and Alma Hodzic, editors, *The Structural Integrity of Carbon Fiber Composites*, pages 319–337. Springer International Publishing, Cham/Switzerland, 2017.
- [41] P. Hallander, M. Akermo, C. Mattei, M. Petersson, and T. Nyman. An experimental study of mechanisms behind wrinkle development during forming of composite laminates. *Composites Part A*, 50:p 54–64, 2013.
- [42] A. Endruweit and P. Ermanni. The in-plane permeability of sheared textiles. experimental observations and a predictive conversion model. *Composites Part A*, 35(4):p 439–451, 2004.
- [43] Fritz Klocke. *Fertigungsverfahren 4*. Springer Verlag, Berlin/Heidelberg, 2017.
- [44] Lucien A. Schmit and Robert H. Mallett. Structural synthesis and design parameter hierarchy. *Journal of the Structural Division*, 89(4):p 269–299, 1963.
- [45] A. C. Long, B. J. Souter, F. Robitaille, and C. D. Rudd. Effects of fibre architecture on deformation during preform manufacture. *Advanced Composites Letters*, 8(6):Art. 096369359900800, 1999.
- [46] Daniel B. Miracle. Fabrics and preforms. In Daniel B. Miracle and Steven L. Donaldson, editors, *Composites*, pages 59–68. ASM International, Novelt/USA, 2001.
- [47] M. Aono, P. Denti, D. E. Breen, and M. J. Wozny. Fitting a woven cloth model to a curved surface: dart insertion. *IEEE Computer Graphics and Applications*, 16(5):p 60–70, 1996.
- [48] Charlotte M. Eisenhauer and Klaus Drechsler. Integration of excess material into a semi-finished product to form complex composite parts. *Textile Research Journal*, 87(19):p 2420–2431, 2017.
- [49] P. Molnár, A. Ogale, R. Lahr, and P. Mitschang. Influence of drapability by using stitching technology to reduce fabric deformation and shear during thermoforming. *Composites Science and Technology*, 67(15-16):p 3386–3393, 2007.
- [50] Mark A. Turk, Bruno Vermes, Adam J. Thompson, Jonathan P.-H. Belnoue, Stephen R. Hallett, and Dmitry S. Ivanov. Mitigating forming defects by local modification of dry preforms. *Composites Part A*, 128:Art. 105643, 2020.



- 
- [51] A. A. Skordos, M.P.F. Sutcliffe, J. W. Klintworth, and P. Adolfsson. Multi-objective optimisation of woven composite draping using genetic algorithms. *Proceedings of the SAMPE Europe 2006*, Paris/France, 2006.
- [52] Hardy Köke, Lennart Weiß, Christian Hühne, and Michael Sinapius. A graph-based method for calculating draping strategies for the application of fiber-reinforced materials on arbitrary surfaces. *Composite Structures*, 162:p 123–132, 2017.
- [53] C. Wang and H. H. Nied. Temperature optimization for improved thickness control in thermoforming. *Journal of Materials Processing and Manufacturing Science*, 8(2):p 113–126, 1999.
- [54] M. Kaufmann, D. Zenkert, and M. Åkermo. Cost/weight optimization of composite prepreg structures for best draping strategy. *Composites Part A*, 41(4):p464–472, 2010.
- [55] Christian Krogh, James A. Sherwood, and Johnny Jakobsen. Generation of feasible gripper trajectories in automated composite draping by means of optimization. *Advanced Manufacturing: Polymer & Composites Science*, 5(4):p 234–249, 2019.
- [56] D. F. Liebau, J.M.J.F. van Campen, T. Sommer-Dittrich, and Middendorf. On the effect of draping strategy on fe-based draping simulation of basic and complex 3d geometries for automated preforming. *Proceedings of the 3rd Aircraft Structural Design Conference, Delft/Netherlands*, 2012.
- [57] Sven Coutandin, David Brandt, Paul Heinemann, Paul Ruhland, and Jürgen Fleischer. Influence of punch sequence and prediction of wrinkling in textile forming with a multi-punch tool. *Production Engineering*, 12(6):p 779–788, 2018.
- [58] Michael Jehrke. *Umformen gewebeverstärkter thermoplastischer Prepregs mit Polypropylen- und Polyamid-Matrix im Pressverfahren*. Dissertation, RWTH Aachen, Aachen, 1995.
- [59] Christian Luebbering and Hauke Lengsfeld. Verfahren und Vorrichtung zur Bereitstellung eines trockenen textilen Vorformlings (Patent DE102010040970 A1), Deutsches Patent- und Markenamt, 2012.
- [60] Daniel Liebau, Soeren Stritzel, Dittrich Thomas Sommer, and Julien van Campen. Spannrahmenvorrichtung, DE102012010497 A1, Deutsches Patent- und Markenamt, 2012.
- [61] Julian Fial, Stefan Carosella, Laurin Ring, and Peter Middendorf. Shear characterization of reinforcement fabrics using textile-applied printed strain sensors. *Procedia Manufacturing*, 47:p 65–70, 2020.
- [62] Joerg Theinert, Horst Mai, and Falko Schmidt. Verfahren zur Herstellung eines Vorformlings aus Fasermaterial, DE102010041179 A1, Deutsches Patent- und Markenamt, 2012.

- [63] Joerg Theinert and Horst Mai. Verfahren und Umformwerkzeug zum Herstellen eines Faserverbund-Vorformlings, DE102010030009 A1, Deutsches Patent- und Markenamt, 2011.
- [64] Joerg Theinert and Falko Schmidt. Verfahren zur herstellung eines zur versteifung eines herzustellenden kunststoffteils vorgesehenen vorformlings, 2011.
- [65] Torsten Flemming, Juergen Scheer, Ulrich Schiefer, Kurt Fraunhofer, and Jürgen Scheer. Verfahren zur Herstellung eines Kunststoffformteils, DE19922799 A1, Deutsches Patent- und Markenamt, 2000.
- [66] S. Chen, O. P. L. McGregor, L. T. Harper, A. Endruweit, and N. A. Warrior. Optimisation of local in-plane constraining forces in double diaphragm forming. *Composite Structures*, 201:p 570–581, 2018.
- [67] Julian Kuntz, Alexander Hemmen, Ralph Männich, and Margarita Wanner. Verfahren und Vorrichtung zur Herstellung eines Faservorformlings, DE102017223875 A1, Deutsches Patent- und Markenamt, 2019.
- [68] Michael Ostgathe, Uwe Habisreiter, and Bernhard Nordmann. Verfahren und Produktionsanlage zum Herstellen von schalenförmigen, fasermattenverstärkten Kunststoffteilen, DE10035237 C1, Europäisches Patentamt, 2001.
- [69] Markus Spoetzl. Clamping frame to hold fabrics or layers of filaments to be passed to a shaping machine, DE10129330 A1, Europäisches Patentamt, 2003.
- [70] Hanno Pfitzer, Christian Dehn, and Danilo Miglierina. Spannrahmen, DE10153035 A1, Deutsches Patent- und Markenamt, 2003.
- [71] Rajan Jagpal, Richard Butler, and Evripides G. Loukaides. Towards flexible and defect-free forming of composites through distributed clamping. *Procedia CIRP*, 85:p 341–346, 2019.
- [72] Sven Schöfer. *Kraftgesteuerte Materialzuführung beim Drapierprozess in der automatisierten Herstellung von Faserverbundwerkstoffen*. Dissertation, RWTH Aachen, Aachen, 2019.
- [73] C. Werner Dankwort, Roland Weidlich, Birgit Guenther, and Joerg E. Blaurock. Engineers' CAx education it's not only CAD. *Computer-Aided Design*, 36(14):p 1439–1450, 2004.
- [74] Biao Liang and Philippe Boisse. A review of numerical analyses and experimental characterization methods for forming of textile reinforcements. *Chinese Journal of Aeronautics*, 34(8):p 143–163, 2021.
- [75] Teik-Cheng Lim and S. Ramakrishna. Modelling of composite sheet forming: a review. *Composites Part A*, 33(4):p 515–537, 2002.

