Simulation in Produktion
und Logistik 2021
Jörg Franke & Peter Schuderer (Hrsg.)
Cuvillier Verlag, Göttingen 2021

Combining Engineering Data, Sensor Data and Artificial Intelligence for Automated Edge Network Infrastructures

Verknüpfung von Engineering-Daten, Sensordaten und künstlicher Intelligenz für automatisierte Edge-Netzwerkinfrastrukturen

André Hürkamp, Franziska Aschersleben, Marvin Czarski, Sebastian Gellrich, Christoph Herrmann, Klaus Dröder, Technische Universität Braunschweig, Institute of Machine Tools and Production Technology, Braunschweig (Germany),
a.huerkamp@tu-braunschweig.de, f.aschersleben@tu-braunschweig.de,
m.czarski@tu-braunschweig.de, s.gellrich@tu-braunschweig.de,
c.herrmann@tu-braunschweig.de, k.droeder@tu-braunschweig.de

Abstract: The paper describes a concept of monitoring and controlling flexible production processes by means of engineering data, sensor data and artificial intelligence for an automatic control by edge-computing. In particular, surrogate models from numerical simulations obtained at the engineering phase are derived using Proper Orthogonal Decomposition and machine learning. Among the six investigated methods, Random Forest yields the best surrogate model in terms of accuracy and real-time feasibility on the edge server for the use case of thermoplastic composites.

1 Introduction

For a real-time capable monitoring and control of flexible production systems, status information on the process and the components at different process steps are needed. Ideally, that information can be obtained without time and cost intensive component testing and without occupying the IT infrastructure with too much raw data.

Decentralized data fusion of real-time capable and valid quality gates at the edge of the factory network can be a solution to overcome these burdens. Process knowledge from available engineering data (i.e., simulation results) together with sensor data and methods of artificial intelligence can be combined to form the basis of virtual quality gates with reduced data streams. Furthermore, the use of quality gates at the individual process and provision of consolidated KPIs at the process chain level (e.g. error probability, source of error, etc.) provides a powerful real-time capable monitoring and control of a flexible production system.

The “Incremental Manufacturing Lab” (IML) of TU Braunschweig is such a flexible manufacturing cell that combines different tool-independent manufacturing technologies. It consists of a welding robot for 3D-printing of steel with high deposition rates, a robot for extrusion of thermoplastics that can also be used for surface machining and handling as well as a milling robot for post-processing. In order to be able to implement a lean solution for improving the part quality and process efficiency, a data processing infrastructure is required that can be universally applied to the process chain, is sufficiently powerful and enables low latencies.

The objective of this contribution is the development of real-time applications on edge devices that are able to learn from simulation data after the engineering phase. In this way, systematic deviations from the simulation model can be corrected online and under real-time conditions with the stream of machine, process, and sensor data. Such an approach is also described as the concept of hybrid twins (Chinesta et al., 2018; Sancarlos et al., 2020).

2 State of the Art

Incremental manufacturing is a concept for a flexible production system that combines additive manufacturing, subtractive manufacturing processes and other production technologies like forming or injection moulding. Low-cost semi-finished parts serve as the basis and functionalities can be added incrementally e.g. by injection moulding, individualized by screw-based extrusion and post-processed by milling. The main advantage over conventional, serially linked manufacturing strategies is the flexibility of the value chain, so that the high customer requirements for individualization of products can be met since the design space is enlarged in contrast to traditional manufacturing technologies and complex geometries can be realized. In addition, the production process is scalable due to the partially tool-free manufacturing, so that a wide variety of materials can be used and product variants can be produced (Reichler et al., 2019; Dröder et al., 2019). However, due to the high flexibility, the process planning, supervision and quality assurance are challenging. Therefore, real-time data is needed for in-process quality monitoring especially for parametrization e.g. for different materials or in changing environments.

