



24th International Conference on Material Forming (ESAFORM 2021)

Deep Neural Networks as Surrogate Models for Time-efficient Manufacturing Process Optimisation

Clemens Zimmerling^{a,*}, Patrick Schindler^a, Julian Seuffert^a and Luise Kärger^a

^a Karlsruhe Institute of Technology (KIT), Institute of Vehicle System Technology, Karlsruhe, Germany

* Corresponding author: Clemens Zimmerling. E-mail address: clemens.zimmerling@kit.edu

Abstract. Manufacturing process optimisation usually amounts to searching optima in high-dimensional parameter spaces. In industrial practice, this search is most often directed by human-subjective expert judgment and trial-and-error experiments. In contrast, high-fidelity simulation models in combination with general-purpose optimisation algorithms, e.g. finite element models and evolutionary algorithms, enable a methodological, virtual process exploration and optimisation. However, reliable process models generally entail significant computation times, which often renders classical, iterative optimisation impracticable. Thus, efficiency is a key factor in optimisation. One option to increase efficiency is surrogate-based optimisation (SBO): SBO seeks to reduce the overall computational load by constructing a numerically inexpensive, data-driven approximation („surrogate“) of the expensive simulation. Traditionally, classical regression techniques are applied for surrogate construction. However, they typically predict a predefined, scalar performance metric only, which limits the amount of usable information gained from simulations. The advent of machine learning (ML) techniques introduces additional options for surrogates: in this work, a deep neural network (DNN) is trained to predict the full strain field instead of a single scalar during textile forming („draping“). Results reveal an improved predictive accuracy as more process-relevant information from the supplied simulations can be extracted. Application of the DNN in an SBO-framework for blank holder optimisation shows improved convergence compared to classical evolutionary algorithms. Thus, DNNs are a promising option for future surrogates in SBO.

Keywords: Neural Networks; Deep Learning; Machine Learning; Optimisation; Surrogate; Draping; Textile forming; Manufacturing; Production

1 Introduction

Optimum operation of production lines in terms of part quality, cycle time or cost generally requires diligent parameterisation of manufacturing processes. In practice, identification of such optimum parameters during production ramp-up usually involves many time- and resource-intensive experimental trials and experiential expert judgment. Thus, an entirely experimental optimisation rapidly becomes cumbersome. This holds all the more for complex processes and delicate materials, e.g. such as technical textiles used in fibre-reinforced components.

High-fidelity process models, e.g. finite element (FE) simulations, offer means for virtual process analysis. In combination with general-purpose optimisation algorithms, e.g. evolutionary algorithms [1], they provide options to systematically and reliably optimise manufacturing. Often termed “virtual process optimisation”, such approaches may help determine promising parameters prior to actual experimental trials. Despite significant process improvements being reported, e.g. [2], reliable models typically require considerable computation times of e.g. hours and days. Iterative optimisation then becomes time-consuming and, in many cases, impracticable. Consequently, time-efficiency is a key factor during optimisation.

One option to reduce the overall computational load is surrogate-based optimisation (SBO). SBO employs numerically efficient approximations of the high-fidelity process model, the “surrogate”, which guide the optimiser in the parameter space [3]. In material forming, as considered in this work, most SBO-applications focus on metal forming, e.g. [4]-[6]. Recent work of the authors additionally addresses textile forming [7],[8]. All studies report a significant speed-up of optimisation.

Although simulations typically provide detailed process information, most surrogate techniques of prior work consider scalar or low-dimensional product attributes only. This „compression“ or „truncation“ of data limits the amount of usable information gained

from simulations. Advanced Machine Learning (ML) techniques, e.g. deep learning (DL), enable novel surrogate types, which do not merely reflect an abstract performance scalar but consider full-field simulation data [7]-[11]. According studies report high surrogate accuracy and thus tacitly expect improved optimisation performance, yet fall short on quantitative comparison: To the authors' knowledge, to date no benchmark against classical, scalar surrogate techniques is available and only [7] gives a brief glimpse on SBO with DL-models. This work aims to substantiate and enhance current findings with numerical evidence regarding optimisation performance.

The scope of this study is twofold: first, predictive accuracy of DL-surrogates is compared to classical surrogates (*full-field* vs. *scalar* surrogates) for different number of training samples. Full-field prediction is achieved with a deep neural network (DNN). It predicts the entire strain field with $n_{el} = 22080$ elements during textile forming. Second, four different SBO-strategies give insight into optimisation performance and the observed convergence is benchmarked against a state-of-the-art evolutionary algorithm (EA).

2 Workflow and Use-case

Due to their superior mechanical properties, continuous-fibre reinforced plastics (CoFRP) have drawn increasing attention in weight-sensitive industries. However, they typically invoke higher cost, not least due to higher engineering effort for defect-free production. Manufacturing of CoFRP-components typically comprises multiple steps, often including a forming step of an initially flat textile ("draping"), e.g. woven fabrics as considered in this work.

Woven fabrics show a comparably low shear stiffness compared to tensile stiffness in warp or weft direction. This makes in-plane shear the predominant deformation mechanism, which is quantified by the in-plane shear angle γ_{12} (Fig. 1 a). For brevity, this work uses $\gamma = \gamma_{12}$. Alike any other material, woven fabrics show a material-dependent forming limit, which is usually quantified by the locking-angle γ_{lock} . Excessive shear beyond γ_{lock} increases the likelihood of unwanted defects, such as wrinkling or poor permeability during subsequent resin infiltration ("dry spots"). Therefore, γ is often minimised in process optimisation.

