Tampere University

ANANDA S. CHAKRABORTI

Graph-based Model Reduction of Machine System Digital Twins

Tampere University Dissertations 1022

Tampere University Dissertations 1022

ANANDA S. CHAKRABORTI

Graph-based Model Reduction of Machine System Digital Twins

ACADEMIC DISSERTATION
To be presented, with the permission of
the Faculty of Engineering and Natural Sciences
of Tampere University,
for public discussion in the auditorium K1702
of the Konetalo building, Korkeakoulunkatu 6, Tampere,
on 3 May 2024, at 12 o'clock.

ACADEMIC DISSERTATION
Tampere University, Faculty of Engineering and Natural Sciences
Finland

Responsible supervisor and Custos	Professor Kari T. Koskinen Tampere University Finland	
Pre-examiners	Professor Gaetano Cascini Politecnico di Milano Italy	Professor Adil Rasheed Norwegian University of Science and Technology Norway
Opponents	Professor Adil Rasheed Norwegian University of Science and Technology Norway	Professor Tommi Mikkonen University of Jyväskylä Finland

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

Cover design: Roihu Inc.

ISBN 978-952-03-3439-0 (print)
ISBN 978-952-03-3440-6 (pdf)
ISSN 2489-9860 (print)
ISSN 2490-0028 (pdf)
http://urn.fi/URN:ISBN:978-952-03-3440-6

HIILINEUTRAALI PAINOTUOTE
ClimateCalc CC-000025/FI
PunaMusta Printing

Carbon dioxide emissions from printing Tampere University dissertations have been compensated.

PunaMusta Oy – Yliopistopaino
Joensuu 2024

PREFACE

The research presented in this thesis was carried out at the Hervanta campus at Tampere University (known as the Tampere University of Technology when the thesis first began) from 2018—2021. The research was principally funded by the TTITO doctoral school grant at Tampere University and partly funded within the scope of the Addendum: FCC-GOV-CC-01 27/EDMSt84557t between CERN and TAU.

I want to express my greatest gratitude to my supervisor Professor Kari T. Koskinen. I am truly thankful for the opportunity that he gave me to do my doctoral studies after my M.Sc. in Factory Automation and Industrial Informatics. I would also like to thank the pre-examiners, Prof. Adil Rasheed and Prof. Gaetano Cascini for their valuable inputs and suggestions for improving the quality of the thesis. Thank you for the feedback and recommendations throughout this thesis and all the inspiring discussions and support. I would also like to thank my colleagues from the Mechatronics Research Group (MRG) for their unconditional support throughout this research, especially Henri Vainio, Saara Salander and Arttu Heininen for their help and support. Furthermore, I would like to thank Prof. Eric Coatanea and his research group and Prof. Inigo Flores Ituarte for their support during the development and implementation of this research work.

I am grateful to have had the support of all my friends and colleagues in Tampere, who made living and studying in Tampere an enjoyable experience. Lastly, I would like to thank my mother Dr. Mridula Chakravarty and my wife Aneesa for being patient with me during this journey of my life and our little son, Ahaan.

ABSTRACT

Digital Twin technology is the talking point of academia and industry. When defining a digital twin, new modeling paradigms and computational methods are needed. Developments in the Internet of Things, Artificial Intelligence and advanced simulation and modeling techniques have provided new strategies for building such complex digital twins. The digital twin is a virtual entity representation of the physical entity, such as a product or a process. This virtual entity is a collection of computationally complex knowledge models that embeds all the information of the physical world. This definition of the virtual entity representation makes it immensely complex with several dependencies across various domains of the real-world object. This virtual entity must meaningfully represent information and deductions from these models.

To that end, this research work presents a graph-based representation of the virtual entity. This graph-based representation provides a method to build a knowledge model that embeds the interactions between several parameters across different modeling domains. For that, both traditional and newer methods of graph-based modeling of multivariate systems are researched and usability of these methods are identified. Thereafter, a new method for digital twin conceptualization with graph-based method is proposed by combining conceptual modeling mechanism known as dimensional analysis conceptual modeling and heuristic methods such as greedy equivalence search. Hence, the virtual entity could be represented as a directed graph. However, such virtual entity graph becomes inherently complex with multiple parameters for a complex multidimensional physical system. This research contributes to the body of knowledge with a novel Graph-based Model Reduction method that simplifies the virtual entity graph by preserving the important parameters in it. The graph-based model reduction method uses spectral decomposition method to segment the knowledge graph into structurally similar chunks. Then, investigation is performed to identify the important nodes of the knowledge graph with node importance algorithms such as weighted PageRank and eigenvector centrality. To consolidate the ranking scores, algorithms from the domain of artificial intelligence such as Dempster-Shaffer theory is applied.

The Graph-based Model Reduction method is validated with two case studies of complex machine systems: (1) grinding wheel wear digital twin and (2) turbo compressor digital twin. In both these case studies, the graph-based modeling method combines information from the physics-based models such as finite element models and system level simulation models, with data-driven models to create a hybrid graphical representation. Then, the graph-based model reduction method is applied on the hybrid graphical representation. The method is benchmarked against other model reduction methods in literature and the results are analysed.

This thesis work provides a detailed analysis and results of the graph-based modeling and model reduction methods of digital twins of complex systems. It is argued that the important area of model reduction has been overlooked by the digital twin community. The thesis work shares the learnings from application of the graph-based modeling and model reduction methods in traditional machine systems. This work promotes the application and integration of graph based methods in digital twin frameworks and software solutions for building efficient digital twins that provide efficient digital services.

Index terms:

Digital Twin; Graph-based modeling, Knowledge Representation; Model Fusion; Model Reduction; Network Theory, Importance Measurement.

ABBREVIATIONS

AI	Artificial Intelligence
AM	Additive Manufacturing
AMB	Active Magnetic Bearing
API	Application Programming Interface
ANN	Artificial Neural Network
BPA	Basic Probability Assignment
BES	Backward Equivalence Search
BN	Bayesian Network
CFD	Computational Fluid Dynamics
CPS	Cyber physical system
CRNN	Convolution Recurrent Neural Network
DACM	Dimensional Analysis Conceptual Modeling
DAG	Directed Acyclical Graph
DES	Discrete Event Simulation
DST	Dempster-Shafer Theory
DT	Digital Twin
EVC	Eigenvector Centrality
FEA	Finite Element Analysis
FEM	Finite Element Method
FES	Forward Equivalence Search
GBMR	Graph Based Model Reduction
IP	Internet protocol
IPv6	Internet protocol version 6
IoT	Internet of Things
IIoT	Industrial Internet of Things
M2M	Machine to machine
M2S	Machine to system
MDO	Multidisciplinary design optimization
MOR	Model Order Reduction
MORA	Model Order Reduction Algorithm
MR	Model Reduction
NN	Neural Network
PCA	Principal Component Analysis
PE	Physical Entity
PHM	Prognostics and Health Management
PR	Page Rank algorithm
RF	Random Forest
RFR	Random Forest Regressor
S&M	Simulation and Modeling
SVD	Singular Value Decomposition

UML	Unified Modeling Language
WPR	Weighted PageRank
VE	Virtual Entity

NOMENCLATURE

d_w	diameter of work piece
d_s	diameter of grinding wheel
C	active grit density (#/mm^2)
r	grit shape factor
v_s	wheel speed (m/s)
v_w	workpiece speed (m/s)
H	hardness of grinding grain
L_s	sliding distance
K	Archard's constant
N_s	rotational wheel speed
t	total grinding contact time
R_r	roughness factor
E_1	elastic Modulus of Wheel
E_2	elastic Modulus of workpiece
v	Poisson's Ratio
a_p	programmed or set depth of cut
b_w	width of grind
d_g	mean grain size
V_s	volume of wheel worn per unit wheel width
V_w	volume of material ground per unit wheel width
t_b	total time
n_b	number of parts
t_g	grinding time
L_w	grinding distance
T_mp	temperature approaching the melting point
k	workpiece thermal conductivity
ϱ	density
c	workpiece heat capacity
k_g	grain thermal conductivity
r_0	grain contact radius
d_e	equivalent wheel diameter
h_cu	uncut chip thickness
e_c	specific grinding energy
f_g	force per grit
F_t	tangential G. Force
F_n	normal G. Force
F'_n	normal G. Force per unit width
μ	coefficient of grinding
A_r	real area of contact
V	volume of wear of the wheel
w_r	wear life cycle
E*	combined elastic Modulus
l_f	deflection contact length
l_g	geometrical contact length
l_c	real contact length
a_e	actual depth of cut
Q'	specific removal rate
P	grinding power
ʌ	stock removal parameter
PRatio	volume of metal ground per unit area of wheel surface
GRatio	volume of material ground per unit wheel width by volume of wheel worn per unit wheel width
T_max	maximum temperature
R_ws	workpiece partition ratio
Rt	roughness
t_c	cycle time
y	non-dimensional pressure rise
ϕ	non-dimensional mass flow rate
Dp	dimensional pressure rise
m	dimensional mass flow rate
r_{01}	density @ inlet condition (ambient)
U	impeller tip speed
A_c	area of the compressor duct
L_c	length of compressor duct
V_p	volume of plenum
a_{01}	speed of sound in @ inlet (ambient) condition
f_c	non-dimensional mass flow rate
f_{th}	throttle mass flow rate
y_p	plenum pressure rise
y_c	compressor pressure rise
t/c	time constant of the compressor
L_{th}	length of throttle duct

A_{th}	cross-sectional area of throttle duct
u_{th}	throttle percentage opening
c_{th}	constant, determined experimentally and depends on valve geometry and properties of the fluid
T_{01}	stagnation temperature
p_{01}	stagnation pressure
C_p	specific heat at constant pressure
$\Delta h_{oc,ideal}$	total specific enthalpy delivered to the fluid
γ	specific heat ratio
$p_{c,ss}$	compressor output pressure at nominal tip clearance $\delta_{cl} = 0$
q_A	predicted cross-coupling stiffness
HP	rated horsepower
B_C	constant for centrifugal compressors determined experimentally
C^*	constant for centrifugal compressors experimentally
ρ_d	discharge gas density per impeller/stage
ρ_s	suction gas density per impeller/stage
D_C	impeller diameter
H_C	minimum of diffuser or impeller discharge width
N	operating speed
V	number of nodes in the algebraic
E	number of edges in an algebraic
acc	accuracy of the neural network
a_s	wear depth of the grinding wheel

ORIGINAL PUBLICATIONS

Publication I — **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

Publication II — **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;93:240–5. https://doi.org/10.1016/j.procir.2020.04.050

Publication III — **Chakraborti A**, Heininen A, Väänänen S, Koskinen KT, Vainio H. Evidential Reasoning based Digital Twins for Performance Optimization of Complex Systems. Procedia CIRP 2021;104:618–23. https://doi.org/10.1016/j.procir.2021.11.104

Publication IV — **Chakraborti A**, Vainio H, Koskinen KT, Lammi J. A Graph-Based Model Reduction Method for Digital Twins. Machines 2023;11:733. https://doi.org/10.3390/machines11070733.

Author's contribution

This thesis includes the scientific outputs from four published publications. The summary (compendium) was prepared by the author of this thesis (later: the Author) Ananda Shankar Chakraborti, and it was reviewed by the responsible supervisor Professor Kari T. Koskinen.

In all publications, the Author conceptualized and designed the methodology, developed the software and validation of the mathematical models, performed the necessary experiments, analysed the data, and wrote the original manuscripts based on the test results. Furthermore, the Author is the main contributor in all four publications included in this thesis. The named co-authors supervised the study and partook in writing, reviewing, and editing the publications.

The Author and co-author contributions are presented as follows:

Publication I The Author was the main contributor to this article developing and implementing dimensionality reduction method of graph-based system representation of superconducting magnet assembly at CERN (Geneva). The model development, optimization and validation were done by the Author. The co-authors provided their feedback and reviewed the final version of the manuscript

Publication II The Author was the main contributor to this article, developing and implementing model reduction method for graph-based representation of digital twins. The model reduction method, related software development and validation with a grinding wheel wear case study were done by the Author. Arttu Heininen supported the work with the Finite Element Modeling of the grinding wheel. The co-authors provided their feedback and reviewed the final version of the manuscript.

Publication III The Author was the main contributor to this article, developing Artificial Intelligence based methods for consolidation of model reduction method of digital twins proposed in Publication II. The mathematical models and related software development were done by the Author. The co-authors provided their feedback and reviewed the final version of the manuscript.

Publication IV The Author is the main contributor to this article, applying heuristic methods to build the graph-based model and an taking an experimental approach to model reduction. The graph-based model reduction (GBMR) method was developed and tested on a turbo compressor case study by the authors. Henri Vainio supported the work by providing random forest method to benchmark the performance of the GBMR method. The co-authors provided their feedback and reviewed the final version of the manuscript.

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

Digital twins (DTs) have been perceived in multiple ways. Several descriptions of the DT exist in scientific literature, many of which go beyond the three-dimensional DT proposed by Michael Grieves [1]. DTs are described as virtual substitutes for real-world objects consisting of virtual representations and communication capabilities, making up smart objects acting as intelligent nodes inside the Internet of Things (IoT) context [2]. DTs have reached beyond the field of product lifecycle management, where they were first conceived, into manufacturing processes, communication and networking, construction, and smart grids. However, the underlying research question remains: How does one best represent a complex multidimensional DT system and define its usability, which comprises advanced simulation models, communication networks, the IoT platform and infrastructure, data models, artificial intelligence, and machine learning-enabled analytics and algorithms [3]? Given such a complex representation, it is imperative to understand how to simplify such a complex multidimensional representation to make the DT more efficient in mimicking real-life scenarios [4]. This will enhance the usability and adoption of DT technology.

The fundamental question revolves around the appropriate timing to construct a multidimensional DT. When the system possesses limited relevant data and physical knowledge, it is encouraged to create a knowledge-driven system utilizing early simulation models. In contrast, if there is a substantial volume of data but insufficient physical knowledge, data-driven models are encouraged. Conversely, when there is a scarcity of relevant data but a comprehensive understanding of the system's physical aspects, the construction of simulation models is preferable [5]. Although this approach does not introduce novel aspects to the system, it facilitates a deeper comprehension of its physical state. Finally, constructing a sophisticated DT, encompassing comprehensive mathematical information and physical system representation, is ideal when both a substantial volume of relevant data and a profound understanding of the system's physical characteristics are available [6]. For example, a complete virtual environment provided by the DNV GL [7] classification in maritime sector serves as a guideline for construction of DTs. This virtual world consists of comprehensive mathematical models of the physical object, including all sensors and actuators [8]. This thesis focuses on the notion of DT, where in-depth physical knowledge can be effectively harnessed through graph-based methods. Additionally, the research considers the presence or prospective implementation of a robust data collection mechanism, including sensor networks, connectivity, and platforms.

The DT serves as a digital counterpart to a physical object, termed the physical entity (PE) [9,10]. In the absence of the PE, the DT remains a mere model. During the design phase, engineers construct prototypes, but this description alone is insufficient to capture the essence of a DT. A model can be linked to a snapshot frozen in time, losing validity as time progresses. In an ideal scenario, a DT embodies a multidimensional physics-based entity that faithfully replicates the physical object and meticulously records all temporal changes, encompassing the state, internal variations, and environmental data [11]. This comprehensive, multidisciplinary digital representation is referred to as the virtual entity (VE) [12]. The VE meticulously captures all pertinent changes over time from the PE, including undesirable alterations like increasing wear in a grinding wheel or surging in a turbo compression system. The DT effectively embodies this VE representation while encompassing all relevant domains.

The construction of a DT holds significant potential benefits across diverse scenarios and industries. However, this endeavor entails substantial complexity, necessitating the resolution of numerous challenges. Key challenges include data collection and integration, complexities related to modeling and computation, real-time performance, interoperability of diverse technologies, data security and privacy concerns, and cost considerations [13]. In this research, particular attention is devoted to the intricacies arising from model and computational complexity. The study highlights the computational challenges associated with accurately representing the physical entity within the DT in real time. Achieving this requires substantial computational capacity to compute target parameters using high-fidelity simulation models and machine-learning techniques [14–16]. Failure to possess adequate computing resources may lead to asynchrony between the DT and the changing state of the physical entity. To address these issues, the development of model reduction and simplification methods and algorithms becomes imperative.

1.1 Objectives and Scope

The DT is a multidimensional entity. In [17], the DT is realized as a five-dimensional living model. It is a collection of simulation models, information models, and IoT data acquisition and processing. Plenty of research is available on the development of these models and data-driven methods [18]. However, the important area that has been overlooked by the DT research community is model and computational complexity and their mitigation in creating an efficient DT. This thesis argues that model reduction is a crucial part of building a DT to address computational complexity. The DT, realized as a multidisciplinary VE representation of a complex machine system, becomes inherently complex. This triggers discussions about the fidelity of the twin and the computational complexity needed to obtain time-sensitive results from the DT. Hence, the objectives of this research work are as follows:

- Conceptualizing the DT as a graph-based complex multi-dimensional, multimodel representation. The DT comprises a fusion of diverse technologies from various domains. To facilitate the conceptualization of how these technologies operate, promote information sharing, and ensure interoperability, a straightforward and simple representation method is essential. This method aims to conceptualize the DT, fostering a shared understanding among multiple technology developers. This thesis work offers a rationale for explaining the significance and suitability of employing graph-based methods for building such multidimensional and multimodel representations.

- Development of the **g**raph-**b**ased **m**odel **r**eduction (GBMR) methodology for the graph-based DT representation. The multidimensional and multimodel representation is inherently complex. Consequently, there is a necessity to develop model reduction methods capable of addressing this complexity in DTs. These model reduction methods should operate on a graph-based representation of the model while maintaining the model's semantics and contextual integrity.

- Application of the graph-based modeling and model reduction methods in various domains of complex engineering systems. The development of graph-based models and model reduction methods should be applicable to engineering systems. Hence, the final objective of this thesis is to apply the graph-based model reduction (GBMR) methodology to different engineering systems to prove the validity of such a method.

The development of graph-based methods for complex DT modeling is done with an established conceptual modeling mechanism known as dimensional analysis and conceptual modeling (section 2.3.4). To simplify the DT, a graph-based model reduction (GBMR) method is proposed (chapter 3). The GBMR method was first conceived as a dimensionality reduction method, but it evolved into a VE representation and optimization tool for the DT. Finally, the GBMR method is implemented, tested, and validated with the help of case studies from the machine systems domain (chapters 4 & 5).

1.2 Research Questions

To achieve the above-mentioned objectives, this thesis attempts to answer the following research questions:

RQ1. How does one build multidimensional digital twins (DTs) with graph-based methods by combining physics-based and experimental/operational data from engineering systems? Is the graph-based method effective in integrating domain-specific knowledge and data from diverse sources into a unified DT framework?

The challenge confronting the digital twin (DT) community lies in effectively defining and constructing digital replicas that faithfully mirror physical assets, systems, or processes, capturing their inherent characteristics and behaviors in a digital realm. These multidimensional DTs offer a detailed representation of systems or processes, furnishing insights into their behaviors, interactions, and interdependencies across diverse dimensions encompassing time, space, and attributes. DTs combine several facets of the physical asset, including physics-based, data-driven, and behavioral elements. Hence, it is imperative to devise methodologies, tools, and techniques for building hybrid representations that faithfully mirror real-world objects. In this research, well-established graph-based methods are utilized to conceptualize the physical knowledge of the DT. A five-dimensional reference model is adopted to capture the various dimensionalities of the DT, such as physical, virtual, data, connectivity, and service. The virtual representation, or twin, is represented as a collection of models from various domains, such as geometrical, physical, and behavioral. By leveraging a graph-based representation, the relationships among parameters across these diverse domains are embedded using a conceptual modeling framework known as dimensional analysis conceptual modeling (DACM), developed at Tampere University. In addition, combining the greedy equivalence search (GES) method with DACM makes the graph truly hybrid. It is demonstrated that such a graph-based representation of the interrelations among model parameters can be leveraged to instantiate a hybrid DT. By comparing the output from graph-based methods with machine learning-based algorithms, the effectiveness of the graph-based method is defined by KPIs such as computational time.

RQ2. How does one achieve model reduction of the multidimensional graph-based DT using computational methods like graph theory algorithms? What methodologies are suitable for propagating and mitigating uncertainties within such graph-based DTs to ensure reliable decision support in dynamic environments?

DTs that utilize advanced modeling and simulation techniques across multiple domains of engineering are inherently complex. There are no common methods that successfully combine information from these models from different domains. A graph-based model reduction (GBMR) method is developed for this purpose. Some consideration will show that the graph-based representation of the physical system is complex. In other words, for a high-fidelity representation (a dense graph consisting of parameters from multiple domains) of the physical system, a complex graph representation is obtained. These graphical models are computationally extensive and unable to compute target parameter values as fast as the DT demands. Hence, model reduction methods based on graphs are needed that can work on graph-based representations of complex systems while preserving the context and semantics of the system. The GBMR is developed from theories and algorithms from the domains of graph clustering, network theory, and artificial intelligence to provide a reduced representation of complex systems.

Uncertainty and its propagation are inherent in graph-based models, as with any other model. This research question also deals with understanding uncertainty and its propagation in the graph and directs the research towards understanding this uncertainty and its mitigation.

RQ3. How can graph-based methods be applied to DT modeling and model reduction in practice? How can one effectively validate the GBMR for DTs with case studies within the domain of complex engineering systems?

Graph-based model reduction (GBMR) has a strong mathematical and physical foundation. What are the real implications of the method? To answer this question, an experimental approach is taken with the help of two case studies. These case studies are (1) application of GBMR in the development of DT for the grinding machine ecosystem and (2) application of GBMR in the development of DT for a turbocompressor system. Both case studies take different approaches to validating and benchmarking the GBMR method against more established methods to understand the efficacy of the GBMR method.

1.3 Contribution and Structure

The principal contribution of this research lies in the conceptualization of the virtual entity (VE) representation of the digital twin (DT) as a multidimensional and multiphysical graph, effectively integrating both physical and data-driven approaches into a unified hybrid entity, and the reduction of such a graph to reduce the computational complexity of a dynamic VE. The idea is to present the GBMR method for addressing this computational complexity of the DT with a two-step approach: (1) providing a graph-based conceptual model representation of the DT with the help of a conceptual modeling framework and (2) reducing the DT graph model by spectral decomposition and identifying the important parameters in it. The novelty lies in representing the complex engineering system as a graph-based model and reducing that graph by finding important parameters dynamically using the DT. The model reduction process helps optimize the VE performance of the DT, as the reduced model uses a subset of parameters to predict the target parameter of the PE. Therefore, this thesis work presents a GBMR method designed to streamline the intricate representation, enhancing the responsiveness and efficiency of the DT.

The structure of this thesis is as follows: Chapter 2 presents the state of the art in DT development, focusing on conceptual graph-based methods. It provides an extensive overview of several technologies employed in modeling DTs and elucidates the challenges faced by the DT community. Moreover, it explores various graph-based modeling aspects that the DT can effectively leverage. In chapter 3, the GBMR method is introduced, elucidating the model-building process and the steps involved in reducing the graph-based model. This chapter provides a detailed account of the GBMR and the tools and techniques needed to support its development. Following that, chapters 4 and 5 present case studies of the GBMR method from two distinct domains, demonstrating the method's generality and applicability. These chapters also present a detailed analysis of the results and key insights derived from the case studies. In chapter 6, a comprehensive discussion of GBMR takes place, delving into its strengths, limitations, and potential improvements. Lastly, chapter 7 concludes the thesis by providing concluding remarks and outlining the future directions for this research work.

2 STATE OF THE ART

In this section, the complexity involved in building the DT is presented. A reference model from the literature is presented to provide context for the VE of the DT. The VE is a collection of complex multidomain models. To interpret and realize the VE, conceptual modeling mechanisms are needed at the early design phase of the VE. Hence, graph-based modeling approaches are presented. A correct balance should be struck between graph-model complexity, speed, and accuracy. This complexity demands the development of novel model-reduction methods. The model-reduction methods should quickly capture the underlying structure of the graph model and generate a reduced representation of that model for faster and less resource-intensive computation of the target quantity. However, the process of reducing a graph-based model is not straightforward; it requires several considerations and the application of methods and algorithms from the graph-theory and network-analysis domains. This section describes the literature behind these components, which are used to build the graph-based model reduction, or GBMR, method.

2.1 Modeling Paradigm for Digital Twins

Digital twin (DT) is at the forefront of the Industry 4.0 revolution, facilitated through advanced technologies such as the Internet of Things (IoT)/Industrial Internet of Things (IIoT), advanced simulation and modeling, connectivity, and seamless integration between these technologies. Both industry and academia are deeply interested in shaping the future of DTs. This is indicated by the growth of highly cited publications as well as DT platforms under development by companies. Due to the widespread popularity of the DT, several definitions and characterizations of the DT are proposed in the literature [6,19–21]. The most commonly used definition of DT is presented in [22]:

"an integrated multi-physics, multiscale, probabilistic simulation of an as-built system, enabled by Digital Thread, that uses the best available models, sensor information, and input data to mirror and predict activities/performance over the life of its corresponding physical twin"

This definition makes DT inherently complex and dependent on several factors such as multiphysics, multiscale and probabilistic simulation of the physical product, fidelity of the DT, speed of operation of the DT, bidirectional data interconnections, and data update frequencies between the physical object and the DT.

In order to conceptualize a DT amidst these complexities, a three-dimensional DT modeling paradigm was proposed by Michael Grieves [20,23–26]. In a white paper published in 2003, Michael Grieves coined the term "digital twin," proposing three layers of the twin: the physical object, the digital representation of the physical object, and the data exchange between the digital and physical representations. In 2019, Fei Tao extended the three-dimensional concept of the DT to a five-dimensional DT representation [9,17,27–33]. The five-dimensional DT consists of the previously mentioned three-dimensional representation; adding to that, there is the connection between the twins and the digital service provided by the DT. The five-dimensional model is of importance because it provides a holistic representation of the DT.

The five-dimensional DT reference model as proposed by Fei Tao is described with the help of figure 1. A grinding wheel example is used to demonstrate the five dimensions of the DT. These five dimensions are as follows:

(1) Physical entity (PE): This consists of the subsystems and sensory devices. This could range from sensors, actuators, and control systems to the whole subsystem, such as motor drives, spindles, and the transmission of the machine. PE guides the process of DT development by providing IoT data from these subsystems. Therefore, the PE also provides communication interfaces, radio frequency identification (RFID) tags, and distributed sensor networks.

(2) Virtual entity (VE): This is the complex virtual representation of the PE. The VE consists of geometric models, analytical or physics-based models, behavioral models, and rule-based models [29,31]. The VE may contain detailed geometric models like 3D CAD models or physics-based models like finite element models. It may contain various behavior modeling methods, such as Markov chains and ontology-based models. Historical data from the PE is used to create rule-based models. The rule-based models provide the VE with the capacity for judgement, optimization, and prediction.

(3) Service: This provides the reason for building a DT, which is digital services. In figure 1, it can be services such as grinding wheel wear monitoring or early warning for wheel change based on the remaining useful life of the wheel. These services fall under the category of prognostics and health management (PHM) services for the grinding machine.

