Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 47 (2020) 847–854

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

23rd International Conference on Material Forming (ESAFORM 2020)

Estimating Optimum Process Parameters in Textile Draping of Variable Part Geometries - A Reinforcement Learning Approach

Clemens Zimmerling[a,*], Christian Poppe[a], Luise Kärger[a]

[a] *Institute of Vehicle Systems Technology (FAST), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany*

* Corresponding author. Tel.: +49-721-608-45409 ; fax: +49-720-608-945905. *E-mail address:* clemens.zimmerling@kit.edu

Abstract

Fine-tuning of manufacturing processes for optimum part quality requires many resource-intensive trial experiments in practice. To reduce the experimental effort, physics-based process simulations in conjunction with optimisation algorithms can be applied, e.g. finite-element-models and evolutionary algorithms. However, they generally require considerable numerical expertise and long computation times. Efficient optimisation of such expensive-to-evaluate models often employs surrogate-based optimisation (SBO). SBO constructs numerically inexpensive approximations of the original model, which guide the optimiser in the parameter space. This allows concentrating costly simulations on the most promising regions. While SBO significantly reduces the computational load in many cases, current SBO-strategies are inevitably problem-specific and cannot be reused in other, even similar situations. Consequently, subtle problem variations, e.g. minor geometry changes in material forming, require an entirely new optimisation and all previous numerical effort is in vain. Thus, surrogate techniques with generalised applicability are an open field of research. Machine Learning techniques using convolutional neural networks (CNNs) are capable of 'learning' complex system dynamics from data. In this work, CNNs are used to extend the predictive capabilities of SBO towards variable instead of fixed manufacturing settings. Specifically, material draw-in optimisation in textile forming ('draping') for variable geometries is studied. Using reinforcement learning, a CNN is trained to estimate optimum positions of pressure pads during draping of a pre-specified class of box-shaped geometries. Once trained, the CNN interprets a forming result and infers beneficial pad positions. Unlike conventional SBO strategies, it can also give recommendations for variable geometries from the selected geometry class. The paper shows that, in principle, CNNs are able to extract information from a range of different forming tasks and apply it to a new, unknown situation. Since they reuse information gained from previous simulations, they are considered a viable option for future, generalised SBO-strategies.

Keywords: Manufacturing Process Optimisation; Textile Forming; Neural Networks; Machine Learning; Reinforcement Learning

1. Introduction

Most industrial manufacturing processes require careful adjustment of process parameters for optimum operation in terms of e.g. part quality, cycle time and cost. In practice, identification of such optimum process parameters often involves many time- and resource-intensive trial experiments. However, an entirely experimental process optimisation quickly becomes cumbersome, especially for complex manufacturing process and delicate materials such as textiles used in fibre reinforced plastics. Under such conditions, high-fidelity process models, e.g. finite element (FE) simulations, in combination with general-purpose optimisation algorithms may help identify promising process parameters prior to actual experimental trials. This is usually referred to as "virtual process optimisation". Although significant part quality improvements are reported, e.g. [1,2], such models typically require substantial numerical expertise for configuration and considerable computation times. This makes virtual process optimisation time-consuming and in many cases impracticable.

Surrogate-based optimisation (SBO) seeks to reduce computational loads when optimising such expensive-to-

10.1016/j.promfg.2020.04.263

evaluate functions [3]. Surrogate models are numerically efficient, phenomenological approximations of the relation between input parameters and observed output of the high-fidelity-model. After initial model training, optimisation is done on the surrogate model in short time, which yields potential optima (candidate solution). In turn, the high-fidelity model (simulation) validates these candidate solutions resulting in a new observation. This 'new' observation is fed back into the database and model training and optimisation is repeated until convergence or triggering of a termination criterion [4].

In material forming, as considered in this work, most SBO-applications focus on metal forming, e.g. [5-7]. In recent work of the authors [8,9], SBO-strategies are also applied to textile forming. All studies report a significant speed-up of optimisation. However, current SBO-strategies are always problem-specific [10] since they solely generate a unique relation between preselected input parameters and corresponding quality output. In consequence, even subtle changes of the optimisation task, e.g. minor geometry variations, instantly invalidate any obtained surrogate model and all previous numerical effort is in vain.

With the advent of techniques of Machine Learning (ML), generalisation of surrogate models beyond component-specific parameters appears achievable: For example, the authors have shown in work [11] that ML-algorithms can estimate process results for flexible geometry features in textile forming processes. Further generalisation is aspired in [12], where image-based ML-techniques estimate the full strain-field during textile forming given the component geometry.

The present work aims at enhancing SBO towards flexible geometries by image-processing ML-techniques. More specifically, optimisation of material draw-in during fabric forming is considered: By iterative interaction with a simplified model, an ML-algorithm learns to give process recommendations for variable geometries. After the training, the algorithm is requested to give process recommendations for 'unknown' geometries that are not part of the training database. FE forming simulations validate these process recommendations. They are found to improve the process result. Hence, ML-techniques may extend current SBO-capabilities and are deemed a viable tool for time-efficient part and process design.

2. Modelling Approach

2.1. Conceptual View

From an optimisation perspective, an FE forming simulation can be seen as a function $\varphi_{\mathrm{sim}}: P \xrightarrow{G} Q$ that maps process parameters $p \in P$ to a part quality metric $q \in Q$ for a given component geometry $g \in G$. In this work, the variable process parameters p are pressure pad positions for control of material draw-in, while the part quality q is quantified by the in-plane shear deformation γ_{12} and local fabric curvature κ (wrinkling), respectively. In general, evaluating φ_{sim} is so costly that a direct optimisation of φ_{sim} using iterative algorithms, e.g. evolutionary algorithms, takes impractically long.