2.1 Simulation Model for Optimisation

This work studies forming of the double-dome geometry, a common benchmark geometry in textile forming. Regarding forming simulation, a macroscopic FE-based modelling approach is applied. It employs constitutive descriptions of the relevant deformation mechanisms by subroutines within the FE-solver ABAQUS/EXPLICIT. See [12]-[14] for modelling and parameterisation details. Superposition of membrane and shell elements ensures decoupling of membrane and bending behaviour and a non-linear shear modulus captures material-specific shear locking. Discrete rigid surfaces model the tool surfaces and the tool closes within $t_{tool} = 2$ s in a single stroke. Figure 1 b) shows an example simulation setup along with an according forming result (shear angles).

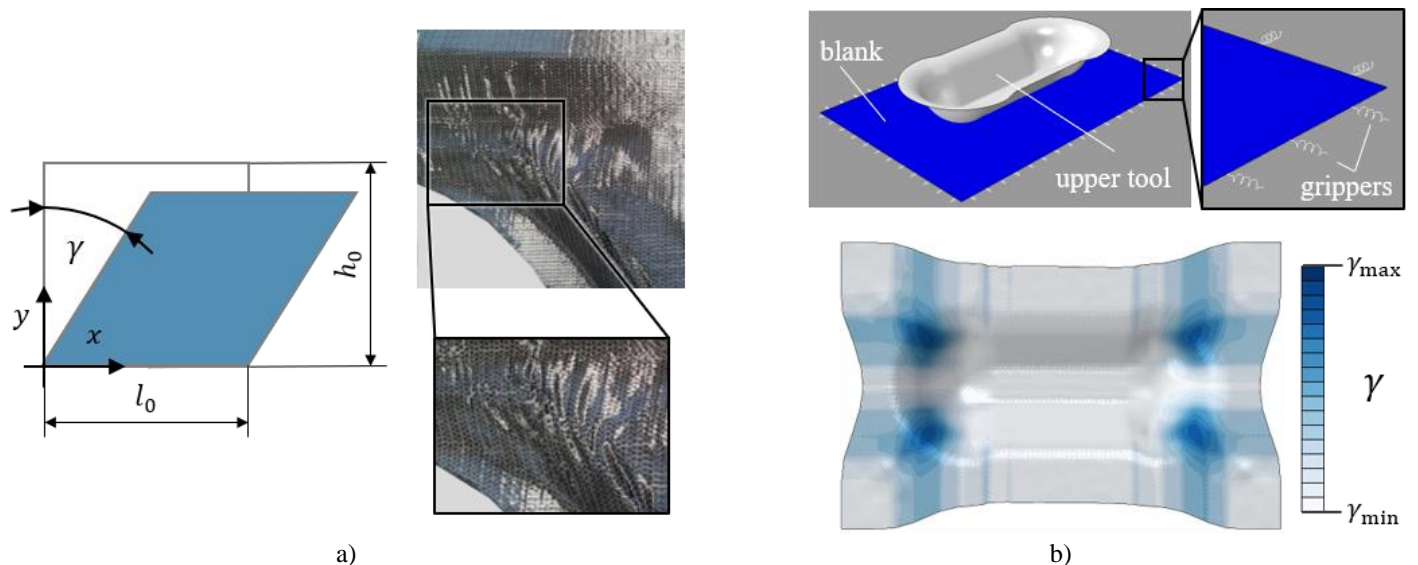


Fig. 1. Visualisation of shear angle and example of textile wrinkling [2] a), forming simulation setup and an example forming result (top view on shear angle distribution) b).

Process manipulation is possible through 60 grippers modelled by springs ($0.01 \frac{N}{mm} \leq c_i \leq 1 \frac{N}{mm}$). They are uniformly distributed around a rectangular blank of thickness $s = 0.3$ mm, cf. Fig. 1 b). Similar to conventional blank holders, they introduce tensile membrane forces into the textile which restrain material draw-in during tool closure. An in-house-developed pre- and postprocessing framework allows for fully-automatic model generation and result analysis during optimisation. Despite symmetry of geometry and material, no symmetry conditions are applied, since – in principle – springs may become asymmetric during optimisation.

2.2 Surrogate Approach

In general, a process simulation may be seen as a function $\varphi_{\text{sim}}: P \mapsto A$ which maps process parameters $\mathbf{p} \in P$ to a part quality attribute $\mathbf{a} \in A$. In many cases, A quantifies part quality by extent of defects, e.g. formation of wrinkles or cracks. This work considers γ as a proxy to wrinkling as outlined above. Please note, that γ is an elemental quantity and thus the overall product quality $\mathbf{a} = \boldsymbol{\gamma} = (\gamma^1, \dots, \gamma^{n_{\text{el}}})^T$ is a vector in $\mathbb{R}^{n_{\text{el}}}$ with n_{el} being the count of fabric element. Analogously, the circumferential spring stiffnesses represent the variable process parameter search space, i.e. $\mathbf{p} = \mathbf{c} = (c_1, \dots, c_{60})^T$.