(4) Data model: The data model creates the schema for data exchange between the PE and the VE. In figure 1, an example is provided. The grinding machine DT requires that sensor data be exchanged between the PE and VE, such as motor torque, motor power, acoustic emission, and wheel wear.

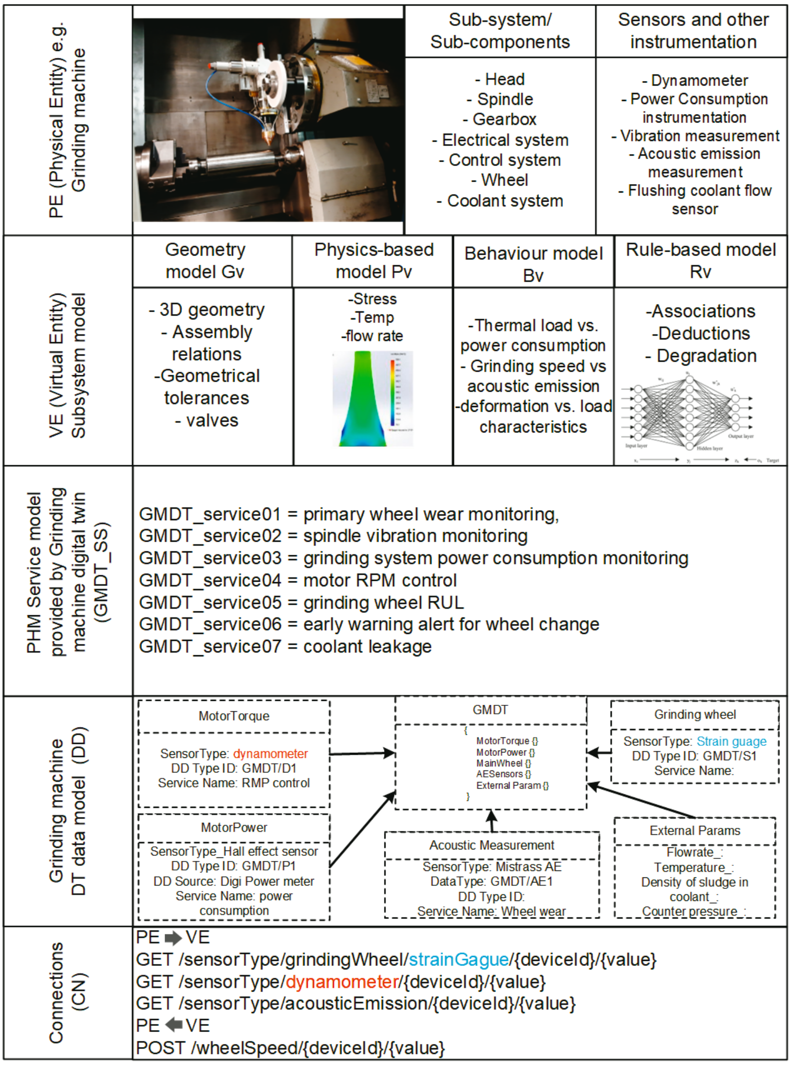
(5) Connections: The connections bind the PE to the VE with the help of the data dimension. PE-to-VE binding consists in acquiring data from the sensors on the grinding wheel with API endpoints. Similarly, VE-to-PE binding provides analytical results to the physical device to perform an action such as grinding wheel speed control.

In the five-dimensional DT reference model, the VE is the key representation. Before building a DT, a deep analysis is needed on what the VE should be and how to build it. Some consideration shows that the VE is not one model but a collection of models. Hence, a model fusion approach is needed to combine the geometrical, physics-based, behavioral, and rule-based models as discussed. Such a model fusion method is demonstrated in the grinding wheel wear case study in chapter 4. According to [28], the model fusion methods have three stages:

i. Construction of the multidimensional model: This indicates the building of unified models based on geometric, physical, or behavioral response principles.

ii. Evaluation and verification of the multidimensional model: After the models are built, a correctness and effectiveness check must be performed. Several tests could be developed for the correctness of the models.

iii. Correlation and mapping mechanism of the multidimensional model: The correlation between the models should be established in the final stage. In this stage, parameters across various models are coupled. By correlating and mapping parameters across various domains, the models can be fused in structure and function to provide a unified VE representation.


PE (Physical Entity) e.g. Grinding machine
Sub-system/ Sub-components
- Head
- Spindle
- Gearbox
- Electrical system
- Control system
- Wheel
- Coolant system
Sensors and other instrumentation
- Dynamometer
- Power Consumption instrumentation
- Vibration measurement
- Acoustic emission measurement
- Flushing coolant flow sensor
VE (Virtual Entity) Subsystem model
Geometry model Gv
- 3D geometry
- Assembly relations
-Geometrical tolerances
- valves
Physics-based model Pv
-Stress
-Temp
-flow rate
Behaviour model Bv
-Thermal load vs. power consumption
- Grinding speed vs acoustic emission
-deformation vs. load characteristics
Rule-based model Rv
-Associations
-Deductions
- Degradation
Input layer
Hidden layer
Output layer
Target
PHM Service model provided by Grinding machine digital twin (GMDT_SS)

GMDT_service01 = primary wheel wear monitoring,
GMDT_service02 = spindle vibration monitoring
GMDT_service03 = grinding system power consumption monitoring
GMDT_service04 = motor RPM control
GMDT_service05 = grinding wheel RUL
GMDT_service06 = early warning alert for wheel change
GMDT_service07 = coolant leakage

PE ➡ VE
GET /sensorType/grindingWheel/strainGague/{deviceId}/{value}
GET /sensorType/dynamometer/{deviceId}/{value}
GET /sensorType/acousticEmission/{deviceId}/{value}
PE ⬅ VE
POST /wheelSpeed/{deviceId}/{value}

Figure 1: Five-Dimensional DT Reference Model of Grinding Wheel

2.2 Complexity in Digital Twin Development

Based on the above discussion, it becomes apparent that the DT is not just an information model or an analytical model. It is a complex, hybrid, cyber-physical system that is formed by the collection of several models from different domains of engineering and data science. The term "hybrid" is often misinterpreted in literature. A hybrid system in this context is a complete representation that integrates the physics-based and data-driven aspects of the system. The DT is a hybrid system that integrates these two aspects according to the popular definitions. Several studies have questioned the misconceptions based around DTs, posing sharp questions regarding the connotations, applications, and research and implementation challenges with the DT [24]. DT has been perceived as a tool for monitoring without physical representation. In other cases, the DT is used as a virtual sensor or 3D model without any information sharing between the simulation models and the PE.

Because of this ambiguous outlook on DTs in the research community and the lack of diverse use cases, attempts have been made to standardize the DT with the help of reference models, such as [34] in manufacturing and risk prediction and prevention [35]. The five-dimensional DT discussed above also serves as a reference model to provide context to the DT and provide guidelines to focus on key areas of the VE.

The key components for building a DT are summarized in figure 2. Figure 2 describes the two pillars of the DT ecosystem: the simulation and modeling (S&M) part and the IoT/IIoT part, which provide a blueprint for the building blocks of the DT. Figure 2 is divided into two parts. The right side of the figure focuses on the IoT part. IoT provides the following components as the building blocks of a DT:

1. Real-time data: The DT depends on robust IoT infrastructure that enables the collection, storage, and analysis of data. The data is obtained from various sensors that are deployed to understand the ground state of the system. Data available from various sources is used to develop and enrich prediction models. Environmental data and other web-based data, such as meteorological data, is used in building predictive models that form an essential part of the virtual entity.

2. Connectivity: The success of a DT relies on the success of the IoT system enabling it. Thus, the choice of connectivity protocols becomes crucial in architecting a DT solution. With the advent of IPv6, any number of devices that have an IP can be connected to the Internet. A DT may demand several application scenarios, such as M2M (machine-to-machine) or M2S (machine-to-systems) communication, and the choice of suitable protocols has the potential to make or break a DT system. From an architectural standpoint, attributes such as interoperability and security are crucial in choosing the right connectivity protocol for the DT.

3. Data-driven models: Data-driven predictive models form the basis of many DTs. Many such data-driven DTs can be found in the literature [36,37]. These predictive models are built for state estimation, behavior

prediction, or causal analysis. Machine learning methods such as convolutional recurrent neural networks (CRNN), artificial neural networks (ANN), and Bayesian networks are used for building these models. These models estimate the state of the system at a point in time in the future. This future state estimation can serve as the input to many simulation models, or it can serve as the output of the data-driven model itself.

4. Model fusion and model reduction: The curse of dimensionality is often experienced in building data-driven models. State-prediction or behavior-estimation models typically contain several parameters that should be monitored, and data should be collected with sensors and a proper connectivity mechanism. Building such high-dimensional, high-fidelity models that replicate reality with a high degree of accuracy is extremely challenging. These models are computationally extensive. Moreover, it is resource consuming to train these models with data from the physical device and develop methods for validating the results. Model fusion provides a means to generate a hybrid model by combining physics-based and data-driven models, and model reduction provides a means to reduce or combine the number of parameters in that hybrid model. In doing so, the hybrid model becomes computationally simplified and takes less time to provide a prediction result. This is crucial for a DT that tries to provide a virtual representation, which is a real-time or near-real-time estimate of the PE.

The left side of figure 2 focuses on simulation and modeling. This part of the figure highlights the advanced simulation models that need to be built to capture the physics of the system. Simulation and modeling create the virtual entity of the five-dimensional representation of the DT [28]. The simulation models can be from one or several domains, such as analytical models, geometrical models, or system-level models. Analytical models such as finite element analysis or computational fluid flow models are necessary to predict the state of the physical entity through software-defined methods such as thermal analysis or stress analysis. Geometrical models purely represent the physical phenomena of the physical entity, such as deformation and buckling. The system-level model combined other types of models to provide system-level information such as efficiency and performance. By combining these advanced simulation models, it is possible to represent the complete state of the PE. The DT demands that these advanced simulation models work in unison and possess the capability to provide the updated state of the system based on real-time data.

Compressing information from these bulky analytical models and making them predict the system state based on real-time data requires further advancement of technology. These simulation models are designed to provide high-fidelity representations of the system without considering the faster prediction of model output. Therefore, it becomes imperative to quickly capture the important facets of these models and solve for the target quantity with reasonable time and resources. For this reason, graph-based modeling methods are used to first build the conceptual representation of the system based on the system architectural blocks of the DT and then, in the second phase, reduce that graph-based representation to determine the important parameters that explain the majority of the system attributes.

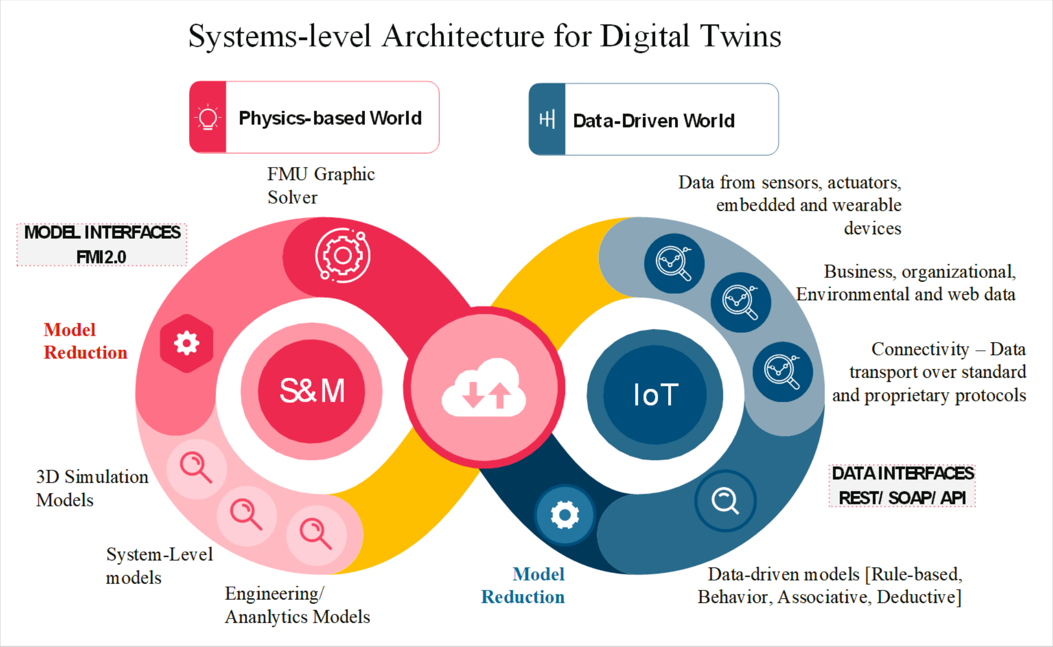

Systems-level Architecture for Digital Twins
Physics-based World
Data-Driven World
FMU Graphic Solver
MODEL INTERFACES FMI2.0
Model Reduction
S&M
IoT
3D Simulation Models
System-Level models
Engineering/ Ananlytics Models
Model Reduction
Data from sensors, actuators, embedded and wearable devices
Business, organizational, Environmental and web data
Connectivity – Data transport over standard and proprietary protocols
DATA INTERFACES REST/ SOAP/ API
Data-driven models [Rule-based, Behavior, Associative, Deductive]

Figure 2: System Architecture for building DTs

It is evident that modeling a cross-domain DT is not as straightforward as other multiphysical problems. This requires building conceptual models that deduce important aspects of the model. On the other hand, a delicate balance of model accuracy and complexity must be achieved. The requirements of model accuracy and model simplicity compete. With complex systems such as a DT, this competition grows largely. To that end, graph-based modeling methodologies are investigated that allow cross-domain conceptual modeling, which is needed in the early phase of VE development. Then, this graph-based representation is analyzed for complexity, and a model reduction method is proposed for that representation. The next section defines the graph-based modeling methods employed in complex system representation.

Graph-based digital twins facilitate multidimensional analysis by integrating data from various sources and dimensions into a unified framework. Such graph-based DTs have already been proposed by [38–40]. These articles provide several advantages for utilizing graph-based representations for DTs. Some of these advantages are as follows: (1) Datapoints can be easily integrated into the graph; (2) graph learning algorithms can be easily applied to the DT graph; (3) relationships between entities across domains in the twin can be easily understood by graphs; and (4) semantics can be largely improved for reasoning in the DT. Given all these advantages, this thesis introduces two novel methods: (i) utilizing a tested framework like DACM for building the DT graph and (ii) model reduction of DT graphs based on graph algorithms such as spectral decomposition and modified PageRank. To the best of our knowledge, graph-based DTs have not employed and tested these methods on engineering use cases. The assumption here is that the larger the number of parameters (nodes in the graph), the more time consuming and extensive computation of specific target parameters becomes. The GBMR helps minimize this computational time by quickly finding those parameters that are more sensitive to the measurement of the target variables.

2.3 Graph-based Modeling and Simulation of Complex System

Graph-based methods are powerful tools for modeling and representing complex systems with several dependencies. In complex systems, there are often numerous interactions and dependencies between different components, which can be effectively captured using edges in a graph. Many simulation platforms used to build DT, such as Modelica, also utilize graph concepts to model system components connected by edges and apply graph-theory concepts such as modularity and inheritance to instantiate and simulate complex systems. Graphs can handle different data types and can represent both physical connections and abstract relationships. Complex systems modeling with graphs provides the benefit of a rich set of algorithms and techniques for analyzing graph structures. These algorithms can be used to extract useful information from the system, such as finding the shortest paths, identifying important nodes or clusters, and detecting patterns. Graph-based models can also be used for predictive modeling and simulation, allowing the forecasting of future behavior or outcomes of the system under different scenarios.

Modern engineering systems consist of a large number of dynamically interactive components that interact nonlinearly with each other. In [41], a detailed account is provided of how graph-based complex systems can be developed for complex systems such as cyber-physical systems. Graph-based modeling and simulation have grown in popularity [42]. The graph-based modeling mechanism, though simplistic in approach, has greater explanatory power. Graphs can capture sophisticated, linear, or nonlinear relationships between system components and represent them meaningfully with superior semantic causal inference. Even in heterogenous complex systems, graph-based modeling methods complement traditional modeling and simulation approaches. Graph-based modeling is applied to a multitude of problems such as engineering design [43], manufacturing [44], reliability measurement, and predictive maintenance [45], to name a few. In all these examples, graph modeling helps in (1) high-level system representation and visualization, (2) reasoning and decision-making, and (3) propagation of information or objectives in the graph.

In high-level system representation and visualization, graph-based modeling is most effective where relationship between entities is the driving force in designing the model. In the graph-based model, components (entities or variables) are represented as nodes and their relationship is embedded in the arrows. Mathematically, a simple graph is represented as $G=(V,E)$, where the elements or variables of the graph are represented as vertices and denoted by V and their relationship is embedded in the edges denoted by E. This is also known as an algebraic graph. A sub-graph of G is represented as $\hat{G}=(\hat{V},\hat{E})$ such that $\hat{V}\subseteq V,\hat{E}\subseteq E$. The adjacency matrix [$A_{ij}$] captures the relationship between all entities in such a graph G, where *i* and *j* are the indices of the adjacency. With the adjacency matrix forms the core of the graph embedding mathematical information about the nature of the graph representation such as, (1) the nature of the graph i.e., directed, or undirected, (2) cyclical, or non-cyclical and (3) loops in the graph. The adjacency matrix transitively connects non-adjacent nodes. The graph Laplacian [*L*], which is calculated from the difference between the adjacency matrix [A_{ij}] and the degree matrix which is denoted as $[D_{ij}]$ provides several useful properties. The following equation provides Laplacian of a simple, undirected graph:

$$[L]= [A_{ij}]-[D_{ij}] \quad \ldots.. (1)$$

Graph modeling is effective where the relationship between entities is a driving force in the design of a data model and where information about data interconnectivity or topology is as important as the data itself [46]. In a complex systems representation in graphical form, complex queries traverse in between nonadjacent nodes (entities) through the edges (relationships) of the graph structure. The edges of the graph represent data objects that are characterized by a high number of permanent and temporary relationships. The size or complexity of such a graph has little impact on the performance of traversal queries. This is the principle of graph algorithms and is widely used in network theory to represent and visualize complex engineering and nonengineering systems.

Reasoning and decision-making are other great contributions of graph-based methods in designing complex systems in modern times. The reasoning and decision-making area can be classified into two subareas: (1) graph theoretic models, which deduce the graph structure with the help of mathematical equations, and (2) machine learning-based methods, which rely on high-quality data.

Graph theoretic models. The edges of the graph embed logical or semantic relationships between nodes. Edge detection and missing relationship identification are possible with the help of graph-based reasoning. By studying the patterns of the missing edges or bridge nodes, it is possible to detect communities and clusters in the graph and hence in the complex system [47–49]. These methods are usually applied to large-scale, complex systems. These methods stem from traditional artificial intelligence methods. Commercial software packages are rare in this case for engineering systems, and the implementation of these methods usually requires development effort.

Graphical machine learning models. With the advent of machine learning, this area of graph-based reasoning has produced many use cases. The most widely used graph-based machine learning methods applied to engineering systems are Bayesian networks [50–52] and, more recently, graph neural networks [53,54]. A Bayesian network is a graph where nodes indicate the variables, and the edges indicate a conditional dependence relationship between those variables [55]. Bayesian networks are good at handling uncertain knowledge and are employed where a high degree of uncertainty is involved in the graph structure. Commercial software packages such as BayesiaLab [56] and open-source packages such as SamIam [57] are available for modeling and as inference engines for engineering systems.

Graph-based as well as data-driven methods have become popular choices in simulation and modeling. This section provides an outline of the modeling and simulation journey of engineering systems. It describes traditional graph-based (visual representation of system components and their relationship) techniques used in modeling engineering systems. The section presents both traditional and newer methods for modeling complex systems. It describes the evolution of graph-based modeling in the engineering domain and how these methods evolved from a purely mathematical or physical representation to graph algorithms. The section also talks about the drawbacks of these methods.

2.3.1 Bond Graph

A bond graph is a graphical system modeling formalism. The bond graph finds application mostly in physical system modeling. It uses the concepts of directed graphs, multiport nodes, and signals and bonds to create a graph-like representation of the physical system. The multiport nodes in a bond graph are characterized by four relevant objects: effort (e), flow(f), power conjugation, and the relation between effort and flow [58]. Readers are directed to [59], where the authors provide a detailed account of the bond graph modeling concepts and the formalism that bond graph system models require. Apart from the formalism of bond graphs, the authors also unambiguously explain rather confusing concepts of bond graph modeling theory, such as conjugate signals and variables, bicausality, and port modulation. This has been the biggest criticism of bond graph theory. Bond graph roots come from the thermodynamic model-based design and control analogy for system modeling [60], which is not at par with designers who are used to more analog system modeling concepts. This has also resulted in the impedance of bond graph modeling methods being adopted by other disciplines apart from physical or mechatronic system models and employed by a few modeling software solutions such as Sim20, ADAM Solver, and AMESim [61]. Compared to general modeling mechanisms, multiport and multibond approaches are difficult to replicate in many problem domains, especially when defining causality, which is an important factor in analyzing system model mechanisms, including bond graphs. Many causality assignment case studies on the bond graph [62] make very domain-specific arguments about its use. This could also be seen in model reduction of bond graph models. Bond graphs employ model reduction algorithms (MORA), model partitioning, and simultaneous order and structure reduction methods. However, it could be argued that these reduction methods are not generic. This is because the bond graph reduction methods rely on bond graph principles such as activity, relative activity, and the energetic contribution index (ECI).

The use of bond graphs for system modeling provides advantages such as consideration of system boundaries and environments and allowing model transformation. It allows the modeling of the dynamic behavior of physical systems by mixing up the configuration structure, physical structure, and conceptual structure of the model. However, the concepts of bond graph modeling methods are not well defined and are outside the domain of many practitioners who use system modeling. There are very few accounts in the literature of applying bond graph methods to CPS or DT development. To that end, this thesis work investigates more generic methods such as graph-based modeling and model reduction, which are easy to adopt and bypass a steep adoption curve perpetuated by bond graphs.

2.3.2 Petri Net

Petri net is a traditional graphical modeling tool used in modeling DES systems, such as industrial automation systems. Petri net was proposed by Adam Petri in 1962 because he wanted to provide a simplistic approach to system modeling, leaving behind the complicated mathematical notations provided by the Markovian process of modeling. Petri net finds application in abstract modeling, formal analysis, and discrete event modeling use cases. A Petri net is a directed graph that provides nodes as tokens and methods known as transitions, which transform the state of the token [63]. It provides several rules by which industrial systems can be modeled, like unified modeling language (UML) models. However, Petri nets are simplistic in nature and mostly restricted to academic research or simple industrial applications such as discrete event system controllers [64]. In the literature, there are very few records of using Petri net as a graph-based tool to model complex systems like DT. However, Petri nets could serve as a good alternative when performing abstract modeling of complex systems.

2.3.3 Metamodeling

Metamodels have several synonyms. They are also called surrogates, auxiliary models, and response surface models. Metamodels quickly capture the underlying phenomena of physics-based models such as finite element models, computational fluid dynamic models, and multiphysical models and embed them into lower-dimensional polynomial models. This definition has led to viewing DTs as metamodels. To capture complex phenomena with the help of a DT, complex physics-based models are needed that take days or even weeks to provide any discernible result. Metamodels, which are mostly empirical in nature, provide results within the required timeframe. Metamodeling is used in the design and optimization stage to simplify computationally intensive simulation of physical systems or phenomena. Popular metamodelling strategies such as 1) Kriging [65], 2) radial basis function [66], 3) neural network (NN) [67] and 4) multivariate adaptive regression splines have been used in several applications [68]. However, the problem with these methods is that with increase in dimensionality of the design problem, new elements of uncertainty are introduced in the approximation model, and the cost of sampling sufficient number of points that can accurately capture the underlying phenomena grows exponentially. Moreover, applying a metamodeling method such as NNs demands a large volume of high-quality empirical data to build accurate surrogates. This adds to the challenges in using metamodeling for DTs.

2.3.4 Dimension Analysis Conceptual Modeling (DACM)

DACM [69–72] is a conceptual modeling mechanism for complex systems. The main goal of this framework is to extract and encode knowledge of different forms in the system with the help of causal representation. DACM derives inspiration from the TRIZ method and the dimensional homogeneity principle to extract causal structure in physical systems. This framework is under active research and development. DACM generates causal graphs that represent, in an unequivocal manner, the chain of events in complex systems. The framework has been successfully applied to case studies in the domains of additive manufacturing (AM), product design, multidisciplinary design optimization (MDO), dimension reduction, and software development.

Dimensional analysis and conceptual modeling (DACM) is a conceptual modeling mechanism used to extract causal relationships between variables in a physics-based simulation environment [73]. This method uses the dimensional homogeneity principle to extract causal relationships between the parameters. DACM is a mature framework, and it is extensively applied by the author and his colleagues to use cases in the fields of additive manufacturing [74] and multidisciplinary design optimization [75]. The DACM framework starts with functional modelling of the system and the assignment of fundamental variables to the different functions of the model. The functions, associated variables, and representative equations are characterized in the causal graph in the form of the cause-and-effect relationship between the fundamental variables of the functional model. The mathematical machinery to check the propagation of an objective in a causal graph is based on Vashy-Buckingham's pi (π) theorem and the dimensional analysis (DA) theory. DACM encodes the domain knowledge of the system in the form of a directed causal graph. Specific checks are run to identify and remove any loops or contradictions in the graph. This ensures a target-driven, directed model. This source of domain knowledge could be from literature, empirical relationships, or analytical models. DACM is combined with machine learning methods such as Bayesian networks for causal inference. The objective of the causal graph provided by DACM is to arrive at the target variable in the directed acyclical graph (DAG) with the help of a set of intermediate dependent and independent variables. Apart from extracting the causal graph, DACM also provides the following checks: (1) generates sets of behavioral equations associated with the causal graphs, (2) simulates qualitative behavior, (3) detects contradictions in systems, and (4) provides a set of analytical concepts for analyzing complex systems.

In addition to originating the causal configuration within intricate systems, the DACM assumes several pivotal functions:

i. The model generates assemblages of behavioral equations that correspond to the causal graphs.

ii. It facilitates qualitative simulation of behaviors.

iii. It identifies inconsistencies inherent in systems.

iv. It furnishes an array of analytical concepts tailored for the examination of intricate systems.

v. It has the capacity to engender a concealed, compact space, like that achieved by autoencoders.

vi. It serves as a preliminary framework conducive to the subsequent application of machine learning methodologies.

2.3.5 Heuristic Methods: Greedy Equivalence Search Methods

The previous methods discussed have one disadvantage. They do not consider the data from the real system in the modeling process. Hence, they are not hybrid methods. To address this limitation in generating the causal graph representation, heuristic methods such as greedy methods are used.

