

Abschlussbericht zum Projekt ML@Karoprod

Vorhaben: Maschinelles Lernen zur Prognose von Prozessparametern und Bauteilqualität in der automobilen Karosserieproduktion (ML@Karoprod)

Förderprogramm: IKT 2020 – Forschung für Innovationen

Förderkennzeichen: 01IS18055

Durchgeführt von: Fraunhofer IWU
SCALE GmbH
TU Chemnitz

Projektlaufzeit: 12/2018 - 01/2022

Projektleiter: Dr. Sebastian Kriechenbauer
Dr. Sebastian Hensel
Dr. Ingolf Lepenies
Prof. Dr. Fred Hamker
Dr. habil. Julien Vitay

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Dieses Forschungs- und Entwicklungsprojekt wurde durch das Bundesministerium für Bildung und Forschung (BMBF) in der Fördermaßnahme „IKT 2020 – Forschung für Innovationen“ (Förderkennzeichen 01IS18055) gefördert und vom Projektträger Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR) betreut. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autorinnen/Autoren.

Autoren:

Dr. Sebastian Kriechenbauer

Dr. Sebastian Hensel

Dr. Ingolf Lepenies

Prof. Dr. Fred Hamker

Dr. habil. Julien Vitay

Aida Farahani

Payam Atoofi

Inhaltsverzeichnis

1	Aufgabenstellung	4
2	Voraussetzungen unter denen das Vorhaben durchgeführt wurde	6
3	Erzielte Ergebnisse	8
3.1	Fraunhofer IWU	8
3.2	SCALE	25
3.3	TU Chemnitz	39
3.4	Fazit	54
4	Wichtigste Positionen des zahlenmäßigen Nachweises	54
5	Notwendigkeit und Angemessenheit der geleisteten Projektarbeiten	55
6	Nutzen für das Unternehmen, insbesondere Verwertbarkeit des Ergebnisses	56
7	Darstellung des während des Vorhabens bekannt gewordenen Fortschritts auf diesem Gebiet bei anderen Stellen	57
8	Veröffentlichungen, Vorträge, Referate	57

1 Aufgabenstellung

Die Digitalisierung der Produktion ist aktuell eines der wichtigsten Handlungsfelder, um Wachstum und Beschäftigung am Standort Deutschland auch zukünftig zu sichern. Die Karosseriefertigung spielt dabei mit den unterschiedlichen verketteten Technologien eine wichtige Rolle und muss im Rahmen der Digitalisierung weiterentwickelt werden, um wettbewerbsfähig zu bleiben. Methoden des Maschinellen Lernens haben ein sehr großes Potenzial, um deutliche Effizienzsteigerungen zu erreichen, aber auch um neue digitale Geschäftsmodelle für KMUs wie Systemhersteller und Zulieferer zu ermöglichen.

Die Verkürzung der Entwicklungszyklen ist ein wesentliches Kriterium, um Wettbewerbsvorteile im globalen Automobilbau zu erreichen. Gleichzeitig steigt durch den wachsenden Einsatz von Mischbaukonzepten im Karosseriebau (z. B. Stahl-/Aluminiumblech) die Komplexität in der Produktion, wodurch sich die Anforderungen an die Einhaltung von Qualitätszielen signifikant erhöhen.

In der Entwicklungsphase bis hin zur Produktion von Karosseriebaugruppen werden durch die zunehmende Digitalisierung entlang des Wertschöpfungsprozesses eine Vielzahl von Daten generiert. Aus dem Entwicklungsprozess resultieren Daten von experimentellen Versuchen als auch numerischen Simulationen. Mithilfe dieser Daten sollen geeignete Prozessparameter für Fertigungsverfahren, wie z. B. die Blechumformung oder die Verbindungstechnik, ermittelt werden. Demgegenüber werden in der Produktion überwiegend Daten zur Qualitätsüberwachung der einzelnen Fertigungsprozesse gesammelt. Hieraus ergibt sich eine hohe Anzahl an zu berücksichtigenden Ein- und Ausgabegrößen zwischen den einzelnen Prozessschritten.

Gegenstand des Forschungsprojekts ist die Modellentwicklung als auch Anwendung des Maschinellen Lernens (ML) zur Beschleunigung des Planungs- und Serienanlaufs bei der Karosseriefertigung. Dabei sollen die in der Prozessentwicklungs- bzw. Produktionsanlaufphase generierten Ein- und Ausgabedaten für die Entwicklung von Prognosemodellen zur Prozessoptimierung bzw. zur Qualitätssicherung in der Produktion genutzt werden. Neben den prozess- bzw. bauteilbezogenen Kenngrößen soll zudem das fertigungstechnische Expertenwissen bei der Modellentwicklung berücksichtigt werden.

Die vier Schwerpunkte bzw. Aufgabenteilgebiete sind:

- Generierung hochqualitativer, repräsentativer und zahlenmäßig ausreichender Daten, die die Einzelprozesse und auftretenden Schwankungen in unterschiedlichen Entwicklungsstufen einer Karosseriefertigung abbilden,

- die automatisierte und strukturierte Erfassung der umfangreichen technologischen Daten aus unterschiedlichen Datenquellen mit unterschiedlichen Datendimensionen in einer Datenmanagement-Applikation,
- die Entwicklung von ML-Prognosealgorithmen, welche auf die Struktur und Menge der im automobilen Karosseriebau auftretenden Daten angepasst sind und für unterschiedliche Entwicklungsstufen eingesetzt werden können,
- die durchgängige, softwaretechnische Abbildung der Wertschöpfungskette am Beispiel eines Produktionsprozesses.

Gemeinsam verfolgen die Projektpartner die Entwicklung von effizienten Methoden für die Akquise der technologischen Daten (Fraunhofer IWU), das Datenmanagement (SCALE GmbH) und die Algorithmenentwicklung (TUC). Nur durch die kontinuierliche und interdisziplinäre Zusammenarbeit der durch die Projektpartner vertretenen Bereiche Technologie, Datenmanagement und ML-Kompetenz konnte das übergeordnete Ziel der datenbasierten Prozessparameter- und Bauteilgeometrieprognose über die gesamte Karosseriefertigungskette erreicht werden, da die Qualität und Struktur der Trainingsdaten sehr großen Einfluss auf die Prognosequalität der ML-Algorithmen besitzen.

Da im Vorhaben unterschiedliche Wissenschaftsbereich zusammenarbeiten, ergeben sich folgenden wissenschaftlichen und technologischen Projektziele.

Wissenschaftliche Ziele in Bezug auf das Forschungsgebiet Produktionstechnik:

- Erfassung aller relevanten technologischen Informationen der Prozesse
- Aufbau von numerischen Modellen für die ausreichend genaue Abbildung der betrachteten Technologien durch FEM-Simulationen
- Abbildung der indirekten und direkten technologischen Wechselwirkungen zwischen Ein- und Ausgabegrößen (z. B. schwankende Halbzeugeigenschaften, Maschinenparameter, definierte Baugruppeneigenschaften)
- Analyse des Einflusses unscharfer (z. B. Chargen-abhängige Blechdicken) bzw. nicht direkt messbarer Größen (z. B. Reibbeiwerte während des Fügeprozesses) auf den jeweiligen Einzel- als auch auf den Gesamtprozess

Wissenschaftliche Ziele in Bezug auf das Forschungsgebiet Maschinelles Lernen:

- Anpassung der Lernalgorithmen auf die Datenstruktur- und Menge der verfügbaren technologischen Informationen aus der Karosseriefertigung
- Fusionierung von technologischen Lerndaten aus verschiedenen Quellen wie Experiment und FEM-Simulation
- Entwicklung von Methoden für das Lernen aus Datensätzen mit geringerer Anzahl bzw. mit unterschiedlichem Informationsgehalt

- Übertragung von gelernten Zusammenhängen auf unterschiedliche Entwicklungsphasen der Karosseriefertigung (Entwicklungsphase, Produktionsanlauf)

Technologische Ziele in Bezug auf die Karosseriefertigung:

- Deutliche Dynamisierung der Entwicklungszyklen von Karosseriebaugruppen
- Nutzung von Daten- und Prognosemodellen der Entwicklungsphase für datengetriebene Modelle zur Qualitätssicherung in der Produktion und somit Steigerung der Produktivität durch die Reduzierung von Ausfallzeiten sowie Produktionsausschuss

Nutzung von datenbasierten Prozessmodellen für technologische Optimierungen der Einzelprozesse, wie z. B. der Planung von neuen Möglichkeiten der adaptiven Prozessregelung

2 Voraussetzungen unter denen das Vorhaben durchgeführt wurde

Die Antragssteller sind bereits seit längerer Zeit in jeweils spezifischen Themengebieten tätig, sodass umfangreichende Expertise im Verbundprojekt bestand:

Fraunhofer IWU

Die Projektinhalte in Bezug auf die technologischen Untersuchungen der Prozesskette wurden von den Abteilungen Prozesskettenanalyse und –regelung sowie Mechanisches Fügen des Fraunhofer IWU in Dresden übernommen. Die Abteilungen haben sich in den letzten Jahren eine umfangreiche Expertise bezüglich der experimentellen sowie numerischen Abbildung der Prozesskette für die Herstellung von Karosseriebaugruppen mit Fokus auf die Maßhaltigkeit erarbeitet [ACK17, ECK12]. Im Rahmen unterschiedlicher industrieller und öffentlicher Projekte konnte dabei das Know-how erarbeitet und das Equipment für die Fertigung von automobilen Baugruppen sowie die entsprechende messtechnische Analyse beschafft werden. Ein weiterer Schwerpunkt der Abteilung sind numerische Sensitivitätsanalysen von Einzelprozessen aus dem Karosseriebau wie dem Umformen [KRI16], Spannen [SCH17] und Fügen [JÄC16]. Diese Kompetenzen sind essenziell für die Gewinnung hochqualitativer technologischer Daten.

SCALE

SCALE bietet umfassende Serviceleistungen bei der Softwareentwicklung sowie bei der Integration von Datenmanagementsystemen für die virtuelle und versuchsgestützte Produktentwicklung. Grundlage für die Entwicklung und den Betrieb von CAx-Datenmanagementlösungen ist ein Software-Framework mit Anwendungsmodulen, welche im Auftrag des Kunden individuell angepasst werden. SCALE ist durch seine kundennahen

Niederlassungen mit Fachabteilungen unterschiedlicher Akteure aus dem Automotive-Umfeld vernetzt und Entwicklungsprozessen und Methoden vertraut. Derzeit beschäftigt SCALE in Deutschland ca. 40 Mitarbeiter, eine ausgewogene Mischung aus Ingenieuren mit CAx-Erfahrung und Informatikern. Durch seine wissenschaftliche Ausrichtung war SCALE als KMU in den vergangenen Jahren bereits an einigen Forschungsprojekten beteiligt, Beispiele sind C³, VAVID und EEGO.

TU Chemnitz (TUC)

Die Professur für Künstliche Intelligenz deckt in Forschung und Lehre Themen der KI, Bildverstehen, Maschinelles Lernen, Robotik und Neurokognition ab. Der Leiter, Prof. Dr. Fred Hamker, wurde im Projekt zusätzlich durch einen Nachwuchswissenschaftler, PD Dr. Vitay unterstützt. Beide konnten aktiv im Projekt mitarbeiten, ihre Kompetenz einbringen und die wissenschaftlichen Mitarbeiter betreuen. Zwei Doktoranden, Aida Farahani und Payam Atoofi, waren während des gesamten Projekts beteiligt und entwickelten neue Ansätze für geometrisches Deep Learning.

3 Erzielte Ergebnisse

Zur übersichtlichen Zuordnung der Ergebnisse, werden diese für jeden Verbundpartner separat vorgestellt.

3.1 Fraunhofer IWU

Definition der Fertigungskette, Konzeptionierung

In einem ersten konstituierenden Schritt wurden gemeinsam mit den Verbundpartnern Scale und TU Chemnitz die zu untersuchende Baugruppe, Materialien, Fügeverfahren, aufzunehmende/auszuwertende Größen sowie erste Fertigungsparameter festgelegt. Die vor Projektbeginn avisierte Baugruppe weist große laterale Abmessungen und einen hohen Komplexitätsgrad auf. Daher wurde mit den Projektpartnern eine Geometrievereinfachung vereinbart, die eine verbesserte Datenlogistik (Elementanzahl) und –standardisierung (Geometriekomplexität) sowie eine geringere Rechenzeit in der Simulation und damit verbessertes Datenhandling verspricht. Abbildung 2 stellt die Fertigungskette samt Baugruppeneinzelteilen dar. Trotz des erhöhten Aufwandes am IWU infolge CAD-Planung der Einzelteile, Tiefzieh- und Beschnittwerkzeuge sowie Spann-, Füge-, und Messvorrichtungen und deren Teilefertigung mit Zusammenbau im Weiteren ergibt sich durch dieses Vorgehen die Möglichkeit die Prozesskette mit allen Schritten sehr projektspezifisch auszulegen und die Datenschnittstellen im Vorfeld zu definieren. Bei der Neuplanung der Prozesskette fand die Integration eines zusätzlichen Prozessschritts für den Beschnitt der Einzelteile statt, um die Fertigungskette näher an den Serienprozess zu rücken. Zudem konnte durch die Neukonstruktion der Fügeprozess nun teilautomatisiert ausgelegt werden, was Fehlereinflüsse infolge manueller Arbeiten reduziert.

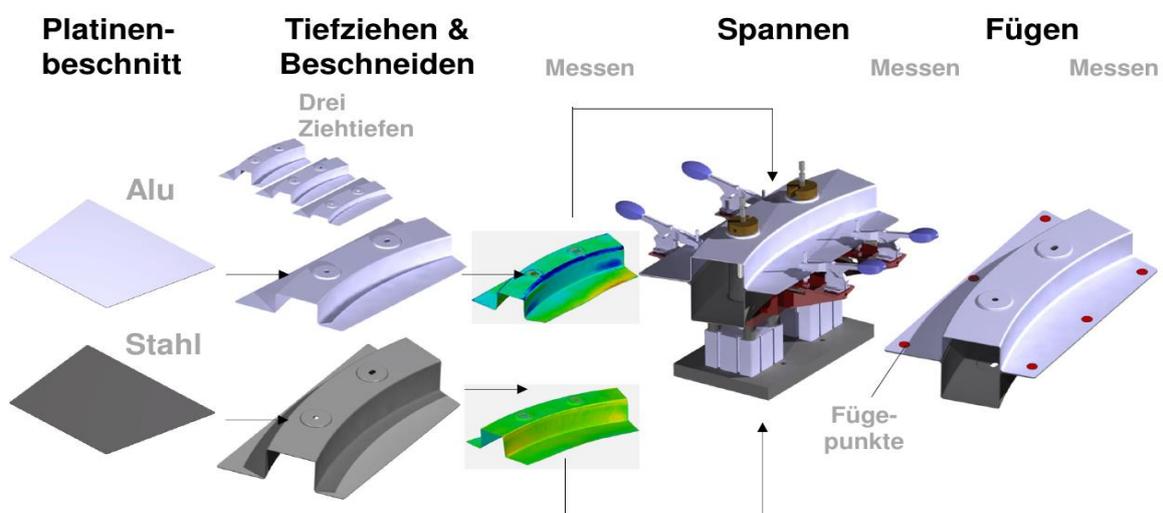


Abbildung 2: Prozess der Baugruppenherstellung

Die hutprofilförmige gebogene Baugruppe besteht aus einem oberen und einem unteren U-Profil, die durch sechs Fügestellen miteinander verbunden sind. Die Baugruppe ist in Mischbauweise (Aluminium-Stahl) ausgeführt. Artgleiche Materialien lassen sich aber ebenfalls verwenden. Nach dem Beschnitt der Platinen erfolgt das Tiefziehen und danach der Beschnitt der umgeformten Einzelteile im Beschnittwerkzeug. In diesem Schritt werden für die spätere definierte Lagerung in der Mess- und Spannvorrichtung ebenfalls Positionierungslöcher eingebracht. Das Lagerungskonzept ist in Anlehnung an die Lagerungskonzepte in der automobilen Prozesskette als RPS-Systematik (RPS entspricht Referenzpunktsystem) mit zueinander orientiertem Rund- und Langloch ausgeführt. Für das Fügen werden die Einzelteile in der Spannvorrichtung positioniert, mit Kniehebelspannern gespannt und mittels C-Bügel-Fügezange mechanisch mit Clinchen oder Halbhohlstanznieten gefügt. Jeweils nach dem Beschnitt, dem Spannen und dem Fügen erfolgt die berührungslose Vermessung der Einzelteil-/ Baugruppengeometrien mittels GOM Atos. Ursprünglich war ein finaler thermischer Prozessschritt angedacht, dessen Realisierung zugunsten einer Prozesskette mit wohldefinierten Abhängigkeiten zurückgestellt wurde, da hiermit üblicherweise zusätzliche Streuungen in den Prozess integriert werden. Abbildung 3 stellt das CAD-Modell von Tiefzieh- und Beschnittwerkzeug und Abbildung 4 das der Spann-/Fügevorrichtung dar. Im Arbeitspaket erfolgte zudem die Beschaffung der Blechwerkstoffe für die nachfolgenden experimentellen Arbeiten des AP 2.1. Es konnten die Platinenwerkstoffe DC01 (1.0 und 1.2 mm Blechdicke), DC04 (1.0 mm), HC340LA (1.0 mm) sowie EN AW-5083 (1.0 und 1.5 mm) im Projekt bereitgestellt werden. Abbildung 5 stellt die Messvorrichtungen für die Einzelteile einerseits und für die Zusammenbaugruppe andererseits dar.

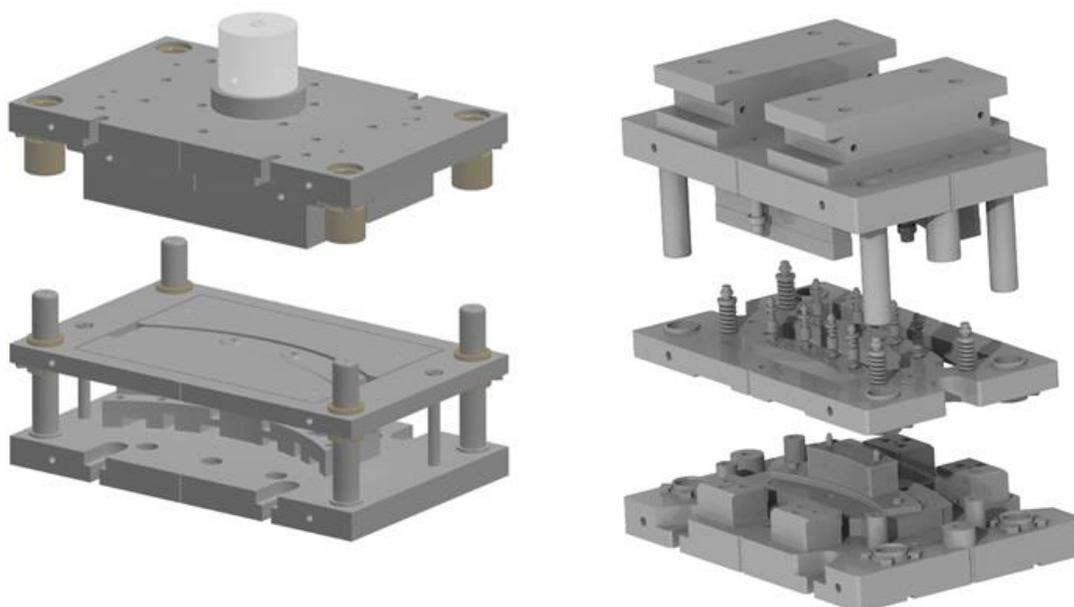


Abbildung 3: Im Projekt entwickelte Werkzeuge mit integrierten Sensoren und RPS-Systematik: Tiefziehwerkzeug (links), Beschnittwerkzeug (rechts).

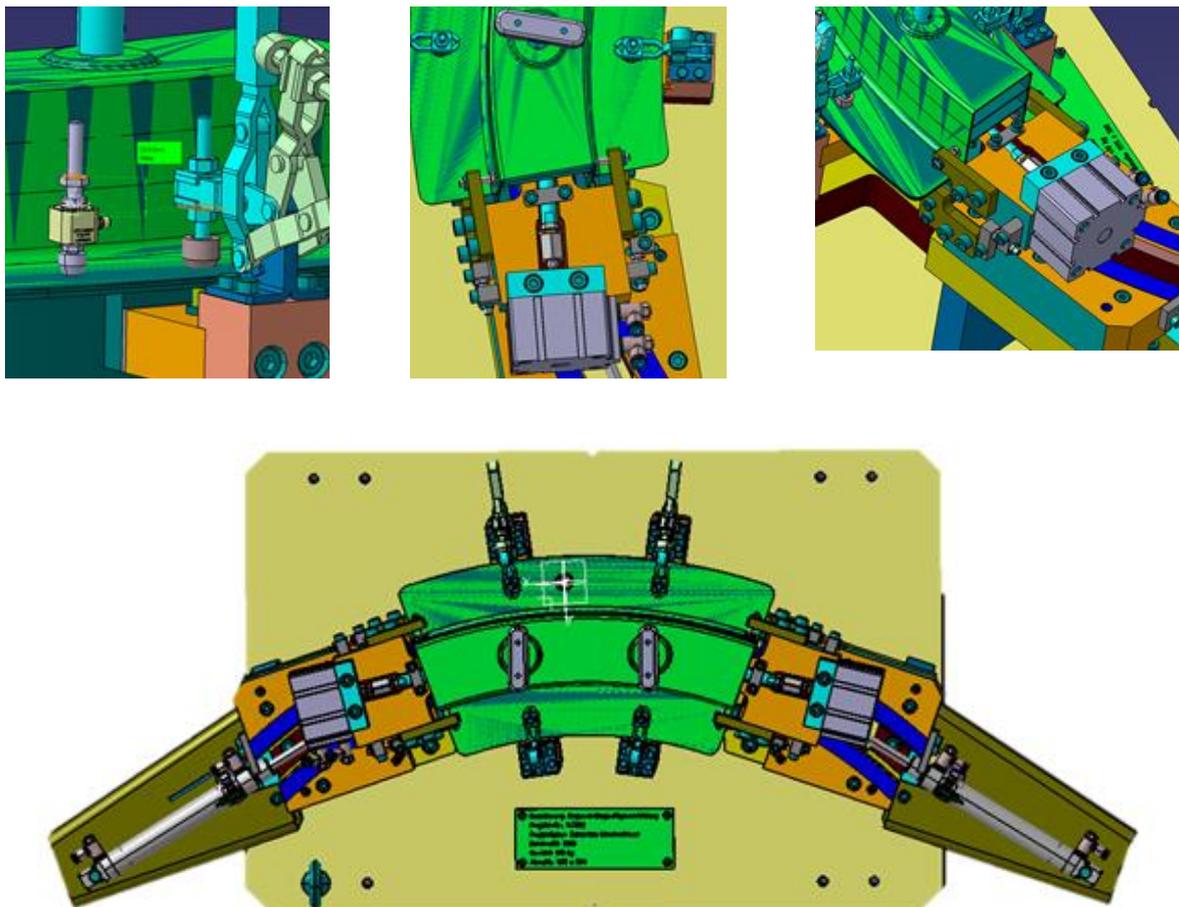


Abbildung 4: Spann- und Fügwerkzeug mit RPS-Systematik: Detailansichten (oben), Gesamtansicht von oben (unten)

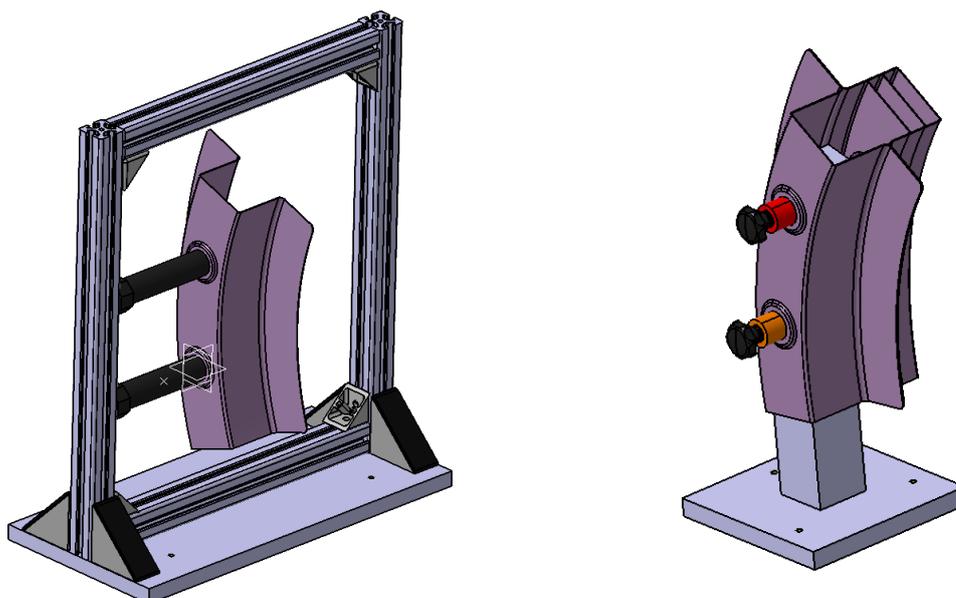


Abbildung 5: Messvorrichtungen mit RPS-Systematik: Einzelteilvermessung (links), Baugruppenvermessung (rechts)

Versuchsplanung und -durchführung

Nach der Planung und Beschaffung der Einzelteile für die Werkzeuge und Vorrichtungen konnten diese verbaut und die Einzelprozesse in ersten Versuchen eingefahren werden (s. Abbildung 6).

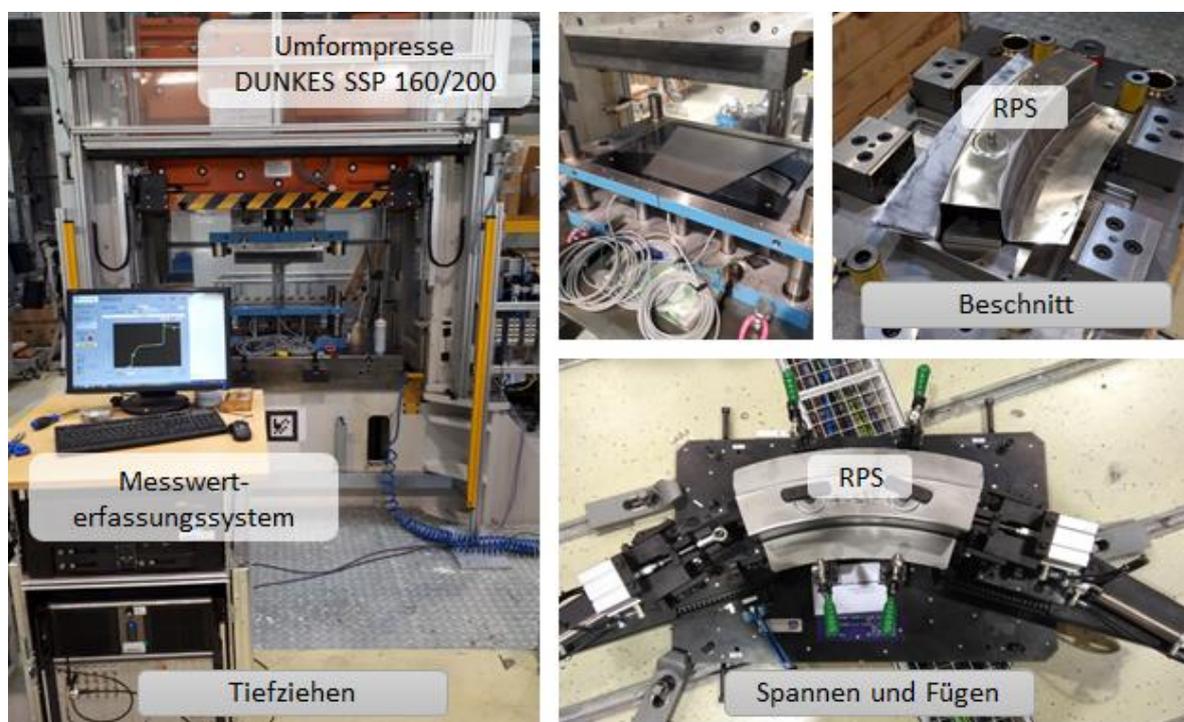


Abbildung 6: Einarbeitung der Werkzeuge.