For time-efficiency, classical SBO pursues constructing a numerically efficient substitute function $\varphi_{\mathrm{ML}}: P \xrightarrow{G} Q$ to approximate φ_{sim}, i.e. $\varphi_{\mathrm{ML}} \approx \varphi_{\mathrm{sim}}$, and perform optimisation on φ_{ML} instead of φ_{sim}. In general, φ_{sim} is a 'black-box'-function for which additional information, e.g. gradients, as required for analytical approximations, may be inaccessible. Thus, only data-driven approximation techniques based on input-output-observations (samples) are eligible.

Although SBO helps reduce the number of simulations for a single optimisation problem, any change of the optimization task invalidates φ_{ML}. Therefore this work investigates the idea of constructing a more generalised function $\Pi_{\mathrm{form}}: G \rightarrow P$ which directly estimates optimum process parameters $p \in P$ given a geometry $g \in G$. Again, Π_{form} is constructed ('learned') using observations. This work uses artificial neural networks to model Π_{form}. Owing to their extraordinary modelling capacity, they show characteristics of universal approximators. That is, given sufficient training data, they will approximate any continuous function, regardless of its complexity and are thus deemed suitable to model Π_{form}. While classical surrogate modelling trains the model function μ_{ML} on a pre-sampled database of input-output examples ('supervised learning'), the idea of this work is to train Π_{form} by interaction with a forming simulation φ_{sim} (Fig. 1). During training, Π_{form} is repeatedly presented different geometries from a geometry database. If Π_{form} gives sound process recommendations, it is rewarded, otherwise penalised.

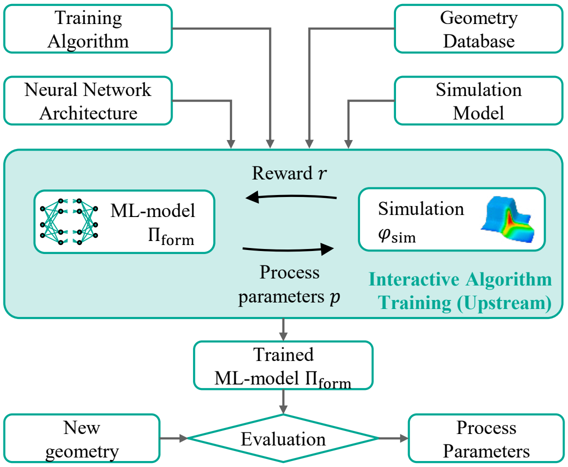
Training Algorithm
Geometry Database
Neural Network Architecture
Simulation Model
Reward r
ML-model Π_{form}
Simulation φ_{sim}
Process parameters p
Interactive Algorithm Training (Upstream)
Trained ML-model Π_{form}
New geometry
Evaluation
Process Parameters

Fig. 1. General workflow to obtain the ML-model function Π_{form} which estimates beneficial process parameters for new geometries.

The reward r is quantified by the reward function and depends on the part quality $q \in Q$ as obtained using φ_{sim}. During training, Π_{form} is stepwise adjusted towards maximisation of the reward r. Its recommendations are initially random, but with the course of training, Π_{form} identifies, which geometry requires which process configuration. After training, the network is used to estimate process parameters for a new geometry, which is not part of the training geometries.

In machine learning, this approach is referred to as "reinforcement learning". Originating in robotics and automation, it has recently been applied for optimisation, e.g. in [13] for optimisation of chemical reactions.

2.2. Data Representation and Algorithm Configuration

In punch-die forming processes, the formed geometry must necessarily be undercut-free to avoid collision-free tool closure. Therefore, without loss of information, a projection of the geometry onto the tool-plane is tractable, which can be encoded in an image. In this work, different greyscale-values quantify the local elevation above tool plane. Analogously, the strain field of the fabric is encoded in an image: Each pixel in the image corresponds to an element of the discretised fabric from the simulation. Plotting the elements' shear strain state onto the undeformed (i.e. flat) fabric yields a 2D representation of the forming result. The data representation is exemplarily visualised in Fig. 2.

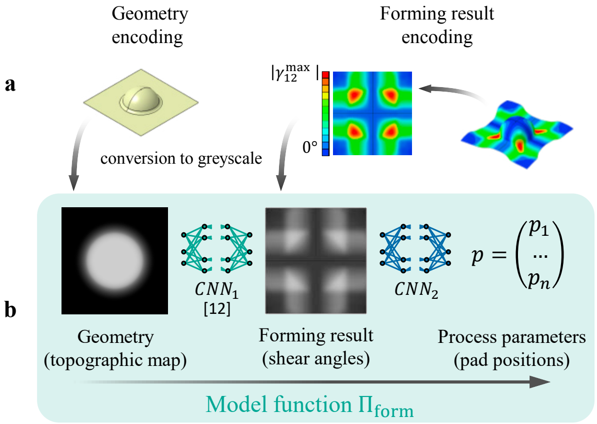
Geometry encoding
Forming result encoding
a
$|\gamma_{12}^{max}|$
0°
conversion to greyscale
b
CNN_1
[12]
CNN_2
$p = \begin{pmatrix} p_1 \\ \cdots \\ p_n \end{pmatrix}$
Geometry (topographic map)
Forming result (shear angles)
Process parameters (pad positions)
Model function Π_{form}

Fig. 2. Image-based data representation according to [12] (a); two-stage processing approach for estimation of process parameters (b).

Apart from a convenient dimensional reduction from 3D to 2D, such an image-based approach grants access to a number of ML-techniques specialised in image-processing, e.g. object recognition, classification, tracking and modification. Also, an image-based geometry description is universal and does not require defining component-specific geometry parameters.

