Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 121 (2024) 49–54

www.elsevier.com/locate/procedia

11th CIRP Global Web Conference (CIRPe 2023)

Framework for automatic production simulation tuning with machine learning

Marvin Carl May[a,*], Alexander Finke[a], Katharina Theuner[a], Gisela Lanza[a,b]

[a]wbk Institute of Production Science, Karlsruhe Institute of Technology (KIT), Kaiserstr. 12, 76131 Karlsruhe, Germany
[b]Global Advanced Manufacturing Institute (GAMI), KIT China Branch, Suzhou 215123, China

* Corresponding author. Tel.: +49 1523-950 2624; fax: +49 721 608-45005. *E-mail address: marvin.may@kit.edu* marvin.may@kit.edu

Abstract

Production system simulation is a powerful tool for optimizing the use of resources on both the planning and control level. However, creating and tuning such models manually is a tedious and error-prone task. Despite some approaches to automate this process, the state-of-the-art relies on the generation of models, by incorporating the knowledge of experts. Nevertheless, effectively creating and tuning such production simulations is, thus, a key driver for reducing costs, carbon footprint, and tardiness and therefore an essential factor in today´s production. Beneficial would be automated and flexible frameworks, since these are applicable to different use cases requiring less effort. Yet, in the age of Industry 4.0, data is ubiquitous and easily available and can serve as a basis for virtual models representing reality. Increasingly, these virtual models shall be interlinked with the current state of real-world systems to form so-called digital twins. As automated and flexible frameworks are missing, this paper proposes a novel approach where observed real system behavior is used and fed into a large-scale machine learning model trained on a plethora of possible parameter sets. The main target is to train this machine learning model to minimize the reality gap between the behavior of the simulated and real system by selecting corresponding simulation system parameters. By estimating those parameters an enhancement of the simulation will emerge. An interlink to real systems can be derived resulting in a digital shadow which is capable to forecast the future similarly to reality. The approach to overcoming the gap between reality and simulation (real2sim) is validated in simulations.

Keywords: Production Simulation; Machine Learning; Simulation Tuning; Simulation Generation

1. Introduction

Worldwide, the pressure for flexible yet efficient production systems is growing rapidly. To optimize those, multidimensional optimization issues have to be tackled and complex dependencies need to be designed [8]. Due to the unchanged mental capacity of human beings, supportive techniques for the planning of these production systems have to be applied.

Increased attention is paid to simulations, serving as tools for decision-making in production control [2]. Panzer et al. [19] provide a review of research regarding neural networks in production planning and control whereby a large number of research was implemented in simulations. As the application of simulation programs, is referblack to as a common tool to improve the efficiency and control of manufacturing systems, they help to fine-tune the production systems and, thus, to foresee the near term behavior of the regarded production system [15]. By applying carefully selected suitable concepts on their application, this can drastically reduce material and resource usage [18]. However, in order to derive meaningful information, these simulations themselves have to be configured in such a way that they closely mimic real behavior.

Due to the high complexity, this modeling process is time consuming and expensive [17]. Nevertheless, current improvements are pointing towards a simulation that follows closely the real environment on the selected level of granularity - a so-called digital twin. [16].

Due to the high amount of data acquired in production systems the originally manual creation process of such simulation environments is prone to errors or oversimplification [17]. In wake of recent breakthroughs in the field of machine learning, this data based task could be assisted or even automated by a concept based on machine learning [4]. Therefore, this paper proposes a framework for simple, automated simulation model creation and parameter tuning based on observable big data and its integration into a machine learning approach.

10.1016/j.procir.2023.11.002

To do so, this paper is structured in seven sections, beginning with the introduction. The state-of-the-art is examined in Section 2, which derives the main research question, based on the research gap. The following Section 3 explains the machine learning based automated simulation tuning framework itself as well as the corresponding building, refining and testing. Section 4 introduces the case study. In the subsequent Section 5 the results of the application of this framework with the introduced testing-approach are shown and analyzed. These results and the corresponding analysis are then discussed in Section 6. Relevant insights as a summary and various subsequent research questions are then pointed out in Section 7.

