A Discrete Event Simulation Based Approach for Digital Twin Implementation

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 54-1 (2021) 414–419

A Discrete Event Simulation Based Approach for Digital Twin Implementation

Lucrezia Morabito * **Massimo Ippolito** * **Erica Pastore** ** **Arianna Alfieri** ** **Francesca Montagna** **

* *Comau S.p.A., Via Rivalta 30, Grugliasco (TO), Italy (e-mail: {lucrezia.morabito, massimo.ippolito}@comau.com).*
** *Department of Management and Production Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino, Italy (e-mail: {erica.pastore, arianna.alfieri, francesca.montagna}@polito.it).*

Abstract: Nowadays, manufacturing companies need to improve their production monitoring and prediction to be more flexible and re-configurable. To do so, the digitization of the manufacturing environment is a very critical issue. This paper proposes an approach to develop digital twins in this environment. Digital twins are virtual systems, real time connected with their physical counterpart, which replicate exactly their behaviour. Discrete event simulation models, connected in real-time with their real system counterparts, are developed in this work. Two industrial use cases are analysed, to show the benefits that this promising technology can bring to the manufacturing industry.

Keywords: Real-time Control, Capacity and Performance Evaluation, Discrete-event Simulation.

1. INTRODUCTION

The rise of the Fourth Industrial (I4.0) revolution has opened an incredible amount of opportunities and has been creating new needs in industries. In particular, companies want to achieve the possibility to control, monitor and predict the performance of their equipment in an effective way to be flexible enough to market needs. Big data, predictive maintenance, Internet of Things (IoT) are some of the most promising enabling technologies of I4.0. However, these technologies still do not really close the loop from the physical to the digital world. From this perspective, to link these two worlds, the cyber physical system (CPS) and cyber physical production system (CPPS) are two key concepts. The first is defined as "systems of collaborating computational entities which are in intensive connection with the surrounding physical world and its on-going processes, providing and using, at the same time, data-accessing and data-processing services available on the internet" (Monostori, 2014). The second consist of "autonomous and cooperative elements and sub-systems that are getting into connection with each other in situation dependent ways, on and across all levels of production, from processes through machines up to production and logistics networks"(Monostori, 2014). Digital twins, i.e., the virtual components of a CPS, are virtual systems that replicate real ones and, consequently, usually represent the connection between physical and virtual worlds. For this purpose, they continuously need data from the real systems to describe their status. These data, in manufacturing context, are shop floor data (Tao and Zhang, 2017). Shop floor data enable digital twins to be synchronized with the field. This communication needs to be bi-univocal (from physical to digital and vice versa) (Lu et al., 2020; Modoni et al., 2019). Generally, a simulator is used to build the digital twin (DT), in particular the virtual shop floor (Tao and Zhang, 2017). This must be able to autonomously update itself when and if the real system changes. Moreover, it has to be able to perform real time simulation in proactive and reactive mode, after deviation or disturbance from the foreseen plan (Monostori et al., 2016).

Nowadays, simulators are mostly used in the building phase of physical systems for performance evaluation purposes. Instead, in the production phase, they are used for performance prediction in what-if analysis, by collecting data from the physical machinery for a period of time that enables the identification of a typical behaviour. Consequently, the majority of simulators are not real time connected with the equipment. In this sense, in the manufacturing context, it is possible to currently identify a gap between the initial installation and commissioning of a production line and its potential connection with its virtual counterpart. In fact, supplier companies of automation systems usually provide production lines that have previously required a simulator. This has been used to prove that the cycle time requested by the customer is achievable. Then the physical system has been installed, tested and finally delivered to the plant. In the majority of the cases, the simulator used at the early stage of the project will not be exploited anymore and not always sold to the customer. In fact, the line managers are used to require the development of a new simulation model during the life cycle of the system for scenarios evaluation.