Chickering, in [76], provides a method for graph structure learning with a two-phase greedy equivalence search (GES) algorithm from data. Graph structure learning is a sequential process that learns the relations between the random variables (nodes of a graph) that are embedded in the edges, simulating a causal influence. The GES algorithm provides a mechanism to obtain such a distribution and represent it in the form of a DAG. The GES approach has an important influence on machine learning methods like Bayesian networks for graph structure learning. Another experimental GES is proposed by [77], called greedy interventional equivalence search (GIES), which generalizes the

GES algorithm. Interventions distort the value of random variables to throw the graph out of its original causal dependencies and make it find the original DAG. In this article, the GES algorithm is used to discover the accurate causal reasoning of the DT graph.

It was proved that for two DAGs δ and λ, where δ is an I-map of λ, there is a finite sequence of edge addition and reversals in λ such that (1) after each edge modification, δ remains an I-map of λ and (2) after all modifications, λ is a perfect map of δ. The two-phase algorithm starts with a graph, assuming there are no dependencies. This is indicated as the zero-edge model. Then, all possible single edges are added until the algorithm reaches a local maxima. The phase of progressively adding single edges in the DAG is known as forward equivalence search (FES), and the corresponding local maxima are known as FES local maxima. Once the FES algorithm stops at a local maximum, a second-phase greedy algorithm is applied that considers at each step all possible single-edge deletions that can be made to the DAG. This phase is known as the backward equivalence search (BES). The algorithm terminates when the BES local maxima are identified. The concept is demonstrated in figure 3.

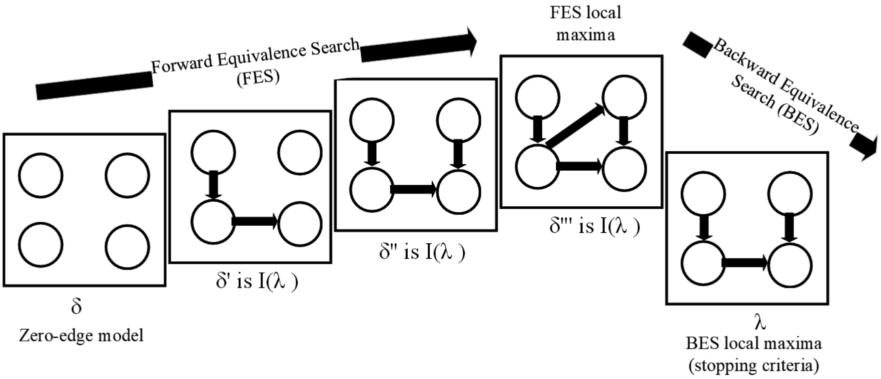

Forward Equivalence Search (FES)
FES local maxima
Backward Equivalence Search (BES)
δ
Zero-edge model
δ' is I(λ)
δ'' is I(λ)
δ''' is I(λ)
λ
BES local maxima (stopping criteria)

Figure 3: Greedy Equivalence Search Algorithm

2.3.6 Hypergraphs

Hypergraphs are mathematical formalism to represent complex systems. The hypergraph has strong descriptive power to describe complex systems and its interconnections. It generalizes many mathematical concepts such as graph concepts and projective planes. Hypergraphs extends the concept of binary interaction between variables in a graph-based model. The readers are directed towards the mathematical foundation of hypergraphs presented in [78]. The author provides examples of applying the concept of hypergraphs in complex informatics systems such as cybernetics, database modeling and network theory. An example hypergraph is shown in the Figure 4. This hypergraph is represented as $H^* = \{V, E\}$ where, $V = \{V_1, ..., V_n\}$ are the vertices of the hypergraph and $E = \{E_1, ..., E_m\}$ represents the set of m hyperedges such that $E \subseteq 2^V$ and $\bigcup_{i=1}^{m} E_i = V$. Hyperedges can contain an arbitrary number of vertices (nodes) and it represents the interactions among the vertices. The properties of the hyperedges differentiates it from other graphs and hypersets. Because of the hyperedge concept, the hypergraph can model the system more efficiently highlighting the interaction of the vertices at different levels.

The concept of hypergraphs has grown in the last decade. Previously, it was restricted to a few areas in the mathematics and computer science domains, for example graph clustering. However, now it is adopted by the systems engineering community and finds application in systems design such as in intelligent transport systems [79], dynamic system modeling [80], and even redesigning the evolution of the supply chain [81].

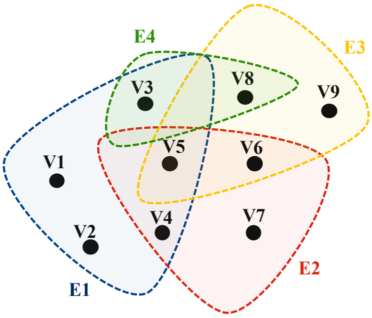


Figure 4: Hypergraphs

2.4 Graph Algorithms

This section describes the state of the art of graph algorithms, especially focusing on graph decomposition and the hierarchical arrangement of nodes in the graph. The spectral method is the most preferred method for structure-preserving graph cuts. However, the method has its own advantages and disadvantages. Node-importance measurement consists of a collection of algorithms that are employed to find the most influential node in the graph. These methods have become an important part of the GBMR process.

2.4.1 Spectral Clustering

Spectral clustering [82–84] is one of the most popular unsupervised learning methods used in graph clustering. Spectral clustering technique finds communities or partitions in a graph based on graph similarity matrix and the graph Laplacian. This method partitions the graph into structurally similar chunks that belong to similar member groups. Spectral clustering outperforms traditional clustering methods like k-means and k Nearest Neighbour (KNN). KNN groups similar nodes in the same part of the graph together. However, proper partitioning of the graph needs to consider all nodes even the ones that are structurally apart. The spectral clustering algorithm separates nodes in different part of the graph according to their similarity indices. To successfully build the spectral clustering algorithm, two important concepts are needed:

(1) Similarity

The spectral clustering algorithm needs a similarity measure between all the nodes in the graph to compute the clusters [85]. The first step of the spectral clustering is to define this similarity matrix. The similarity is how the nodes in a complex graph can be divided into several groups based on pairwise similarity between them. There is no best way to produce the similarity matrix for a given graph clustering problem. But generic methods could be used such as Euclidean similarity, cosine similarity, k-nearest neighbour, and Gaussian similarity. The simplest similarity measure is the Euclidean distance. The Euclidean distance is a method of measuring the proximity between the datapoints of the nodes in the graph. It is mostly used for unweighted graphs. Cosine similarity is widely used by graph clustering algorithm such as spectral clustering and NN [86]. The goal of k-nearest neighbour is to connect the vertex v_i with v_j such that v_j is amongst the nearest neighbour v_i having a defined threshold of k. k-nearest neighbour is the most common method for graph clustering. But it lacks the potential for clustering complex graphs. Gaussian similarity index is used for fully connected undirected and unweighted graphs.

(2) Graph Laplacian

Graph Laplacian provides the mechanism by which the spectral clustering algorithm functions, and it is defined as:

$$[L]=[D]-[W] \quad \ldots. (2)$$

Where, [L] is the unnormalized Laplacian, [D] is the Degree matrix and [W] is the weighted adjacency matrix. This definition holds when G is undirected meaning $[W_{ij}]=[W_{ji}]>0$. However, the causal graph obtained from graph-based modeling methods is a DAG. Hence the Laplacian must be normalized. This is done based on Chung criterion [87] where the normalized Laplacian of a directed graph $[L_N]$ is defined as:

$$[L_N]=[D]^{-\frac{1}{2}}[L][D]^{\frac{1}{2}}=[D]^{-\frac{1}{2}}[W][D]^{\frac{1}{2}} \quad \ldots (3)$$

When the similarity matrix and the normalized Laplacian are defined, spectral clustering algorithm can be constructed. The number of clusters to be constructed is defined as k. Based on $[L_N]$, the first k eigenvectors are computed for the graph. The second corresponding set of eigen vectors for zero eigenvalue is utilized to construct the clusters.

2.4.2 Node Importance Measurement Methods

Identifying the importance of nodes in complex graphs is an active field of research in artificial intelligence. Several studies and algorithms have been published to estimate the importance of nodes in a graph [88]. 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 graph, and finding bridge nodes to detect communities or clusters. Experimental studies have proven the validity of such systems applied to complex networks [89]. The PageRank algorithm is a popular algorithm for graph centrality measurement in directed graphs. It is a network ranking method developed to compute the ranks of webpages in Google's search engine results. The PageRank algorithm iteratively converges to a point for the most influential nodes. Hence, it creates a hierarchical node-importance ranking system in a DAG. An improved version of this algorithm is used in applications that go beyond search engine ranking, which include impact analysis of graph-based system requirements and graph-based feature selection [90]. PageRank is a simple yet versatile method for determining the relative importance of a node in the directed network. It finds application in smart grids [91], recommender systems [92], and requirement analysis [93]. In 1998 by Brian and Page defined PageRank as given a network of N nodes $i=1,2,...,N$ in a directed graph, the PageRank P_i of the i^{th} node is given as:

$$P(i)^n = \frac{1-\alpha}{k} + \alpha \sum_{j=1}^{q} a_{ij} \frac{P(j)^{n-1}}{k_j} \quad \dots (4)$$

Where, α is the damping factor usually 0.8 < α < 1.0, k_j is the number of out links. This definition shows that the PageRank score of a link $P(i)^n$ is dependent of the link it points to i.e., $P(j)^{n-1}$. If $P(j)^{n-1}$ is highly ranked, it influences the rank of $P(i)^n$. The algorithm starts with evenly distributed scores for all the nodes and iteratively all nodes gain a score that hierarchically arranges the nodes. The problem with this approach is, the distribution of weights should not be uniform when the iterations happen. There should be a preference given to the most important node in that iteration to continue the next. Hence, a weighted pagerank (WPR) method was proposed. The traditional PageRank algorithm can be modified to fit use cases better to weighted PageRank score [94]:

$$WPR(i)^n = \frac{1-\alpha}{k} + \alpha \sum_{j=1}^{q} \frac{P(j)^{n-1}}{k_j} . w \quad \dots (5)$$

Where, w is the weight added to propagate a PageRank score from node to node. This w is known as relative popularity of node and should be modelled based on the problem. Convergence criteria of *WPR* defines if there would be a unique solution and all ranks will converge. That is determined with the help of equation 6. In this equation, the difference between the score of the nth node $WPR(x)_n$ and the one before are used to check for convergence. η is a predefined parameter for convergence.

$$\frac{WPR(x)_{n-1} - WPR(x)_n}{WPR(x)_n} \leq \eta \quad ... (6)$$

Eigenvector centrality (EVC) [95] is another popular graph centrality measuring algorithm that finds application in engineering systems such as structural analysis, communication networks, and transportation networks to measure critical nodes in large connected graphs. EVC identifies nodes by the number of the neighbours and their importance. EVC is calculated as:

$$\lambda x_i = \sum_{j=1}^{n} A_{ij} x_j \quad \ldots (7)$$

Where the largest $[A_{ij}]$ is the adjacency matrix, λ is the largest eigenvalue of $[A_{ij}]$ and n is the number of vertices. Node importance measurement algorithms are best suited to compute the importance of vertices in the case where the topology of the graph is defined [96,97]. Node ranking measures can provide an indication of the important nodes in the graph when all parameters mentioned above in defined for the use case.

2.5 Evidential Reasoning

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) Bayes theory, and (2) Dempster-Shaffer theory (DST) or evidence theory-based frameworks. Bayesian methods are widely used techniques based on conditional probability to reason under uncertainty [55]. Bayesian networks provide a mechanism to probabilistically infer the likelihood of an event occurring. However, a major criticism of the Bayesian method is that it cannot handle ignorance nor incomplete, or imprecise information. In the node-importance scenario described above, the absence of experimental data for the variable in the DT graph affects the determination of their conditional probabilities, which makes Bayesian methods inconsistent. In data fusion, effective results can be obtained with Bayesian methods only if adequate and appropriate a priori and conditional probabilities are available. In contrast, DST is a generalization of the 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. Belief functions are extracted to model the problem. Based on the belief functions, a generalized Dempster's combination rule is formed. The results generated by Demeter's combination rule are fault tolerant and better characterize a decision-making problem. [98] provides a comparative analysis between Bayesian methods and evidence theory for failure diagnosis in knowledge-based systems. [99] provides a tutorial on the DST application for online engine diagnostics based on information obtained from multiple sensors such as accelerometers and acoustic emission sensors.

The graph centrality and node importance approaches, though mathematically accurate, often are general and yield contradicting results. This results in disparity in ranking system, introducing uncertainty and incompleteness in the ranking system. Dempster-Shafer theory (DST) [100,101] deals with this uncertainty and incomplete behaviour of any ranking system. Having its roots in probability theory, DST uses data fusion and combinatorial rules to provide belief function to a set of elements in the domain. DST is used to combine the information available regarding the nodes in

DT graph and their relative importance obtained from the node importance scores. 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: $\{h, l, \Omega\}$. Next, the mass functions are determined by adopting a technique similar to the one described in [102] for directed networks. The frame of discernment contains all the possible combination where the combination lies. If there are three hypotheses possible $\phi = \{\theta_1, \theta_2, \theta_3\}$, the set of all combination where the solution lies are:

$$2^{\phi} = \{\Omega\{\theta_1\},\{\theta_2\},\{\theta_3\},\{\theta_1,\theta_2\},\{\theta_2,\theta_3\},\{\theta_3,\theta_1\},\{\theta_1,\theta_2,\theta_3\}\} \ldots (8)$$

The maximum and minimum values of the corresponding ranking is used to compute the **mass functions** with the following formulae:

$$m_{C(i)}(h) = \frac{C_i - C_m}{C_M - C_m + \omega} \qquad \ldots (9)$$

$$m_{C(i)}(l) = \frac{C_i - C_M}{C_M - C_m + \omega} \qquad \ldots (10)$$

$$m_{C(i)}(\phi) = 1 - m_{C(i)}(h) - m_{C(i)}(l) \qquad \ldots (11)$$

Where is a tuneable parameter which is chosen to avoid the denominator becoming zero. Repeating the steps in equation 1-3 creates basic probability assignment (BPA) for each node in the form:

$$M_{C(i)} = \{m_{C(i)}(h), m_{C(i)}(l), m_{C(i)}(\phi)\} \quad \ldots (12)$$

There will be same number of BPA created as there are number of nodes in the sets. Now, all node importance scores obtained from different centrality metrics can be combined with the help of Dempster's combination rule to generate a new combined ranking of the nodes. Dempster's combination rule used in the field of IoT sensor fusion [100] is modified to obtain the new metric for node based on the evidence whether the node is high importance or low importance:

$$m_i(h) = \frac{1}{1-k} \sum_{C(i)=h}^{n} m_{C(i)}(h) \qquad \ldots. (13)$$

$$m_i(l) = \frac{1}{1-k} \sum_{C(i)=l}^{n} m_{C(i)}(l) \qquad \text{…. (14)}$$

where,

$$k = \sum_{C(i)=\phi}^{n} m_{C(i)}(\phi) \qquad \text{…. (15)}$$

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:

$$M_{evidential}(i) = m_i(h) - m_i(l) \qquad \text{…. (16)}$$

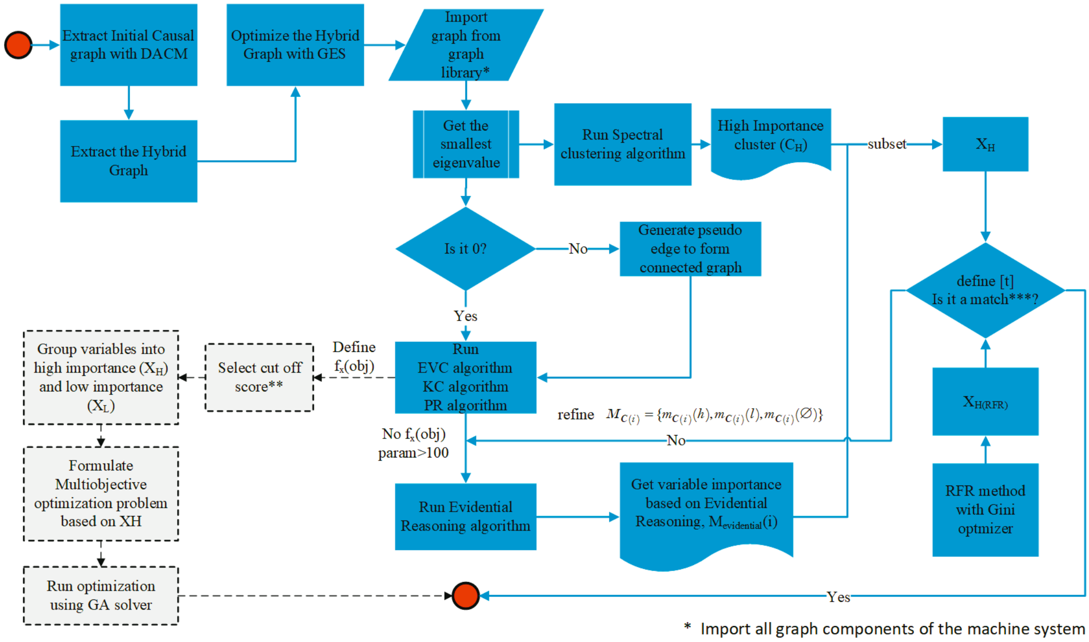
A criticism of DST is it cannot express variability and fluctuations in data at a given phase of time. Hence studies are available on generalizing the DST combination rule that tries to reduce this variability [103]. In this study the author modelled a mass function with the help of complex numbers that has more powerful ability to represent the uncertain information. It is shown that by modifying the Dempster's method such as using complex numbers to generate the Dempster's combination rule or keeping the value of $k < 1$, it is possible also to explain the variability in the data.

3 GRAPH-BASED MODEL REDUCTION METHOD

Graph-**b**ased **m**odel **r**eduction (GBMR) is a method used to simplify and streamline complex graphical models while preserving the essential behavior and characteristics of the original engineering system. This approach leverages graph and network theory algorithms to identify and eliminate redundant or less-significant components from a model, resulting in a more efficient representation that is easier to analyze and simulate. These reduced models are computationally less demanding, allowing for quicker simulations and predictions of state variables and parameters than the DT demands. The reduced graphical representation is also more amenable to design and optimization tasks because of the streamlined representation of the system's behavior.

This section elucidates the intricacies of the GBMR methodology. The procedures integral to GBMR are discussed, elucidating the method's sequential progression. The GBMR approach combines the principles of graph-based modeling, model fusion, and model reduction. Firstly, the procedure for constructing the initial graphical representation of the system is explained. Subsequently, the methodology for a hybrid graph formulation is provided. Upon the development of the hybrid system graph, the GBMR method is applied to it. The GBMR methodology encompasses a sequence of strategic, mathematical, and empirically derived stages, namely (1) spectral clustering, (2) evidential reasoning, and (3) node importance quantification. The outcomes derived through the GBMR methodology undergo validation against machine learning approaches for model reduction, aiming to ascertain the method's precision and accuracy.

The DT is a living, hybrid, cross-disciplinary model of a real entity. For this, many cross-platform parameters need to be coupled together. Model fusion facilitates this coupling process by bringing several high-fidelity models from domains described in the DT reference model in figure 1 to create a unified model that provides specific digital services. Model fusion is a two-stage process where the first stage is building the multidisciplinary model by coupling model parameters. The second stage is simplifying that model for faster interpretability and prediction capacity of target parameters with model reduction. The GBMR method was first presented in [104]. This section describes the GBMR method as a model-fusion strategy for building faster and more accurate DTs. The GBMR method is shown in figure 5, and sections 3.1 through 3.6 explain the building blocks of the method.


Extract Initial Causal graph with DACM
Optimize the Hybrid Graph with GES
Import graph from graph library*
Extract the Hybrid Graph
Get the smallest eigenvalue
Run Spectral clustering algorithm
High Importance cluster (C_H)
subset
X_H
Is it 0?
No
Generate pseudo edge to form connected graph
define [t]
Is it a match***?
Yes
Group variables into high importance (X_H) and low importance (X_L)
Select cut off score**
Define f_x(obj)
Run EVC algorithm KC algorithm PR algorithm
$X_{H(RFR)}$
refine $M_{C(i)} = \{m_{C(i)}(h), m_{C(i)}(l), m_{C(i)}(\varnothing)\}$
No f_x(obj) param>100
No
Formulate Multiobjective optimization problem based on XH
Run Evidential Reasoning algorithm
Get variable importance based on Evidential Reasoning, $M_{evidential}$(i)
RFR method with Gini optmizer
Run optimization using GA solver
Yes

* Import all graph components of the machine system
** Earlier approach of selecting cutoff on node importance score
*** The benchmarking process takes place here

Figure 5: The **G**raph-**B**ased **M**odel **R**eduction (GBMR) Method

3.1 Initial Graph

The initial graph is the causal graph extracted from the physics-based analytical models of the PE. The analytical models are the physics-based models of the physical entity (PE), which is defined as P_v. Parameters from physical systems, such as FEM models, and parameters from literature are used as inputs to the DACM method. The output of the DACM method is the causal graph, which contains all nodes and edges from physics-based equations. An alternative to DACM for building the initial network is graph-structure learning algorithms from data, such as Bayesian networks. The GBMR is combined with a Bayesian network to obtain variable importance in the causal graph [89]. In the Bayesian approach, the conditional probability of each node on the causal graph is calculated based on available data. Then, sensitivity analyses are performed to determine the most responsive parameters using a Bayesian inference engine. The initial graph contains information about the relationship between nodes and weighted edges. It also contains information about the target variables that the DT needs to optimize.

3.2 Hybrid Graph

The GBMR process starts with building the initial graph. However, this initial graph is a physics-based representation. The DT is a hybrid representation. There is a need to inject process data into the initial graph generated in the previous step to build such a hybrid graph. Hence, process parameter data is collected with IoT platforms. When the parameter graph and data are available, an analogy modeling technique is followed to generate the relationship between the parameters. For example, the relationship between power consumption (P), voltage (V) and current (I) of a machine could be established based on the popular relationship $P = V.I$ with the Initial graph as shown in Figure 6A. If the data obtained from the machine contains datasets like Active Power (A(P)), Active Voltage (A(V)) and Active Current (A(I)), the datapoints could be appended to the initial graph and a hybrid graph could be established as shown in 6B.

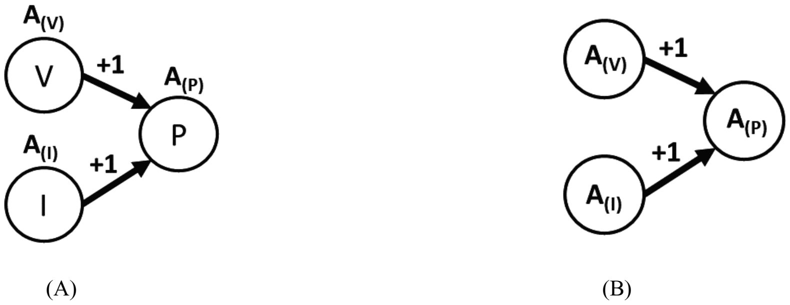


Figure 6: (A): DAG Representation from Initial Graph; (B): Hybrid Graph Representation

3.3 Heuristic Method for Hybrid Graph

The hybrid graph consists of several parameters. Though the relationship is defined by the underlying physics, it is not necessary that a relation exists between two entities. Hence, a heuristic approach is taken. Greedy search algorithm such as GES as described in the section 2.3.5 is applied to construct the final graph. The GES progressively generates and removes edges in the graph with FES. It finally stops at the point where the BES hits the local maxima. This process is computationally extensive. For graphs with high number of nodes, this process is done in stages. The

GES algorithm is suited for high-dimensional datasets. GES was applied in causal model discovery in directed graphs like the hybrid network in [105]. The GES provides the final graph that serves as an input to the model reduction method. The process of GES is demonstrated in Algorithm 1. The input to the algorithm are data collected from the PE, the nodes or variables involved (n_i), an order of deletion that defines to what extent deletion of edges can be done (k) and a deletion score (λ) which is the terminating parameter. The FES and BES methods are constructed to obtain the final graph. The section marked with "**A**" in the algorithm is constructed with python library known as sempler [106]. This library is used in the field of causal discovery in real data set where relationships between the variables are unknown.

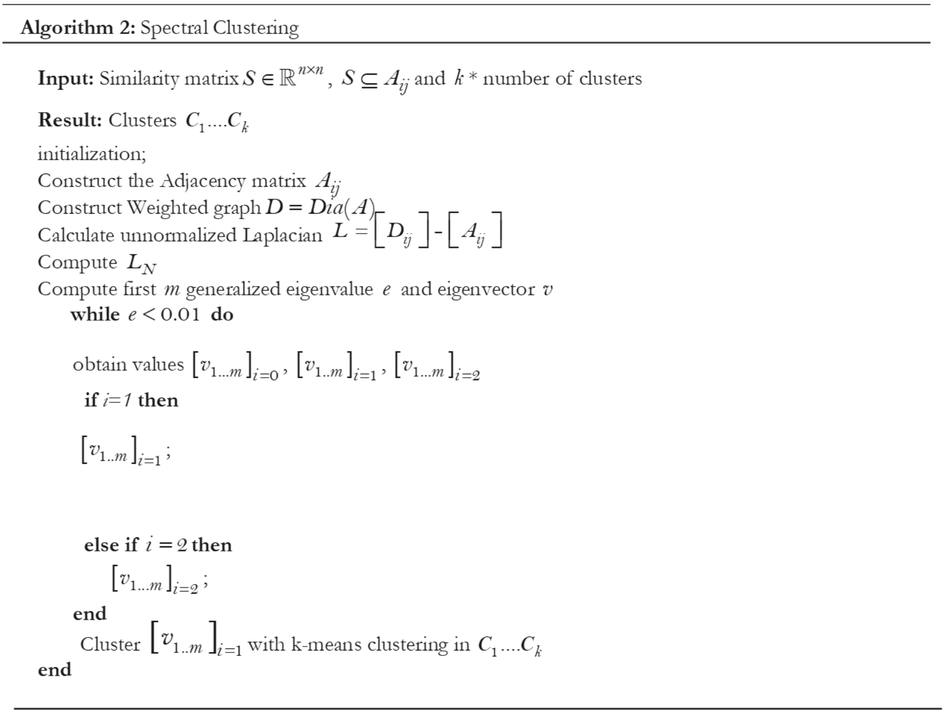
Algorithm 1: Greedy Equivalence Search

```
Input: Data $D$, order of deletion $k$, nodes $n_i$, deletion score $\lambda$
Result: Final DAG $G$
initialization;
Define I-MAP of $G$

    | Forward Equivalence Search ($n_i, D, k, \lambda$):
    |       Generate maximal cliques $C = C_{n_1}, ..., C_{n_n}$
    |
    | Backward Equivalence Search ($C$):
 A  |       Repeat
    |               Edge deletion with $k$
    |               If $\lambda < 0$ && every node in $G \le k$
    |                       Return $G$
    |           Else
    |               $k = k + 1$

end
```

3.4 Graph Spectral Clustering

The evolution of the GBMR method continued when more fundamental questions were raised about the causal graph structure. The hybrid graph input to the GBMR method assumes that the causal graph structure is complete when the node-importance measuring algorithms run on it. This is indicated by a * sign in the Figure 4 when importing the graph from the graph library. That means the graph structure does not change at runtime (no nodes or edges are added or removed). Hence, it becomes possible to segment the bigger graph into structurally similar chunks. A spectral clustering-based graph-cut method for DAGs was used for this purpose [107]. Spectral clustering method learns the graph structure and provides the hierarchical cluster based on the graph Laplacian [108]. The Spectral clustering algorithm classifies the parameters into cluster membership based on the adjacency matrix.

