

## Research Article

Riku-Pekka Nikula\*, Marko Paavola, Mika Ruusunen, and Joni Keski-Rahkonen

# Towards online adaptation of digital twins

<https://doi.org/10.1515/eng-2020-0088>

Received Jun 19, 2019; accepted Jun 10, 2020

**Abstract:** Digital twins have gained a lot of attention in modern day industry, but practical challenges arise from the requirement of continuous and real-time data integration. The actual physical systems are also exposed to disturbances unknown to the real-time simulation. Therefore, adaptation is required to ensure reliable performance and to improve the usability of digital twins in monitoring and diagnostics. This study proposes a general approach to the real-time adaptation of digital twins based on a mechanism guided by evolutionary optimization. The mechanism evaluates the deviation between the measured state of the real system and the estimated state provided by the model under adaptation. The deviation is minimized by adapting the model input based on the differential evolution algorithm. To test the mechanism, the measured data were generated via simulations based on a physical model of the real system. The estimated data were generated by a surrogate model, namely a simplified version of the physical model. A case study is presented where the adaptation mechanism is applied on the digital twin of a marine thruster. Satisfactory accuracy was achieved in the optimization during continuous adaptation. However, further research is required on the algorithms and hardware to reach the real-time computation requirement.

**Keywords:** adaptation, differential evolution, digital twin, optimization, surrogate model

## 1 Introduction

Digital twin (DT) technology is the core component of cyber-physical systems (CPS) [1]. Such technologies have emerged from the rapid advances in ICT technologies and from national strategies and initiatives, such as ‘Industrie 4.0’, ‘Industrial Internet of Things’ (IIoT), and ‘Made in China 2025’, to name a few [1]. Grieves defined the DT originally in 2003 [2], although the definition by NASA was widely adopted later [3]. NASA defined DT as an integrated multi-physics, multi-scale, probabilistic ultra-fidelity simulation that reflects the state of the physical twin in a timely manner by using the best available physical models, sensor updates, and historical data [3]. Since then, various additional definitions have emerged [1].

The backbone technology of digital twins is real-time and multi-source data collection [4]. Information about the system operations, history, behavior, and the current state are required to create digital simulation models that dynamically update and change along with their physical counterparts. DT also adopts modern data visualization schemes such as virtual reality (VR) and augmented reality (AR), providing illustrative and user-friendly views [4].

The general and standard architecture for the digital twin [2] is composed of three components: physical objects, digital models, and the data that connect the physical and digital domains. The physical objects provide the basis for the building of the digital models. The digital models support the control, simulation, and decision-making of the physical objects. The data are the requirement for creating new knowledge. Additional components to the architecture have been proposed as well [5].

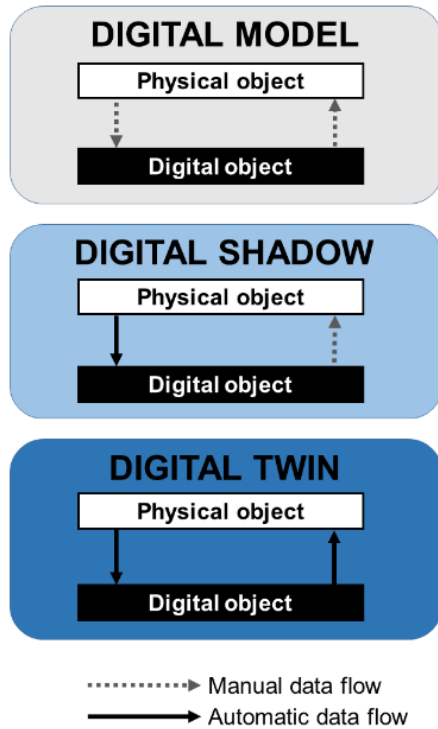
Although a common perspective is that digital twins are virtual copies of physical systems [1–3], there are some differences in the definitions of the data integration levels [6]. Kritzinger *et al.* [6] defined three levels of data integration, as illustrated in Figure 1. The digital model is a digital representation of the physical object without automated data exchange between the physical and digital objects. The digital shadow has an automated data flow from the physical object to the digital object, but not vice versa. Finally, the digital twin has fully integrated data flows in both directions. In the recently reported research, the interaction between the physical and virtual objects is mainly