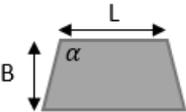
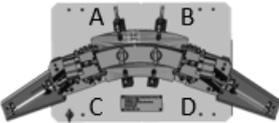
Bei der Konzeptionierung der Werkzeuge und des Messkonzepts wurde die spätere Variierung von Prozessparametern berücksichtigt. In Vorplanungen wurden die in Tabelle 1 zusammengestellten Parameter mit ihrer Schwankungsbreite und Stufung festgelegt.

Als Platinenform vor dem Tiefziehen wurde die Trapezform gewählt, da die Einzelteile infolge konkaver und konvexer Krümmung ebenfalls eine längere und eine kürzere Seite aufweisen. Im Rahmen der Werkzeugeinarbeitung erfolgte die systematische Variierung von Länge, Breite und Innenwinkel der Trapezform, um eine gegenüber der Riss- und Faltenbildungsneigung robuste Grundform zu ermitteln. Es wurden eine Breite von 350 mm, eine Länge von 214 mm und ein Innenwinkel von 108° als robuste Abmaße für die Ausgangsplatine ermittelt.

Für die spätere Simulation der Einzelprozesse werden Materialparameter benötigt. Daher wurden für die sechs Blechmaterial-/Blechkicken-Kombination jeweils acht taillierte Zugproben in Walzrichtung, jeweils vier in 45° -Richtung zur Walzrichtung und vier in 90° -Richtung nach DIN EN ISO 6892-1 angefertigt und im Zugversuch geprüft, um E-Moduli, Fließkurven und charakteristische Punkte der Fließkurve (Streckgrenze, Gleichmaßdehnung, Anisotropiekennwerte) aufzunehmen. Weiterhin fand die Anfertigung und Prüfung von jeweils

fünf Rundproben pro Blechmaterial-/Blehdicken-Kombination im Stauchversuch nach DIN 50106 statt, um ebenfalls das Materialverhalten unter Druckbelastung zu charakterisieren. Mit den Ergebnissen konnten walzrichtungsabhängige Materialmodelle extrahiert werden. Es kam dabei die Extrapolation der Fließkurven nach Voce zum Einsatz. Für die Auswertung der insgesamt 126 Materialgrundversuche konnte ein im Projekt erstelltes Skript genutzt werden, um eine automatisierte Materialmodell-Parameteridentifikation durchzuführen.

Tabelle 1: Parametermatrix

	Parameter	Variationsbereich	Einheit	Prozessgrößen
1 Beschnitt der Platine 	Breite B	250; 275; 300; 325	mm	
	Länge L	50; 60; ...; 300	mm	
	Winkel α	90; 91; ...; 110	°	
	Blechdicke	DC01 (1,00; 1,25) DC04	mm	
	Material <i>nur als Kombination verfügbar</i>	(1,00) AA5083 (1,00; 1,50) HC340LA (1,00)		
2 Tiefziehen	Niederhalterkraft	10; 20; ...; 500	kN	Stempelkraft
	Stempelgeometrie (Ziehspalt)	0,8 (S1) / 1,6 (S2) / 2,4 (S3)	mm	
	Matrizengeometrie (Ziehtiefe)	30 (T3) / 50 (T2) / 70 (T1)	mm	Ziehkissenkraft
	Einlegeposition DLR	-5; -4; ...; 5	mm	Stößelkraft Stößelweg
3 Bauteilbeschnitt				
4 Spannen 	A Position Z-Richtung	-5; -4; ...; 5	mm	
	B Position Z-Richtung	-5; -4; ...; 5	mm	
	CD Position Z-Richtung	-5; -4; ...; 5	mm	
	Spannzustand A, B, C, D	AUF / ZU		
5 Fügen 5.1 HSN	<i>(nur als bestimmte Paarungen ffügbar*)</i>			
	Niederhalterkraft	5; 6; ...; 15	kN	Fügekraft
	Stempelposition	0; 0,1; ...; 2	mm	Hinterschnitt
	Fügereihenfolge			
	Anzahl Fügepunkte			
5.2 Clinchen	Niederhalterkraft	5; 6; ...; 15	kN	Hinterschnitt
	Stempelposition	0; 0,1; ...; 2	mm	Bodendicke
	Fügereihenfolge			Halsdicke
	Anzahl Fügepunkte			

Für den letzten Prozessschritt Mechanisches Fügen im Experiment musste eine Bemusterung für die Fügeverbindungen durchgeführt werden, um eine qualitätsgerechte und industriennahe Verbindung zu erzeugen. Die Bemusterung orientiert sich immer an einer konkreten Blechmaterialpaarung (Kombination zweier Bleche in Material, Dicke und Fügerichtung).

Somit müssen für die im Projekt geplanten 2 Fügeverfahren Clinchen und Halbhohlstanznieten und 12 Blechmaterial-Dicken-Fügerichtung--Paarungen sowie 2 Fügerichtungen $2 \cdot 12 \cdot 2 = 48$ Bemusterungen mit jeweils drei Einzelversuchen zur Absicherung der Wiederholgenauigkeit ausgeführt werden. In der Bemusterung werden Füge- und Niederhalterkräfte, Stempelformen und -durchmesser, Matrizenformen und -durchmesser sowie Stempelwege festgelegt. Für die Startpunkte der Auslegung konnte auf die Expertise des IWU zurückgegriffen werden. Die Qualität der Verbindung wird mittels angefertigter Schliffbilder sowie dem Ausmessen und Evaluieren der sich einstellenden Fügepunktgeometrie beurteilt (s. exemplarische Schliffbildervermessung Abbildung 7). Nicht alle Paarungen lassen sich mit den gewählten Verfahren prozesssicher fügen. Beim Halbhohlstanznieten war es möglich 14 der 24 Paarungen und beim Clinchen 10 der 24 Paarungen fehlerfrei zu verbinden. In der Schnittmenge dieser beiden Mengen befanden sich sechs mögliche Paarungen, die für beide Fügeverfahren und –richtungen prozesssicher gesetzt werden können. Für diese erfolgte abschließend die Hauptbemusterung auf einer am Roboter befestigten industriellen Fügeanlage (Fügezange Fa. Eckold), da die Vorbemusterung auf einer feststehenden Fügeanlage ausgeführt wurde (s. Tabellen 2 und 3).

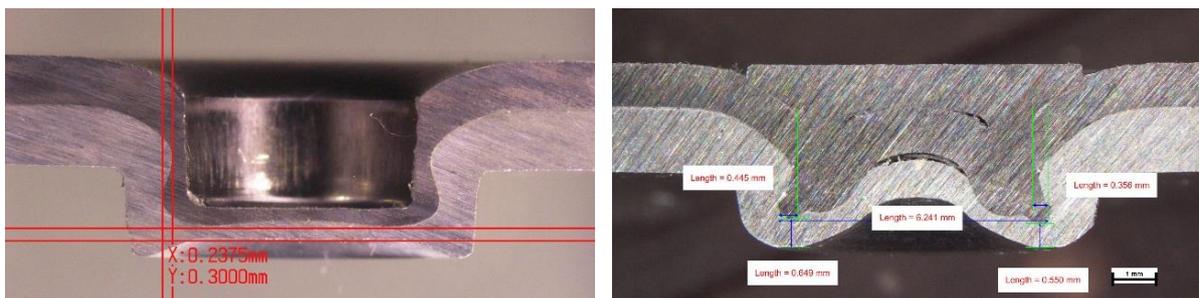


Abbildung 7: Schliffbildervermessung: Clinchen (links), Halbhohlstanznieten (rechts).

Tabelle 2: Hauptbemusterung der Fügepaarungen

Verbindung	Oberblech		Unterblech			Clinche $\overleftarrow{\text{T}}$	HSN $\overleftarrow{\text{T}}$
2	DC01	1,00	EN AW-5083	1,00	2,00	x	x
12	EN AW-5083	1,00	HC340LA	1,00	2,25	x	x
14	DC01	1,00	EN AW-5083	1,50	2,50	x	x
15	DC01	1,25	EN AW-5083	1,50	2,75	x	x
16	DC04	1,00	EN AW-5083	1,50	2,50	x	x
17	HC340LA	1,00	EN AW-5083	1,50	2,10	x	x
22	EN AW-5083	1,50	DC04	1,00	2,50	x	x

Tabelle 3: Hauptbemusterung der Fügepaarungen: Aktivteilbezeichnungen Stempel / Matrize

	Fügeteile								Prozessparameter		
	Oberblech				Unterblech				Variante A (Stempel A)		
	Material ID	Werkstoff	Charge/Fach	Blechdicke	Material ID	Werkstoff	Charge/Fach	Blechdicke	Stempel	Matrize	Fügekraft
C	A	DC01	625/7	1.0	F	AA5083	630/33	1.0	TOX 10.180.160627	TOX_14.25.258087	54kN/ 180bar
	I	F	AA5083	630/33	1.0	D	HC340LA	628/15	1.0	TOX 10.180.154501	TOX_14.25160BD8014
n	A	DC01	625/7	1.0	E	AA5083	629/33	1.5	TOX 10.180.228806	TOX_14.25.258087	54kN/ 180bar
	c	B	DC01	626/7	1.2	E	AA5083	629/33	1.5	TOX 10.180.228806	TOX_14.25.258087
h	C	DC04	627/7	1.0	E	AA5083	629/33	1.5	TOX 10.180.228806	TOX_14.25.258087	54kN/ 180bar
	e	D	HC340LA	628/15	1.0	E	AA5083	629/33	1.5	TOX 10.180.156547	TOX_14.25.258087
n	E	AA5083	629/33	1.5	C	DC04	627/7	1.0	TOX 10.180.160627	TOX_14.25.258087	56kN/ 190bar
H	A	DC01	625/7	1.0	F	AA5083	630/33	1.0	Henrob C5,3x4,0	DO.082.10.013	52kN
	S	F	AA5083	630/33	1.0	D	HC340LA	628/15	1.0	Henrob C5,3x4,0	DO.082.10.013
N	A	DC01	625/7	1.0	E	AA5083	629/33	1.5	Henrob C5,3x4,0	DO.082.10.013	54kN
	S	B	DC01	626/7	1.2	E	AA5083	629/33	1.5	Henrob C5,3x4,5	DO.082.10.013
N	C	DC04	627/7	1.0	E	AA5083	629/33	1.5	Henrob C5,3x4,5	DO.082.10.013	58kN
	D	HC340LA	628/15	1.0	E	AA5083	629/33	1.5	Henrob C5,3x4,0	DO.082.10.013	54kN
	E	AA5083	629/33	1.5	C	DC04	627/7	1.0	Henrob C5,3x5,0	DO.087.18.042	56kN

Parallel erfolgte der Modellaufbau für die Einzelschrittsimulationen Tiefziehen, Rückfederung I, Beschnitt, Einlegen und Spannen, Fügen, Rückfederung II. Da die Modellerstellung für die Einzelprozesse mit variierenden Parametern für spätere Arbeiten weitestgehend automatisch erfolgen muss, entstand in diesem Arbeitspaket ein Programm (Eigenname DynamoPre, Weiterentwicklung aus (Kriechenbauer 2022)), mit dem unter Zuhilfenahme auch weiterer LS-Prepost-Makros ein vollständiges LS-Dyna-Inputdeck automatisiert erstellt werden kann. Die Modellerstellung wird dann für jede Parameterkonfiguration erneut via LS-Prepost unter Nutzung des Makros ausgeführt. Teil des Skripts ist ebenso die Berücksichtigung der sich verändernden kinematischen Modellrandbedingungen, Materialzuordnungen und Geometrikombinationen, die durch die Variation der Prozessparameter entstehen. Die Modelldaten werden dann in einer Ordnerumgebung abgelegt und mit LS-Dyna berechnet. Ein weiteres Programm (Eigenname DynamoPost) entstand, welches die nach der Berechnung vorliegenden Daten aus den LS-Dyna-Ergebnisdateien zusammenstellt, extrahiert und in Dateien mit einheitlich definiertem Format archiviert und damit für die Weiterverarbeitung speichert. Insbesondere bei der Schnittstellendefinition, der Datenstandardisierung und der Erstellung der Softwareumgebung unterstützte der Projektpartner Scale mit Expertise. Das Gesamtskript Dynamo basiert auf der lizenzfreien Sprache Python und dessen Bibliotheken.

Erste vergleichende Einzelschrittsimulationen dienten der Validierung der Simulationsmodelle mit den experimentellen Ergebnissen (erste Topologievermessungen mit GOM Atos). Die Prozessschrittmodelle wurden unter spezieller Maßgabe der Daten- und Rechenzeitreduktion aufgebaut, um die Handhabbarkeit bei der großen Anzahl der geplanten Variationen abzusichern. Die Gesamtprozesskette untergliedert sich in zwei wesentliche Bereiche: Die Einzelteilfertigungs- und die Zusammenbausimulation. Das Modell für das Tiefziehen (s. Abbildung 8) als erster Schritt in der Einzelteilfertigungssimulation erfolgte mit dem expliziten Solver von LS-Dyna. Für Werkzeuge und Blech wurde die Halbsymmetrie und für das Blech zusätzlich adaptive Neuvernetzung genutzt, um die Berechnungszeit zu senken. Die Neuvernetzung bereitete bei der späteren Datenverarbeitung für ML Schwierigkeiten, da über

die verschiedenen Designs variierende Anzahlen und Nummerierungen der Knoten und Elemente entstehen. Daher mussten zuverlässige Mapping-Methoden von der Firma SCALE entwickelt werden.

In Abbildung 8 ist die trapezförmige Platine gelb, die Matrize transparent im linken Bild (im rechten Bild ausgeblendet), der Niederhalter blau und der feststehende Stempel rot dargestellt. Alle folgenden Simulationen entlang der gesamten Prozesskette werden mit dem impliziten Solver von LS-Dyna berechnet.

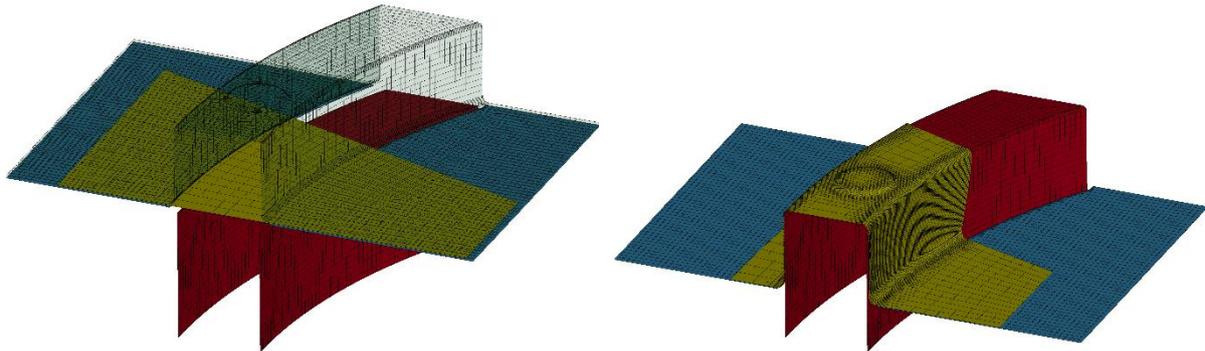


Abbildung 8: Tiefziehprozessmodell: Prozessstart (links), Prozessende (rechts)

An die Tiefziehsimulation schließen sich in zwei weiteren Simulationen der Beschnitt und die Rückfederung an. Abbildung 9 zeigt diese in Form der v.Mises-Vergleichsspannung. Die Halbsymmetrie wurde für die Visualisierung durch Spiegelung der Ergebnisse wieder in das Komplettbauteil überführt. Die aus der letzten Rückfederungssimulation ausgelesenen und weiter in die Zusammenbausimulation übergebenen Daten enthalten ebenfalls zusätzlich die gespiegelten Daten, da nach der Einzelteilsimulation die Halbsymmetrie durch unsymmetrische Randbedingungen verletzt wird.

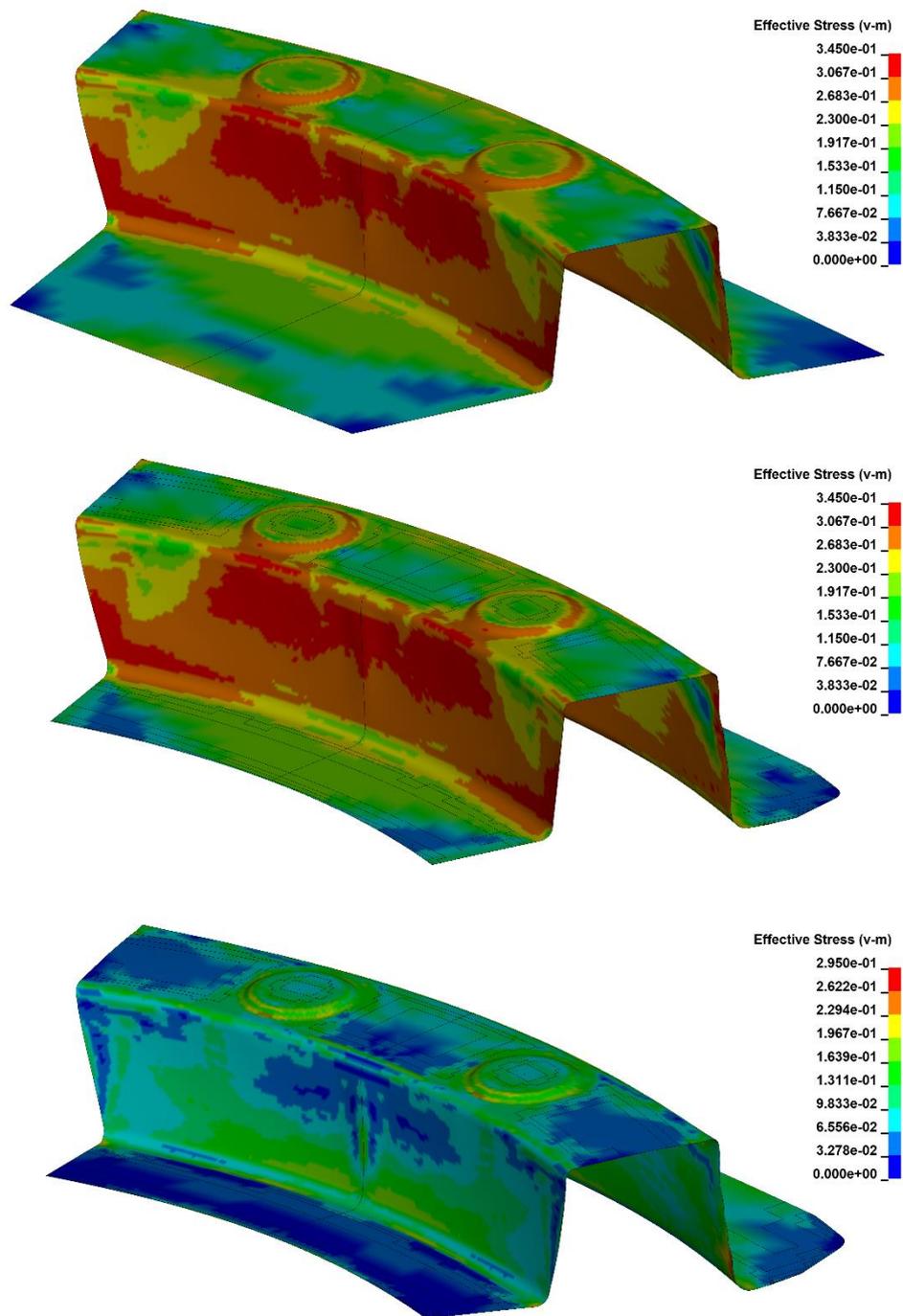


Abbildung 9: Kette für die Einzelteilfertigungssimulation, v. Mises-Vergleichsspannung Tiefziehen (oben, Max. 345 MPa), Beschnitt (mittig, Maximum 345 MPa), Rückfederung (unten, Maximum 295 MPa)

Die gesamte Zusammenbausimulation untergliedert sich in vier Spannsimulationen, das Fügen mit sechs Untersimulationen und das abschließende Rückfedern. In einem ersten Schritt, dem ersten Spannen, erfolgt das Einpressen des Unterblechs in die Anlageflächen entsprechend Werkzeuganlagengeometrie im Experiment (s. Abbildung 10). Die unteren runden Spannerflächen sind in der ersten Spannsimulation schon enthalten. Die zweite Spannsimulation positioniert das Oberblech auf dem eingepressten Unterblech und schließt die oberen runden Spannerflächen. Hiermit wird das Oberblech entsprechend gewählter

Spannerpositionierung des Versuchsplans auf das Unterblech gepresst (s. Abbildung 11). Infolge zum Teil extrem gegenläufiger Parameterkonfigurationen bestehend aus Spannerhöhenpositionen, Ziehtiefen, Material und Einzelteilgeometrien nach dem Tiefziehen können extreme Paarungen aus Ober- und Unterblech entstehen, die zum Teil lange Rechenzeiten von bis zu 8 Stunden pro Rechnung auf 8 Prozessoren verursachen und in 10 % der Fälle zum vorzeitigen Abbruch der Berechnung durch Nichtkonvergieren führen. Die langen Berechnungszeiten entstehen durch teilweise geringere Berechnungsstabilität einerseits und die großen Spalte, die mit konstanten Spannervorgangsgeschwindigkeiten geschlossen werden andererseits. Die Bandbreite an Spaltmaßen schuf die Notwendigkeit über ein Skript das jeweilige Spaltmaß exakt vor der Berechnung auszumessen und den Verfahrensweg für die Spannflächen entsprechend anzupassen.

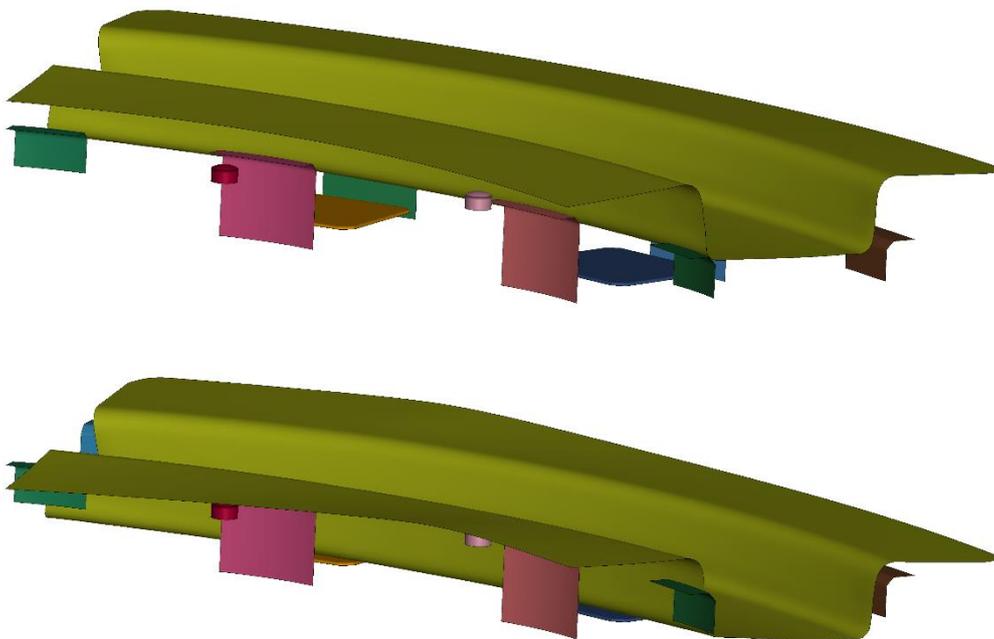


Abbildung 10: Spannsimulation Unterblech mit runden unteren Spannerflächen und weiteren Anlageflächen: Prozessbeginn (oben), Prozessende (unten)

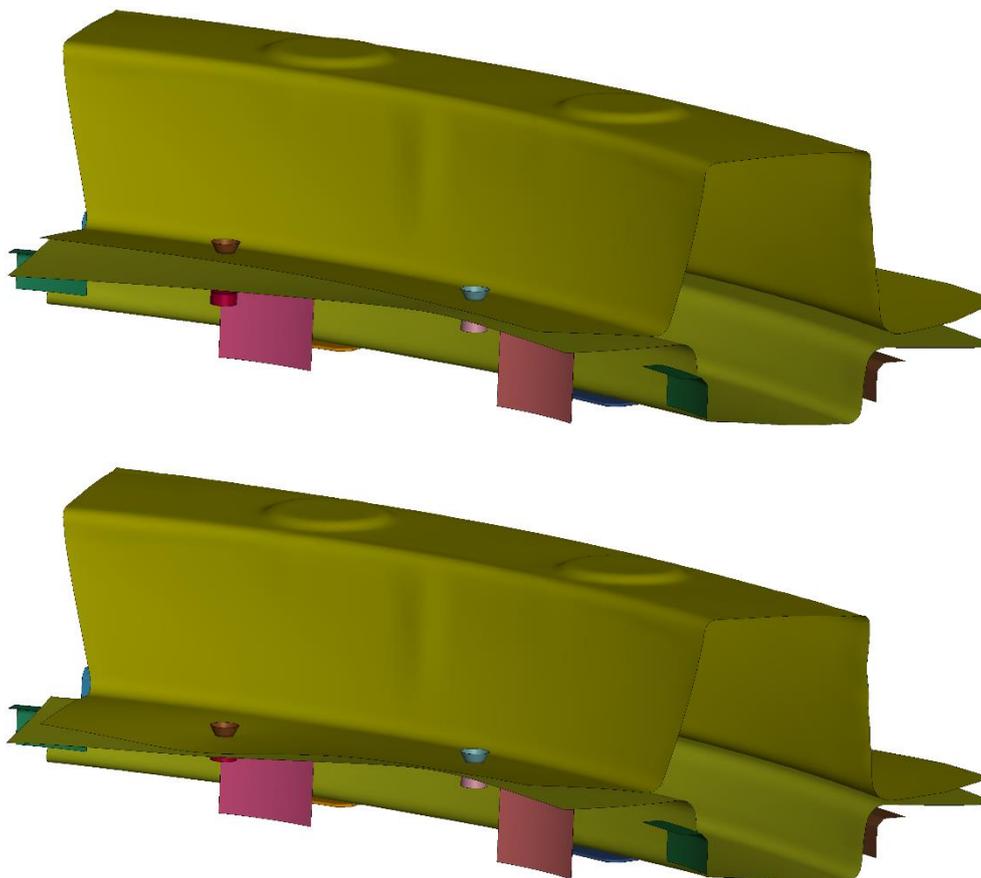


Abbildung 11: Spannsimulation Unter- und Oberblech mit allen runden Spannerflächen und den weiteren Anlageflächen: Prozessbeginn (oben), Prozessende (unten)

Nach den Spannsimulationen schließt sich die Fügungssimulation an. Für die Nachbildung der Fügungswirkung, also der lokal-punktuellen Verbindung beider Bleche miteinander, mussten diverse Methoden in LS-Dyna getestet werden, um eine hinsichtlich Berechnungszeit, Komplexitätsgrad und Konvergenz optimale Ersatzmodellierung für die Fügung aufzufinden. Als am geeignetsten erwies sich eine Beam-Elementformulierung in Verbindung mit *CONTACT_SPOTWELD. Für jeden der sechs separaten Fügungspunkte wurde entsprechend der in der Bemusterung festgelegten Fügungspunktreihenfolge ein Fügungsspanner (Niederhalter) geschlossen und die Ersatzmodellierung der Fügung in einer separaten Simulation aufbauend auf der Vorgängersimulation eingebracht. Nach jedem Einbringen von jedem der sechs Fügungspunkte erfolgt entsprechend Experiment jeweils abermals eine Rückfederungssimulation. Nach dem Fügen folgt das finale Rückfedern. Die punktuellen Verbindungen zwischen Ober- und Unterblech zusammen mit den Deformationen, die durch das Zusammenpressen der vorverformten Einzelteile eingebracht wurden, sorgen für eine sehr individuelle Entspannung der Baugruppe in der Rückfederungssimulation und damit Baugruppenverformung. Es kommt zu Verdrillungen, lokaler Taschenbildung (Spalt zwischen Ober- und Unterblech nach der Rückfederung) und Verkippungen. Hier führen extreme Kombinationen ebenfalls zu langen Berechnungszeiten und einem Ausfall von 15 % der

Berechnungen. Beim Schließen der Fügspanner kam abermals das Skript zum Ausmessen von Spaltmaßen zwischen den Blechen zum Einsatz, um die Berechnungszeiten auf ein Minimum zu reduzieren. Abbildung 12 zeigt das Ergebnis der Rückfederungssimulation nach dem Fügen am Beispiel des Zusammenbaudesigns Nr. 0487 (laufende Nummer) von Abbildung 9 bestehend aus den Einzelteilen Nr. 0945 (Oberblech, Abbildungen 8 und 9) und Nr. 0975 (Unterblech, nur in Abbildung 10 und 11 dargestellt) in Form der v.Mises-Vergleichsspannungsverteilung.

