CIRP Annals - Manufacturing Technology 70 (2021) 349–352

Contents lists available at ScienceDirect

CIRP Annals - Manufacturing Technology

journal homepage: https://www.editorialmanager.com/CIRP/default.aspx

An intelligent agent-based architecture for resilient digital twins in manufacturing

Rok Vrabič (2)[a,*], John Ahmet Erkoyuncu (2)[b], Maryam Farsi[b], Dedy Ariansyah[b]

[a] *Faculty of Mechanical Engineering, University of Ljubljana, Slovenia*
[b] *School of Aerospace, Transport and Manufacturing, Cranfield University, United Kingdom*

ARTICLE INFO

Article history:
Available online 11 June 2021

Keywords:
Manufacturing system
Digital twin
Machine learning

ABSTRACT

Digital twins (DTs) offer the potential for improved understanding of current and future manufacturing processes. This can only be achieved by DTs consistently and accurately representing the real processes. However, the robustness and resilience of the DT itself remain an issue. Accordingly, this paper offers an approach to deal with uncertainty and disruptions, as the DT detects these effectively and self-adapts as needed to maintain representativeness. The paper proposes an intelligent agent-based architecture to improve the robustness (including accuracy of representativeness) and resilience (including timely update) of the DT. The approach is demonstrated on a case of cryogenic secondary manufacturing.

1. Introduction

Digital twins (DTs) are increasingly gaining interest from industry as an enabler for enhanced asset and process performance [1]. The vision for DTs has often been stated as offering a comprehensive physical and functional description of a component, product, or system that contains more or less all information useful in the current and subsequent lifecycle phases [2]. Moreover, DTs are often assumed to provide accurate and comprehensive representations of the physical asset or process [3]. However, the lack of approaches for adaptively connecting existing systems and their data limits the accuracy of DTs [4].

Franciosa et al. [5] demonstrated the results of using the first fully digitally developed remote laser welding process for aluminium doors, which yielded a right-first-time rate of >96% for door assembly cell development. Soderberg et al. [2] emphasise that accuracy challenges in DTs, in the context of real-time geometry assurance, are driven by faster optimization algorithms, higher computer power, and increased amount of available data. Erdos et al. [6] propose that by improving the twin closeness through multilevel calibration methods, an accurate DT of the as-built cell can be established in which the sufficient accuracy of offline planned robotic operations is guaranteed. Preuveneers et al. [7] investigated the extent to which faults in DTs can compromise intelligent cyber-physical production processes, and suggested that DTs should be equipped with safety features such as feature toggles and software circuit breakers capable of intercepting local faults and preventing them from propagating and cascading to other systems. Stark et al. [4] highlight that real-time simulation takes place as the DT controls the manufacturing process with the information about the status of the process and the positions of every object. However, there is a significant assumption that the positions of moving objects are accurately and continuously updated for the real-time simulation of the DT in the context of the mill and the pick-and-place robot.

While there are gaps in terms of the mechanisms to increase the accuracy of DTs, Helu et al. [1] emphasise the need to complement simulation and support optimal decision making throughout the lifecycle, as means to align and contextualize as-planned and as-executed process data dynamically using semantically-rich, open standards. There are numerous research gaps related to 1) real-time identification of the DT accuracy status, 2) development of an optimisation framework to decide how to self-correct the DT with respect to the data feed and its models, and 3) mechanisms to self-adapt the DT [4,8,9].

Tomiyama and Moyen [10] focus on the role resilience can play in minimising downtime of cyber-physical production systems. Their approach to this challenge was to develop a design framework to implement resilience in order to avoid catastrophic operational disruptions. This relied on the production system itself responding resiliently and autonomously to disruptions and failures and maintaining its functions at an acceptable level.

The existing literature presented emphasises the issues of DT accuracy, calibration, safety and reliability, simulation capability, and the impact of DTs on the resilience of the underlying system. However, to the authors' knowledge, the concept of resilience has not yet been applied to DTs themselves and represents the novelty of this work. The authors argue that resilience should be considered not only in physical manufacturing processes, but also in the digital assets including DTs, as it can enable the relevance of the digital support assets to be sustained in decision making over time. In this context, resilience refers to the ability to respond to disruptions experienced in the DT that may arise due to various reasons such as missing data, a temporary disruption in the data feed, or models that are not up to date given the evolving asset or its processes.