- [76] C. Mack and H. M. Taylor. The fitting of woven cloth to surfaces. *Journal of the Textile Institute Transactions*, 47(9):p T477–T488, 1956.
- [77] R.E Robertson, T.-J Chu, R.J Gerard, J.-H Kim, M. Park, H.-G Kim, and R.C Peterson. Three-dimensional fiber reinforcement shapes obtainable from flat, bidirectional fabrics without wrinkling or cutting. part 1. a single four-sided pyramid. *Composites Part A*, 31(7):p 703–715, 2000.
- [78] R.E Robertson, T.-J Chu, R.J Gerard, J.-H Kim, M. Park, H.-G Kim, and R.C Peterson. Three-dimensional fiber reinforcement shapes obtainable from flat, bidirectional fabrics without wrinkling or cutting. part 2: a single n-sided pyramid, cone, or round box. *Composites Part A*, 31(11):p 1149–1165, 2000.
- [79] K. Golden, T. G. Rogers, and A.J.M. Spencer. Forming kinematics of continuous fibre reinforced laminates. *Composites Manufacturing*, 2(3-4):p 267–277, 1991.
- [80] B. P. van West, R. B. Pipes, and M. Keefe. A simulation of the draping of bidirectional fabrics over arbitrary surfaces. *The Journal of The Textile Institute*, 81(4):p 448–460, 1990.
- [81] Abdelhakim Cherouat, Houman Borouchaki, and Jean-Louis Billoët. Geometrical and mechanical draping of composite fabrics. *Revue Européenne des Éléments Finis*, 14(6-7):p 693–707, 2005.
- [82] F. van der Weeën. Algorithms for draping fabrics on doubly-curved surfaces. *International Journal for Numerical Methods in Engineering*, 31(7):p 1415–1426, 1991.
- [83] Benedikt Fengler. *Manufacturing-constrained multi-objective optimization of local patch reinforcements for discontinuous fiber reinforced composite parts*. Dissertation, Karlsruher Institut für Technologie, Karlsruhe, 2019.
- [84] P. Badel, S. Gauthier, E. Vidal-Sallé, and P. Boisse. Rate constitutive equations for computational analyses of textile composite reinforcement mechanical behaviour during forming. *Composites Part A*, 40(8):p 997–1007, 2009.
- [85] D. J. Mortell, D. A. Tanner, and C. T. McCarthy. A virtual experimental approach to microscale composites testing. *Composite Structures*, 171:p 1–9, 2017.
- [86] Damien Durville. Simulation of the mechanical behaviour of woven fabrics at the scale of fibers. *International Journal of Material Forming*, 3(S2):p 1241–1251, 2010.
- [87] Mathias Engelfried, Julian Fial, Manuel Tartler, Patrick Böhler, Dominik Hägele, and Peter Middendorf. A mesoscopic approach for draping simulation of preforms manufactured by direct fibre placement. *Proceedings of the 20th ESAFORM*, page Art. 030016, Dublin/Ireland, 2017.

- [88] G. Creech and A. K. Pickett. Meso-modelling of non-crimp fabric composites for coupled drape and failure analysis. *Journal of Materials Science*, 41(20):p 6725–6736, 2006.
- [89] S. Allaoui, P. Boisse, S. Chatel, N. Hamila, G. Hivet, D. Soulat, and E. Vidal-Salle. Experimental and numerical analyses of textile reinforcement forming of a tetrahedral shape. *Composites Part A*, 42(6):p 612–622, 2011.
- [90] S. P. Haanappel, R.H.W. ten Thije, U. Sachs, B. Rietman, and R. Akkerman. Formability analyses of uni-directional and textile reinforced thermoplastics. *Composites Part A*, 56:p 80–92, 2014.
- [91] F. J. Schirmaier, D. Dörr, F. Henning, and L. Kärger. A macroscopic approach to simulate the forming behaviour of stitched unidirectional non-crimp fabrics (UD-NCF). *Composites Part A*, 102:p 322–335, 2017.
- [92] Gabriele Barbagallo, Angela Madeo, Fabrice Morestin, and Philippe Boisse. Modelling the deep drawing of a 3d woven fabric with a second gradient model. *Mathematics and Mechanics of Solids*, 22(11):p 2165–2179, 2017.
- [93] P. Boisse, R. Bai, J. Colmars, N. Hamila, B. Liang, and A. Madeo. The need to use generalized continuum mechanics to model 3D textile composite forming. *Applied Composite Materials*, 25(4):p 761–771, 2018.
- [94] Renzi Bai, Julien Colmars, Naim Naouar, and Philippe Boisse. A specific 3d shell approach for textile composite reinforcements under large deformation. *Composites Part A*, 139:Art. 106135, 2020.
- [95] Bastian Schäfer, Dominik Dörr, and Luise Kärger. Potential and challenges of a solid-shell element for the macroscopic forming simulation of engineering textiles. *Proceedings of the 24th ESAFORM*, permanent URL: <https://popups.uliege.be/esaform21/index.php?id=883>, 2021.
- [96] Christian Poppe, Constantin Krauß, Fabian Albrecht, and Luise Kärger. A 3d process simulation model for wet compression moulding. *Composites Part A*, 145:Art. 106379, 2021.
- [97] Christian Poppe. *Process simulation of wet compression moulding for continuous fibre-reinforced polymers*. Dissertation, Karlsruher Institut für Technologie, Karlsruhe, 2021.
- [98] Nahiène Hamila and Philippe Boisse. Locking in simulation of composite reinforcement deformations - analysis and treatment. *Composites Part A*, 53:p 109–117, 2013.
- [99] R.H.W. ten Thije and R. Akkerman. Solutions to intra-ply shear locking in finite element analyses of fibre reinforced materials. *Composites Part A*, 39(7):p 1167–1176, 2008.

