Inverse Process-Structure-Property Mapping

Norbert Link[1], Tarek Iraki[1], and Johannes Dornheim[1]

Intelligent Systems Research Group,
Karlsruhe University of Applied Sciences
[tarek.iraki, johannes.dornheim, norbert.link]@hs-karlsruhe.de

Abstract. Workpieces for dedicated purposes must be composed of materials which have certain properties. The latter are determined by the compositional structure of the material. In this paper, we present the scientific approach of our current DFG funded project Tailored material properties through microstructural optimization: Machine Learning Methods for the Modeling and Inversion of Structure-property relationships and their application to sheet metals. The project proposes a methodology to automatically find an optimal sequence of processing steps which produce a material structure that bears the desired properties. The overall task is split in two steps: First find a mapping which delivers a set of structures with given properties and second, find an optimal process path to reach one of these structures with least effort. The first step is achieved by machine learning the generalized mapping of structures to properties in a supervised fashion, and then inverting this relation with methods delivering a set of goal structure solutions. The second step is performed via reinforcement learning of optimal paths by finding the processing sequence which leads to the best reachable goal structure. The paper considers steel processing as an example, where the microstructure is represented by Orientation Density Functions and elastic and plastic material target properties are considered. The paper shows the inversion of the learned structure-property mapping by means of Genetic Algorithms. The search for structures is thereby regularized by a loss term representing the deviation from process-feasible structures. It is shown how reinforcement learning is used to find deformation action sequences in order to reach the given goal structures, which finally lead to the required properties.

Keywords: Computational Materials Science, Property-Structure-Mapping, Texture Evolution Optimization, Machine Learning, Reinforcement Learning

1 Introduction

The derivation of processing control actions to produce materials with certain, desired properties is the "inverse problem" of the causal chain "process control" - "microstructure instantiation" - "material properties". The main goal of our current project is the creation of a new basis for the solution of this problem by using modern approaches from machine learning and optimization.

The inversion will be composed of two explicitly separated parts: "Inverse Structure-Property-Mapping" (SPM) and "microstructure evolution optimization". The focus of the project lies on the investigation and development of methods which allow an inversion of the structure-property-relations of materials relevant in the industry. This inversion is the basis for the design of microstructures and for the optimal control of the related production processes. Another goal is the development of optimal control methods yielding exactly those structures which have the desired properties. The developed methods will be applied to sheet metals within the frame of the project as a proof of concept. The goals include the development of methods for inverting technologically relevant "Structure-Property-Mappings" and methods for efficient microstructure representation by supervised and unsupervised machine learning. Adaptive processing path-optimization methods, based on reinforcement learning, will be developed for adaptive optimal control of manufacturing processes.

We expect that the results of the project will lead to an increasing insight into technologically relevant process-structure-property-relationships of materials. The instruments resulting from the project will also promote the economically efficient development of new materials and process controls.

2 Related Work

In general, approaches to microstructure design make high demands on the mathematical description of microstructures, on the selection and presentation of suitable features, and on the determination of structure-property relationships. For example, the increasingly advanced methods in these areas enable Microstructure Sensitive Design (MSD), which is introduced in [1] and [2] and described in detail in [3].

The relationship between structures and properties descriptors can be abstracted from the concrete data by regression in the form of a Structure-Property-Mapping. The idea of modeling a Structure-Property-Mapping by means of regression and in particular using artificial neural networks was intensively pursued in the 1990s [4] and is still used today. The approach and related methods presented in [5] always consist of a Structure-Property-Mapping and an optimizer (in [5] Genetic Algorithms) whose objective function represents the desired properties.

The inversion of the SPM can be alternatively reached via Generative Models. In contrast to discriminative models (e.g. SPM), which are used to map conditional dependencies between data (e.g. classification or regression), generative models map the composite probabilities of the variables and can thus be used to generate new data from the assumed population. Established, generative methods are for example Mixture Models [6], Hidden Markov Models [7] and in the field of artificial neural networks Restricted Boltzmann Machines [8]. In the field of deep learning, generative models, in particular generative adversarial networks [9], are currently being researched and successfully applied in the context of image processing. Conditional Generative Models can generalize the probability of occurrence of structural features under given material properties. In this way, if desired, any number of microstructures could be generated.