This paper proposes real use cases of manufacturing systems' DT developed in the design phase and still used in

10.1016/j.ifacol.2021.08.164

the control and production phase. An autonomous trigger mechanism is developed to continuously detect anomalous behavior of the production lines during control and production phase. Thus, production lines are real time connected with their virtual counterpart. If an irregularity is observed, the simulators are run and fed with data collected from the field to predict the systems performance. Also, the DTs are useful to autonomously evaluate alternative scenarios to increase performance, based on the actual functioning of real systems.

The remainder of the paper is structured as follows. Section 2 shows the relevant literature and section 3 the contribution of this work. The description of the approach and of the use cases are reported in Section 4 and 5, respectively. Finally, conclusions are drawn in Section 6.

2. LITERATURE REVIEW

The digital twin objectives can be declined in different ways and the related literature does not show the same technological features and functionalities (Cimino et al., 2019). On one hand, the DT is defined not only for a prognostic assessment at a design stage (static perspective), but also for a continuous check and control (dynamic perspective) (Negri et al., 2017). In this context, DTs are used to reflect the current status of the system and to perform real-time monitoring and optimization, decision making, predictive and remote maintenance (Tao et al., 2018). According to this perspective, DT can be defined as a decision support tool when it is eventually linked to a Manufacturing Execution System (MES) (Kunath and Winkler, 2018). Also, DTs can be defined as a support tool on the experimental phase of a system life cycle (Dahmen and Rossmann, 2018; Sujová et al., 2019).

Simulation and simulators play a significant role in the development of a DT (Negri et al., 2017). The simulation model can have different shapes: 3D dynamical Finite Element Method (FEM)-based models or multi physical models, prediction models, discrete event simulation (DES) models or Simulink models (Cimino et al., 2019). In addition, the so called information model could be based essentially on ISO format (Lu et al., 2020). However, focusing on the applications in which a DES model is used as a digital twin, various approaches can be identified. In fact, it can be used in synergy with other models such as an agent-based one. In this case, when an error occurs, the agent-based model uses agents that compete for executing the previously failed, unfinished task, while the DES model forecasts show the diversion from the planned course (Beregi et al., 2018). In other scenarios, DES models can be used as a virtual counterpart to evaluate performances in what-if scenarios, even without a real time connection with the field or, at most, by collecting data only for a limited amount of time to identify typical functioning of the system (Sujová et al., 2019; Karanjkar et al., 2018). Some applications exist to connect in real time the DES model with the field, with the purpose of mirroring the physical equipment functioning (Vachálek et al., 2017; Garrido and Sáez, 2019).

Moreover, DTs are applied in various industries: there are use cases in manufacturing facilities or automotive companies (Karanjkar et al., 2018; Nonaka et al., 2015; Sujová et al., 2019) and others on *"ad hoc"* systems (e.g., production lines installed in laboratory environment) (Beregi et al., 2018; Vachálek et al., 2017; Cimino et al., 2019).

In conclusion, there are few cases presented within the literature that deals with industrial applications of discrete event simulation combined with a continuous real time connection with the field with the aim of forecasting system performances. Moreover, there is a lack of common vision and architecture that has to be followed to implement applications.

3. CONTRIBUTION

This paper addresses the implementation of DTs through simulators connected in real time with their physical counterpart, to be used during the production phase. The simulator is a DES model connected with the physical machinery thanks to an additional software layer. This is essentially composed by an autonomous trigger mechanism able to detect anomalous behavior of the physical system, a data pre-processing that extracts the actual functioning of the equipment in terms of statistical distributions, and an interface to feed and run the simulator without human intervention. Therefore, the DT is able to predict the production performances and give the plant staff an effective decision support tool. Also, sensitivity analysis of the actual behaviour of the line can be automatically performed with the model. That analysis can also autonomously evaluate different scenarios to find the best configuration on the system. Moreover, two use cases are adopted for the design and validation and are represented by two assembly production lines of Comau (an Italian company of industrial automation based in Turin, Italy) and in a Comau customer plant.

4. APPROACH DEFINITION

In order to implement the DT, three modules are developed in terms of architecture: the input data management, the simulation model and the output analysis. Each module contains micro-services, dedicated to single operations.