Ultimately, an objective function $f: A \mapsto Q$ must map the part quality attributes to a scalar performance metric $q \in Q \subset \mathbb{R}$. Virtual process optimisation then amounts to finding

$$\mathbf{p}^* = \arg \min(q) = \arg \min(f(\mathbf{a})) = \arg \min(f(\varphi_{\text{sim}}(\mathbf{p}))) \quad . \quad (1)$$

However, evaluating φ_{sim} is often so costly that a direct optimisation using iterative algorithms takes prohibitively long. For increased efficiency, surrogate-based optimisation (SBO) proposes devising an easy-to-evaluate approximation $\mu_{\text{surr}}: P \mapsto Q$ with

$$\mu_{\text{surr}} \approx f(\varphi_{\text{sim}}(\mathbf{p})) \quad \forall \mathbf{p} \in P_{\text{adm}} \subset P \quad . \quad (2)$$

Therein, P_{adm} denotes the search space for optimisation. Consequently, optimisation takes place on μ_{surr} instead of $f(\varphi_{\text{sim}})$.

In general, φ_{sim} is a „black-box“-function, i.e. it can be evaluated but it otherwise unknown. In such cases data-driven approximations based on n input-output-observations $D^n = \{(\mathbf{p}_1, f(\mathbf{a}_1)), \dots, (\mathbf{p}_n, f(\mathbf{a}_n))\}$ are suitable. For this, a plethora of different techniques exists [15], e.g. polynomial regression, Support-Vector techniques or Neural Networks. Differences in model function aside, they all follow the notion of tuning model parameters $\boldsymbol{\theta} \in \boldsymbol{\theta}$ towards minimisation of an error metric ε_{err} , e.g. mean squared error (MSE)

$$\varepsilon_{\text{err}}^{\text{MSE}}(D^n, \boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n (\mathbf{a}_i - \hat{\mathbf{a}}_i)^2 = \frac{1}{n} \sum_{i=1}^n (\mathbf{a}_i - \mu_{\text{surr}}(\mathbf{p}_i, \boldsymbol{\theta}))^2 \quad (3)$$

, which is also used in this work.

This work concentrates on deep neural networks (DNN) since they pose several advantages: first, they are universal approximators [16]. That is, given sufficient data they can reproduce any continuous function irrespective of its complexity and thus promise general suitability as surrogates. Additionally, over the last decades a large community of researchers developed specialised sub-types of DNNs for specific tasks (image-recognition, time-series-analysis,...) and embodied them in novel ML-algorithms, e.g. advanced Reinforcement Learning techniques. For the authors, these developments can be means to more capable engineering surrogates beyond ‘simple’ input-output-relations. Building an understanding for their fundamental behaviour in engineering tasks – as pursued in this work – certainly is a prerequisite to this. In general, DNNs consist of complex parallel and series connections of so-called „neurons“, whose individual parameters constitute the model parameters $\boldsymbol{\theta}$. Within k_{trn} “training episodes” $\boldsymbol{\theta}$ is gradually adjusted to minimise $\varepsilon_{\text{err}}^{\text{MSE}}(D^n, \boldsymbol{\theta})$. See [17] for details on DNNs and their training.

Prior work focuses on emulating the scalar objective function $f(\varphi_{\text{sim}})$ only. However, in intricate cases, results are not as convincing [7]. Therefore, this work suggests bringing the surrogate closer to simulation results. More precisely, instead of training the surrogate to mimic the scalar objective function, i.e.

$$\mu_{\text{surr}}(\mathbf{p}) \approx f(\mathbf{a}) = f(\varphi_{\text{sim}}(\mathbf{p})) \quad , \quad (4)$$

this work trains the surrogate to predict the complete strain field

$$\mu_{\text{surr}}(\mathbf{p}) = \hat{\mathbf{a}} \approx \mathbf{a} = \varphi_{\text{sim}}(\mathbf{p}) \quad . \quad (5)$$

Thereby, additional positional information is introduced to the surrogate: for example, the influence of each spring mainly affects its immediate vicinity on the textile. Such local influence cannot be resolved in a global scalar metric, which consequently leads to a loss of information in the database. Training the surrogate on field-data retains this information and may thus increase accuracy.

2.3 Surrogate-based Optimisation

The obtained surrogate model μ_{surr} from Section 2.2 can be used for SBO. However, being an entirely statistical model, μ_{surr} inevitably introduces deviations compared to the original function φ_{sim} . Therefore, a single optimisation on μ_{surr} may not yield the true optimum of $f(\varphi_{\text{sim}}(\mathbf{p}))$. SBO aims at iterative removal of these deviations by sequential updates with new observations (simulations). A common approach is to directly evaluate the found optimum $\mathbf{p}_{\text{surr}}^*$ of μ_{surr} , i.e. evaluating $\mathbf{a}_{\text{sim}}^* = \varphi_{\text{sim}}(\mathbf{p}_{\text{surr}}^*)$. The new observation $(\mathbf{p}_{\text{surr}}^*, \mathbf{a}_{\text{sim}}^*)$ is then fed back into the database D^n and training continues for k_{trn} episodes (gradient-descents). Thereby the surrogate refines in vicinity of potential optima and explores its most promising parameter regions until triggering of a termination criterion. Figure 2 illustrates the approach schematically.