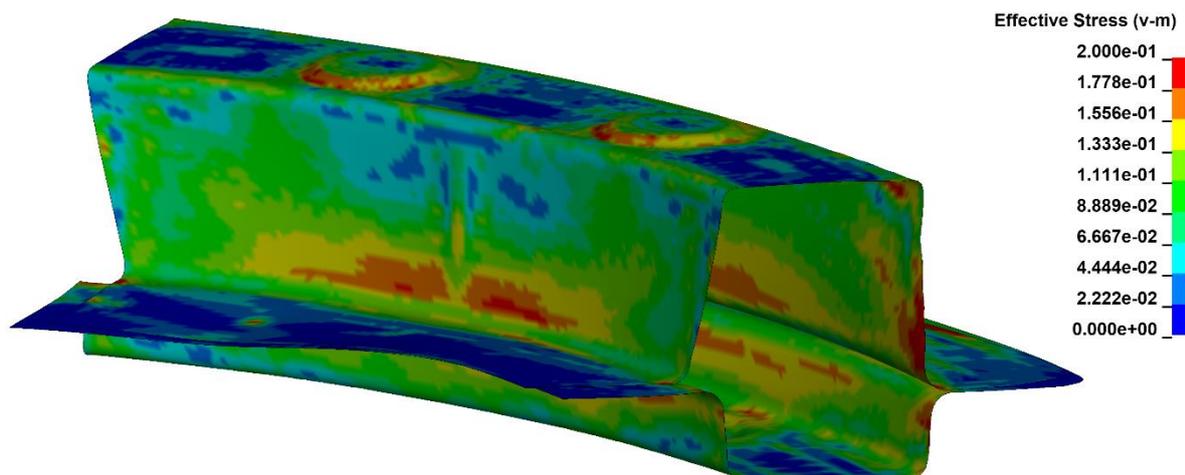


Abbildung 12: V.Mises-Vergleichsspannung nach Fügen und Rückfederung

Datenakquise Gesamtprozesskette

In der experimentellen Prozesskette wurden nach dem Einarbeiten der Werkzeuge und Einfahren der Prozesse alle 160 hergestellten Einzelteilen mit GOM Atos vermessen. Hierbei erfolgte die Einspannung der Proben in die Einzelteilmessvorrichtung, um die wiederholgenaue Positionierung im CAD-Koordinatensystem zu gewährleisten (RPS-Systematik). Die Bauteile weichen entsprechend der jeweiligen Parameterkonfiguration von der CAD-Geometrie wie geplant ausreichend ab und liefern somit eine große Anzahl variierender Einzelteilgeometrien für den späteren Zusammenbau. Abbildung 13 zeigt für neun Designs die Abweichung von der CAD-Geometrie. Die Einzelteile wurden beidseitig vermessen, sodass jeweils eine Hüllgeometrie aus der Messung resultiert. Aus diesen Hüllgeometrien wurde mittels eigens programmierten Skripts die Blechdicke auf dem Bauteil extrahiert, um Blechausdünnung bzw. -aufdickung infolge Tiefziehens zu bewerten. Abbildung 14 zeigt die Blechdickenverteilung wiederum für weitere neun Designs mit unterschiedlichen Herstellungsparametern. Maximum und Minimum der Skalen wurden entsprechend der untersuchten Ausgangsblechdicken (1.0 mm, 1.2 mm und 1.5 mm) gesetzt (Nominaldicke ± 0.05 mm).

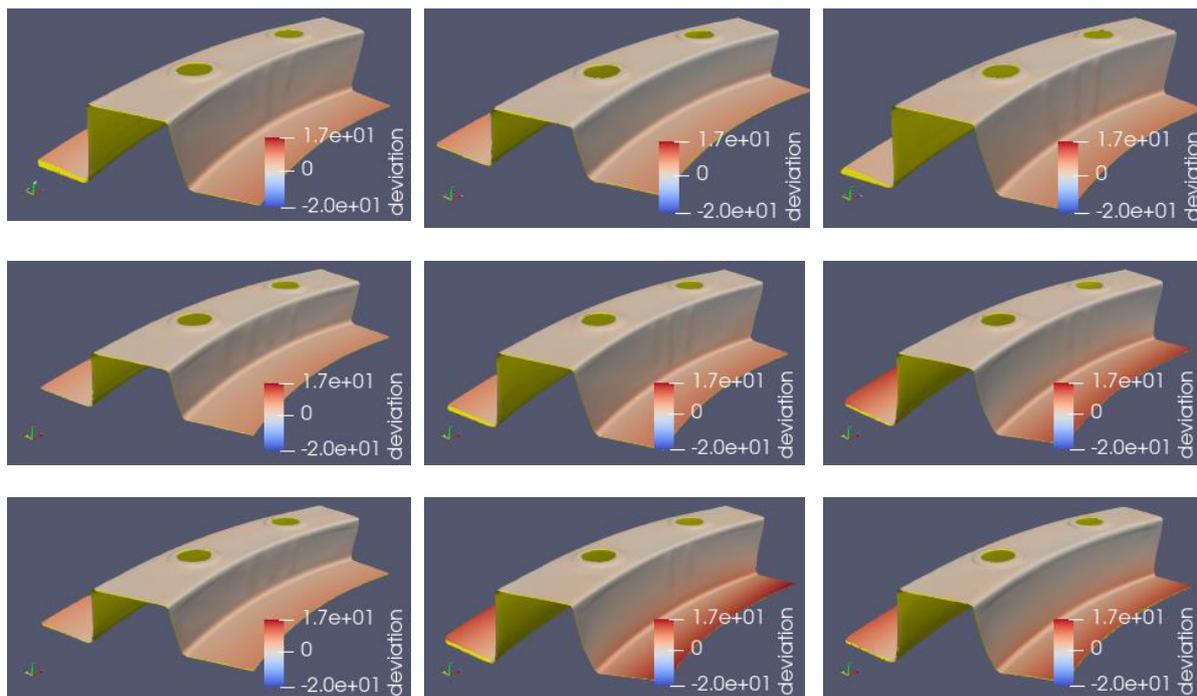


Abbildung 13: Geometrievermessung Einzelteile: Abweichung von CAD (mm)

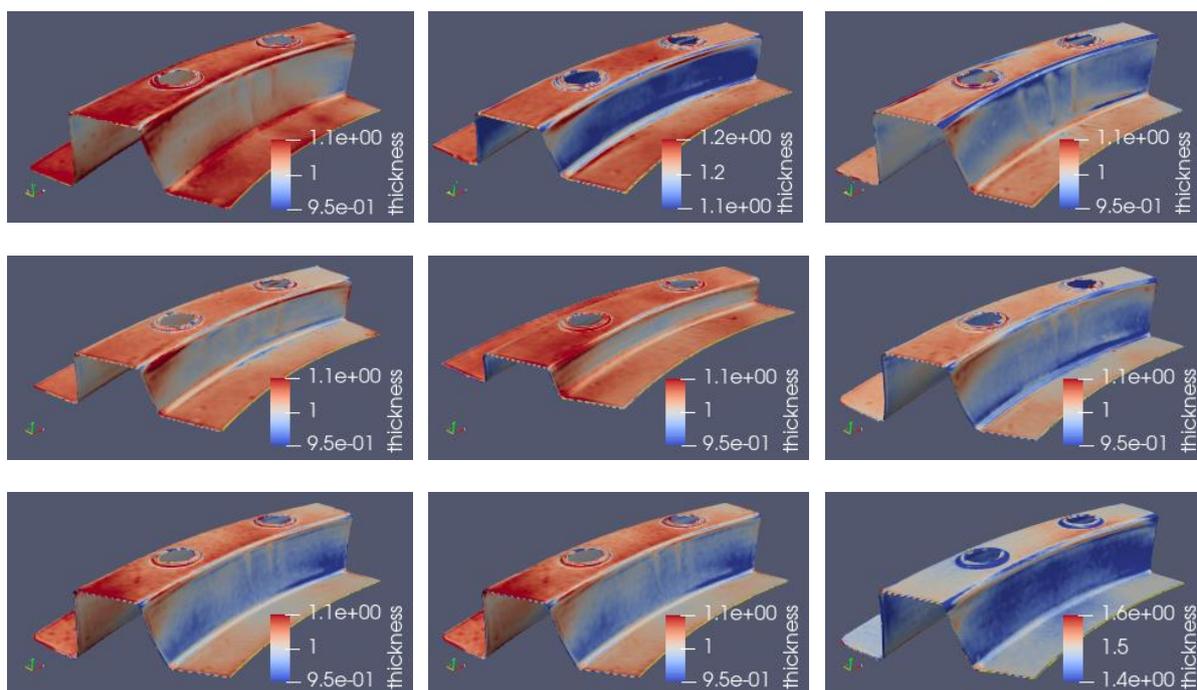


Abbildung 14: Geometrievermessung Einzelteile: Blechdickenverteilung (mm)

Die Vermessung der Baugruppen nach dem Positionieren, Spannen und Fügen fand in der Zusammenbau-Messvorrichtung statt. Aus 160 Einzelteilen konnten 80 Baugruppen mit jeweils einem Ober- und einem Unterteil gewonnen werden. Der Fügeprozessschritt besteht dabei aus sechs einzelnen Fügeoperationen, die entsprechend der vorher in der Bemusterung festgelegten Fügereihenfolge ausgeführt worden sind. Es kam abermals das Messsystem

GOM Atos für die Geometrievermessung zum Einsatz. Abbildung 15 stellt die Abweichung der ersten neun Einzelteilkombinationen (Baugruppendesigns) dar. Die Skala reicht von -3 mm bis +3 mm. Die Verteilung der Abweichung gegenüber CAD zeigt eine beträchtliche Bandbreite. Mit den streuenden Eingangsparametern der experimentellen Prozesskette lässt sich somit eine große Geometrievarianz als Ausgangsgröße erzeugen.

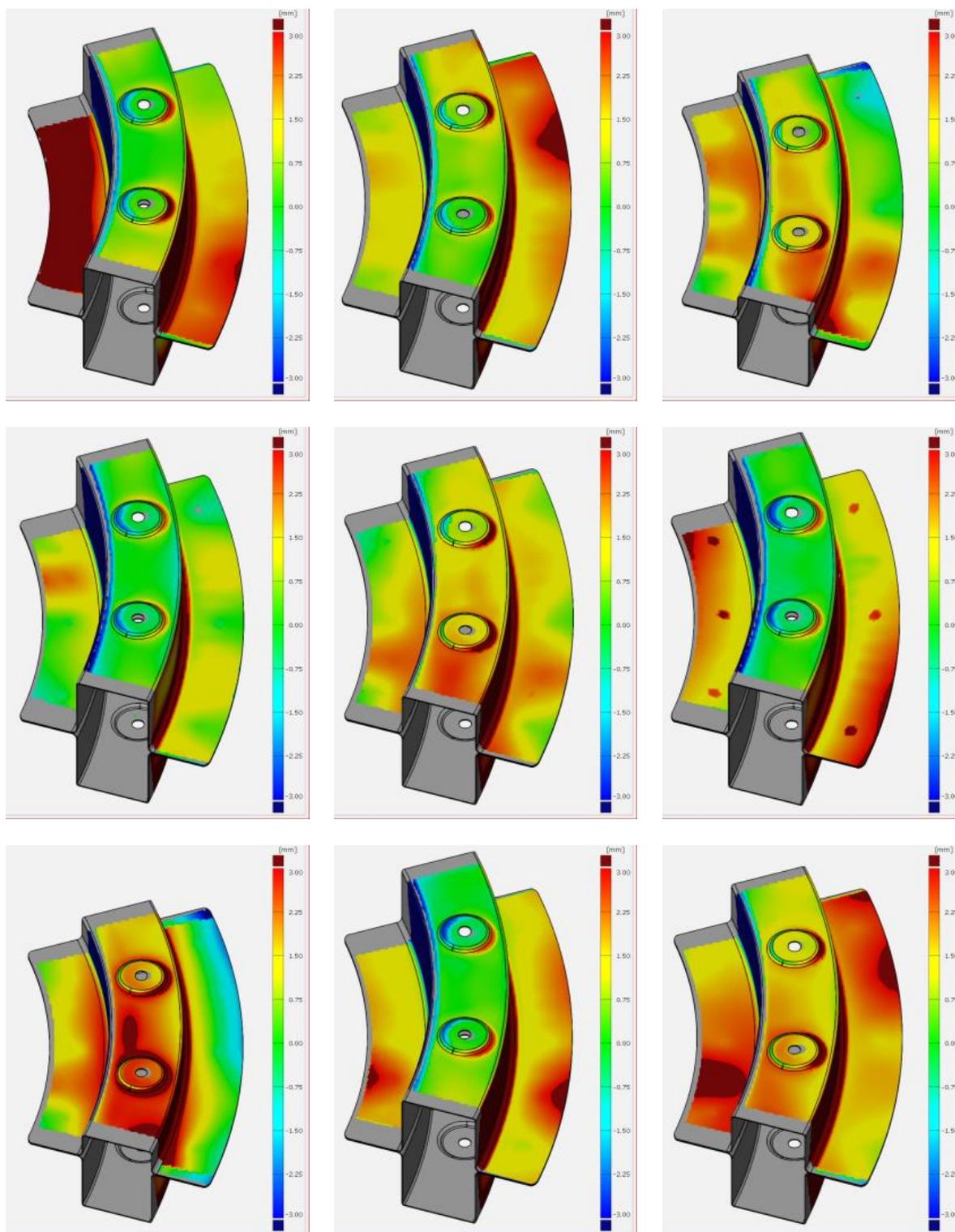


Abbildung 15 Geometrievermessung Baugruppe: Abweichung von CAD (mm)

Mit den validierten Einzelprozessschrittmodellen konnten die Berechnungen zur Datenakquise für die ML-Methoden (TU Chemnitz) durchgeführt werden, um die einflussreichsten Parameter zu identifizieren. Alle Teilprozessmodelle wurden verknüpft und der Informationsfluss von Modell zu Modell und die Dateiablagestruktur organisiert. Es kamen verschachtelte Skripte auf Basis von Python, Matlab und DOS zum Einsatz, um die Datenstruktur zu verwalten und die Solver zu starten sowie die Ergebnisse auszuwerten und abzulegen. Um Berechnungslastspitzen abzubauen, wurde Kontakt zum Hochleistungsrechenzentrum der TU Dresden geknüpft. Ein Teil der Berechnungen konnte somit auf das Cluster der TU Dresden ausgelagert werden. Die variierten Parameter der Datenakquise wurden entsprechend Tabelle 4 gestuft. Es entstanden somit 1000 Datensets pro Einzelprozessschritt bis zum Spannen mit unterschiedlichen Parameterkombinationen, die den Partnern zur Verfügung gestellt werden konnten. In den sich anschließenden Simulationen zur Zusammenbausimulation erfolgte die automatisierte Teilepositionierung gefolgt von den oben dargestellten Simulationen zum Spannen und Fügen. Mit den 1000 Einzelteilsimulationen sollten 500 Baugruppensimulationen durchgeführt werden. Infolge Ausfalls einiger Berechnungen durch Nichtkonvergenz konnten jedoch nur knapp 400 Baugruppensimulationen erfolgreich absolviert werden. Diese Daten wurden den Projektpartnern auf dem projektinternen Server zur Verfügung gestellt.

Tabelle 4: Stufung der Parameter in den Datenakquisesimulationen

Lfd. Nr. (-)	Mat.-ID (-)	Blechdicke (mm)	Niederhalterkraft (kN)	Ziehspalt (mm)	Ziehtiefe (mm)	Stempel-ID (-)	Matrizen-ID (-)	Einlegeposition (mm)
1,2,..., 1000	1,2,.. 6	0.99, 1.00,1.01, 1.19,1.48	10,20,..500	1.6,2.4	30,50, 70	2,3	1,2,3	-5,-4,..,5

Für das Tiefziehen wurde eine Sensitivitätsanalyse der Einflussparameter (ANOVA) durchgeführt. Hierfür wurde ein Algorithmus für das Mapping von Berechnungsergebnissen programmiert und in der am IWU entwickelten Software MetaField genutzt (Schwarz et al. 2018). Infolge der sich in jeder Simulation ändernden Topologien und Knotenzuordnungen war diese Ergebnisübertragungsmethodik nötig, um die Vergleichbarkeit zu wahren. Die umfangreiche Software kann für jeden Punkt auf einem beliebigen Bauteil die gegenseitigen Abhängigkeiten von Eingangsparametern wie Niederhalterkraft, Ziehtiefe, Werkzeuggeometrien, Einlege- und Spannerpositionen zu wählbaren Ausgangsgrößen wie Abweichung zu CAD, Blechausdünnung und Verfestigung korrelieren. Bei MetaField war eine Erweiterung der Sensitivitätsanalyse auf die Gesamtbaugruppe nicht möglich, da der Mappingalgorithmus bei den zum Teil sehr großen Abweichungen der Baugruppengeometrie von CAD nicht mehr zuverlässige Ergebnisse erbringt, wodurch die Vergleichbarkeit bei der

Ergebnisübertragung nicht mehr gegeben ist. Die Abbildungen 16 und 17 zeigen verschiedene Sensitivitätseinstellungen, die mit der Software vorgenommen werden können. Mit den weiter unten beschriebenen ML-Ansätzen sind auch Sensitivitätsanalysen von Ergebnisse mit stark vom Plan abweichender Geometrie möglich, was das Potential dieser Methoden aufzeigt.

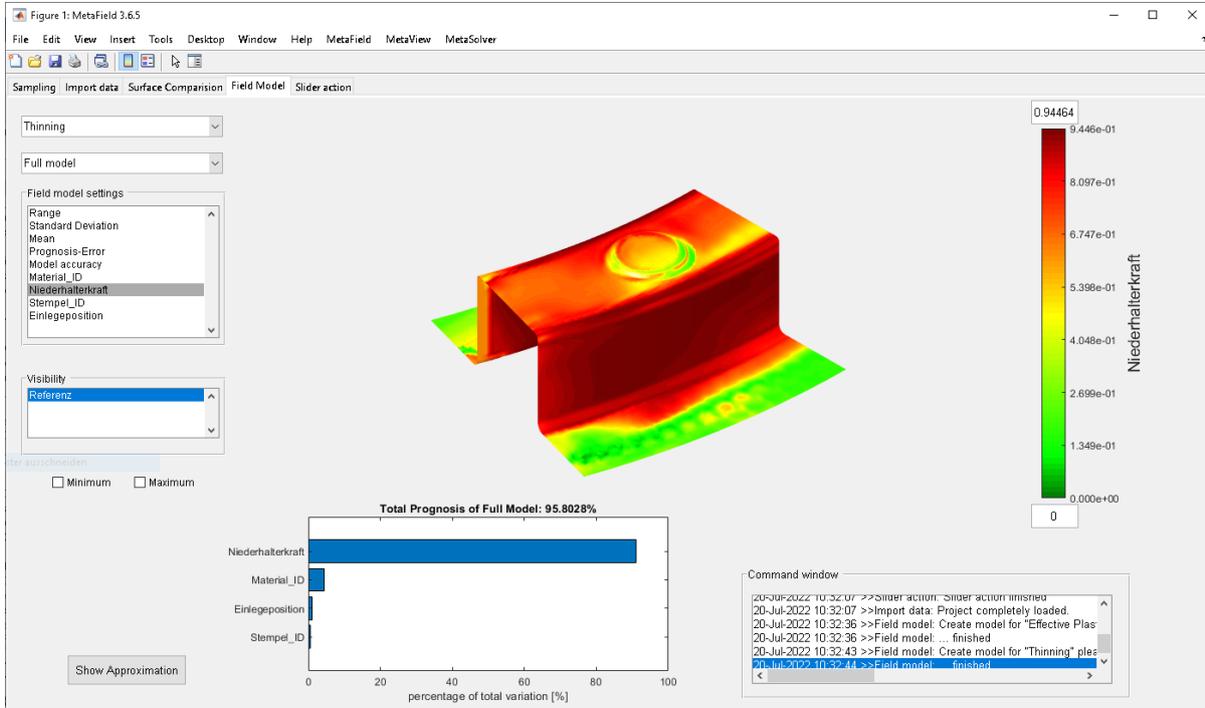


Abbildung 16: Sensitivitätsvisualisierungssoftware MetaField: lokaler Einfluss der Niederhalterkraft auf die Blechausdünnung.

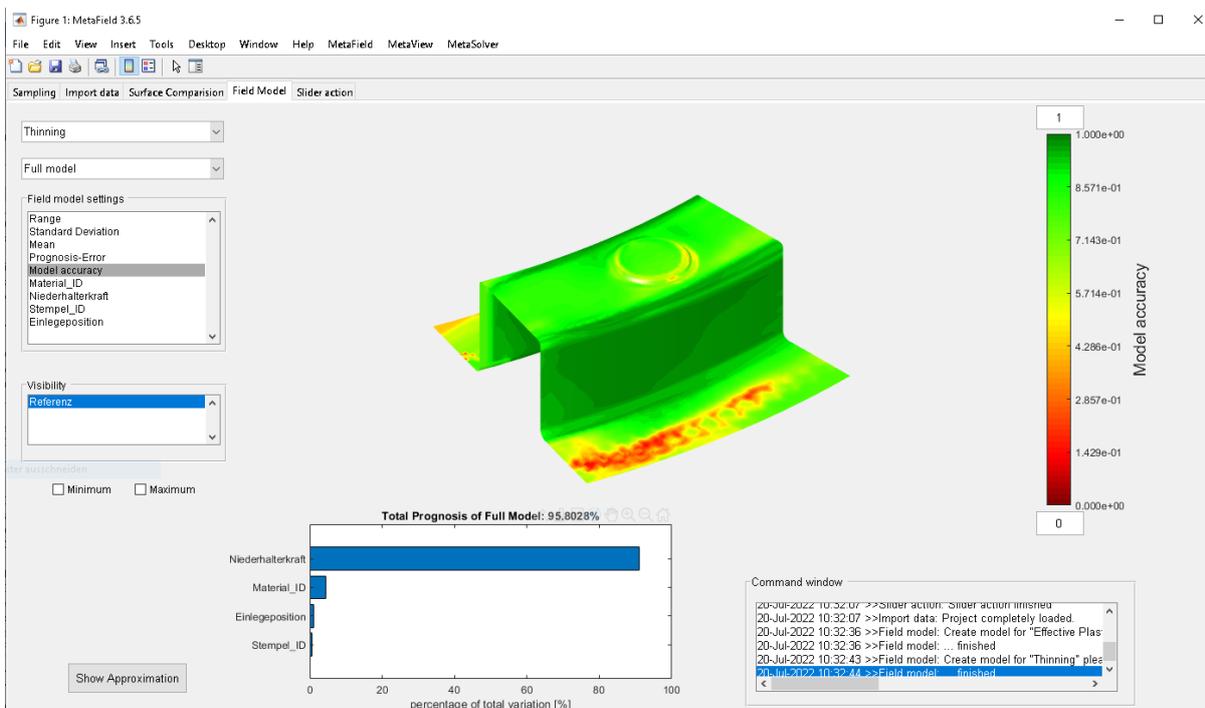


Abbildung 17: Sensitivitätsvisualisierungssoftware MetaField: lokale Prognosegüte des Modells bezüglich Blechausdünnung

Die automatisierte Berechnung und Auswertung der gesamten Prozesskette mit allen Einzelschritten ist eine hochkomplexe Aufgabe, die sehr viele Verknüpfungen und logische Abhängigkeiten erfordert. Mittels Skripts müssen Ergebnisse sehr unterschiedlicher Prozessschritte aufwendig standardisiert werden. Skripte, jegliche verwendete Software und Schnittstellen müssen hochflexibel ausgelegt sein. Trotz durchgeführter Maßnahmen zur Reduktion des benötigten Speicherplatzbedarfs kann das Datenvolumen, das verwaltet werden muss, extrem hoch sein. Ein hoher zeitlicher Aufwand beim Aufbau der durchgängigen virtuellen Prozesskette wird eingeführt durch die Tatsache, dass bei Veränderungen während des Aufbaus und nachträglich in der Kette alle Einzelberechnungen erneut durchgeführt werden müssen. Die Verkettung verursacht komplexe Abhängigkeiten, wodurch die Aufwände für Fehlersuche und –korrektur stark ansteigen. Im Rahmen des Projektes konnte eine robuste virtuelle Prozesskette entwickelt werden, die für die beschriebenen Prozesse flexibel genug ist, um z. B. die Bauteilgeometrie einfach austauschen zu können. Der Modellaufbau, die Berechnung und Auswertung der Simulation erfolgen automatisch. Abbildung 18 zeigt das Informationssteuerungsmodell der virtuellen Prozesskette mit allen Abhängigkeiten.

