Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 104 (2021) 618–623

www.elsevier.com/locate/procedia

54th CIRP Conference on Manufacturing Systems

Evidential Reasoning based Digital Twins for Performance Optimization of Complex Systems

Ananda Chakraborti[a,*], Arttu Heininen[a], Saara Väänänen[a], Kari T. Koskinen[a], Henri Vainio[a]

[a]*Automation Technology and Mechanical Engineering, Tampere University, 33720 Tampere, Finland*

* Corresponding author. Tel.: +358 413692130; *E-mail address:* ananda.chakraborti@tuni.fi

Abstract

Digital twins (DTs) are fast becoming an important technology in manufacturing companies for predicting failures of critical assets. However, such a digital twins is a hybrid representation with multiple parameters which need to be monitored to predict complex phenomena occurring in the asset in real time. This high-fidelity model of the twin makes the computation of the output extensive. Therefore, it is necessary to develop model reduction methods that simplify the high-fidelity model for faster computation with an acceptable degree of error. Such a method was proposed in previous studies to identify important nodes in graph-based DT representation. This article provides an improvement of previous method, considering the uncertainty in important node selection with Dempster-Shafer Theory (DST). The method is demonstrated with a grinding case study.

Keywords: Failure prediction, Dempster-Shafer Theory, Digital Twins, Artificial Intelligence

1. Introduction

Digital twins (DTs) are the talking point in manufacturing industry at the moment. Manufacturers, software developers and academicians all understand the concept of digital twins in different ways. Some define DTs as replica of a manufacturing asset that mimics its functionalities in real time [1] whereas, others define DTs as a bi-directional communication process between manufacturing machinery and their simulation models [2]. Efforts made by companies like GE (Predix), Siemens (MindSphere) and IBM (Watson analytics) has helped digital twin developers to have a process management platform-oriented mindset of digital twins. On the other hand, companies like ANSYS and PTC prefer a product-oriented analytical approach. In academic literature, DTs have been viewed as physical systems and their virtual equivalent with the system data threading them together [3]. In other cases, DTs have been conceptualized as a five-dimensional entity [4]. This three or five-dimensional viewpoint of DTs introduces a bigger systems integration challenge where it is not enough to put the physical and digital components together, but integrate bigdata, connections and services between them. Due to the versatility of definitions, opinions, utility and cost, DTs poses several challenging tasks to overcome in the manufacturing industry as well as academia [5–7].

Nomenclature

API	Application programming interface
BPA	Basic probability assignment
B_v	Behaviour model
CBM	Condition based maintenance
C_i	Centrality score obtained from respective algorithms
C_m	Minimum value of C_i
C_M	Maximum value of C_i
C_H	Cluster containing the high importance variables

10.1016/j.procir.2021.11.104

CN	DT connections
DD	DT data model
DAG	Directed Acyclical Graph
DACM	Dimension Analysis Conceptual Modeling
DTs	Digital Twins
DST	Dempster-Shafer Theory
EVC	Eigenvector centrality
G_v	Geometrical model
GA	Genetic Algorithm
h	High importance elements
IIoT	Industrial internet of things
l	Low importance elements
P_v	Physics-based model
PE	Physical entity
PHM	Prognostics and health management
PR	PageRank algorithm
R_v	Rule-based model
RUL	Remaining useful life
RTU	Remote terminal unit
SS	DT Services
Ω	Frame of discernment
VE	Virtual entity
X_H	Matrix containing set of high importance variables
X_L	Matrix containing set of low importance variables

2. Case Study: Grinding Digital Twin

One such important challenge is the trade-off between model fidelity and computational speed. High fidelity analytical models require several hours even days to solve for quantities of interest even with powerful processors. This raises the question of real-time (or near real-time) predictions made by the twin. Adding to that are questions regarding the latency of data, availability of network and internet speed. Hence, some authors have proposed that a very high-fidelity model which is computationally intensive should not be the goal while building a responsive digital twins for process optimization [8]. Rather, focusing on a subset of important parameters that explain the behavior of the physical system and predict its future state is more valuable. However, determining these selected few parameters that explains the majority of impact on the target parameters and in turn the final outcome of the system is not a trivial task.