- 
- [100] Christian Poppe, Dominik Dörr, Frank Henning, and Luise Kärger. Experimental and numerical investigation of the shear behaviour of infiltrated woven fabrics. *Composites Part A*, 114:p 327–337, 2018.
- [101] Dominik Dörr, Frank Henning, and Luise Kärger. Nonlinear hyperviscoelastic modelling of intra-ply deformation behaviour in finite element forming simulation of continuously fibre-reinforced thermoplastics. *Composites Part A*, 109:p 585–596, 2018.
- [102] Dominik Dörr, Fabian J. Schirmaier, Frank Henning, and Luise Kärger. A viscoelastic approach for modeling bending behavior in finite element forming simulation of continuously fiber reinforced composites. *Composites Part A*, 94:113–123, 2017.
- [103] Benedikt Fengler, Luise Kärger, Frank Henning, and Andrew Hrymak. Multi-objective patch optimization with integrated kinematic draping simulation for continuous–discontinuous fiber-reinforced composite structures. *Journal of Composites Science*, 2(2):Art. 22, 2018.
- [104] Clemens Zimmerling, Christian Poppe, Oliver Stein, and Luise Kärger. Optimisation of manufacturing process parameters for variable component geometries using reinforcement learning. *Materials & Design*, 214:Art. 110423, 2022.
- [105] Constantin Krauß and Luise Kärger. Tensor interpolation in virtual manufacturing chains for fiber reinforced composites. *International Journal of Mechanical Sciences*, 226:Art. 107378, 2022.
- [106] Luise Kärger, Alexander Bernath, Florian Fritz, Siegfried Galkin, Dino Magagnato, André Oeckerath, Alexander Schön, and Frank Henning. Development and validation of a cae chain for unidirectional fibre reinforced composite components. *Composite Structures*, 132:p 350–358, 2015.
- [107] S. M. Afazov. Modelling and simulation of manufacturing process chains. *CIRP Journal of Manufacturing Science and Technology*, 6(1):p 70–77, 2013.
- [108] Luise Kärger, Siegfried Galkin, Clemens Zimmerling, Dominik Dörr, Fabian Schirmaier, A. Oeckerath, Klaus Wolf, and J. Linden. Continuous CAE chain for composite design, established on an HPC system and accessible via web-based user-interfaces. In *Proceedings of the NAFEMS World Congress 2017*. Stockholm/Sweden, 2017.
- [109] Johannes Görthofer, Nils Meyer, Tarkes Dora Pallicity, Ludwig Schöttl, Anna Trauth, Malte Schemmann, Martin Hohberg, Pascal Pinter, Peter Elsner, Frank Henning, Andrew Hrymak, Thomas Seelig, Kay Weidenmann, Luise Kärger, and Thomas Böhlke. Motivating the development of a virtual process chain for sheet molding compound composites. *Proceedings Applied Mathematics and Mechanics (PAMM)*, 19(1):Art. e201900124, 2019.

- [110] Johannes Görthofer, Nils Meyer, Tarkes Dora Pallicity, Ludwig Schöttl, Anna Trauth, Malte Schemmann, Martin Hohberg, Pascal Pinter, Peter Elsner, Frank Henning, Andrew Hrymak, Thomas Seelig, Kay Weidenmann, Luise Kärger, and Thomas Böhlke. Virtual process chain of sheet molding compound: Development, validation and perspectives. *Composites Part B*, 169:p 133–147, 2019.
- [111] Martin Hohberg, Luise Kärger, Andrew Hrymak, and Frank Henning. Process simulation of sheet molding compound (SMC) as key for the integrated simulation chain. In *Proceedings of Simulation von Composites 2016*. Hamburg, 2016.
- [112] Frédéric Masseria, André Berger, and Benjamin Kaiser. Virtual composite manufacturing simulation chain. *Materials Science Forum*, 825-826:p 671–676, 2015.
- [113] Natalie Mayer, Jens Prowe, Tamas Havar, Roland Hinterhölzl, and Klaus Drechsler. Structural analysis of composite components considering manufacturing effect. *Composite Structures*, 140:p 776–782, 2016.
- [114] Natalie Mayer, Björn van den Broucke, Jens Prowe, Tamas Havar, and Roland Hinterhölzl. Finite element mapping for incompatible fe meshes of composite structures. *Advances in Engineering Software*, 99:81–88, 2016.
- [115] Hans J. Fahrenwaldt, Volkmar Schuler, and Jürgen Twrdek. *Praxiswissen Schweißtechnik*. Springer Fachmedien, Wiesbaden, 2014.
- [116] Gerhard Oehler and Kaiser. *Schnitt-, Stanz- und Ziehwerkzeuge*. Springer Verlag, Berlin/Heidelberg, 1993.
- [117] David M. Anderson. *Design for manufacturability*. Routledge Taylor & Francis, New York/USA, 2 edition, 2020.
- [118] M. H. A. Bonte, A. H. van den Boogaard, and J. Huétink. An optimisation strategy for industrial metal forming processes. *Structural and Multidisciplinary Optimization*, 35(6):p 571–586, 2008.
- [119] Phil Reeves, Chris Tuck, and Richard Hague. Additive manufacturing for mass customization. In Flavio S. Fogliatto and Giovanni J. Da Silveira, editors, *Mass customization*, Springer series in advanced manufacturing, pages p 275–289. Springer, London/United Kingdom, 2011.
- [120] Patrick Moll, Markus Schäfer, Sven Coutandin, and Jürgen Fleischer. Reconfigurable modular production plant for thermoplastic hybrid composites. *Production Engineering*, 13(3-4):469–477, 2019.
- [121] Mohamed Abosaf, Khamis Essa, Ali Alghawail, Abror Tolipov, Shizhong Su, and Duc Pham. Optimisation of multi-point forming process parameters. *The International Journal of Advanced Manufacturing Technology*, 92(5-8):p 1849–1859, 2017.

- 
- [122] Daniel Opritescu and Wolfram Volk. Variation of components by automated driving. *International Journal of Material Forming*, 9(1):p 9–19, 2016.
- [123] Sigrún Andradóttir. A review of simulation optimization techniques. In *Proceedings of the 30th Conference on Winter Simulation*, pages 151–158, Washington DC, USA, 1998. IEEE Computer Society Press.
- [124] Carl Geiger and Christian Kanzow. *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer Verlag, Berlin/Heidelberg, 2002.
- [125] David G. Green. *Dual Phase Evolution*. Springer Verlag, New York/USA, 2014.
- [126] Peter W. Christensen and A. Klarbring. *An Introduction to Structural Optimization*, volume 153. Springer Netherlands, Dordrecht, 2008.
- [127] N. van Dijk, M. Langelaar, and F. van Keulen. Critical study of design parameterization in topology optimization. In *Proceedings of the 2nd International Conference on Engineering Optimization*, 2010.
- [128] Uri Kirsch. *Structural Optimization*. Springer Verlag, Berlin/Heidelberg, 1993.
- [129] R. Kriechbaum. CAIO (Computer Aided Internal Optimization). In *First European Conference on Smart Structures and Materials*, pages 281–284. SPIE Digital Library, 1992.
- [130] D. Zink, C. Awe, and P. Middendorf. Automated design approach and potential assessment of composite structures: A fast analytical engineering tool for multiple load cases. *Proceedings of ECCM17*, München, 2016.
- [131] Clemens Zimmerling, Benedikt Fengler, Constantin Krauß, and Luise Kärger. Optimisation of layup type and fibre orientation in continuous-fibre reinforced components via anisotropy analysis. *Composite Structures*, 287:Art. 115347, 2022.
- [132] Stéphane Alarie, Charles Audet, Aïmen E. Gheribi, Michael Kokkolaras, and Sébastien Le Digabel. Two decades of blackbox optimization applications. *Journal on Computational Optimization*, 9:Art. 100011, 2021.
- [133] Andrew R. Conn, Katya Scheinberg, and Luis N. Vicente. *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, Philadelphia/USA, 2009.
- [134] Oliver Stein. *Nichtlineare Optimierung I und II: Skript zur Vorlesung am Karlsruher Institut für Technologie*. Lecture notes, Karlsruher Institut für Technologie, Karlsruhe, 2014.