In material forming, a close spatial relationship prevails between local component geometry, material strains and the effect of restraining forces during material draw-in. Convolutional neural networks (CNN), a special type of feedforward networks, are specifically designed to extract patterns in such spatially structured data. As the name hints, CNNs slide ('convolve') kernel matrices ('templates') across target images while continuously computing the degree of coincidence. Matches between template and target image activate neurons in subsequent layers. Upon layerwise repetition, complex relationships can be encoded ('learned'), e.g. estimation of forming results.

The model function Π_{form} in this work comprises two CNNs: First, the component geometry is evaluated by CNN_1 (cf. Fig. 2 b). It takes the geometry (greyscale image) as input and yields a full-field estimation of the resulting shear angles for free forming, i.e. without pads, also encoded in a greyscale-image. This image-estimation is then parsed to CNN_2, which interprets the strain field estimation and infers beneficial process parameters (scalar values).

This work focuses in particular on the implementation of CNN_2 since CNN_1 has already been studied in previous work. Detailed information on implementation, training and application of CNN_1 can be found in [12]. Thus an existing, pre-trained version of CNN_1 is used.

2.3. Forming Simulation Setup and Quality Assessment

This work studies virtual optimisation of material draw-in during forming of plain-weave fabrics for variable geometries. Specifically, forming of box-shaped geometries is considered, whose corners are deliberately pronounced (fillet radii $r = 20$ mm) to provoke severe shear deformations and forming defects. While the height is fixed to 150 mm, the geometry can be varied by means of length l and width w between $50\,\text{mm} \leq l, w \leq 250\,\text{mm}$. The geometry definition along with example variations are visualised in Fig. 3 a.

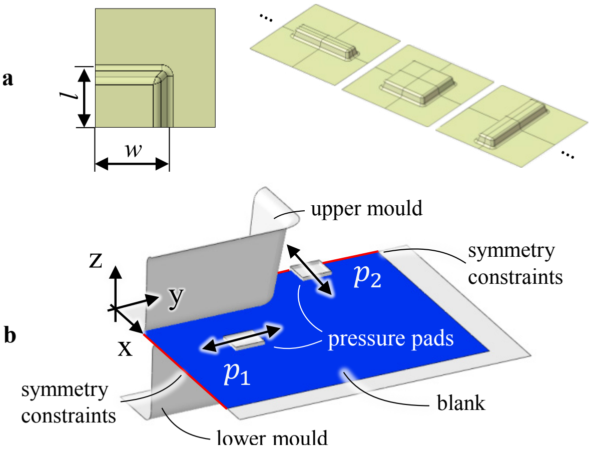

Fig. 3. (a) Visualisation of shape variation (a); (b) draping simulation setup.

Regarding forming simulation, a macroscopic FE-based modelling approach is applied which employs constitutive descriptions of the relevant deformation mechanisms during forming, usually categorised according to intra-ply and interface mechanisms [14]. All constitutive models are implemented within the commercially available FE solver ABAQUS/EXPLICIT by means of several in-house user-subroutines. While modelling on the macroscopic scale, intra-ply mechanisms, namely membrane and bending behaviour, have to be described in a decoupled fashion. This is achieved by means of superimposed membrane and shell elements to represent the single ply of the stacked laminate [14].

Membrane behaviour is implemented for woven fabrics using hypervisco-elastic constitutive equations according to Poppe et al. [15], which correctly account for fibre reorientation under large deformations. To account for material-specific shear locking, a non-linear shear modulus is used and parameterised via experiments. Regarding bending behaviour, a hypoviscoelastic model with a Voigt-Kelvin approach is utilised in a non-orthogonal, curvilinear fibre-parallel frame [16]. Thereby, fibre-orientation is correctly accounted for during bending. Both intra-ply mechanisms are parametrised for the same material (fabric thickness $t =$

0.3 mm). Since the influence of the material parameters is beyond the scope of this investigation, they are kept constant in all runs. Eventually, interface behaviour between fabric and tool is implemented via a built-in contact in ABAQUS/EXPLICIT.

A fully automatised pre- and post-processing framework is applied during model generation. To save computation time, symmetry conditions are applied. The tool is modelled by means of discrete rigid surfaces and is closed in a single stroke of two seconds. For process variation, two quadratic pads (30 × 30 mm) are integrated in the forming simulation. Their position along the punch perimeter can be varied, whilst maintaining a 30-mm-distance to the punch opening line. The pads' downforces are constant ($F_p = 5\,N$ each). Fig. 3 b) shows an example forming simulation setup for visualisation.

Compared to their relatively high tensile stiffness, fabrics show low shear resistance, which makes in-plane shear the predominant deformation mechanism until a material dependent forming limit $\gamma_{12}^{\mathrm{lock}}$ ('locking'). Excessive shear causes fibre compaction and therefore high membrane stresses. This increases the likelihood of unwanted defects, e.g. wrinkling and poor resin infiltration. Therefore, the maximum shear angle $\gamma_{12}^{\max}$ after forming is often used to assess the draping quality and is also evaluated in this work.

Apart from in-plane $\gamma_{12}^{\max}$, this work also evaluates the local fabric curvature κ to quantify wrinkling. In process optimization, usually the maximum of a quality attribute is evaluated. However, for wrinkling assessment evaluating the maximum curvature $\kappa_{\max}$ proved unstable and prone to physically implausible outliers.

To overcome this, more global quality metrics are proposed in literature, e.g. [17], and adapted in this work: The curvature values of each element in the fabric are sorted into a histogram. Then, a Weibull-distribution is fitted and its 99.5 % percentile κ_{p995} is determined, cf. Fig. 4. This global metric delimits the influence of individual elements and any spurious outliers do not distort the part quality result. This approach is found to give reproducible results.