\*Corresponding Author: **Riku-Pekka Nikula:** Control Engineering, Environmental and Chemical Engineering, University of Oulu, P.O. Box 4300, 90014 Oulu, Finland; Email: riku-pekka.nikula@oulu.fi

**Marko Paavola:** VTT Technical Research Centre of Finland, P.O. Box 1100, 90571 OULU, Finland

**Mika Ruusunen:** Control Engineering, Environmental and Chemical Engineering, University of Oulu, P.O. Box 4300, 90014 Oulu, Finland

**Joni Keski-Rahkonen:** Kongsberg Maritime Finland Oy, P.O. Box 220, 26101 Rauma, Finland



**Figure 1:** Data flow in digital model, digital shadow, and digital twin. (Modified from [6]).

off-line interaction, lacking continuous and online interaction [6, 7].

The first digital twin applications originated in the aerospace sector. At present, various types of DT including product digital twins, process digital twins and operation digital twins have been developed [7]. The case study in this paper focuses on the maintenance and lifecycle management areas of product digital twins. In these areas, digital twins have been proposed for the anomaly detection and condition monitoring of machines [1, 5], the monitoring of material deformations [8], and the modeling of the reliability of physical systems [9], for example. Moreover, digital twins have been proposed for mirroring the entire life of a physical object digitally. This includes tasks, such as virtual commissioning, where the lifetime of old machines is extended [10]. Furthermore, the environmental conditions and external factors require consideration in studies focusing on the long-term behavior of the system [11, 12]. Although digital twins could provide significant benefits to industry [6, 13, 14], the practical applications are still rare due to the challenges arising from the real-time adaptation. Moreover, the maintenance and calibration of the model are challenging but essential tasks in real-life applications with limited amount of reliable reference data.

The coupling between simulation and optimization is an essential subject for current research [15]. Practical challenges arise from the continuous and real-time data integration in digital twins. Zhang *et al.* [16] inferred that the calculation time in real-time optimization has decreased from hours to minutes recently. The minute level is adequate for certain applications such as chemical process control with large time constants. However, in certain applications such as in rotating machines, the significant phenomena may occur at rates from hertz to several kilohertz. The optimization algorithms face great challenges to meet the real-time requirements in such cases.

Moreover, physics-based simulators tend to be complex and relatively slow. Such simulators are used for gaining insight into how the system behaves and for analyzing the system outputs. The simulators are not necessarily optimized for fast iterative computations. Lighter modeling approaches such as surrogate models [17–19] are therefore needed in practical online applications.

In this study, an online adaptation mechanism for digital twins is proposed. A DT of a driveline in a marine thruster is used for testing the mechanism. A linear regression model is applied as a surrogate model to substitute for the physics-based Simulink<sup>®</sup> model of the driveline in order to speed up the computation. Differential evolution (DE) is used as the optimization algorithm due to its good convergence properties in global optimization [20].

In principle, the online adaptation is a parameter estimation task in moving time windows [21]. Model adaptation is a well-known research area, which has been studied from various perspectives such as concept drift adaptation [22], soft sensor adaptation [23], state estimation [24], adaptive control [25], physical model adaptation with statistical correction models [26], disturbance model adaptation [27], and the adaptation to different local operating regimes [28]. The mechanism proposed in this study uses global optimization and is a model-independent solution, which makes it a general solution and suited for a variety of different simulators, and both physics-based and data-based models.

The research objective in this paper is to study the real-time applicability and accuracy of the proposed adaptation mechanism. The need for this approach originated from the objective to estimate torque anywhere on the thruster driveline without a physical sensor in the corresponding location. Although such estimations can be done through modeling in this simulated case, direct modeling is not always possible, and a more general solution is therefore needed for digital twins. Moreover, the digital models or simulators that are run parallel to the physical system may operate under assumptions that do not fully match

the real events. In such a case, the model needs online adaptation to respond correctly to the events confronted by the physical system. This creates practical challenges for real-time simulation and adaptation, which are demonstrated in this study.

The rest of the paper is organized as follows. Section 2 introduces the studied product digital twin and the proposed adaptation mechanism. The results from the tests are shown and discussed in Section 3. Finally, the study is summarized in Section 4.