2. State-of-the-art

As discussed in Section 1, simulations are used to investigate production systems. They are typically based on discrete event simulation systems and can be modeled with various software available. This process is normally executed by an expert of simulation creation. To tackle the problem of complexity, one normally adheres to guidelines, such as VDI-3633 [25]. These guidelines support the creator with various methods and granular steps, but the process itself remains time consuming and, therefore, expensive. Especially the time needed for data collection and preparation grows proportionally with the complexity of the production system. For instance, the VDI-3633 [25] shows, that information of different domains has to be aggregated such as plant-layout, technical machinery data, processing sequences, operations planning or production planning. Therefore, the person in charge needs to interview experts, consult machine suppliers, read documentations, observe the current production processes and access various databases.

To reduce the time needed for simulation-creation, different toolboxes and frameworks are proposed throughout the scientific community. In the 1990s so called expert systems were developed to aid in the process of simulation creation or production control as mentioned in Egresitis et al. [5], Ikkai et al. [7] or Westkämper et al. [26]. In toady's complex world, production control approaches extend towards the idea of a digital twin [15]. These in turn are based on complex real-time simulations. The old approaches are not flexible enough to be able to react to rapidly changing conditions. Therefore, improved approaches, ideally based on the plethora of available data and typically enhanced through machine learning have to be developed. The system model can be enhanced by a permanent and flexible responding estimation of parameters. Consequently, the application of even more powerful machine learning would be achieved and thus a reduction of human work and error.

Reaching a high performance while keeping computing power low as well as considering changing circumstances are *challenges in ML*. To increase the ability to handle complex systems, automatic solutions are desirable [4]. In addition, modeling accuracy has to achieve the desired objective [25]. In order to overcome these challenges, various requirements are specified for one approach. In addition, this paper introduces so-called framework maturity, leading to five different requirements. The focus remains on the ability to perform detailed analysis with reduced manual interaction. Meanwhile, the required computing power ought to be low in order to be technically feasible. Hereafter various more recent approaches are discussed, regarding those requirements.

For instance, von Rueden et al. [22], propose various approaches to enhance simulations with the help of machine learning. First off, one can use machine learning to modify the model itself and delete irrelevant, but overly complex parts of it. The approach referred to as initial model causes an enhanced adjustment of the simulation model, but could lack diversity in further investigations. Second, one can use machine learning to find the optimal parameters to configure the simulation, without altering the model itself. This process, referred to as input parameters saves time while still preserving all features of the simulation. Both could be used to reduce the time consumed in the creation process but still lack detail to make the framework directly feasible.

Kasim et al. [10] combine both ideas into the concept of simulation emulation. In this concept, a neural network is used to train the correlations of input parameters to output parameters and, thus, replace the simulation entirely. With this approach, however, the loss of information leads to disadvantages. The decisions cannot be reproduced despite similar results so the requirements for the detailed analysis are hard to be achieved.

Elbattah and Molloy [6] propose a concept in which only parts of the simulation are replaced by emulations to reduce complexity. In a production context these parts could be agents such as machines or autonomous transportation vehicles as proposed by Bergmann et al. [2]. The main downside of this approach is the lack of actual insights into the decision process and, hence, the missing ability to try out new features. A similar approach uses reinforcement learning to make decisions in the actual production system, which increases efficiency, but does not provide further insights into the system itself [21]. Thus, both approaches [2] and [21] fail to meet the requirements for analysis detail. Another framework proposed by Bergmann and Straßburger [3] is based on the idea of automatic model creation, based on a premodeled description of the simulation system and, therefore, shifts the problem of complexity towards the premodeled description language. However, a major disadvantage is that for this approach the computing power cannot be decreased sufficiently to satisfy the requirement.

The fulfillment levels and disadvantages of the aforementioned approaches are presented in Table 1. It shows that only one of the solutions available (ML for parameter tuning) fulfills the requirements regarding lesser manual interaction while maintaining a high level of analysis detail [22]. But the maturity of this solution is not high enough as successful implementations and case study solutions are missing. Thus, it represents only a starting point for further investigations which are presented within our paper.