* Corresponding author.
E-mail address: rok.vrabic@fs.uni-lj.si (R. Vrabič).

https://doi.org/10.1016/j.cirp.2021.04.049
0007-8506/

This paper applies principles of resilience engineering for DTs to bridge the gap between the as-planned and the as-executed manufacturing process. The focus is on how to increase the consistent and accurate representativeness of the DT over time given the dynamic behaviour of the component, product, or system. The novelty of the work lies in the proposed innovative approach that integrates resilience and machine learning to self-learn and adapt the representativeness of the DT.

2. A learning agent architecture for resilient digital twins

Alignment of the DT and the real system is an ongoing research topic. The DT must respond to the changes and disruptions the real system experiences. It is important that the response is accurate and that the recovery is rapid in order to minimise the delta between the DT and the real system. Simple optimisation approaches are often insufficient for adaptation because the real system inherently operates under uncertainty, making it difficult to pinpoint a source when a deviation from expected behaviour occurs and to distinguish between actual changes and random fluctuations.

The robustness and resilience of a DT can be improved over time if the DT has a learning capability that allows it to improve its adaptability based on historical operational data and what-if scenario simulations. This can give the DT intelligence that allows it to predict future behaviour, effectively. In turn, the learning capability can also be used to detect anomalies, i.e., to detect when the behaviours of the real system and the DT diverge. In the proposed approach, the learning capability is realised by introducing a learning agent [11]. The goal of the agent is to restore DT fidelity after a change occurs in the real system that causes a deviation between the real system and the DT and represents a disruption in DT accuracy. The disruption in the accuracy of the DT could be caused by any change in the real system that is not captured in the DT. The agent continuously performs the following tasks: (1) detecting disruptions, (2) determining the source and magnitude of the disruptions, (3) formulating a response strategy by simulating what-if scenarios, and (4) autonomously adapting of the DT.

The learning agent *monitors* a set of properties of the real system - observables - and compares their values with those predicted by the DT. Observables can be obtained directly from the sensors or, for example, represent system states as recorded by the information system. The difference between the real system and DT is evaluated by an *anomaly detection* mechanism based on a *performance standard*, which in this case is DT accuracy. When an anomaly is detected, the DT needs to be updated. This is supported by *learning*, the goal of which is to determine how the difference in values can be explained, whether by looking at historical data (something similar has already been observed in the past) or by looking at simulated what-if scenarios (this would be observed if the DT had been modified). The scenarios are generated continuously and simulated in parallel without affecting the operation of the DT, which is governed by the same control inputs (decisions) as the real system. To capture the results of the scenario evaluation, the learning mechanism uses a neural network which allows it to learn over time and extrapolate beyond the particular scenario simulation results. The architecture of the learning agent is shown in Fig. 1.

The *performance standard* should be independent of the real system, therefore, the ranges of values of the observables should be normalised. By monitoring the observables of the real system, their distribution can be determined. The mean of the distribution is mapped to 0, while the standard deviation is mapped to the interval $[-1, 1]$. Then, the root mean square deviation (RMSD) is calculated over the normalised values of the observables and DT predictions. The deviation is thus expressed in units of standard deviation from the normal behaviour of the system. 1 − RMSD is proposed as the DT accuracy measure. Given N normalised observables at time t, $\boldsymbol{O}^t$, and corresponding normalised DT predictions for time t, $\boldsymbol{P}^t$, the measure for DT accuracy is defined as shown in Eq. (1).

$$d(t) = 1 - \sqrt{\frac{\sum_{i=1}^{N}\left(o_i^t - p_i^t\right)^2}{N}} \quad (1)$$

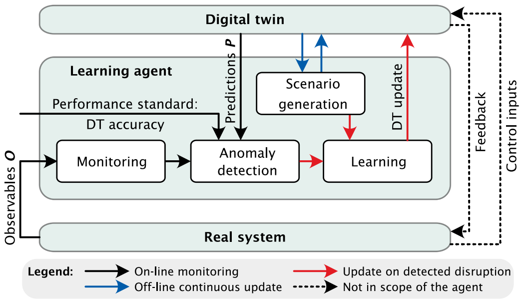

Fig. 1. Architecture of the learning agent.