The input data management module is dedicated to data collection from the field. In particular, data have different types of sources: Manufacturing Execution System, warehouse, real time operational, Work In Progress (WIP), Material Requirements Planning (MRP) and Enterprise Resource Planning (ERP). These data are collected with specific connectors (e.g., database connectors, real time protocols for industrial communications, sensors and so on), and then pre-processed to extract as much knowledge as possible. In this context, we are interested in: 1) the cycle time for each operation in the production line (the real and the target); 2) the list of failures and downtime occurred in the line (failure data); 3) the warehouse availability (WIP data); 4) the order list and the related delivery time and the staff planning. In particular, the cycle time data are processed in order to verify if a trend exists leading to a deviation from the targeted behavior, to calculate its statistical distribution and to feed the DES model. Through a micro-service that exploits the Moving Average Convergence Divergence (MACD) methodology

(Papadamou and Tsopoglou, 2001), it is possible to isolate the subset of data collected after the starting of the trend. Then, to obtain the slope (i.e., the direction and steepness of the cycle time time series), an ordinary least squares model or a Theil-Sen estimator are used. If this slope is significantly different from zero, the fitting distribution micro-service calculates the best statistical distribution for that data, using Kolmogorov Smirnov hypothesis test with a subset of possible continuous distributions.

The simulation model represents the second and crucial part of the architecture. It consists of a DES model able to use the above mentioned real time data as initial settings. The model plays the role of the digital twin having access to that information and perfectly replicating the actual behavior of the plant. Once created with the native blocks of the software (e.g., sources, buffers, conveyors, processes and so on), it is enriched with functions and connectors that allow the automatic insertion of parameters, the start of execution and the export of results. In particular, methods are implemented to allow the processing block to use the right processing time for the specific entity it has to work on, by reading this value from the imported data. Moreover, the DES model is self-resetted and runs with a user specified duration, based on the specific needs of the use case. Finally, when the simulation stops, a method extracts the values of specific KPIs (e.g., throughput per day, per hour, per minute for each entity type or machine usage) in order to export them automatically. Finally, a configuration matrix is added, to enable the user to change the flow of the model evaluating alternative production strategies. For example, it is possible to add resources to a process, activate backup stations, change the buffer capacity or the number of working shifts without manually modifying the DES model.

The data coming from the model execution are visualized in a graphical user interface (GUI), to show to the final users the consequences of the actual behavior of the line. In particular, the results of the simulation are compared with the expected target performances in terms of throughput, machine usage, availability, and revenues. Moreover, in the GUI the user can create a scenario to be simulated with real time data, in order to find how to improve the performances previously estimated. This arrangement is essentially a modification of the previously mentioned configuration matrix. Also in this case, results are shown in terms of target and forecast KPI comparison. To summarize, the main goal of the GUI is to give the user a decision support tool that provides suggestions immediately applicable in the plant to control its performances. The resulting architecture is depicted in Fig. 1.

5. USE CASES FOR APPROACH VALIDATION

To test and validate the proposed approach, the DT is implemented in two different use cases. The first is a manual assembly line in Comau, headquarter in Turin, while the second is a line located at a customer plant. The first line is a brownfield case, i.e., an already existing manufacturing line in which the proposed tool has been added. The second case is a greenfield case, i.e., a new physical system on an automate cell in the final assembly phase of an automotive production line. The proposed