Process mining techniques have been increasingly employed in more recent approaches regarding model generation [1]. Lugaresi and Matta [11] generate simulation-based digital twins by information derivation from data logs of production systems. Pourbafrani et al. [20] propose an approach based on event data

Table 1. Comparison of different approaches

Approach	Low Computing Power	Less Manual Interaction	Framework Maturity	Applicability on different scenarios	Analysis Detail
Standard procedure (e.g. [25])	◐	○	●	●	●
ML for model adjustment [22]	○	◐	○	●	○
ML for parameter tuning [22]	○	◐	○	●	●
simulation emulation [10]	●	○	◐	◐	○
partial simulation emulation [2]	◐	◐	◐	◐	○
RL for decision making[21]	○	◐	◐	◐	◐
automatic model creation [3]	◐	○	◐	●	●

Legend: ●fulfilled, ◐partial fulfilled, ○not fulfilled or regarded

to generate system dynamics simulation models. However, the goal of this work is to develop a model that can react flexibly and quickly. Obtaining parameters from process mining (e.g. processing times) results in waiting for a complete trace to be run before effects can be captured and reacted to. Therefore, this paper distinguishes itself from the newer approaches by not creating the infrastructure based on event logs and resulting process models but still manually by an expert in order to ensure the necessary flexibility. This paper addresses the existing gap between industrial requirements and the actual state of the art in research and proposes a framework that addresses the aforementioned requirements.

The basis of the framework is built upon the idea of finding the optimal input parameters for the simulation through machine learning, in a similar vein to one of the approaches mentioned in von Rueden et al. [22]. Thus, analyses of the production system are still possible, while maintaining the advantages of an easier simulation creation.

3. Methods

To accomplish the aim of a mostly automatic simulation creation through machine learning, the resulting degrees of freedom for the algorithm have to be designed. This process is based on the purpose of building an ideal simulation adjusted to reality and, thus, a digital twin of the real production system. Other systems should not be taken into account, to reduce the degrees of freedom for the algorithm. As aforementioned, the expert has to model the structure of the system manually. This structure consists of every determined part in the system, such as the number of machines and transportation systems or the required process steps for certain products. Such information is usually accessible and known. Due to the determined character of these parameters, the complexity is still manageable for the simulation expert.

Based on this structure, ML is used to train the algorithm in order to find the behavioral parameters in the system that generate KPIs that hardly differ from the real KPIs. The needed data can be collected from past production log files. The estimated behavior parameters are fed into the simulation. Comparing the thereby calculated KPIs to real KPIs an error can be derived that serves for the adjustment of the ML algorithm.

The main challenge of this approach is backpropagation. To achieve a desired prediction, with backpropagation, the error needs to be mapped to specific weights of the neural network. Based on the simulation with stochastic elements in the backpropagation cycle, this cannot be realized. Therefore, this paper proposes three different approaches to solve this problem. The KPIs addressed in each approach refer to supported KPIs of the simulation based on Kang et al. [9], Stricker et al. [23] and May et al. [12].

One implementation approach consists of building a second neural network, which emulates the simulation. Therefore, the backpropagation can be calculated by being able to derive the error function over the neural network emulating the simulation. Hence, the problematic stochastic elements in the cycle are cut.

Another possible approach makes use of reinforcement learning. In this solution, the error does not need to be derived, but rather a simple feedback has to be implemented. This number is also known as reward and measures the quality of the given output of the neural network.

The third implementation approach is based on generalization. In this approach, the neural network is trained on huge amounts of presimulated data and, therefore, known correlations between behavioral parameters and resulting KPIs. As a result, the neural network learns these correlations. When being fed with the real KPIS it uses the correlations to find the corresponding behavioral parameters. Thus, it generalizes from known correlations. Due to the implementation simplicity, this paper uses a generalization technique and, therefore, a supervised learning approach to test and evaluate the proposed framework. Due to the framework style of this paper, the following insights can still be easily transferred.

The main framework is shown in Figure 1. The ANN estimates the parameters which are fed into the simulation. The simulation calculates the predicted KPIs which are then compared to the real KPIs. The error which occurs here serves for the adjustment of the ML algorithm.

Evaluating this approach according to the aforementioned requirements introduced in Section 2 the following results can be derived. While the requirements according to the applicability on different scenarios and analysis detail can be reached, framework maturity and computing power are regarded on a medium level. Almost none manual interaction is needed.