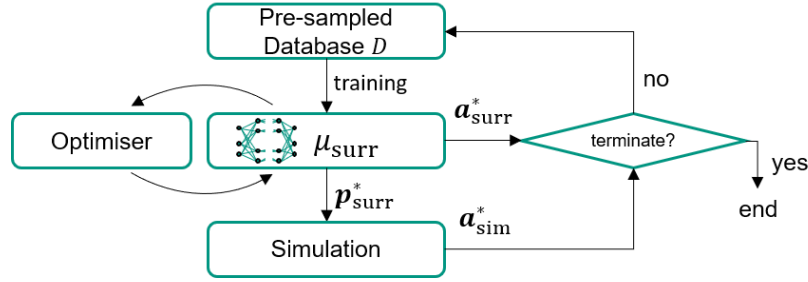


Fig. 2. Scheme of surrogate-based optimisation as applied in this work.

3 Results and Discussion

3.1 Surrogate Construction

This work investigates the effect of different surrogate strategies at the example of artificial neural networks (ANNs). More specifically, two effects are studied: first, the effect of depth of ANNs (i.e. number of layers) and, second, the effect of full-field-data instead of scalar-data during model training. To this, three types of ANNs are considered as shown in Fig. 3: the SS-type (shallow network, scalar information), the DS-type (deep network, scalar information) and DF-type (deep network, field information).

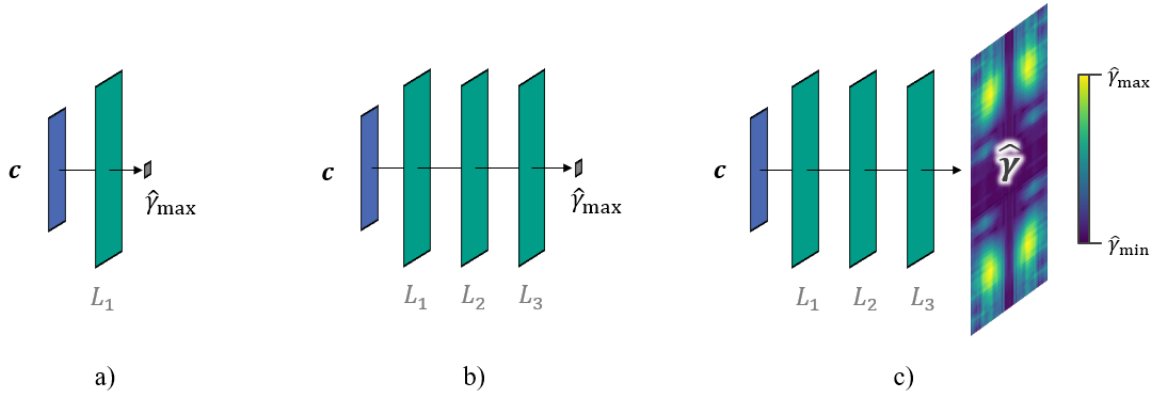


Fig. 3. Visualisation of the three considered network types: a) SS-type (shallow network, scalar information), b) DS-type (deep network, scalar information) and c) DF-type (deep network, field information)

Each network is a feed-forward network whose layers are fully connected. All neurons use ReLu-activation. For each network type (SS, DS, DF) an extensive hyperparameter study was performed to determine an optimal number of neurons and layers. The studied networks range from 25 to 10 000 neurons per layer and 2 to 5 hidden layers. Since the number of parameters is much larger than the number of supplied data points, i.e. highly flexible network, measures were investigated to prevent overfitting, such as dropout, L1- and L2-regularisation, mini-batches and batch normalisation. Yet, only the mini-batches and batch normalisation proved useful. The selected network architectures are summarised in Fig. 4 a).

To evaluate each network’s data efficiency, different-sized databases $n \in \{100, 250, 500, 1000\}$ are sampled. Performance is evaluated on an additional, separate validation set with 100 samples. For both, training and validation set, Latin Hypercube sampling is used. Figure 4 b) and c) visualise the findings.