Since the values are normalised to the standard deviations, the interpretation of the measure is straightforward. When the observed and predicted values are equal, the value of the measure is 1. The value decreases as the deviation increases, e.g., it is 0 when the values differ by one standard deviation on average. During the operation of the system, the measure is expected to stabilise at a steady-state value related to the intrinsic uncertainty present in the system and to the prediction horizon of the DT. Systems with a lot of uncertainty will tend to deviate more from 1 because their DTs are unable to capture the exact future behaviour.

The DT accuracy measure d can then be used by the *anomaly detection* mechanism. When the value of the measure deviates from the steady state, this signifies that a change has occurred in the real system and that the DT should be updated. To determine the source of the anomaly and formulate a response strategy, the agent generates and simulates what-if scenarios for various disruptions using the DT. The agent encodes the knowledge gained in this way into a neural network that learns how to predict the parameters of the DT based on past and present observables. The input to the neural network is a vector containing observables at subsequent times of observation, $(\boldsymbol{O}^{t-1}, \boldsymbol{O}^t)$, while the output is a vector of the parameters of the DT. The neural network is thus able to determine how the DT should be updated so that the behaviour of DT matches the observed behaviour of the real system.

The agent randomly generates scenarios of disruptions that may occur in the future and simulates them to continuously feed the neural network with new data. The disruptions are simulated as random changes in the parameter values of the DT. The scenarios are simulated off-line, i.e., independently of the operation of the real system and the DT. The results are used as training data for the neural network in addition to the data that is generated by the DT during operation. When a disruption is detected, the learning element uses the neural network to update the DT.

In order to quantify the response, measures of robustness and resilience need to be defined in terms of the DT accuracy measure d. This is based on established literature [12] and shown in Fig. 2. In this context, robustness refers to the loss in accuracy after the disruption occurs with respect to the steady-state value and is calculated as $1 - (d_{lost}/d_{ss})$. Resilience, on the other hand, is defined through the

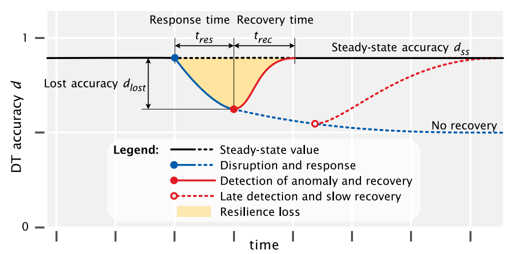

Fig. 2. Expected response to disruption in terms of the DT accuracy.

resilience loss, which is the integral of the lost accuracy due to the disruption (highlighted in yellow on Fig. 2). Resilience is then defined as the lost resilience (as a percentage with respect to the total possible loss) subtracted from 1. For a given DT and its response to disruption, robustness can be improved by faster anomaly detection, while resilience can additionally be improved by faster recovery.

3. Case study: cryogenic secondary manufacturing

A case study from one of the UK's cell and gene therapy (CGT) secondary manufacturing facilities was selected to test the proposed intelligent agent-based architecture [13]. Secondary manufacturing refers to the storage, packaging, and distribution of advanced therapy medicinal products (e.g. the cryogenic warehousing processes following the product manufacture). To collect the required information, 5 site visits, 13 meetings and 5 interviews with the Global Director, Head of Operations, and the Project Team Manager were conducted. The manufacturing system consists of three main phases: receipt and inventory (R&I), storage & monitoring (S&M), and distribution (D). There are three groups of workers on the shop floor including 18 technicians (G1), 2 service (G2), and 12 quality control (G3) staff members. At R&I phase, different types of materials, i.e., cryoproducts and shippers (containers for the cryoproducts), are delivered to the warehouse with the average daily rate of $\lambda_{RI} = 12$, and are receipted at the receipt stage by the service staff. The material types are summarised in Table 1.