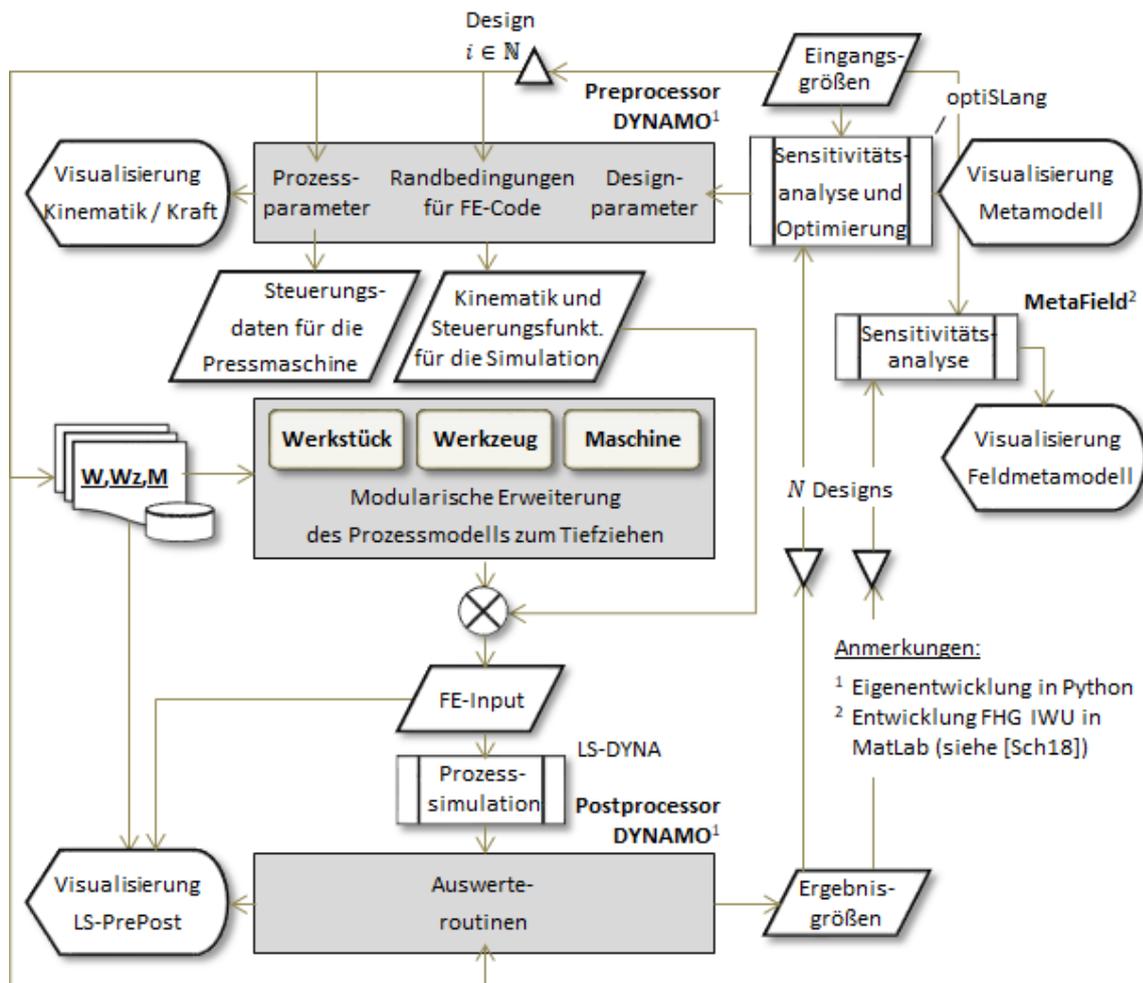


Abbildung 18: Informationssteuerungsmodell der virtuellen Prozesskette

3.2 SCALE

Softwarekonzept und Konzept zur Anwendung der Machine-Learning-Methoden

In dem ersten Arbeitspaket wurde ein Integrationskonzept zur softwaretechnischen Prozessabbildung einer Baugruppenherstellung entwickelt (AP1.3). Ziel war die Verknüpfung der Teilprozesse zu einem Gesamtprozess, welcher sich an realistischen Anwendungsfällen orientiert, vgl. Abb. 19 und Abb. 20.

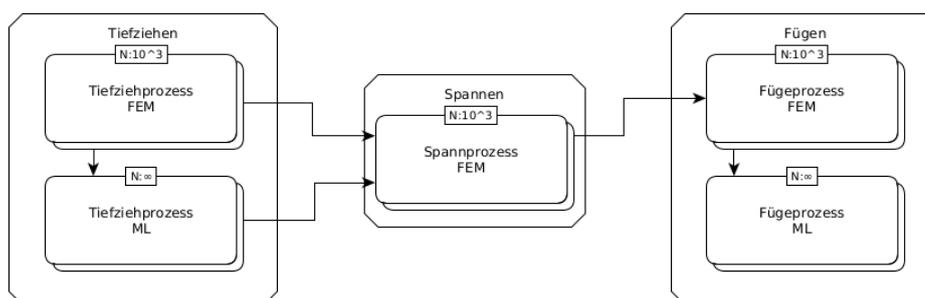


Abbildung 19: Prozessabbildung der Baugruppenherstellung mit FEM- und ML-Methoden

Das entwickelte Softwarekonzept definiert Schnittstellen und Datenmodelle inkl. der Datenformate für eine Modularisierung des Prozesses. Die relevanten Daten der Prozessschritte wurden identifiziert und ein Datenmanagementkonzept entwickelt. Weiterhin wurden mögliche Aufgabenstellungen für die Machine-Learning-Methoden herausgearbeitet.

Beginnen mit einer punktförmigen Prognose von Abweichungen des Bauteils bezüglich einer Referenzgeometrie wurden Prognosen von Bauteilschnitten angestrebt. Aufgrund der hervorragenden Prognosegüten der entwickelten 0-d und 1-d-Modelle wurde der Projektplan erweitert. Das neue Ziel war die Prozesse des Tiefziehens der Einzelteile und des Fügens der Gesamtbaugruppe mittels Methoden des maschinellen Lernens vollständig zu approximieren, um Bauteileigenschaften inkl. der finalen Bauteilgeometrie in Echtzeit basierend auf den gewählten Prozessparametern prognostizieren zu können, vgl. Abb. 20.

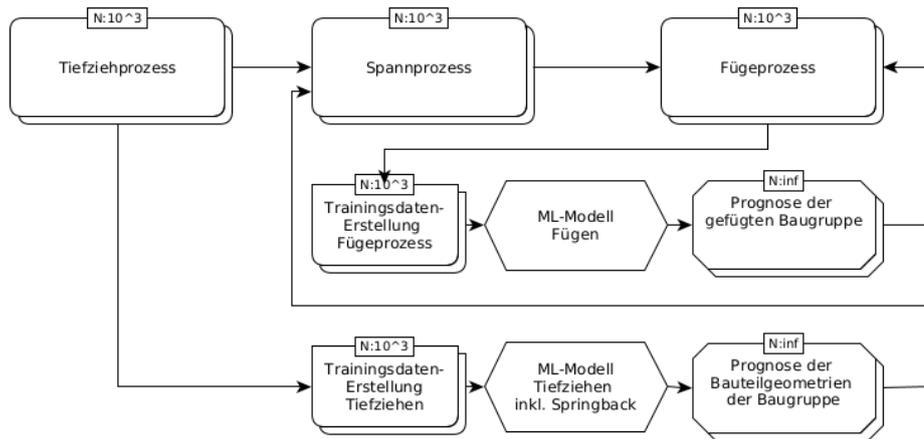


Abbildung 20: Prozessabbildung der Baugruppenherstellung

Abbildung 20 zeigt die experimentell und simulativ abgebildete Baugruppenherstellung aus Tiefziehen-Spannen-Fügen. Die Prozessschritte des Tiefziehens und des Fügens sollen mittels Machine-Learning-Modelle abgebildet werden. Der Projektpartner IWU liefert hierzu die experimentelle und simulativ gewonnenen Prozessdaten. Diese Massendaten müssen als Trainingsdaten für die ML-Modelle aufgearbeitet werden. Weiterhin wurde ein Konzept zur Implementierung der vom Projektpartner TUC bereitgestellten ML-Methoden in geeignete Software-Applikationen entwickelt.

Datenaufbereitung

Das Arbeitspaket AP2.2 widmete sich im besonderen Maße der Datenaufbereitung. Das Datenmanagement stellte aufgrund des großen Datenaufkommens eine besondere Herausforderung dar. Beispielsweise produziert eine Tiefziehsimulation ca. 2GB Roh-Ergebnisdaten. Die LS-DYNA-Simulationsdaten wurden in einem 1. Schritt in das dynain-Format konvertiert. Danach wurden diese Daten in ein offenes, solverunabhängiges Format (ML@Karoprod-hdf5-Format) übertragen, wobei alle Ergebnisse Element- und Knoten-basiert abgelegt worden sind. Dieses neuentwickelte Format zeichnet sich durch die Plattformunabhängigkeit, Performance und Dateigröße besonders aus. Alle Rohdaten aus Experiment und Simulation wurden in dieses Datenformat konvertiert und als Eingabedatenformat für nachfolgende Prozesse genutzt. Dieses standardisierte Datenformat stellt auch die Ausgangsbasis für die spezifischen ML-Trainingsdatenformate dar, vgl. Abbildung 21.

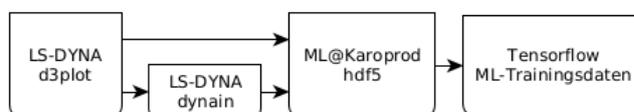


Abbildung 21: Datenmanagement & Datenkonvertierung

Die Benutzung des LS-Dynain-Formates (ASCII) des Finite-Element-Solvers hat sich hinsichtlich der Prozesszeit und Datenmengen, trotz der gegebenen Flexibilität und guten Lesbarkeit, als unvorteilhaft erwiesen. Deshalb wurde ergänzend zum Projektplan ein performantes Python-Modul zum Auslesen der LS-DYNA-D3Plot-Dateien entwickelt, welches die binären FE-Solver-Daten direkt in das binäre ML@Karoprod-hdf5-Format konvertieren kann, vgl. Abb. 21.

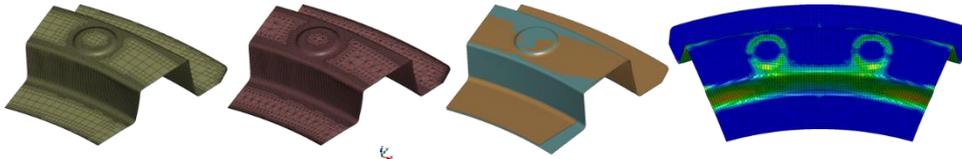


Abbildung 22: Datenmanagement & Datenkonvertierung

Die Prozess-Simulationsdaten des Tiefziehens lagen als Finite-Element-Netz-Daten vor, vgl. Abb. 22. Analog zur Praxis wurden adaptive, nicht-mannigfaltige FE-Netz-Topologien in den Tiefziehsimulationen verwendet. Für die Benutzung der Daten musste im Anschluss an die Prozess-Simulation ein Mapping der Prozessergebnisdaten auf neu generierte mannigfaltige Netzrepräsentationen erfolgen, vgl. Abb. 23. Alternativ wurde auch das Mappen auf ein Referenznetz untersucht (xyz->xyz_ref).

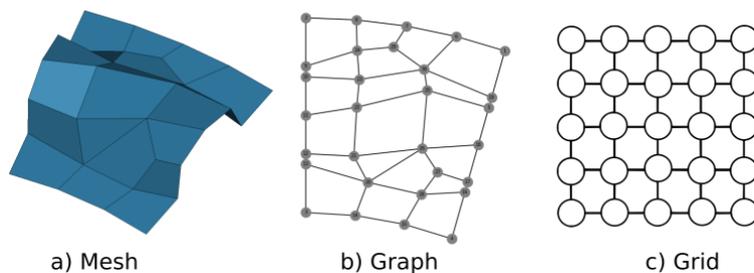


Abbildung 23: Datenrepräsentationen

ML-Methoden benötigen vergleichbare Datenstrukturen. Ein Schwerpunkt der Arbeiten lag demzufolge auf einer versuchsübergreifenden Ortsbeschreibung der Ergebnisdaten auf dem Bauteil. Neben dem Mappen der Ergebnisdaten auf ein Referenznetz wurden auch Projektionsmethoden und Flächenabwicklungsmethoden untersucht.

Abbildung 24 zeigt exemplarisch eine Darstellung des untersuchten Bauteils in Zylinderkoordinaten. Für ausgewählte Teilbereiche der Struktur lassen sich hiermit einfach Trainingsdaten für die 0-d, 1-d und bereichsweise 2-d-ML-Modelle generieren. Die Methode ist jedoch nur für eine kleine Klasse an Tiefziehbauteilen anwendbar.

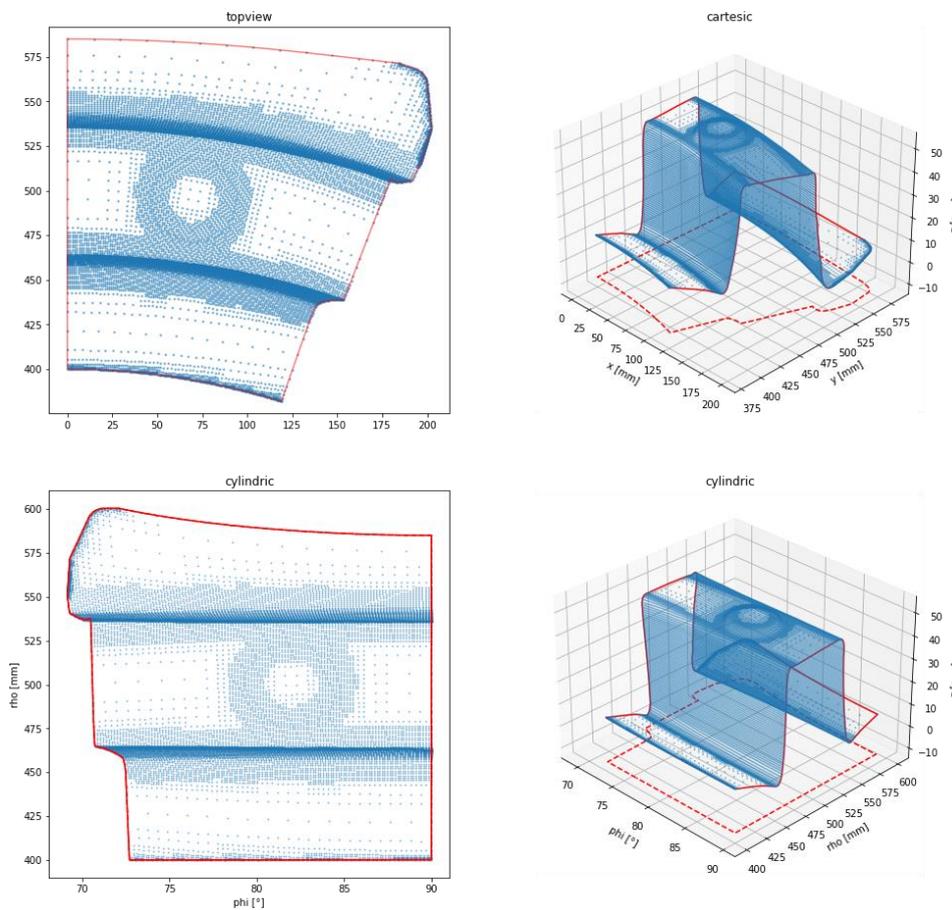


Abbildung 24: Beispiel Datentransformationen in Zylinderkoordinaten

Als vielversprechender haben sich Abwicklungsmethoden erwiesen, welche eine 3d-Flächen-Geometrie (x, y, z) zwei-dimensional abbilden (u, v). Abbildung 25 zeigt die exemplarisch die Verteilungen der Blechdicken und plastischen Dehnungen in einer 2d-grid-Datenrepräsentation basierend auf einem regelmäßigen Referenznetz. Hierzu wurden die Daten aus der Prozess-Simulation auf ein Referenznetz übertragen.

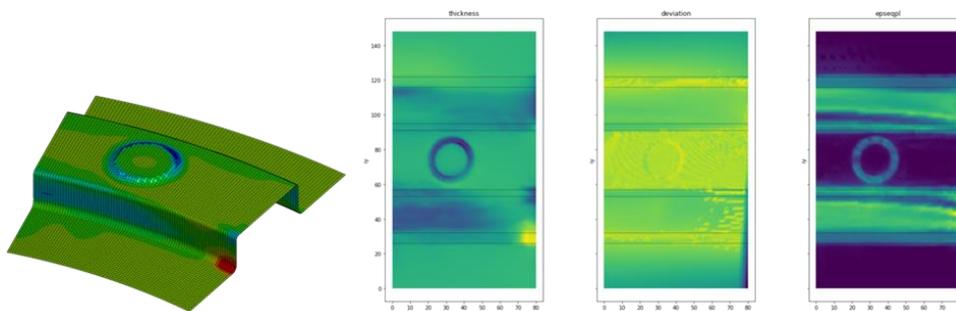


Abbildung 25: Grid-basierte Datenrepräsentation

Alternativ wurden mehrere UV-Mapper hinsichtlich ihrer Anwendbarkeit untersucht. Jedem Finite-Element-Knoten des unregelmäßigen Prozess-Simulation-Netzes kann eine 2d-Koordinate zugeordnet werden, vgl. Abb. 26.

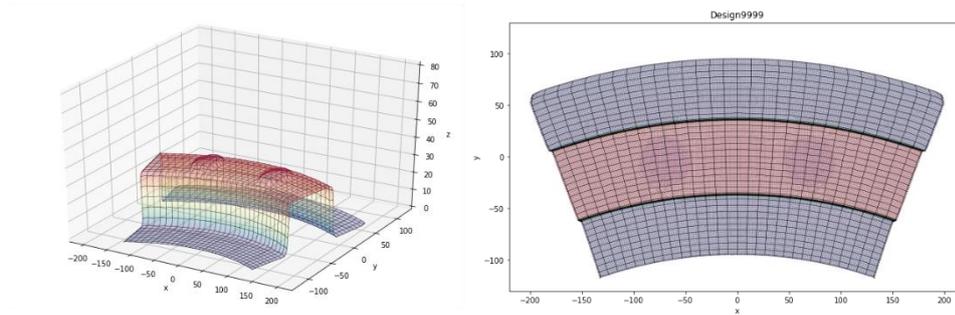


Abbildung 26: Grid-basierte Datenrepräsentation

Die 3d-Netze aus der Prozess-Simulation lassen sich somit in (ebene) 2d-Netze konvertieren. Für das im Projekt untersuchte Bauteil wurde zusätzlich eine Transformation der ebenen 2d-Koordinaten in einen normierten Parameterraum ($u=[0,1]$, $v=[0,1]$) durchgeführt, vgl. Abb. 27.

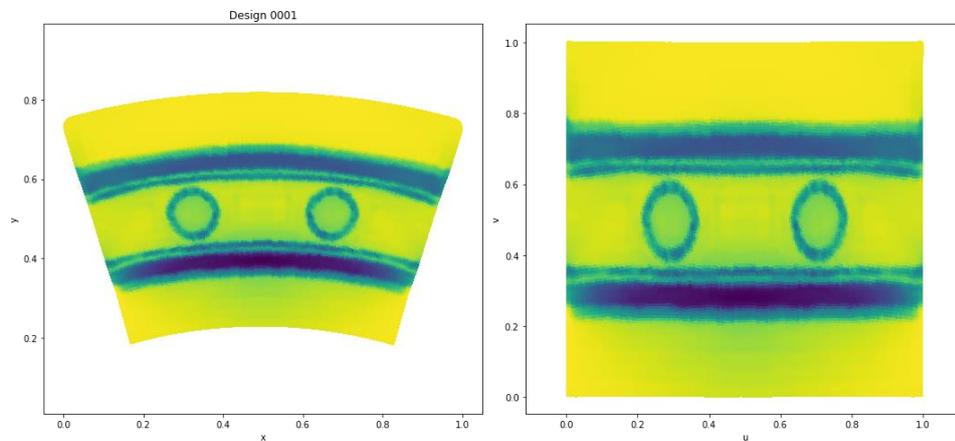


Abbildung 27: Grid-basierte Datenrepräsentation

Für jede Bauteilvariante lässt sich somit beispielsweise die Kontur mit $u=0$, $u=1$, $v=0$ oder $v=1$ identifizieren. Diese Positionsparameter stellen neben den Prozessparametern den Eingangsparametervektor für die ML-Modelle dar.

Datenmanagement

Der Workflow der Datengenerierung, Datenaufbereitung, ML-Datenerzeugung war in kontinuierlicher Entwicklung während des Projektlaufes. Die Versionierung der entwickelten Softwarekomponenten war deshalb besonders zu berücksichtigen. Mittels des Versionsmanagementsystems git gelang das sehr zufriedenstellend mit allen Projektpartnern. Die Systemtests erfolgten kontinuierlich und stellten die Funktionsfähigkeit der softwaretechnischen Prozessabbildung während der gesamten Projektbearbeitung sicher.

Ein Projektziel war die Erfassung und Speicherung aller relevanten technologischen Informationen der Einzelprozesse. Hierfür wurde in der Projektphase 2 ein webbasiertes Datenmanagementsystem entwickelt, welches die relevanten Prozessketten-Ergebnisdaten

und ML-Modelle speichern und visualisieren kann, vgl. Abb. 28. Die Ergebnisse des Prozessschrittes Tiefziehen können z.B. per Drag & Drop in den Web-Client importiert werden. Weiterhin können die Ergebnisgrößen der Versuche und Simulationen nach Attributen gefiltert und auf vielfältige Weise visualisiert werden.

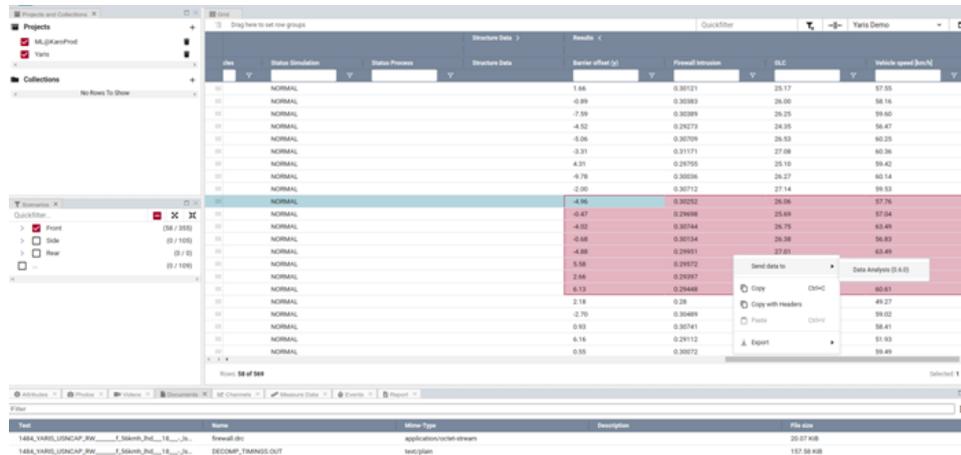


Abbildung 28: Web-Client in der Versuchsansicht

Ein Projektziel war die Implementierung von Machine-Learning-Modellen in das Datenmanagementsystem. Hierfür wurde u.a. das Modul „Data Analysis“ im Web-Client implementiert, vgl. Abb. 29. Die zu untersuchenden Daten können mittels Filter in der Versuchsansicht des Web-Clients selektiert werden.

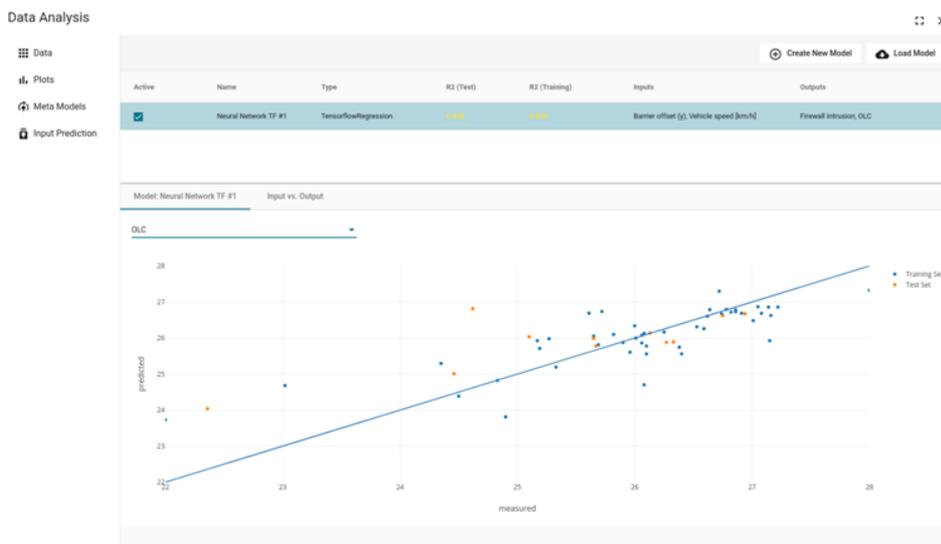


Abbildung 29: Web-Client in der Versuchsansicht

Aufbauend auf einer „Outlier Detection“ können ML-Modelle trainiert bzw. fertige ML-Modelle importiert werden. Hierfür wurde das ML-Framework „Tensorflow“ genutzt. Das Modul „Data Analysis“ erlaubt neben der Daten-Visualisierung mittels 2d-Scatter-Plots und Parallel-Plots auch eine 3d-Visualisierung der ML-Modelle, vgl. Abb 30.

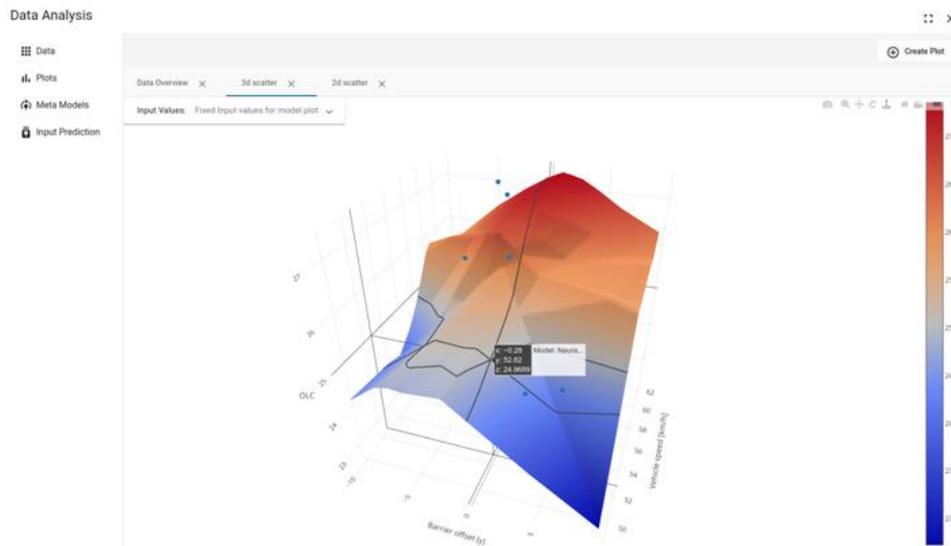


Abbildung 30: Exemplarische 3d-Visualisierung der Prognosedaten des ML-Modells

Für den Web-Clienten wurde auch eine Modellvisualisierung realisiert (Abb. 31). Die Modelldaten wurden im vtu-Format bereitgestellt. Der Anwender kann die Ergebnisse der Prozess-Simulation interaktiv studieren. Z.B. können die Verteilungen der Bauteildicken bzw. die plastischen Verzerrungen auf dem Bauteil visualisiert werden (AP6.2).