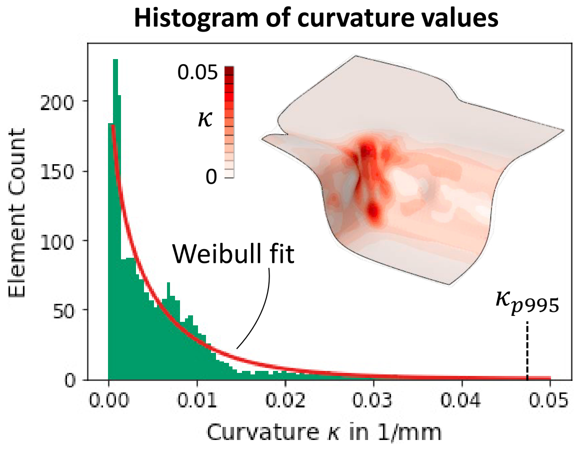

Fig. 4. Simulation result example with significant wrinkling effects, according curvature histogram, Weibull-distribution and 99.5 % percentile.

2.4. Reward Function Definition

Technically, pressure pads introduce local tensile membrane forces, similar to blank holders, and reduce aforementioned compressive stresses in the fabric. To assess and quantify the influence of pads on the forming result, an extensive parametric study is performed, see Fig. 5 for an example visualisation. It shows the influence of different pad positions on the draping quality for three example geometries with variable width w of $w_{\mathrm{geo1}} = 100\,\mathrm{mm}$, $w_{\mathrm{geo2}} = 150\,\mathrm{mm}$ and $w_{\mathrm{geo3}} = 200\,\mathrm{mm}$, respectively. The corner radii of the geometries have been chosen deliberately severe, so that the fabric is at its forming limit, i.e. $\gamma_{12}^{\max} \cong \gamma_{12}^{\mathrm{lock}}$ independent of the pads. Since the pads do not influence $\gamma_{12}^{\max}$, draping quality is solely quantified by κ_{p995} (wrinkling). For comparison, straight horizontal lines mark κ_{p995} for free forming, i.e. without pads. Dashed vertical lines mark the minimum of κ_{p995} for each geometry.

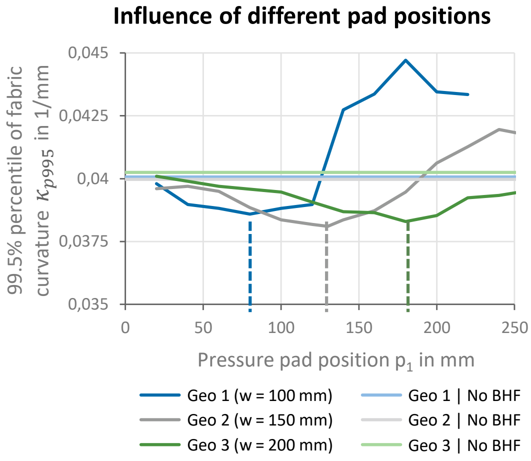

Fig. 5. Influence of the pad positions position on the forming result for three exemplary geometries.

A significant impact of the pads is observed for each geometry. Position-dependent, they reduce or increase wrinkling tendencies: Compared to free forming, a reduction of κ_{p995} by ≈ 6% can be achieved, while an increase by ≈ 12% is observed for adverse positions. Overall, pad-positioning offers sufficient optimisation potential in this work.

Fig. 5 also shows that the optimum pad-positions directly move with width w, and analogously length l, of the box geometry, i.e. position of the corner. For instance, as $w_{\mathrm{geo1}} = 100\,\mathrm{mm}$ changes to $w_{\mathrm{geo3}} = 200\,\mathrm{mm}$ the optimum pad-position analogously shifts from $p = 80\,\mathrm{mm}$ to $p = 180\,\mathrm{mm}$. That is, a constant 'offset' of $\Delta_l = \Delta_w \approx 20\,\mathrm{mm}$ between optimum pad-position and geometry parameters is observed. This observation allows deducing an optimality-criterion for rapid evaluation of process parameters without the need of computation-intensive simulations. Specifically, an explicit formula for the reward $r(p)$ is cast as

$$\begin{aligned} r_x &= -1 + 2.5\, exp\left(\left(p_x - (l - \Delta_l)\right)^2\right) \\ r_y &= -1 + 2.5\, exp\left(\left(p_y - (w - \Delta_w)\right)^2\right) \\ r\ &= r_x + r_y \end{aligned} \tag{1}$$

in which p_x and p_y denote the position of the pads, l and w the geometry parameters and Δ_l and Δ_w denote the observed 'offset' between corner position and optimum pad-position.

This definition of the reward function ensures that a positive reward r is returned, when the algorithm Π_{form} recommends pad-position near the optima (i.e. leading to reduced κ_{p995}) and is otherwise penalised (negative reward).

The analytical formulation beneficially reduces the computational load during methodology development in this work. However, such a rule may only be tractable for simple geometries, wherefore actual forming simulations must replace this explicit reward function in future applications.

2.5. Algorithm Training Scheme

Training of CNN_2 is done using an actor-critic-approach as described in [18]. In essence, two networks are utilized during training: The desired "actor"-network CNN_2, and an auxiliary network required for actor-training, the so-called "critic"-network. Loosely speaking, during algorithm training the actor CNN_2 gives process recommendations while the critic informs the actor how to improve its recommendations. Fig. 6 visualises the training scheme and is outlined in the following.