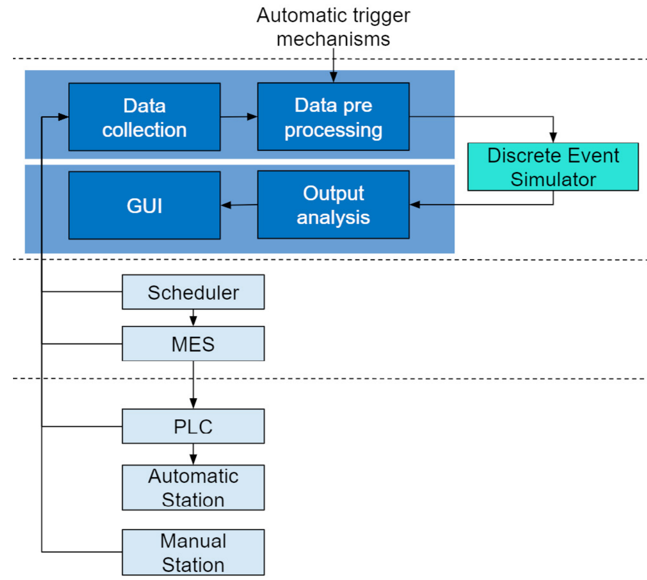

Fig. 1. Architecture of digital twin.

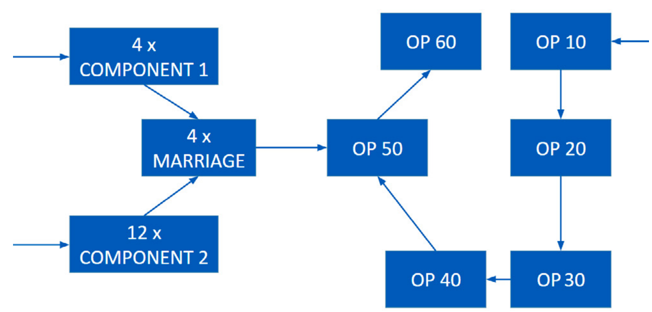

Fig. 2. Assembly production line scheme.

architecture is applicable in both cases with promising results.

The input data management and the output analysis are developed in Python and are the same for each use case, except for specific functions for data pre-processing. Instead, the choice of the software to implement the DES models actually depends on company requirements. The Comau (brownfield) use case was implemented in Flexsim, as it is internally used for research scope. The second (greenfield) use case was implemented in Plant Simulation, as it was a customer requirement. The use of different softwares proves that the approach is agnostic to a specific simulation engine. Finally, the GUI is implemented in Python Plotly and Angular.

5.1 Brownfield case

The use case addresses an assembly line composed of four main parts, as shown in Fig. 2. Specifically, on the left side there are two parallel operations for the first and second component assembly (with 4 and 12 parallel stations, respectively) and the marriage of these two parts (with 4 parallel stations). On the right side, there are the third component assembly, with 4 operations in series, the final marriage of the three components and a final operation. All of these operations are manual and, hence, a Programmable Logic Controller (PLC) is not present. At the end of each of them, a human operator confirms the success of the OP with a barcode reader that registers the event in the central database. The aim of the DT in this use case is to predict the line throughput per shift, day and week, and anomaly deviations from the target cycle time. In addition, the DT is used to find efficiency measures (e.g., add a shift or increase the resources available in OPs) to minimize the production delay to deliver the goods on time to customers or at least decrease it.

The first step is to collect and pre-process data coming from the Comau central database. This contains the operational data of the line, but also the production plan (i.e., the list of jobs to be done during the day, week and month). The first micro-service of the input data management module analyses the first type of data. It calculates the cycle time of every OP using the timestamp of two consecutive confirmations and eventually removes outlier values (based on their quantiles). The output of this function is reported in Table 1. The *timestamp* column represents the time the production of a piece of *product variant* ends in station *object* that had a duration, in seconds, equal to *cycle time*. These data are processed in order to detect a possible trend or deviation from the target behavior and, consequently, to find the best statistical distribution for the process, which presents the anomaly, using the specific micro-service. Finally, the best result is translated into Flexsim syntax (i.e., *normal(3600, 900)* represents the normal distribution with mean 3600 and standard deviation 900, both expressed in seconds) and inserted into a table of a dedicated database reached from the DES model. On the other hand, the production plan is read from the database and then transformed in Flexsim format to be readable in the Source block, as shown in Tables 2 and 3, and inserted in the previously mentioned database. In these tables, each row represents a batch of items to be created in the simulation, which is characterized as follow: *item name* takes the default value "Product", the *quantity* is the batch size and *optional label* specifies a property of the item. In our case, this represents the variant of the piece. Table 3 contains an additional column, *arrival time*, to specify the time that the entity has to be created in the simulation.
This first step of the methodology is repeated with a daily frequency by considering data collected with one week horizon. On one hand, the definition of this strategy is aligned with the (roughly) one-hour cycle time of this line; on the other hand, it was aligned with line personnel opinion.