## 2 Digital twin adaptation

### 2.1 Thruster driveline digital twin

Figure 2 shows a simplified drawing of an azimuth thruster, which can be applied in towboats, for example. This thruster driveline is operated by an electric motor. The driveline can be divided into three parts: the input shaft, vertical shaft, and propeller shaft. The driveline includes two reduction gears, the propeller, and a flexible coupling in the input shaft.

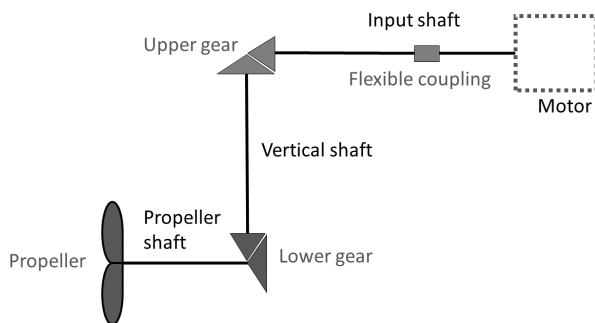


Figure 2: Simplified drawing of thruster driveline.

Transient torsional vibration models were used on the physical modeling of the driveline. The complete driveline was compiled in the Simulink<sup>®</sup> environment using Simscape<sup>™</sup> components. The Simulink<sup>®</sup> model can be used to estimate torque in any part of the driveline. A detailed representation of the system and its model structure is omitted in this study due to proprietary limitations.

### 2.2 Adaptation mechanism

Figure 3 portrays the digital-physical interaction of a digital twin and the position of the adaptation mechanism in

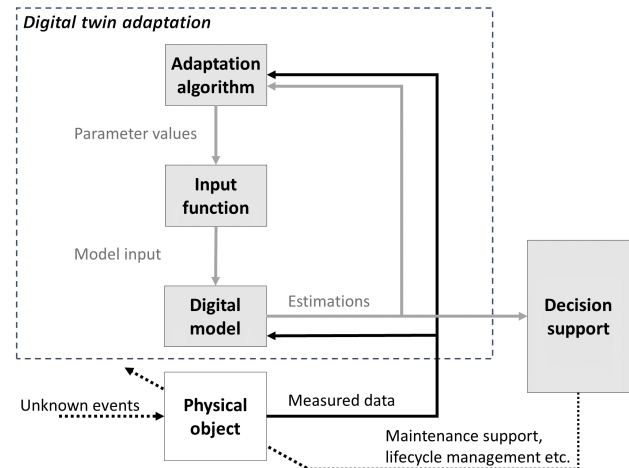


Figure 3: Schema for online adaptation of digital twins.

the overall schema. The physical object is the real system, which is monitored online and generates measured data. It is affected by unknown events that cannot be predicted in advance. The system has a digital model (twin), which acquires data from the input function and the measurements. The input function receives estimated (parameter) values from the adaptation algorithm (differential evolution). The input adaptation is an iterative procedure, where the deviation of the measured system output and the estimated output is minimized. When the data integration level of a digital shadow is applied, the estimates can be used in decision support, for example. If the estimates are used to control the system operation automatically, the highest data integration level is obtained [6].

In this study, the digital model is used to estimate the values of a variable, of which in situ measurement is avoided due to practical considerations, such as low cost efficiency. The estimated variable is the torque from the propeller (see Figure 2). The measured and estimated torque values of the driveline could be used for maintenance support and lifecycle management in a practical application. However, the practical use of the information provided by the digital twin is beyond the scope of this study.

The adaptation is done in short time windows and therefore the real-time requirements are strict. Here, the input function estimates a constant value for each time window. However, a more complicated input function, such as a polynomial function, can be used as well. In this study, only one variable of the model was adapted. However, the mechanism is not limited to the adaptation of a single variable.

### 2.2.1 Differential evolution

Differential evolution, developed by Storn and Price [20], is a heuristic approach to global optimization with real numbers [29]. It has been used successfully in diverse domains [30–33], and therefore provides a well-founded basis for the general mechanism of digital twin adaptation.

A DE algorithm consists of four basic steps, which include the initialization of the population, mutation, crossover, and selection. The initial vector population (individual trial solutions) is chosen randomly and should cover the entire parameter space. A new population is produced at the end of each iteration (or generation) by mutation and crossover operations on individual vectors [29]. In the selection stage, individuals of the new population are compared with individuals of the old population. The best individual based on the value of the objective function is selected as a competent individual of the population for the next generation. The termination of optimization is done based on a fixed number of iterations or by attaining a pre-specified objective function value, for example.