The spectral clustering algorithm defines three parameters: number of clusters in which the graph should be split, the affinity or adjacency matrix of the graph and random state used to initialize the graph decomposition method. Because of this graph decomposition, the nodes which belong to the similar clusters can be identified that might have similar behaviour in the graph.



Algorithm 2: Spectral Clustering

```
Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$ , $S \subseteq A_{ij}$ and $k$ * number of clusters
Result: Clusters $C_1 ....C_k$
initialization;
Construct the Adjacency matrix $A_{ij}$
Construct Weighted graph $D = Dia(A)$
Calculate unnormalized Laplacian $L = \left[ D_{ij} \right] - \left[ A_{ij} \right]$
Compute $L_N$
Compute first $m$ generalized eigenvalue $e$ and eigenvector $v$
  while $e < 0.01$ do
    obtain values $\left[v_{1...m}\right]_{i=0}$, $\left[v_{1..m}\right]_{i=1}$, $\left[v_{1...m}\right]_{i=2}$
    if $i=1$ then
      $\left[v_{1..m}\right]_{i=1}$;
    else if $i = 2$ then
      $\left[v_{1...m}\right]_{i=2}$;
    end
    Cluster $\left[v_{1..m}\right]_{i=1}$ with k-means clustering in $C_1 ....C_k$
end
```

The first step of the algorithm is to obtain the similarity matrix S which can be the adjacency matrix $[A_{ij}]$. The input to the spectral clustering algorithm also contains the number of clusters k in which the graph should be divided. The unnormalized Laplacian $[L]$ is computed with the help of equation 1 and the Laplacian in normalized with the Chung criteria to obtain the normalized Laplacian $[L_N]$ with equation 3. When $[L_N]$ is obtained, the eigenvalues e

are computed. Then the eigenvectors v corresponding to the smallest eigenvalues are computed. The value of k is determined based on the number of peaks in the eigenvector plot. Figure 7 shows the graph cuts based on the spectral method in a graph of 183 nodes used in the turbo compressor case study.

The peaks in figure 7 (first and second eigenvector plots with 0 eigenvalue) indicate sharp change in values, indicating a possible graph cut. These peaks decide the value of k in the k - means algorithm. k - means is used to construct the clusters and group variables in it. Spectral clustering provides an idea of how structurally similar the variables in the graph are. Spectral clustering method is combined with the Node ranking method to identify the important variables.

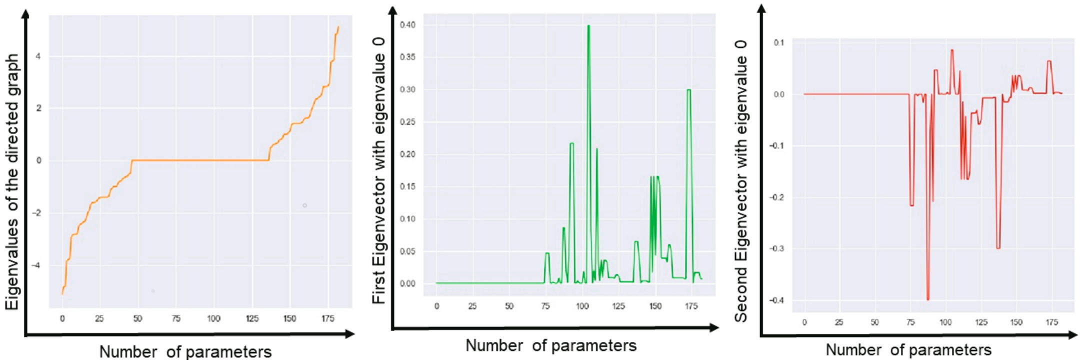


Figure 7: Spectral Graph Cuts

3.5 Importance Measurement

In GBMR, graph centrality methods such as WPR and EVC are used for the hybrid graph to identify the important nodes. If the smallest eigenvalue is zero, the node's importance can be measured. When these nodes are identified according to the WPR score, they are compared with the results of the spectral clustering algorithm. When both the node centrality algorithms and spectral clustering rank the node highly, the node is declared an important node [109]. If the eigenvalues are not zero, the hierarchical node rankings can be applied. This check is made in the GBMR method soon after the hybrid graph is obtained. If the hybrid graph provides zero or negative eigenvalues, the graph is not complete. There are some edges connected in the wrong locations, and some nodes are not connected at all. In either case, the input graph is invalid. If there are nodes that cannot be connected in a legitimate graph with all other nodes and edges in place, a pseudo-node is generated for the nearest neighbor to make the graph valid.

The WPR algorithm used in GBMR is shown in algorithm 3. The WPR is calculated with a method known as power iteration method (PIM). The method takes several factors as inputs such as the DAG G, the damping factor α, maximum number of iterations the algorithm should run $i_{\max}$, and the weight w needed for weighting the edges of the PageRank. For check the convergence of the algorithm an error tolerance e_{tol} is also defined. If w is not mentioned, it is assumed to be 1. $i_{\max}$ is usually kept as 100, as the algorithm converges before 100 iterations. The damping factor α is kept in between 0.85 and 1 which indicates a gradual convergence towards the final score. In the section marked as "**A**" in the algorithm, the validity of the DAG is checked. As mentioned previously, the graph must be a DAG. Then in the "**B**" section of the algorithm, equation (5) is used to compute the WPR score. The value of w can be altered to get the best WPR score depending on the case at hand. In many cases $1.5 \leq w \leq 2$ provide good rank distribution. Finally, in section "**C**" of the algorithm, the convergence is checked, and the final score is returned for all nodes based on equation (6). So, the result of the WPR is a hierarchically ordered matrix. This matrix can be decomposed into high importance matrix [X$_H$] and low importance matrix [X$_L$] based on suitable threshold of WPR score. The algorithm is applied to a case study of superconducting magnets and described in **Publication I** [110].

Algorithm 3: Weighted PageRank with PIM ($G, \alpha, i_{\max} = 100, w$)

```
Input: DAG ($G$), damping factor ($0.85 \leq \alpha \leq 1$), maximum iterations ($i_{\max}$ = 100), weight($w$),
        error tolerance ($e_{tol}$)
Result: Weighted PageRank vector $WPR(x)_n$
initialization.
Define DAG $G$
Check if DAG is valid.
      if len($G$) ==0
A             return null
      if $G$ is directed
      else
              return {}
    Run the power method with $x$ as the initial guess:
B         Power Iteration Method ($i_{\max}$ = 100) :
          Calculate WPR based on the equation 5
    Check convergence:
C         Calculate convergence based on equation 6
    if $\eta < e_{tol}$:
              return $WPR(x)_n$
end
```

3.6 Analysis of Importance

The GBMR method further evolved when the eigen vector centrality algorithms such as EVC and Katz were compared with the output obtained from modified PageRank algorithm, the match of important parameters were less than 60%. It was discovered that the accuracy of the node's importance depends on the ranking method selected. To address this issue, evidential reasoning technique such as Dempster-Shaffer theory was used for consolidating the node importance scores. DST complements the GBMR method by providing a belief structure to decide when a node is considered high importance. The mass functions are calculated and $M_{evidential}(i)$ was computed. Even though the node is differentially ranked by the node ranking algorithm, DST helps decide by fusing the available information of the nodes which are contradictory in their ranking. An aggregated node ranking is achieved with the application of DST which is used to select the parameters for DT optimization problem [104]. The DST method was built with the python library known as pyds [111]. This library provides prebuilt functions for DST for computing mass functions, belief functions, plausibility, and commonality functions.

The outcome of the GBMR method was benchmarked against machine learning based methods such as random forest regressor (RFR) and Gini importance. RFR with Gini importance is a fast and accurate way of analysing feature importance in high dimensional data. The benchmarking process provides an estimate of relevance of the important parameters. Gini provides a superior method of feature importance measurement than other methods such as PCA or SVD. RFR and Gini optimizer provide a parameter ranking stored in [X_H(RFR)]. [X_H(RFR)] is compared to [X_H] to check the number of parameters declared important by both methods. A threshold *t* is defined at this stage as shown in Figure 5. If the match percentage is below *t*, the mass functions and BPA is revaluated, and the process is run again.

An alternative approach to the RFR method is to analyse the GBMR method by formulation of an optimization problem. This is shown in Figure 5 by the dotted grey boxes. The problem was formulated as a multi-objective optimization (MOO) case. The objective function is setup with the target variables of the DAG and the parameters from [X_H] used to attain the target variables. An empirical approach was taken to determine a pareto efficient solution. However, formulating such a multi-objective optimization problem is challenging in situation containing high number of parameters. In section 5, a MOO is set up for validating the output of the GBMR method.

The GBMR method is validated with the help of two case studies: (1) DT of a Grinding Machine, described in **Publication II & Publication III** [104,112], and (2) DT of turbocompressor, described in **Publication IV**. In both these case studies, graph-based system models are developed with the help of techniques mentioned in section 2.3. Model fusion principles are used to combine models from different domains. In these case studies, two different graph-based approaches are taken for the model development part. Finally, the GBMR method is applied to these graphs to simplify the system by identifying the important nodes. Both the case studies use assumptions to make the GBMR process applicable. Validation by benchmarking is done by competing the GBMR method against machine learning-based methods such as RFR, as mentioned in the previous section.

4 CASE STUDY I: GRINDING WHEEL DIGITAL TWIN

The case study and its findings are published in **Publication II**, **Publication III** and partly in **Publication I**. The case study describes the need for the GBMR method when building multidimensional DTs. The graph-based model development is done from a finite element model built to study the physics of grinding wheel wear. This finite element model is an input to this case study, and it is published in parts in [113]. This case study is the outcome of the Business Finland-funded digital ecosystem development project ÄVE. The study was conducted in between 2020 and 2021.

4.1 Graph-Based Model Development

In this case study, the reference model presented in figure 1 is implemented on a cylindrical surface grinding case study. In grinding operations, an important concern is the life of the grinding wheel. The PE consists of various subsystems of the grinding machine; several sensors and IoT/IIoT devices are integrated with the machine. Acoustic emission sensors, accelerometers, temperature sensors, and sensors for current and voltage measurement are used to capture data from the PE. The VE is the entire virtual representation of the grinding system, unifying several models such as the geometric model $\mathbf{G_v}$, the physics-based model $\mathbf{P_v}$, the behaviour model $\mathbf{B_v}$ and the rule-based model $\mathbf{R_v}$. The models and their role in the VE are described as:

- $\mathbf{G_v}$ is the simplest graphical model. It contains geometric information such as dimensions. G_v contains information like the diameter of the grinding wheel, bore-to-hole ratio, width of the bore, and workpiece diameter. Another parameter that can be obtained from the geometry models of the grinding wheel and the workpiece is the equivalent diameter. This measure allows a comparison between two grinding applications. From the relationship of the equivalent grinding diameter, the graphical model G_v is constructed. G_v is connected to P_v through the geometry information it contains regarding the equivalent diameter.

- $\mathbf{P_v}$ is the graphical representation of the multi-physical phenomena in the grinding process, and it is solved with finite element method to obtain wheel wear and volume of material ground. P_v is generated based on dimensional analysis conceptual modeling framework along with well understood grinding physics available in literature. Firstly, analytical models are built for the grinding wheel such as FEM and Multiphysics models. The parameters used in the FEM models and their relationship are extracted from literature and used as an input to the DACM method. The P_v is also analysed with the help of heuristic methods such as GES algorithm.

- **B_v** represents the dynamic behaviour model of the grinding process. B_v determines the variation of dynamic response such as the grinding force, vibration, and grinding power. This dynamic behaviour model effects grinding quality parameters such as roughness or residual stresses. In [114], the dynamic behaviour of grinding process is defined mathematically as a function of chip thickness and contact length between wheel and workpiece. In high-speed grinding, variation in vibration or work piece runout are dynamic behaviour patterns that influences the grinding quality. B_v can be built as a function of maximum uncut chip thickness and contact length between the wheel and workpiece. B_v is connected to P_v through common parameters governing process physics and dynamic behaviour such as volume of wheel worn V_s and volume of material ground V_w.

- **R_v** is a data-driven model which defines the deduction rule between the grinding wheel wear and the input parameters. Data collection is done from the grinding process with the help of acoustic emission sensors and accelerometers. The input parameters are rotational speed, tangential force and vibration along X, Y and Z. An artificial neural network (ANN) is implemented similar to [115] to associate specific levels of the sensor signals with the grinding wheel wear. The feed forward propagation of the ANN is a logistical regression method, but the back propagation is the actual learning event when the weights of the ANN are learnt to predict grinding wheel wear based on sensor data. To connect R_v to the rest of the models, accuracy of the ANN is chosen. When the accuracy of the ANN is sufficiently high, R_v can be connected to P_v through grinding power [116].

Figure 8 describes the DT test bed setup in Tampere University Mechatronics research lab [113]. The PE is shown in the figure is a grinding machine spindle which hold the grinding wheel and the bottom spindle that contains the work piece. The process parameters such as grinding wheel rotation speed (*v_s*), workpiece rotation speed (*v_w*), feed rate and coolant delivery are continuously measured from the PE. Parameters such as tangential force (*F_t*), normal force (*F_n*), acoustic emission (*a01*) and max temperature (*T_max*) are measured online with specialized IoT sensors. And other parameters such as volume of the wheel worn (*V_s*) or volume of material ground (*V_w*) are measured offline. To build the multi-dimensional graphical model described above, following models are combined:

i. Finite element analysis (FEA) model of the grinding wheel: The mathematical models used to build the FEA are utilized to build the graph-based model. The FEA contains both geometrical and physics-based model information. FEA provides the model parameters. The relationship between these parameters is not obtained from the FEA model. Hence, the relationship between the parameters is modelled with the help of the graph-based method mentioned in section 3.

ii. ii. AI-based model for prediction of process parameters: An ANN-based model is used to predict the state of the process parameters obtained from the FEA. The inputs to the ANN model are online- and offline-measured parameter values. The predicted values of the parameters are continuously compared to the measured parameters to estimate the error. This is indicated by the accuracy (*acc*) of the ANN in figure 9.

iii. Behavior models are established based on different physical phenomena. In the case of the grinding wheel DT, Barkhausen noise is used to tune the parameters and build the behavioral model. The behavioral model of development is described in [97].

In this study, a temperature-based behavior model is generated with grinding power as the main parameter considered for the DT. The temperature is measured with a Hall-effect sensor. The response time for this sensor is 0.5µs. The key challenge for an efficient real-time DT in grinding is to obtain the model's response times within the same order as the sensor acquisition frequency. Such DT requires integrating models that are computationally efficient, thus the need for developing model reduction methods that can be integrated with the grinding DT. This also limits the application of standard model order reduction algorithms, which are built into modern commercial software packages, as they could not be extended to the AI-based or behavior model.

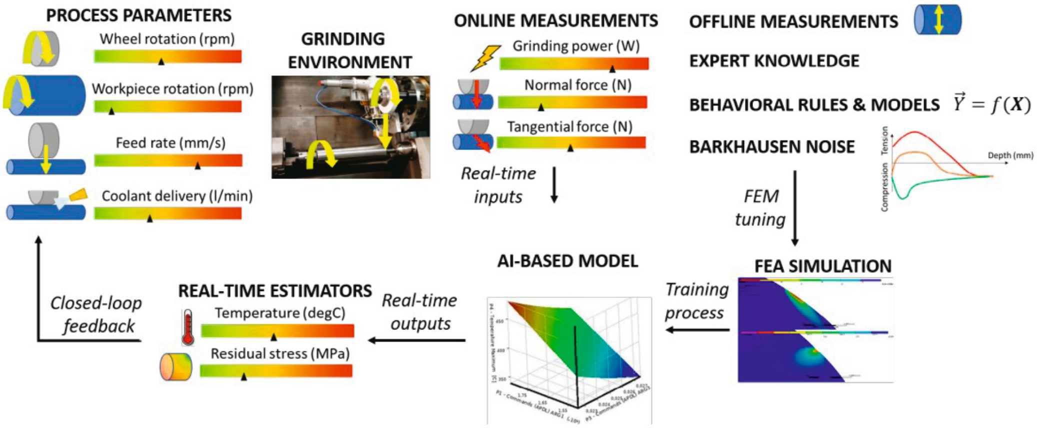


Figure 8: Grinding Wheel Physical Entity.

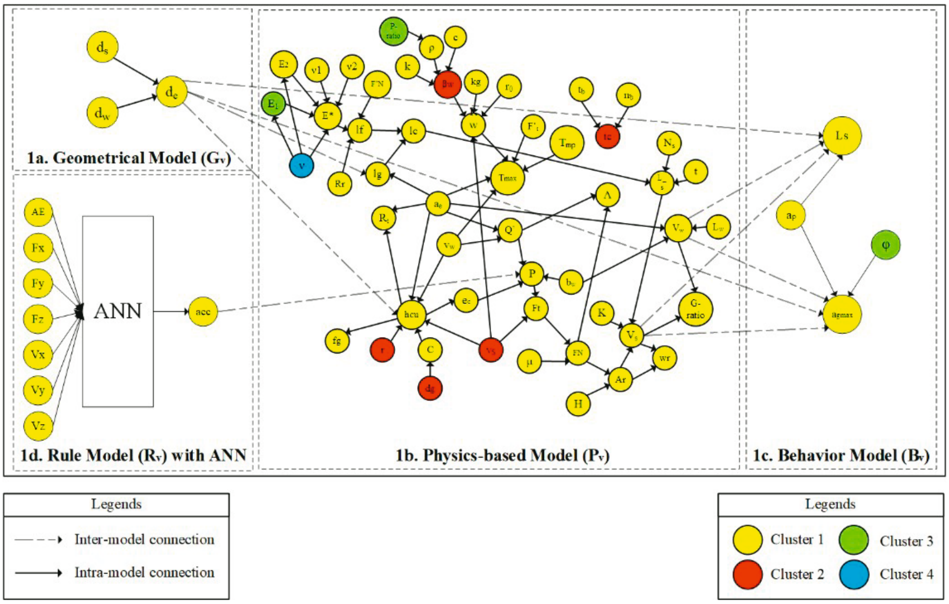


Figure 9: The Fusion Model G_F of the VE; (a) Geometry model G_v; (b) Physics-based model P_v; (c) Behaviour model B_v; (d) Rule model R_v

Graph-based methods described in section 3 is utilized to build and extract the causal relationship between the process, online and offline measured parameters for the grinding wheel wear. The graph-based method provides a causal graph which is a knowledge model of the parameters. The physical parameters needed to build the causal graph are extracted from literature and could be found in **Publication II** [112]. In Figure 9, graph-based method is used to build the fused model of P_v, G_v and B_v and R_v. The physics-based model P_v approximates the Jaeger's analytical model described in [97]. The Jaeger analytical model is used in grinding temperature profiling. The Jaeger model, maximum temperature approximation of grinding and triangular heat flux distribution is used to build the FEA model. The mathematical notations and explanation of these models are provided in Appendix B equation 27 to 45. The P_v is extracted from this FEA model by application of DACM framework.

Fusion of the multidimensional models is an important step in the construction and operation of the VE. These models embed large amounts of information, such as product geometry, process physics, functional behavior, and failure rules, in different formats. Hence, realization of the fusion process is a challenging but essential task. Different software is used to build different models, which might not be compatible with each other. For example, the geometry models are constructed with 3D computer-aided design (CAD) software, whereas the physics-based models are built with finite element modeling (FEM) software, and the rule-based models are constructed with an ANN model. For this reason, a graph-based approach is used for the fusion model shown in figure 9. To fuse such a multidomain model, the correlation and mapping processes of these parameters are analyzed. The interdependence of the model parameters is identified from the FEM model and data collected from the grinding test rig shown in figure 8. An ANN rule-based model is used to correlate the missing relationships into the model core. The models are integrated into their structural topology and functions. Figure 9 shows the fused model, called G_F.

4.2 GBMR Application

The fusion model of the VE contains a large number of variables whose states and values must be known to obtain correct prediction and simulation results from the VE. Hence, the GBMR method is applied to obtain a reduced model of the fused model shown in figure 9 so that knowledge of a smaller number of variables is required to compute and optimize the target variables. This improves the performance of the simulation process of the VE, making it faster and computationally less expensive. The GBMR method used in this case uses spectral clustering and the node importance method to obtain the cluster importance.

Algorithm 1 from section 3 is applied to obtain the spectral clusters. The Laplacian graph is normalized with the help of equation 3. The eigenvector distribution shows there are four peaks, and it is reflected in the spectral graph cut. Algorithm 3 from section 3 is then utilized to obtain the node ranking with WPR. When spectral clustering techniques are used in conjunction with a graph centrality algorithm, such as WPR, the cluster hierarchy can be determined according to their impact on the target nodes. WPR is a class of eigenvector centrality measures. There are other eigenvector centrality measures that take the same eigenvector approach as WPR, such as Katz [117] and eigenvector methods, as indicated in section 2.4.2 and equation 7. However, upon further investigation, it was found that the similarly ranking algorithms used to compute node importance ranking do not agree with each other regarding 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 must be normalized before the application of clustering and the node importance algorithms. The results obtained from both these algorithms are presented in section 5.

Although the three methods— (1) WPR, (2) EVC, and (3) Katz—fall under the class of probabilistic eigenvector distribution methods 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 does not agree completely. This difference in ranking order is shown in figure 11A. 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 results of the system. Hence, there is a need to tackle this uncertainty in the node ranks. DST is applied to combine the results from different centrality metrics based on evidence of whether a node is important.

To validate this approach, performance variables marked in red in figure 9 are optimized based only on the variables from cluster 1 and the values are compared with other grinding case studies from the literature. There are conflicting objectives in the VE as the variable V_s should be minimized and V_w should be maximized in the grinding operation. Hence a multi-objective optimization problem is formulated and solved to demonstrate the significance of cluster 1 variables on the performance of the VE.

4.3 Results

A spectral clustering algorithm is implemented to obtain the graph partition of G_F. G_F consists of 62 variables in total. The first step in the algorithm is to define a similarity matrix. The similarity matrix is chosen as the adjacency of G_F. From this adjacency matrix, the graph Laplacian is computed, and the eigenvalues and eigenvectors are computed from the Laplacian. As G_F is a directed graph, the Laplacian is normalized with help of equation 3 to obtain the graph cuts. The number of clusters are decided based on the eigenvectors, and it is shown in Figure 10. Presence of 4 sharp peaks in the eigenvector graph indicate the presence of four partitions. The cluster is reconstructed with the graph along with k-means clustering where k is decided by the eigenvector values. For G_F, k=4 was selected. Figure 9 shows the clusters from spectral clustering algorithm for G_F. The clusters are colour coded. The cluster marked in yellow is the first and biggest cluster with 53 variables in it, which cannot be subdivided further when the value of k is 4. The variables marked in red belong to the second cluster that contains many variables whose values depend upon the variables in the first cluster. The green and blue clusters contain other less important variables that do not contribute significantly to the prediction and simulation results of the VE. The spectral clustering already sets a hierarchy of variables.

Algorithm 3 is used to find the relative importance of the clusters. This algorithm has shown stability in node ranking and has been popularly used for variable screening in physical networks. The WPR score is shown in Figure 10. It is found that higher WPR scores are obtained for variables lying in the first cluster. Whereas the scores of variables lying in other clusters are low and fail to qualify as important variables when a threshold of 0.025 is selected. This threshold of 0.025 is selected based on the percentile WPR score of all the variables. Also, applying classical Pareto principle on the percentile scores is done.

It can be observed from the spectral clustering pattern that cluster 1 contains the target variables of the physics-based model P_v. At this point, it is desirable to reformulate the VE model based only on the variables belonging to cluster 1 in yellow. Variables in other clusters are placed in $[X_L]$ matrix as they contain dependent variables and their knowledge itself does not alter the computing or simulation time of the VE by a large amount. Clusters 2, 3 and 4 contain variables, which have a low WPR score, and it could be inferred that they do not contribute significantly to the target variables. Cluster 1 is therefore the most important cluster containing all the important variables that should be optimized to obtain higher performance of the twin.

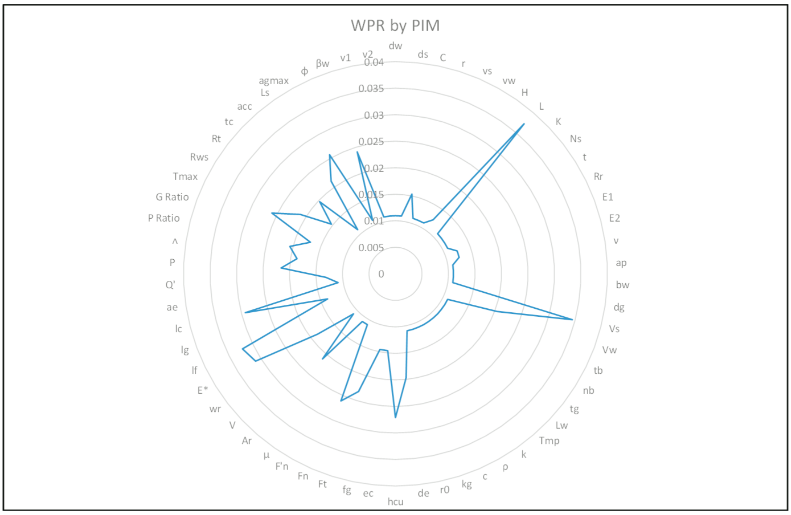


Figure 10: WPR Score for the Graph-based Model.

Moving further on in the GBMR process, the consolidation of the ranking must be done by spectral clustering and node ranking because of the reasons mentioned in section 3.6. The evidential reasoning method is applied for this, as described in section 2.5. The result of the evidential reasoning-based score for the grinding case study is shown in figures 11A and 11B. From figure 11A, it is found that the node importance score based on evidential reasoning aggregates the scores provided by other centrality techniques. Figure 11B shows the disagreement between the three-node importance measurement algorithms. It can be found that in some nodes, the algorithms agree with each other.

The DST score does not affect those nodes. The nodes, which are ranked lower in DST, have more disagreement among the centrality metrics. This introduces a high degree of uncertainty in those nodes that are considered important. On the other hand, the nodes for which all centrality metrics have ranked higher with little disagreement have a higher score from DST. Thus, assuming that 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_s* and *V_w* in the DT graph.