To make all the above data available for the DES, a connector is configured in the model in order to allow information to be caught from external sources. In this way, the model is able to import the process actual cycle time and the scheduling of the production, and also to export result data (i.e., predicted throughput per hour, day and shift, machine usage and availability). The DES model, shown in Fig. 3, is built using three different entity sources for stations that represent the components assembly. All of them follow the same production plan. Then, to model the assembly and marriage operations, process and combiner blocks are used. The baseline model is configured with two working shifts, one first component assembly station and one marriage station disabled, according to the real functioning of the physical line.

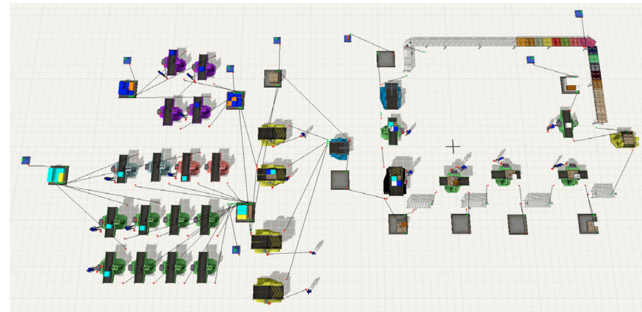

Fig. 3. Flexsim simulation model.

To conclude, the results are shown in the GUI that visualizes the input data analysis with histograms and boxplots, and the results of trend detection and fitting distribution in table format, as reported in Fig. 4. Then, the results of the simulation, in terms of throughput, and the results of different scenarios evaluation are reported in Figures 5 and 6, respectively. This use case addresses the following scenarios: three working shifts instead of two, activate one more station for the first component assembly, activate one more station of component marriage. The definition of such scenarios followed a discussions with the plant personnel and represents the most feasible and applicable line improvements.

The most important feature of this approach is the enabled possibility to make sensitivity analysis of the input data and of the actual behavior on the output KPIs. First, it is possible to determine how much and in what time a deviation of the cycle time significantly affects production in terms of throughput. Moreover, it is possible to understand if this deviation produces an unexpected bottleneck in the system that could influence the target station usage or availability as well as the throughput. Finally, thanks to the possibility of evaluating different scenarios and model configurations, the user can understand how to increase the efficiency of the line based on the actual behavior.

Table 1. Operational data structure after input.

timestamp	object	cycle time	product variant
10:12:15 22/10/2020	op10	3600	A
11:12:15 22/10/2020	op10	3600	A
12:12:15 22/10/2020	op20	4250	B
...	...	...	...

Table 2. Flexsim Source data acceptable formats.

item name	quantity	optional label
Product	1	label value
Product	1	label value

Table 3. Flexsim Source data acceptable formats.

arrival time	item name	quantity	optional label
0	Product	1	label value
0	Product	1	label value

5.2 Greenfield case

The second use case is under development by Comau in a customer plant whose layout cannot be displayed due