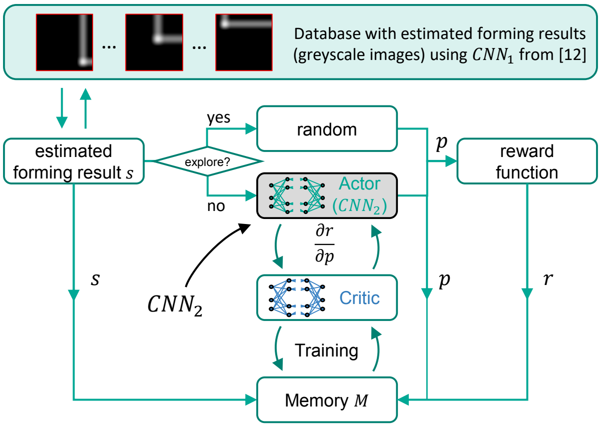
Database with estimated forming results (greyscale images) using CNN_1 from [12]
...
...
estimated forming result s
explore?
yes
no
random
Actor (CNN_2)
$\frac{\partial r}{\partial p}$
Critic
Training
Memory M
reward function
p
p
r
s
CNN_2

Fig. 6. Visualisation of the algorithm training scheme.

One training iteration consists of two steps, an observation step and a network training step. To obtain a new observation, a greyscale image of the shear angle distribution s (process 'situation') is drawn from the database resulting from CNN_1 and parsed to CNN_2. CNN_2 analyses the image s and returns a parameter recommendation $p = (p_x \quad p_y)^T$. The quality of this parameter recommendation p is then evaluated using the reward function r defined in equation (1). Finally, the estimated forming result s , the recommended process parameters p and the obtained reward r are collated in an 'observation'-tuple $\{s, p, r\}$ and stored in a process database M ('memory'). This is repeated for the remaining images in the database, i.e. in each iteration each image is evaluated once.

During the subsequent algorithm training, at first the critic network is updated using the observations stored in the memory M. This is done by adjusting weights and biases of the neurons' interconnections (i.e. model parameters $\underline{\theta_c}$) so that predictions match better the observations in the memory M. That is, $\underline{\theta_c}$ is tuned to minimise the 'loss'-function $L(\hat{r}, r)$ between predicted reward $\hat{r}(\underline{\theta_c}, \{s, p\})$ and the 'true' reward $r(\{s, p\})$ that was originally received for the process recommendation p for in process situation s. Formally, this is cast as

$$\underline{\theta_c}^* = arg \min_{\underline{\theta_c} \in \mathbb{R}} L(\hat{r}, r) \tag{2}$$

While different loss functions $L(\hat{r}, r)$ are applicable in general, this work uses the L$_2$-loss defined by

$$L = L_2 = \frac{1}{n} \sum_{i=1}^{n} L_{2_i} = \frac{1}{n} \sum_{i=1}^{n} (\hat{r}_i - r_i)^2 \tag{3}$$

with n being the number of observations in the memory. Subsequently, the model parameters of CNN_2 ('actor') $\underline{\theta_a}$ are updated in direction of increasing reward r using the expectation E of

$$\frac{\partial r}{\partial \theta_{a\,j}} = E\left[\left(\frac{\partial r_i}{\partial p_i} \frac{\partial p_i}{\partial \theta_{a\,j}}\right)_{i=1\ldots n}\right] \tag{4}$$

, wherein $\frac{\partial r_i}{\partial p_i}$ is the gradient of reward r_i with respect to the process recommendation p_i for the i-th of n observations in the memory M. The variable $\theta_{a\,j}$ denotes the j-th model parameter in $\underline{\theta_a}$.

The networks' weights are initialised randomly. Therefore, evaluations of the networks will be of low significance at training start. Consequently, any gradient information $\frac{\partial r_i}{\partial p_i}$ may be misleading for the actor-network within the first iterations. More important, adjusting the actor network at the very start, often leads to poor results, since the network directly approaches the minimum in closest proximity without globally exploring other options. This is known as exploration-vs-exploitation dilemma in literature. Therefore, this work delays updating the actor network CNN_2 (equation (4)) until the critic-network's predictions have reached a minimum accuracy measured by the L$_2$-loss. Empirically, a threshold of $L_2 < L_{2\,Thr} = 1.0$ proved sufficient. Until then, in each iteration process parameters p are chosen randomly to facilitate exploration and build a rather global process memory M before exploiting promising strategies.

Once, L_2 is sufficiently small and actor training starts, random actions are selected only with a certain, constantly decreasing probability $\varepsilon \in [0,1]$. In this work, an exponential decay is chosen, i.e. $\varepsilon = {\varepsilon_0}^k$, with $\varepsilon_0 = 0.95$ and k the current training iteration number. This approach is known as ε-greedy exploration [18] and seeks to balance exploration, i.e. probing random process parameters, and exploitation, i.e. refining current network strategy.

3. Results

3.1. Algorithm training progress

In total 20 box geometries are generated, converted to greyscale-images and evaluated by the existing CNN_1 presented in [12]. This results in 20 greyscale-images $s_{1\ldots20}$ of the estimated shear angle distributions (cf. Fig. 2). As length l and width w are known for each geometry, in each case a reward function according to equation (1) is established, resulting in 20 reward functions $r_{1\ldots20}$.

The images $s_{1\ldots20}$ and the according reward functions $r_{1\ldots20}$ are stored in a database $\boldsymbol{S} = \{\{s_1, r_1\}, \{s_2, r_2\}, \ldots\}$. It is common practice to split $\boldsymbol{S}$ into two non-intersecting subsets, i.e. $\boldsymbol{S}_T \cup \boldsymbol{S}_V = \boldsymbol{S}$ and $\boldsymbol{S}_T \cap \boldsymbol{S}_V = \{\}$. Training is performed on the training set $\boldsymbol{S}_T$ only, while the algorithm's performance on the 'unknown' validation subset $\boldsymbol{S}_V$ allows for an assessment of the generalisation capabilities. In this work 5 geometries are held out for validation, while 15 geometries are used for training.