For this purpose, a methodology was proposed previously which realizes DTs as complex graphs. The methodology describes a graph-based system identification and model reduction technique to locate the important parameters and their optimization [9]. The methodology was applied to a grinding wheel wear case study. Fig. 1 shows the conceptual DT of the grinding system based on framework proposed by other authors [10]. This is used as a reference model. The PE in Fig. 1 is the physical grinding machine with sensors, actuators and RTUs. The VE representation is the DT model. However, the VE is not a single model. It is a collection of models such as geometrical, physics-based, rule-based and behavior model. Digital service model defines the prognostics and health management services provided by the twin for the grinding wheel. The Data model provides detailed description of the data and its source that is generated by the PE and consumed by the VE models. Connections model describe the API endpoints where the data from various components of PE is posted to on-premise server or to the cloud. It should also describe the connectivity of the simulation models with specific data sources.

The VE section of Fig. 1 works as the platform for the complex graph representation of the DT. At this stage, the functional variables influencing different dimensions of the digital twins are identified and associated with respective models of VE. Then, a graph-based conceptual modeling mechanism known as dimensional analysis conceptual modeling (DACM) [11] was used to represent the virtual entity in the form of a causal graph which is directed and acyclic in nature. The model reduction method was implemented on this causal graph in two stages; (1) spectral clustering method with normalized graph Laplacian to fit the directed acyclical graph (DAG), and (2) identifying important nodes (variables) in the graph with graph centrality metrics based on eigenvector methods. However, on further investigation, it was found that different graph centrality metrics that use eigenvector methods do not yield the same results. Hence, an uncertainty is induced whether a variable is important or not. Also, the DT graph contains exogenous variables. Their selection as important nodes by the algorithm also induces uncertainty regarding the outcome of the model reduction method.

Therefore, in this article, a new node importance evaluation method is introduced with the help of Dempster-Shafer theory (DST) or evidential reasoning which takes the uncertainties into account while selecting the important nodes. DST is a well-known data fusion method which has been widely used in predicting failure and decision making under uncertainty [12]. As a result, a python package is developed for selecting the important nodes in a graph-based representation of grinding DTs with evidential reasoning for optimization of performance metrics.

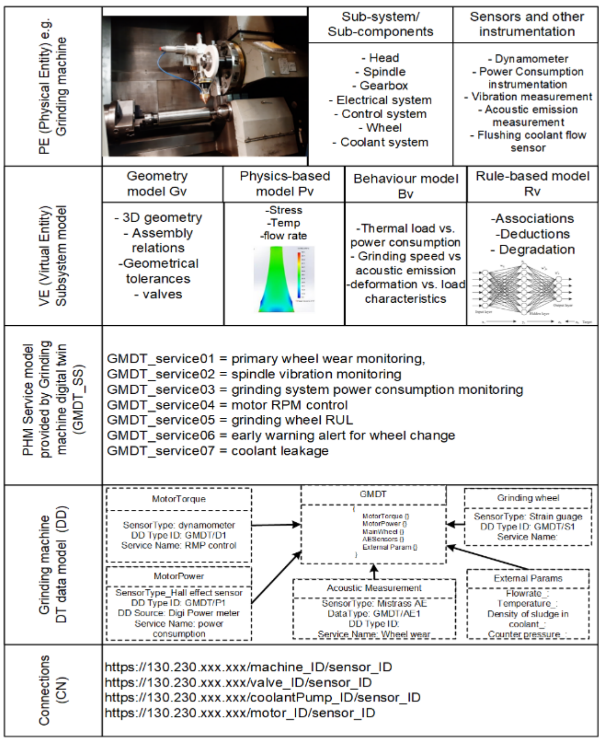

Fig. 1. DT reference model for grinding wheel wear

This article is organized as follows. Section 3 presents the node importance identification under uncertainty with the help of DST. Section 4 proposes an improved model reduction method based on evidential reasoning. Section 5 introduces a python package for implementing the model reduction on graph-based DTs and finally, section 6 concludes the article and proposes the future work in this direction. (Please note that the terms graph and network are used interchangeably in this article).