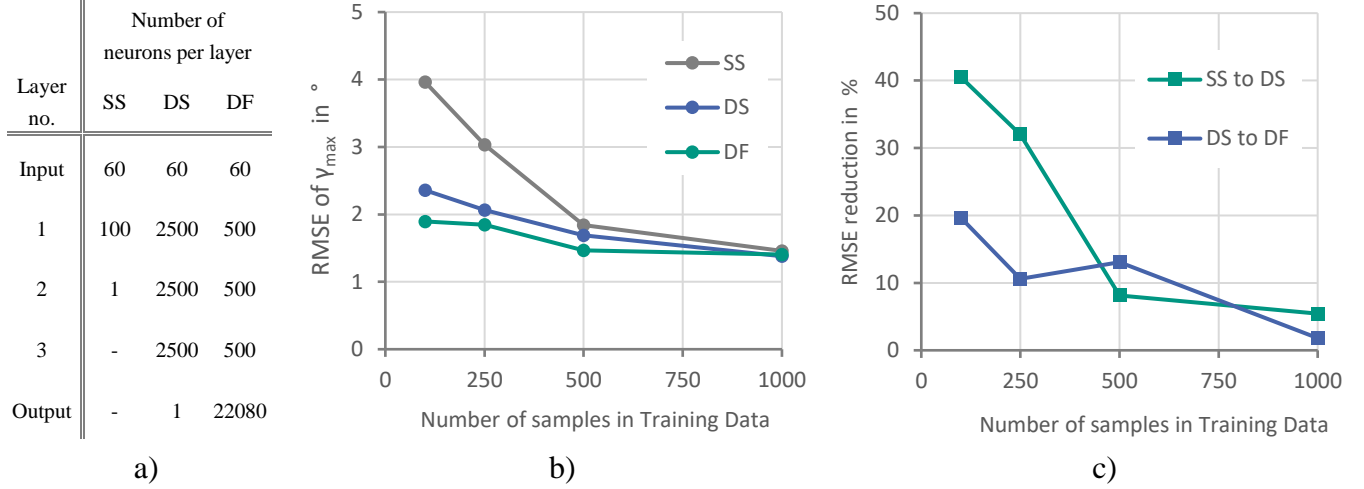


Fig. 4. Performance comparison of three different network types with a summary of their layer-architecture a) and b) absolute and c) relative predictive error.

More specifically, Fig. 4 b) shows the evolution of prediction accuracy as measured by root mean square error (RMSE) for each ANN. For all ANNs the RMSE reduces with more available data, underpinning validity of the universal approximation theorem. However, large performance differences appear when data becomes sparse, e.g. $n = 100$: although trained on the same data, the SS-type cannot capture γ as accurately as the DS- and the DF-type and results in a constantly higher RMSE. Similar holds for DS- and DF-type, albeit at lesser extent. Figure 4 c) quantifies this reduction: using additional layers, i.e. changing from SS to DS, reduces the error by $\approx 40\%$ (100 samples). Yet, this advantage gradually becomes less significant as more data becomes available – loosely speaking, the SS-network „catches up“. Similar hold for the additional change from DS to DF since an additional $\approx 20\%$ (100 samples) is apparent. Therefore, it may be stated, that both measures, deepening the network and training on full-field data, significantly improves predictive accuracy, especially in sparse-data situations.

3.2 Optimisation performance

In SBO, sequential surrogate refinement with new samples is essential, for which two different paradigms prevail: samples can be placed either in parameter regions with little evidence to facilitate discovery of new, potentially better optima (“exploration”), or near already localised optima for perfection of parameter combinations (“exploitation”).

This work studies the exploration-exploitation-balance by two different hyperparameters for optimisation configuration: one hyperparameter is the initial database size n . In general, greater values of n introduce more prior information to the surrogate. Thus, it can directly exploit the most promising regions and spend less effort on additional exploration. Obviously, this comes at the cost of increased effort prior to optimisation. The second hyperparameter is the number of ANN-training episodes k_{trn} during SBO-loops (cf. Fig. 2). In general, the higher k_{trn} , the more emphasis lies on new samples during optimisation and the stronger the attraction of an optimum, i.e. stronger exploitation. Both parameters comprise two levels, $n \in \{n_{min}; n_{max}\} = \{100; 1000\}$ and $k_{trn} \in \{k_{min}; k_{max}\} = \{2; 25\}$, respectively.

The vector-norm $p_m = \|\mathbf{y}\|_m = (\sum_i |\gamma_i|^m)^{-m}$ constitutes the objective function. While it includes the maximum norm ($m = \infty$) and the sum of all values ($m = 1$) as limit cases, this work employs $m = 4$ as a tradeoff between suppression of maximum shear and formation of shear angles in general. Figure 5 shows the evolution of the objective function $p_4(\mathbf{a}_{sim}^*)$ and the surrogate prediction $p_4(\mathbf{a}_{surr}^*)$ during optimisation. To allow investigation of long term behaviour, no automatic stopping criterion is set, but optimisations are terminated manually when both, a minimum iteration number $i_{min} = 450$ and a minimum predictive error of $\Delta p_4 = |p_4(\mathbf{a}_{sim}^*) - p_4(\mathbf{a}_{surr}^*)| \approx 3^\circ$, are reached.