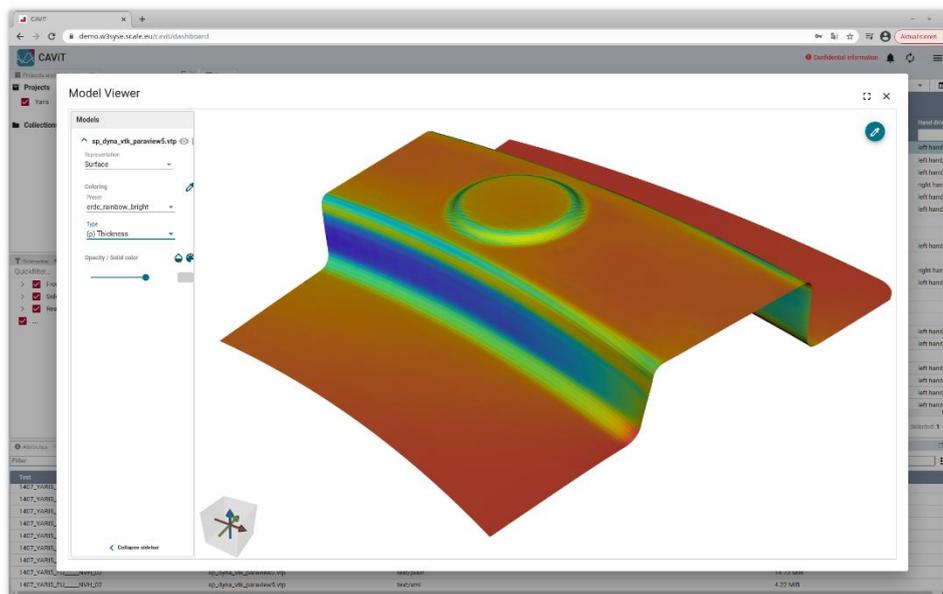


Abbildung 31: Datenmanagement-App

Machine-Learning-Modelle (ML-Modelle)

Gemeinsam mit dem Projektpartners TUC wurde die entwickelten ML-Modelle in dem Softwarepaket "ML-Karoprod-Predictor" implementiert. Hier erwies sich die git-basierte Versionsverwaltung des Softwareprojektes als sehr zielführend.

Implementiert wurde u.a. eine offenes Tensorflow-Serialisierungsformat, welches neben den trainierten ML-Modell auch die spezifischen Konfigurationseinstellungen des Modells in einer Datei abbilden kann. Dies ist für einen effektiven und sicheren Modellaustausch essentiell.

Exemplarisch seien hier die entwickelten ML-Modelle des Prozessschrittes Tiefziehen aufgeführt. Abbildung 32 zeigt die prognostizierten Daten der 0-d, 1-d, 2-d und 3-d-ML-Modelle.

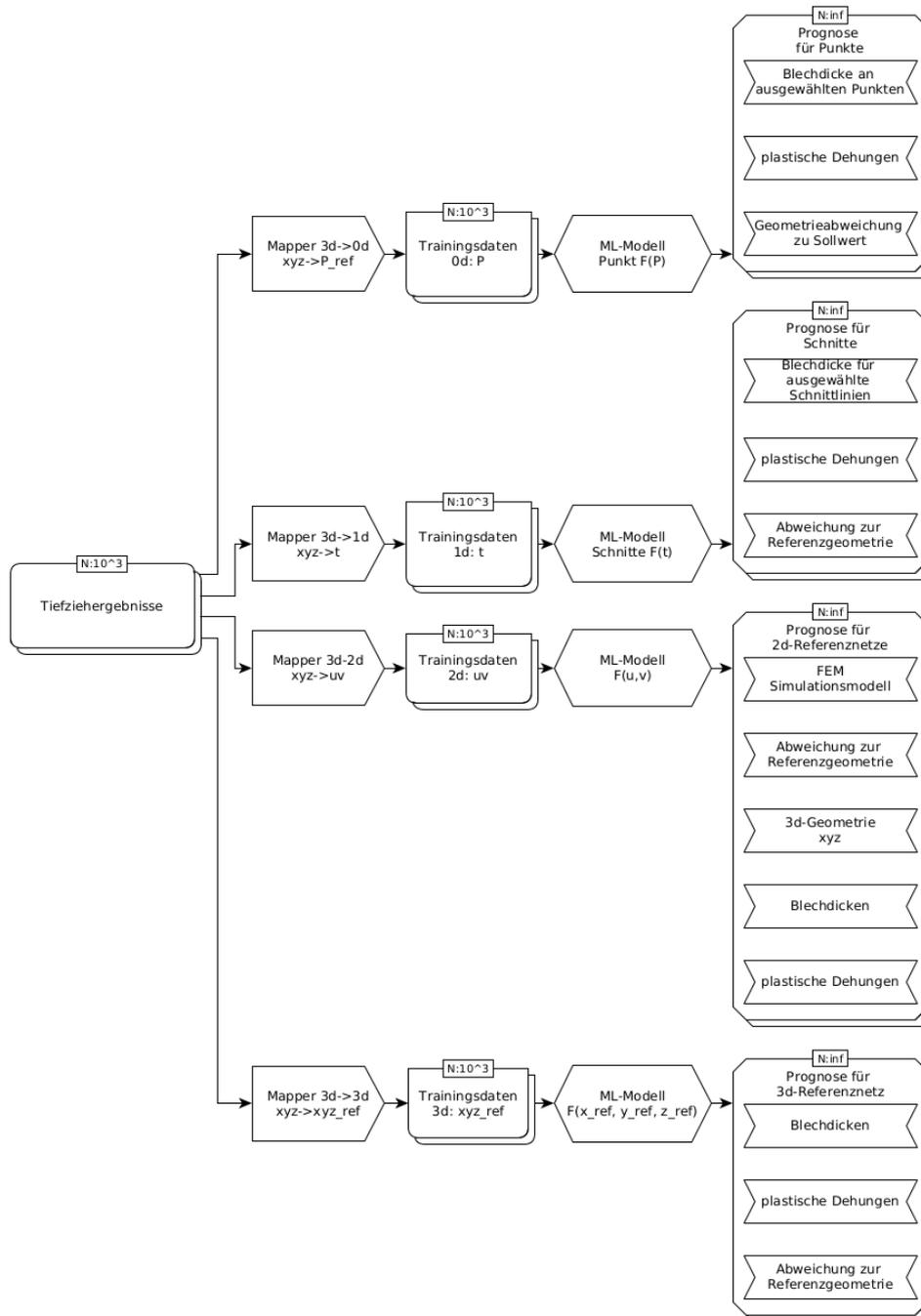


Abbildung 32: ML-Modelle für den Tiefziehprozess

1d-ML-Modelle

In der Anfangsphase des Projektes wurden für ausgewählte Punkte P des untersuchten Bauteils ML-Modelle $F(P)$ entwickelt. Der Nachteil liegt in der Festlegung der Punkte durch den Anwender vor der ML-Modell-Erstellung. In einem zweiten Schritt erfolgte das Trainieren von 1d-ML-Modellen, welche die Bauteileigenschaften an einem Schnitt durch das Bauteil betrachten. Hierzu wurde eine Ortvariable tp in dem Schnitt eingeführt, welche versuchsübergreifend, für jede Ziehtiefe jede Position im Schnitt eindeutig identifizieren kann, vgl. Abb. 33.

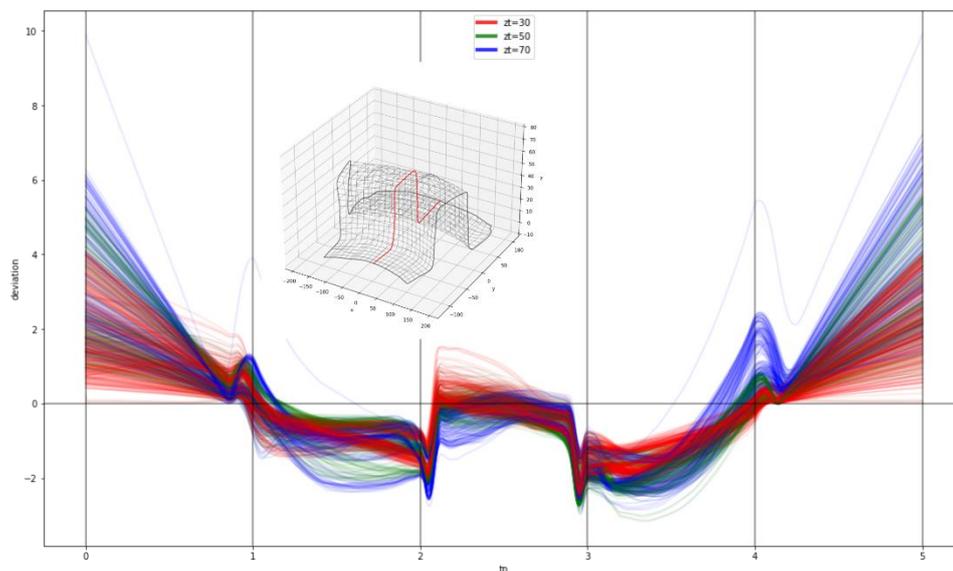


Abbildung 33: generierte Schnitte $x=0$ aus 1000 Simulationen

Basierend auf diesen Daten wurde von der TUC ein 1d-ML-Modell entwickelt, welches die Grundlage für die 1d-Prognose-App “ml_x0cut” darstellte, vgl. Abb. 34. Die Prognosegüten für die Geometrie des betrachteten Schnittes sind sehr zufriedenstellend (AP3.4).



x0cut: ground truth prediction

Design 949

output: deviation

	doe_id	Material_ID	Blechdicke	Niederhalterkraft	Ziehspalt	Stempel_ID
	842	949	5	1.4800	300.0000	2.4000
						3

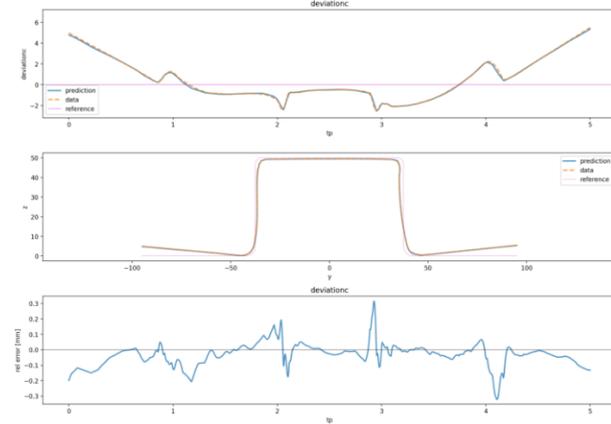
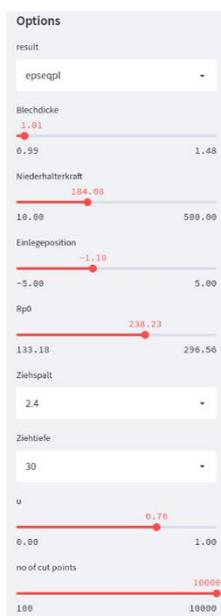


Abbildung 34: 1d-Prognose-App “ml_x0cut”

2d-Modelle

Mittels der entwickelten Beschreibung jedes Punktes auf dem Bauteil mit uv-Koordinaten konnten 2d-ML-Modelle entwickelt werden, welche neben den 3d-Koordinaten (xyz) auch die Bauteileigenschaften (Blechdicken, plastische Verzerrung) prognostizieren können. Abb. 35 zeigt die interaktiv bedienbare Applikation “ml_uvxyz” zum Studium der prognostizierten Bauteileigenschaften. Der Nutzer kann ausgehend von den gewählten Prozessparametern das Bauteil sowohl als 2d- und 3d-Geometrie als auch an gewählten Schnitten durch das Bauteil analysieren.



springback prediction

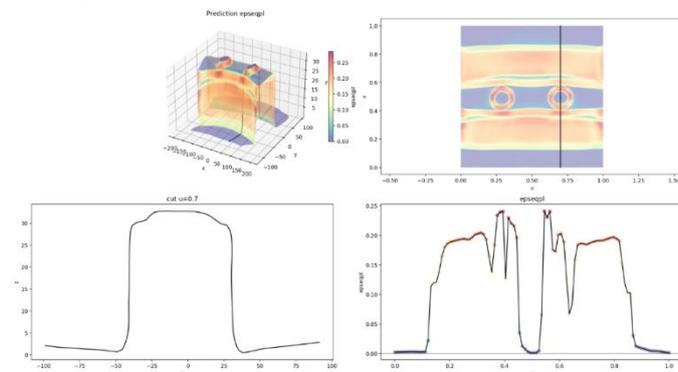


Abbildung 35: 2d-Prognose-App “ml_uvxyz”

Darüber hinaus kann ein Finite-Element-Modell aus den Prognosedaten analog zur Prozess-Simulation mittels LS-DYNA generiert werden, vgl. Abb. 36. Damit kann das ML-Modell den simulativen Prozessschritt des Tiefziehens vollständig ersetzen. Für eine Prognose wird weniger als eine Sekunde benötigt, wogegen eine vergleichbare FE-Simulation mehrere Minuten benötigt. Die enorme Beschleunigung der Bauteilprognose mittel ML-Modellen erlaubt dem Anwender eine interaktive Analyse des Prozesses in nahezu Echtzeit. Ein wesentlicher Fortschritt liegt in der wesentlich verkürzten Zeit, die ein Entwicklungs-Ingenieur benötigt, um ein gutes Modellverständnis zu erlangen und damit robuste Herstellungsparameter abzuleiten kann.

Für die nachgelagerten Prozesse stehen Finite-Element-Modelle mit den prognostizierten Blechdicken und plastischen Vorverformungen zur Verfügung, vgl. Abb. 36.

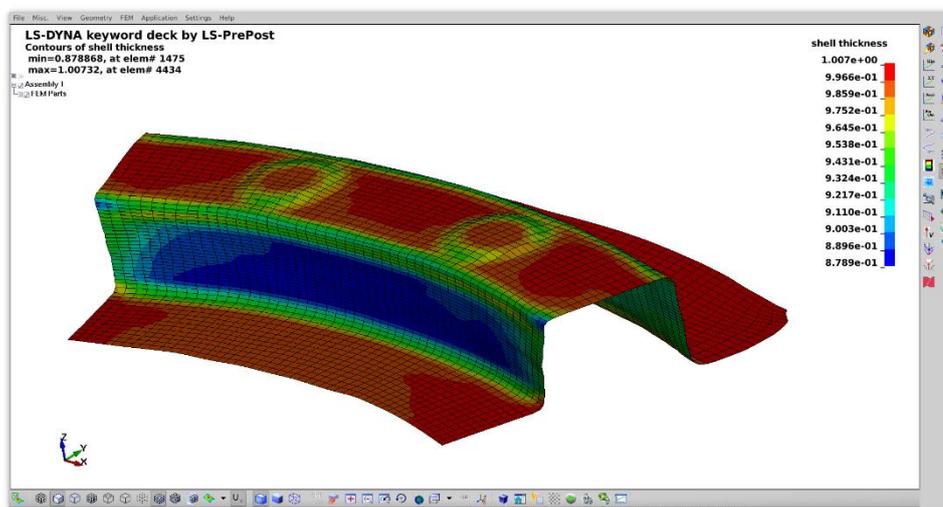


Abbildung 36: prognostiziertes Finite-Element-Modell

Es lassen sich auch 3d-Volumenmodelle mittels des ML-Prognose-Modells erstellen. Abb. 37 zeigt exemplarisch ein generiertes Finite-Element-Modell mit Volumenelementen.

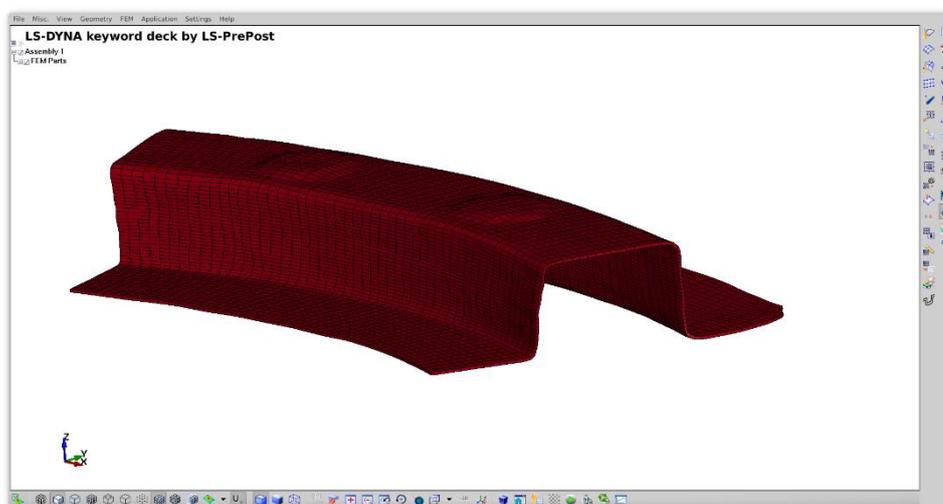


Abbildung 37: prognostiziertes Finite-Element-Modell

Robustheitsanalysen

In diesem Arbeitspaket wurde eine prototypische Applikation für eine Robustheitsanalyse entwickelt. Ziel war es, dem Anwender ein Werkzeug bereitzustellen, welches den Einfluss von Prozessparameterunschärfe auf die Bauteileigenschaften analysieren kann.

Grundlage bildet das von der TUC entwickelte ML-Modelle zur Prognose der 3d-Bauteilgeometrie basierend auf 2d-Positionsdaten (uv), vgl. Abb. 32. Entwickelt wurde die 2d-Prognose-Applikation “ml_robustness” (AP5.4). In der Applikation bietet dem Anwender die Möglichkeit in Echtzeit beliebige Schnitte durch das Bauteil zu legen und für die gewählten Prozessparameterbereiche 3d-Geometrien inkl. der zugehörigen Blechdicken- und Dehnungsverteilungen in den Bauteil zu prognostizieren, vgl. Abb. 38.

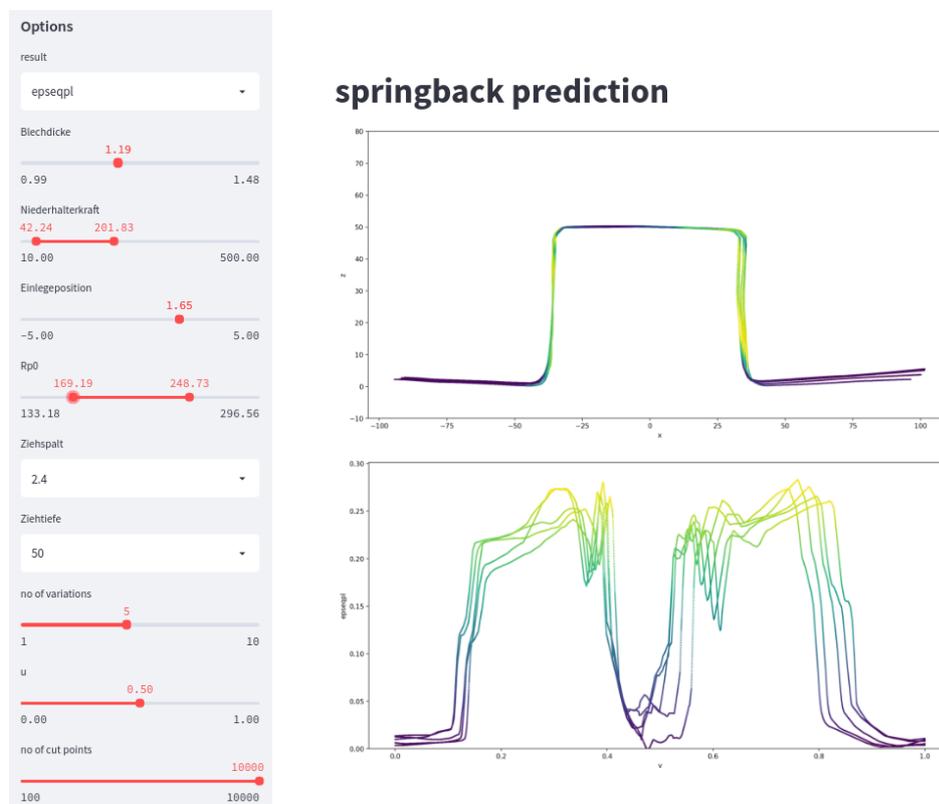


Abbildung 38: 2d-Prognose-App “ml_robustness”

Hierdurch ist nicht nur eine Parameteroptimierung des Tiefziehprozesses mit Berücksichtigung der Rückfederung möglich, sondern durch die extraordinary Performanz der Prognosemodelle kann der Anwender spielerisch ein Modellverständnis entwickeln und Parameterbereiche für eine robuste Prozessauslegung leicht ermitteln (AP4.2). Weiterhin können Prozessschwankungen unter Berücksichtigung der Prozess-Parameter-Unschärfe anschaulich ermittelt und dargestellt werden.

Prognose der Baugruppeneigenschaft

Für den Prozessschritt des Fügens der untersuchten Baugruppe wurde die Applikation “ml_joint” entwickelt (s. Abb. 39). Der Anwender kann interaktiv in Echtzeit den Einfluss der Spanner-Positionen auf das Fügeergebnis studieren. Neben der Geometrie der Baugruppe können Blechdickenverteilungen prognostiziert werden. Auch ist eine Verknüpfung mit den Prognoseergebnissen des ML-Modells für den Tiefziehvorgang möglich (z.B. plastische Dehnungsfelder).

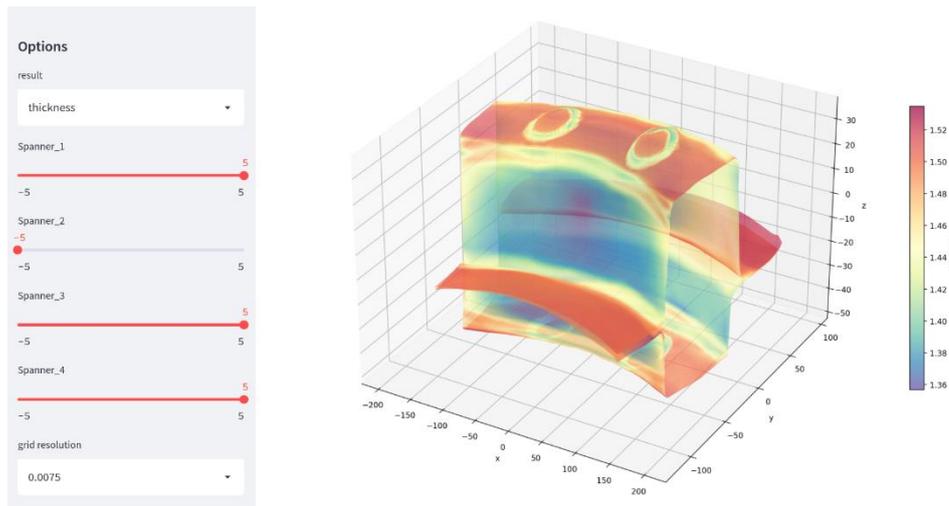


Abbildung 39: App “ml_joint”: Prognose der Baugruppengeometrie inkl. Blechedicken und plast. Verformungen

Für den Anwender sind auch Fragestellungen hinsichtlich der “Taschenbildung” zwischen den gefügten Bauteilen nach dem Lösen der Spanner interessant, vgl. Abb. 40, welche mit Hilfe des ML-Modells beantwortet werden können.

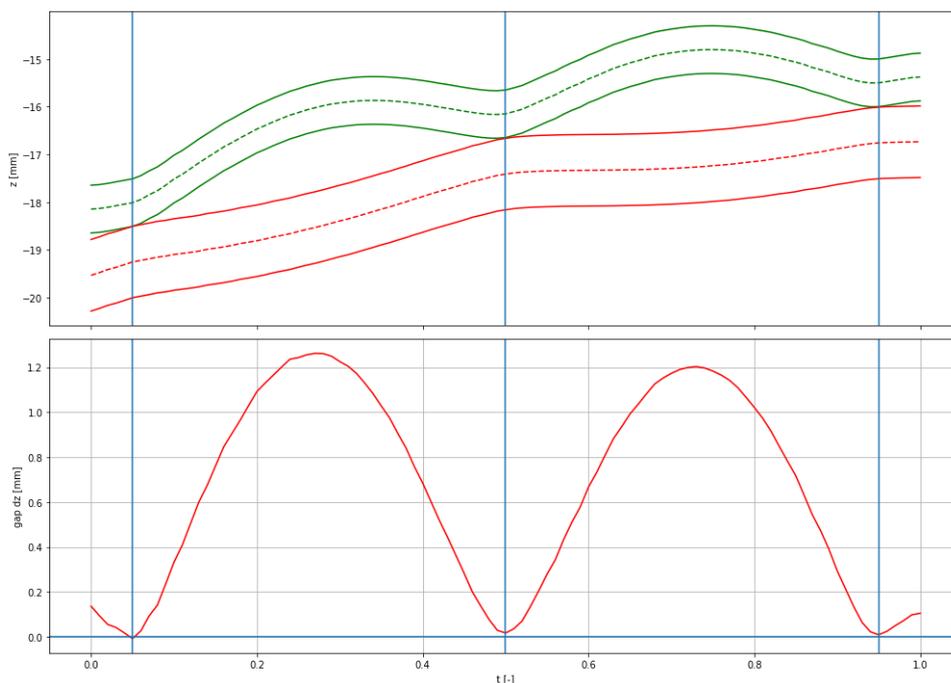


Abbildung 40: “Taschenbildung” zwischen den Bauteilen der gefügten Baugruppe

Abschließend wurde ein 3d-Modellgenerator für Schalen und Volumenelementen zusätzlich zum Projektplan implementiert. Abbildung 41 zeigt ein generiertes Finite-Element-Modell der Baugruppe mit Volumenelementen. Der Export von "wasserdichten" CAD-Modellen (z.B. stl) ist auch möglich.

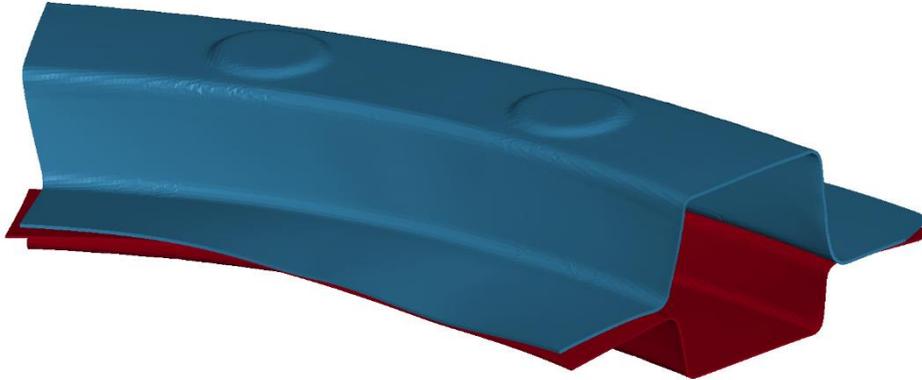


Abbildung 41: Datenexport als Finite-Element-Volumenmodell oder CAD-Modell

3.3 TU Chemnitz

Geometric Deep Learning für 3D Daten

Tiefe neuronale Netze haben bei der Verarbeitung von Daten wie Bildern und Videos großes Potenzial gezeigt. Die Bereitstellung von 3D-Strukturen für die neuronalen Netze ist jedoch nach wie vor eine Herausforderung. Geometric Deep Learning (GDL) ist ein Zweig des maschinellen Lernens (ML), der sich mit 3D-Daten für verschiedene Zwecke wie Klassifizierung, Komprimierung und Segmentierung befasst. Obwohl 3D-Modelle informativer sind, um die Umgebung zu beschreiben, sind gängige 3D-Darstellungen leider nicht leicht mit neuronalen Netzen zu kombinieren, und die Methoden zur Verarbeitung geometrischer Deformationen befinden sich noch in der Entwicklung.