3. Node Importance

Identification of important nodes in a graph is an active field of research in artificial intelligence. Researchers have applied different graph algorithms to figure out which are central nodes in the graph and how sensitive are these nodes to the objective variables. Graph centrality is a diverse topic in network theory with several algorithms available to study different network phenomena in complex graphical systems such as; finding the shortest path from a given node to the target node, predicting links between nodes, understanding the relative importance of nodes in a network and finding bridge nodes to detect communities or clusters to name a few. Several experimental studies have proven the effectiveness of such algorithms in complex systems [13,14]. However, because of this diversity of graph algorithms and their specific area of application, the context in which a centrality metric is used becomes critical.

In this section, DT described in the previous section is viewed as a complex graphical representation of the grinding system with the target of monitoring and predicting wear in the grinding wheel. Firstly, such a DAG is clustered with the help of unsupervised learning techniques such as spectral clustering to figure out the similarity between the nodes or find out those nodes that stay together when the graph is partitioned. When spectral clustering techniques are used in conjunction with the graph centrality algorithm such as PageRank, cluster hierarchy could be determined according to their impact on the target nodes. PageRank is a class of eigenvector centrality measure. There are other eigenvector centrality measures which takes the same eigenvector approach as PageRank. However, upon further investigation, it was found that the similar ranking algorithms to compute node centrality do not agree with each other for the grinding system graph. One interesting aspect to mention here is that the algorithms for undirected and directed graphs are different. This is because a directed graph does not have a symmetrical adjacency matrix. Hence, directed graphs such as the DT graph have to be normalized before application of clustering and node importance algorithms.

The DT graph is clustered with the spectral clustering method. Then three methods of centrality for directed graphs are applied on it which are (1) EVC, (2) Katz centrality and (3) PageRank algorithm. Though the three methods fall under the class of eigenvector method where the relative importance of a node depends on the importance of its neighbors and the degree distribution in the network, the ranking order of the nodes do not agree with each other completely. This difference in ranking order shown in Fig. 2. This is problematic because a definite ranking system cannot be followed and depending on the selection of the method there will be a recommendation to optimize completely different sets of nodes. This affects the end result of the system. Hence, there is a need to tackle this uncertainty in the node ranks. That is why DST is applied to combine the results from different centrality metrics based on the evidence that a node is important or not.

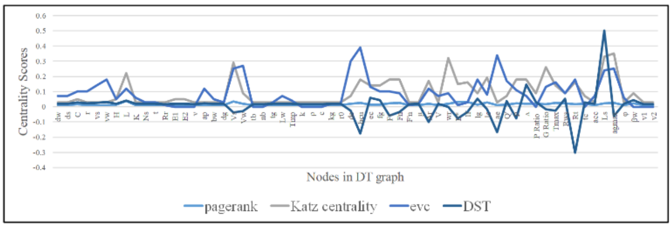

Fig. 2. Comparison of node importance by centrality methods and DST

3.1. Dempster-Shafer Theory (DST)

The area of condition-based monitoring and intelligent failure detection, targeted towards reasoning and decision making under uncertainty, can be broadly classified under two frameworks: (1) those implementing Bayes theory, and (2) DST or evidence theory-based frameworks. Bayesian methods are widely used techniques based on conditional probability to reason under uncertainty [15]. Bayesian networks provide a mechanism to probabilistically infer the likelihood of an event occurring. However, a major criticism of Bayesian method is that it cannot handle ignorance, incomplete or imprecise information. In the node importance scenario described above, the absence of experimental data of the variable in the DT graph affect the determination of their conditional probabilities which makes Bayesian methods inconsistent. In data fusion, effective results can be obtained with Bayesian methods only if the adequate and appropriate priori and conditional probabilities are available. In contrast, DST is a generalization of Bayesian theory of subjective probability, used to combine information obtained from multiple sources. In DST, 'belief' is assigned to a set of elements rather than assigning probabilities to individual variables in the graph. The concept of belief is not the same as chance and can be updated based on evidence obtained about the elements. [16] provides a comparative analysis between Bayesian methods and evidence theory for failure diagnosis in knowledge-based systems. [17] provides a tutorial on DST for online diagnostics of engines based on information obtained from multiple sensors such as accelerometers and acoustic emission sensors.