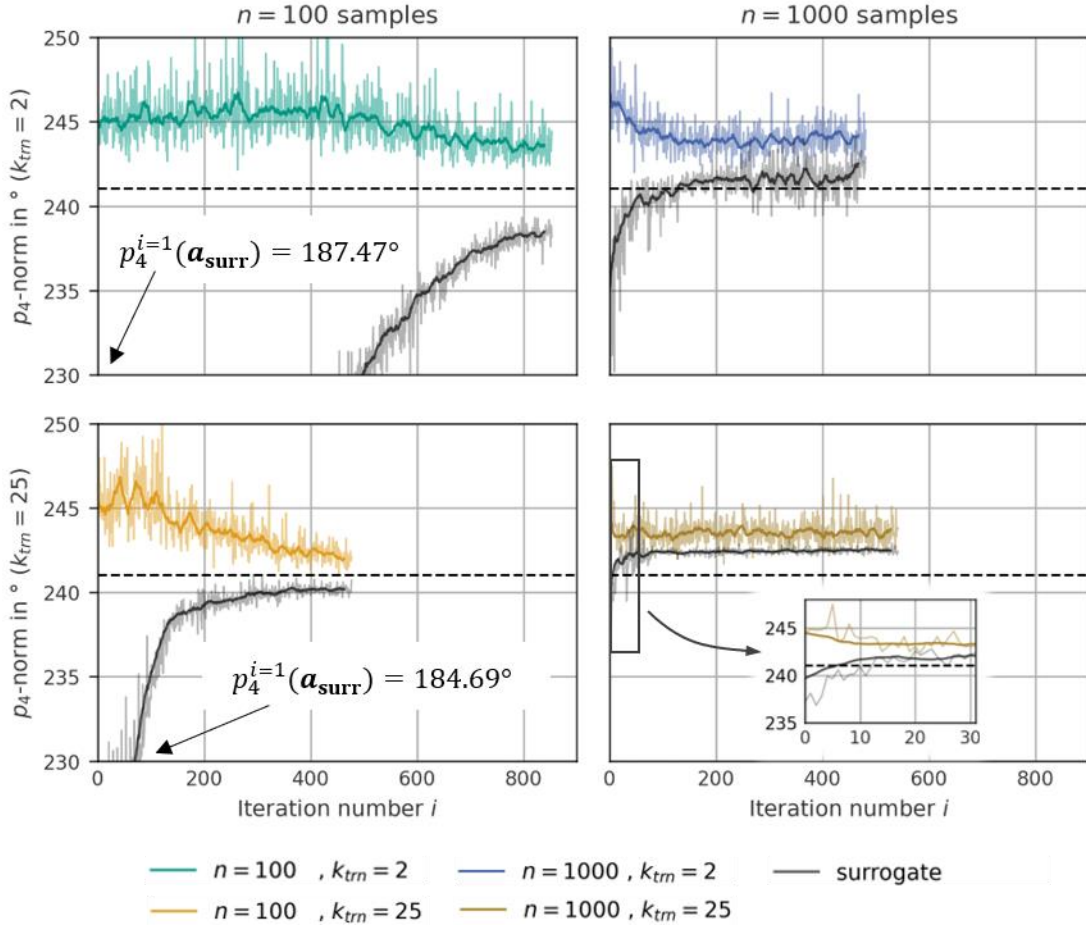


Fig. 5. Optimisation progress for each SBO-configuration. Surrogate predictions $p_4(\mathbf{a}_{\text{surr}})$ are given in black (moving average) and gray, while the coloured graphs represent simulation results $p_4(\mathbf{a}_{\text{sim}})$. The initial $p_4(\mathbf{a}_{\text{surr}})$ -values of the left column are omitted in the plot for readability. Their respective values are directly printed for reference.

Each subplot refers to a combination of hyperparameters and shows two different graphs, the surrogate prediction $p_4(\mathbf{a}_{\text{surr}}^*)$ and the actual simulation result $p_4(\mathbf{a}_{\text{sim}}^*)$. To alleviate the erratic appearance of the graphs, a moving-average over $m_{\text{avg}} = 15$ iterations smoothes each curve. The dashed horizontal line denotes the best value of the objective function $p_4^*(\mathbf{a}_{\text{surr}}^*)$ ever found in this work. It is deemed the best available approximation of the true – but unknown – optimum and is thus used for reference during subsequent algorithm comparison.

Some volatility aside, all graphs show three common characteristics: first, the objective function $p_4(\mathbf{a}_{\text{sim}}^*)$ overall decreases which validates the general suitability of “full field”-DNNs as surrogate models. Second, the surrogate predictions approach the simulation results, corroborating the successful learning process on new samples. Third, the surrogate constantly underestimates simulation results.

Main differences between the graphs lie in their shape with two different, column-wise characteristics. Overall, in both columns the graphs maintain their general shape, yet, the bottom plots are significantly compressed to the left: In the left column ($n = 100$ samples), the graphs of $p_4(\mathbf{a}_{\text{sim}}^*)$ initially waver around an approximately constant value, while the surrogate $p_4(\mathbf{a}_{\text{sim}}^*)$ constantly improves its accuracy. In iteration $i \approx 550$ (top) and $i \approx 130$ (bottom), the surrogate improvements become slower and the objective function $p_4(\mathbf{a}_{\text{sim}}^*)$ begins to decline. This hints that the surrogate first eliminates its bias (“exploration”) and, with sufficient accuracy, starts converging to an optimum.

The opposite holds for the right column (large database with $n = 1000$), where both graphs show an initial descent and, from iteration $i \approx 120$ (top) or $i \approx 10$ (bottom) onwards, some wavering around a constant value. Since the surrogate is relatively accurate already at start it may directly converge to an optimum without further exploration. However, the absolute value of the objective function is higher than on the left, which implies a local rather than a global optimum.

For final efficiency assessment, Fig. 6 visualises the convergence of each configuration along with a (non-surrogate) evolutionary algorithm (EA) from the DAKOTA-toolbox on default settings [18]. The graphs show the evolution of $p_4(\mathbf{a}_{sim}^*)$, i.e. the FE-simulation with the so-far-best quality metric. Formally, it is the lower envelope of the erratic graphs in Fig. 5. Note that due to initial database-sampling the graphs are offset by $n = 100$ and $n = 1000$ simulations, respectively.