Was diesen Bereich von seinem Pendant in 2D-Bildern unterscheidet, ist die zugrunde liegende Struktur der Daten, bei der die Daten nicht auf einem Gitter liegen. Dies hat zu Forschungen über nicht-euklidische Domänen, wie z.B. Mannigfaltigkeiten oder Graphen, geführt. Die Vielfalt der Darstellungen mit jeweils eigenen Vor- und Nachteilen hat insbesondere die Entscheidung über die Wahl des geeigneten Datentyps zu einer neuen Herausforderung gemacht (Ahmed et al., 2019). Man könnte zum Beispiel die folgenden nennen:

- Multi-view (mehrere 2D-Bilder eines 3D-Objektes): sie löst das Problem der Okklusion, aber die ausreichende Anzahl von Ansichten kann nicht im Voraus festgelegt werden, außerdem kann sie nur dort angewendet werden, wo nur minimale Verformungen auftreten.
- Volumetrisch (im Grunde ein regelmäßiges Gitter im 3D-Raum): diese Darstellung der Daten ist zwar einfacher zu bearbeiten, kann aber die Geometrie nicht erhalten.
- 3D-Punkte (eine Menge von Punkten im 3D-Raum, d.h. eine Punktwolke): Die Leichtigkeit, mit der solche Daten erzeugt werden können, ist ihr Hauptvorteil, doch ihre fehlende Struktur und das Fehlen von Informationen über die Konnektivität zwischen den Punkten erfordert eine andere Herangehensweise.
- Mesh (eine Menge von Eckpunkten mit ihren Konnektivitätsinformationen): eine bevorzugte Darstellung vor allem in der Computergrafik, jedoch sind globale euklidische Operationen nicht leicht zu definieren, z.B. Faltungen.

Ein 3D-Scanner liefert eine 3D-Punktwolke, die dann gerendert oder verarbeitet werden könnte, um eine Annäherung an das gescannte Objekt zu erhalten. Diese Methode könnte jedoch unterschiedliche Netze ausgeben, wenn die Punktwolke desselben 3D-Objekts variiert, selbst wenn die Geometrie nicht geändert wird, könnte sich die Topologie in diesem Fall unterscheiden. Aus diesem Grund werden viele 3D-Punktwolkendatensätze strengen

Methoden unterzogen, um aus der ursprünglichen Version der gescannten Objekte einen sauberen und adäquaten Datensatz zu erstellen, wobei ein Teil der Informationen zwangsläufig verloren geht. Darüber hinaus sind die Kosten für die Erfassung und Erstellung eines wünschenswerten Datensatzes allein durch das Scannen von 3D-Objekten enorm hoch, sowohl was die Zeit als auch die Ressourcen betrifft, wenn ein solcher Prozess durchgeführt werden soll.

Eine alternative Lösung könnte die Generierung eines Datensatzes durch Simulation sein, die sowohl in Bezug auf Zeit als auch auf Ressourcen wesentlich effizienter ist. Obwohl die gleichen Probleme hinsichtlich Inkonsistenz der Meshgröße und/oder der Topologie bestehen, ist es einfacher, die gewünschte Geometrie in einer Simulation zu handhaben. Die im Projekt zur Verfügung gestellten Daten enthielten sowohl 3D gescannte Informationen als auch das aus den LS-DYNA-Simulationen gewonnene Mesh. Die Daten hatten eine hohe Anzahl von Eckpunkten und damit eine hohe Anzahl von Flächen. Wenn ein neuronales Netz auf einem Datensatz trainiert werden soll, dessen Elemente/Samples enorm groß sind, konnte man schnell an die Grenze des Speicherplatzes des Prozessors stoßen, was das Training und die Entwicklung unpraktisch machte.

Solche expliziten geometrischen Deep-Learning-Methoden, die ganze Punktwolken oder Meshes als Trainingsmuster verwenden, wurden im ersten Teil des Projekts untersucht. Ein alternativer Ansatz, die so genannten impliziten Repräsentationen, wurde während der Projektlaufzeit eingeführt und hat in verschiedenen Disziplinen wie der Computergrafik viel Aufmerksamkeit erregt. Im Rahmen des Projekts haben wir diese neuartige Methode auf die Modellierung von Formveränderungen ausgedehnt.

Explizit 3D Deep Learning Ansatz

Convolutional Mesh Autoencoder (CoMA)

Ein entscheidender Aspekt beim Training eines neuronalen Netzes ist die Möglichkeit, Daten zu erzeugen, ohne die reale Simulation/Experiment zu verwenden, entweder weil die Durchführung der Simulation/Experiment ein teurer Prozess ist oder weil es schneller ist, die Ergebnisse aus einem trainierten Netz als aus einer FEM-Simulation zu erhalten. Einer der bemerkenswerten generativen Netztypen ist der Variational AutoEncoder (VAE), der lernen kann, Stichproben aus der den Daten zugrundeliegenden Verteilung zu generieren. Ein weiteres Interesse an der Verwendung unüberwachter neuronaler Netze wie Autoencoder besteht darin, die hochdimensionalen Daten in einem niedrigdimensionalen Bereich, dem sogenannten latenten Raum, komprimieren zu können. Der latente Raum extrahiert die notwendigen Informationen, um die Originaldaten so genau wie möglich zu rekonstruieren.

Seit dem Aufkommen der Autoencoder sind viele verschiedene Varianten eingeführt worden. Eines der Beispiele heißt Convolutional Mesh Autoencoder (CoMA, Ranjan et al, 2018). Er definiert einen Autoencoder zur Rekonstruktion des 3D-Modells des menschlichen Gesichts, das in Form eines Meshes in das Netzwerk eingespeist wird (Abb. 42). CoMA ist so konzipiert, dass die topologische Struktur des Meshes mit unterschiedlichen Maßstäben im Netz erhalten bleibt, wobei Mesh-Samplingoperationen verwendet werden. Im Enkoderteil wird das Netz durch Minimierung des Oberflächenfehlers nach unten abgetastet. Die upsampled-Meshes werden ebenfalls gleichzeitig aufgebaut, so dass die entsprechende Schicht im Dekoder die gleiche Größe wie der Encoder hat. Der Faltungsoperator für das Netz wird im Fourier-Raum definiert und mit Hilfe rekursiver Tschebyscheff-Polynome vereinfacht. Die Sampling Schichten sind zwischen Faltungsschichten verschachtelt. Wie in (Ranjan et al, 2018) erläutert, hat das 3D-Modell der Fläche 5023 Knoten im Netz, das erfolgreich auf nur 8 Variablen im Latentraum komprimiert wird, um ihr Modell zu rekonstruieren.

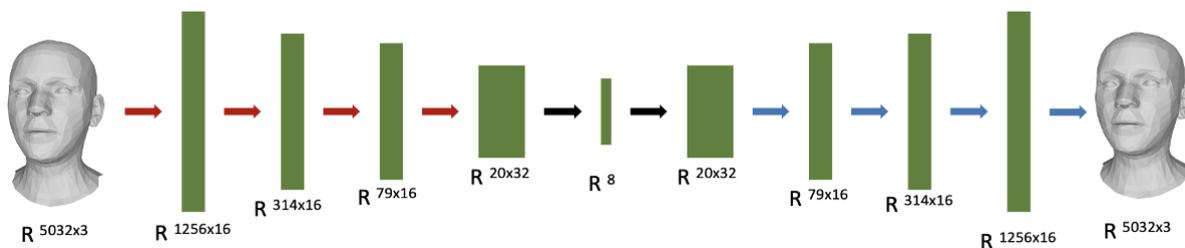


Abbildung 42: CoMA-Architektur - Downsampling (in rot), Upsampling (in blau) und Faltungsschichten (Rechtecke).

Generierung von FEM-Simulationsdaten

Die meisten FEM-Datensätze, die beim überwachten Lernen verwendet werden, haben eine inkonsistente Anzahl von Scheitelpunkten und/oder Flächen (z. B. ModelNet) oder sie werden manuell mit Software wie Blender verfeinert und bereinigt, bevor sie in einem Mesh-basierten neuronalen Netz verwendet werden. Dies schafft eine zusätzliche Komplexität für die Mesh-basierten neuronalen Netze, da sie Auffüllmethoden für fehlende oder neue Scheitelpunkte verwenden müssen, was die Qualität der extrahierten Merkmale beeinträchtigt, wie wir in unseren ersten Experimenten beobachten konnten.

Wir verwendeten FreeCAD, eine Open-Source-Anwendung zur Lösung von Aufgaben im Maschinenbau, um einen Simulator zu entwerfen, der die Konsistenz der generierten Mesh garantiert. Die Möglichkeit, ein Makro in Python zu definieren, das in FreeCAD geladen werden kann, war zusätzlich entscheidend, da die meisten Frameworks für maschinelles Lernen (Tensorflow, Pytorch) nur in Python verfügbar sind. Der von uns entwickelte Simulator RLFEM, der auf [Github](https://github.com/hex41434/FreeCAD_RL_FEM_Environment) frei verfügbar ist (https://github.com/hex41434/FreeCAD_RL_FEM_Environment), erlaubt es uns, wiederholt Aktionen auf ein Blech anzuwenden und die resultierenden Netze zu beobachten.

RLFEM bietet eine Gym-ähnliche Schnittstelle, die die Standard-Umgebungs-API in der Reinforcement Lernen (RL) Forschung ist und uns erlaubt, jede RL-Implementierung transparent auf unser Problem anzuwenden. RLFEM erlaubt es, eine Umgebung für den Agenten zu erstellen, in der die Zustände der aktuelle Zustand eines Blechs sind und die Aktionen die Höhe und der Ort der Kraft sind, die auf das zu verformende Blech angewendet werden soll. Die aufeinanderfolgenden Aktionen können entsprechend der Strategie, die der Agent verfolgt, angewendet werden, und sobald eine bestimmte Anzahl von Aktionen erreicht ist, würde die Episode enden. Die Belohnung bei jedem Zustand ist die Ähnlichkeit des Zustands mit dem gewünschten Zustand. In Abb. 43 werden verschiedene Aktionen ausgewählt und die Ergebnisse gezeigt. Aktionen waren hier die Erzeugung von kreisförmigen Druckpunkten mit unterschiedlichem Radius und/oder Koordinaten des Mittelpunkts.

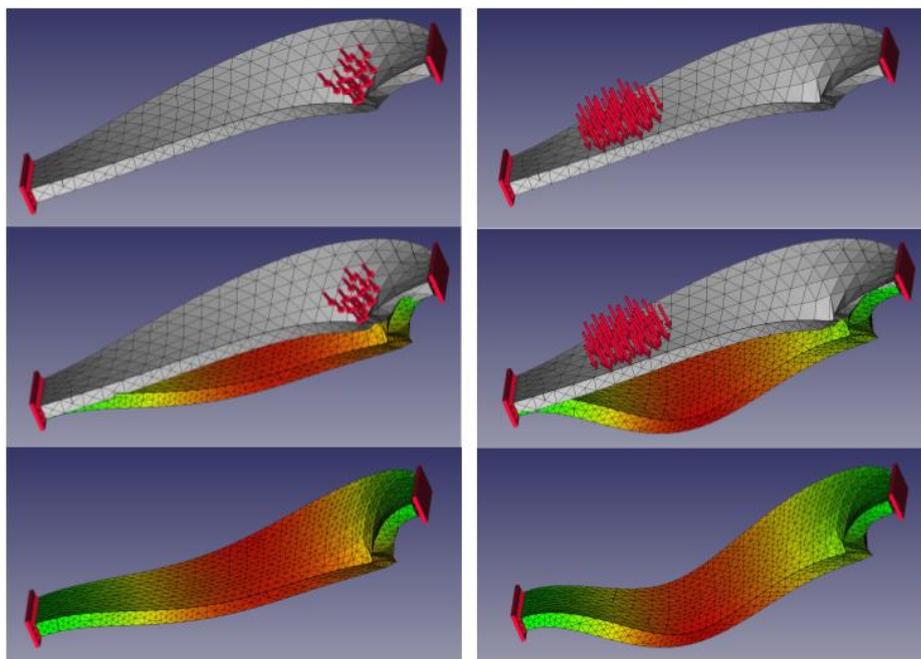


Abbildung 43: Mit RLFEM können wir eine Folge von Einwirkungen auf ein Blech anwenden und jeden einzelnen Zustand mit einer festen Topologie beobachten.

Rekonstruktion deformierter Meshes

Mit unserem RLFEM Simulator haben wir 6228 Proben (deformierte Meshes) erstellt, wobei jedes Mesh 381 Knoten hat. Eine CoMA-Architektur mit 19 Schichten (4 Faltungsschichten und 5 Down- oder Up-Sampling-Schichten für den Encoder und den Decoder sowie eine Schicht für den Latentraum) wurde in Tensorflow implementiert. Wir wählten eine Größe von 16 für den Latentraum, 250 Trainingsepochen und eine anfängliche Lernrate von 0,05. In Abb. 44 stellen wir ein beispielhaftes Mesh und seine Rekonstruktion dar.

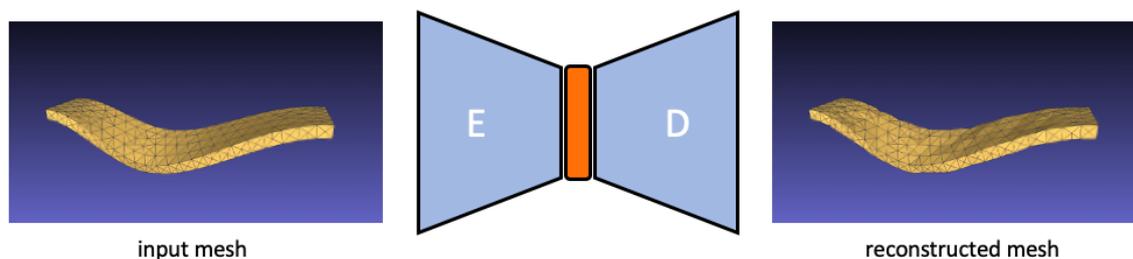


Abbildung 44: Mesh-Rekonstruktion unter Verwendung der CoMA-Autoencoder-Architektur.

Dies zeigt, dass die CoMA-Architektur in der Lage ist, Netze mit 381 Knoten unter Verwendung von nur 16 Werten im Latentraum korrekt zu rekonstruieren. Allerdings setzen netzbasierte Methoden entweder eine feste Topologie für die Eingabe voraus oder sind nicht in der Lage, das 3D-Netz ohne die Adjazenz zu rekonstruieren. Dies ist eine offene Frage im Bereich des Geometric Deep Learning, um ein verallgemeinertes Modell zu entwickeln, welches für variierende Topologien geeignet ist. Während unser RLFEM-Simulator es uns erlaubt mit diesen Einschränkungen umzugehen, stellt sich die Frage nach der Verallgemeinerung unserer vorgeschlagenen Methode auf andere FEM-Simulationen. Wir haben daher einen anderen Ansatz untersucht, der nicht allein auf Dreiecksnetze angewiesen ist, da diese auf eine bestimmte Topologie beschränkt sind.

Implizit 3D Deep Learning Ansatz

Deep SDF: Signed Distance Functions

Die Signed Distance Function (SDF) ist eine Darstellung von 3D-Formen, bei der jeder Punkt in 3D einen Abstandswert hat, unabhängig davon, ob er sich innerhalb (-), außerhalb (+) oder auf der Formoberfläche (0) befindet. Diese kontinuierliche Interpretation des Raums ermöglicht es, Formen in jeder gewünschten Auflösung zu rekonstruieren und Formverformungen ohne großen Speicherbedarf zu bewahren. Die Idee, SDF mit tiefen neuronalen Netzen zu kombinieren, wurde 2019 vorgestellt und DeepSDF genannt (Park et al., 2019).

Ein neuronales Netz kann trainiert werden, um den SDF-Wert einer Form S für jede Abfrage-Eingabeposition im 3D-Raum zu schätzen (Abb. 45, links). Dieses neuronale Netz ist eine eingebettete Darstellung einer einzelnen Form. Um das Netz auf mehrere Formen zu verallgemeinern, wird ein Code als Bedingung zur Netzeingabe hinzugefügt (Abb. 45, rechts). Diese Formkodierung wird durch eine geschichtete Autodecoder-Architektur realisiert. Ähnlich wie latente Repräsentationen in der Autodecoder-Architektur sind die "Codes" eines Autodecoders eingebettete Repräsentationen der Formen. Für jeden Abfragepunkt (x, y, z) sagt das Netz den SDF-Wert voraus, der der angegebenen Form entspricht. Durch gleichmäßige Abtastung des 3D-Bereichs und Ableitung der Mannigfaltigkeit, bei der der SDF-Wert gleich 0 ist, kann die Oberfläche des Objekts als Punktwolke und sogar als 3D-Netz

rekonstruiert werden. Diese kontinuierliche Interpretation des Raums ermöglicht es, Formen in jeder gewünschten Auflösung zu rekonstruieren und Formverformungen ohne großen Speicherbedarf zu bewahren. Mit dieser Methode kann das Problem der effizienten Einbettung von Formen wirksam angegangen werden, und die Größe des Netzes ist vergleichbar mit klassischen Ansätzen zur Verarbeitung beliebig großer Netze mit beliebiger Topologie.

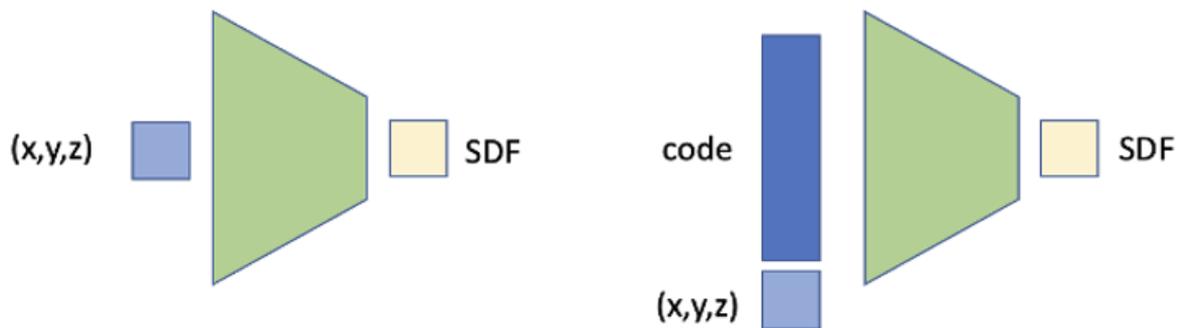


Abbildung 45: DeepSDF, Deep Signed Distance Function. Das Netzwerk würde lernen, die SDF-Werte vorherzusagen.

Deep SDF Netze für geometrische Formverformung

Unter Verwendung der 6228 von unserem RLFEM-Simulator erzeugten Proben geben wir die räumlichen Koordinaten eines Punktes und den auf die Form ausgeübten Kraftvektor als Eingabe in das Netz ein und sagen den entsprechenden SDF-Wert als Ausgabe voraus. Wir trainieren ein vollverbundenes neuronales Netz mit sechs Eingaben (x -, y -, z -Koordinaten eines Punktes, x -, y -Position der auf die Oberfläche ausgeübten Kraft und Kraftgröße). Nach einer Bayes'schen Hyperparametersuche unter Verwendung der optuna-Bibliothek besteht das optimale neuronale Netz aus 4 versteckten Schichten (130-118-150-148) unter Verwendung der LeakyRelu-Aktivierungsfunktion und einem linearen Ausgangsneuron. Der mittlere quadratische Fehlerverlust wird mit dem Adam-Regularizer und einer Lernrate von 0,0005 minimiert. Das Netz wird für 150 Epochen trainiert, wobei ein Validierungssatz aus 20 % der Daten besteht. Am Ende des Trainings liegen sowohl die Trainings- als auch die Validierungsverluste unter 10^{-6} .

Das Netz liefert eine kontinuierliche Funktion, die die SDF für jeden Abfragepunkt im Raum darstellt, so dass für die endgültige Formfindung ein weiterer Schritt erforderlich ist. Marching Cube (MC; Lorenzen und Cline, 1987) ist der gängigste Ansatz zur Extraktion des Netzes in unterschiedlichen Auflösungen. Durch die Änderung der "Würfelgröße" als Eingabeparameter für den Algorithmus werden die SDF-Werte innerhalb eines Einheitswürfels diskretisiert, um die Oberfläche der Form in verschiedenen Auflösungen zu rekonstruieren. In Abb. 46 sind zwei generierte Netzmuster und die entsprechenden Grundwahrheitsnetze aufeinander dargestellt (für eine Würfelgröße von 95^3).

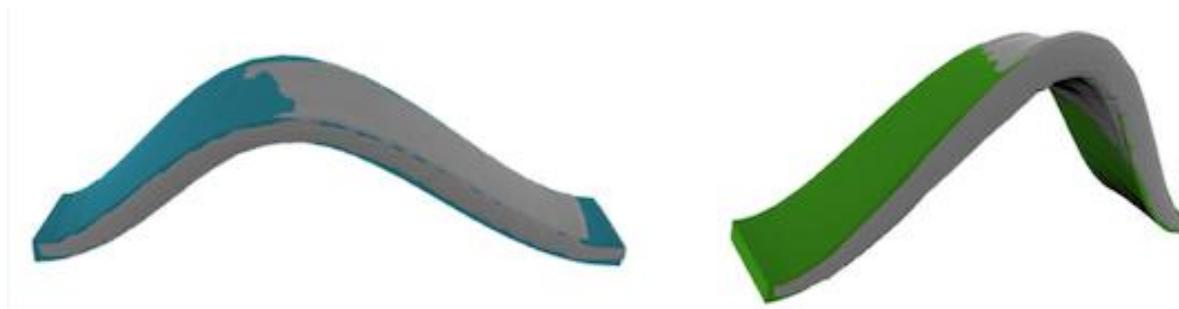


Abbildung 46: Das aus der NN-Vorhersage rekonstruierte Netz (in grau) überlagert sich bei zwei verschiedenen Stichproben mit dem tatsächlichen Netz (in Farbe).

Zur Bewertung der Qualität des rekonstruierten Netzes im Vergleich zum Original verwenden wir eine beliebige Metrik, die Chamfer Distanz (CD). Wir haben 116 Stichproben aus dem Testsatz zufällig ausgewählt und das Netz anhand der Netzvorhersagen rekonstruiert. Der Mittelwert der CD für 30000 Stichprobenpunkte ist in der folgenden Tabelle 5 dargestellt. Wie erwartet verringert sich die CD, wenn die Anzahl der Würfel im MC-Algorithmus erhöht wird (was zu feineren Netzen führt).

Tabelle 5: Bewertung der Qualität des rekonstruierten Netzes

Anzahl von Würfeln	Chamfer Distanz
85^3	0.0009595
95^3	0.0009171
105^3	0.0008937
120^3	0.0008754
130^3	0.0008747

Die Verwendung der SDF-Darstellung hat die folgenden Vorteile: Jedes simulierte Netz oder CAD-Modell kann leicht in eine SDF-Darstellung umgewandelt werden, so dass die vorhandenen Datensätze für das Training verwendet werden können. Im Gegensatz zu expliziten Darstellungen ist diese Darstellung NN-freundlich und kann große Netze mit beliebiger Topologie verarbeiten. Außerdem kann eine große Anzahl von Formen als trainiertes neuronales Netz gespeichert werden, was Speicherplatz spart. Nach dem Training des Netzes werden im Vergleich zu numerischen Ansätzen weniger Rechenleistung und Zeit für die Verarbeitung großer Netze benötigt. Unseres Wissens ist dies das erste Mal, dass diese Darstellung mit neuronalen Netzen für die Verarbeitung deformierbarer Objekte kombiniert wird. Diese Arbeit wurde auf der KI-2022-Konferenz in Trier angenommen.

Trotz dieser Vorteile haben SDF-Darstellungen zwei große Schwierigkeiten zu bewältigen: Die 3D-Netzwerke müssen wasserdicht sein, um den 3D-Raum in Innen- und Außenbereiche zu unterteilen. Leider sind viele CAD-Modelle nicht wasserdicht, und es sind einige Änderungen

am Algorithmus erforderlich, um mit nicht wasserdichten Netzen kompatibel zu sein. Das zweite Problem ist der zusätzliche Schritt, der am Ende hinzugefügt wird, um den Raum zu diskretisieren und eine explizite Darstellung der Form zu extrahieren, z. B. ein Netz oder eine Punktwolke. Der letzte Schritt ist leider von der erforderlichen Maschengröße abhängig.

Anwendung beim Tiefziehen, Spannen und Fügen

Struktur des ML-Regressors

Wir haben auf der Idee impliziter Darstellungen aufgebaut, um die im Rahmen des Projekts gewonnenen Daten zu verarbeiten. Der Hauptvorteil gegenüber expliziten Ansätzen besteht darin, dass wir nicht mehr nur über einige hundert simulierte Experimente verfügen, um das neuronale Netz zu füttern (was höchstwahrscheinlich zu einer Überanpassung führt), sondern über Tausende von Punkten pro Experiment, die als einzelne Eingaben verwendet werden können. Außerdem verfügen wir bereits über einen Code, der die Form für die Tiefzieh-, Spann- und Fügeoperationen darstellt: die Prozessparameter, die zur Erstellung der FEM-Simulationen verwendet wurden. Auf der Grundlage der 1D-, 2D- oder 3D-Koordinaten eines Knotens in einem simulierten Netz können wir die entsprechende Ausgangsvariable (xyz-Koordinaten, Dicke, Ausdünnung, plastische Dehnung usw.) an dieser bestimmten Stelle vorhersagen (Abb. 47). Anstatt sich auf die nicht-euklidische Struktur des Eingaberaums zu verlassen und teure Operationen wie Faltungen auf dem gesamten Eingabernetz durchzuführen, gewährleistet dieser Ansatz eine effiziente Regularisierung des NN durch die Verwendung einzelner Positionen als Trainingsmuster. Dieser Ansatz erwies sich als sehr robust für die Modellierung von FEM-Simulationen in verschiedenen Dimensionalitäten. Nachdem das Netz trainiert wurde, können neue Werte für die Prozessparameter vom Benutzer definiert werden. Eingangspositionen (1D, 2D oder 3D) können abgetastet werden und die entsprechenden Simulationen können vom neuronalen Netz in wenigen hundert Millisekunden nachgebildet werden, ohne dass die entsprechende FEM-Simulation ausgeführt werden muss, so dass ein Ingenieur sehr schnell Gedankenexperimente und Robustheitsuntersuchungen durchführen kann.

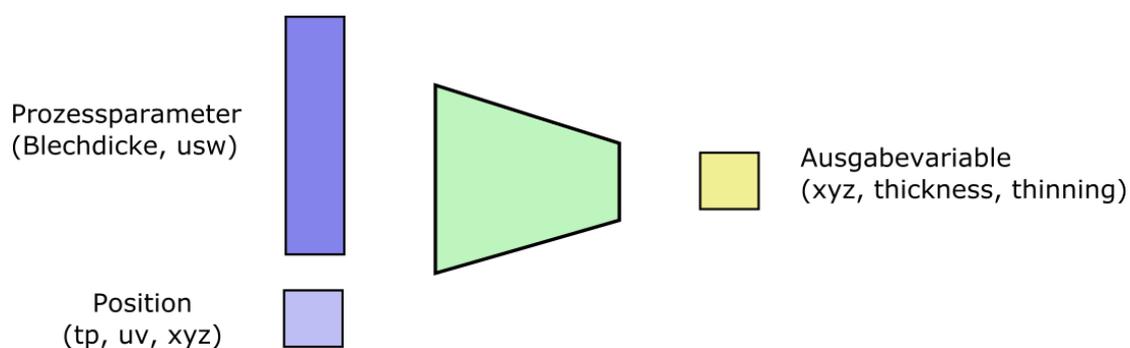


Abbildung 47: Struktur des ML-Regressors.