Table 1
Material arrival proportions by type.

Delivery type	Symbol	Arrival proportion	Failure rate λ_F
Empty used shippers	EUS	89%	0%
Empty new shippers	ENS	3%	10%
Empty damaged shippers	EDS	2%	1%
Shippers with new products	SNP	3%	0%
Shippers with returned products	SRP	3%	0%

Some of the shippers are disposed due to the potential faults at the R&I phase, with a failure rate of λ_F (see Table 1). The suitable shippers are recycled, and the products are quarantined in the pre-storage stage. Afterwards, the products are stored in cryogenic tanks and documented. The shippers are stored separately, and their data loggers are then documented at the logger stage. After both product and shipper documentation is completed, inventory records are updated at the end of the S&M phase. In parallel with the first two phases, clinics and hospitals place different orders with the average daily rate of $\lambda_D = 12$, at phase D. The orders are documented and scheduled for dispatch at the order planning stage. Accordingly, the shippers and the products are collected and brought to the dispatch area. Finally, the orders are quality checked, documented, and dispatched. The flow of the processes in these three phases is shown in Fig. 3.

Data on process cycle times and corresponding workforce resources are summarised in Table 2. It is worth noting that the following assumptions and limitations are made: The simplified model used in this study does not consider multitasking for the workforce, and sharing of equipment resources within the shop floor.

Table 2
Cycle times and resources for the processes.

Events	Resources	Cycle-time (minutes)
Phase I: R&I		
Receipt	G1	Triangular: T(6.8,30.7,13.6)
Phase II: S&M		
Pre-storage	G1	Exponential: E(4.25)
Logger process	G3	Triangular: T(5,10,7)
Shipper storage	G1	Poisson: P(619.0)
Product storage	G1	Poisson: P(71.5)
Documentation	G3	Triangular: T(74.0,90.9,83.2)
Phase III: D		
Order planning	G1	Triangular: T(4, 6, 5)
Collect shipper	G1	Poisson: P(14.3)
Collect product	G1	Poisson: P(37.5)
Quality check	G3	35.6
Dispatching	G2	120.0

4. Results

The proposed DT resilience architecture was applied to the described case. A discrete event simulation model of the manufacturing system was created in Python, based on the real system data presented in Tables 1 and 2 and on the original simulation model that has 98.962% accuracy [13]. Three types of disruptions were considered, based on the priorities identified in the case study: (1) a temporary 20% increase in the material arrival rate, (2) a external temporary 20% increase in the order arrival rate, and (3) a temporary unavailability of 20% of staff members from group G1. In each simulation run, the simulation model is instantiated twice: one instance represents the real system, i.e., the ground truth, and the other represents the model used by the DT. For the observables, the queue levels of processes shown in Fig. 3 are selected. This means that the agent cannot directly observe the disruptions (changes in the model parameters), but can only indirectly observe their effects on the queue levels. The predictions of the queue levels are generated by the DT every 10 min for a prediction horizon of 10 min.

The simulation model is run 1000 times with default parameters and no disruption to obtain the ranges for the observables, which serve as the basis for normalisation of their values. Then, the neural network is created and pre-trained on 1000 runs of the simulation model, but with randomised material arrival, order arrival, and number of G1 staff members. The neural network uses a feedforward topology, two densely connected layers with 64 neurons each, and relu activation function. The outputs of the neural network are the material arrival rate, the order arrival rate, and the numbers of staff members in groups G1, G2, and G3.