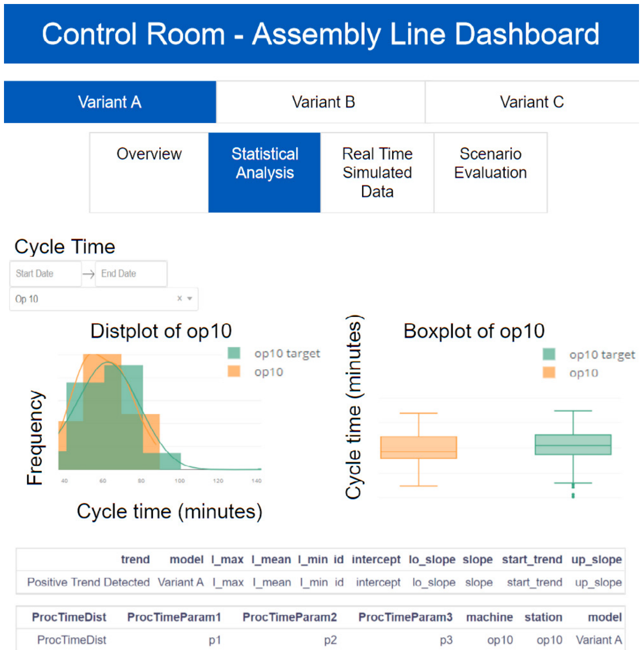

Fig. 4. Data management module visualization.

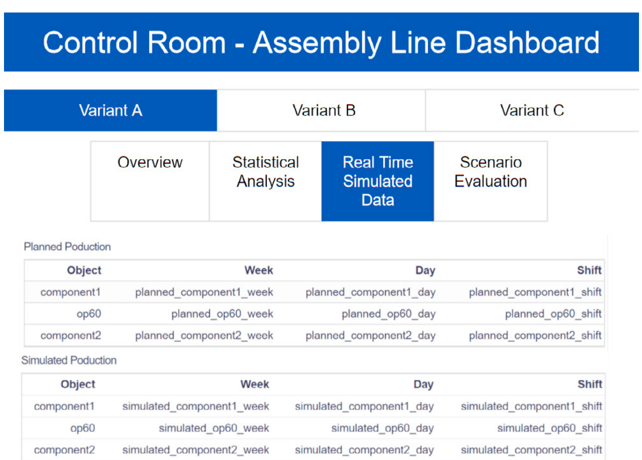

Fig. 5. Simulated throughput visualization.

to patent confidentiality reasons. The system is an automated cell in the final assembly phase of an automotive production line. All the stations are robotic and, due to the fact that the flow is fixed and without buffers within the stations, there is backup equipment activated only in the case of faults on the primary machinery. Differently from the previous use case, here a PLC exists. As Comau has followed the cell development since its design stage, Comau control engineers are able to program the PLC in order to record on specific variables, called tags, the cycle time of each operation. In this way, this time is available for the analysis.

With the specific micro-service, input data are aggregated to obtain the duration of a series of operations that represents a unique block of the simulation model (e.g., all the operations that compose the pick or the place operation of a robot, such as the validation of the vision system of the right positioning or the right vacuum engage). Then,

Fig. 6. User interface to configure a new scenario and its simulated throughput visualization.

the trend detection micro-service tests if a deviation from the target cycle time exists with a hourly frequency and by using data collected through one day and, in this case, the fitting distribution micro-service calculates the best distribution for the cycle times. Notice that these services, deployed on the Azure Cloud, enable this approach not to be a stand-alone implementation but part of the customer information technology system. Finally, these data feed the simulation model implemented in Plant Simulation. To interact with the simulator, a micro-service based on Plant Simulation native internal web server was implemented. In fact, the data exchange was done by HTTP POST requests.

Regarding the results, also in this case, the sensitivity analysis that can be done is very promising. First, the impact of micro stoppages on the throughput of the production line can be evaluated. Second, it can be assessed whether and when to activate the backup equipment can improve performance by losing only an acceptable number of jobs per hour (JPH). Finally, if the input analysis shows that, with a particular type of product, the production line has a systematic increase on the cycle time, it is possible to change the production plan to temporarily stop the production of that variant of product.

6. CONCLUSIONS AND FUTURE WORKS

Digital twins are some of the most promising technologies of I4.0, due to their potential capacity to make manufacturing systems more flexible and re-configurable. The key concept is the real time connection with the field in order to replicate into the digital environment the physical world. This paper proposes an approach to implement the DT in a manufacturing environment using a simulator as virtual counterpart (specifically, a DES model). In fact, usually a simulator of the manufacturing environment already exists because it has been developed during the design of the physical system but, until now, it has not been connected and exploited during the production phase. In this paper, the simulator is used as DT from the design to the production phase. In fact, an approach to connect the DES model in real time is proposed; the DES model is developed at the beginning of the system life cycle, with the physical equipment, analysing and pre-processing data coming from the field. Afterwards, the results of the simulation are shown to give at the user a decision support tool to compare the predicted performances of the line with the planned ones. In conclusion, the two proposed industrial use cases reveal the promising benefits of the followed approach.