In this section, DST is used to combine the information available regarding the nodes in DT graph and their relative importance obtained from the node importance scores described in section 2. There are two possible outcomes for each node. The nodes can be high importance (h) or low importance (l). Hence, the frame of discernment (which is a non-empty set containing all mutually exclusive and exhaustive elements) is defined as: $\Omega = \{h, l\}$ and the power set (which is a set of all possible combinations of the problem in the frame of discernment) is defined as: $\{h, l, \varnothing\}$. Next, the mass functions are determined by adopting a technique similar to the one described in [18] for directed networks. The maximum and minimum values of the corresponding ranking is used to compute the mass functions with the following formulae:

$$m_{C(i)}(h) = \frac{C_i - C_m}{C_M - C_m + \omega} \tag{1.1}$$

$$m_{C(i)}(l) = \frac{C_i - C_M}{C_M - C_m + \omega} \tag{1.2}$$

$$m_{C(i)}(\varnothing) = 1 - m_{C(i)}(h) - m_{C(i)}(l) \tag{1.3}$$

ω is a tunable parameter which is chosen to avoid the denominator becoming zero. Repeating the steps in equation 1.1-1.3 creates basic probability assignment (BPA) for each node in the form:

$$M_C(i) = \{m_{C(i)}(h), m_{C(i)}(l), m_{C(i)}(\varnothing)\} \tag{1.4}$$

As there are 62 nodes in the original graph, 62 BPA sets were obtained. Now, all node importance scores obtained from different centrality metrics can be combined with the help of Dempster's combination rule [19] to generate a new combined ranking of the nodes. Dempster's combination rule (rule of evidence combination) is modified to obtain the new metric for node based on the evidence whether the node is high importance or low importance:

$$m_i(h) = \frac{1}{1-k} \sum_{C(i)=h} m_{C(i)}(h) \tag{1.5}$$

$$m_i(l) = \frac{1}{1-k} \sum_{C(i)=l} m_{C(i)}(l) \tag{1.6}$$

$$\text{Where, } k = \sum_{C(i)=\varnothing} m_{C(i)}(\varnothing) \tag{1.7}$$

The factor k is a normalization constant known as conflict coefficient of two BPAs. Higher the value of k, more conflicting are the sources of evidence and lesser information they combine. Finally, the combined scores of each node based on evidential reasoning is obtained as:

$$M_{evidential}(i) = m_i(h) - m_i(l) \tag{1.8}$$

The result of the evidential reasoning-based score is shown in Fig. 2. From the figure, it is found that node importance score based on evidential reasoning aggregates the scores provided by other centrality techniques. Those nodes are ranked lower, which have bigger disagreement amongst the centrality metrics. This indicates the presence of a high degree of uncertainty in those nodes to be the important node. On the other hand, the nodes which all centrality metrics have ranked higher with little disagreement has a higher score from DST. Thus, assuming the graph-based representation of the grinding system wear prediction and monitoring is complete, a hierarchy of the nodes in that graph is obtained considering the uncertainty in that ranking system. Hence, those high importance or high impact nodes can be obtained that contribute significantly to the target variables of the grinding system such as V, V_s and V_w in the DT graph.

4. Model reduction

The digital twins are living hybrid model. It is a combination of IIoT data with advanced physics-based or system level simulation models. A little consideration will show that such a model of machinery is a high-fidelity model with a large number of parameters needing real-time or near real-time optimization. Hence, a model reduction method is highly desirable that simplifies the computational challenge and focuses on optimizing those variables that have a higher impact on the target variables and in turn the final outcome of the model. In machine learning literature, conventional methods for model reduction could be found such as singular value decomposition and principal component analysis. However, there is a need for development of new methods for reducing the graphical DT models, that limits the number of nodes in the graph to high importance nodes, especially when imprecise or no information about node values is available. In this section, such a model reduction methodology is proposed based on network theory algorithms for node importance obtained from the previous section.