DE utilizes  $NP$  parameter vectors as a population for each generation  $G$ . Each vector is  $D$ -dimensional, where  $D$  is the number of parameters to be manipulated, i.e., the number of input function parameters in this case.  $NP$  is a user-specified number, such as  $10 \cdot D$ . The method has several variants for the mutation strategies and all of them are based on the mathematical subtraction operation with parameter vectors. In this study, the classical strategy of ‘DE/rand/1/bin’ was selected [34].

Other user-specified parameters include the crossover probability constant ( $CR$ ) from the interval  $[0, 1]$  and the factor that controls the amplification of differential variation ( $F$ ) from the interval  $[0, 2]$ . The values  $F = 0.8$  and  $CR = 0.5$  were tested in this study. The maximum number of iterations (itermax) and the value to reach (VTR) in the objective function were defined to reach the end of optimization. Practical advice on defining the parameter values is given in [34].

The original form of DE was modified here with the addition of a constraints check to minimize the calculation work. In the constraints check, the vectors of a new population were pre-checked and rejected if the trial solutions were outside the user-specified range, thus avoiding the generation of non-feasible solutions.

### 2.2.2 Objective function

The objective of model input adaptation is to fit the model output values to the reference values by adjusting the

model input values. An objective function that describes the problem well is necessary for successful optimization. In this study, the difference between the model output and reference values was minimized based on the mean relative error (MRE). The objective function  $J$  can therefore be defined as

$$J(\theta) = \left[ \frac{1}{N} \sum_{t=1}^N \left( \frac{|\hat{y}_t - y_t|}{|y_t|} \right) \right] \cdot 100\% \quad (1)$$

where  $\theta$  is the evaluated parameter subset,  $\hat{y}_t$  is the model output value,  $y_t$  is the reference value, and  $N$  is the number of time instants in the time series. In this study, the parameter subset includes only one value, because a single constant was estimated. The reference data were generated by the surrogate model using the input data generated by the Simulink<sup>®</sup> model.

In this study, the adapted variable was the torque from the propeller. The variable  $y$  used in the objective function was the torque from the flexible coupling (see Figure 2). However, the mechanism is not limited to the use of a single variable in the objective function owing to the global optimizer.

### 2.2.3 Surrogate modeling

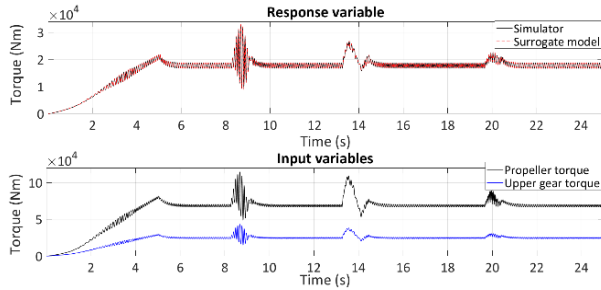
In this study, a surrogate model of the simulator was generated in order to speed up the adaptation procedure. The model is a linear regression model, which can be defined as

$$\hat{y} = a_1 \cdot x_1 + a_2 \cdot x_2 + a_3 \quad (2)$$

where  $x_1$  is the value of the torque from the propeller;  $x_2$  is the value of the torque from the upper gear;  $a_1$  and  $a_2$  are the regression coefficients and  $a_3$  is the intercept defined using least squares fitting [35]. The defined values are  $a_1 = -0.2798$ ,  $a_2 = 1.5025$ ,  $a_3 = 27.1046$ . The model response  $\hat{y}$  is the estimate of the torque from the flexible coupling ( $y$ ).

The Simulink<sup>®</sup> model was used to generate the data for the surrogate model fitting. The data and the surrogate model response ( $\hat{y}$ ) are shown in Figure 4. The simulated signals include an acceleration phase (0–5 s) and a constant phase, which is disturbed by unknown events around the time instants of 9, 14, and 20 s. The unknown events are simulated ice loads on the propeller, which were generated based on the definitions made by the classification society. The sampling frequency in this study was set to 1 kHz.