- [135] Sascha Dähne and Christian Hühne. Gradient based structural optimization of a stringer stiffened composite wing box with variable stringer orientation. In Axel Schumacher, Thomas Vietor, Sierk Fiebig, Kai-Uwe Bletzinger, and Kurt Maute, editors, *Advances in Structural and Multidisciplinary Optimization*, pages 814–826. Springer International Publishing, Cham/Switzerland, 2018.
- [136] Odeh Dababneh, Timoleon Kipouros, and James Whidborne. Application of an efficient gradient-based optimization strategy for aircraft wing structures. *Aerospace*, 5(1):Art. 3, 2018.
- [137] Panos M. Pardalos, Varvara Rasskazova, and Vrahatis, Michael, N. *Black Box Optimization, Machine Learning and No-Free Lunch Theorems*, volume 170. Springer International Publishing, Cham/Switzerland, 2021.
- [138] R. Kathiravan and R. Ganguli. Strength design of composite beam using gradient and particle swarm optimization. *Composite Structures*, 81(4):p 471–479, 2007.
- [139] Nicolas Lachenmaier, Daniel Baumgärtner, Heinz-Peter Schiffer, and Johannes Kech. Gradient-free and gradient-based optimization of a radial turbine. *International Journal of Turbomachinery, Propulsion and Power*, 5(3):Art. 14, 2020.
- [140] Natalia M. Alexandrov, John E. Dennis, Robert Michael Lewis, and Virginia Torczon. A trust-region framework for managing the use of approximation models in optimization. *Structural optimization*, 15(1):p 16–23, 1998.
- [141] Stefan M. Wild, Rommel G. Regis, and Christine A. Shoemaker. ORBIT: Optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing*, 30(6):p 3197–3219, 2008.
- [142] Anthony Giunta and Michael Eldred. Implementation of a trust region model management strategy in the DAKOTA optimization toolkit. In *8th Symposium on Multidisciplinary Analysis and Optimization*, American Institute of Aeronautics and Astronautics, Reston/USA, 2000.
- [143] Slawomir Koziel and Sigmar D. Unnsteinsson. Expedited design closure of antennas by means of trust-region-based adaptive response scaling. *IEEE Antennas and Wireless Propagation Letters*, 17(6):p 1099–1103, 2018.
- [144] Charles Audet, Andrew R. Conn, Sébastien Le Digabel, and Mathilde Peyrega. A progressive barrier derivative-free trust-region algorithm for constrained optimization. *Computational Optimization and Applications*, 71(2):p 307–329, 2018.
- [145] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):p 308–313, 1965.



- 
- [146] J. R. Perez and J. Basterrechea. Comparison of different heuristic optimization methods for near-field antenna measurements. *IEEE Transactions on Antennas and Propagation*, 55(3):p 549–555, 2007.
- [147] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary computation*, 8(2):p 173–195, 2000.
- [148] A. Muc and W. Gurba. Genetic algorithms and finite element analysis in optimization of composite structures. *Composite Structures*, 54(2-3):p 275–281, 2001.
- [149] Benedikt Fengler, Andrew Hrymak, and Luise Kärger. Multi-objective CoFRP patch optimization with consideration of manufacturing constraints and integrated warpage simulation. *Composite Structures*, 221:Art. 110861, 2019.
- [150] S. Chen, A. Endruweit, L. T. Harper, and N. A. Warrior. Inter-ply stitching optimisation of highly drapeable multi-ply preforms. *Composites Part A*, 71:p 144–156, 2015.
- [151] S. Chen, L. T. Harper, A. Endruweit, and N. A. Warrior. Formability optimisation of fabric preforms by controlling material draw-in through in-plane constraints. *Composites Part A*, 76:p 10–19, 2015.
- [152] Julius Pfrommer, Clemens Zimmerling, Jinzhao Liu, Luise Kärger, Frank Henning, and Jürgen Beyerer. Optimisation of manufacturing process parameters using deep neural networks as surrogate models. *Procedia CIRP*, 72:p 426–431, 2018.
- [153] Slawomir Koziel and Leifur Leifsson. *Surrogate-Based Modeling and Optimization: Applications in Engineering*. Springer Science+Business Media, New York/USA, 2013.
- [154] Bruno Sudret, Stefano Marelli, and Joe Wiart. Surrogate models for uncertainty quantification: An overview. In *11th European Conference on Antennas and Propagation (EUCAP)*, pages 793–797. Paris/France, 2017.
- [155] Maliki Moustapha and Bruno Sudret. Surrogate-assisted reliability-based design optimization: a survey and a unified modular framework. *Structural and Multidisciplinary Optimization*, 60(5):p 2157–2176, 2019.
- [156] Georgios Balokas, Benedikt Kriegesmann, and Raimund Rolfes. Data-driven inverse uncertainty quantification in the transverse tensile response of carbon fiber reinforced composites. *Composites Science and Technology*, 211:108845, 2021.
- [157] Georgios Balokas, Benedikt Kriegesmann, Steffen Czichon, and Raimund Rolfes. A variable-fidelity hybrid surrogate approach for quantifying uncertainties in the nonlinear response of braided composites. *Computer Methods in Applied Mechanics and Engineering*, 381:Art. 113851, 2021.

- [158] S. Dey, T. Mukhopadhyay, A. Spickenheuer, S. Adhikari, and G. Heinrich. Bottom up surrogate based approach for stochastic frequency response analysis of laminated composite plates. *Composite Structures*, 140:p 712–727, 2016.
- [159] Benedikt Kriegesmann and Julian K. Lüdeker. Robust compliance topology optimization using the first-order second-moment method. *Structural and Multidisciplinary Optimization*, 60(1):p 269–286, 2019.
- [160] Minjie Zhu and Michael H. Scott. Direct differentiation of the particle finite-element method for fluid–structure interaction. *Journal of Structural Engineering*, 142(3):Art. 04015159, 2016.
- [161] Hamed Ebrahimiyan, Rodrigo Astroza, and Joel P. Conte. Extended Kalman filter for material parameter estimation in nonlinear structural finite element models using direct differentiation method. *Earthquake Engineering & Structural Dynamics*, 44(10):p 1495–1522, 2015.
- [162] Karl Siebertz, David van Bebber, and Thomas Hochkirchen. *Statistische Versuchsplanung: Design of Experiments (DoE)*. Springer-Verlag, Berlin Heidelberg, 2010.
- [163] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):p 409–423, 1989.
- [164] V. Roshan Joseph. Space-filling designs for computer experiments: A review. *Quality Engineering*, 28(1):p 28–35, 2016.
- [165] A. Olsson, G. Sandberg, and O. Dahlblom. On Latin Hypercube sampling for structural reliability analysis. *Structural Safety*, 25(1):p 47–68, 2003.
- [166] Matthew David Marko. The impact of lubricant film thickness and ball bearings failures. *Lubricants*, 7(6):Art. 48, 2019.
- [167] Alexander Kaplun and Arkadiy Meshalkin. Simple self-empirical equation of state of liquid and gas for engineering calculations. *Journal of Chemical & Engineering Data*, 56(4):p 1463–1467, 2011.
- [168] Yuen Kuan Yong, Tien-Fu Lu, and Daniel C. Handley. Review of circular flexure hinge design equations and derivation of empirical formulations. *Precision Engineering*, 32(2):p 63–70, 2008.
- [169] A. Krishnakumari, A. Elayaperumal, M. Saravanan, and C. Arvindan. Fault diagnostics of spur gear using decision tree and fuzzy classifier. *International Journal of Advanced Manufacturing Technology*, 89(9-12):p 3487–3494, 2017.
- [170] Barbara Hammer and Kai Gersmann. A note on the universal approximation capability of support vector machines. *Neural Processing Letters*, 17(1):p 43–53, 2003.