In today's product and production engineering, simulation tools are a matter of course to support development processes. Computer simulations are able to analyse the material behaviour of products and processes using numerical methods based on physical principles (Panchal et al., 2013; Allison et al., 2006). However, those simulations are computational expensive and time consuming (Zimmerling et al., 2020). Due to that, surrogate modelling can be applied to reach computationally feasible models (Han and Zhang, 2012). Applications can be found in the field of biomechanics (Liang et al., 2018), composite manufacturing (Pfrommer et al., 2018; Zambal et al., 2018; Hürkamp et al., 2020a) and in the context of virtual quality gates (Hürkamp et al., 2020b).

The advent of low-cost information and communication technology (ICT) has enabled various industries to employ data gathering and data processing for their production. With the goal to automatically make decisions and predict outcomes regarding the quality and progression of a product (Thiede et al., 2019; Filz et al., 2020). A challenge is the increasing complexity, the high dimensionality of the available data and the mostly chaotic structures. A viable solution for solving these problems is the application of artificial intelligence (AI) methods, especially with supervised AI models (Wuest et al., 2016). A manufacturer usually employs centralized or cloud computing to train AI models. Since there are limitations imposed by the network infrastructure regarding the data transfer it may not be possible to deploy the AI models, because they require a constant stream of live-data and so it may be necessary to perform on-site signal pre-processing (Wang et al., 2015).

A technology to perform the processing on site is edge computing. The objective of edge computing is to process data as close as possible to the source. With edge computing the computational load can be distributed between edge devices in one network to reduce the computation time. It can extend one manufactures capabilities in the domains: computation, networking and storage with the benefit of reducing latency and bandwidth (Wu et al., 2017).

3 Methods

The proposed method is based on the edge computing concept for flexible manufacturing, which is displayed in Figure 1 on the right side. The foundation is the flexible process chain in the bottom.

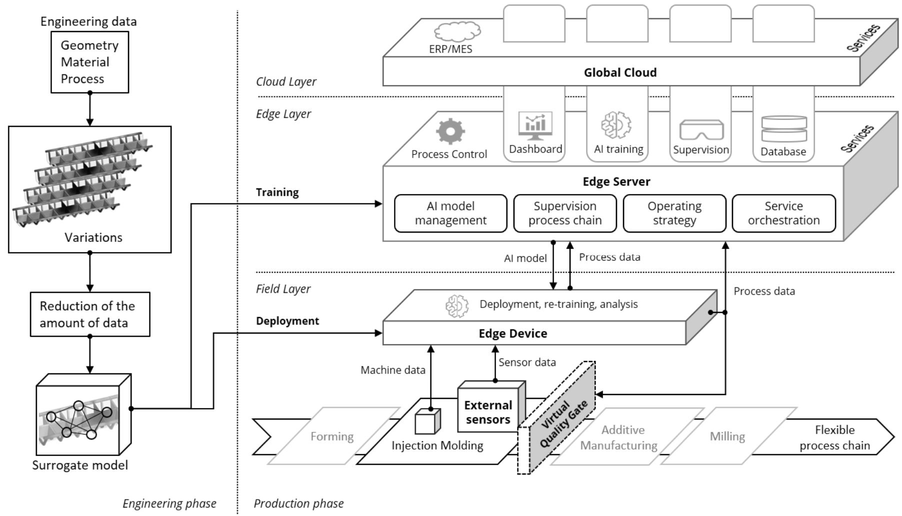

Figure 1: *Edge computing concept for flexible manufacturing cell*

An exemplary chain is shown with focus on the here utilized injection moulding. Each process step is equipped with external sensors and internal measurement capabilities. These sensor data are then transferred to an edge device that runs a pre-processing to decrease and aggregate the large amount of data to be further processed. AI methods are deployed for a fast and intelligent analysis. Especially simulation data from the engineering phase (Figure 1 on the left side) are used to train a surrogate model that is able to predict process results on an edge device in real time.