To evaluate the fitness of the surrogate model, the following values were computed based on the model response (period 1–25 s): MRE = 0.09%, RMSE = 35.45 Nm



**Figure 4:** The values of model variables including the estimates from the surrogate model.

(root-mean-square error), and  $R^2 = 1.00$  (coefficient of determination) [35]. The first second of the 25-second simulation was omitted, because the modeling error was relatively large at the start and would give a false impression of the overall performance. Otherwise, the surrogate model emulates the original simulator performance accurately in the presented operating conditions.

Practical modeling approaches require validation, such as cross-validation [36], but in this case the model was used to test the adaptation mechanism only. Model validation on separate data sets was therefore omitted. The main objective in the surrogate modeling was to develop as simple a model as possible for testing purposes only.

### 3 Results and discussion

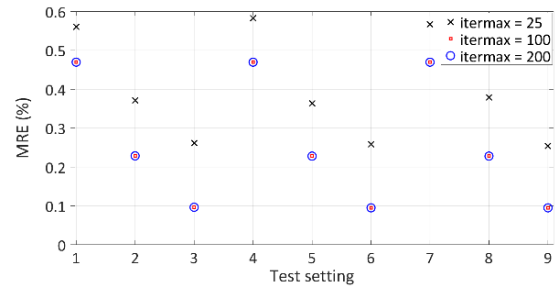
Nine different tests, presented in Table 1, were done using a varying maximum number for the iterations in DE,  $\text{itermax} = \{25, 100, 200\}$ . The tests included a varying value to reach (VTR) and time window length. The tested values were  $\text{VTR} = \{0.05\%, 0.01\%, 0.002\%\}$  and the number of time instants in a window were  $\{2, 5, 10\}$ . The estimated parameter value

**Table 1:** Test settings.

Setting number	Value to reach, VTR (%)	Time instants in window
1	0.05	10
2	0.05	5
3	0.05	2
4	0.01	10
5	0.01	5
6	0.01	2
7	0.002	10
8	0.002	5
9	0.002	2

(a.k.a. the value of the input variable,  $x_1$ ) was constrained in the range of  $-10000$ – $200000$  Nm in DE. The values outside this range were rejected from the new population. The computations were done using Matlab R2017a on Dell PowerEdge R720 with a 2.8 GHz 10-core-E5-2680v2 Xeon processor and 512 GB memory.

Figure 5 presents the MRE values for the adapted input variable (propeller torque) in each test. The adapted values were compared with the values generated by the simulator (see Figure 4). The MRE values were computed based on the data points from the period 1–25 s. The results indicate that the error decreased together with a decreasing window size. This is accounted for with the use of a constant value in each time window. The resolution is better in the shorter windows, resulting in smaller error values.

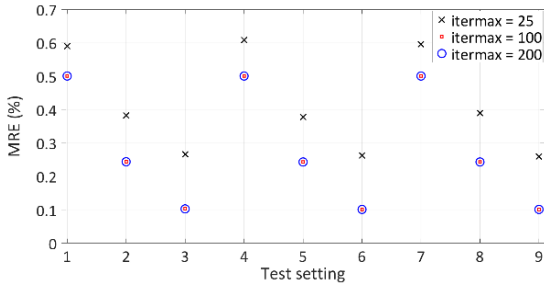


**Figure 5:** MRE of the adapted input variable with different test settings.

Interestingly, the VTR value did not have a major influence on the result. This can be inferred by comparing the MRE of test settings 3, 6, and 9, for example. This indicates that the error in the model output did not decrease after a certain limit. This is confirmed in Figure 6, which shows that the error in the model output did not decrease substantially when the VTR was reduced. The reason for the minor effect of VTR originates from the fact that DE reached the achievable optimum already at the highest tested VTR (0.05%). This indicates that the defined VTR could not be attained in every time window by estimating a constant in the input function.

The results in Figures 5 and 6 indicate that the optimization was completed typically when the maximum number of iterations was attained. The lowest number of iterations ( $\text{itermax} = 25$ ) was clearly too low, because more accurate estimates could be reached with a higher number of iterations. Then again, two hundred was too much, because the error did not decrease when compared with the case of one hundred iterations.