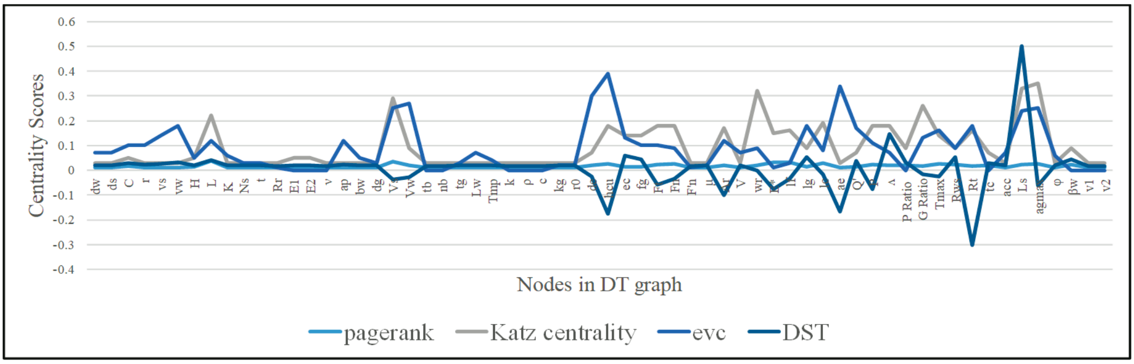


Figure 11A: Comparison of Node Importance by Centrality Methods and DST

To validate the GBMR approach, performance variables V_s and V_w are optimized based only on the variables which are subsets of high DST score and lying in the cluster 1. The optimized values of these performance variables are then compared with other grinding case studies from the literature to understand how close the value of the performance variables is by considering just the important variables with GBMR method.

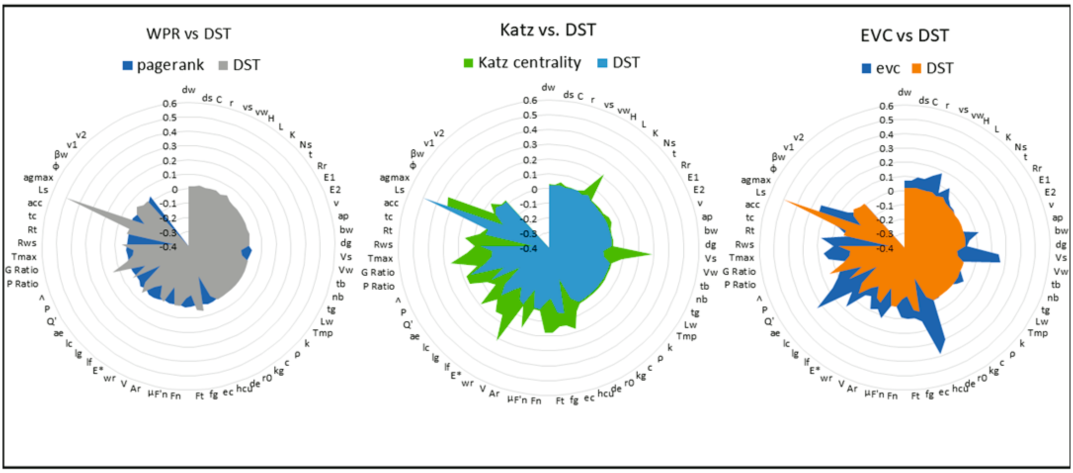


Figure 11B: Disagreement Between Node Important Measurement Algorithms

4.4 Parameter Optimization

Parameter optimization serves two purposes. Firstly, it validates the GBMR method and determines whether the reduced set of parameters can provide the target value accurately. Secondly, the parameter optimization provides the error in prediction in the target parameters. Hence, an optimization problem is set up with the reduced set of parameters. In **Publication I**, similar parameter optimization was set up and solved with the help of a genetic algorithm. In the range of values for the reduced parameters in [X$_H$], the values of the target parameters were noted when the solution hit the pareto front. In **Publication II**, there are conflicting objectives in the VE as the target parameter V_s should be minimized and V_w should be maximized in the grinding operation. Hence, a multi-objective optimization problem is formulated and solved to demonstrate the significance of cluster 1 variables on the performance of the VE. Here is a snapshot of objective functions selection:

f(cluster 1)

objective 1

$$\min : f(V_s)$$

$$\text{where, } V_s = b_w\, a_s\, \pi\, d_s$$

objective 2

$$\max f(V_w)$$

$$\text{where, } V_w = b_w\, a_e\, L_w$$

objective 3

$$\min f(R_t)$$

$$\text{where, } R_t \approx \left(\frac{v_w}{v_s} . \frac{1}{C\,r\sqrt{d_e}} \right)^{\frac{2}{3}} \text{ and } d_e = \frac{d_s d_w}{d_s \pm d_w}$$

$$\text{with model reduction } (v_s \approx 1);\ R_t \approx \left(\frac{v_w}{C\,r\sqrt{d_e}} \right)^{\frac{2}{3}}$$

subjected to;

$$200 \le d_s \le 355\,\text{mm}$$

$$150 \le L_w \le 300\,\text{mm}$$

$$0.0505 \le a_e \le 0.1\,\text{mm}$$

In objective 1, V_s is defined as a function of a_S, which is the wear depth of the grinding wheel. As no standard value for as was found in literature, it is assumed to be the lowest value of the variable a_e. In objective 3, the value of C is chosen as 32 μ/mm2 and grit size r is chosen as 500 based on ANSI B74.16, 1995. Also, for surface grinding, $d_e \approx d_s$. The optimization problem is solved with the help of a MATLAB based genetic algorithm solver and data collected from the digital environment for grinding with IoT devices from the test rig shown in Figure 8. The values of the variables, whose data are not collected from the digital grinding environment, are derived from previous work on grinding parameter optimization such as the case study presented in [118]. Acoustic emission data is collected from the grinding machine with acoustic emission sensors. The acoustic emission intensity is related to the grinding power P to compute volume of wheel wear V_w, if a significant accuracy is obtained from the rule-based model R_v. The fitness function for multi-objective optimization is defined according to the mathematical expression of the performance variables. The optimization result is shown in Table 1, for n_b = 20.

Table 1. Optimization Results for Target Parameters with [X_H]

Variables	**Optimized value**
V_s	24.485 mm^3
V_w	1419.4 mm^3
R_t	0.8 µm

To compare the optimization results with the case study from literature, *GRatio* is used. *GRatio* is a measure of the ability of a grinding wheel to remove materials and it is given by the ratio of V_w/V_s. *GRatio* obtained from Table 1 is 57.95 and *GRatio* obtained from the case study is 60. Hence, less than 10% error is obtained. This indicates that the variables selected by the GBMR could be sufficient to provide meaningful information on performance optimization of the VE.

The trade-off between the model fidelity and computational complexity requires a case based analysis. However, the discussion in this case study is focused around whether a high fidelity model is relevant for the DT of the grinding system. The high fidelity simulation models introduce several computational complexities such as space complexity and time complexity. The high fidelity FEA model for grinding does not consider the memory or storage requirements of the model as the input size grows. This can involve tracking the amount of memory needed for variables, data structures, and intermediate computations. The time complexity of the FEA model is quadratic in nature because of the large number of operations the model needs. The FEA model shown in Figure 8 requires 516.884 min to solve for all target parameters in the high performance computing cluster known as “Galerkin” at Tampere University. The graph-based multi-domain fusion model mitigates this space and time complexity however, it introduces other types of time complexities. As discussed in chapter 3, factor graph and graph decomposition algorithms add to the time complexity. The time complexity of these algorithms is often related to the number of parameters and the graph structure. By simplifying the graph structure and reducing the number of parameters, GBMR contributes to the reduction in computational complexity of the DT.

The service dimension of the DT reference framework shown in figure 1 is the service provided by the GBMR service based on the application. In the grinding wheel wear case, the DT provides several services. For example, a predictive maintenance service by predicting the remaining useful life of the grinding wheel based on continuous monitoring of various parameters such as grinding forces, temperatures, vibrations, and wear patterns. The influential parameters obtained from the GBMR anticipate when the wheel will need replacement or maintenance, minimizing downtime and maximizing productivity. With the help of GBMR, the DT can optimize grinding parameters such as wheel speed, feed rate, depth of cut, and coolant flow rate to maximize material removal rate while minimizing wheel wear and energy consumption. Similarly, the DT provides sustainability services such as improving energy efficiency and reducing environmental impact of the grinding process. By selecting the energy consumption related parameters as target parameters, the GBMR can efficiently find and optimize those parameters. GBMR can suggest alternative process parameters or equipment configurations that minimize energy usage while meeting production requirements. Finally, the DT of the grinding system can serve as a test bed for training operators and engineers.

5 CASE STUDY II: TURBO COMPRESSOR DIGITAL TWIN

This case study and findings are published in **Publication IV**. The case study describes the evolution of the GBMR method from modeling the twin to obtaining the reduced representation. This case study is the outcome of the Business Finland-funded project SNOBI. The study was conducted in 2021 and 2022.

5.1 Turbocompressor Physics

Many advancements have taken place in compression technology over the last decade. One example is the use of active magnetic bearing (AMB)-driven centrifugal compression, which is controlled with variable speed drives (VSD) [119]. AMBs use the principle of magnetic levitation, that is, using magnetic forces generated by electrical coils acting on the conductive materials in the shaft to levitate the rotor within the bearing tolerances, thus reducing the frictional losses in the rotor to almost zero. This principle allows the rotor to operate without any mechanical contact between the static and rotating parts of the centrifugal compressor. As a result, the compressors do not need complex lubrication systems and can run for longer periods without regular maintenance.

Although there are several advantages to a VSD-based centrifugal compressor with AMB, the inherent instabilities in the compression system cannot be totally removed. Instability in the compression system occurs when a small disturbance in the transient gas flow does not return the system to its original steady equilibrium state. There are predominantly two types of instabilities in compression systems: static and dynamic. Dynamic instabilities are more common in compressors, overpowering static instabilities. There are two types of dynamic instabilities known as stalls and surges.

Stalling is a complex flow instability originating from regions of flow stagnation that are created near the impeller blade confinements of the centrifugal compressor, known as stall cells. These stall cells interact with each other and travel around the impeller axis with a speed of up to 70% of the impeller speed, significantly dropping the performance and efficiency of the compressor. Hence, AMB-based compressors are at risk of developing stall cells and eventual impeller failure. Stalling can be progressive or abrupt in nature. The progressive stall is the more common and riskier form of stall.

Stalling is a precursor to surge, which is the principal destabilizing phenomenon in centrifugal compressors with or without AMB. Due to stalling, the compressor cannot generate enough pressure at the outlet to match the pressure built up inside. This forces the compressed air to flow back towards the inlet, resulting in a rapid asymmetrical oscillation known as surge oscillation. Surge is a concern for compressor manufacturers, as it affects the entire compression system, including the compressor and all components upstream and downstream in the air delivery path. A surge is recognized by violent vibration and high gas temperatures that eventually cause structural damage to the compressor. Hence, surge mitigation requires precise and accurate modeling and control methods. Appendix A provides a list of mathematical expressions used to model the surge and stalling behavior of the compressor with AMB. For this, the Greitzer compression system model is utilized as a foundation for building the graphical model [120]. The DACM transforms the Greitzer compression system model into the initial graph of the VE.

5.2 Graph-Based Model Development

The GBMR method is applied to the graph-based representation of the system. The initial graph is constructed with DACM, as described in figure 5. The target parameters are identified by the propagation of strategic objectives in the causal graph. The causal graph is checked for loops and contradictions, and they are removed. The parameters from the initial graph are appended to the system-level model, and the hybrid graph is established. The hybrid graph is treated as the input to the GES algorithm provided in algorithm 1. The final graph is the knowledge model, which consists of details from both the physics-based and data domains. The final graph is used as an input to the model reduction process. The following section describes the physics of turbocompressors for initial graph development. (All standard notations are used in this section. An explanation of the notations is presented in appendix A, equations 17 to 26).

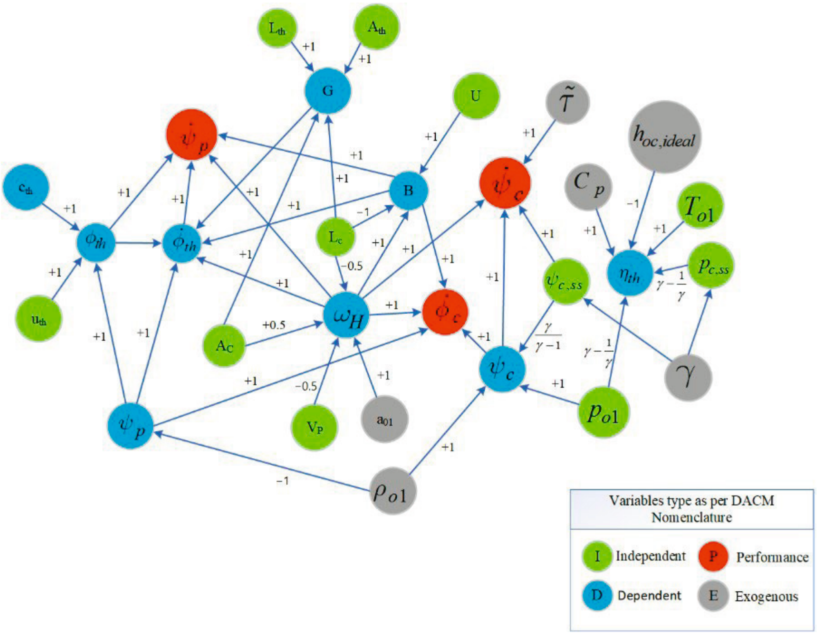


Figure 12: The Physics-based Initial Graph of Turbo Compressor

Initial graph: Surge is a gradual built-up phenomenon whose occurrence can be understood by monitoring pressure rise in the compressor, the plenum pressure rises, mass flow rate of compressed air and mass flow rate at the throttle. These variables and their interrelationship are defined by the Greitzer compression system model which provides a lumped parameter model of the compression system. The mathematical equation needed from literature to build the graph-based Greitzer compression system model with DACM is described in equations (17) through (26) in Appendix A. This model provides the basic physics-based causal formulation of turbocompressor system.

The causal model is used to extract and represent the causal relationship between the variables in the Greitzer Compression System Model. The result graph obtained is shown in figure 12. This network is built to optimize the stability of the compression system. Stability is quantified with ψ_p, ψ_c, ϕ_c and ϕ_p from the Greitzer Compression System Model. This is indicated with red-colored variables. These are called performance variables in the DACM nomenclature. The variables in red are the target variables defined by the system. The blue variables are the dependent variables, which are dependent on the green variables, known as independent variables. The black variables are exogenous variables that are not affected by external or internal change. The weights on the edge of the network are assigned as per the power law and utilized in the GBMR approach.

Hybrid graph*:* The initial graph captures the physics-based representation of the system; however, it is bound to the steady state of the system, and its implication is theoretical. In earlier studies, efforts were made to define the practical usage of this causal graph entity using theories from artificial intelligence. System-level models, as in figure 13C, represent the dynamic state of the system that is related to the physical phenomena guiding it. In the VSD unit of turbofans and motor systems shown in figure 5B, the variables from the dynamic system model are organized, and the causality between these variables is extracted, guided by the initial graph (i.e., the Greitzer compression system model). By embedding parameters or networks from figure 5 into relevant sections of figure 13C, a hybrid of the physics-based and the systems-level models is created. This combination of the physics-based and systems-level models is called the hybrid graph. This hybrid graph is the dynamic system-level model in DAG form.

The methodology described in section 3.2 is used to build the hybrid graph. The factors affecting the pressure differential parameters in the three turbomotors and compressor systems in the systems-level model—pressure differential, maximum free air delivery, motor speed, and motor stator temperature—are associated with nondimensional pressure rise, dimensional mass flow rate, impeller tip speed, and stagnation temperature, respectively, from the initial graph. Similarly, the surge limit is connected to the set pressure, and motor rpm is connected to the active power based on the physics of the system. The same methodology is repeated to create the causal model of the whole system. Additional parameters where this methodology cannot be applied, such as state variables and ON/OFF signals, are considered independent if they can be connected to a dependent variable. If the variable has no association with the physics of the system, the assumption is made to consider it exogenous.

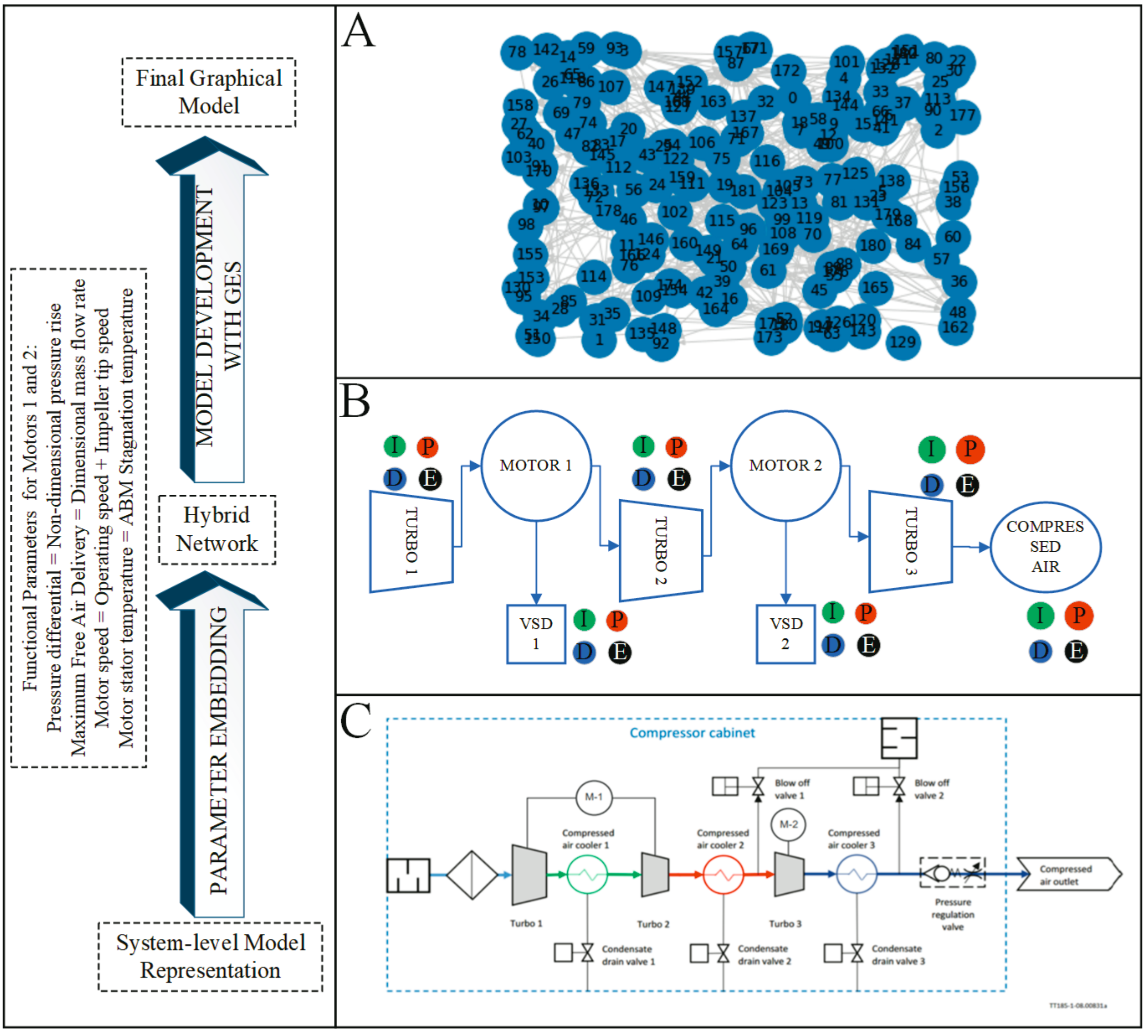


Figure 13: Parameter Assignment and Causality Extraction of Hybrid Model. A: Hybrid Model, B: Parameter Assignment and Causality Extraction Process, C: System-level Model

The GES algorithm (algorithm 1) is applied to the hybrid graph model, and the result is shown in figure 13A, which was constructed in stages, considering every section of the system-level model in 13C. The GBMR method is applied to the hybrid graph shown in figure 13A. The results are discussed in detail in section 5.

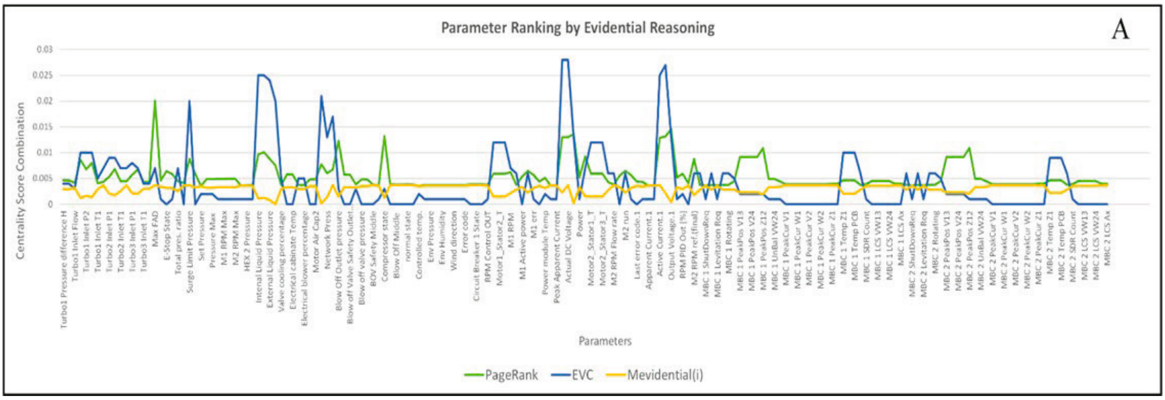
5.3 GBMR Application

The hybrid network developed in 13A is a complete knowledge graph in line with the VE representation of the DT, embedding the relationship obtained from the underlying physical phenomena, the system-level information model, and the data collected from the turbocompressor system. The final network contains 183 nodes.

The result of the ranking algorithms is shown in figure 14A. The spectral method and node ranking methods are done simultaneously. The spectral clustering algorithm is applied to obtain the structure-preserving graph cuts. The spectral clustering algorithm is developed with a Python API that uses clustering of normalized Laplacian. The eigenvector plot clearly indicates that there are five peaks with zero eigenvalues, indicating the presence of five ways the graph can be partitioned. The cluster membership result of the parameters is indicated in figure 14B. Cluster 0 contains 112 parameters; the second-largest cluster, cluster 3, contains 33 parameters, and the third-largest cluster, cluster 2, contains 28 parameters. The two remaining small clusters contain five parameters each. If the clusters are reconstructed with the number of clusters at six, no change is observed in cluster 0, cluster 2, or cluster 3. The smaller clusters start fragmenting as the number of clusters increases. It is interesting to note that the small clusters contain nodes that are farther from the center of the graph, where the target variables are located.

WPR and EVC scores are shown in figure 14A. The WPR and EVC algorithms for weighted edges are a modified version of the NetworkX Python library implementation [121]. The WPR scores of nodes in cluster 0 are consistent. The intersection of the high WPR scores with cluster 0 is agreeable except for the environmental parameters, which were added to the dataset externally. The highest WPR score is given to maximum FAD. This is consistent with the hybrid graph, as FAD is considered a target variable. WPR also nominated motor speed as an important factor associated with FAD in the hybrid graph. EVC agrees with the WPR in many cases; however, EVC differs in the ranking scheme given by the WPR. State variables such as motor state and turbo state differ considerably between EVC and WPR. Hence, the consolidated ranking results are obtained from the node-importance algorithms with DST. This is indicated by the difference between the blue and green lines in figure 14A. DST is applied to this hierarchical ranking system to find a combined ranking. This is indicated by the yellow line and marked as the final evidential mass, or $M_{evidential}(i)$. DST can consider all the available pieces of evidence in deciding whether a node is important or not, can give a trusted final decision. After application of DST, the nodes are rearranged, and the nodes lying in the intersection of cluster 0, cluster 2 and cluster 3 with $M_{evidential}(i)$ are considered as important nodes. These nodes are stored in a high importance parameter matrix $[X_H]$ and the other variables are kept in a low importance parameter matrix $[X_L]$.

Because of the many variables declared important (173), a cutoff is set for the score. When a threshold of a lower limit of 25% of the score is selected, the number of important parameters obtained is 145. Therefore, a 22.4% reduction in parameters is obtained. That is, with 22.4% fewer parameters, the hybrid graph can compute the target variables. The values of the target variables are measured from the reduced model, and they are computed and compared with the values from the literature presented in [119]. These values are Free Air Delivery, pressure differential for all turbo compressor (*Turbox_OUT_P – Turbox_IN_P*), where x is 1, …, 3. The error in the Free Air Delivery is ($e = FAD - \dot{\phi}_C$) and it is less than 4%. The error in pressure differential is e= (Turbox_OUT_P – Turbox_IN_P) $-\dot{\psi}_C$ and it is less than 6%. The maximum value from DST scores is 0.0037. Considering a lower threshold than 25% results in selection of 173 parameters or 2.73% reduction. This makes the GBMR very inefficient. On the other hand, a higher threshold eliminates important parameters ranked high by both EVC and WPR. This result is valid for the selected threshold, which is obtained by a numerical analysis. For a different threshold, the percentage reduction varies. The threshold selection method is proposed by combining two methods: (1) based on cross-validation experiments to evaluate the performance of different thresholds on multiple subsets of the turbocompressor data and (2) incorporating domain knowledge from experts in the shopfloor. A more accurate method of obtaining the threshold should be realized as a future direction of this research. These parameters are used to construct a benchmarking procedure for the turbocompressor case study.



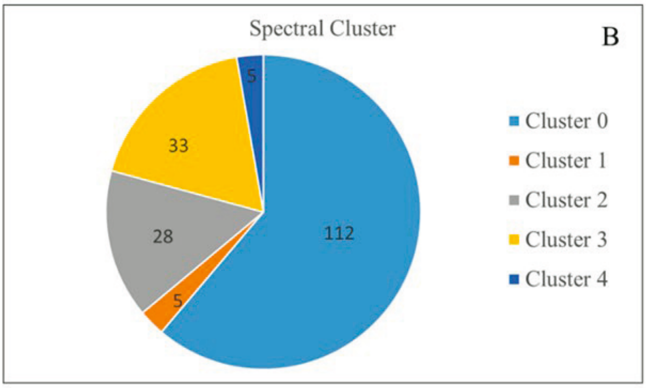


Figure 14: (A) Parameter Ranking by Evidential Reasoning Algorithm; (B) Cluster Membership of Nodes.

The DT provides several services tailored for turbocompresors such as in the fields of sensor integration, data fusion techniques, and predictive analytics algorithms targeted towards minimize downtime, and improve overall equipment effectiveness. With the help of GBMR, the DT can integrate physics-based models, data-driven algorithms, and machine learning techniques, providing services such as predict potential failures and optimize operational parameters. By capturing the relationships between parameters and their dependencies, the GBMR enables a holistic view of the turbocompressor system, facilitating fault detection, performance optimization, and decision support.

5.4 Results

A validation by benchmarking approach is designed for the GBMR method when applied to the VE representation of the DT. The benchmarking method compares the GBMR method to machine learning-based approaches. Classical methods such as RFR [122] or a mixture of RFR and deep learning methods such as CRNN [123] can be used to find statistically significant parameters that contribute most to explaining the target outcomes when a large number of feature sets are present in the data. Such approaches become necessary because, unlike other domains, large training datasets are not readily available for identifying important variables for compressor surge prediction.