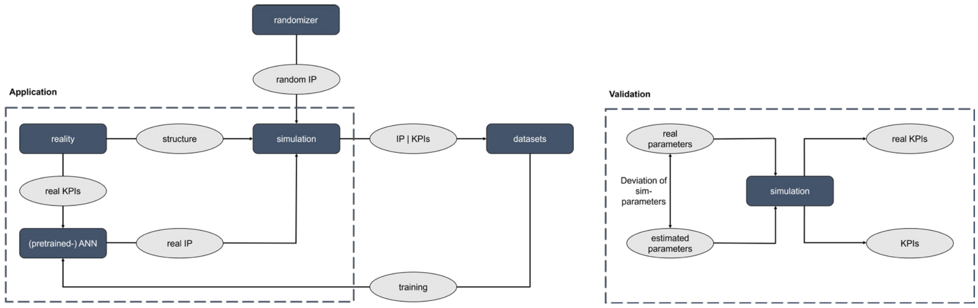

Fig. 1. Explanation of the proposed framework with the focus within the dashed lines - IP is the abbreviation for Input Parameters

Table 2. Overview of used production systems and measured parameters on different structures

Production system			Parameters on different structures	
ID	Machines	Products	ID	Varied parameters
1xx	1	2	xx0	all possible
2xx	4	2	xx1	order amount, process time
3xx	4	8	xx2	+ process variance
4xx	8	8	xx3	+ setup time
5xx	8	16	xx4	+ handling times

4. Case study

This paper makes use of different production types, function as a categorization of a variety of production structures as well as parameter combinations and is formed by three digits. Whereas the first digit indicates the underlying production structure, the latter two refer to the parameter configuration. The different production structures are shown in Table 2. They reach from simple to complex systems and build a reasonable base to discuss the quality of the framework. As aforementioned, this paper applies a wide range of KPIs based on the KPI system proposed by Kang et al. [9] and Stricker et al. [23]. The used input parameters for the simulation system consist of the structural and behavioral parameters introduced in Table 2. These are designed to be able to map the corresponding KPIs. An example could be implementing the failure probability for a machine, if there is a KPI describing the machine downtime. The number of used parameters is also varied over several different systems. As previously illustrated, the third digit represents the variation of the parameters of the production type. They can be gradually increased in complexity analogously to the production types as shown in Table 2. Based on the five different production structures together with parameter configurations, twenty-five possible production types result.

The method described is used within a situation aware, knowledge graph based DES, OntologySim [13]. It is an event discrete simulation, generating KPIs by appropriate parameterization and can later serve as a digital twin. It is capable of logging aggregated KPIs, agent based KPIs and time step based KPIs [13].

For the evaluation, a reasonable range for the behavioral parameters for every production system was set. This step tremendously reduces the number of possible combinations and makes use of the already existing knowledge about the specific system [18]. Based on this range, huge amounts of randomized behavioral parameters were created and fed into the simulation to obtain the corresponding KPIs, a standard approach in using machine learning techniques for production planning and control optimization [24]. The resulting KPIs and the corresponding input parameters are saved as sample. Based on these, the neural network is trained. Previously, one of these was separated for evaluation to represent the real production system being simulated. The entered number of configurations, an expert estimation, and the real data set are generated for each of the 25 production types. To generate data a process simulation program is used. During training, the early-stopping approach is chosen to avoid overfitting, as well as to optimize the required time. After the training, the process is evaluated by feeding the representing real input parameters into the simulation and comparing the resulting KPIs with the representing real KPIs. While doing so, optimal hyperparameters are searched systematically. The aim of this search is to find the best parameters for a high accuracy as well as efficiency.

5. Results

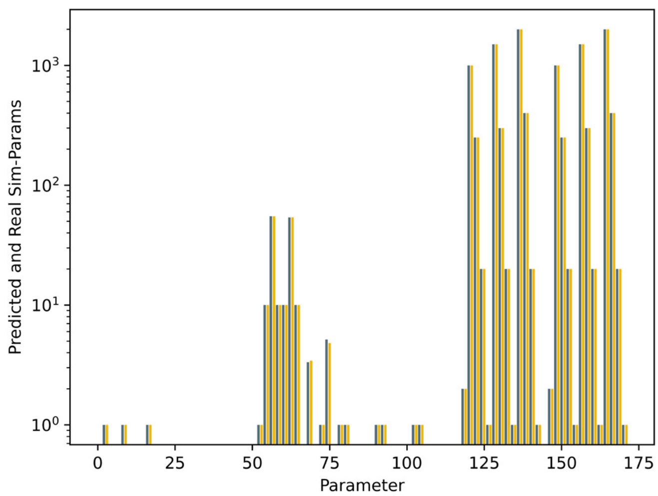

Fig. 2. Detailed comparison for parameters of system 101 (blue) with the proposed ML model prediction (yellow)