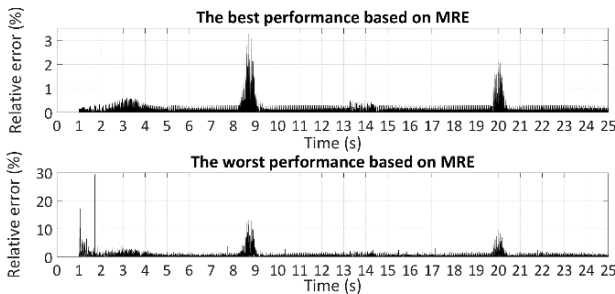
The upper plot in Figure 7 shows the relative errors for each time instant (1–25 s) from the case where the MRE was



**Figure 6:** MRE of the surrogate model output variable using the adapted input variable.

the smallest (0.09%) on the adapted variable (see Figure 5). One hundred iterations and test setting no. 9 were used. The results show that the highest error of 3.25% was obtained during the first ice load at the time instant of 8.7 s. Other simulation stages with relatively high errors include the acceleration stage, and the two other ice load instances. Obviously, a changing system state resulted in higher errors than the constant state. However, in general the relative errors were small.

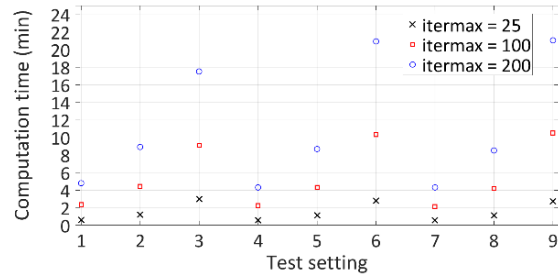
The lower plot in Figure 7 shows the relative errors for each time instant from the case where the MRE was the highest (0.58%) on the adapted variable (see Figure 5). Twenty-five iterations and test setting no. 4 were used. The maximum error (27.78%) was obtained during the acceleration phase at the time instant of 1.7 s. As in the upper plot, the highest errors were obtained during the ice load instances and the acceleration phase in general. A constant system state resulted in relatively small error values.



**Figure 7:** Best performance (above) and worst performance (below) on the adapted input variable.

Figure 8 presents the computation time of each test. The number of iterations and the number of time instants in a time window had a large effect on the computation time. The computation time increased together with the increase of itermax. In fact, the computation time was almost directly proportional to the number of iterations. One hundred iterations took roughly half of the time needed

for two hundred iterations and 25 iterations took close to a quarter of the time required for one hundred iterations. Moreover, the number of time instants (i.e., data points) in a window had a strong negative correlation with the computation time. Five time instants in a window took around double the time required for ten instants. Two instants took around 2.5 times the duration needed for five time instants. In conclusion, the computation time increased when smaller time windows were used.



**Figure 8:** Computation time with the applied settings.

In summary, the results show that relatively small errors were obtained in the model input by using the proposed adaptation mechanism. The presented case study was a simplified demonstration where the adaptation of a single constant in short time windows was tested. In order to obtain even more accurate estimates, more complicated input functions, such as linear or polynomial functions, could be used. The approach is not limited to using a single variable in adaptation. On the contrary, several variables can be optimized in more complex applications due to the suitability of the differential evolution algorithm for multi-parameter global optimization tasks [29].

Based on these results, the proposed adaptation mechanism has great potential for online adaptation in the future. The adaptation of the complete 25-second signal with the smallest error (setting no. 9, itermax = 100) took 630 seconds, which is around 25.2 times the duration of the signal. The fastest adaptation (setting no. 7, itermax = 25) took 34 seconds, which resulted in an MRE of 0.57% in the adapted input variable. Parallel computing was not applied in this study, but such a solution could potentially improve the real-time performance of the method. Moreover, the adaptation can be restricted solely to time instances with anomalous events if the typical performance of the physical object is simulated accurately enough at other time instances. The performance of the adaptation mechanism has a trade-off between the desired accuracy and computation time.

## 4 Conclusions

In this paper, the shortcomings of practical DT applications for industrial monitoring were studied, namely continuous and real-time data integration and model adaptation. A mechanism based on differential evolution was proposed for the online adaptation of digital twins with the aim of achieving real-time performance. The mechanism was demonstrated by estimating constants that represented model input variable values in short time windows. A surrogate model of the physics-based driveline model of a marine thruster was used to simplify the problem and to reduce the computational time. The results revealed that small errors were achieved in the model output and in the estimated input variable. However, the real-time requirement for the kilohertz rate was not achieved with the applied settings. Therefore, additional research should be done on parallel processing in order to reduce the computational time. Other possibilities include hardware-based speed improvements, algorithm optimization, and the use of more efficient programming techniques. Finally, the application of the proposed mechanism in more complex problems and other practical digital twin applications is recommended.