Fig. 7 a) shows the evolution of the loss L_2 of the critic network during algorithm training on a logarithmic scale. Within the first $k \leq 10$ iterations, a precipitous drop of the L_2-loss is observed, i.e. the critic networks rapidly improves its accuracy on estimating the reward for given process parameters. Then, a plateau is observed for another approximately 20 iterations ($10 \leq k \leq 30$). This occurs, because a growing number of observations with new information on the system dynamics becomes available in the memory. From approximately 30th iteration on, the loss decreases in a monotonous manner apart from small spikes. This hints that new observations are now recurring rather than new events and do not convey as much new information to the network as before. Loosely speaking, the network has sufficient information on the underlying process dynamics.

In iteration $k = k_0 = 74$, the L_2-loss becomes smaller than the pre-defined threshold $L_{2Thr} = 1.0$ and training of the actor network CNN_2 starts according to equation (4). At the same time, the exploration probability ε starts decaying, cf. Fig. 7 b). That is, the algorithm follows more and more its own policy rather than performing exploration by acting randomly. After 150 iterations the exploration probability ε is less than 5 % and CNN_2 acts virtually deterministic.

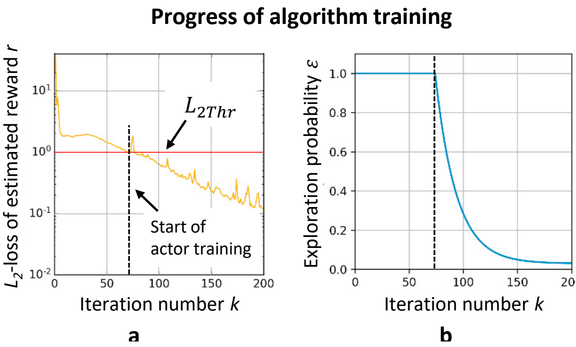

Fig. 7. Evolution of loss (a) and exploration probability (b).

The quality of the process recommendations p of CNN_2 is quantified by the obtained reward r. Since in each iteration multiple images are evaluated, the average reward per iteration r_{avg} is used to assess the performance of CNN_2. Note, that only data from $\boldsymbol{S}_T$ is used for adjusting network parameters. When evaluating data from the validation set $\boldsymbol{S}_V$, the network parameters $\underline{\theta}_a$ and $\underline{\theta}_c$ are kept constant and not adjusted. Thus, data from the validation set $\boldsymbol{S}_V$ is always 'new' to the algorithm and the according reward $r_{\text{avg V}}$ allows an assessment of the performance on unknown geometries.

Fig. 8 shows the evolution of r_{avg} during training. Two graphs are plotted, the average reward on the training geometries $r_{\text{avg T}}$ and the average reward on the held out validation geometries $r_{\text{avg V}}$. During the initial exploration phase (grey shade), random process parameter combinations are drawn only. Consequently, the average reward $r_{\text{avg T}}$ is entirely random and serves only to build the memory M. The average reward on validation geometries $r_{\text{avg V}}$ is set to zero. Since no training on the actor network CNN_2 is done during exploration, any evaluation of CNN_2 would be meaningless. Once exploration is terminated (iteration $k = 74$), training of CNN_2 starts and its model parameters $\underline{\theta}_a$ are stepwise adjusted in direction of increasing reward.

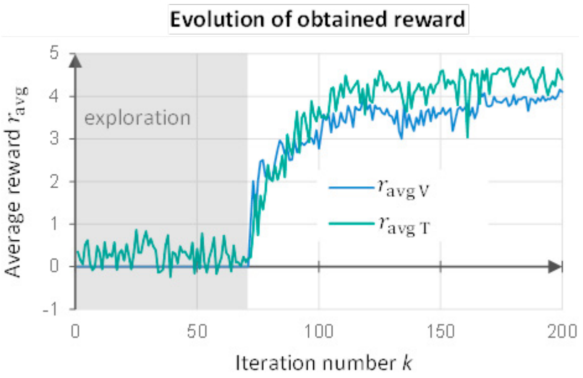

Fig. 8. Evolution of average reward r_{avg}.

Both graphs, $r_{\text{avg T}}$ and $r_{\text{avg V}}$, show a similar evolution. Initially, a steep ascent of $r_{\text{avg T}}$ and $r_{\text{avg V}}$ within the first additional iterations ($74 \leq k \leq 90$) is apparent. Further training ($90 \leq k \leq 110$) still increases both rewards, though at a lower rate. In the end ($k \geq 110$), both rewards approach an approximately constant reward-level in an asymptotic manner.

The reward-level on the training geometries $r_{\text{avg T}}$ is almost constantly higher than the validation-reward $r_{\text{avg V}}$. This is a common observation in surrogate-modeling and ML: While the surrogate/ML-model constantly improves on supplied training data, transferring obtained relationships to additional (validation) data usually introduces some error. In extreme, this can lead to a nearly perfect approximation of training data, while performing poor on additional (validation) data. This phenomenon is known as 'overfitting' in statistical modelling.

Overall, the increasing validation curve $r_{\text{avg V}}$ clearly shows, that CNN_2 is able to extract process 'knowledge' from a set of geometries and apply it to new 'unknown' situations.

3.2. Performance on unknown box-geometries

Validation of the ML-assisted process is done for a geometry g_{val} ($l = 120$ mm , $w = 180$ mm) from the validation set $\boldsymbol{S}_V$. First, the ML-assisted recommendation is obtained by Π_{form}. After conversion of geometry g_{val} to a greyscale image, evaluation of Π_{form} comprises two steps (cf. Fig. 2):

- Estimate forming result s_{val} from geometry-image by CNN_1 (pre-trained available as described in [12])
- Estimate optimum process parameters p using CNN_2

To validate the quality of the ML-predictions, FE-forming simulations are used. Please refer to section 2.3 for model description. Two different simulation setups are considered, free forming of fabric, i.e. no pads, and forming with pads positioned according to ML-recommendation (Fig. 9, left). The forming results are evaluated with regard to fabric curvature κ, as a measure of wrinkling. Fig. 9 (right) shows the resulting curvature-metrics of both simulations.