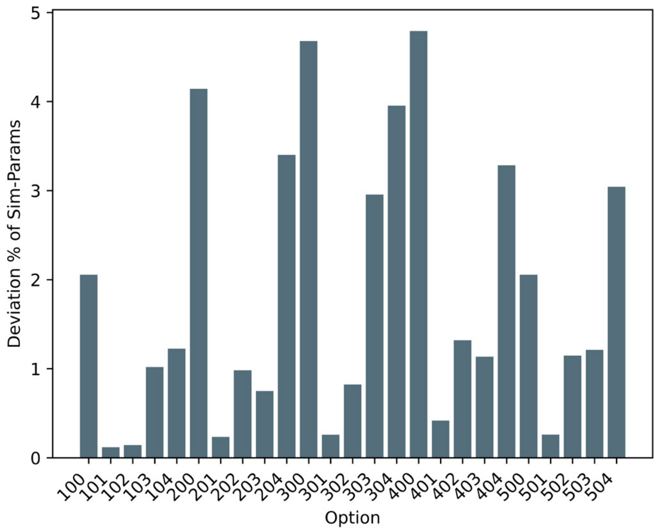

Fig. 3. Parameter prediction performance comparison of all different systems

By searching systematically for the best hyperparameters with the lowest generalization error, the parameters of Table 3 were the most accurate and efficient. Therefore, the following results were recorded using these hyperparameters.

Hyperparameters	Value
samples	500
epochs	10.000.000
batch-size	≥samples
learning rate	0.1
momentum	0.1
hidden layers	1
loss function	MSE
activation function	None
optimizer	adamax
Nesterov	True
dropout	False
patience	1000

Table 3. Hyperparameters used in this evaluation

With this configuration, Figure 2 shows the predicted parameters in relation to the real parameters of an example "system 101". Deep diving into the various investigated systems shows excellent results. Even parameters in a wide range can be fitted as indicated by the logarithmic scale. Especially on systems with a low to medium complexity, such as 101, the behavioral parameters match the real ones almost perfectly, despite the fact that the observed parameters are not present in the training data set.

Nonetheless, the accuracy of the neural network drops with increasing complexity when analyzing the overall performance as shown in Figure 3. For example, the accuracy decreases when observing gradually more complex production structures (100, 200, 300, 400, 500). Simpler derivatives of these base systems, such as 501, clearly show a better performance than their root system with more variables to handle.

6. Discussion

To interpret the quality of the proposed framework, the given requirements are compared with the actual results of the exemplary implementation. As stated in Section 5, the deep dive into single systems shows excellent results, especially in low complexity systems. Therefore, the resulting simulation is optimally adapted to reality and can be used for complex scenarios, such as production planning based on realtime simulation.

Problematically is the drop of performance on systems of higher complexity. This is due to the fact that the training took place on a single machine with limited resources and could not be tested on huge amounts of data. The data has to be created artificially and can not be extracted out of already existing logs from physical systems. In industrial scenarios this data is already available from the past and more resources for computing can be allocated. However, in contrast to simulative data that is used to obtain the presented results, real world systems can include a higher degree of abnormalities. Furthermore, the exemplary implementation itself is not yet perfectly optimized. According to this, the bottleneck of highly complex systems in the exemplary implementation can be neglected when analyzing the overall framework.

The tendency is that more data can improve the accuracy to some extent, as shown by "systems 101"'s deepdive. This accuracy is the unique selling point of the framework, along with the reduced labor required to create a simulation. When the algorithm is implemented and the structure remains the same, the process of simulation creation can be automated completely. Therefore, the framework itself fulfills the requirements as outlined in the literature review and delivers a mature solution for automatic simulation creation. Nonetheless, the implementation has to be further optimized to move from a solution for now to a high fidelity "four know" machine learning application as introduced by Chen et al. [4].

7. Summary and Outlook

The analysis shows that the proposed framework works well in scenarios with sufficient available data and that a good implementation of the machine learning algorithm is feasible. Sufficient data is a term depending on the complexity of the present production system.