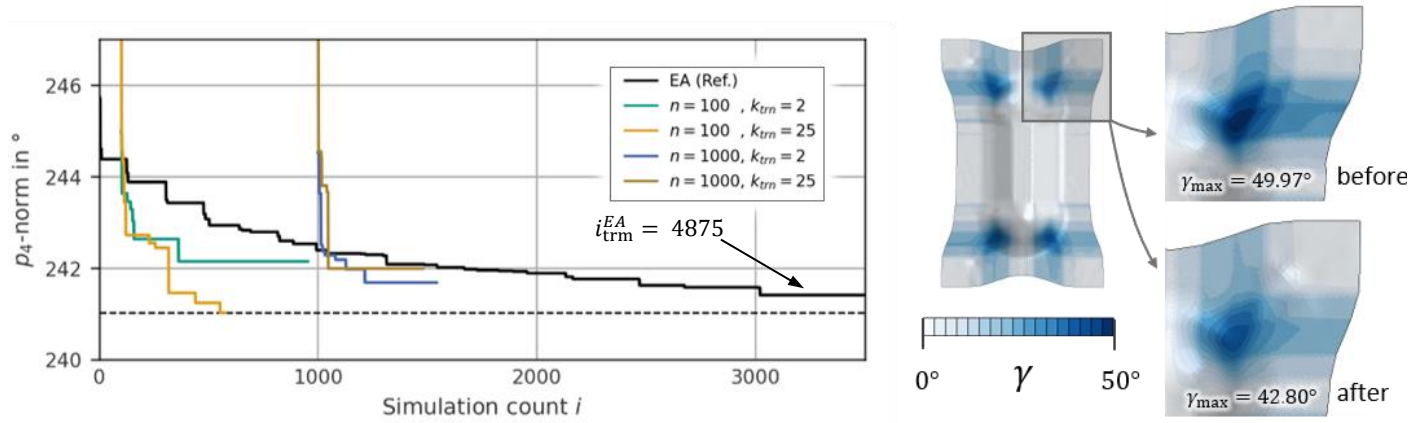


Fig. 6. Convergence of a classical evolutionary algorithm (EA) and each SBO-configuration (left). The EA terminated in iteration $i_{trm}^{EA} = 4875$ without further improvement (omitted for readability). Forming results before and after optimisation show a successful reduction of γ_{max} (right).

The graphs distil the essence of the previous plots: despite a faster descent of the objective function, the optimisation results with the large database D^{1000} are inferior to the smaller database D^{100} . Not just in terms of quality but also efficiency: While configuration “ $n = 100, k_{trn} = 25$ ” has found its final result in iteration $i \approx 600$, the “ $n = 1000$ ”-configurations require $i \approx 1100$ and $i \approx 1300$ iterations, respectively. Please note, that each SBO outperforms the EA in terms of efficiency and “ $n = 100, k_{trn} = 25$ ” also in terms of quality. Please also note, that “ $n = 100, k_{trn} = 2$ ” might have given a similarly good result, but was manually terminated due to excessive computation time (>10 weeks). At this time, the objective function was still descending, albeit at slow rate (cf. Fig. 5 top left). From an engineering perspective, Fig. 6 (right) shows a successful reduction of maximum shear by $\approx 7.2^\circ$ or $\approx 14.3\%$, respectively. After optimisation, the grippers restrain the material draw-in such as to avoid local shear concentration but make the deformation stretch over a wider expanse.

4 Summary and Conclusion

This work examines the use of deep neural networks as surrogate models in virtual manufacturing process optimisation at the example of gripper-assisted textile forming. Different network types are compared. Best prediction performance is achieved using a deep neural network which predicts the full strain field instead of just a single performance scalar. The network is also integrated in an SBO-framework to study suitability and convergence behaviour during optimisation. Four SBO-configurations with different exploration-vs-exploitation balances are investigated. In each case, the developed SBO-framework outperformed a current state-of-the-art evolutionary algorithm in terms of efficiency. One case gave an even better result. Results further hint that “online”-simulations during SBO-loops contribute significantly more to convergence than “offline”-simulations from prior sampling.

Further research is still envisaged. The presented results show that a smaller database – and thus a less accurate surrogate (!) – can indeed lead to better optimisation results. This observation requires a more comprehensive investigation and more ideally, a quantifiable criterion towards selecting the size of the initial database. It also implies that global accuracy metrics, e.g. MSE, might not be the best measure for surrogate quality assessment in SBO. In the long term, surrogate models may also be equipped with additional capabilities: as shown in [19], convolutional neural networks (CNN) are able to learn system dynamics from data and predict physical effects in real-world engineering problems. First results for textile draping appear promising: [20] and [21] hint that CNNs can learn to assess formability of new components from generic draping examples. [22] further shows that – in principle – CNNs can additionally be used to estimate optimal process parameters for new components. Thus, DL-techniques appear a promising and efficient tool for process design at early stages of product development.