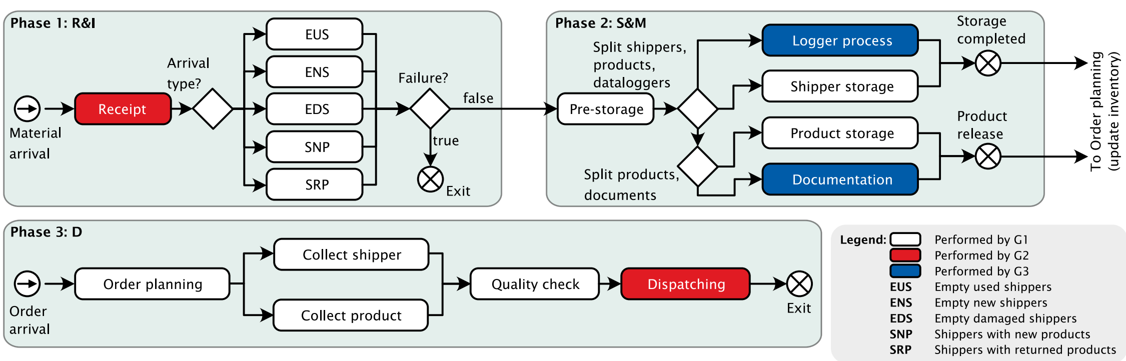

Fig. 3. Simplified process model of CGT secondary manufacturing.

After the neural network is pre-trained, the learning agent is introduced into the simulations. The simulations are then run 100 times to obtain averages and standard deviations. In each simulation, a scenario is randomly selected.

When a scenario with a disruption is simulated, the same disruption occurs twice: at $t = 60$h and at $t = 160$h, with a duration of 20h. Before the start of the second disruption event, the neural network is trained with additional data of the DT run up to that point to check whether the response at the re-occurence of the disruption is improved through learning.

Fig. 4a illustrates how the DT accuracy decreases due to the disruptions. As a baseline, the dark blue line shows the steady-state behaviour of the system without disruptions, along with the corresponding standard deviation in light blue. After settling to the steady state in the first 40h of simulated time, the DT remains 87.2% accurate throughout the simulation ($d_{ss} = 0.872$). When disruptions are introduced and no corrections are made to DT, the DT accuracy drops to approximately one standard deviation from the steady-state value, as shown by the red dashed line.

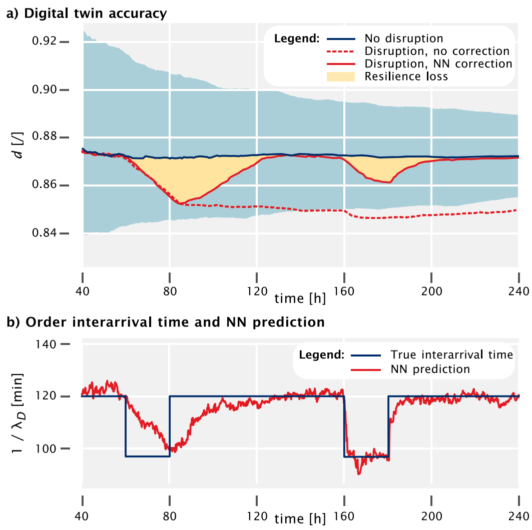

Fig. 4. a) response to (repeated) disruption and b) example of neural network (NN) predictions for order interarrival time.

Using the proposed approach (red line), the accuracy drops when the disruption is introduced ($d_{lost} \approx 2.3\%$) but it quickly improves afterwards ($t_{rec} \approx 45$h). Due to the training with additional data before the same disruption is reintroduced at time $t = 160$h, the response is more robust, with a smaller drop in DT accuracy ($d_{lost} \approx 1.3\%$) and more rapid with a shorter time to get back to steady-state ($t_{rec} \approx 25$h). The main reason for this can be seen in Fig. 4b, which shows the corresponding neural network predictions for the cases where the introduced disruption is the increase in the order arrival rate. It can be clearly seen that the neural network is able to predict the drop in the interarrival time more accurately at the reoccurence of the disruption.

The results show that the DT robustness and resilience are improved by using the proposed approach. When the disturbance occurs for the first time, the robustness is slighly improved (red vs. dashed red line in Fig. 4a), but is further improved when the disturbance occurs again. The yellow highlights in Fig. 4a, which visualise the resilience loss of the DT, clearly show that not only the resilience is improved compared to the case without correction, where the accuracy never recovers, but also that the agent is able to improve the DT resilience when the disturbance reoccurs by learning from historical data. The resilience loss is almost halved, from 1.26% on the first occurrence to 0.69% on the second one.