All in all, the proposed framework fulfills its claims and can automatically tune a simulation to match reality in order to facilitate digital twins that are fine-tuned based on authentic parameters rather than comparably output. This can be used in various scenarios, such as simulation based production control systems, digital twins or future analysis. The framework still uses a simulation as a core element and enables to analyze the decision-making in the system while still using all the advantages a machine learning algorithm provides. Therefore, it solves the problem of time consuming simulation creation of complex systems and offers a high maturity to be directly applicable. In addition, the complete manual effort can be saved.

In future directions, the approach should be adapted in order to be applied to more complex and flexible structures such as job shops. As the process starts with simulating a variety of system descriptions, the ANN is locked to the regarded system, as it was designed. The implemented production systems and parameters require expansion by adding the routes and the more complex characteristics of the job shops. However, this requires just a couple of parameters to be included in the training, while the basic framework remains the same. In addition, this framework can be adapted or more efficient implementations can be tested by making use of novel machine learning algorithms, cleverer and knowledge infused state-space modeling [4] or the integration of natural-language processing capabilities in the front end or alike transformer machine learning models [14]. It can also be compared with frameworks based on the other proposed solutions for the problem of backpropagation.

Simulation will remain challenging in the future since this approach is dependent on data quality and availability. The method still needs improvements and will benefit from experts. However, the need to involve experts has already been reduced to a certain degree.

Acknowledgment

This research work was undertaken in the context of DIGIMAN4.0 project. DIGIMAN4.0 is a European Training Network supported by Horizon 2020, the EU Framework Programme for Research and Innovation (Project ID: 814225).

References

[1] van der Aalst, W.M., 2018. Process mining and simulation: A match made in heaven!, in: SummerSim, pp. 4–1. doi:10.22360/summersim.2018.scsc.005.

[2] Bergmann, S., Stelzer, S., Strassburger, S., 2014. On the use of artificial neural networks in simulation-based manufacturing control. Journal of Simulation 8, 76–90. doi:10.1057/jos.2013.6.

[3] Bergmann, S., Straßburger, S., 2020. Automatische Modellgenerierung – Stand, Klassifizierung und ein Anwendungsbeispiel, in: Mayer, G., Pöge, C., Spieckermann, S., Wenzel, S. (Eds.), Ablaufsimulation in der Automobilindustrie. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 333–347. doi:10.1007/978-3-662-59388-2_23.

[4] Chen, T., Sampath, V., May, M.C., Shan, S., Jorg, O.J., Aguilar Martín, J.J., Stamer, F., Fantoni, G., Tosello, G., Calaon, M., 2023. Machine learning in manufacturing towards industry 4.0: From 'for now'to 'four-know'. Applied Sciences 13, 1903. doi:10.3390/app13031903.

[5] Egresitis, C., Monostori, L., Hornyák, J., 1998. Multistrategy learning approaches to generate and tune fuzzy control structures and their application in manufacturing. Journal of Intelligent Manufacturing 9, 323–329. doi:10.1023/A:1008922709029.

[6] Elbattah, M., Molloy, O., 2018. ML-Aided Simulation, in: Quaglia, F. (Ed.), Proceedings of the 2018 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, ACM, [Place of publication not identified]. pp. 33–36. doi:10.1145/3200921.3200933.

[7] Ikkai, Y., Ohkawa, T., Komoda, N., 1996. Recursive learning method for knowledge-based planning system. Journal of Intelligent Manufacturing 7, 405–410. doi:10.1007/BF00123918.

[8] Kandler, M., Dierolf, L., Bender, M., Schäfer, L., May, M.C., Lanza, G., 2022. Shopfloor management acceptance in global manufacturing. Procedia CIRP 115, 190–195. doi:10.1016/j.procir.2022.10.072.

[9] Kang, N., Zhao, C., Li, J., Horst, J.A., 2016. A Hierarchical structure of key performance indicators for operation management and continuous improvement in production systems. International Journal of Production Research 54, 6333–6350. doi:10.1080/00207543.2015.1136082.

[10] Kasim, M.F., Watson-Parris, D., Deaconu, L., Oliver, S., Hatfield, P., Froula, D.H., Gregori, G., Jarvis, M., Khatiwala, S., Korenaga, J., Topp-Mugglestone, J., Viezzer, E., Vinko, S.M., 2020. Building high accuracy emulators for scientific simulations with deep neural architecture search. URL: http://arxiv.org/pdf/2001.08055v1.