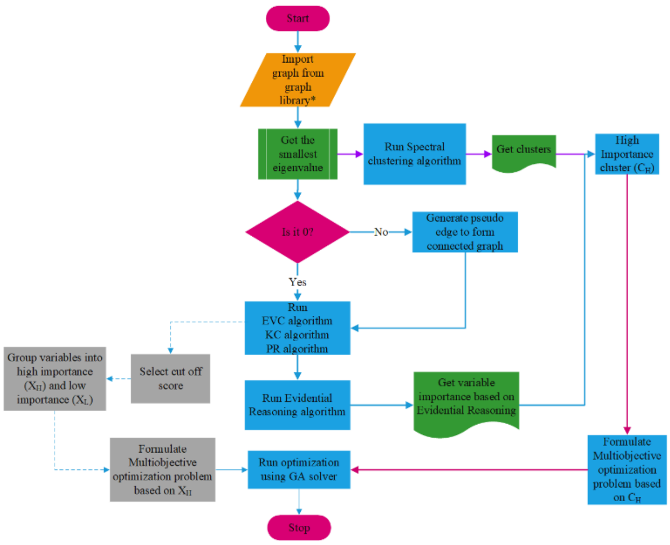

Fig. 3. The Model Reduction Method

The model reduction method is shown in Fig. 3. This model reduction method should be read in conjunction with the model reduction method proposed in [20]. Previously, such a method was proposed only based on metrics that could screen the parameters into high importance or low importance. However, the validity of such a method was challenged when it was used on complex graphs such as the DT graph. The new method has three steps.

1. *Generating the DAG*: The model reduction method can work on DT of any machinery provided a DAG representation of it exists. In the grinding DT case, the DAG is generated with DACM. It is possible to generate a graph library for different machinery for this purpose. The graph is generated from the VE representation of specific physics-based phenomenon. Hence, a graph library item can be created for phenomenon such as grinding wheel wear. This is shown in the orange box of Fig. 3. * sign in the box

indicates that such a library is under development and not available during the time of writing this article.

2. *Application of graph algorithms to locate important nodes*: After the graph is built and imported into the python environment, a test is run to check if the imported graph is a fully connected graph. This is important because if there are discontinuities in the graph, the clustering and centrality measures will not yield proper results. Then the spectral clustering and centrality metrics computation are done parallelly. In Fig. 3, this is indicated by purple arrows for spectral clustering and blue arrows for centrality algorithms. The spectral clustering algorithm is implemented to generate the clusters in the DT graph. In the other direction, three different eigenvector centrality methods are applied to the graph to generate the importance scores. These scores are evidentially combined with the help of DST as mentioned in section 2.1. When the output from the DST and spectral clustering are combined, those clusters of nodes are obtained which contains the high importance nodes known as C_H or high importance cluster.
3. *Formulation of the optimization problem*: In the final stage, a multi-objective optimization problem is formulated to test and validate the model reduction method. This is indicated by red arrows in Fig. 3. Previously, a threshold score was used to classify whether a node is important or not. Then two matrices were generated X_H and X_L which contained the high and low importance nodes respectively. The optimization problem was formulated with variables in X_H. This is indicated by the grey boxes in Fig. 3. This system works well when there is one centrality metric available that will yield a perfect result. Also, the selection of the threshold was done based on the data and justification of selecting the threshold was weak. The new method applies evidential reasoning-based node importance selection and combines the result with the spectral clustering output. This method bypasses the need to select any arbitrary threshold as spectral methods groups similar nodes together. This means if a high importance node is located in the cluster it is likely that the other nodes are high importance as well as similar nodes were grouped together by the clustering algorithm. Hence, a cluster can be found C_H, that contains the maximum number of high importance variables. This is defined as the most important cluster and the variables it contains will have maximum impact on the target when optimized.

Thus, optimizing the variables in C_H has the largest impact on the target variables. This model reduction methodology is a fast way to determine the important variables. Also, this method is generalizable. When a graph library of phenomena exist (phenomena model), such as the grinding wheel wear phenomenon mentioned above, important nodes can be determined from it. The obvious disadvantage of such method is its accuracy is low because some variables are consciously omitted from the final set of variables that is optimized. However, the utility of this method lies in finding the important nodes quickly with a reasonable degree of error. In the grinding DT, the model reduction method found the important nodes that were most sensitive to the change in grinding ratio with less than 5% error [6].