Bei dieser generischen Architektur bleiben mehrere Freiheitsgrade übrig: 1) Welche Prozessparameter haben die größte Bedeutung für die Vorhersage der Netztopologie? 2) Welche Architektur des neuronalen Netzes ist am besten für die verfügbaren Daten geeignet? Für die Tiefziehdaten wurde eine auf Random Forests basierende "Permutation Feature Importance"-Routine angewendet, um die wichtigsten Parameter zu identifizieren. Blechdicke, Niederhalterkraft, Ziehspalt, Einlegeposition, Ziehtiefe und R_p0 erwiesen sich als die am besten vorhersagenden Prozessparameter. Dies wurde dann an den individuell trainierten Modellen verifiziert, da das Hinzufügen anderer Merkmale die Leistung bei den Testdaten nicht verbesserte, während das Entfernen eines dieser Merkmale die Leistung beeinträchtigte. Diese Merkmale werden dann in den Experimenten zur Kodierung der Tiefziehform verwendet. Ziehtiefe und Ziehspalt nehmen kategoriale Werte an ($\{30, 50, 70\}$ für Ziehtiefe, $\{1.6, 2.4\}$ für Ziehspalt) und sind daher vor der Einspeisung in das neuronale Netz one-hot kodiert. Die anderen Merkmale sind standardisiert (Mittelwert 0, Varianz 1). Die Eingangspositionen, entweder 1D, 2D oder 3D, sowie die Ausgangsvariablen sind auf $[0, 1]$ normiert.

Die Architektur des neuronalen Netzes und seine Hyperparameter (Anzahl der Schichten und Neuronen, Lernrate, Dropout-Level) müssen an jeden Datensatz angepasst werden. Mit Hilfe der optuna-Bibliothek wurde ein AutoML-Ansatz auf der Grundlage der Bayes'schen Optimierung entwickelt. Das vollverbundene neuronale Netz kann zwischen 3 und 6 Schichten haben, wobei jede Schicht zwischen 64 und 512 Neuronen umfasst. Da das Training eines einzelnen Netzes nur wenige Minuten dauert, wurden in den folgenden Experimenten jeweils 100 Durchläufe gemacht, um die optimale Netzstruktur zu finden. 10 % der verfügbaren Simulationen wurden systematisch als Testsatz verwendet (leave-N-out cross-validation), da eine zufällige Auswahl von Positionen keine Erkennung einer Überanpassung ermöglichen würde. Der mittlere quadratische Fehler der Vorhersagen in den Testexperimenten wurde als Ziel der Optimierung verwendet. Das neuronale Netzwerk wird mit Tensorflow implementiert und so trainiert, dass der mittlere quadratische Fehler über die Trainingsmenge mit Hilfe des Adam-Regulierers für 100 Epochen minimiert wird.

Tiefziehen: 1d-ML-Modelle

Die Daten bestehen aus 879 Tiefziehsimulationen, bei denen jeweils 1000 Stichproben entlang des x_0 -Schnitts genommen werden. Jede Eingabeposition tp ist mit der entsprechenden Abweichung von der Referenzform sowie den entsprechenden y - und z -Koordinaten im 3D-Raum verbunden. Zusätzlich zu den 6 Prozessparametern hat das Netz also einen Eingang und drei Ausgänge. Die Vorhersage der Abweichung und der yz -Koordinaten mit separaten Netzen hat die Genauigkeit nicht verbessert.

Abb. 48 zeigt die Leistung des besten Netzes während und nach dem Training. In der Abbildung oben links liegen die Trainings- und Validierungsverluste nahe beieinander und

erreichen einen niedrigen Wert ($5e-5$), was trotz des Leave-N-Out Kreuzvalidierungsverfahrens keine Anzeichen für eine Überanpassung zeigt. In den drei anderen Diagrammen werden die vorhergesagten Ausgabewerte mit der Grundwahrheit auf dem Testset verglichen. Die Abweichung weist mehr sichtbare Fehlprognosen auf als bei y und z, aber dies betrifft nur eine kleine Minderheit der Testdaten.

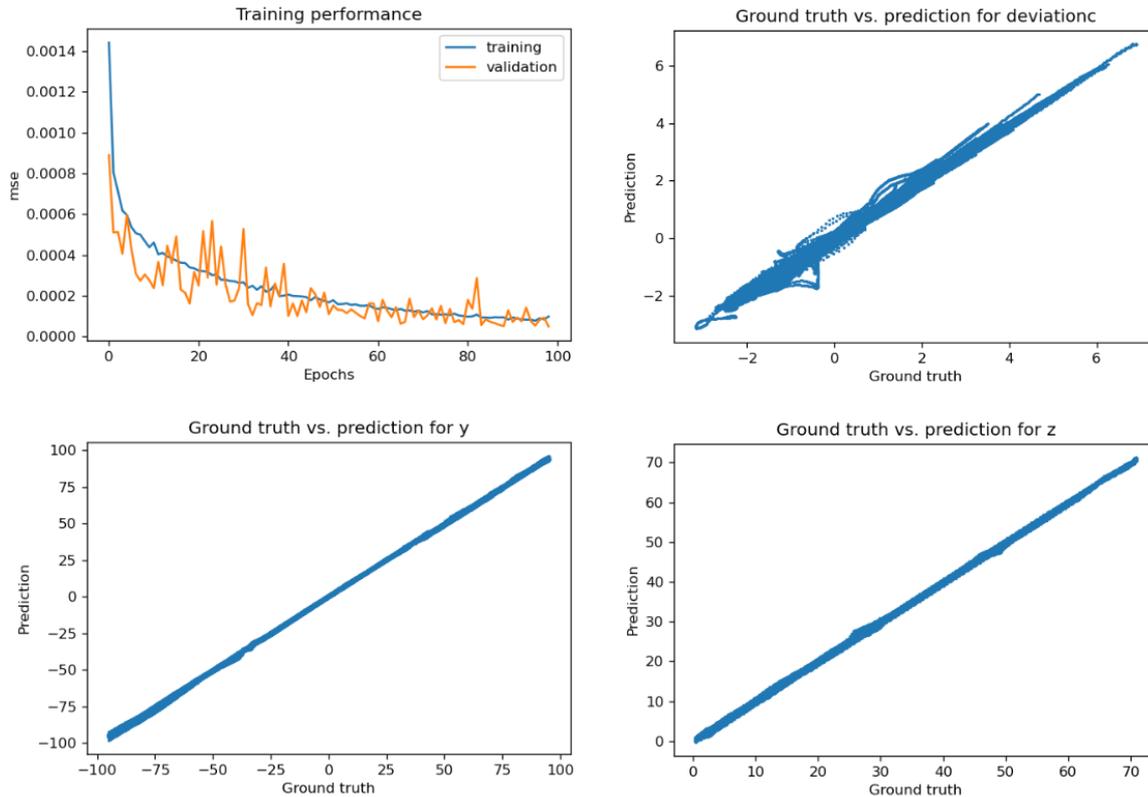


Abbildung 48: Oben links: Entwicklung des Trainings- und Testverlusts während der 100 Trainingsepochen (batch size 2048). Die drei anderen Diagramme zeigen die vorhergesagten Werte für Abweichung, y und z gegen die Grundwahrheit im Testset. Der Idealfall ist eine gerade diagonale Linie.

Abb. 49 zeigt die Vorhersage für ein einzelnes Experiment im Testset (#883). Die Fehlprognosen treten meist an stark gekrümmten Stellen auf, beeinträchtigen aber nicht die Vorhersage der allgemeinen Form des Schnitts.

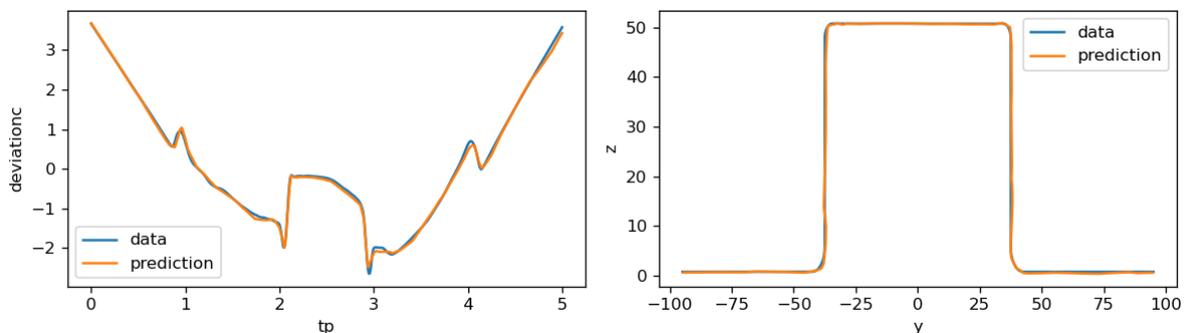


Abbildung 49: Grundwahrheit und Vorhersage des x0-Schnitts für das Testexperiment #883.

Tiefziehen: 2d-ML-Modelle

Wir verwenden nun eine abgeflachte 2D-Darstellung der Oberfläche der Tiefziehexperimente unter Verwendung der von der Scale GmbH erstellten uv-Koordinatensysteme. Es wurden 725 Tiefziehexperimente verwendet (mit den gleichen Prozessparametern wie zuvor), von denen 72 das Testset bildeten. Die uv-Koordinaten sind bereits zwischen 0 und 1 skaliert. Zwei separate Netzwerke wurden optimiert und trainiert, um 1) die xyz-Koordinaten des einer uv-Position entsprechenden Knotens und 2) die entsprechende Dicke, Ausdünnung und plastische Dehnung vorherzusagen. Beide Netze konvergierten mit einer hohen Genauigkeit (unter $1e-5$) und zeigten keine Anzeichen von Überanpassung. Kombiniert man die beiden Netze, kann man nicht nur die Form des Tiefziehteils für jede beliebige Stelle auf der Oberfläche vorhersagen (da u und v gleichmäßig abgetastet werden können), sondern auch die entsprechenden Bauteileigenschaften wie Dicke, Ausdünnung und plastische Dehnung (Abb. 50).

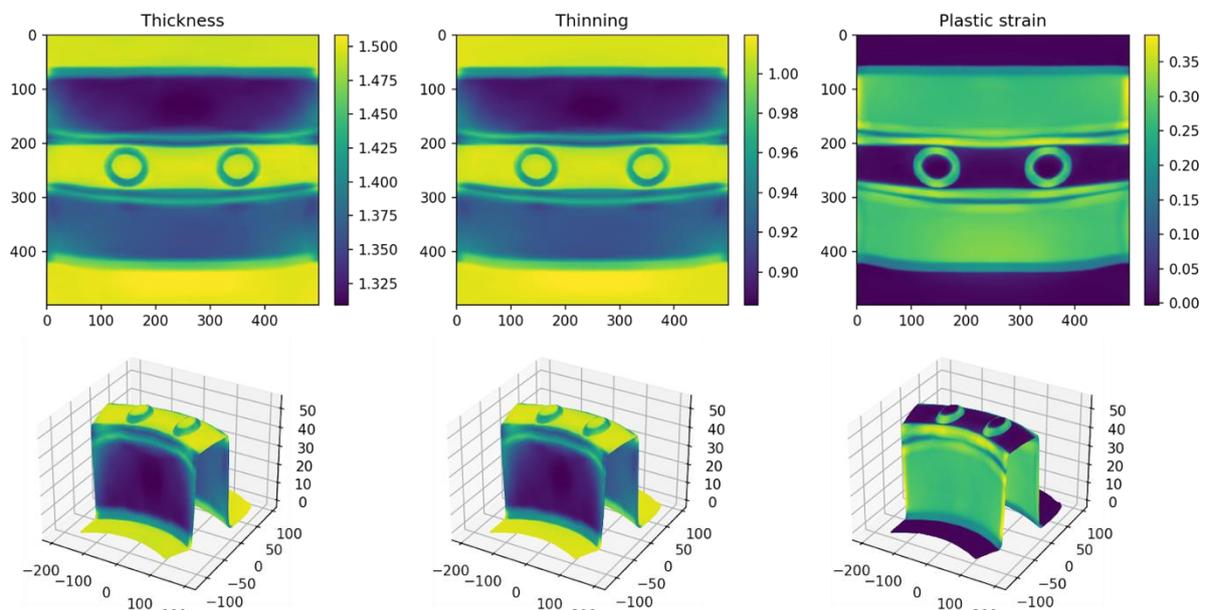


Abbildung 50: 2D-Vorhersage von Dicke, Ausdünnung und plastischer Dehnung für das Testexperiment #883. Oben: vorhergesagter Wert im uv-Raum. Unten: Abbildung auf die vorhergesagten xyz-Koordinaten.

Tiefziehen: 3d-ML-Modelle

Wir untersuchen nun, ob es möglich ist, die Bauteileigenschaften direkt aus den xyz-Koordinaten vorherzusagen. Das Lernen ist zwar möglich, stellt aber ein Problem für die Inferenzphase mit neuen Prozessparametern dar, da die xyz-Koordinaten jedes Knotens unbekannt sind. Die Bauteileigenschaften können jedoch auf das Referenznetz projiziert werden (für die drei Ziehtiefen 30, 50 und 70 mm), wobei die Knoten feste xyz-Koordinaten haben, die als Eingabe des Modells während der Inferenz verwendet werden können. Abb. 51 zeigt die Vorhersage des 3D-Modells für die Abweichung und Dicke eines einzelnen Versuchs

(#883), für die xyz-Koordinaten der Knoten des Referenznetzes (Ziehtiefe 50mm). Im Vergleich zu Abb. 50 ist die vorhergesagte Dicke in Abb. 51 ähnlich, aber die Seiten der Form nehmen ungewöhnliche Werte an. Dies ist auf die Abbildung des tatsächlichen Meshes auf das Referenzmesh zurückzuführen, das aufgrund des verwendeten “nearest neighbours” Algorithmus anormale Werte erzeugt. Aus diesem Grund kommen wir zu dem Schluss, dass die 2D-UV-Projektion für die Darstellung der Tiefziehexperimente besser geeignet ist und behalten diesen Ansatz für die Spannen- und Füge-experimente bei.

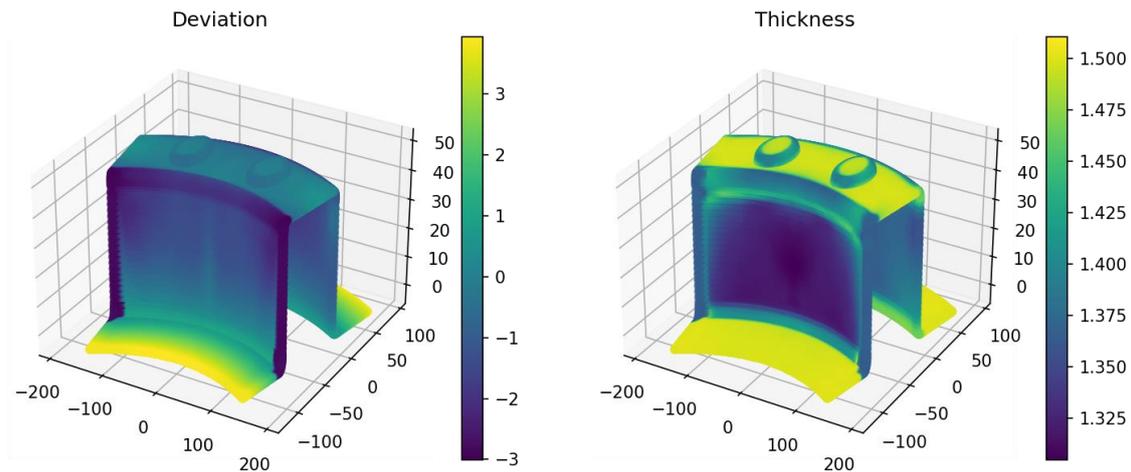


Abbildung 51: 3D-Vorhersage von Ausweichung und Dicke für das Testexperiment #883.

Spannen und Fügen: 2d-ML-Modelle

Die Spann- und Füge-Operationen bestehen darin, zwei tiefgezogene Formen zu nehmen und sie zusammenzufügen. Das obere Teil hat immer eine Ziehtiefe von 50 mm, während das untere Teil entweder 30, 50 oder 70 mm hoch ist. Zusätzlich zu den Prozessparametern des Tiefziehens jedes Teils wird der Vorgang durch die Z-Position der vier Spanner sowie eine Clinch-ID für jedes Teil charakterisiert. Damit ergeben sich insgesamt 18 Prozessparameter für den Fügevorgang, die 321 erfolgreichen Simulationen beschreiben. 32 Versuche werden zu Testzwecken aufbewahrt. Die Ober- und Unterteile haben ihre eigenen uv-Koordinatensysteme. Da wir die Eigenschaften der Ober- und Unterteile getrennt vorhersagen, wird je nach Teil ein zusätzlicher binärer Index an uv gekoppelt. Wie zuvor werden zwei Netze optimiert und trainiert, um 1) die xyz-Koordinaten der Knoten des Teils und 2) die Bauteileigenschaften vorherzusagen. Zum Zeitpunkt der Inferenz werden die Prozessparameter festgelegt, u und v werden gleichmäßig abgetastet und der Index wird abwechselnd auf 0 und 1 gesetzt, um die jeweiligen Größen der beiden Teile vorherzusagen.

Abb. 52 zeigt die Vorhersage der beiden Netze für die Geometrie der zusammengefügte Teile und die entsprechenden Bauteileigenschaften für ein völlig neues Design (die Unter- und Oberteile wurden in den Trainingsdaten nie zusammengespannt). Trotz der getrennten

Vorhersage stimmen die vorhergesagten Koordinaten für die beiden Teile überein, insbesondere in den Kontaktzonen.

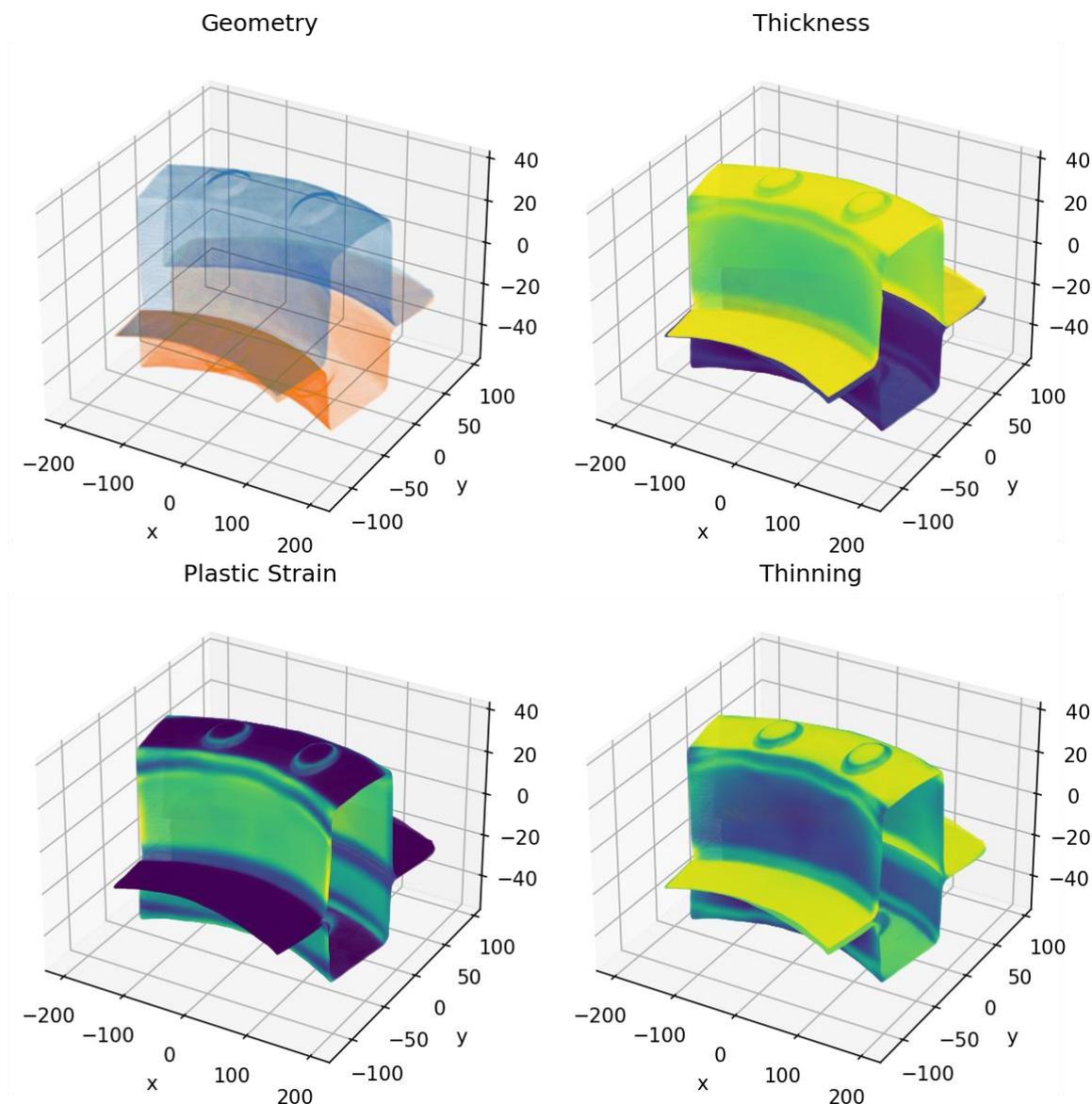


Abbildung 52: 3D-Vorhersage für ein neues Testexperiment: Spanner [-3, -3, -5, 3], MID [5, 3], Oberblech (Dicke 1.48, Niederhalterkraft 400, Ziehspalt 2.4, Einlegeposition -4, Ziehtiefe 50, RP0 235.160326), Unterblech (Dicke 1.01, Niederhalterkraft 410, Ziehspalt 2.4, Einlegeposition -5, Ziehtiefe 30, RP0 138.226960).

Neben der einfachen Verwendung als Eingabe für das neuronale Netz zur Vorhersage von 3D-Formen und -Eigenschaften hat das uv-Koordinatensystem auch den Vorteil, dass die Formen entlang von 1D-Achsen geschnitten werden können, indem u oder v auf einen festen Wert gesetzt werden. Abb. 53 zeigt den x0-Schnitt, der aus dem vorherigen Fügeexperiment abgeleitet wurde, sowie die Taschenbildung.

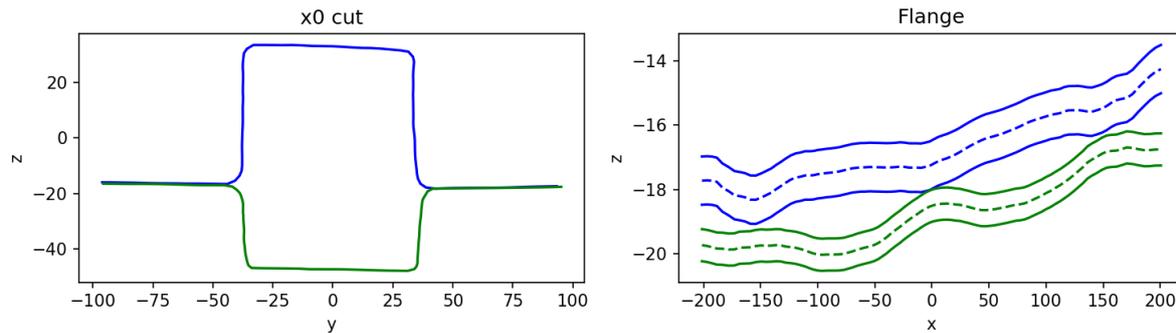


Abbildung 53: x0-Schnitt und "Taschenbildung" zwischen den gefügten Bauteilen.

Suche nach optimalen Prozessparametern

Nach dem Training der Netze müssen nur noch die Prozessparameter eingegeben werden, da die Eingangsvariablen tp oder uv abgetastet werden. Scale GmbH hat eine Prognose-App entwickelt, die es den Ingenieuren ermöglicht, mit den Modellen zu interagieren und die verschiedenen Bauteileigenschaften zu visualisieren, ohne die teuren FEM-Simulationen durchführen zu müssen (Digital Twin). Die Tatsache, dass die neuronalen Netze nur einige hundert Millisekunden benötigen, um eine Vorhersage zu treffen, ermöglicht es uns auch, automatisch nach den Prozessparametern zu suchen, die ein bestimmtes Kriterium minimieren (z. B. Abweichung an bestimmten Stellen, Taschengröße usw.). Dies lässt sich mit Hilfe der Bayes'schen Optimierung über Tausende von Versuchen leicht bewerkstelligen. Abb. 54 zeigt ein Beispiel, bei dem die Blechdicke auf 1,01 mm festgelegt wurde, die anderen Parameter sich jedoch in ihrem Bereich frei bewegen konnten. Das 1D-Modell wird verwendet, um die Abweichung entlang des x0-Schnitts vorherzusagen, und der Optimierer versucht, entweder die mittlere absolute Abweichung (damit das Teil im Durchschnitt am korrektesten ist) oder die maximale absolute Abweichung (um die "Flügel" zu vermeiden) zu minimieren. Vergleichen Sie das Ergebnis dieser Optimierung mit Experiment #889 (Abb. 49).

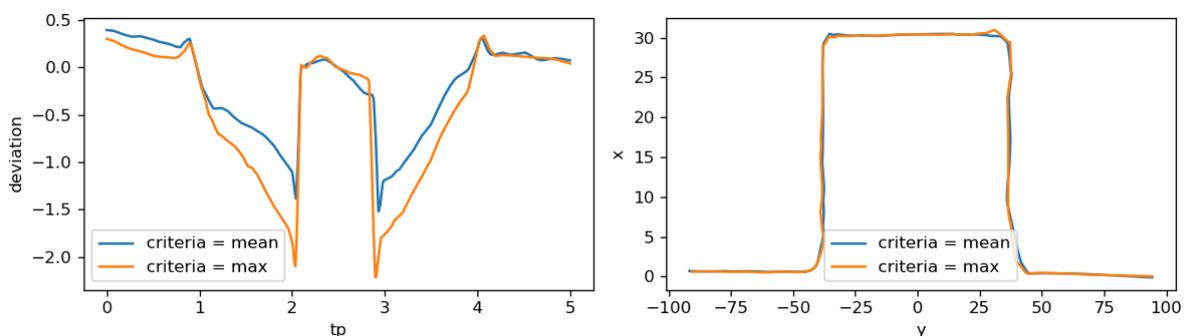


Abbildung 54: Ergebnis der Optimierung des x0-Schnitts im Hinblick auf die maximale bzw. mittlere absolute Abweichung. Die Blechdicke ist mit 1.01mm konstant.

Für die Fügeversuche konnten mehrere Kriterien verwendet werden. In Abb. 55 haben wir als Kriterium den durchschnittlichen Abstand zwischen den Flügeln der beiden Teile entlang des x0-Schnitts gewählt. Dies führt dazu, dass die beiden Flügel zumindest entlang des x0-Schnitts

fest zusammengeklemt werden. In Abb. 56 wird die Taschenbildung stattdessen entlang des Flansches minimiert. Bauingenieure könnten verschiedene Kriterien kombinieren, um die bestmöglichen zusammengefügte Teile zu erhalten.