The edge device is mounted closely to the process, so that a very low latency can be achieved. Therefore, it is still assigned to the field layer. In the second layer, the process-superordinate calculations take place on the edge server. The forwarded data from the edge devices are processed to derive parameter adjustments for the process. In addition, the process data is visualized here and the entire process chain is monitored by evaluating virtual quality gates, which are provided after each process step, based on the processed data. Possibilities for compensating for errors in the subsequent process are included here. The edge server is therefore the main computing unit, where also management and training of AI models is performed due to a large computational capacity. In addition, the process data required for training is stored in a database here. All these modules are executed as a service to maximize the flexibility by using Kubernetes. This means that services can be moved to the cloud layer in order to use central computing units, which leads to better resource utilization and but also an increase in latency. The associated orchestration evaluates the requirements for the service, such as the maximum tolerated latency and the computing overhead in order to distribute the services reasonably. This data is subsequently transformed into a real-time capable surrogate model by machine learning that can be applied to edge devices. Embedded into networking interfaces this model can function as the digital representation of the production system (digital twin).

In order to improve this digital twin continuously in the further course of production, suitable sensors provide data with which additional production data is collected. For this purpose, algorithms are being developed to provide robust and reliable automation of flexible manufacturing processes. They will thus provide solutions for automated and decentralized edge infrastructure processes to achieve better process automation and in-line quality control. The goal is to provide methods and infrastructure that can leverage services with high performance and scalability requirements.

3.1 Numerical Modelling and Data Sampling

For the development of the digital twin, numerical parameter studies are performed during the engineering phase. These methods are able to represent the physical interactions of process and material properties and to determine three-dimensional process-dependent product properties. As a use case, the thermoplastic composite structure shown in Figure 2 a) is used. It consists of a continuously fibre reinforced thermoplastic (organo sheet) as a planar part insert and injected polymer ribs on top of it. The structure is designed for the manufacturing of testing specimens in order to experimentally determine the interface bond strength since this is limiting factor in such thermoplastic composites.

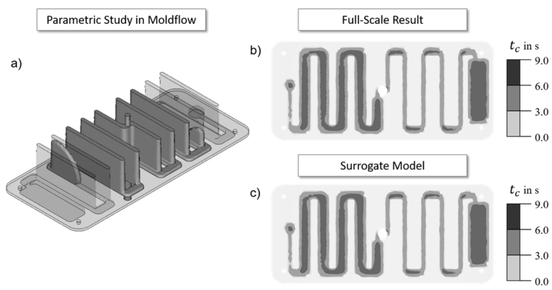

Figure 2: *a) Investigated thermoplastic composite structure and exemplary result of b) the full-scale computation and c) the corresponding surrogate*

During the engineering phase, potential process and material variations are captured in the resulting data set of numerical studies that cover the entire process parameter space. Within a Latin Hypercube Sampling (LHS), the process parameters part insert temperature $20 \leq T_{insert} \leq 240$°C, the mould temperature $30 \leq T_{mould} \leq 80$°C and the flow rate $10 \leq v_{inj} \leq 100$ cm^3/s are varied. For the use case, 100 parameter combinations are used as input for simulations of the injection moulding process. As target value, the contact time t_c is computed for each individual simulation. It describes the time the interface between part insert and injected polymer is heated above melting temperature of the used polymer. This variable has been identified to correlate directly with the final bond strength and hence the part quality (Hürkamp et al., 2021). The interface between organo sheet and injected polymer consists of 8196 nodes. Hence, each solution contains 8196 values describing the spatial distribution of the contact time t_c at the interface (cf., Figure 2 b) and c)).