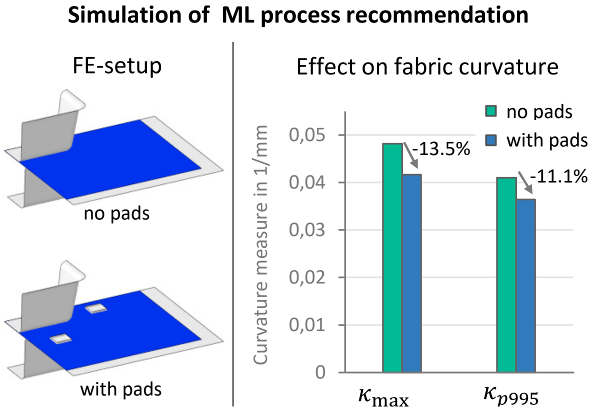

Fig. 9. Example visualisation of simulation setup for validation and column-plot of the improvement.

Two different curvature measures are plotted, the 99.5% percentile of the Weibull-fit κ_{p995} and the maximum curvature $\kappa_{\max}$ for reference. Compared to free forming, the ML-assisted process recommendation by Π_{form} successfully reduces the fabric curvature by 13.5 % ($\kappa_{\max}$) or 11.1% ($\kappa_{995\text{p}}$), respectively. That is, the overall function Π_{form} has made meaningful inference on process parameters given a 'new' geometry from the geometry class. Therefore, process knowledge has successfully been transferred from previous forming tasks to a new forming task.

3.3. Process recommendations on double-dome geometry

Another important aspect is, whether Π_{form} can make inference on a geometry, that lies outside the training geometry class of 'boxes'. To this, the double-dome geometry g_{DD} is evaluated as shown in Fig. 10.

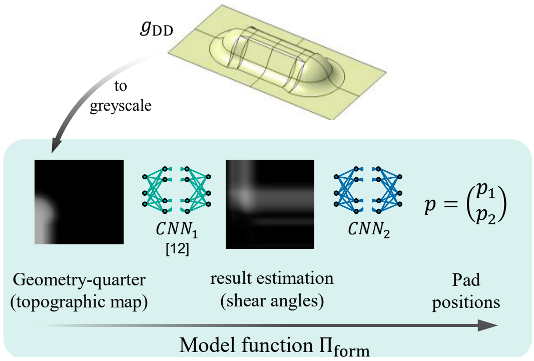
g_{DD}
to greyscale
CNN_1
[12]
CNN_2
$p = \binom{p_1}{p_2}$
Geometry-quarter (topographic map)
result estimation (shear angles)
Pad positions
Model function Π_{form}

Fig. 10: Visualisation of the double-dome geometry as a test case and according evaluation scheme by Π_{form} (CNN_1 and CNN_2).

Forming of the double-dome does only evoke moderate shear deformations below the fabric's deformation limit, i.e. $\gamma_{12}^{\max} < \gamma_{12}^{\text{lock}}$. Therefore wrinkling did not occur and local curvature κ is not significantly influenced. Thus, the shear angles γ_{12} are evaluated instead. Fig. 11 visualises the effect of different pad-positions. The straight horizontal line denotes $\gamma_{12}^{\max}$ in free forming. Again, introducing pads gives a position-dependent reduction of $\gamma_{12}^{\max}$ by up to 9.1 %, except for the outermost position $p = 180$ mm . In this case, pads even increase $\gamma_{12}^{\max}$, i.e. deteriorating the forming result.

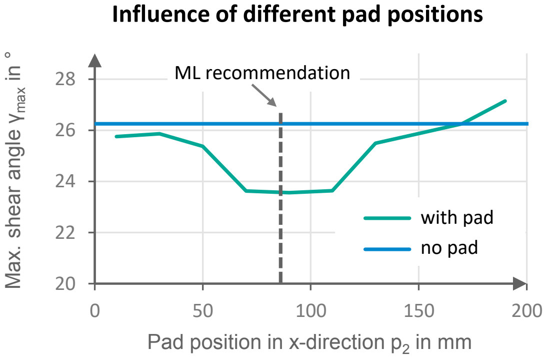

Fig. 11: Visualisation of the effect of different pad positions on the maximum shear angle during forming.

The dashed vertical line marks the recommendation $p \approx 90$ mm by Π_{form}. Comparing the recommendation p with the parametric study, reveals that p lies in a promising range. Thus, Π_{form} has transferred process knowledge from previous process-observations (box-geometries) and made meaningful inference on a new geometry from a different geometry class.

4. Conclusion and Outlook

A machine-learning-assisted approach is presented, which rapidly estimates beneficial process parameters (pad positions) in textile forming for variable component geometries using an image-based approach. To this, a function Π_{form} is proposed, which comprises two main steps: First, an image-processing convolutional neural network (CNN_1) analyses a greyscale-image of the geometry of interest and yields an image-estimation of the shear angle distribution after forming. Then, a second network CNN_2 interprets this estimation and infers beneficial pad positions.

The function Π_{form} is trained using Reinforcement Learning on a simplified phenomenological model deduced from textile-forming simulations. To facilitate comprehensibility of the algorithm behaviour during development, convex, box-shaped geometries are considered. After training, Π_{form} is validated on two test scenarios. First, it must make inference on an 'unknown' box-geometry that was held out during algorithm training. Second, a common benchmark geometry in textile draping, the "double-dome", is used to test the generalisation capability of Π_{form}. In both cases, an FE forming simulation validates the process recommendations of Π_{form}.