- 
- [171] Dustin Tran, Rajesh Ranganath, and David M. Blei. The Variational Gaussian Process, *arXiv:1511.06499*, 2015.
- [172] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):p 251–257, 1991.
- [173] Sandia National Laboratories. DAKOTA user’s manual - version 6.15, [www.dakota.sandia.gov/sites/default/files/docs/6.15/Users-6.15.0.pdf](http://www.dakota.sandia.gov/sites/default/files/docs/6.15/Users-6.15.0.pdf), 2021, last accessed: 31.08.2022.
- [174] The MathWorks Inc. Statistics and Machine Learning toolbox documentation, [www.mathworks.com/help/stats](http://www.mathworks.com/help/stats), 2022, last accessed: 31.08.2022.
- [175] Stefano Marelli and Bruno Sudret. UQLab: A framework for uncertainty quantification in Matlab. In Michael Beer, Siu-Kui Au, and Jim W. Hall, editors, *Vulnerability, Uncertainty, and Risk*, pages 2554–2563. American Society of Civil Engineers, Reston/USA, 2014.
- [176] David H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):p 1341–1390, 1996.
- [177] Jianguang Fang, Guangyong Sun, Na Qiu, Nam H. Kim, and Qing Li. On design optimization for structural crashworthiness and its state of the art. *Structural and Multidisciplinary Optimization*, 55(3):p 1091–1119, 2017.
- [178] S. Dey, T. Mukhopadhyay, and S. Adhikari. Metamodel based high-fidelity stochastic analysis of composite laminates: A concise review with critical comparative assessment. *Composite Structures*, 171:p 227–250, 2017.
- [179] Hu Wang, Fan Ye, Lei Chen, and Enying Li. Sheet metal forming optimization by using surrogate modeling techniques. *Chinese Journal of Mechanical Engineering*, 30(1):p 22–36, 2017.
- [180] Clemens Zimmerling, Christian Poppe, and Luise Kärger. Virtuelle Produktentwicklung mittels Simulationsmethoden und KI. *Lightweight Design*, 12(6):p 12–19, 2019.
- [181] Clemens Zimmerling, Patrick Schindler, Julian Seuffert, and Luise Kärger. Deep neural networks as surrogate models for time-efficient manufacturing process optimization. *Proceedings of the 24th ESAFORM*, permanent URL: [popups.uliege.be/esaforum21/index.php?id=3882](http://popups.uliege.be/esaforum21/index.php?id=3882), 2021.
- [182] J. W. Bandler, Q. S. Cheng, S. A. Dakrouy, A. S. Mohamed, M. H. Bakr, K. Madsen, and J. Sondergaard. Space mapping: The state of the art. *IEEE Transactions on Microwave Theory and Techniques*, 52(1):p 337–361, 2004.
- [183] T. Jansson, A. Andersson, and L. Nilsson. Optimization of draw-in for an automotive sheet metal part. *Journal of Materials Processing Technology*, 159(3):p 426–434, 2005.

- [184] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):p 455–492, 1998.
- [185] Mohamed Amine Bouhlel, Nathalie Bartoli, Rommel G. Regis, Abdelkader Otsmane, and Joseph Morlier. Efficient global optimization for high-dimensional constrained problems by Kriging models combined with the partial least squares method. *Engineering Optimization*, 50(12):p 2038–2053, 2018.
- [186] Raphael T. Haftka, Diane Villanueva, and Anirban Chaudhuri. Parallel surrogate-assisted global optimization with expensive functions – a survey. *Structural and Multidisciplinary Optimization*, 54(1):p 3–13, 2016.
- [187] Hany El Kadi. Modeling the mechanical behavior of fiber-reinforced polymeric composite materials using artificial neural networks - a review. *Composite Structures*, 73(1):p 1–23, 2006.
- [188] Elia Hamouche and Evripides G. Loukaides. Classification and selection of sheet forming processes with machine learning. *International Journal of Computer Integrated Manufacturing*, 31(9):p 921–932, 2018.
- [189] Navdeep Kumar and Pankaj Garg. Recognition of distorted CAD objects using neural networks. *International Journal of Computer Applications*, 14:p 18–22, 2011.
- [190] K. G. Swift and J. D. Booker. *Manufacturing process selection handbook*. Butterworth-Heinemann, Oxford/United Kingdom, 2013.
- [191] Tom M. Mitchell. *Machine learning*. McGraw-Hill, New York/USA, 2010.
- [192] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc, 2012.
- [193] Thomas McLeay, Michael S. Turner, and Keith Worden. A novel approach to machining process fault detection using unsupervised learning. *Proceedings of the Institution of Mechanical Engineers: Part B*, 235(10):p 1533–1542, 2021.
- [194] Christian Schwarz, Patrick Ackert, and Reinhard Mauermann. Principal component analysis and singular value decomposition used for a numerical sensitivity analysis of a complex drawn part. *International Journal of Advanced Manufacturing Technology*, 94(5-8):p 2255–2265, 2018.
- [195] Richard S. Sutton and Andrew Barto. *Reinforcement learning: An introduction*. The MIT Press, Cambridge/USA and London/United Kingdom, 2 edition, 2018.