The applications of this method can be (1) selecting and optimizing performance indicators for CBM of complex machine systems, (2) a tool for maintenance engineers to quickly locate most probable failure zones with parameters most likely to result in a failure, and (3) resource optimization in monitoring complex systems.

5. A Python package for model reduction

A python package is developed for computing the important nodes in the DT graph with the help of evidential reasoning method. This package can be readily imported by machine designers, manufacturing, and maintenance engineers to run a check for the important nodes. The package uses standard libraries and dependencies which are easy to implement. This package contains following modules:

- *graph.py*: This module generates the graph of the PE with 'Networkx' python library for directed graphs. In the grinding case, the graph is developed and imported manually. But as mentioned in the previous section, this module is under development. This module can be expanded to a library of items itself, containing a graph-based representation of the VE of any machinery desired by the user.

- *spectral.py*: This module spectrally clusters the imported graph. It implements several functions and dependencies for generating the graph Laplacian, calculating eigenvalues and eigenvectors, grouping the nodes into clusters (C_1, C_2, C_3, ..., C_n) and using *k*-means to create the clusters. This module uses popular python packages such as 'sklearn.cluster' using methods such as 'SpectralClustering' and 'kMeans' to generate the clusters.

- *central.py*: This module contains submodules for calculating different centrality measures. This module works parallelly to the spectral.py module generating the node importance scores irrespective of the clustering details. The submodules independently compute different centrality scores using 'Networkx' and 'NumPy' libraries.

- *evidence.py*: This module imports the importance scores from central.py and combines the scores with the help of DST to generate a new set of ranking for the nodes. This module uses a prebuilt 'pyds' library for performing DST calculations. 'pyds' library provides methods to build the mass functions and powerset with 'MassFunction' and 'powerset' modules for all the nodes based on equations 1.1-1.3 as mentioned in section 2.1.

- *final.py*: This module combines the results obtained from evidence.py and spectral.py in order to obtain C_H and other clusters.

The following process of building the multi-objective optimization problem is not a part of this package however, in the future this could be integrated into this package as well. There is also a need to connect this model reduction method to the PE and data obtained from continuous measurement from the grinding wheel as shown in Fig 1. Hence, a database plug-in functionality will be developed in the future so that the model reduction method can be integrated with measurement data or any other framework that analyzes measurement data.

6. Conclusion and Future Work

In this article, a methodology is presented to reduce complex VE models of graphical DT representation. Previously, a model reduction method of graph-based representation of complex systems was demonstrated with the help of spectral methods and centrality measures. It was found that the method was not optimal, and the reduced model was dependent on the choice of centrality method. Therefore, an evidential reasoning approach is undertaken with the help of DST to combine the results from centrality metrics and generate a ranking of node importance considering the uncertainty in selecting an important node. Then the spectral method and the evidential method were combined to obtain a subgraph which explains the majority of impact on the outcome of the model with reasonable accuracy. A python package was developed to combine the steps in the model reduction method. This package provides a readymade solution for engineers and managers in small and medium scale industries who are building digital twins for complex machines and facing challenges with monitoring and optimizing a large number of parameters provided by high-fidelity simulation models. Some functionalities of this package are under development. In the future, it will be possible to import graph-based representation of the entire machine system and select the important nodes that explains the majority of impact on the output. The performance of complex machine systems can be optimized by tuning these important parameters.

Acknowledgements

The support of ÄVE-project and Business Finland in making this research possible is greatly acknowledged.

References

[1] Söderberg R. Toward a Digital Twin for real-time geometry assurance in individualized production. R So 2017:4.

[2] Tao F. Digital twin-driven product design, manufacturing and service with big data. Int J Adv Manuf Technol 2018:14.

[3] Grieves M, Vickers J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In: Kahlen F-J, Flumerfelt S, Alves A, editors. Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches, Cham: Springer International Publishing; 2017, p. 85–113. https://doi.org/10.1007/978-3-319-38756-7_4.

[4] Qi Q. Enabling technologies and tools for digital twin n.d.:19.