3.2 Combining Proper Orthogonal Decomposition and Data-driven Surrogate Modelling

The LHS produces a large amount of data that needs to be processed. For this purpose, a Proper Orthogonal Decomposition (POD) is used, which on the one hand reduces the amount of data and on the other hand provides a basis for model reduction and surrogate modelling. POD has been proven to be a powerful tool in combination with machine learning for the design of surrogates with physical constraints (Swischuk et al., 2019). The POD of an arbitrary solution vector U can be written as

$$U(x,p) \approx \sum_{i=1}^{k} \phi_{i(x)} \cdot a_i(p), \tag{1}$$

where $a_i = \phi_i^T u$ denotes the POD coefficients. The vector $\phi_i(x)$ is referred to as spatial mode ad contains the geometric information. In contrary, the coefficients $a_i(p)$ depend only on the input p, which in the present case represents the set of injection moulding parameters. A reduced representation of the solution is obtained when $k < n$ (with n denoting the number of precomputed solutions) modes are used in the series expansion (1). The goal of the subsequent machine learning procedure is the prediction of the POD coefficients a_i in dependence of the process parameters p.

Consequently, in contrast to previous studies on surrogate modelling of FEA data (Hürkamp et al., 2020b), within this study the reduced FEA data set by POD is taken as input for machine learning. This leads to three major differences in the data set characteristics. First, the target variable is no longer a physical property (e.g., temperature or contact time), but multiple POD coefficients per target variable (here: first 50 POD coefficients a_i). Due to this, a multi-target regression approach is pursued. Second, the feature set only consists of the simulated (and measurable) process parameters, e.g. flow rate, mould temperature and part insert temperature as the geometrical information is embedded within the spatial modes $\phi_i(x)$. Third, due to the removed geometrical data, only one data point per process parameter variation is needed, which significantly decreases the data volume. In total, the training data set (if a share of 20 % is hold out for testing) is sized 80 x 3 with 50 target variables, in contrast to 655,680 x 6 and a single target variable for FEA. Within pre-processing, the input features are standardized by MinMax scaling. As for FEA in the previous study, a benchmark across six data-driven approaches is performed, which are AdaBoost, Decision Tree, Gradient Boosting (GradBoost), Polynomial Regression, RandomForest and Extreme Gradient Boosting (XGBoost). The training is implemented by the scikit-learn library (Pedregosa et al., 2011) of python using a 5-fold cross-validation. As shown in Table 1 a sound hyperparameter optimization is performed. The POD and FEA columns list the hyperparameters that result in best scores for R^2 for both approaches. The comparison shows that the hyperparameters differ between the two approaches, which is reasonable, because of the significantly changed properties of the underlying data sets of the same thermoplastic composite structure.

Table 1: *Overview of applied hyperparameter optimization of six data-driven methods for the POD and FEA surrogate modelling approaches*

Method	Hyperparameters	Range	# Steps	POD	FEA
AdaBoost	learning rate	0.1-1	10	0.9	0.7
	# estimators	10-100	7	90	30
Decision Tree	max depth	10-None	7	200	None
	min samples leaf	1-10	10	3	6
GradBoost	learning rate	0.1-1	10	0.1	0.7
	# estimators	10-500	7	300	500
Polynomial Reg.	degree	1-7	7	3	4
RandomForest	max depth	10-70	4	30	50
	min samples leaf	1-10	5	1	1
	# estimators	10-500	9	300	100
XGBoost	learning rate	0.1-1	10	0.1	0.7
	# estimators	10-500	7	500	500

3.3 Preparing the Surrogate to run on the edge

A challenge when developing software solutions to run on the edge is that the development is usually different from the deployment environment. In this case, the development of the simulation and the surrogate model was on personal computers (workstation) and the result needs to be deployed on a separate edge-device. One technology to transfer software from one environment to the other without compatibility issues is Docker containers. Containers are packaged software environments made up of different layers. The bottom layer is the Host-OS usually a Linux distribution with additional software, libraries and configurations on top. The main benefit of containers is that they can be seamlessly shared between different host systems independent from the local environment. For this study the base image with the Host-OS is python3.6:slim-buster from docker-hub (public, online container repository). The image contains Linux Debian 10.9 with Python 3.6 already installed. In the next step, the surrogate application comprised of the Python code and trained machine learning models, is added to the container. Additional libraries (e.g., XGBoost) are installed. In order to access the surrogate model inside the container an application programmable interface (API) is employed. The API takes simulation parameters and the machine learning method as an input. With the input the API executes the surrogate model inside the container and provides the output in a structured format (JSON). Request and response are transferred using the http protocol, which is very widespread and makes the API integrable in nearly any other software.