**Acknowledgement:** Our research work was supported by funding from Business Finland during the Reboot IoT Factory program. We would like to thank the personnel of Kongsberg Maritime Finland Oy (Rauma, Finland) for their collaboration and the Simulink® model.

## References

- [1] Negri E, Fumagalli L, Macchi M. A review of the roles of Digital Twin in CPS-based production systems. *Proc Manuf.* 2017;11:939-48.
- [2] Grieves M. Digital twin: manufacturing excellence through virtual factory replication, white paper. 2014 [cited 2020 Jun 11]. Available from: [https://www.researchgate.net/publication/275211047\\_Digital\\_Twin\\_Manufacturing\\_Excellence\\_through\\_Virtual\\_Factory\\_Replication](https://www.researchgate.net/publication/275211047_Digital_Twin_Manufacturing_Excellence_through_Virtual_Factory_Replication).
- [3] Shafto M, Conroy M, Doyle R, Glaessgen E, Kemp C, LeMoigne J, et al. DRAFT Modeling, simulation, information technology & processing roadmap. *Technology Area 11.* 2010 [cited 2020 Jun 11]. Available from: [https://www.nasa.gov/pdf/501321main\\_TA11-MSITP-DRAFT-Nov2010-A1.pdf](https://www.nasa.gov/pdf/501321main_TA11-MSITP-DRAFT-Nov2010-A1.pdf).
- [4] He Y, Guo J, Zheng X. From surveillance to digital twin: Challenges and recent advances of signal processing for the industrial Internet of Things. *IEEE Signal Proc Mag.* 2018;3:120-9.
- [5] Tao F, Zhang M, Liu Y, Nee AYC. Digital twin driven prognostics and health management for complex equipment. *CIRP Ann-Manuf Techn.* 2018;67:169-72.
- [6] Kritzinger W, Karner M, Traar G, Henjes J, Sihn W. Digital Twin in manufacturing: A categorical literature review and classification. *IFAC PapersOnLine.* 2018;51(11):1016-22.
- [7] Bao J, Guo D, Li J, Zhang J. The modelling and operations for the digital twin in the context of manufacturing. *Enterp Inf Syst.* 2019;13(4):534-56.
- [8] Yang J, Zhang W, Liu Y. Subcycle fatigue crack growth mechanism investigation for aluminum alloys and steel. *Proceedings of the 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference; 2013 Apr 8-11; Boston, Massachusetts, USA. AIAA;* 2013. p. 946-52.
- [9] Gockel BT, Tudor AW, Brandyberry MD, Penmetza RC, Tuegel EJ. Challenges with structural life forecasting using realistic mission profiles. *Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures Structural Dynamics and Materials Conference; 2012 Apr 23-26; Honolulu, Hawaii, USA. AIAA;* 2012. p. 7185-95.
- [10] Ayani M, Ganebäck M, Ng AHC. Digital Twin: applying emulation for machine reconditioning. *Proc CIRP.* 2018;72:243-8.
- [11] Gabor T, Belzner L, Kiermeier M, Beck MT, Neitz A. A simulation-based architecture for smart cyber-physical systems. *Proceedings of the 2016 IEEE International Conference on Autonomic Computing (ICAC); 2016 Jul 17-22; Wurzburg, Germany. IEEE;* 2016. p. 374-9.
- [12] Grieves M, Vickers J. Digital Twin: mitigating unpredictable, undesirable emergent behavior in complex systems. In: Kahlen FJ, Flumerfelt S, Alves A, editors. *Transdisciplinary perspectives on complex systems.* Cham: Springer; 2017. p. 85-113.
- [13] Lindström J, Larsson H, Jonsson M, Lejon E. Towards intelligent and sustainable production: combining and integrating online predictive maintenance and continuous quality control. *Proc CIRP.* 2017;63:443-8.
- [14] Weyer S, Harms C, Ruskowski M. Seamless integration of CPS-based factories across various engineering and simulation disciplines. *Proceedings of the 6th International Conference on Advanced Technology and Sciences; 2017 Sep 12-15; Riga, Latvia;* 2017. p. 137-43.
- [15] Uhlemann THJ, Lehmann C, Steinhilper R. The digital twin: realizing the cyber-physical production system for Industry 4.0. *Proc CIRP.* 2017;61:335-40.
- [16] Zhang H, Liu Q, Chen X, Zhang D, Leng J. A digital twin-based approach for designing and multi-objective optimization of hollow glass production line. *IEEE Access.* 2017;5:26901-11.
- [17] Stephens DW, Gorissen D, Crombecq K, Dhaene T. Surrogate based sensitivity analysis of process equipment. *Appl Math Model.* 2011;35:1676-87.
- [18] Cozad A, Sahinidis NV, Miller DC. Learning surrogate models for simulation-based optimisation. *AIChE J.* 2014;60:2211-27.
- [19] Golzari A, Sefat MH, Jamshidi S. Development of an adaptive surrogate model for production optimization. *J Petrol Sci Eng.* 2015;133:677-88.
- [20] Storn R, Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim.* 1997;11:341-59.
- [21] Ohenoja M, Ruusunen M, Koistinen A, Kaartinen A, Paaso J, Isokangas A, et al. Towards mineral beneficiation process chain intensification. *IFAC PapersOnLine.* 2018;51(21):201-6.
- [22] Gama J, Žliobaite I, Bifet A, Pechenizkiy M, Bouchachia A. A Survey on concept drift adaptation. *ACM Comput Surv.* 2014;46(4):1-37.