The RFR is a type of ensemble method machine learning algorithm. It consists of several decision trees, each constructed based on a randomly selected subset of the training dataset. CART, a popular training algorithm, is used for this. The training indicates that the individual trees learned some features from the training data, and the ensemble of all trees learned the features present in the turbocompressor dataset. The estimation is carried out by voting or sampling some statistical value from the estimations of the individual trees [124]. A by-product of the training of the

RFR is combined feature importance. Feature importance is a method of ranking features based on how much each feature reduces the impurity of the estimation through the nodes of the decision trees. The importance of each feature is calculated by normalizing the total reduction in impurity the feature causes. The feature importance identification can be done by two methods: Gini importance [125] and permutation importance [126]. It was observed experimentally that the importance distribution in Gini can be skewed by ranking some features much higher than others. Hence, permutation importance is used to estimate the importance of the features. Permutation importance is calculated using a trained estimator and dataset, such as the RFR. The algorithm first uses the estimator to calculate a baseline output using some metric on the dataset, then it permutes a feature of the dataset and recalculates the output metric. This is repeated with the other features. The difference between these metrics then indicates the importance of each feature in the dataset with respect to the estimator output.

5.4.1 Experiment Design for Validation

The RFR method is chosen in this study to benchmark the results obtained from the GBMR method. The RFR algorithm was built with a python-based machine learning library scikit-learn. An experiment was designed to achieved that. The RFR was trained with 517902 samples, using the 183 variables as inputs. The target of the training was the power variable calculated as a sum of power variables from the two-motor assembly in the VSD compressor line. A hold out test data set of 129475 was left out of the training data. The hyperparameters of the regressor training were as follows:

- number of trees = 200
- maximum depth of a tree = 10
- the minimum samples required for a split = 2
- The accuracy of the trained model on the training data = 0.9998
- Holdout test dataset accuracy = 0.9997

The top 20 important variables from permutation importance and GBMR are listed in Table 2. The parameters are presented in a high to low order. Figure 15 presents the top 20 parameters obtained from the two methods. The common parameters selected by two methods are more important than the order of the parameters. The parameters with higher importance score are the most important parameters obtained from both the methods. In the benchmarking process, the parameter set obtained from the permutation importance serves as a control.

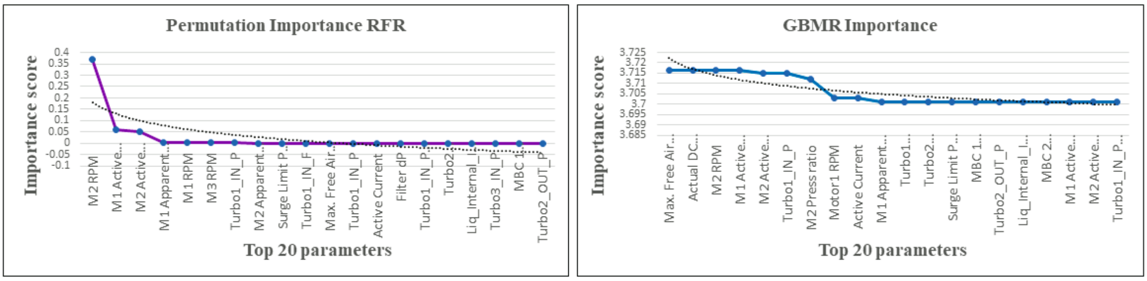


Figure 15: Permutation Importance vs. GBMR Importance

Table 2: Parameter Benchmarking between GBMR and RFR Methods.

Node Importance Rank	GBMR	Gini Permutation Importance Rank of nodes	RFR
1	Max. Free Air Delivery (cal.)	1	M2 RPM
2	Actual DC Voltage	2	M1 Active power
3	M2 RPM	3	M2 Active power
4	M1 Active power	4	M1 Apparent Current
5	M2 Active power	5	M1 RPM
6	Turbo1_IN_P	6	M3 RPM

7	M2 Press ratio	7	Turbo1_IN_P
8	Motor1 RPM	8	M2 Apparent Current
9	Active Current	9	Surge Limit P (bar)
10	M1 Apparent Current	10	Turbo1_IN_F
11	Turbo1 Pressure ratio	11	Max. Free Air Delivery (cal.)
12	Turbo2 Pressure ratio	12	Turbo1_IN_P
13	Surge Limit P (bar)	13	Active Current
14	MBC 1 PeakCur W1	14	Filter dP
15	Turbo2_OUT_P	15	Turbo1_IN_P_DIFF_H
16	Liq_Internal_IN_P	16	Turbo2 Pressure ratio
17	MBC 2 Rotation	17	Liq_Internal_IN_P
18	M1 Active Current	18	Turbo3_IN_P
19	M2 Active Current	19	MBC 1 PeakCur W1
20	Turbo1_IN_P_DIFF_H	20	Turbo2_OUT_P

6 DISCUSSIONS

As outlined in this thesis, the DT represents a complex modeling system encompassing the modeling of the physical entity and the tracking of its behavior over time. Consequently, intricate models with numerous parameters are constructed to address target quantities. This research demonstrates that graph-based methods can effectively serve as a modeling mechanism for building DTs while providing a clear rationale at each step regarding the causality underlying the twin's construction. To manage the computational complexity inherent in DT development and operation, novel methods are essential. Traditional approaches to model reduction and computational complexity assessment have typically been confined to individual domains, resulting in siloed methods such as model order reduction in fluid dynamics and principal component analysis in design optimization. In contrast, graph-based methods are gaining popularity due to their simplicity in representing complex information. Moreover, these methods are enriched by machine learning and graph algorithms, surpassing traditional techniques in terms of speed and accuracy. The case studies presented in the preceding section serve as evidence of these advantages.

In the context of graph-based representation and reduction of multidimensional DT models, researchers have employed various techniques. The literature review conducted for this thesis work revealed several methods commonly used by researchers for reducing graph-based models. These methods are outlined below:

1. Graph truncation: This involves the removal of less significant nodes or edges from the graph that have minimal impact on the overall system behavior.
2. Clustering and aggregation: This entails grouping similar nodes into clusters and replacing them with aggregate representations, leading to a reduction in the overall node count.
3. Dimensionality reduction: Dimensionality reduction techniques decrease the graph's dimensionality, making it more manageable.
4. Model order reduction: This utilizes mathematical methods like balanced truncation to derive lower-order models that capture the essential dynamics of the original graph-based model.
5. Sensitivity analysis with Bayesian methods: This identifies and eliminates noncritical edges or nodes based on probabilistic models through sensitivity analysis, focusing on the most influential components.

These methods are integral components of the GBMR to varying degrees. However, it is crucial to acknowledge that these methods cannot be used in isolation. For example, graph truncation may not be suitable for graphs with a high number of nodes and edges, and pre-mathematically driven methods may yield unreliable results. These methods do not acknowledge the physical relationship between the nodes and edges of the graph. Moreover, these methods may lead to contradictory outcomes. To achieve the optimal outcome from the model reduction process, an integrated and proactive approach to model reduction becomes imperative. Such an approach would consider the strengths and limitations of each method and adopt a comprehensive strategy tailored to the specific characteristics of the graph-based DT model and its application.

The preceding case studies exemplify a comprehensive strategy for DTs based on the GBMR framework. Case I, described in chapter 4, addresses challenges related to grinding wheel wear measurement, outlining the strategy adopted by the GBMR to represent the wheel wear problem using the graph modeling paradigm based on Jaeger's analytical model for grinding profiles. The GBMR framework utilizes spectral decomposition and node importance to identify significant nodes. Unlike conventional methods that focus solely on determining important nodes, the core idea here is to establish a comprehensive strategy for determining significant nodes based on contextual considerations and optimizing the grinding wheel wear process. Case II, described in chapter 5, adopts a similar approach in both the graph model and the application of GBMR. The graph model represents the Greitzer model for turbocompressors, and node importance scores are determined using the Dempster-Shaffer theory, which presents an alternative perspective to Bayesian methods. The outcomes of applying GBMR in these two case studies published in Publications I, II, III, and IV are subsequently discussed.

Case Study I:

This case study proposes a unique graph-based modeling and model reduction mechanism for the DT. This is tested with the grinding machine DT. The grinding machine case study demonstrates that the GBMR reduced the number of parameters by 14.5% with the DT to provide effective values for the target variables. This reduction was directly correlated with resource optimization and reduced effort in the measurement of the parameters, as described in section 4, leading to a simplification of the model structure and a reduction in the computational complexity of the VE of the DT. This is a clear indication of the benefit of the GBMR.

The parameters from [X_H] matrix that are chosen to build the objective function are considered the most significant contributors to the target parameters V_S, W_v and R_t. The challenging part though is to build these objective functions as many times it was found that a direct relation does not exist between the variables and the target parameters. It is possible to simulate the relationship with purely data driven methods like ANN. A genetic algorithm solver was used to obtain the result. The [X_H] matrix consisted of 53 parameters which resulted in a 4% error in the final calculation of the target variables. This is 14.5% reduction in the number of parameters. However, if a lower error percentage is desirable, the threshold for importance is reduced that reduces the reduction percentage.

The application of the GBMR method to the grinding wheel case study demonstrates that a DT can benefit from the advantages of graph-based methods. These methods help in identifying patterns and the parameters that have the maximum impact on the DT. Dynamic optimization of these parameters becomes possible, which helps simplify the computational complexity of the VE representation. This improves the visualization of the system by synchronizing the VE with the PE. This case study demonstrates how the graph-based representation of the VE can be done by combining several models from various domains of the grinding system. The GBMR method was further evolved and applied in case study II to prove its efficacy.

Case study II:

The second case study demonstrates how a DT can be created by combining system-level simulation models with the GBMR method. The GBMR method identifies the important parameters that need to be measured to optimize the performance of the VE. This was done by extracting a causal graph (called a hybrid graph) from the system-level representation and identifying the important parameters in this hybrid graph. The case study demonstrates that GBMR can be successfully applied to machine-system DTs to not only meaningfully represent the DT but also reduce the computational complexity of that representation.

To demonstrate the GBMR, a sample of the top 20 parameters was taken from the 183 parameter set. The accuracy of the GBMR method was benchmarked against the RFR method. Both methods captured the significant parameters from three motors and the main compression parameters, as mentioned in chapter 5. Both methods returned active current and active power as important parameters. This is in line with the hybrid graph, where current and voltage were considered inputs to several parameters, including turbomotor pressure and temperature. Both methods indicate the motor RPM as an important parameter; however, RFR indicates that it is the most significant variable by a large margin, with a mix of power and current variables also indicated. This is because of the permutation importance, and the peak is distinctly noticeable in figure 15. The significance of motor-2 RPM is likely since it explains the operational condition of the motor that runs the last compressor stage and therefore works with the highest pressures. This is clearly linked to the highest energy consumption rate, making apparent current, active current, output voltage, and active power crucial parameters in surge control. The compressor state is correlated to max free air delivery, which in turn is related to the active power of the system. The air pressure value at the intake and compressor control-related variables are also seen as important parameters. A calculated parameter, the surge limit, is marked as an important parameter by both RFR and GBMR. The surge limit is computed from the pressure rise in the compressor and max free air delivery. This is because of the use of node importance methods to rank order parameters. The parameter pointing to important parameters is also considered important.

In the sample of the top 20 parameters, 15 parameters matched both methods. This amounts to a 75% match between the two methods. The RFR is a data-driven method, and it does not contain any knowledge of the physics of the parameters involved. The GBMR, on the other hand, works on hybrid representation. The 75% match indicates that both methods attempt to identify the most important common feature in the parameters. The percentage match between parameters obtained from two methods increases up to 78% when the top 100 parameters are considered. The motor currents and voltages are identified as important parameters in the [X_H] set, as they point to the high-importance parameter active power. The parameters in [X_L] are ignored for benchmarking purposes. When the values of [X_L] are considered, the benchmark percentage varies. To understand the simplification of the computational complexity obtained by GBMR, it is possible to compare the training time of the RFR method with the total execution time of the GBMR method. The training time of the RFR method is 81.3471 min. The total execution time of the GBMR method, which is a summation of graph structure learning, spectral decomposition, and importance measurement, was 72.2318 min. Hence, GBMR is 9.1153 minutes faster at identifying the reduced model. GBMR still provides a fast method to quickly capture the influential system parameters.

The GBMR method provides several advantages over traditional model reduction methods, such as those described in section 2.3. More traditional methods used for system-level model order reduction, such as balanced truncation, principal component analysis, and singular value decomposition [54], also benefit from the graph-based methods. Graph-based methods are more flexible; they use generic algorithms that are applicable to any graph. The graph-algorithms combined with machine learning methods like Bayesian networks [38] and methods from the artificial intelligence domain, such as evidence theory [80-83], provide better alternatives than traditional methods. The design, development, and testing of graph-based model reduction methods are faster than deploying traditional methods as the GBMR uses standard libraries and applies several general-purpose algorithms. The graph based methods are applicable to any category of problems because of their flexibility. Chapter 5 proves this by applying GBMR method of two problems from different engineering domains.

In the graph-based DTs, there are generic and a nongeneric parts. For example, the model of the grinding wheel or compressor is the same for all grinding wheels and compressors. That is the generalizable part. It is even possible to build a component library in the final software package (like DNV GL classification) of the DT platform. However, when the hybrid model is built with the data obtained for the device and the GES algorithm, it becomes a unique, living representation of that device.

Publication Summary

Publication I provides the outcome of applying the dimensionality reduction method to a superconducting magnet case study based on DACM and graph centrality algorithms. The mathematical relationships needed for the casual model construction in the DACM phase are obtained from three conceptual designs in the design phase of 16T superconducting magnets: the cosine theta design, the block design, and the common coil design. Research has found that the set of important variables from the DACM casual model need to optimize the performance of simulation models to effectively design the superconducting magnet assembly. The important parameters were found with a modified PageRank algorithm. After the important variable set [X_H] was found, a multi-objective optimization problem was formulated based on the literature and mathematical relationships of the design variables. The contradictory objectives are identified in the causal model with the DACM propagation of the objective method and can be found in [72]. The dimensionality reduction results are included such that, if there are parameters from [X_H] in the fitness function, they are used as they are. If there are parameters from the low importance set [X_L], they are ignored in the fitness function. The objective functions for stress are convex ($f'(\sigma), f''(\sigma) > 0$). The cost function is calculated by assuming the volume of the key and the per unit cost of the key. Which is a function of C_m and C_t. Other costs are not affected by the volume of the key. The optimization problems are simplified for the purpose of validation of the dimensionality reduction method and the convexity of the problem is checked ($f'(C_T), f''(C_T) > 0$). The constraints defined in the optimization problem are well formed and obtained from literature. With this modification, the multi-objective optimization was run with the "gamultiobj" solver from global optimization toolbox from MATLAB. "gamultiobj" can be used to solve multiobjective optimization problems with several constraints. It aims to find solutions that approximate the Pareto front, which represents the set of non-dominated solutions. These solutions are considered optimal because no other solution in the feasible region simultaneously improves all objectives. "gamultiobj" is based on genetic algorithms. Genetic algorithms are global optimization algorithms, meaning they can search the entire solution space for the best solutions rather than getting stuck in local optima. Genetic algorithms are particularly useful in case of non-convex problems. The solution obtained from the solver is shown in figure 16.

The solution represents the trade-off frontier, showcasing the best compromise between the cost and the stress to design the bladder and key mechanism for superconducting magnets. Solution with heuristic method like genetic algorithm provides an unevenly distributed solution as shown in the figure below. But no dominance of one solution was found. To validate the dimensionality reduction method, pre-determined values of d_{key} and l_{key} were used. Samples in between $d_{key} = [1, 1.5]$ i.e. $6794.06 < f(\sigma) < 15286.62$ was selected to compute the cost function. The value of the cost function was compared with the actual cost. In the sample range of $f(\sigma)$, the error in $f(C_T)$ was observed as less than 6%.

Publication I provides a direction for the development of the GBMR method. It provides a technique for modeling systems in the form of graphs with the help of the DACM method. It also provides the platform for the application of graph theory algorithms such as centrality measurement to introduce the concept of node importance measurement in system graphs.

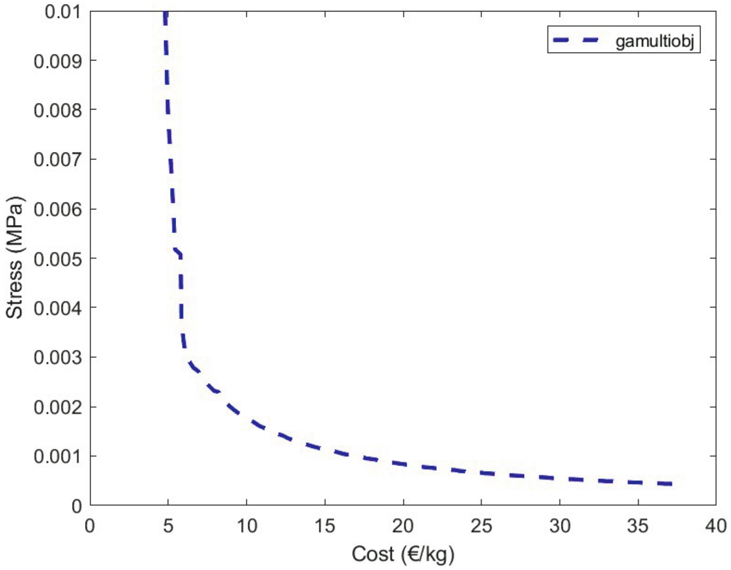


Figure 16: Solution obtained with gamultiobj solver for bladder and key design for superconducting magnets at CERN.

Publication II continues the work proposed in Publication I to reduce the graph-based models of complex systems. In this article, a novel approach is presented to enhance the efficiency of DTs through a model fusion and reduction method. Specifically, we explore the utilization of spectral decomposition to group similar variables in multidimensional graphical representations of the DT, thereby streamlining the optimization process and reducing computational overhead. We focus on the significance of this approach in optimizing the performance variables governing the overall performance of the DT, with particular emphasis on reducing the time required for simulation model updates based on IIoT data from the grinding wheel case study.

Model fusion with graph-based methods involves leveraging the structure and relationships within a graph to combine predictions from multiple models, such as the {G_v, P_v, R_v, B_v} models described in section 2.1. The model reduction method leverages the spectral decomposition of directed graphs obtained from the model fusion method, a powerful technique commonly used in machine learning and data mining, to identify and group similar variables within the virtual entity (VE) of the DT. The mathematical models used to build the casual graph were obtained from Jaeger's computational model and are available in appendix B. By clustering variables based on their similarity, this method aims to analyze their importance in optimizing performance variables critical to the DT operation. To the best of the author's knowledge, such an approach of graph-based model fusion and model reduction (GBMR) is unique and not available in the literature. The reasons for utilizing the GBMR methodology are as follows:

(i) these methods can capture complex dependencies and interactions between data points that may not be captured by traditional methods,

(ii) graphs provide a unified framework for integrating heterogeneous information from different data sources,

(iii) graphs are scalable and can handle very complex data structures ideal for representing complex engineering systems,

(iv) graph theory algorithms can be applied to complex systems. By analysing the graph structure and node attributes, researchers can gain a better understanding of the underlying patterns in the data and the factors influencing the predictions,

(v) Once a graph structure is constructed and algorithms are developed, they can be applied to similar problems with minimal modifications, making them reusable and cost-effective.

One of the key performance indicators of the GBMR is the time required for the simulation models to update themselves based on IIoT data. This is crucial for ensuring real-time responsiveness and accuracy of the DT in reflecting changes observed on the shop floor. By identifying fewer sensitive variables through spectral clustering, the GBMR method enables the elimination of redundant data collection and complex analytics, thereby streamlining the VE representation of the DT.

The results of Publication II are discussed in detail in section 4.3. A similar multi objective optimization based approach is undertaken for validating the methodology. The optimization the target parameters is a trade-off between V_s , V_w and R_t . Contradictory objectives are identified as target parameters in V_s should be minimized as V_w and R_t should be maximized. In the objective function, R_t has dependency on v_s but according to the GBMR method, $v_s \in [X_L]$. Hence, the value of v_s is ignored in the objective function. The outcome of the multi objective optimization is given in table 1. To validate the GBMR method, the optimized value from table 1 is used to calculate known quantities of the grinding system such as G_{Ratio} with an error of less than 10% from the original value.

The results from GBMR applied to the grinding example shows that GBMR successfully identified the parameters in [X_H] that produces a result with 10% error. [X_H] in this case contains 53 variables and [X_H + X_L] contains 62 parameters, which is a reduction of 14.5%. [X$_L$] contains parameters such as v_s and c which are non-trivial and requires effort for data collection. The reduction in computational effort is calculated as the total execution time for the GBMR i.e. summation of computing time for graph structure learning, spectral decomposition, and importance measurement.

Publication III describes the GBMR framework and provides the building blocks of the method. This article is based on the grinding machine case study. Previous studies in Publications I and II propose methods to simplify high-fidelity DT by identifying important nodes within their graph-based representations. By selectively monitoring key nodes, these methods aim to achieve faster computation without compromising predictive accuracy. However, traditional approaches often overlook the inherent uncertainty associated with node selection, potentially leading to suboptimal results.

In response to this challenge, incorporating uncertainty analysis into the node selection process using Dempster-Shafer theory (DST) is proposed in Publication III. Dempster-Shafer theory provides a framework for reasoning under uncertainty, allowing for the representation and combination of uncertain information. By leveraging DST, it is possible to develop enhanced model reduction techniques that account for the uncertainty inherent in DT graphical representation. This publication presents an improved method for selecting important nodes in graph-based DT representations, utilizing Dempster-Shafer theory to address uncertainty that arises when different graph algorithms are used to compute node importance scores. The proposed approach offers a more reliable means of identifying important nodes, thereby enhancing the overall effectiveness of the GBMR method.

The outcome of the publication is providing a framework for GBMR, which is presented in figure 5. The core components of the GBMR method are defined in the publication, such as graph development, running checks for spectral decomposition, node importance measurements, and dealing with the uncertainty in the node's importance. Firstly, the high-fidelity model is structured with mathematical models defined in appendix B. Then, DACM is utilized for extracting the casual relationship for such mathematical models.

The model order reduction is obtained with spectral decomposition. Spectral decomposition leverages the spectral properties of graphs, as explained in section 2.4.1, to identify dominant modes of behavior and reconstruct reduced-order models with reduced computational complexity. The retained dominant eigenvectors serve as the basis for reconstructing reduced-order models of the graph-based system. To avoid the complexity common in graph algorithms, spectral clustering approximates the solution using a generalized eigenvalue problem with graph Laplacians, as discussed in section 3.4. While generic clustering algorithms exist, they cannot be directly applied to networks based on the mathematical representation of complex systems without considering the physics involved. Therefore, this class of algorithms, specifically algorithm 2, utilizes the DACM methodology. These reduced-order models capture the essential dynamics of the original system while neglecting higher-frequency modes that contribute less to the overall behavior of the system. In the final stage, the model reduction problem is formulated from the perspective of multi-objective optimization, constructing a lower-order network model that minimizes a certain reduction error.

The grinding case study serves as a tangible example of how the GBMR method can be applied to optimize asset monitoring with DT technology. By effectively identifying key nodes within the DT representation, users can proactively address potential issues before they escalate, minimizing downtime and maximizing operational efficiency. The integration of DST ensures that the DT is informed by a comprehensive understanding of the underlying data and its associated uncertainties.

The GBMR method is based on clear, well-defined, and well-published methodologies. They specify the steps involved in spectral decomposition, including the construction of the graph, computation of the Laplacian matrix, eigenvalue analysis, and building a causal graph based on the DACM framework. The majority of the GBMR implementation is based on open-source software packages such as Python libraries like NetworkX or MATLAB toolboxes. These software tools provide researchers with access to standardized implementations of reduction algorithms, making it easier to reproduce results across different studies and platforms. The reduced model with the GBMR method has undergone rigorous benchmarking and validation procedures against full-order systems reported in Publication IV to assess its performance and reliability. The GBMR benchmarking results are reported transparently. GBMR provides a set of instructions and code examples on how to replicate the research results. Users need to create the graph-based representation of their complex systems with DACM, and the output of that can easily utilize GBMR. In the future, like DNV GL classification, detailed component libraries could be built for complex machinery such as compressors, grinding machines, or superconducting magnets and provided to the user in one software package. Because of the reasons mentioned above, GBMR is reproducible and reusable.

7 CONCLUSIONS AND FUTURE WORK

This thesis work presents a well-researched claim about viewing DTs as graph-based entities. Graph-based digital twins provide a comprehensive representation of complex systems or processes by capturing the relationships and interactions between components or entities using graph structures. As indicated in the thesis work, much research has already been presented on graph-based DTs. However, this thesis work presents the graph-based model reduction, or GBMR, method, which utilizes graph theory algorithms to identify important parameters that need to be optimized to optimize the output obtained from the DT. Graph-based DTs are flexible and scalable, allowing for the modeling of diverse systems across different scales and complexities. This flexibility enables stakeholders to adapt digital twins to evolving requirements and integrate new data sources or dimensions as needed.

The research demonstrates activities regarding the VE optimization of the DT with the help of a model fusion technology, which is a combination of graph-based model development and model reduction. The VE is a computationally complex entity, comprising models from different domains, that tries to faithfully replicate the state of the PE. The VE combines advanced simulation models, like system-level models, with data-driven prediction models to predict the state of the PE. This thesis work describes the GBMR method for optimizing the performance of the VE with a two-step approach: (1) providing a graph-based conceptual model representation of the VE and (2) reducing the VE graph model by identifying the important parameters in it. The GBMR embeds all the parameters and their relationships in a graph model and facilitates the application of graph algorithms for measuring node importance. Therefore, GBMR facilitates the modeling of physical systems in the form of graphs and the reduction of such models based on graph algorithms. The GBMR makes the VE more efficient, as the reduced model uses a subset of parameters to predict the target parameters in the PE.

Methods like GBMR become important in the context of the DT because they help simplify the VE development while still capturing both the physics-based and data-driven aspects of the twin. With the help of the GBMR, the DT becomes more context aware by knowing the important parameter that it needs to monitor and optimize to obtain the fastest result and reduce its computational complexity. The GBMR method aids in the fast computation of target parameters that the DT is trying to replicate by capturing the intricacies of a multidomain system. It is possible to convince PE owners of what is essential in their system with GBMR and how resource allocation and optimization can be done effectively with DT. The GBMR is developed as a Python-based software package that can be used as virtual emulators for PEs or a prototyping tool where quick estimation regarding the system and the effort needed to build a VE representation can be analyzed effectively. The conclusion provides an answer to the research questions as follows:

RQ1. How does one build multidimensional digital twins (DTs) with graph-based methods by combining physics-based and experimental/operational data from engineering systems? Is the graph-based method effective in integrating domain-specific knowledge and data from diverse sources into a unified DT framework?