[5] Qi Q, Tao F, Zuo Y, Zhao D. Digital Twin Service towards Smart Manufacturing. Procedia CIRP 2018;72:237–42. https://doi.org/10.1016/j.procir.2018.03.103.

[6] Samir K, Maffei A, Onori MA. Real-Time asset tracking; a starting point for Digital Twin implementation in Manufacturing. Procedia CIRP 2019;81:719–23. https://doi.org/10.1016/j.procir.2019.03.182.

[7] Monostori L. Cyber-physical Production Systems: Roots, Expectations and R&D Challenges. Procedia CIRP 2014;17:9–13. https://doi.org/10.1016/j.procir.2014.03.115.

[8] Hartmann D, Herz M, Wever U. Model Order Reduction a Key Technology for Digital Twins. In: Keiper W, Milde A, Volkwein S, editors. Reduced-Order Modeling (ROM) for Simulation and Optimization: Powerful Algorithms as Key Enablers for Scientific Computing, Cham: Springer International Publishing; 2018, p. 167–79. https://doi.org/10.1007/978-3-319-75319-5_8.

[9] Chakraborti A, Heininen A, Koskinen KT, Lämsä V. Digital Twin: Multi-dimensional Model Reduction Method for Performance Optimization of the Virtual Entity. Procedia CIRP 2020:6.

[10] Tao F, Zhang H, Liu A, Nee AYC. Digital Twin in Industry: State-of-the-Art. IEEE Trans Ind Inf 2019;15:2405–15. https://doi.org/10.1109/TII.2018.2873186.

[11] Coatanea E, Roca R, Mokhtarian H, Mokammel F, Ikkala K. A Conceptual Modeling and Simulation Framework for System Design. Comput Sci Eng 2016;18:42–52. https://doi.org/10.1109/MCSE.2016.75.

[12] Yang B-S, Kim KJ. Application of Dempster–Shafer theory in fault diagnosis of induction motors using vibration and current signals. Mechanical Systems and Signal Processing 2006;20:403–20. https://doi.org/10.1016/j.ymssp.2004.10.010.

[13] Zhang WY, Zhang S, Guo SS. A PageRank-based reputation model for personalised manufacturing service recommendation. Enterprise Information Systems 2017;11:672–93. https://doi.org/10.1080/17517575.2015.1077998.

[14] You K, Tempo R, Qiu L. Distributed Algorithms for Computation of Centrality Measures in Complex Networks. IEEE Trans Automat Contr 2017;62:2080–94. https://doi.org/10.1109/TAC.2016.2604373.

[15] Panicker S, Nagarajan HPN, Mokhtarian H, Hamedi A, Chakraborti A, Coatanéa E, et al. Tracing the Interrelationship between Key Performance Indicators and Production Cost using Bayesian Networks. Procedia CIRP 2019;81:500–5. https://doi.org/10.1016/j.procir.2019.03.136.

[16] Verbert K, Babuška R, De Schutter B. Bayesian and Dempster–Shafer reasoning for knowledge-based fault diagnosis–A comparative study. Engineering Applications of Artificial Intelligence 2017;60:136–50. https://doi.org/10.1016/j.engappai.2017.01.011.

[17] Basir O, Yuan X. Engine fault diagnosis based on multi-sensor information fusion using Dempster–Shafer evidence theory q. Information Fusion 2007:8.

[18] Mo H, Deng Y. Identifying node importance based on evidence theory in complex networks. Physica A: Statistical Mechanics and Its Applications 2019;529:121538. https://doi.org/10.1016/j.physa.2019.121538.

[19] Bappy MM, Ali SM, Kabir G, Paul SK. Supply chain sustainability assessment with Dempster-Shafer evidence theory: Implications in cleaner production. Journal of Cleaner Production 2019;237:117771. https://doi.org/10.1016/j.jclepro.2019.117771.

[20] Chakraborti A, Nagarajan HPN, Panicker S, Mokhtarian H, Coatanéa E, Koskinen KT. A Dimension Reduction Method for Efficient Optimization of Manufacturing Performance. Procedia Manufacturing 2019;38:556–63. https://doi.org/10.1016/j.promfg.2020.01.070.