- 
- [196] Johannes Dornheim, Lukas Morand, Samuel Zeitvogel, Tarek Iraki, Norbert Link, and Dirk Helm. Deep reinforcement learning methods for structure-guided processing path optimization. *Journal of Intelligent Manufacturing*, 1(12):Art. 4022, 2021.
- [197] Johannes Günther, Patrick M. Pilarski, Gerhard Helfrich, Hao Shen, and Klaus Diepold. Intelligent laser welding through representation, prediction, and control learning: An architecture with deep neural networks and reinforcement learning. *Mechatronics*, 34:p 1–11, 2016.
- [198] Zhenpeng Zhou, Xiaocheng Li, and Richard N. Zare. Optimizing chemical reactions with deep reinforcement learning. *ACS central science*, 3(12):p 1337–1344, 2017.
- [199] C. Lataniotis, S. Marelli, and B. Sudret. Extending classical surrogate modelling to high-dimensions through supervised dimensionality reduction: a data-driven approach, *arXiv:1812.06309v3*, 2018.
- [200] Amir Barati Farimani, Joseph Gomes, and Vijay S. Pande. Deep learning the physics of transport phenomena, *arXiv:1709.02432v1*, 2017.
- [201] C. O. S. Sorzano, J. Vargas, and A. Pascual Montano. A survey of dimensionality reduction techniques, *arXiv:1403.2877v1*, 2014.
- [202] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge/USA, 2016.
- [203] François Chollet and community. Keras (open source project), <https://keras.io>, 2015, last accessed: 09.09.2022.
- [204] Wolfgang Ertel. *Grundkurs Künstliche Intelligenz*. Springer Fachmedien, Wiesbaden, 2016.
- [205] Warren McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:p 115–133, 1943.
- [206] Tomasz Szandała. Review and comparison of commonly used activation functions for deep neural networks, *arXiv:2010.09458v1*, 2021. 2021.
- [207] Brosnan Yuen, Minh Tu Hoang, Xiaodai Dong, and Tao Lu. Universal activation function for machine learning. *Scientific reports*, 11(1):Art. 18757, 2021.
- [208] K. Fukushima. Neocognitron: A self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):p 193–202, 1980.
- [209] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):p 541–551, 1989.

- [210] Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM*, pages 481–490. 2016.
- [211] Clemens Zimmerling, Daniel Trippe, Benedikt Fengler, and Luise Kärger. An approach for rapid prediction of textile draping results for variable composite component geometries using deep neural networks. In *AIP Conference Proceedings*, 2113: Art. 020007, 22nd ESAFORM, Vitoria-Gasteiz/Spain, 2019.
- [212] Hamid Reza Attar, Haosu Zhou, Alistair Foster, and Nan Li. Rapid feasibility assessment of components to be formed through hot stamping: A deep learning approach. *Journal of Manufacturing Processes*, 68:p 1650–1671, 2021.
- [213] Haosu Zhou, Qingfeng Xu, Zhenguo Nie, and Nan Li. A study on using image-based machine learning methods to develop surrogate models of stamp forming simulations. *Journal of Manufacturing Science and Engineering*, 144(2):Art. 021012, 2022.
- [214] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):p 533–536, 1986.
- [215] Léon Bottou. Online learning and stochastic approximations. In David Saad, editor, *Online Learning in Neural Networks*, pages 9–42. Cambridge University Press/USA, 2010.
- [216] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):p 1929–1958, 2014.
- [217] Marc Claesen and Bart De Moor. Hyperparameter search in machine learning, *arXiv:1502.02127v2*, 2015.
- [218] Mohammad Tabatabaei, Jussi Hakanen, Markus Hartikainen, Kaisa Miettinen, and Karthik Sindhya. A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods. *Structural and Multidisciplinary Optimization*, 52(1):p 1–25, 2015.
- [219] Clemens Zimmerling, Julius Pfrommer, Jinzhao Liu, Jürgen Beyerer, Frank Henning, and Luise Kärger. Application and evaluation of meta-model assisted optimisation strategies for gripper-assisted fabric draping in composite manufacturing. In *Proceedings of the 18th ECCM*. Athens/Greece, 2018.
- [220] John Henry Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, Ann Arbor/USA, 2 edition, 1975.
- [221] Thomas Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, Oxford/United Kingdom, 1996.

- 
- [222] John Eddy and Kemper Lewis. Effective generation of pareto sets using genetic programming. In *Proceedings of DETC 2001*, pages 783–791. Pittsburgh/USA, 2001.
- [223] Paul Levi. *Einführung in die Informatik: Für Naturwissenschaftler und Ingenieure*. Hanser Verlag, München, 4 edition, 2002.
- [224] R. Storn. On the usage of differential evolution for function optimization. In *Proceedings of NAFIP 1996*, pages 519–523, Berkeley/USA, 1996.
- [225] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the 14th Conference on Artificial Intelligence and Statistics*, pages 315–323, Fort Lauderdale/USA, 2011.
- [226] Axel Schumacher. *Optimierung mechanischer Strukturen*. Springer Vieweg, Berlin/Heidelberg, 3 edition, 2020.
- [227] Guocheng Pan, Dave Gaard, Ken Moss, and Tim Heiner. A comparison between cokriging and ordinary kriging: Case study with a polymetallic deposit. *Mathematical Geology*, 25(3):p 377–398, 1993.
- [228] Boukje M. de Gooijer, Jos Havinga, Hubert J. M. Geijselaers, and Anton H. van den Boogaard. Evaluation of POD based surrogate models of fields resulting from nonlinear FEM simulations. *Advanced Modeling and Simulation in Engineering Sciences*, 8(1):Art: 25, 2021.
- [229] Liang Liang, Minliang Liu, Caitlin Martin, and Wei Sun. A deep learning approach to estimate stress distribution: A fast and accurate surrogate of finite-element analysis. *Journal of the Royal Society, Interface*, 15(138):Art. 20170844, 2018.
- [230] Saeed Mohammadzadeh and Emma Lejeune. Predicting mechanically driven full-field quantities of interest with deep learning-based metamodels. *Extreme Mechanics Letters*, 50:Art. 101566, 2022.
- [231] Fabian Albrecht, Clemens Zimmerling, Christian Poppe, Luise Kärger, and Frank Henning. Development of a modular draping test bench for analysis of infiltrated woven fabrics in wet compression molding. *Key Engineering Materials*, 809:p 35–40, 2019.
- [232] Patrick Schindler. *Untersuchung zum Konvergenzverhalten metamodel-unterstützter Optimierungsmethoden am Beispiel der RTM-Formfüllung*. Master thesis (in German, mentoring by C. Zimmerling and J. Seuffert), Karlsruher Institut für Technologie, Institute of Vehicle Systems Technology, Karlsruhe, 2019.

- [233] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide and deep Learning for recommender systems, *arXiv:1606.07792v1*, 2016.
- [234] Daniel Opretescu and Wolfram Volk. Automated driving for individualized sheet metal part production—a neural network approach. *Robotics and Computer-Integrated Manufacturing*, 35:p 144–150, 2015.
- [235] Christoph Hartmann, Daniel Opretescu, and Wolfram Volk. An artificial neural network approach for tool path generation in incremental sheet metal free-forming. *Journal of Intelligent Manufacturing*, 30(2):p 757–770, 2019.
- [236] Joost R. Dufflou, Anne-Marie Habraken, Jian Cao, Rajiv Malhotra, Markus Bambach, Dave Adams, Hans Vanhove, Amirahmad Mohammadi, and Jack Jeswiet. Single point incremental forming: State-of-the-art and prospects. *International Journal of Material Forming*, 11(6):p 743–773, 2018.
- [237] Clemens Zimmerling, Dominik Dörr, Frank Henning, and Luise Kärger. A meta-model based approach for rapid formability estimation of continuous fibre reinforced components. In *AIP Conference Proceedings*, 1960: Art. 020042, ESAFORM 2018, Palermo/Italy.
- [238] Clemens Zimmerling, Dominik Dörr, Frank Henning, and Luise Kärger. A machine learning assisted approach for textile formability assessment and design improvement of composite components. *Composites Part A*, 124:Art. 105459, 2019.
- [239] Hong Yao, Brad L. Kinsey, and Jian Cao. Rapid design of corner restraining force in deep drawn rectangular parts. *International Journal of Machine Tools and Manufacture*, 40(1):p 113–131, 2000.
- [240] Mark Colgan and John Monaghan. Deep drawing process: analysis and experiment. *Journal of Materials Processing Technology*, 132(1-3):p 35–41, 2003.
- [241] R. Padmanabhan, M. C. Oliveira, J. L. Alves, and L. F. Menezes. Influence of process parameters on the deep drawing of stainless steel. *Finite Elements in Analysis and Design*, 43(14):p 1062–1067, 2007.
- [242] H. Zein, M. El-Sherbiny, M. Abd-Rabou, and M. El Shazly. Effect of die design parameters on thinning of sheet metal in the deep drawing process. *American Journal of Mechanical Engineering*, 1(2):p 20–29, 2013.
- [243] N. M. Yusop and A. B. Abdullah. Evaluation of square deep drawn AA6061-T6 blank based on thinning pattern. *IOP Conference Series: Materials Science and Engineering*, 670(1):Art. 012079, 2019.