[11] Lugaresi, G., Matta, A., 2021. Automated manufacturing system discovery and digital twin generation. Journal of Manufacturing Systems 59, 51–66. doi:10.1016/j.jmsy.2021.01.005.

[12] May, M.C., Fang, Z., Eitel, M.B., Stricker, N., Ghoshdastidar, D., Lanza, G., 2023. Graph-based prediction of missing kpis through optimization and random forests for kpi systems. Production Engineering 17, 211–222. doi:10.1007/s11740-022-01179-y.

[13] May, M.C., Kiefer, L., Kuhnle, A., Lanza, G., 2022a. Ontology-based production simulation with ontologysim. Applied Sciences 12, 1608. doi:10.3390/app12031608.

[14] May, M.C., Neidhöfer, J., Körner, T., Schäfer, L., Lanza, G., 2022b. Applying natural language processing in manufacturing. Procedia CIRP 115, 184–189. doi:10.1016/j.procir.2022.10.071.

[15] May, M.C., Overbeck, L., Wurster, M., Kuhnle, A., Lanza, G., 2021. Foresighted digital twin for situational agent selection in production control. Procedia CIRP 99, 27–32. doi:10.1016/j.procir.2021.03.005.

[16] Negri, E., Fumagalli, L., Macchi, M., 2017. A review of the roles of digital twin in cps-based production systems. Procedia Manufacturing 11, 939–948. doi:10.1016/j.promfg.2017.07.198.

[17] Overbeck, L., Brützel, O., Teufel, M., Stricker, N., Kuhnle, A., Lanza, G., 2021. Continuous adaption through real data analysis turn simulation models into digital twins. Procedia CIRP 104, 98–103. doi:10.1016/j.procir.2021.11.017.

[18] Overbeck, L., Rose, A., May, M., Lanza, G., 2022. Nutzungskonzept für digitale zwillinge von produktionssystemen. Zeitschrift für wirtschaftlichen Fabrikbetrieb 117, 244–248. doi:10.1515/zwf-2022-1035.

[19] Panzer, M., Bender, B., Gronau, N., 2022. Neural agent-based production planning and control: An architectural review. Journal of Manufacturing Systems 65, 743–766. doi:10.1016/j.jmsy.2022.10.019.

[20] Pourbafrani, M., van Zelst, S.J., van der Aalst, W.M., 2020. Supporting automatic system dynamics model generation for simulation in the context of process mining, in: Business Information Systems: 23rd International Conference, BIS 2020, Colorado Springs, CO, USA, June 8–10, 2020, Proceedings 23, Springer. pp. 249–263. doi:10.1007/978-3-030-53337-3_19.

[21] Rinciog, A., Meyer, A., 2021. Fabricatio-rl: a reinforcement learning simulation framework for production scheduling, in: 2021 Winter Simulation Conference (WSC), IEEE. pp. 1–12. doi:10.1109/WSC52266.2021.9715366.

[22] von Rueden, L., Mayer, S., Sifa, R., Bauckhage, C., Garcke, J., 2020. Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions, in: Berthold, Birukou (Eds.), Advances in Intelligent Data Analysis XVIII, Springer International Publishing, [S.l.]. pp. 548–560. doi:10.1007/978-3-030-44584-3_43.

[23] Stricker, N., Echsler, F., Lanza, G., 2017. Selecting key performance indicators for production with a linear programming approach. International Journal of Production Research 55, 5537–5549. doi:10.1080/00207543.2017.1287444.

[24] Valet, A., Altenmüller, T., Waschneck, B., May, M.C., Kuhnle, A., Lanza, G., 2022. Opportunistic maintenance scheduling with deep reinforcement learning. Journal of Manufacturing Systems 64, 518–534. doi:10.1016/j.jmsy.2022.07.016.

[25] VDI-3633, 2014. Simulation von Logistik-, Materialfluss und Produktionssystemen: Grundlagen - Blatt 1.

[26] Westkämper, E., Pirron, J., Schmidt, T., 1998. Simulation based on learning methods. Journal of Intelligent Manufacturing 9, 331–338. doi:10.1023/A:1008926825868.