4 Results

Machine Learning Results

Regarding the prediction of each single POD coefficient, it can be seen that significantly weak predictions can occur by Polynomial Regression. Further, there are no clear performance differences or trends across the 50 POD coefficients for all six models. Therefore, in Figure 3, a comparison of the R^2 scores for the two approaches POD (x-axis) and FEA (y-axis) across the six data-driven methods is depicted. The depiction shows that in contrast to the FEA approach all six models are feasible for the POD surrogate modelling task as DecisionTree performs weakest for POD with $R^2 = 0.893$ and a mean of all models of $R^2 = 0.936$ (mean FEA: 0.646). For both approaches RandomForest ($R^2_{POD} = 0.969$; $R^2_{FEA} = 0.925$) outperforms all other models, although GradBoost ($R^2_{POD} = 0.968$) and XGBoost ($R^2_{POD} = 0.966$) are comparable, especially for POD-based surrogate modelling.

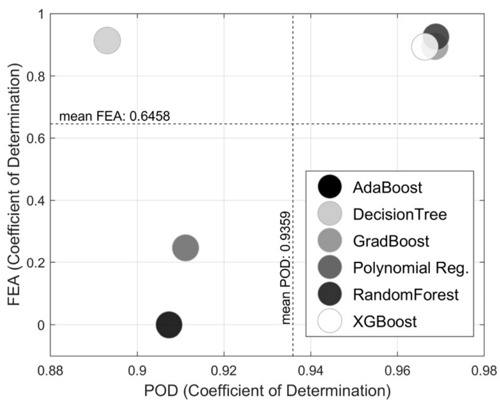

Figure 3: *Comparison of R2 between the approaches FEA and POD surrogate modelling (right) for all six modelling approaches.*

Approximation Quality

From the estimated POD coefficients, the solution given in Eq. (1) can be computed. Since it represents an approximation of the full-scale solution, the corresponding error versus the number of modes k used in the reduced representation is displayed in Figure 4.

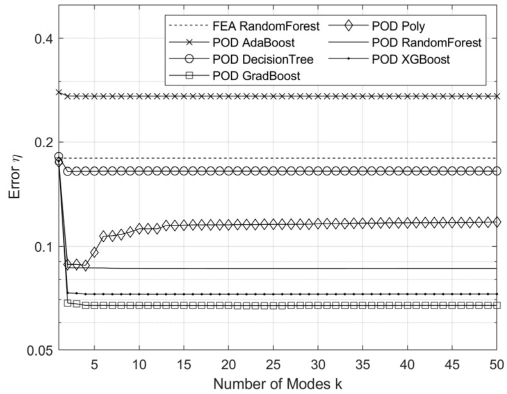

Figure 4: *Comparison of Model error with respect to full-scale simulations.*

As reference, the error achieved by Random Forest used for FEA is displayed. Note, that the plotted errors are the average of 100 results. It can be seen, that POD performs in general better (except AdaBoost) than the direct FEA approach. DecisionTree shows the lowest coefficient of determination. However, the error is even slightly smaller than for the best FEA approach. In the present study, GradBoost and XGBoost achieve the smallest error. Even though Random Forest performed best in approximating the POD coefficients (cf., Figure 3), these two methods lead to a smaller error compared to the full-scale simulation. In this context, it should be mentioned that with increasing number of modes, the influence of the individual modes decreases. Hence, the first modes are most important for the POD approximation. When the approximation of the first mode is already insufficient, a comparable large model error can be expected. From the graph, we observe that each method converges after five modes (except Polynomial Regression which in fact shows an error increase when using more than five modes), which implies that the number of target variables can be significantly decreased and only the first modes are important to predict.