Future works should address data pre-process, not only for the evaluation of cycle times, but also for all the other types of data used to feed the simulator. In fact, feeding the DT with more and more information will increase its adherence to the real system. Moreover, efforts should focus on the implementation of a methodology that, starting from the input data, is able to autonomously create the DES model and to catch the variations of the simulator from the real system. Finally, to close the loop from the virtual to the physical equipment, methods to enhance the communication between the DT to the real system should be developed, to apply corrective methods to improve the performances of the production line in real time.

REFERENCES

Beregi, R., Szaller, Á., and Kádár, B. (2018). Synergy of multi-modelling for process control. *IFAC-PapersOnLine*, 51(11), 1023–1028.

Cimino, C., Negri, E., and Fumagalli, L. (2019). Review of digital twin applications in manufacturing. *Computers in Industry*, 113, 103130.

Dahmen, U. and Rossmann, J. (2018). Experimentable digital twins for a modeling and simulation-based engineering approach. In *2018 IEEE International Systems Engineering Symposium (ISSE)*, 1–8. IEEE.

Garrido, J. and Sáez, J. (2019). Integration of automatic generated simulation models, machine control projects and management tools to support whole life cycle of industrial digital twins. *IFAC-PapersOnLine*, 52(13), 1814–1819.

Karanjkar, N., Joglekar, A., Mohanty, S., Prabhu, V., Raghunath, D., and Sundaresan, R. (2018). Digital twin for energy optimization in an smt-pcb assembly line. In *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, 85–89. IEEE.

Kunath, M. and Winkler, H. (2018). Integrating the digital twin of the manufacturing system into a decision support system for improving the order management process. *Procedia Cirp*, 72, 225–231.

Lu, Y., Liu, C., Kevin, I., Wang, K., Huang, H., and Xu, X. (2020). Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*, 61, 101837.

Modoni, G.E., Caldarola, E.G., Sacco, M., and Terkaj, W. (2019). Synchronizing physical and digital factory: benefits and technical challenges. *Procedia CIRP*, 79, 472–477.

Monostori, L. (2014). Cyber-physical production systems: Roots, expectations and r&d challenges. *Procedia Cirp*, 17, 9–13.

Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., and Ueda, K. (2016). Cyber-physical systems in manufacturing. *Cirp Annals*, 65(2), 621–641.

Negri, E., Fumagalli, L., and Macchi, M. (2017). A review of the roles of digital twin in cps-based production systems. *Procedia Manufacturing*, 11, 939–948.

Nonaka, Y., Suginishi, Y., Lengyel, A., Nagahara, S., Kamoda, K., and Katsumura, Y. (2015). The s-model: A digital manufacturing system combined with autonomous statistical analysis and autonomous discrete-event simulation for smart manufacturing. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 1006–1011. IEEE.

Papadamou, S. and Tsopoglou, S. (2001). Investigating the profitability of technical analysis systems on foreign exchange markets. *Managerial Finance*.

Sujová, E., Čierna, H., and Żabińska, I. (2019). Application of digitization procedures of production in practice. *Management Systems in Production Engineering*, 27(1), 23–28.

Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., and Sui, F. (2018). Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology*, 94(9-12), 3563–3576.

Tao, F. and Zhang, M. (2017). Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing. *Ieee Access*, 5, 20418–20427.

Vachálek, J., Bartalskỳ, L., Rovnỳ, O., Šišmišová, D., Morháč, M., and Lokšík, M. (2017). The digital twin of an industrial production line within the industry 4.0 concept. In *2017 21st international conference on process control (PC)*, 258–262. IEEE.