The DT combines methods from the physics-based and data-driven worlds to build a multidimensional representation. The analytical or physics-based models are obtained from engineering models such as FEM and mathematical models from literature. These are combined with the help of a conceptual model mechanism to extract the causal graph representation of the physical model. Then, the data from the PE is utilized along with heuristic search mechanisms such as the GES algorithm to determine the best structure of the graph. Chapters 2 and 3 of this thesis work describe in detail the process and methods needed to build the graph-based DT. Chapter 3 provides the detailed process of building a hybrid model by combining graph-based methods and process data. This hybrid method combines the physics-based and data-driven models. **Publications II** and **IV** describe the process of building such hybrid models. **Publication II** also deals with the efficiency of the DT and provides an account of the challenges of optimizing a large number of parameters online with the DT. **Publications II, III, and IV** apply the hybrid method of building DTs to case studies from different engineering domains.

The research question has two parts. The first part is answered by building graph-based representations with DACM and GES methods. DACM effectively provides physical knowledge of the systems, and GES combines this with operational data, creating the hybrid representation as discussed in **Publication IV**. The second part discusses the outcomes of methods such as DACM and GES. To what degree are these methods successful in embedding physical knowledge in the graph? Based on the outcome, it can be said that the methods are effective.

RQ2. How does one achieve model reduction of the multidimensional graph-based DT using computational methods like graph theory algorithms? What methodologies are suitable for propagating and mitigating uncertainties within such graph-based DTs to ensure reliable decision support in dynamic environments?

The DT graph is reduced by a novel method developed in this research work and known as GBMR. GBMR forms the core of this research work. GBMR has several dependencies. It is a combination of methods and graph algorithms that inputs a complex knowledge graph, such as the multidimensional DT graph, and provides a reduced presentation of that graph. The main contribution of this thesis work is to answer this research question with the help of the GBMR method. The GBMR method can be positioned as a framework with a collection of methods and algorithms aimed at tackling the problem of multiple parameters influencing the system in various ways. The GBMR method is described in detail in chapter 3 of this thesis. The GBMR method is discussed in different capacities in **Publications I, II, III, and IV**.

The research question has two parts. Firstly, model reduction based on graph algorithms is achieved with GBMR. Secondly, uncertainty quantification and mitigation are other important aspects that the GBMR provides. In **Publications III and IV**, the GBMR method utilizes DST for the quantification of uncertainty. DST provides a means for reducing the uncertainty arising from interconnecting different domains based on the parameters. The uncertainty quantification is also discussed in other publications, such as [**55,71,127**].

RQ3. How can graph-based methods be employed for DT modeling and model reduction in practice? How can one effectively validate the GBMR for DTs with case studies within the domain of complex engineering systems?

Application of the GBMR to case studies from engineering systems is challenging. The GBMR has evolved along with the applications. However, the GBMR is designed such that the core of the methodology remains the same, that is, to identify the important parameters in the engineering system based on graph-structure learning. The case studies sections of **Publications II and IV** deal with the application in detail. In this thesis, the application of GBMR to grinding wheel wear detection, early warning, and mitigation is described in chapter 4. In chapter 5, GBMR is applied to the turbocompressor case study, which includes the development of the hybrid method by combining graph-based and heuristic methods. The GBMR needed modification in both of these cases, but the core of the GBMR process remains the same. The shortcomings of the method were identified, analyzed, and improved for GBMR. Although the GBMR is demonstrated with the help of two case studies, its applicability is not limited to these areas. The GBMR can be integrated with any DT platform with little modification.

The research questions were answered with the help of the case studies of grinding wheel wear and turbocompressors explained in chapters 4 and 5. The research question was successfully answered, as in both cases, graph-based DT was achieved, and model reduction was done with the spectral decomposition method. To compare the outcome of the GBMR, it was benchmarked against machine learning methods such as random forest regression and convolutional recurrent neural networks (CRNN), with less than 5% error in both cases.

The KPI computational time is noted in the benchmarking exercise. The training time of the RFR method was 81.3471 min. The total execution time of the GBMR method, which is a summation of graph structure learning, spectral decomposition, and importance measurement, was 72.2318 min. Hence, GBMR is 9.1153 minutes faster at identifying the target parameters, with a 5% error margin.

7.1 Future Work

Based on the discussion in this thesis work, the future of model reduction frameworks such as GBMR can be described as follows:

Improving computational efficiency of engineering models: The future of model reduction frameworks will be aimed at simplification of high-fidelity models to create simplified models that retain the essential characteristics and behavior of the system, thus reducing computational costs. In the future, even more efficient algorithms and techniques that can handle larger and more complex models while maintaining accuracy can be expected. These advancements would enable real-time simulations and optimizations for complex engineering systems. With the rapid growth of machine learning and data-driven modeling, it is likely that physical model reduction frameworks such as GBMR will begin to incorporate machine learning techniques. Data-driven model reduction methods will complement traditional physics-based approaches, allowing the exploitation of large datasets generated from simulations, experiments, or sensor measurements. Frameworks like GBMR will play a vital role in such hybrid models, laying the foundation for successful DT development. New modeling and simulation paradigms such as hypergraphs and hyperedges could be more extensively used in DT design to represent the complex relationships between entities more flexibly and expressively. The complexity between entities will grow to a greater extent in the near future, and having a functional DT will necessitate the application of novel technologies such as quantum computing.

Multifidelity and multiscale techniques: Future model reduction frameworks may focus on seamlessly integrating models of different fidelity and scales. This would enable combining high-fidelity, detailed models with low-fidelity, coarse models, allowing for accurate simulations while maintaining computational efficiency. This is essential for DT development. Hence, GBMR should also take these multifidelity and multiscale approaches into account.

Uncertainty quantification: The GBMR framework addresses the uncertainties in declaring the parameters as important with different methods. However, an accurate representation of uncertainties involved in the parameter selection that governs the behavior of the engineering systems is crucial, especially in safety-critical applications such as turbocompression systems. The future model reduction frameworks or extensions of the GBMR framework should incorporate uncertainty quantification techniques to provide reliable predictions and sensitivity analyses. Integrating uncertainty measurements is a challenging task, but it is an essential extension of this research work, and future efforts should be focused on quantification and minimizing the uncertainty between the PE and VE.

Computing resources: Inspiration will be drawn from the fields of ML and AI to build self-explanatory causal models. Closer interactions between system model developers and the AI/ML community are needed to build a DT that is efficient and effective. Model reduction is still in the research and proof-of-concept stages today. The GBMR will be integrated with larger DT frameworks under research and in commercial software products. The GBMR has the potential to increase the efficiency of traditional algorithms. With the increase in computing power and the advent of communication technologies such as 5G and 6G, it is possible to integrate low-cost measurement systems with larger machine systems such as the grinding machine rig or turbocompression system. The future of GBMR is likely to involve more extensive use of parallel and distributed computing resources. Leveraging high-performance computing and cloud infrastructure would enable handling larger datasets and performing computationally demanding model reduction tasks.

Finally, it should be mentioned that different engineering disciplines, such as aerospace, automotive, and energy systems, have unique challenges and requirements. Future model reduction frameworks might be tailored to address specific industry needs, resulting in more effective and specialized tools. In this research work, the GBMR proves its generality by combining case studies from different domains of engineering systems. However, the GBMR needs to be customized according to the use case that is governed by the industry. GBMR creates the avenue for future model reduction efforts to seamlessly integrate the PE and VE to have an efficient and effective DT, just like people have imagined in fiction.

REFERENCES

[1] 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.

[2] Sun W, Lei S, Wang L, Liu Z, Zhang Y. Adaptive Federated Learning and Digital Twin for Industrial Internet of Things. IEEE Trans Ind Inf 2021;17:5605–14. https://doi.org/10.1109/TII.2020.3034674.

[3] Broo DG, Schooling J. Digital twins in infrastructure: definitions, current practices, challenges and strategies. International Journal of Construction Management 2023;23:1254–63. https://doi.org/10.1080/15623599.2021.1966980.

[4] Maulik S, Riordan D, Walsh J. Dynamic Reduction-Based Virtual Models for Digital Twins—A Comparative Study. Applied Sciences 2022;12:7154. https://doi.org/10.3390/app12147154.

[5] Zhang L, Zhou L, Horn BKP. Building a right digital twin with model engineering. Journal of Manufacturing Systems 2021;59:151–64. https://doi.org/10.1016/j.jmsy.2021.02.009.

[6] VanDerHorn E, Mahadevan S. Digital Twin: Generalization, characterization and implementation. Decision Support Systems 2021;145:113524. https://doi.org/10.1016/j.dss.2021.113524.

[7] Smogeli Ø. FEATURE FEBRUARY 2017 n.d.

[8] Pedersen TA, Glomsrud JA, Ruud E-L, Simonsen A, Sandrib J, Eriksen B-OH. Towards simulation-based verification of autonomous navigation systems. Safety Science 2020;129:104799. https://doi.org/10.1016/j.ssci.2020.104799.

[9] Qi Q, Tao F. Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison. IEEE Access 2018;6:3585–93. https://doi.org/10.1109/ACCESS.2018.2793265.

[10] 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.

[11] Liu J, Ji Q, Zhou H, Du C, Liu X, Li M. A multi-dimensional evolution modeling method for digital twin process model. Robotics and Computer-Integrated Manufacturing 2024;86:102667. https://doi.org/10.1016/j.rcim.2023.102667.

[12] Liu X, Jiang D, Tao B, Xiang F, Jiang G, Sun Y, et al. A systematic review of digital twin about physical entities, virtual models, twin data, and applications. Advanced Engineering Informatics 2023;55:101876. https://doi.org/10.1016/j.aei.2023.101876.

[13] Botín-Sanabria DM, Mihaita A-S, Peimbert-García RE, Ramírez-Moreno MA, Ramírez-Mendoza RA, Lozoya-Santos JDJ. Digital Twin Technology Challenges and Applications: A Comprehensive Review. Remote Sensing 2022;14:1335. https://doi.org/10.3390/rs14061335.

[14] Ferko E, Bucaioni A, Behnam M. Architecting Digital Twins. IEEE Access 2022;10:50335–50. https://doi.org/10.1109/ACCESS.2022.3172964.

[15] Mihai S, Yaqoob M, Hung DV, Davis W, Towakel P, Raza M, et al. Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects. IEEE Commun Surv Tutorials 2022;24:2255–91. https://doi.org/10.1109/COMST.2022.3208773.

[16] Schluse M, Priggemeyer M, Atorf L, Rossmann J. Experimentable Digital Twins—Streamlining Simulation-Based Systems Engineering for Industry 4.0. IEEE Trans Ind Inf 2018;14:1722–31. https://doi.org/10.1109/TII.2018.2804917.

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

[18] Yang X, Ran Y, Zhang G, Wang H, Mu Z, Zhi S. A digital twin-driven hybrid approach for the prediction of performance degradation in transmission unit of CNC machine tool. Robotics and Computer-Integrated Manufacturing 2022;73:102230. https://doi.org/10.1016/j.rcim.2021.102230.

[19] Semeraro C, Lezoche M, Panetto H, Dassisti M. Digital twin paradigm: A systematic literature review. Computers in Industry 2021;130:103469. https://doi.org/10.1016/j.compind.2021.103469.

[20] Rasheed A, San O, Kvamsdal T. Digital Twin: Values, Challenges and Enablers From a Modeling Perspective. IEEE Access 2020;8:21980–2012. https://doi.org/10.1109/ACCESS.2020.2970143.

[21] Rasheed A, San O, Kvamsdal T. Digital Twin: Values, Challenges and Enablers 2019.

[22] Wright L, Davidson S. How to tell the difference between a model and a digital twin. Adv Model and Simul in Eng Sci 2020;7:13. https://doi.org/10.1186/s40323-020-00147-4.

[23] Jones D, Snider C, Nassehi A, Yon J, Hicks B. Characterising the Digital Twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology 2020;29:36–52. https://doi.org/10.1016/j.cirpj.2020.02.002.

[24] Fuller A, Fan Z, Day C, Barlow C. Digital Twin: Enabling Technologies, Challenges and Open Research. IEEE Access 2020;8:108952–71. https://doi.org/10.1109/ACCESS.2020.2998358.

[25] Boje C, Guerriero A, Kubicki S, Rezgui Y. Towards a semantic Construction Digital Twin: Directions for future research. Automation in Construction 2020;114:103179. https://doi.org/10.1016/j.autcon.2020.103179.

[26] Zheng Y, Yang S, Cheng H. An application framework of digital twin and its case study. J Ambient Intell Human Comput 2019;10:1141–53. https://doi.org/10.1007/s12652-018-0911-3.

[27] Tao F, Zhang H, Liu A, Nee AYC. Digital Twin in Industry: State-of-the-Art. IEEE Transactions on Industrial Informatics 2019;15:2405–15. https://doi.org/10.1109/TII.2018.2873186.

[28] Tao F, Zhang M, Nee AYC. Chapter 6 - Cyber–Physical Fusion in Digital Twin Shop-Floor. In: Tao F, Zhang M, Nee AYC, editors. Digital Twin Driven Smart Manufacturing, Academic Press; 2019, p. 125–39. https://doi.org/10.1016/B978-0-12-817630-6.00006-0.

[29] Tao F, Zhang M, Liu Y, Nee AYC. Digital twin driven prognostics and health management for complex equipment. CIRP Annals 2018;67:169–72. https://doi.org/10.1016/j.cirp.2018.04.055.

[30] Tao F, Cheng J, Qi Q, Zhang M, Zhang H, Sui F. Digital twin-driven product design, manufacturing and service with big data. Int J Adv Manuf Technol 2018;94:3563–76. https://doi.org/10.1007/s00170-017-0233-1.

[31] Tao F, Xiao B, Qi Q, Cheng J, Ji P. Digital twin modeling. Journal of Manufacturing Systems 2022;64:372–89. https://doi.org/10.1016/j.jmsy.2022.06.015.

[32] Qi Q, Tao F, Hu T, Anwer N, Liu A, Wei Y, et al. Enabling technologies and tools for digital twin. Journal of Manufacturing Systems 2019:S027861251930086X. https://doi.org/10.1016/j.jmsy.2019.10.001.

[33] Copyright. Digital Twin Driven Smart Manufacturing, Elsevier; 2019, p. iv. https://doi.org/10.1016/B978-0-12-817630-6.00014-X.

[34] Lu Y, Liu C, Wang KI-K, Huang H, Xu X. Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. Robotics and Computer-Integrated Manufacturing 2020;61:101837. https://doi.org/10.1016/j.rcim.2019.101837.

[35] Bevilacqua M, Bottani E, Ciarapica FE, Costantino F, Di Donato L, Ferraro A, et al. Digital Twin Reference Model Development to Prevent Operators' Risk in Process Plants. Sustainability 2020;12:1088. https://doi.org/10.3390/su12031088.

[36] Darvishi H, Ciuonzo D, Eide ER, Rossi PS. Sensor-Fault Detection, Isolation and Accommodation for Digital Twins via Modular Data-Driven Architecture. IEEE Sensors Journal 2021;21:4827–38. https://doi.org/10.1109/JSEN.2020.3029459.

[37] Selvaraj P, Radhakrishnan P, Adithan M. An integrated approach to design for manufacturing and assembly based on reduction of product development time and cost. The International Journal of Advanced Manufacturing Technology 2009;42:13–29. https://doi.org/10.1007/s00170-008-1580-8.

[38] Mortlock T, Muthirayan D, Yu S-Y, Khargonekar PP, Abdullah Al Faruque M. Graph Learning for Cognitive Digital Twins in Manufacturing Systems. IEEE Trans Emerg Topics Comput 2022;10:34–45. https://doi.org/10.1109/TETC.2021.3132251.

[39] Ramonell C, Chacón R, Posada H. Knowledge graph-based data integration system for digital twins of built assets. Automation in Construction 2023;156:105109. https://doi.org/10.1016/j.autcon.2023.105109.

[40] Sahlab N, Kamm S, Muller T, Jazdi N, Weyrich M. Knowledge Graphs as Enhancers of Intelligent Digital Twins. 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), Victoria, BC, Canada: IEEE; 2021, p. 19–24. https://doi.org/10.1109/ICPS49255.2021.9468219.

[41] Jalving J, Cao Y, Zavala VM. Graph-based modeling and simulation of complex systems. Computers & Chemical Engineering 2019;125:134–54. https://doi.org/10.1016/j.compchemeng.2019.03.009.

[42] Lambiotte R, Rosvall M, Scholtes I. From networks to optimal higher-order models of complex systems. Nat Phys 2019;15:313–20. https://doi.org/10.1038/s41567-019-0459-y.

[43] Kurtoglu T, Tumer IY. A Graph-Based Fault Identification and Propagation Framework for Functional Design of Complex Systems. Journal of Mechanical Design 2008;130:051401. https://doi.org/10.1115/1.2885181.

[44] Kuhn M, Franke J. Data continuity and traceability in complex manufacturing systems: a graph-based modeling approach. International Journal of Computer Integrated Manufacturing 2021;34:549–66. https://doi.org/10.1080/0951192X.2021.1901320.

[45] Xia L, Zheng P, Li X, Gao RobertX, Wang L. Toward cognitive predictive maintenance: A survey of graph-based approaches. Journal of Manufacturing Systems 2022;64:107–20. https://doi.org/10.1016/j.jmsy.2022.06.002.

[46] Weise J, Benkhardt S, Mostaghim S. A Survey on Graph-based Systems in Manufacturing Processes. 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 2018, p. 112–9. https://doi.org/10.1109/SSCI.2018.8628683.

[47] Fortunato S, Hric D. Community detection in networks: A user guide. Physics Reports 2016;659:1–44. https://doi.org/10.1016/j.physrep.2016.09.002.

[48] Malliaros FD, Vazirgiannis M. Clustering and community detection in directed networks: A survey. Physics Reports 2013;533:95–142. https://doi.org/10.1016/j.physrep.2013.08.002.

[49] Community Detection in Multi-Layer Graphs. SIGMOD Record 2015;44.

[50] Scanagatta M, Salmerón A, Stella F. A survey on Bayesian network structure learning from data. Prog Artif Intell 2019;8:425–39. https://doi.org/10.1007/s13748-019-00194-y.

[51] Maier M, Zwicker R, Akbari M, Rupenyan A, Wegener K. Bayesian optimization for autonomous process set-up in turning. CIRP Journal of Manufacturing Science and Technology 2019. https://doi.org/10.1016/j.cirpj.2019.04.005.

[52] Khodakarami V, Abdi A. Project cost risk analysis: A Bayesian networks approach for modeling dependencies between cost items. International Journal of Project Management 2014;32:1233–45. https://doi.org/10.1016/j.ijproman.2014.01.001.

[53] Park N, Kan A, Dong XL, Zhao T, Faloutsos C. Estimating Node Importance in Knowledge Graphs Using Graph Neural Networks. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '19 2019:596–606. https://doi.org/10.1145/3292500.3330855.

[54] Scarselli F, Gori M, Ah Chung Tsoi, Hagenbuchner M, Monfardini G. The Graph Neural Network Model. IEEE Trans Neural Netw 2009;20:61–80. https://doi.org/10.1109/TNN.2008.2005605.

[55] 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.

[56] Conrady S, Jouffe L. Bayesian networks and BayesiaLab: a practical introduction for researchers. 2015.

[57] SamIam n.d.

[58] Broenink JF. Introduction to Physical Systems Modelling with Bond Graphs 1999:31.

[59] Borutzky W, editor. Bond Graph Modelling of Engineering Systems: Theory, Applications and Software Support. New York, NY: Springer New York; 2011. https://doi.org/10.1007/978-1-4419-9368-7.

[60] Gawthrop PJ, Bevan GP. A TUTORIAL INTRODUCTION FOR CONTROL ENGINEERS n.d.

[61] AMESim https://plm.sw.siemens.com/en-US/simcenter/systems-simulation/amesim/ n.d.

[62] Low CB, Wang D, Arogeti S, Zhang JB. Causality Assignment and Model Approximation for Hybrid Bond Graph: Fault Diagnosis Perspectives. IEEE Transactions on Automation Science and Engineering 2010;7:570–80. https://doi.org/10.1109/TASE.2009.2026731.

[63] Zurawski R, Zhou M. Petri nets and industrial applications: A tutorial. IEEE Trans Ind Electron 1994;41:567–83. https://doi.org/10.1109/41.334574.

[64] Holloway LE. A Survey of Petri Net Methods for Controlled Discrete Event Systems n.d.

[65] Kleijnen JPC. Kriging metamodeling in simulation: A review. European Journal of Operational Research 2009;192:707–16. https://doi.org/10.1016/j.ejor.2007.10.013.

[66] Hussain MF, Barton RR, Joshi SB. Metamodeling: Radial basis functions, versus polynomials. European Journal of Operational Research 2002;138:142–54. https://doi.org/10.1016/S0377-2217(01)00076-5.

[67] Chambers M, Mount-Campbell CA. Process optimization via neural network metamodeling. International Journal of Production Economics 2002;79:93–100. https://doi.org/10.1016/S0925-5273(00)00188-2.

[68] Shan S, Wang GG. Metamodeling for High Dimensional Simulation-Based Design Problems. J Mech Des 2010;132:051009. https://doi.org/10.1115/1.4001597.

[69] Coatanea E, Roca R. Dimensional Analysis Conceptual Modeling Supporting Adaptable Reasoning in simulation-based training. 2018 13th Annual Conference on System of Systems Engineering (SoSE), Paris: IEEE; 2018, p. 245–52. https://doi.org/10.1109/SYSOSE.2018.8428785.

[70] Mokhtarian H, Coatanéa E, Paris H, Ritola T, Ellman A, Vihinen J, et al. A Network Based Modelling Approach Using the Dimensional Analysis Conceptual Modeling (DACM) Framework for Additive Manufacturing Technologies. Volume 1A: 36th Computers and Information in Engineering Conference, Charlotte, North Carolina, USA: ASME; 2016, p. V01AT02A046. https://doi.org/10.1115/DETC2016-60473.

[71] Coatanéa E, Nagarajan HPN, Panicker S, Prod'hon R, Mokhtarian H, Chakraborti A, et al. Systematic manufacturability evaluation using dimensionless metrics and singular value decomposition: a case study for additive manufacturing. Int J Adv Manuf Technol 2020. https://doi.org/10.1007/s00170-020-06158-0.

[72] Coatanéa E, Nonsiri S, Ritola T, Tumer IY, Jensen DC. A Framework for Building Dimensionless Behavioral Models to Aid in Function-Based Failure Propagation Analysis. Journal of Mechanical Design 2011;133:121001. https://doi.org/10.1115/1.4005230.

[73] Coatanéa E, Roca R, Mokhtarian H, Mokammel F, Ikkala K. A Conceptual Modeling and Simulation Framework for System Design. Computing in Science Engineering 2016;18:42–52. https://doi.org/10.1109/MCSE.2016.75.

[74] Mokhtarian H, Coatanéa E, Paris H, Mbow MM, Pourroy F, Marin PR, et al. A Conceptual Design and Modeling Framework for Integrated Additive Manufacturing. Journal of Mechanical Design 2018;140:081101. https://doi.org/10.1115/1.4040163.

[75] Wu D, Coatanéa E, Wang G. EMPLOYING KNOWLEDGE ON CAUSAL RELATIONSHIP TO ASSIST MULTIDISCIPLINARY DESIGN OPTIMIZATION. Journal of Mechanical Design 2018. https://doi.org/10.1115/1.4042342.

[76] Chickering DM. Optimal Structure Identification With Greedy Search n.d.:48.

[77] Hauser A. Characterization and Greedy Learning of Interventional Markov Equivalence Classes of Directed Acyclic Graphs n.d.:56.

[78] Bretto A. Hypergraph Theory: An Introduction. Heidelberg: Springer International Publishing; 2013. https://doi.org/10.1007/978-3-319-00080-0.

[79] Khalil W, Merzouki R, Ould-Bouamama B, Haffaf H. Hypergraph Models for System of Systems Supervision Design. IEEE Trans Syst, Man, Cybern A 2012;42:1005–12. https://doi.org/10.1109/TSMCA.2012.2183350.

[80] Carletti T, Fanelli D, Nicoletti S. Dynamical systems on hypergraphs. J Phys Complex 2020;1:035006. https://doi.org/10.1088/2632-072X/aba8e1.

[81] Suo Q, Guo J-L, Sun S, Liu H. Exploring the evolutionary mechanism of complex supply chain systems using evolving hypergraphs. Physica A: Statistical Mechanics and Its Applications 2018;489:141–8. https://doi.org/10.1016/j.physa.2017.08.002.

[82] von Luxburg U. A tutorial on spectral clustering. Stat Comput 2007;17:395–416. https://doi.org/10.1007/s11222-007-9033-z.

[83] Jia H, Ding S, Xu X, Nie R. The latest research progress on spectral clustering. Neural Comput & Applic 2014;24:1477–86. https://doi.org/10.1007/s00521-013-1439-2.

[84] Ghojogh B, Ghodsi A, Karray F, Crowley M. Laplacian-Based Dimensionality Reduction Including Spectral Clustering, Laplacian Eigenmap, Locality Preserving Projection, Graph Embedding, and Diffusion Map: Tutorial and Survey 2022.

[85] Schaeffer SE. Graph clustering. Computer Science Review 2007;1:27–64. https://doi.org/10.1016/j.cosrev.2007.05.001.

[86] Qian G, Sural S, Gu Y, Pramanik S. Similarity between Euclidean and cosine angle distance for nearest neighbor queries. Proceedings of the 2004 ACM symposium on Applied computing, Nicosia Cyprus: ACM; 2004, p. 1232–7. https://doi.org/10.1145/967900.968151.

[87] Chung F. Laplacians and the Cheeger Inequality for Directed Graphs. Ann Comb 2005;9:1–19. https://doi.org/10.1007/s00026-005-0237-z.

[88] Chen D, Lü L, Shang M-S, Zhang Y-C, Zhou T. Identifying influential nodes in complex networks. Physica A: Statistical Mechanics and Its Applications 2012;391:1777–87. https://doi.org/10.1016/j.physa.2011.09.017.

[89] Hu P, Fan W, Mei S. Identifying node importance in complex networks. Physica A: Statistical Mechanics and Its Applications 2015;429:169–76. https://doi.org/10.1016/j.physa.2015.02.002.

[90] Henni K, Mezghani N, Gouin-Vallerand C. Unsupervised graph-based feature selection via subspace and pagerank centrality. Expert Systems with Applications 2018;114:46–53. https://doi.org/10.1016/j.eswa.2018.07.029.

[91] Ma Z, Shen C, Liu F, Mei S. Fast Screening of Vulnerable Transmission Lines in Power Grids: A PageRank-Based Approach. IEEE Trans Smart Grid 2019;10:1982–91. https://doi.org/10.1109/TSG.2017.2785267.

[92] 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.

[93] Mokammel F, Coatanea E, Bakhouya M, Christophe F, Nonsiri S. Impact analysis of graph-based requirements models using PageRank algorithm. 2013 IEEE International Systems Conference (SysCon), Orlando, FL: IEEE; 2013, p. 731–6. https://doi.org/10.1109/SysCon.2013.6549964.

[94] Jiang B. Ranking spaces for predicting human movement in an urban environment. International Journal of Geographical Information Science 2009;23:823–37. https://doi.org/10.1080/13658810802022822.

[95] Bonacich P, Lloyd P. Eigenvector-like measures of centrality for asymmetric relations. Social Networks 2001;23:191–201. https://doi.org/10.1016/S0378-8733(01)00038-7.