Based on the work on the SPM, the process path optimization in the context of the MSD is treated depending on the material properties. For this purpose, the process is regarded as a sequence of structure-changing process operations which correspond to elementary processing steps. Shaffer et al. [10] construct a so called *texture evolution network* based on process simulation samples, to represent the process. The texture evolution network can be considered as a graph with structures as vertices, connected by elementary processing steps as edges. The structure vertices are points in the structure-space and are mapped to the property-space by using the SPM for property driven process path optimization. In [11] one-step deformation processes are optimized to reach the most reachable element of a texture-set from the inverse SPM. Processes are represented by so called *process planes*, principal component analysis (PCA) projections of microstructures reachable by the process. The optimization then is conducted by searching for the process plane which best represents one of the texture-set elements. In [12], a generic ontology based semantic system for processing path hypothesis generation (MATCALO) is proposed and showcased.

3 Research Concept

3.1 Inverse Structure-Property-Mapping (SPM)

The required mapping of the structures to the properties is modeled based on data from simulations. The simulations are based on Taylor models. The structures are represented using textures in the form of orientation density functions (ODF), from which the properties are calculated. In the investigations, elastic and plastic properties are considered in particular. Structural features are extracted from the ODF for a more compact description. The project uses spectral methods such as generalized spherical harmonics (GSH) to approximate the ODF. As an alternative representation we investigate the discretization in the orientation-space, where the orientation density is represented by a histogram.

The solution of the inverse problem consists of a Structure-Property-Mapping and an optimizer: As [4] described, the SPM is modeled by regression using artificial neural networks. In this investigation, we use a multilayer perceptron.

Differential evolution (DE) is used for the optimization problem. DE is an evolutionary algorithm developed by Rainer Storn and Kenneth Price [13]. It is a optimization method, which repeatedly improves a candidate solution set under consideration of a given quality measure over a continuous domain. The DE algorithm optimizes a problem by taking a population of candidate solutions and generating new candidate solutions (structures) by mutation and recombination existing ones. The candidate solution with the best fitness is considered for further processing. So, for the generated structures the reached properties are determined using the SPM.

The fitness $\mathcal{F}$ is composed of two terms: The property loss $\mathcal{L}_{\mathcal{P}}$, which expresses, how close the property of a candidate is to the target property, and the structure loss $\mathcal{L}_{\mathcal{S}}$, which represents the degree of feasibility of the candidate structure in the process

$$\mathcal{F}(\vec{s}, \hat{\vec{s}}, \vec{p_r}, \vec{p_d}) = \mathcal{L}_{\mathcal{P}}(\vec{p_r}, \vec{p_d}) + \mathcal{L}_{\mathcal{S}}(\vec{s}, \hat{\vec{s}}), \tag{1}$$

The property loss is the mean squared error (MSE) between the reached properties $p_r \in \vec{p_r}$ and the desired properties $p_d \in \vec{p_d}$:

$$\mathcal{L}_{\mathcal{P}}(\vec{p_r}, \vec{p_d}) = \frac{1}{N} \sum_{i=1}^{N} (p_{ri} - p_{di})^2 \tag{2}$$

Considering the goal that the genetic algorithm generates reachable structures, a neural network is formed which functions as an anomaly detector. The data basis of this neural network are structures that can be reached by a process. The goal of anomaly detection is to exclude unreachable structures. The anomaly detection is implemented using an autoencoder [14]. This is a neural network (see Fig. 1) which consists of the following two parts: the encoder and the decoder. The encoder converts the input data to an embedding space. The decoder converts the embedding space as close as possible to the original data. Due to the reduction to an embedding space, the autoencoder uses data compression and extracts relevant features. The cost function for the structures is a distance function in the ODF-space, which penalizes the network if it produces outputs that differ from the input. The cost function is also known as the *reconstruction loss*:

$$\mathcal{L}_{\mathcal{S}}(\vec{s}, \hat{\vec{s}}) = \sum_{i=1}^{N} \frac{(s_i - \hat{s}_i)^2}{(s_i + \hat{s}_i + \lambda)}, \tag{3}$$

with $s_i \in \vec{s}$ as the original structures, $\hat{s}_i \in \hat{\vec{s}}$ as the reconstructed structures and $\lambda = 0.001$ to avoid division by zero.