- [244] Philippa M. Horton, Julian M. Allwood, Christopher Cleaver, and Adam Nagy-Sochacki. An experimental analysis of the relationship between the corner, die and punch radii in forming isolated flanged shrink corners from Al 5251. *Journal of Materials Processing Technology*, 278:Art. 116486, 2020.
- [245] Hamid Reza Attar, Nan Li, and Alistair Foster. A new design guideline development strategy for aluminium alloy corners formed through cold and hot stamping processes. *Materials & Design*, 207:Art. 109856, 2021.
- [246] Patrick Böhler, Frank Härtel, and Peter Middendorf. Identification of forming limits for unidirectional carbon textiles in reality and mesoscopic simulation. *Key Engineering Materials*, 554-557:p 423-432, 2013.
- [247] S. G. Hancock and K. D. Potter. Inverse drape modelling - an investigation of the set of shapes that can be formed from continuous aligned woven fibre reinforcements. *Composites Part A*, 36(7):p 947-953, 2005.
- [248] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, Cambridge/USA, 2006.
- [249] Giuseppina Ambrogio, L. de Napoli, Luigino Filice, and M. Muzzupappa. Experimental evidences concerning geometrical accuracy after unclamping and trimming incrementally formed components. *Key Engineering Materials*, 344:p 535-542, 2007.
- [250] Giuseppina Ambrogio, Claudio Ciancio, Luigino Filice, and Francesco Gagliardi. Innovative metamodelling-based process design for manufacturing: an application to incremental sheet forming. *International Journal of Material Forming*, 10(3):p 279-286, 2017.
- [251] Bayraktar E and Altıntaş S. Square cup deep drawing experiments. In *Proceedings of the 2nd Numisheet Conference*, pages 441-446, Isehara/Japan, 1993.
- [252] Y. C. Fung, Pin Tong, and Xiaohong Chen. *Classical and Computational Solid Mechanics*. World Scientific Publishing, Singapore/Singapore, 2017.
- [253] John P. Lewis. Fast template matching. In *Proceedings of the Vision Interface*, pages 120-123, Quebec City/Canada, 1995.
- [254] Clemens Zimmerling, Benedikt Fengler, and Luise Kärger. Formability assessment of variable geometries using machine learning - analysis of the influence of the database. *Key Engineering Materials*, 926:p 2247-2257, 2022.
- [255] Ravi Kumar Gupta and Balan Gurumoorthy. Classification, representation, and automatic extraction of deformation features in sheet metal parts. *Computer-Aided Design*, 45(11):p 1469-1484, 2013.

- [256] Ravi Kumar Gupta, Yicha Zhang, Alain Bernard, and Balan Gurumoorthy. Generic classification and representation of shape features in sheet-metal parts. In S. Kumar and H. M. A. Hussein, editors, *AI applications in sheet metal forming*, pages 15–26. Springer, Singapore/Singapore, 2017.
- [257] Vishakh Hegde and Reza Zadeh. FusionNet: 3D object classification using multiple data representations, *arXiv:1607.05695v4*, 2016.
- [258] Dan Dolonius. *Sparse Voxel DAGs for Shadows and for Geometry with Colors*. Graduation Thesis (approx. equiv. to PhD), Chalmers University of Technology, Gothenburg, Sweden, 2018.
- [259] Lovedeep Gondara. Medical image denoising using convolutional denoising autoencoders, *arXiv: 1608.04667v2*, 2016.
- [260] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation, *arXiv:1505.04597*, 2015.
- [261] Daniel Trippe. *Untersuchung der Eignung tiefer neuronaler Netze zur zeiteffizienten Bewertung der Drapierbarkeit endlosfaserverstärkter Bauteile*. Master thesis (in German, mentoring by C. Zimmerling), Karlsruher Institut für Technologie, Institute of Vehicle Systems Technology (KIT-FAST), Karlsruhe, 2019.
- [262] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th ICML*, pages 1096–1103. Helsinki/Finland, 2008.
- [263] Prem S. Mann. *Mann’s introductory statistics*. Wiley, Hoboken/USA, 2017.
- [264] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning *arXiv:1509.02971v6*, 2015.
- [265] Clemens Zimmerling, Christian Poppe, and Luise Kärger. Estimating optimum process parameters in textile draping of variable part geometries - a reinforcement learning approach. *Procedia Manufacturing*, 47:p 847–854, 2020.
- [266] Rajan Jagpal, Evangelos Evangelou, Richard Butler, and Evripides G. Loukaides. Preforming of non-crimp fabrics with distributed magnetic clamping and bayesian optimisation. *Journal of Composite Materials*, Art. 002199832211036, 2022.
- [267] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures, *arXiv:1206.5533v2*, 2012.
- [268] D. R. Wilson and T. R. Martinez. The need for small learning rates on large problems. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 115–119. IEEE Xplore Digital Library, New York City/USA, 2001.

- [269] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):Art. 60, 2019.
- [270] Christoph Geisendörfer. *Effiziente Optimierung aufwendiger Finite-Volumen-Simulationsmodelle am Beispiel der Formfüllsimulation von Hochleistungsfaserverbunden*. Master thesis (in German, co-mentored by C. Zimmerling from KIT-FAST), Karlsruher Institut für Technologie, Institute for Operations Research (KIT-IOR), Karlsruhe, 2020.
- [271] Jorge J. Moré and Stefan M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):p 172–191, 2009.
- [272] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving non-linear partial differential equations. *Journal of Computational Physics*, 378:p 686–707, 2019.
- [273] Sina Amini Niaki, Ehsan Haghghat, Trevor Campbell, Anoush Poursartip, and Reza Vaziri. Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture. *Computer Methods in Applied Mechanics and Engineering*, 384:Art. 113959, 2021.
- [274] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks, *arXiv:2002.09405v2*, 2020.
- [275] Jonas Zoll. *Injection Molding Simulation based on Graph Neural Networks*. Bachelor thesis, Karlsruher Institut für Technologie, Institute for Program Structures and Data Organization (IPD), Karlsruhe, 2022.