5. Conclusions and future work

The paper offers a novel process to evaluate and improve the robustness and resilience of DTs in manufacturing processes. It achieves this by developing an intelligent agent-based architecture that can be used to 1) detect any disruption to the DT, 2) evaluate the disruption, 3) determine a response, and 4) self-adapt the DT. The simulation-based case study demonstrates that the robustness and resilience of the DT can be improved in presence of uncertainties and disruptions. Without the developed learning agent, the accuracy never recovers after a disruption. Moreover, the agent learns on operational data and is able to almost halve the resilience loss when the same disruption re-occurs.

A key question remains how to reliably determine causal relationships in cases where the effects of different causes are similar. By introducing an agent that monitors the behavior of the system over longer time scales and performs root-cause analysis, it would be possible to more accurately discern between different causes. However, this would require greater computational resources and, possibly, the involvement of expert knowledge, and remains the subject of further research. For future work, it would also be worthwhile to investigate how the introduction of additional sensing affects the DT accuracy and contributes to the correct identification of causes. An approach should be developed on how to parameterize the structure of DT as well, and to incorporate domain knowledge to assess its feasibility. Most importantly, however, future work should enable feedback from DT back to the real system. The proposed approach could be extended by introducing the ability to suggest changes that could improve the performance and resilience of the real system in the face of uncertainty and disruptions.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The research was partially supported by the EPSRC funded project DigiTOP (EP/R032718/1) and the Slovenian Research Agency (P2-0270).

References

[1] Helu M, Joseph A, Hedberg T (2018) A standards-based approach for linking as-planned to as-fabricated product data. *CIRP Ann. – Manuf. Technol.* 67(1): 487–490.
[2] Söderberg R, Wärmefjord K, Carlson JS, Lindkvist L (2017) Toward a Digital Twin for real-time geometry assurance in individualized production. *CIRP Ann. – Manuf. Technol.* 66(1):137–140.
[3] Erkoyuncu JA, Amo IF del, Ariansyah D, Bulka D, Vrabič R, Roy R (2020) A design framework for adaptive digital twins. *CIRP Ann. – Manuf. Technol.* 69(1):145–148.
[4] Stark R, Fresemann C, Lindow K (2019) Development and operation of Digital Twins for technical systems and services. *CIRP Ann. – Manuf. Technol.* 68(1): 129–132.
[5] Franciosa P, Sokolov M, Sinha S, Sun T, Ceglarek D (2020) Deep learning enhanced digital twin for Closed-Loop In-Process quality improvement. *CIRP Ann. – Manuf. Technol.* 69(1):369–372.
[6] Erdos G, Imre P, Bence T (2020) Transformation of robotic workcells to digital twins. *CIRP Ann. – Manuf. Technol.* 69(1):149–152.
[7] Preuveneers D, Joosen W, Ilie-Zudor E (2018) Robust digital twin compositions for Industry 4.0 smart manufacturing systems. *IEEE 22nd International Enterprise Distributed Object Computing Workshop*, 69–78.
[8] Abramovici M, Göbel J, Dang H (2016) Semantic data management for the development and continuous reconfiguration of smart products and systems. *CIRP Ann. – Manuf. Technol.* 65(1):185–188.
[9] Tomiyama T, Lutters E, Stark R, Abramovici M (2019) Development capabilities for smart products. *CIRP Ann. – Manuf. Technol.* 68(2):727–750.
[10] Tomiyama T, Moyen F (2018) Resilient architecture for cyber-physical production systems. *CIRP Ann. – Manuf. Technol.* 67(1):161–164.
[11] Russell SJ, Norvig P (1995) *Artificial Intelligence: a Modern Approach*, Prentice hall.
[12] Hosseini S, Barker K, Ramirez-Marquez JE (2016) A review of definitions and measures of system resilience. *Reliab. Eng. Syst. Saf.* 145:47–61.
[13] Farsi M, Erkoyuncu JA, Steenstra D, Roy R (2019) A modular hybrid simulation framework for complex manufacturing system design. *Simul. Model. Pract. Theory* 94:14–30.