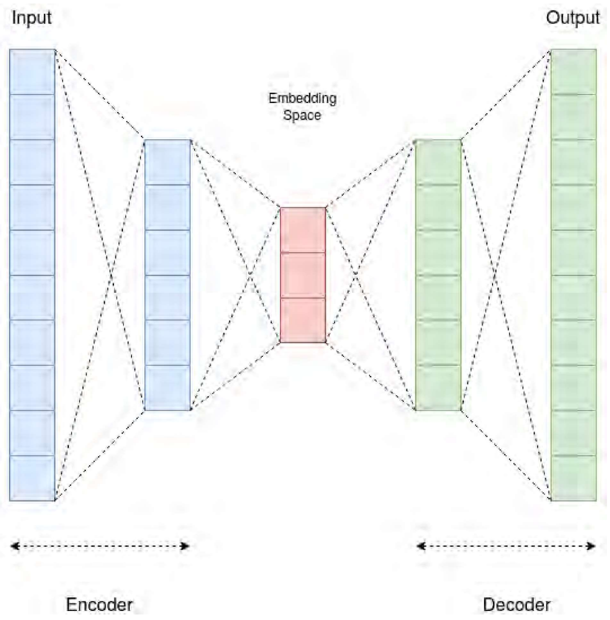

Fig. 1. Autoencoder for determining the structure loss

When using the anomaly detection, the autoencoder determines a high reconstruction loss if the input data are structures that are very different from the reachable structures. The overall approach is shown in Fig. 2 and consists of the following steps:

1. The genetic algorithm generates structures.
2. The SPM determines the reached properties of the generated structures.
3. The structure loss $\mathcal{L}_{\mathcal{S}}$ is determined by the reconstruction loss of the anomaly detector for the generated structures with respect to the reachable structures.
4. The property loss $\mathcal{L}_{\mathcal{P}}$ is determined by the MSE of the reached properties and the desired properties.
5. The fitness is calculated as the sum of the structure loss $\mathcal{L}_{\mathcal{S}}$ and the property loss $\mathcal{L}_{\mathcal{P}}$.

The structures, resulting from the described approach form the basis for optimal process control.

3.2 Texture Evolution Optimization

Due to the forward mapping, the process evolution optimization based on texture evolution networks ([10]) is restricted to a-priori sampled process paths. [11] relies on linearization assumptions and is applicable to short process sequences only. [12] relies on a-priori learned process models in the form of regression trees and is also applicable to relatively short process sequences only.

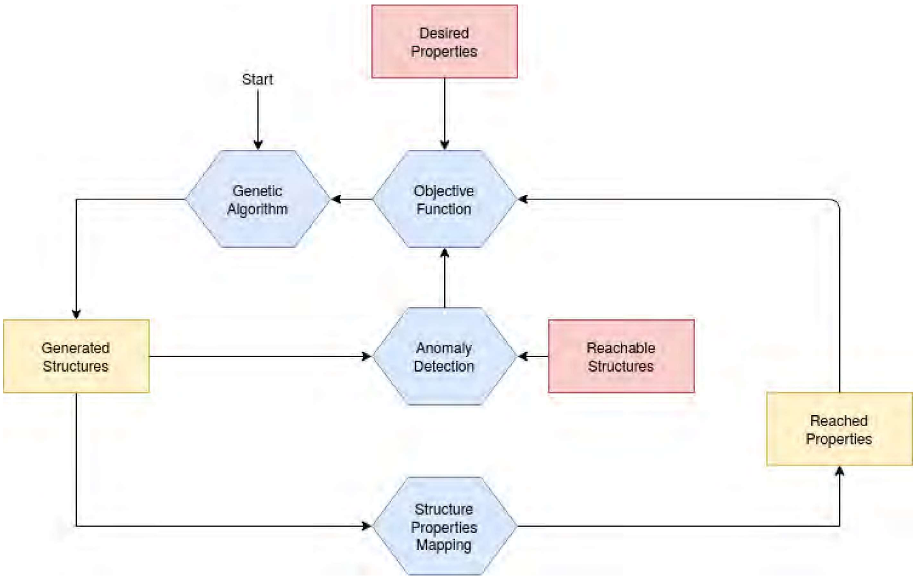

Fig. 2. Overview of our approach to find reachable structures with desired properties

As an adaptive alternative for texture evolution optimization, that can be trained to find process-paths of arbitrary length, we propose methods from reinforcement learning. For desired material properties $\vec{p_d}$. The inverted SPM determines a set of goal micro-structures $\vec{s_d} \in G$, which are very likely reachable by the considered deformation process. The texture evolution optimization objective is then to find the shortest process path $\mathcal{P}^*$ starting from a given structure $\vec{s_0}$, and leading close to one of the structures from G.

$$\mathcal{P}^* = \arg\min |\mathcal{P}| : E(s_0, \mathcal{P}) \in G_\tau, \tag{4}$$

where $\mathcal{P} = (a_k)_{k=0,...,K}; K \leqq T$ is a path of process actions a, T is the maximum allowed process length. The mapping $E(s, \mathcal{P}) = s_k$ delivers the resulting structure, when applying $\mathcal{P}$ to the structure s. Here, for the sake of simplicity, we assume the process to be deterministic, although the reinforcement learning methods we use are not restricted to deterministic processes. G_τ is a neighbourhood of G, the union of all open balls with radius τ and center points from G.