Testing the container on an Edge Device and the Edge Server

In order to verify the edge suitability of the container with the surrogate model it is executed on the Edge Server (Workstation with 3.9 GHz Xeon W2245, 128 GB RAM) and the Edge Device (1.9 GHz i7-3517U, 4 GB RAM). Both are running on a Windows 10 OS and have a x84-64 CPU architecture. The Server can handle the FEA and the POD approach, but the device is not able to compute the FEA surrogate model because it is running out of RAM. Figure 5 depicts the various calculation times depending on the machine learning method for POD surrogates. Every method was tested with 100 different parameter configurations. The bar shows the mean calculation time and the error bar shows the standard deviation, which is significantly larger for the Edge Device. As expected, the Edge Server vastly outpaces the Edge Device. Most methods take below 1 second to compute in both environments with the exception of GradBoost an XGBoost.

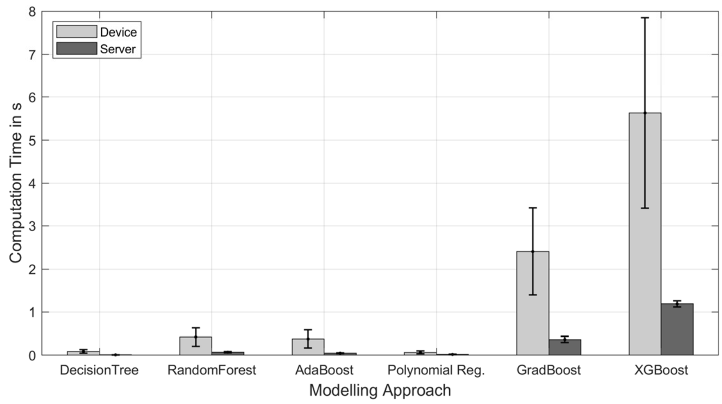

Figure 5: *Calculation times for the POD surrogate models on the edge server and on the edge device.*

5 Concluding Remarks

The combination of sensor data and real time-feasible surrogate models obtained from engineering data in flexible production systems shows great potential. Since continuous data streams need to be evaluated, low-data demanding and fast algorithms are necessary to define virtual quality gates. Utilizing POD from offline obtained simulation results extracts the main features of the underlying physics and forms the basis for a reduced representation. The advantage for this kind of surrogate modelling lies in the recued number of data that need to be processed during training and evaluation which makes it suitable for edge computing. Furthermore, it has been shown in the given use case of a thermoplastic composites that the surrogate error even decreases when using a POD approach instead of a pure data driven modelling of the FEA data set. The smallest error in this cases is obtained with GradBoost. However comparing the overall metrics and the calculation time on the edge server, RandomForest is most suitable. In future works, the investigated process chain will be extended and the edge computing concept for flexible manufacturing cells will be implemented and tested within the IML.

References

Allison, J.; Backman, D.; Christodoulou, L.: Integrated computational materials engineering: A new paradigm for the global materials profession. JOM 58 (2006) 11, pp. 25–27.

Chinesta, F.; Cueto, E.; Abisset-Chavanne, E.; Duval, J.L.; Khaldi, F.E.: Virtual, Digital and Hybrid Twins: A New Paradigm in Data-Based Engineering and Engineered Data. Archives of Computational Methods in Engineering (2018).

Dröder, K.; Reichler, A.-K.; Mahlfeld, G.; Droß, M.; Gerbers, R.: Scalable Process Chain for Flexible Production of Metal-Plastic Lightweight Structures. Procedia CIRP 85 (2019), pp. 195–200.

Filz, M.-A.; Gellrich, S.; Turetskyy, A.; Wessel, J.; Herrmann, C.; Thiede, S.: Virtual Quality Gates in Manufacturing Systems: Framework, Implementation and Potential 2020.