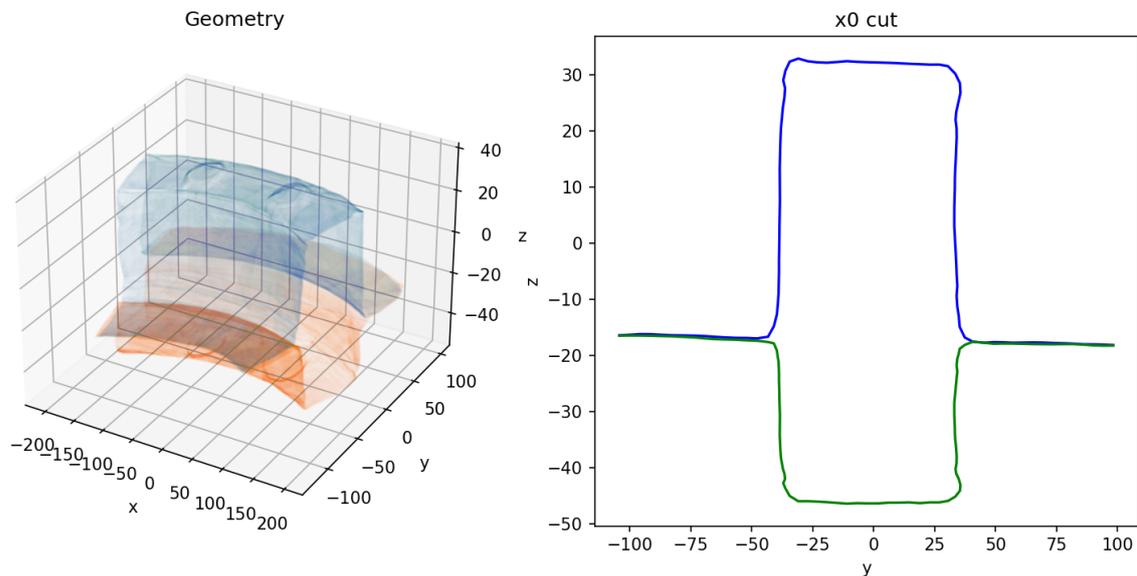


Abbildung 55: Ergebnis der Optimierung des x0-Schnitts. Die Blechdicke ist mit 1.01 mm konstant, die Ziehtiefe ist 30 mm unten, 50 mm oben.

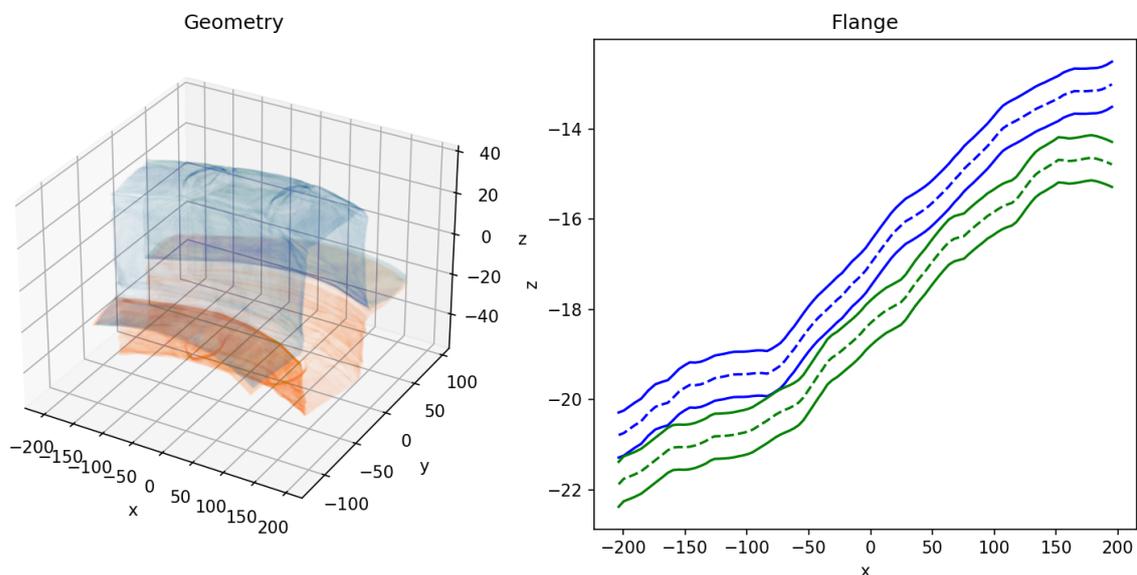


Abbildung 56: Ergebnis der Optimierung der Taschenbildung. Die Blechdicke ist mit 1.01mm konstant, die Ziehtiefe ist 30mm unten, 50mm oben.

Code-Repository

Der Code, der die Vorhersage von Bauteileigenschaften aus FEM-Simulationen ermöglicht, wurde als freie Bibliothek (MIT-Lizenz) organisiert, die für andere Datensätze wiederverwendet werden kann: <https://github.com/hamkerlab/ML-Karoprod-MeshPredictor>. Die Design-of-Experiments-Tabellen und die Daten für jedes Experiment müssen als Pandas Dataframes zur

Verfügung gestellt werden, aber der Rest läuft automatisch ab: Vorverarbeitung der Eingaben, Training des neuronalen Netzes mit Tensorflow, Bayes'sche Optimierung mit Optuna, usw. Diese Bibliothek kann ein sehr wertvolles und vielseitiges Werkzeug für Materialingenieure sein, die ihre eigenen FEM-Simulationen überprüfen und optimieren wollen. Die Dokumentation ist verfügbar unter <https://hamkerlab.github.io/ML-Karoprod-MeshPredictor/>.

3.4 Fazit

Die gesteckten Ziele konnten erreicht werden. Es ist gelungen den Prozess vollständig experimentell und virtuell abzubilden. Es wurde eine komplexe umfangreiche Prozesskette geplant, konstruiert und aufgebaut. Nach Inbetriebnahme konnten virtuelle Einzelschrittmodelle zur kompletten automatisierten virtuellen Prozesskette verbunden werden. Mit dieser erfolgte in der Datenakquise die automatische Generierung vieler Datensätze, die in den ML-Modellen genutzt wurden. Mit den Methoden des maschinellen Lernens können nicht nur selektive Bauteileigenschaften prognostiziert werden, sondern auch die gesamte Bauteilgeometrie (inkl. der Blechdickenverteilung und plastische Verformungen) in Echtzeit vorhergesagt werden. Der Prozessschritt des Tiefziehens konnte vollständig durch ein ML-Modell ersetzt werden. Dies übertrifft die Ziele des Antrages signifikant. Bemerkenswert sind die außergewöhnlich guten Prognosegüten der entwickelten 1d-, 2d- und 3d-Machine-Learning-Modelle für die Prozesse Tiefziehen und Fügen.

Es sind intuitiv benutzbare, webbasierte Software-Applikationen in dem Projekt entwickelt worden, welche prototypisch die neuen Möglichkeiten einer KI-unterstützten Prozessauslegung veranschaulichen. Hierbei wurde besonders auf eine einfache Übertragbarkeit der Methodik auf weitere Anwendungsgebiete geachtet.

Die Echtzeit-Prognose der Bauteileigenschaften mit den entwickelten Software-Applikationen kann die Produktentwicklung wesentlich beschleunigen. Etablierte Finite-Element-Simulationen würde erst nach mehreren Stunden vergleichbare Ergebnisse liefern können. Die Anwendung der entwickelten Softwaretools führt zu einem besseren und schnelleren Modellverständnis bei der Prozessauslegung. Dies führt neben einer besseren Nutzerakzeptanz zu qualitativ hochwertigeren Bauteilen und robusteren Herstellungsprozessen.

4 Wichtigste Positionen des zahlenmäßigen Nachweises

Wesentliche Positionen des zahlenmäßigen Nachweises im Projekt ML@KaroProd für das IWU bestanden in den Personalkosten, Kosten für Fremdleistungen, Material- und Reisekosten. Die Personalkosten machten laut Projektplan 91.3 % der IWU-Projektkosten

aus. Eine Veränderung im Projektverlauf war nicht nötig. Die Kosten für Fremdleistungen betragen 4.3 % der Gesamtkosten und mussten auf 6.1 % erhöht werden. Der Grund hierfür lag in der Anpassung der realen Prozesskette, um eine erhöhte Serienprozessnähe zu erreichen. Der Anteil der Materialkosten am Gesamtvolumen war mit 2.7 % gering und konnte durch Rückgriff auf am IWU bestehendes Material auf 2.4 % reduziert werden. Die Reisekosten waren mit 1.7 % der Gesamtprojektkosten geplant. Infolge der Corona-Pandemie fanden fast alle der Projektmeetings und geplanten Teilnahmen an Konferenzen online statt bzw. wurden im Falle der Konferenzen verschoben. Dadurch sank der Anteil an den Gesamtkosten auf 0.3 %.

Bei SCALE fanden gegenüber der Planung keine Veränderungen der wesentlichen Kostenposition (Personalkosten 100 %) in der Projektbearbeitung statt.

Der Personalkostenanteil am Gesamtvolumen des Projekts der TU Chemnitz betrug 93.7 %, die Reisekosten 2.6 %. Der Übertrag von Reise- in Personalkosten aus ebenfalls den oben genannten Gründen und entstandenem personellen Mehraufwand erhöhte den Personalkostenanteil auf 96.2 %. Es entfielen 3.7 % auf zu beschaffende Rechentechnik, die in dieser Form realisiert werden konnte.

Jeweils in Summe wurden am IWU, SCALE und TUC die Kostenplanungen eingehalten.

5 Notwendigkeit und Angemessenheit der geleisteten Projektarbeiten

Im Projekt fanden wie geplant Methoden des Maschinellen Lernens Anwendung. Diese müssen üblicherweise mit umfangreichen Datensätzen trainiert und verifiziert werden bevor sie bei weiteren interessierenden Datensätzen für das Auffinden von Zusammenhängen nutzbar sind. Hierfür erfolgte die Planung, Konstruktion und Einarbeitung einer Prozesskette zur Baugruppenfertigung. In der Prozesskette verwendete Werkzeuge wurden über Fremdleistung beschafft. Mit der physischen Prozesskette konnten Finite-Elemente-Modelle der gesamten Fertigungslinie aufgebaut und verknüpft werden, mit der die Datensätze für das Maschinelle Lernen bereitgestellt werden konnten. Ein nicht zu vernachlässigender Anteil entfiel bei den Arbeiten auf Datenstandardisierung, -strukturierung, Aufbau von Datenbanken und Skripting, da bei dem großen numerischen Aufwand alle Berechnungsstarts samt Übergabe von Startparametern, Informationsfluss zwischen den Einzelberechnungen und Simulationsauswertung weitestgehend automatisiert ausgeführt werden müssen und für das Maschinelle Lernen eine strikte Datenstrukturierung vorliegen muss. Ohne das Programmieren dieser Skripte und Anwendung von weiteren informationstechnischen

Methoden wäre das Zeitziel nicht zu halten gewesen. Alle Arbeiten für das dargestellte Vorgehen waren nötig und angemessen, um die Projektziele zu erreichen.

6 Nutzen für das Unternehmen, insbesondere Verwertbarkeit des Ergebnisses

Am IWU konnte eine durchgängige Prozesskette real und als virtuelles Abbild mit Finite-Elemente-Modellen entwickelt werden. Mit der realen Prozesskette kann künftig auf weitere Aspekte im Umfeld der Automobilproduktion wie Einfluss von Fügetechnologien oder Positionierungsstrategien aber auch in anderen Branchen wie Anlagenbau, Klimatechnik und Brennstoffzellenfertigung fokussiert und intensiv unter kontrollierten Bedingungen untersucht werden. Die Prozesskette am IWU eröffnet die Möglichkeit, alle interessierenden Einstellungen vor Ort am IWU vorzunehmen und deren Wirkung zu analysieren, sodass laufende Serienprozesse beim Kunden nicht für die Sensorierung und Datenaufnahme unterbrochen werden müssen. Für den virtuellen Prozesskettenpfad konnten flexible Prozesskettenteilmodelle auf Finite-Elemente-Basis automatisiert erstellt und miteinander verknüpft werden. Die Datenauswertung und –weiterleitung erfolgt ebenfalls automatisiert, was eine umfangreiche Datenakquise absichert. Die dabei erstellten Skripte und Programme zur Informationsauswertung und –management können in Folgeprojekten nachgenutzt werden. Aus dem Projekt heraus entstand die Software MetaField, mit welcher Einflüsse von Sensitivitäten visualisiert und mit Geometrie einschränkungen auch Vorhersagen auf neue Parameterkombinationen getroffen werden kann. Die Software wurde mittlerweile erfolgreich in den Bereichen Kunststoffspritztechnik und Schweißnahtausbildung an weiteren Fallbeispielen getestet. Eine künftige Zusammenarbeit mit Teilnehmern dieser Märkte ist geplant.

Die im Rahmen des Projektes bei SCALE entwickelten Datenaufbereitungsmethoden von Finite-Element-Simulationsergebnissen ermöglichen die Benutzung von Maschine-Learning-Modellen, welche bisher bei der simulationsgestützten Produktentwicklung keine Anwendung finden konnten. Insbesondere von der Bereitstellung des praxisnahen Beispielprozesses erhoffen wir uns eine bessere Kundenakquise, da der potenzielle Kunde die Leistungsfähigkeit der SCALE-Softwarelösungen direkt testen kann. Ferner liefern die außergewöhnlich guten Prognosegüten der entwickelten ML-Modelle hinsichtlich der Bauteilverformungen und der Bauteileigenschaften (z.B. Blechdicken, plastische Vorverformungen infolge der Fertigung) dem Entwicklungsingenieur ein besseres Verständnis bei der Produktentwicklung und erhöhen die Nutzerakzeptanz der entwickelten Methoden signifikant.

Die Allgemeingültigkeit des Ansatzes für die Modellierung von FEM-Simulationen eröffnet viele Möglichkeiten für Forschung und Entwicklung in den Ingenieurwissenschaften. Die TUC hat

das erworbene Know-how im Jahr 2021 in einer Zusammenarbeit mit Porsche Leipzig eingesetzt, mit dem Ziel, die Positionierung verschiedener Autoteile in der Produktionslinie zu unterstützen. Darüber hinaus ermöglichte die erworbene Expertise der TUC, auf den BMBF-Aufruf zu KI-Servicezentren mit einem gemeinsam mit Fraunhofer ENAS und IWU verfassten Antrag "Edge-AI und Smart Manufacturing" zu reagieren und plant, lokale Unternehmen im Raum Chemnitz bei eingebetteten KI-Projekten zur Prozessoptimierung, Planung und Steuerung zu unterstützen. TUC akquirierte während des Projekts zwei weitere BMBF-geförderte Projekte, WAIKIKI zur Cybersicherheit in smart grids und SmartAirSense zur intelligenten Steuerung von IoT-Luftsensoren, die auf dem im Rahmen von ML@Karoprod erworbenen Wissen aufbauen.

Direkt aus dem Projekt heraus konnte eine Vielzahl an Veröffentlichungen realisiert werden (s. Kapitel 8). Weitere werden derzeit eingereicht bzw. befinden sich in Planung. Aufbereitete Ergebnisse des Projekts werden an den Standorten Chemnitz (TU) und Dresden (HTW) in die Lehre überführt. Messeauftritte und Konferenzteilnahmen wurden Corona-bedingt zurückgestellt und werden aktuell nachgeholt.

7 Darstellung des während des Vorhabens bekannt gewordenen Fortschritts auf diesem Gebiet bei anderen Stellen

Es sind keine wesentlichen Fortschritte auf diesem Gebiet bei anderen Stellen bekannt geworden.

8 Veröffentlichungen, Vorträge, Referate

Veröffentlichungen

Atoofi, P., Vitay, J., Hamker, F.H. (2021). Geometric Deep Learning: Graph Neural Networks, Challenges, and Breakthroughs. Workshop 3D-NordOst 2021. Tagungsband 23. Anwendungsbezogener Workshop zur Erfassung, Modellierung, Verarbeitung und Auswertung von 3D-Daten, Berlin, 02./03.12.2021:115-124. ISBN: 978-3-942709-27-9.

Farahani, A., Vitay, J., Hamker, F.H. (2021). Geometric Deep Learning and solutions for the industry. Workshop 3D-NordOst 2021. Tagungsband 23. Anwendungsbezogener Workshop zur Erfassung, Modellierung, Verarbeitung und Auswertung von 3D-Daten, Berlin, 02./03.12.2021: 105-113. ISBN: 978-3-942709-27-9.

Schwarz, C., Kriechenbauer, S., Mauermann, R., Drossel, W. G. Field meta modelling for process design in complex sheet metal forming. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2022. S. 012069.

Kriechenbauer, S., Müller, P., Mauermann, R., Drossel, W. G. Evolutionary optimization of deep-drawing processes on servo screw presses with freely programmable force and motion functions. *Procedia CIRP*, 2021, 104. Jg., S. 1482-1487.

Kriechenbauer, S., Jäckel, M. Karosserieproduktion: Bessere Bauteilqualität dank Maschinellem Lernen. Jahresbericht 2019/2020, IWU, Chemnitz

In press:

Farahani, A., Vitay, J., Hamker, F.H. (accepted). Deep Neural Networks for Geometric Shape Deformation. 45th German Conference on Artificial Intelligence, September 19th-23rd 2022, Trier (Germany).

Farahani, A., Vitay, J., Kriechenbauer, S., Hamker, F.H. (accepted). A Deep Learning Approach to Predict the Deviation of Sheet Metal Parts in Deep Drawing Process. 16th LS-DYNA Forum, October 11th-13rd 2022, Bamberg (Germany).

Kriechenbauer, S.; Hensel, S., Lepenies, I. (accepted). Framework für die numerische Sensitivitätsanalyse von Prozessketten zur Herstellung komplexer Karosseriebauteile. 16th LS-DYNA Forum, October 11th-13rd 2022, Bamberg (Germany).

Vorträge

Das Projekt wurde zum BMBF-Statustreffen beim All-Hands-Meeting in Dortmund am 05. Juni 2019 präsentiert.

Der im Projekt erarbeitete Ansatz wurde am 06. Februar 2020 auf dem von der IHK Dresden organisierten Follow Up Workshop Künstliche Intelligenz (<https://www.energy-saxony.net/veranstaltungen/follow-up-kuenstliche-intelligenz.html>) vorgestellt. Der Workshop versammelte 60 Teilnehmer aus dem sächsischen industriellen Umfeld, die großes Interesse an unserem Ansatz zeigten.

Für die Internationale Conference on Numerical Simulation of 3D Sheet Metal Forming Processes (NUMISHEET 2022) ein Abstract eingereicht worden, welches akzeptiert wurde („“). Eine Veröffentlichung erfolgt somit demnächst.

Websites

<https://www.scale.eu/de/aktuelles/forschungsprojekte/ml-karoprod>

<https://www.autoland-sachsen.com/selbstlernende-karosseriefertigung/>

<https://github.com/hamkerlab/ML-Karoprod-MeshPredictor>

Literatur

Ackert, P.; Schwarz, C.; Mauermann, R.: Geometric Accuracy of a Design Based Pro-cess Chain Simulation in Comparison th the Results of Real Car Body Manufacturing. Automotive Engineering Congress, Nürnberg, 2017.

Ahmed, E., Saint, A., Shabayek, A. E. R., Cherenkova, K., Das, R., Gusev, G., Aouada, D., & Ottersten, B. (2019). A survey on Deep Learning Advances on Different 3D Data Representations (arXiv:1808.01462).

Eckert, A.: Prognose der Maßhaltigkeit punktförmig mechanisch gefügter Karosseriean-bauteile. Dissertation, Chemnitz 2012.

Jäckel, M.; Falk, T.; Landgrebe, D.: Concept for Further Development of Self-pierce Riveting by Using Cyber Physical Systems. In: Procedia CIRP, Jg. 44 (2016); 293-297.

Kriechenbauer, S. Landgrebe, D.: Methode zur Auslegung von Sonderverfahren in der Blechumformung mit hoher Prozessdynamik am Beispiel der Kissen-Stößel-Pulsation, Hannover: EFB, 2016.

Kriechenbauer, S. (2022). Beitrag zur Analyse und Synthese von variablen Kraft- und Wegfunktionen beim Tiefziehen auf Servo-Spindelpressen. Berichte aus dem IWU. Band 122. Verlag Wissenschaftliche Scripten. Auerbach/Vogtland

Lorensen, W. E., & Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. ACM SIGGRAPH Computer Graphics, 21(4), 163–169. doi:10.1145/37402.37422

Park, J. J., Florence, P., Straub, J., Newcombe, R., & Lovegrove, S. (2019). DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. (arXiv:1901.05103).

Ranjan, A., Bolkart, T., Sanyal, S., & Black, M. J. (2018). Generating 3D faces using Convolutional Mesh Autoencoders. (arXiv:1807.10267).

Schwarz, C.; Ackert, P.; Mauermann, R.: Principal component analysis and singular value decomposition used for a numerical sensitivity analysis of a complex drawn part. In: The International Journal of Advanced Manufacturing Technology, 2017, DOI: 10.1007/s00170-017-0980-z

Schwarz, C., Ackert, P. & Mauermann, R. Principal component analysis and singular value decomposition used for a numerical sensitivity analysis of a complex drawn part. Int J Adv Manuf Technol 94, 2255–2265 (2018). doi: 10.1007/s00170-017-0980-z

Kurzbericht zum Projekt ML@Karoprod

- Vorhaben:** Maschinelles Lernen zur Prognose von Prozessparametern und Bauteilqualität in der automobilen Karosserieproduktion (ML@Karoprod)
- Förderprogramm:** IKT 2020 – Forschung für Innovationen
- Förderkennzeichen:** 01IS18055
- Durchgeführt von:** Fraunhofer IWU
SCALE GmbH
TU Chemnitz
- Projektlaufzeit:** 12/2018 - 01/2022
- Projektleiter:** Dr. Sebastian Kriechenbauer
Dr. Sebastian Hensel
Dr. Ingolf Lepenies
Prof. Dr. Fred Hamker
Dr. habil. Julien Vitay

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Aufgabenstellung

Wie in anderen Industriezweigen auch, verspricht der Einsatz von Methoden des Maschinellen Lernens (ML-Methoden) einen schnelleren Aufbau von Prozessen und ganzer Prozessketten einerseits sowie eine Verbesserung von Regelungsstrategien zur Aufrechterhaltung und Erhöhung der Qualität von Produkten andererseits. Im Projekt soll eine Zusammenbauprozesskette analysiert und durch eine virtuelle Entsprechung in Berechnungsmodellen nachgebildet werden. In der Prozesskette erfolgt die Herstellung von Einzelteilen und der Zusammenbau, in dem die Teile durch mechanisches Fügen dauerhaft miteinander zu einer Baugruppe verbunden werden. Hierbei kann es durch streuende Prozesse und Materialien zu Abweichungen von der geplanten Geometrie kommen. Im Projekt sollte in einem ersten Schritt die Prozesskette eingehend untersucht werden. Mit genauer Kenntnis der Einzelprozesse sollte dann die Gesamtprozesskette mit Finite-Elemente-Methoden modelliert und ganz gezielt Schwankungen in den Einzelprozessschritten eingebracht werden. Die hierbei anfallenden Daten liefern die Basis für das Trainieren von ML-Modellen für die Gesamtprozesskette, mit denen die wichtigsten Eingangsgrößen mit den interessierenden Ausgangsgrößen korreliert werden.

Wissenschaftlich-technischer Stand

Bisher erfolgt der Aufbau und das Einfahren von Prozessketten mit hohem Aufwand, um robuste Prozessparameter aufzufinden. Bereits in der Planungsphase vorher unterstützen Finite-Element-Modelle in Einflussanalysen. Zum Teil kann das mit sehr komplexen Gesamtprozessketten-Modellen mit Finiten Elementen geschehen. Die Modellerstellung hierfür ist aufwändig und die Berechnungszeiten immens groß. Insbesondere Prozesskettenschwankungen lassen sich nur durch umfangreiche Sensitivitätsstudien integrieren. Beim Betrieb des realen Produktionsprozesses kann es zur kritischen und dauerhaften Veränderungen von Parametern kommen, wodurch es nötig wird, diese Veränderungen durch Stellmaßnahmen zu kompensieren. Auch dieser Vorgang ist hochkomplex und erfordert viel Expertenwissen. Moderne Methoden des maschinellen Lernens, wie z. B. Deep Networks, erfordern feste Eingaben mit bekannten Topologien, was nicht gut zu Finite-Element-Simulationen passt. Im Laufe des Projekts wurden Methoden des geometrischen Deep Learning entwickelt, um mit solchen unregelmäßigen Datentypen umzugehen, wobei entweder explizite oder implizite gelernte Darstellungen verwendet werden.

Ablauf des Vorhabens

Zu Beginn des Projekts fanden Planung, Konstruktion und Anfertigung von Komponenten für eine Prozesskette im Labormaßstab statt. Der Produktionsablauf umfasst Einzelteilerfertigung mittels Tiefziehen, Positionieren der Teile sowie Spann- und Fügeprozesse. Die Prozessket-

te aus den Komponenten aufgebaut, getestet und eingefahren. Zudem erfolgte die Anfertigung der Spann-/Fügevorrichtung und weiterer Messvorrichtungen zur Geometrievermessung an verschiedenen Punkten in der Kette zur Aufnahme der sich einstellenden Geometrieabweichungen. Entsprechend eines Versuchsplans konnten die geplanten Experimente entlang der Prozesskette mit variierenden Eingangsgrößen durchgeführt werden. Parallel erfolgte die Modellierung aller Einzelprozesse mit Finite-Elemente-Modellen und deren Verknüpfung zum Gesamtprozess. In einem Datenakquisearbeitspunkt fand mit den Modellen die Generierung von vielen Designs mit unterschiedlichen Parameterkonfigurationen entsprechend numerischem Versuchsplan statt. Insbesondere der Informationsfluss zwischen den Einzelprozessschritten, die Ergebnisauswertung der Simulationen sowie die Datenstrukturierung und -ablage mussten sorgfältig und hochflexibel organisiert werden, da die Datenbeschaffung automatisiert erfolgt. Für das Anlernen von ML-Algorithmen müssen zudem hochstrukturierte Daten bereitgestellt werden, die von Design zu Design identische Positionen für auswertbare Ergebnisgrößen, z. B. Geometrieabweichungen, auf den Bauteilen aufweisen. Dies steht einer üblichen Finite-Elemente-Prozesssimulation entgegen, da infolge lokaler Netzverfeinerung sehr unterschiedliche Auswertepositionen auf den Bauteilen entstehen. Somit musste ein Algorithmus programmiert werden, der diese Schwierigkeiten überwindet. Dieser basiert auf einer Dimensionsreduktion aus dem 3d- in den 2d-Raum in Verbindung mit einem Mapping der Ergebnisdaten auf ortsfeste Punkte im 2d-Raum. Hiermit konnten diverse ML-Methoden getestet und bewertet werden. Die vielversprechendsten Methoden können basierend auf den Eingangsdaten sehr zuverlässig die interessierenden Ausgangsdaten der Prozesskette vorhersagen.

Wesentliche Ergebnisse

Es konnte eine komplexe Prozesskette geplant, umgesetzt und untersucht werden, um Prozesskettenteilmodelle aufzubauen. Nach Verknüpfung zu einem Gesamtprozesskettenmodell fand eine umfangreiche und automatisiert ausgeführte Datenakquise statt. Die Ergebnisdaten mussten aufbereitet und strukturiert werden, um als Eingangsgröße für das Training von ML-Modellen fungieren zu können. Es ist gelungen die Prozesskette vollständig experimentell und virtuell mit Finite-Elemente-Modellen abzubilden. Damit konnten ML-Modelle trainiert werden, die in der Lage sind - basierend auf Eingangsparameterkombinationen - die Ausgangsgrößen sicher zu prognostizieren. Im Ergebnis lassen sich die trainierten Modelle auch auf neue Fälle anwenden und bieten dadurch die Möglichkeit komplexe Finite-Elemente-Simulationen der Prozesskette durch sehr schnelle ML-Modelle zu substituieren. Dies erleichtert Planung im Anlauf und Parameteranpassungen beim Betrieb von Produktionsprozessen.