To solve the optimization problem by reinforcement learning approaches, it must be reformulated as markov decision process (MDP), which is defined by the tuple (S, A, P, R). In our case S is the space of structures $\vec{s}$, A is the parameter-space of the deformation process, containing process actions $\vec{a}$, $P : S \times A \mapsto S$ is the transition function of the deformation process, which we assume to be deterministic. $R_g : S \times A \mapsto \mathbb{R}$ is a goal-specific reward function. The objective of the reinforcement learning agent is then to find the optimal goal-specific policy $\pi_g^*(s_t) = a_t$ that maximizes the discounted future goal-specific reward

$$\mathcal{V}_g(s_t) = \sum_{k=t}^{K} \gamma^{k-t} R_g(s_k, a_k), \tag{5}$$

where $\gamma \in [0, 1]$ discounts early attained rewards, the policy $\pi_g(s_k)$ determines a_k and the transition function $P(s_k, a_k)$ determines s_{k+1}.

For a distance function d in the structure space, the binary reward function

$$R_g(s,a) = \begin{cases} 1, & \text{if } d(P(s,a),g) < \tau \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

if maximized, leads to an optimal policy π_g^* that yields the shortest path to g from every s for $\gamma < 1$. Moreover, if $\mathcal{V}_g$ is given for every microstructure from G, $\mathcal{P}$ from eq. 4 is identical with the application of the policy π_ζ^*, where $\zeta = \arg\max_g[\mathcal{V}_g]$.

π_g^* can be approached by methods from reinforcement learning. Value-based reinforcement learning is doing so by learning expected discounted future reward functions [15]. One of these functions is the so called *value-function* V. In the case of a deterministic MDP and for a given g, this expectation value function reduces to $\mathcal{V}_g$ from eq. 4 and ζ can be extracted if V is learned for every g from G. For doing so, a generalized form of expectation value functions can be learned as it is done e.g. in [16].

This exemplary MDP formulation shows how reinforcement learning can be used for texture evolution optimization tasks. The optimization thereby is operating in the space of microstructures and does not rely on a-priori microstructure samples. When using off-policy reinforcement learning algorithms and due to the generalization over goal-microstructures, the functions learned while solving a specific optimization task can be easily transferred to new optimization tasks (i.e. different desired properties or even a different property space).

References

1. Adams, B.L., Henrie, A., Henrie, B., Lyon, M., Kalidindi, S., Garmestani, H.: Microstructure-sensitive design of a compliant beam. Journal of the Mechanics and Physics of Solids **49**(8) (2001) 1639–1663
2. Kalidindi, S.R., Houskamp, J.R., Lyons, M., Adams, B.L.: Microstructure sensitive design of an orthotropic plate subjected to tensile load. International Journal of Plasticity **20**(8-9) (2004) 1561–1575
3. Fullwood, D.T., Niezgoda, S.R., Adams, B.L., Kalidindi, S.R.: Microstructure sensitive design for performance optimization. Progress in Materials Science **55**(6) (2010) 477–562
4. Sumpter, B.G., Noid, D.W.: On the design, analysis, and characterization of materials using computational neural networks. Annual Review of Materials Science **26**(1) (1996) 223–277
5. Kuroda, M., Ikawa, S.: Texture optimization of rolled aluminum alloy sheets using a genetic algorithm. Materials Science and Engineering: A **385**(1-2) (2004) 235–244
6. McLachlan, G., Peel, D.: Finite mixture models. John Wiley & Sons (2004)
7. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2) (1989) 257–286
8. Smolensky, P.: Information processing in dynamical systems: Foundations of harmony theory. Technical report, Colorado Univ at Boulder Dept of Computer Science (1986)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014) 2672–2680
10. Shaffer, J.B., Knezevic, M., Kalidindi, S.R.: Building texture evolution networks for deformation processing of polycrystalline fcc metals using spectral approaches: applications to process design for targeted performance. International Journal of Plasticity **26**(8) (2010) 1183–1194
11. Acar, P., Sundararaghavan, V.: Linear solution scheme for microstructure design with process constraints. AIAA Journal (2016) 4022–4031
12. Picklum, M., Beetz, M.: MATCALO: Knowledge-enabled machine learning in materials science. Computational Materials Science **163**(December 2018) (2019) 50–62

13. Storn, R., Price, K.: Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces (1995)
14. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
15. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
16. Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, O.P., Zaremba, W.: Hindsight experience replay. In: Advances in Neural Information Processing Systems. (2017) 5048–5058