Overall, Π_{form} is found to give meaningful recommendations, which improve the forming result. Thus, it may be concluded that Π_{form} is generally able to extract process knowledge from process examples of different geometries and apply it to 'new' components. This significantly extends current surrogate capabilities, which are always component-specific and limited to a certain, pre-specified parameter selection. Also, results of this work hint that surrogate training could be decoupled from the optimisation task to a certain extent: Given sufficient computational resources, Π_{form} could be pre-trained on a sufficiently large database and potential users may apply Π_{form} without the need of modelling and training a surrogate themselves. Although training such a model would require considerable computational effort, the benefit of immediate process recommendation and ease of application may outweigh the expenses in the long term.

Further development is still envisaged. First, the simplified phenomenological reward-function needs to be replaced by actual forming simulations. Even if similar functions may be tractable for other simple geometries, e.g. ellipsoids or cylinders, this appears unlikely for arbitrary geometries.

Additionally, the scope of geometries requires extension, since Π_{form} only makes valid process recommendations for the range of previously seen geometry classes (e.g. convex shapes). Therefore, future training databases require the deliberate inclusion of additional geometry shapes for full generalisation, e.g. convex-concave shapes.

Finally, a remaining research question pertains to component-specific optimisation: Although results show that Π_{form} gives reasonable 'near-optimum' recommendations, they do not necessarily have to be strict optima. To achieve an optimum solution, a component-specific refinement of Π_{form} is required. Future research may investigate, if additional training of Π_{form} on a single geometry further improves initial recommendations and converges to an actual optimum.

Overall, the proposed ML-based process optimisation approach appears as a promising enhancement of current SBO-strategies for fast and efficient process design at early stages of product development.

Acknowledgment

The authors would like to thank the German Federal State Ministry of Science, Research and the Arts of Baden-Württemberg (MWK) for funding the project "Forschungsbrücke Karlsruhe-Stuttgart", which the presented work is carried out for. The work is also part of the Young Investigator Group (YIG) "Tailored Composite Materials for Lightweight Vehicles", generously funded by the Vector Stiftung.

References

[1] Kärger L, Galkin S, Zimmerling C, Dörr D, Linden L, Oeckerath A, Wolf K. Forming optimisation embedded in a CAE chain to assess and enhance the structural performance of composite components. Composite Structures 2018;192:143-152.

[2] Chen S, Harper LT, Endruweit A, Warrior NA. Formability optimisation of fabric preforms by controlling material draw-in through in-plane constraints, Composite Part A 2015;76:10-19.

[3] Forrester I, Keane AJ. Recent advances in surrogate-based optimization, Progress in Aerospace Sciences 2009;45-1:50–79.

[4] Han ZH, Zhang KS. Surrogate-Based Optimization. In: Roeva O, editor. Real-World Applications of Genetic Algorithms. London/UK: IntechOpen Ltd.; 2012, p. 343-362.

[5] Jakumeit J, Herdy M, Nitsche M. Parameter optimization of the sheet metal forming process using an iterative parallel Kriging algorithm. Structural and Multidisciplinary Optimization 2005;29:498-507.

[6] Bonte MHA, van den Boogaard AH, Huétink J. A Metamodel Based Optimisation Algorithm for Metal Forming Processes. In: Banabic D, editor. Advanced Methods in Material Forming. Springer Berlin/Heidelberg; 2007, p. 55-72.

[7] Wang H, Chen L, Li, E. Time dependent sheet metal forming optimization by using Gaussian process assisted firefly algorithm. International Journal of Material Forming, 2018;11-2:279-295.

[8] Pfrommer J, Zimmerling C, Liu J, Kärger L, Henning F, Beyerer J. Optimisation of manufacturing process parameters using deep neural networks as surrogate models, Procedia CIRP 2018;72:426–431.

[9] Zimmerling C, Pfrommer J, Liu J, Beyerer J, Henning F, Kärger L. Application and evaluation of meta-model assisted optimisation strategies for gripper-assisted fabric draping in composite manufacturing, Proceedings of the 18th ECCM 2018.

[10] El Kadi H. Modeling the mechanical behavior of fiber-reinforced polymeric composite materials using artificial neural networks - a review. Composite Structures 2006;73-1:1-23.

[11] Zimmerling, Dörr D, Henning F, Kärger L. A Machine learning assisted approach for textile formability assessment and design improvement of composite components. Composites Part A 2019;124:105459.

[12] Zimmerling C, Trippe D, Fengler B, Kärger L. An approach for rapid prediction of textile draping results for variable composite component geometries using deep neural networks. AIP Conference Proceedings 2019;2113:020007.

[13] Zhou Z, Li X, Zare RN. Optimizing Chemical Reactions with Deep Reinforcement Learning. ACS Central Science 2017;3-12:1337-1344.

[14] Dörr D, Schirmaier FJ, Henning F, Kärger L. A viscoelastic approach for modeling bending behavior in finite element forming simulation of continuously fiber reinforced composites. Composites Part A 2017;94:113–123.

[15] Poppe C, Dörr D, Henning F, Kärger L. Experimental and numerical investigation of the shear behaviour of infiltrated woven fabrics, Composite Part A 2018;14:327-337.

[16] Poppe C, Rosenkranz T, Dörr D, Kärger L. Comparative experimental and numerical analysis of bending behaviour of dry and low viscous infiltrated woven fabrics. Composite Part A 2019;124:105466.

[17] Chen S, Endruweit A, Harper LT, Warrior NA. Inter-ply stitching optimisation of highly drapeable multi-ply preforms, Composites Part A 2015;71:144-156.

[18] Sutton RS, Barto AG. Reinforcement Learning: An Introduction. 2nd ed. Cambridge/MA/USA: MIT Press; 2018.