Han, Z.-H.; Zhang, K.-S.: Surrogate-Based Optimization. In: Roeva, O. (Hrsg.): Real-World Applications of Genetic Algorithms: InTech 2012,

Hürkamp, A.; Dér, A.; Gellrich, S.; Ossowski, T.; Lorenz, R.; Behrens, B.-A.; Herrmann, C.; Dröder, K.; Thiede, S.: Integrated computational product and production engineering for multi-material lightweight structures. The International Journal of Advanced Manufacturing Technology 110 (2020a) 9-10, pp. 2551–2571.

Hürkamp, A.; Gellrich, S.; Dér, A.; Herrmann, C.; Dröder, K.; Thiede, S.: Machine Learning and Simulation Based Surrogate Modelingfor Improved Process Chain Operation. The International Journal of Advanced Manufacturing Technology (2021).

Hürkamp, A.; Gellrich, S.; Ossowski, T.; Beuscher, J.; Thiede, S.; Herrmann, C.; Dröder, K.: Combining Simulation and Machine Learning as Digital Twin for the Manufacturing of Overmolded Thermoplastic Composites. Journal of Manufacturing and Materials Processing 4 (2020b) 3, pp. 92.

Liang, L.; Liu, M.; Martin, C.; Sun, W.: A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis. Journal of the Royal Society, Interface 15 (2018) 138.

Panchal, J.H.; Kalidindi, S.R.; McDowell, D.L.: Key computational modeling issues in Integrated Computational Materials Engineering. Computer-Aided Design 45 (2013) 1, pp. 4–25.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; others: Scikit-learn: Machine learning in Python. the Journal of machine Learning research 12 (2011), pp. 2825–2830.

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 72 (2018), pp. 426–431.

Reichler, A.-K.; Gerbers, R.; Falkenberg, P.; Türk, E.; Dietrich, F.; Vietor, T.; Dröder, K.: Incremental Manufacturing: Model-based part design and process planning for Hybrid Manufacturing of multi-material parts. Procedia CIRP 79 (2019), pp. 107–112.

Sancarlos, A.; Cameron, M.; Abel, A.; Cueto, E.; Duval, J.-L.; Chinesta, F.: From ROM of Electrochemistry to AI-Based Battery Digital and Hybrid Twin. Archives of Computational Methods in Engineering (2020).

Swischuk, R.; Mainini, L.; Peherstorfer, B.; Willcox, K.: Projection-based model reduction: Formulations for physics-based machine learning. Computers & Fluids 179 (2019), pp. 704–717.

Thiede, S.; Turetskyy, A.; Kwade, A.; Kara, S.; Herrmann, C.: Data mining in battery production chains towards multi-criterial quality prediction. CIRP Annals 68 (2019) 1, pp. 463–466.

Wang, P.; Gao, R.X.; Fan, Z.: Cloud Computing for Cloud Manufacturing: Benefits and Limitations. Journal of Manufacturing Science and Engineering 137 (2015) 4.

Wu, D.; Liu, S.; Zhang, L.; Terpenny, J.; Gao, R.X.; Kurfess, T.; Guzzo, J.A.: A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing. Journal of Manufacturing Systems 43 (2017), pp. 25–34.

Wuest, T.; Weimer, D.; Irgens, C.; Thoben, K.-D.: Machine learning in manufacturing: advantages, challenges, and applications. Production & Manufacturing Research 4 (2016) 1, pp. 23–45.

Zambal, S.; Eitzinger, C.; Clarke, M.; Klintworth, J.; Mechin, P.: A digital twin for composite parts manufacturing: Effects of defects analysis based on manufacturing data. In: 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), 2018, pp. 803–808.

Zimmerling, C.; Poppe, C.; Kärger, L.: Estimating Optimum Process Parameters in Textile Draping of Variable Part Geometries - A Reinforcement Learning Approach. Procedia Manufacturing 47 (2020), pp. 847–854.