Acknowledgment

The authors would like to thank the German 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] S. Chen, L. T. Harper, A. Endruweit, N. A. Warrior: „Formability optimisation of fabric preforms by controlling material draw-in through in-plane constraints”, *Composite Part A*, 76, pp. 10-19, 2015
- [2] L. Kärger, S. Galkin, C. Zimmerling, D. Dörr, J. Linden, A. Oeckerath, K. Wolf: Forming optimisation embedded in a CAE chain to assess and enhance the structural performance of composite components. *Comp. Struct.* 192, pp. 143-152, 2018, DOI: 10.1016/j.compstruct.2018.02.041
- [3] Z. H. Han, K.S. Zhang: “Surrogate-Based Optimization” in *Real-World Applications of Genetic Algorithms*, Olympia Roeva, IntechOpen, DOI: 10.5772/36125
- [4] J. Jakumeit, M. Herdy, M. Nitsche: „Parameter optimization of the sheet metal forming process using an iterative parallel Kriging algorithm“ in *Structural and Multidisciplinary Optimization*, (29, 2005), pp. 498-507.
- [5] M.H.A. Bonte, A. H. van den Boogaard, J. Huétink: “A Metamodel Based Optimisation Algorithm for Metal Forming Processes” in *Adv. Methods in Material Forming* (Springer, Berlin/Heidelberg, 2007), pp. 55-72.
- [6] H. Wang, L. Chen, E. Li: „Time dependent sheet metal forming optimization by using Gaussian process assisted firefly algorithm“, *International Journal of Material Forming*, Vol. 11-2, pp. 279–295, 2018
- [7] J. Pfrommer, C. Zimmerling, J. Liu, L. Kärger, F. Henning, J. Beyerer: “Optimisation of manufacturing process parameters using deep neural networks as surrogate models”, *Proc. CIRP* 72, pp. 426–431, 2018, DOI: 10.1016/j.procir.2018.03.046
- [8] C. Zimmerling, J. Pfrommer, J. Liu, J. Beyerer, F. Henning, L. Kärger: “Application and evaluation of meta-model assisted optimisation strategies for gripper-assisted fabric draping in composite manufacturing” in *Proceedings of the 18th ECCM, Athens/Greece, 2018*, DOI: 10.5445/IR/1000099317
- [9] B. Gooijer, J. Hazrati, B. Geijselaers, T. van den Boogaard: On the choice of basis in proper orthogonal decomposition-based surrogate models. In: *Proceedings of 22nd ESAFORM 2019, Vitoria-Gasteiz, Spain, AIP Publishing (AIP Conference Proceedings)*, S. 100002, 2019, DOI: 10.1063/1.5112635
- [10] G. Misiun, C. Wang, H. Geijselaers, and A. van den Boogaard, “Interpolation of final geometry and result fields in process parameter space,” in *Numiform 2016*, Vol. 80 (2016), pp. 16010–1 – 16010–6.
- [11] M. Hamdaoui, G. Le Quilliec, P. Breitkopf, P. Villon: POD surrogates for real-time multi-parametric sheet metal forming problems. In: *Int J Mater Form* 7 (3), S. 337–358, 2014, DOI: 10.1007/s12289-013-1132-0.
- [12] D. Dörr, F.J. Schirmaier, F. Henning, L. Kärger: “A viscoelastic approach for modeling bending behavior in finite element forming simulation of continuously fiber reinforced composites”, *Composites Part A*, 94, pp. 113–123, 2017, DOI: 10.1016/j.compositesa.2016.11.027
- [13] C. Poppe, D. Dörr, F. Henning, L. Kärger: “Experimental and numerical investigation of the shear behaviour of infiltrated woven fabrics”, *Composite Part A*, 14, pp. 327-337, 2018, DOI: 10.1016/j.compositesa.2018.08.018
- [14] C. Poppe, T. Rosenkranz; D. Dörr, L. Kärger: “Comparative experimental and numerical analysis of bending behaviour of dry and low viscous infiltrated woven fabrics”, *Composite Part A*, 124, 2019, DOI: 10.1016/j.compositesa.2019.05.034
- [15] S. Koziel, L. Leifsson: “Surrogate-based Modeling and Optimization – Applications in Engineering”, Springer New York, New York/NY, 2013
- [16] K. Hornik: Approximation capabilities of multilayer feedforward networks. In: *Neural Networks* 4 (2), , 1991. DOI: 10.1016/0893-6080(91)90009-T
- [17] Y. LeCun, Y. Bengio, G. Hinton: Deep learning. In: *Nature* 521 (7553), S. 436–444, 2015, DOI: 10.1038/nature14539
- [18] B.M. Adams, W.J. Bohnhoff, K.R. Dalbey et al.: "Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.11 User's Manual", Sandia Technical Report SAND2014-4633, November 2019
- [19] X. Guo, W. Li, F. Iorio: Convolutional Neural Networks for Steady Flow Approximation. In: *Proceedings of the 22nd ACM*, S. 481–490, 2016
- [20] C. Zimmerling, D. Dörr, F. Henning, L. Kärger: A Machine learning assisted approach for textile formability assessment and design improvement of composite components; *Composites Part A*, 124, 2019, DOI: 10.1016/j.compositesa.2019.05.027
- [21] C. Zimmerling, D. Trippe, B. Fengler, L. Kärger: “An approach for rapid prediction of textile draping results for variable composite component geometries using deep neural networks”, *AIP Conference Proceedings*. Vol. 2113. No. 1. AIP Publishing, 2019, DOI: 10.1063/1.5112512
- [22] C. Zimmerling, C. Poppe, L. Kärger: Estimating Optimum Process Parameters in Textile Draping of Variable Part Geometries - A Reinforcement Learning Approach. In: *Procedia Manufacturing* 47, S. 847–854, 2020 DOI: 10.1016/j.promfg.2020.04.263.