# List of publications

## Journal articles

- [C. Zimmerling](#), D. Dörr, F. Henning, L. Kärger: A machine learning assisted approach for textile formability assessment and design improvement of composite components. *Composites Part A*, 124: Art. 105459, 2019
- [C. Zimmerling](#), C. Poppe, O. Stein, L. Kärger: Optimisation of manufacturing process parameters for variable component geometries using reinforcement learning. *Materials and Design*, 214: Art. 110423, 2022
- [C. Zimmerling](#), B. Fengler, C. Krauß, L. Kärger: Optimisation of layup type and fibre orientation in continuous-fibre reinforced components via anisotropy analysis. *Composite Structures*, 287: Art. 115347, 2022

## Conference contributions

### With conference proceedings

- [C. Zimmerling](#), D. Dörr, F. Henning, L. Kärger: A meta-model based approach for rapid formability estimation of continuous fibre reinforced components. *AIP Conference Proceedings*, 1960: Art. 020042, ESAFORM 2018, Palermo/Italy, 2018

- [C. Zimmerling](#), J. Pfrommer, J. Liu, J. Beyerer, F. Henning, L. Kärger: Application and evaluation of meta-model assisted optimisation strategies for gripper-assisted fabric draping in composite manufacturing. *KITopen*, KITopen-ID: 1000099317, ECCM 2018, Athens/Greece, 2018
- [C. Zimmerling](#), D. Trippe, B. Fengler, L. Kärger: An approach for rapid prediction of textile draping results for variable composite component geometries using deep neural networks. *AIP Conference Proceedings*, 2113: Art. 020007, ESAFORM 2019, Vittoria-Gasteiz/Spain, 2019
- [C. Zimmerling](#), C. Poppe, L. Kärger: Estimating optimum process parameters in textile draping of variable part geometries - A reinforcement learning approach. *Procedia manufacturing*, 47: p 847–854, ESAFORM 2020, Cottbus/Germany, 2020
- [C. Zimmerling](#), P. Schindler, J. Seuffert, L. Kärger: Deep neural networks as surrogate models for time-efficient manufacturing process optimisation. *PoPuPS of ULiège Library*, DOI: 10.25518/esaf orm21.3882, ESAFORM 2021, Liège/Belgium, 2021
- [C. Zimmerling](#), B. Fengler, L. Kärger: Formability Assessment of Variable Geometries using Machine Learning – Analysis of the Influence of the Database. *Key Engineering Materials*, 926: p 2247-2257, ESAFORM 2022, Braga/Portugal, 2022

## Without conference proceedings

- [C. Zimmerling](#), L. Kärger, S. Carosella, P. Middendorf, F. Henning: Zeit- und kosteneffiziente Prozess- und Produktentwicklung für den Hochleistungs-Faserverbundleichtbau unterstützt durch Techniken des Maschinellen Lernens (in German). *6. Technologietag Hybrider Leichtbau*, Stuttgart, 2019

- C. Zimmerling, B. Fengler, H. Wen, Z. Fan, L. Kärger: Rapid Determination of Suitable Reinforcement Type in Continuous-Fibre-Reinforced Composites For Multiple Load Cases. *KITopen*, KITopen-ID: 1000124624, ICCS/MECHCOMP 2020, Porto/Portugal, 2020

## Co-authored journal articles

- L. Kärger, S. Galkin, C. Zimmerling, D. Dörr, J. Linden, A. Oeckerath, K. Wolf: Forming optimisation embedded in a CAE chain to assess and enhance the structural performance of composite components. *Composite Structures*, 192: p 143-152, 2018

## Co-authored conference contributions

### With conference proceedings

- L. Kärger, S. Galkin, C. Zimmerling, D. Dörr, F.J. Schirmaier, A. Oeckerath, K. Wolf, J. Linden: Continuous CAE chain for composite design, established on an HPC system and accessible via web-based user-interfaces. *Proceedings NAFEMS World Congress*, Stockholm/Sweden, 2017.
- J. Pfrommer, C. Zimmerling, J. Liu, F. Henning, L. Kärger, J. Beyerer: Optimisation of manufacturing process parameters using deep neural networks as surrogate models. *Procedia CIRP*, 72: p 426–431, 2018
- F. Albrecht, C. Zimmerling, C. Poppe, L. Kärger, F. Henning: Development of a modular draping test bench for analysis of infiltrated woven fabrics in wet compression molding. *Key Engineering Materials*, 809: p 35–40, 2019

- D. Herz, C. Krauß, C. Zimmerling, B. Grupp, F. Gauterin: Estimation of Load-Time Curves Using Recurrent Neural Networks Based On Can Bus Signals. *WCCM-ECCOMAS2020*, Art. 1700, Paris/France (virtual), 2021

## Without conference proceedings

- L. Kärger, D. Dörr, C. Poppe, J. Seuffert, A. Bernath, S. Galkin, C. Zimmerling, F. Henning: Continuous Process Simulation for Continuous Fiber Reinforced Composites. *International VDI Conference - Simulation in Automotive Lightweight Engineering*, Amsterdam/Netherlands, 25.-26.04.2018
- D. Dörr, C. Poppe, C. Zimmerling, C. Krauß, B. Schäfer, F. Henning, L. Kärger: Advanced macroscopic modelling approaches for FE composite forming simulation using Abaqus. *Simulia Regional user meeting*, Hanau, KITopen-ID: 1000099359, 04.12.2018

## Other

- C. Zimmerling: Maschinenlernen als neues Werkzeug - Techniken des Maschinenlernens unterstützen Prozessauslegung und Gestaltung von Faserverbundbauteilen (in German). *Carbon Composites Magazin*, 01/2019: p 19-20, 2019
- C. Zimmerling, C. Poppe, L. Kärger: Virtuelle Produktentwicklung mittels Simulationsmethoden und KI (in German). *Lightweight Design* 6/2019: p 12-19, 2019.
- C. Zimmerling, C. Poppe, L. Kärger: Virtual Product Development Using Simulation Methods and AI. *Lightweight Design worldwide* 6/2019: p 13-19, 2019.



- [C. Zimmerling](#), C. Poppe, F. Albrecht, J. Fial, M. Engelfried, M. Draskovic, S. Carosella, L. Kärger, P. Middendorf, F. Henning: Einsatzmöglichkeiten des Maschinenlernens zur zeit und kosten-effizienten Prozessoptimierung, *Forschungsbrücke Abschlussympo-sium*, Karlsruhe (online presentation), 03.12.2020

## Awards

- *ThinKing-Leichtbauidee des Monats: Der virtuelle Prozessexperte*  
Leichtbau BW GmbH, Stuttgart, März 2019,  
[www.leichtbau-bw.de/fileadmin/user\\_upload/PDF/Presse/ThinKing/2019/03/ThinKing\\_Maerz\\_2019\\_V01R01.pdf](http://www.leichtbau-bw.de/fileadmin/user_upload/PDF/Presse/ThinKing/2019/03/ThinKing_Maerz_2019_V01R01.pdf), last accessed 09.09.2022,
- *Best Communication Award* (Bronze medal)  
awarded at ESAFORM 2020 in Cottbus/Germany (virtual), 2020,  
<https://esaform.org/post-2>, last accessed 09.09.2022