- [23] Kadlec P, Grbic R, Gabrys B. Review of adaptation mechanisms for data-driven soft sensors. *Comput Chem Eng.* 2011;35:1-24.
- [24] Maasoumy M, Moridian B, Razmara M, Shahbakhti M, Sangiovanni-Vincentelli A. Online simultaneous state estimation and parameter adaptation for building predictive control. *Proceedings of the ASME 2013 Dynamic Systems and Control Conference*; 2013 Oct 21-23; Palo Alto, USA. ASME; 2013.
- [25] Dougherty D, Cooper D. A practical multiple model adaptive strategy for single loop MPC. *Control Eng Pract.* 2003;11:141-59.
- [26] Schirmacher V, Schmiechen P, Knaak M, Ohata A, Kaneko S. An approach for physical model adaptation based on transient measurements. *Proceedings of the 2009 ICCAS-SICE*; 2009 Aug 18-21; Fukuoka, Japan. IEEE; 2009. p. 480-3.
- [27] Wang W, Zhao J, Xu Z. On parameter design for predictive control with adaptive disturbance model. *Proceedings of the 31st Chinese Control Conference*; 2012 Jul 25-27; Hefei, China. IEEE; 2012. p. 4160-5.
- [28] Ruusunen M, Leiviskä K. Fuzzy modelling of carbon dioxide in a burning process. *Control Eng Pract.* 2004;12:607-14.
- [29] Das S, Suganthan P. Differential evolution: a survey of the state-of-the-art. *IEEE T Evolut Comput.* 2011;15:4-31.
- [30] Keskitalo J, Leiviskä K. Application of evolutionary optimisers in data-based calibration of activated sludge models. *Expert Syst Appl.* 2012;39:6609-17.
- [31] Sun Z, Wang N, Bi Y, Srinivasan D. Parameter identification of PEMFC model based on hybrid adaptive differential evolution algorithm. *Energy.* 2015;90:1334-41.
- [32] Evangeline SJ, Kumar SS, Jayakumar J. Torque modeling of switched reluctance motor using LSSVM-DE. *Neurocomputing.* 2016;211:117-28.
- [33] Guedes JJ, Castoldi MF, Goedel A, Agulhari CM, Sanches DS. Parameters estimation of three-phase induction motors using differential evolution. *Electr Pow Syst Res.* 2018;154:204-12.
- [34] Differential evolution homepage [Internet]. Berkeley: International Computer Science Institute [cited 2020 Apr 25]. Available from: <http://www1.icsi.berkeley.edu/~storn/code.html>
- [35] James G, Witten D, Hastie T, Tibshirani R. An introduction to statistical learning with applications in R. New York: Springer-Verlag; 2013.
- [36] Picard RR, Cook RD. Cross-validation of regression models. *J Am Stat Assoc.* 1984;79:575-83.