[96] Yan E, Ding Y. Discovering author impact: A PageRank perspective. Information Processing & Management 2011;47:125–34. https://doi.org/10.1016/j.ipm.2010.05.002.

[97] Saxena A, Iyengar S. Centrality Measures in Complex Networks: A Survey 2020.

[98] 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.

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

[100] Ghosh N, Paul R, Maity S, Maity K, Saha S. Fault Matters: Sensor data fusion for detection of faults using Dempster–Shafer theory of evidence in IoT-based applications. Expert Systems with Applications 2020;162:113887. https://doi.org/10.1016/j.eswa.2020.113887.

[101] 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.

[102] 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.

[103] Xiao F. Generalization of Dempster–Shafer theory: A complex mass function. Appl Intell 2020;50:3266–75. https://doi.org/10.1007/s10489-019-01617-y.

[104] Chakraborti A, Heininen A, Väänänen S, Koskinen KT, Vainio H. Evidential Reasoning based Digital Twins for Performance Optimization of Complex Systems. Procedia CIRP 2021;104:618–23. https://doi.org/10.1016/j.procir.2021.11.104.

[105] Ramsey J, Glymour M, Sanchez-Romero R, Glymour C. A million variables and more: the Fast Greedy Equivalence Search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. Int J Data Sci Anal 2017;3:121–9. https://doi.org/10.1007/s41060-016-0032-z.

[106] Sempler https://sempler.readthedocs.io/en/latest/ n.d.

[107] von Luxburg U. A tutorial on spectral clustering. Stat Comput 2007;17:395–416. https://doi.org/10.1007/s11222-007-9033-z.

[108] Kang Z, Shi G, Huang S, Chen W, Pu X, Zhou JT, et al. Multi-graph fusion for multi-view spectral clustering. Knowledge-Based Systems 2020;189:105102. https://doi.org/10.1016/j.knosys.2019.105102.

[109] Rafailidis D, Constantinou E, Manolopoulos Y. Landmark selection for spectral clustering based on Weighted PageRank. Future Generation Computer Systems 2017;68:465–72. https://doi.org/10.1016/j.future.2016.03.006.

[110] 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.

[111] pyds https://pyds.sohi.link/ n.d.

[112] 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;93:240–5. https://doi.org/10.1016/j.procir.2020.04.050.

[113] Heininen A, Prod'hon R, Mokhtarian H, Coatanéa E, Koskinen K. Finite element modelling of temperature in cylindrical grinding for future integration in a digital twin. Procedia CIRP 2021;104:875–80. https://doi.org/10.1016/j.procir.2021.11.147.

[114] ma Y, yang J, li B, lu J. An analytical model of grinding force based on time-varying dynamic behavior. Int J Adv Manuf Technol 2017;89:2883–91. https://doi.org/10.1007/s00170-016-9751-5.

[115] Wu D, Jennings C, Terpenny J, Gao RX, Kumara S. A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests. Journal of Manufacturing Science and Engineering 2017;139:071018. https://doi.org/10.1115/1.4036350.

[116] Badger J, Murphy S, O'Donnell GE. Acoustic emission in dressing of grinding wheels: AE intensity, dressing energy, and quantification of dressing sharpness and increase in diamond wear-flat size. International Journal of Machine Tools and Manufacture 2018;125:11–9. https://doi.org/10.1016/j.ijmachtools.2017.11.007.

[117] Sciarra C, Chiarotti G, Laio F, Ridolfi L. A change of perspective in network centrality. Sci Rep 2018;8. https://doi.org/10.1038/s41598-018-33336-8.

[118] Slowik A, Slowik J. Multi-objective optimization of surface grinding process with the use of evolutionary algorithm with remembered Pareto set. Int J Adv Manuf Technol 2008;37:657–69. https://doi.org/10.1007/s00170-007-1013-0.

[119] Yoon SY, Lin Z, Allaire PE. Control of Surge in Centrifugal Compressors by Active Magnetic Bearings. London: Springer London; 2013. https://doi.org/10.1007/978-1-4471-4240-9.

[120] Giarré L, Bauso D, Falugi P, Bamieh B. LPV model identification for gain scheduling control: An application to rotating stall and surge control problem. Control Engineering Practice 2006;14:351–61. https://doi.org/10.1016/j.conengprac.2005.01.013.

[121] NetworkX https://networkx.org/ n.d.

[122] Grömping U. Variable Importance Assessment in Regression: Linear Regression versus Random Forest. The American Statistician 2009;63:308–19. https://doi.org/10.1198/tast.2009.08199.

[123] Choi K, Fazekas G, Sandler M, Cho K. Convolutional recurrent neural networks for music classification. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, p. 2392–6. https://doi.org/10.1109/ICASSP.2017.7952585.

[124] Breiman L. Random Forests. Machine Learning 2001;45:5–32. https://doi.org/10.1023/A:1010933404324.

[125] Menze BH, Kelm BM, Masuch R, Himmelreich U, Bachert P, Petrich W, et al. A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. BMC Bioinformatics 2009;10:213. https://doi.org/10.1186/1471-2105-10-213.

[126] Altmann A, Toloşi L, Sander O, Lengauer T. Permutation importance: a corrected feature importance measure. Bioinformatics 2010;26:1340–7. https://doi.org/10.1093/bioinformatics/btq134.

APPENDIX A

Mathematical model for Surge graph modeling based on Greitzer Compression model

Greitzer compression system model

$$\psi = \frac{\Delta p}{\frac{1}{2}\rho_{o1}U^2} \qquad \text{.......(17)}$$

$$\phi = \frac{m}{\rho_{o1}\, U\, A_c} \qquad \text{.......(18)}$$

Helmholtz frequency (ω_H)

$$\omega_H = a_{o1}\sqrt{\frac{A_c}{V_p\, L_c}} \qquad \text{......(19)}$$

Original Greitzer compression system model with non-dimensional variables

$$\left.\begin{aligned}\frac{d\phi_c}{dt} &= B\omega_H(\psi_c - \psi_p)\\ \frac{d\phi_{th}}{dt} &= \frac{B\omega_H}{G}(\psi_p - \psi_{th})\end{aligned}\right] \textit{conservation of momentum of fluid in compressor and throttle duct}$$

$$\left.\frac{d\psi_p}{dt} = \frac{\omega_H}{B}(\phi_c - \phi_{th})\right] \textit{conservation of mass in plenum volume} \qquad \text{....(20)}$$

$$\left.\frac{d\psi_c}{dt} = \frac{\omega_H}{\tilde{\tau}}(\psi_{c,ss} - \psi_c)\right] \textit{behavior of dynamic compressor settling}$$

Greitzer stability parameter (*B*) governs the intensity of surge instability in Greitzer model

$$B = \frac{U}{2\omega_H L_c}$$

$$G = \frac{L_{th}A_c}{L_c A_{th}} \qquad \text{......(21)}$$

For subsonic flows

$$\phi_{th} = c_{th} u_{th} \sqrt{\psi_p} \qquad ......(22)$$

Curve fitting to determine steady state pressure and flow rate measurements.

$$\psi_c(\phi) = \psi_{c0} + \tilde{H}\left(1 + \frac{3}{2}\left(\frac{\phi}{W} - 1\right) - \frac{1}{2}\left(\frac{\phi}{W} - 1\right)^3\right) \qquad ......(23)$$

ψ_{c0} : pressure @ 0 flow

$\tilde{H}$, W: constant computed from pressure rise and flow rate corresponding to surge point (predicted params from curve fitting in the stable region and used to correct ψ_{c0})

Variation of impeller tip clearance with AMB. Isentropic efficiency of compressor

$$\eta_{th} = \frac{T_{o1} C_p \left(\left(\frac{p_{c,ss}}{p_{o1}} \right)^{\frac{\gamma - 1}{\gamma}} - 1 \right)}{\Delta h_{oc,ideal}} \qquad ......(24)$$

ψ_c (non-dimensional compressor pressure rise) as a function of $\psi_{c,ss}$ and δ_{cl}

$$\psi_c = \frac{p_{o1}}{\frac{1}{2}\rho_{o1} U^2} \left(\left(1 + \frac{\left(\frac{\frac{1}{2}\rho_{o1} U^2}{p_{o1}} \psi_{c,ss} + 1 \right)^{\frac{\gamma}{\gamma - 1}} - 1}{1 - k_0 \frac{\delta_{cl}}{b_2}} \right)^{\frac{\gamma}{\gamma - 1}} - 1 \right) \qquad ....(25)$$

Level I stability analysis (initial screening to identify safe compressor operations)

$$q_A = HP \frac{B_C C \rho_d}{D_C H_C N \rho_s} \qquad ....(26)$$

APPENDIX B

Mathematical model for Grinding wheel wear graph modeling based on Jaeger’s analytical model.

Grinding time: $t_c = \frac{t_b}{n_b}$ … (27)

Equivalent diameter: $d_e = \frac{d_s \times d_w}{d_s \pm d_w}$ … (28)

Uncut chip thickness: $h_{cu} = \sqrt{\frac{v_w}{v_s} \times \frac{1}{C \times r} \sqrt{\frac{a_e}{d_e}} h_{cu}}$ … (29)

Equivalent chip thickness: $h_{eq} = \frac{a_e \times v_w}{v_s}$ … (30)

Grinding force: $f_g \propto h_{cu}^{1.7} \propto \left\{ \left\{ \frac{v_w}{v_s} \times \frac{1}{C.r} \right\} \sqrt{\frac{a_e}{d_e}} \right\}^{0.85}$ … (31)

Specific grinding energy: $e_c \propto \frac{1}{h_{Cu}^{n}} \propto \sqrt{\left(\frac{v_s}{v_w} .C.r \times \sqrt{\frac{d_e}{a_e}} \right)}$ … (32)

Material removal rate: $Q' = Q_w^{'} = a_e v_w$ … (33)

Grinding power: $P = e_c Q' b_w$ … (34)

Tangential force: $F_t = \frac{P}{v_s} = \frac{e_c Q' b_w}{v_s}$ … (35)

Grinding coefficient: $\mu = \frac{F_t}{F_n}$ …

(36)

Roughness factor: $R_t \propto \frac{h_{Cu}^{\frac{4}{3}}}{a_e^{\frac{1}{3}}} \approx \left(\frac{v_w}{v_s} \times \frac{1}{C.r.\sqrt{d_e}} \right)^{\frac{2}{3}}$ …(37)

Geometric contact length: $l_g = \sqrt{a_e.d_e}$ … (38)

$\wedge = \frac{Q'}{F_n}$ … (39)

Real contact length: $l_c = \sqrt{l_g^2 + l_f^2}$ … (40)

Deflection contact length: $l_f^2 = \frac{8R_r^2 F_n' d_e}{\pi E^*} l$ … (41)

Combined elastic modulus: $\frac{1}{E^*} = \frac{1-v_1^2}{E_1} + \frac{1-v_2^2}{E_2}$ … (42)

Volume of wheel wear per unit wheel width:

$V_S = L_c.N_s.t$ … (43)

Real area of contact: $A_r = \frac{F_n}{H}$ … (44)

Volume of wheel wear: $V = K\frac{F_n}{H} L\_s$ …(45)

PUBLICATIONS

PUBLICATION I

A Dimension Reduction Method for Efficient Optimization of Manufacturing Performance

Ananda Chakraborti, Hari P. N. Nagarajan, Suraj Panicker, Hossein Mokhtarian, Eric Coatanéa, Kari T. Koskinen

29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2019), June 24-28, 2019, Limerick, Ireland.
https://doi.org/10.1016/j.promfg.2020.01.070

Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 38 (2019) 556–563

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2019), June 24-28, 2019, Limerick, Ireland.

A Dimension Reduction Method for Efficient Optimization of Manufacturing Performance

Ananda Chakraborti[a]*, Hari P. N. Nagarajan[a], Suraj Panicker[a], Hossein Mokhtarian[a], Eric Coatanéa[a], Kari T. Koskinen[a]

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

Abstract

Increased competitiveness in the manufacturing industry demands optimizing performance at each level of an enterprise. Optimizing performance in terms of indicators such as manufacturing cost requires knowledge of cost-inducing variables from product design and manufacturing, and optimization of these variables. However, the number of variables that affect manufacturing cost is very high and optimizing all variables is time intensive and computationally difficult. Thus, it is important to identify and optimize select few variables that have high potential for inducing cost. Towards that goal, a dimension reduction method combining dimensional analysis conceptual modelling framework and graph centrality theory is proposed. The proposed method integrates existing knowledge of the cost inducing variables, their interactions, and input-output relationship for different functions or behavior of a system, in the form of a causal graph. Propagation of optimization objectives in the causal graph is checked to identify contradictory influences on the variables in the graph. Following the contradiction analysis, graph centrality theory is used to rank the different regions within the graph based on their relative importance to the optimization problem and to cluster the variables into two optimization groups namely, less important variables and most important variables relative to optimizing cost. The optimization problem is formulated to fix less important variables at their highest or lowest levels based on their interaction to cost and to optimize the more important variables to minimize cost. The proposed dimension reduction method is demonstrated for an optimization problem, to minimize the production cost of the bladder and key mechanism for a high-field superconducting magnet at CERN, capable of producing a 16 Tesla magnetic field. It was found that the graph region representing the electromagnetic force and resultant stress generated during energizing of the magnet ranked highest for influence on the bladder and key manufacturing cost. An optimization of the stress and its associated variables to minimize the manufacturing cost is performed using a genetic algorithm solver in Matlab.

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

10.1016/j.promfg.2020.01.070

Keywords: Computer integrated manufacturing; Conceptual modelling; Dimension reduction; Multi objective optimization

1. Introduction

Continuous improvement has become means of subsistence for a successful manufacturing enterprise. In recent times, the growth of information and communications technology (ICT) and its integration into manufacturing has further enabled businesses to fulfill customer’s needs in an economical and sustainable manner. To remain competitive in the market, manufacturers must ensure optimized utilization of resources, energy and consumables, as well as prolonged life of the product. Thus, optimization of a process or product design for performance improvement is crucial for an enterprise. Manufacturers measure performance with the help of key performance indicators such as cost, quality, productivity, etc. These indicators are often functions of design and manufacturing variables that need to be modelled and optimized. Hence, sufficient knowledge about the product design and manufacturing process is needed to model their influence on performance indicators. Difficulty arises when trying to model these performance indicators, which are inherently complex and span across multiple domains including product behavior, manufacturing process physics, and production planning. Moreover, optimizing a performance indicator require manufacturers to find optimal values for all influencing variables. Choosing optimal values for all variables irrespective of their level of influence on performance is computationally cost intensive. Dimension reduction in the field of design optimization or machine learning is not a new phenomenon. Nevertheless, there is a lack of systematic techniques that embed domain knowledge in manufacturing performance optimization models. Thus, a multi-domain approach is required with the help of conceptual modelling and simulation to optimize performance indicators. Such approaches would help manufacturers 1) to characterize interrelationships between variables in a system to model performance, 2) integrate knowledge of different domains for modelling performance indicators, and 3) to reduce the number of variables that are needed to optimize performance.

This research focuses on developing a new dimension reduction methodology, which combines graph based modelling of performance indicators across different domains using the dimensional analysis conceptual modelling framework (DACM), and variable clustering using the centrality concept in graph theory and network analysis. The method is tested with a case study of manufacturing the key and bladder mechanism for a new superconducting magnet at CERN. The remainder of the manuscript is organized as follows; Section 2 provides a background discussion on conceptual modelling, dimension reduction in optimization, and centrality concept of graph theory and network applications. Section 3 describes the developed methodology applied to the case study and briefly discusses the results of the case study. Finally, Section 4 describes the conclusions of the work and briefly discusses future development efforts.

Nomenclature

B	Magnetic Flux Density (Nm/A) [$MT^{-2}I^{-1}$]
q	Electric charge (C) [IT]
I	Electric current (A) [I]
J	Current density (I/mm^2) [IL^{-2}]
L	Inductance (H) [$ML^2T^{-2}I^{-2}$]
U	Total energy for self-Inductance [ML^2T^{-2}]
n	Number of poles
L_m	Length of the magnet (m) [L]
θ	Angle made by the conductor with the field (degree)
F_L	Lorentz force (N) [MLT^{-2}]
p_m	Magnetic pressure (N/mm^2) [$ML^{-1}T^{-2}$]
μ	Permeability (H m^{-1}) [$MLT^{-2}I^{-2}$]
σ	Normal stress (N/mm^2) [$ML^{-1}T^{-2}$]

σ_v	von Mises stress (N/mm^2) [$ML^{-1}T^{-2}$]
t	Time of current flow (sec) [T]
ε	Normal strain
τ	Shear stress (N/mm^2) [$ML^{-1}T^{-2}$]
δ_l	Elongation of the magnet due to F_L (mm) [L]
d_{key}	Diameter of the stainless steel key (mm) [L]
t_{key}	Thickness of stainless steel key lamination (mm) [L]
A_{key}	Cross-sectional area of the key (mm^2) [L^2]
V_{shell}	Volume of the key (mm^3) [L^3]
ρ_{SS}	Material density of stainless steel [ML^{-3}]
n_{key}	Number of keys
f	Feed rate of milling tool (mm) [LT^{-1}]
v	Velocity of milling tool (mm) [LT^{-1}]
k	Constant representing cost components within low importance variable list X_L
c_t	Tooling cost (€/unit)
c_m	Material cost (€/unit)
c_l	Labour cost (€/unit)
c_o	Overhead cost (€/unit)
t_s	Milling setup time (min)[T]
c_T	Motal cost of manufacturing of keys (€)

2. Background

2.1. Conceptual modelling and simulation

Conceptual modelling is the abstraction of a model from a real or proposed system [1]. DACM framework as a conceptual modelling mechanism originally developed as a specification and verification technique for complex systems, but has been applied to many different cases such as additive manufacturing [2], machine learning [3], and multidisciplinary design optimization [4]. The main aim of DACM is to extract and encode knowledge of different forms (expert literature, empirical/experimental, and equations) in the form of a causal graph. The DACM framework starts with functional modelling of the system and assigning of fundamental variables to the different functions of the model. The functions, associated variables, and representative equations are characterized in the causal graph in the form of the cause-effect relationship between the fundamental variables of the functional model. The mathematical machinery to check propagation of an objective in a causal network is based on the Vashy-Buckingham's Pi (π)-theorem and the dimensional analysis (DA) theory [5], [6]. An adjacency matrix or multiple domain matrix (MDM) can be obtained from the causal network representing multiple domains. A MDM is a systematic extension of the design structure matrix (DSM), popularly used in system decomposition, integration, and design of complex systems. This matrix encodes a rich data structure able to represent knowledge extracted from the multi-domain system variables [7]. In this research, the matrix representation is evaluated qualitatively to check for contradictory influences in the objective imposed by the system variables. The qualitative analysis is followed by a dimension reduction method to rank order the system variables and form the optimization objective function.

2.2. Dimension reduction

High dimensionality is a universal challenge during computational analysis in the field of science and engineering. In manufacturing, computationally expensive and resource demanding optimization methods are needed to simulate high-dimensional problems such as production planning, scheduling, and performance optimization. Shan and Wang [8] provided a survey of popular strategies such as decomposition, which is to break up the optimization problem into simpler and smaller steps, and screening of variables to identify more important and less important variables as potential means to tackle the high dimensionality in engineering problems. In this research, a screening using the graph centrality theory and node ranking is performed to classify variables as either low impact or high impact depending on their influence on the performance objectives. Following the classification, the optimization problem is decomposed into two stages. First, the low impact variables are fixed at their highest or lowest values based on their connection to the target variables in the performance objectives. Second, the optimization is performed for only the high impact variables based on the performance criteria having fixed the low impact variable values. The screening using graph centrality and node ranking is presented in the next subsection 2.3.

2.3. Graph centrality and node ranking

In a graph-based representation of a system consisting of a large number of nodes, the user is often interested to know what the most important nodes are. This helps the user to direct their attention towards that part of the graph (or network) which has the most influence on the system represented. Graph centrality measures are used to rank nodes and find the most influential nodes within a complex network. Its most notable applications include wireless network applications, network traffic reduction, and social media network analysis. Many measures exist for graph centrality, Freeman [9] provides one of the earliest empirically based measures of centrality in complex networks. The author identifies three measures, namely, degree, betweenness, and closeness, which can be used to obtain a score for centrality in a graph. Borgatti and Everett [10] developed a unified framework for measuring centrality scores in complex social networks. They describe centrality as the node's contribution to the cohesiveness of the network. They also provide mathematical expressions for computing the centrality score in complex networks. The method of ranking the different nodes could be automated using a centrality measurement algorithm similar to the PageRank algorithm, proposed by Brin and Page [11] of Google. PageRank is a network ranking method developed to compute ranks of webpages in Google's search engine results. An improved version of this algorithm is used in applications that go beyond search engine ranking, which include impact analysis of graph-based system requirements and graph-based feature selection [12], [13]. In this article, a dimension reduction strategy is developed based on the causal graph representation of the system. Node ranking algorithm along with graph centrality measurements are used to identify most influential variables in the system. The variables in the causal graph are classified into two groups; high ranking/high impact variables and low ranking/low impact variables. Thus, only a smaller subset of the complete variable list that have high impact are used for optimizing the performance objectives, reducing computational cost. The methodology for dimension reduction using node ranking is explained using a case study of manufacturing the key and bladder mechanism for superconducting magnets in Section 3.

3. Combined Conceptual Modelling and Dimension Reduction Methodology

European Center for Nuclear Research (CERN) has been developing prototype designs of superconducting magnets which have the field strength of 16 Tesla (twice as much field strength compared to the current working designs operated in the Large Hadron Collider). In the conceptual design phase, three designs are under consideration for prototype manufacturing; the cosine theta design, the block design, and the common coil design [14]. In these three designs, the magnet and its support structure differ in size, performance, manufacturing, and assembly process. When the superconducting magnet is energized, electromagnetic forces try to expand the coil. The coil itself is unable to support these forces in tension. Hence, to counter these force during operation, and to have good control during assembly, a bladder and key (made of stainless steel, SS) mechanism is proposed by researchers to produce cost effective magnets [15]. The design and manufacturing of the various components which constitute the magnet becomes challenging considering that it is a multi-criteria design optimization problem. Larger the number of design variables, design constraints, material selection requirements, manufacturing parameters, and functional requirements of finished product, the bigger the optimization problem becomes computationally.

In this research, the cosine theta design structure for the magnet is used for the case study. The methodology developed is shown as a three step approach in Figure 1 which include, modelling the system using DACM, dimension reduction approach to find most influential variables, and solving the optimization problem. The DACM framework is used to model the behavior of the magnet (expansion of magnet coil due to electromagnetic force or Lorentz's force) during energizing.

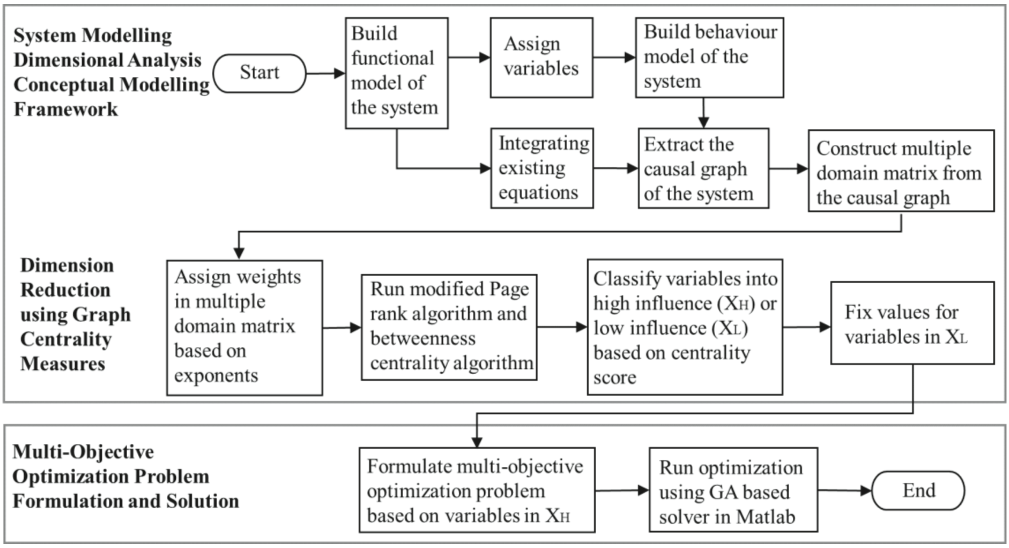

System Modelling Dimensional Analysis Conceptual Modelling Framework
Start
Build functional model of the system
Assign variables
Build behaviour model of the system
Integrating existing equations
Extract the causal graph of the system
Construct multiple domain matrix from the causal graph
Dimension Reduction using Graph Centrality Measures
Assign weights in multiple domain matrix based on exponents
Run modified Page rank algorithm and betweenness centrality algorithm
Classify variables into high influence (XH) or low influence (XL) based on centrality score
Fix values for variables in XL
Multi-Objective Optimization Problem Formulation and Solution
Formulate multi-objective optimization problem based on variables in XH
Run optimization using GA based solver in Matlab
End

Figure 1: Combined conceptual modelling and dimension reduction methodology

The step-by-step conceptual modelling methodology using DACM is explained in earlier research [5]. The electromagnetic force is counterbalanced by the different support structures of the magnet to prevent magnet displacement. For simplification in this case study, it is assumed that the electromagnetic force is counterbalanced force by the pre-stress provided by the bladder and key mechanism. The functional model of the system is built from an abstract concept to a fidelity model based on elementary phenomena from the domain of classical electrodynamics and theory of failure as shown in Figure 2a. The resultant causal graph of the variables developed based on existing equations representing the different functions of the system is represented in Figure 2b. A simplified version of the cosine theta magnet design used in this case study is shown in Figure 2c. The causal graph maps the variables from the functional domain to the technical domain (used by designers to represent design parameters such as the geometry of the product and material), and from technical domain to process domain (used by manufactures to assign process parameters and compute cost of manufacturing). The nodes are color coded as green (independent variables), blue (intermediate variables), black/grey (exogenous variables), and red (target variables). The arcs that connects the nodes represent the interconnection between different variables as well the exponent of that connection from existing equations. A "+" is assigned on the node connection which denote the relationship, if increase in the nth node increases the (n+1)th node and a "–" relationship if the increase in the nth node decreases the (n+1)th node. Next, a multiple domain matrix is developed from the causal graph. The MDM is a sparse, square matrix representation of the system's structure that condenses knowledge of all the variables across the functional, design, and manufacturing process domains with their weights obtained from the causal graph (network). The MDM can be considered as a collection of design structure matrices (DSM) in each domain, mapping variables in the domain to itself, as well as variables from other domains to represent the entire structure of the system. The first column of the MDM consists of all the variables in the system. The MDM is a scalable matrix capable of handling very large structures, where each DSM can contain any finite number of variables. The matrix representation of the system facilitates application of ranking algorithms and graph centrality measures to find the most influential variables of the system.

A weighted PageRank algorithm and betweenness centrality scores are used to rank the nodes in the causal network based on the MDM. The PageRank algorithm computes a probabilistic rank vector that provides an importance estimate of all the nodes in the network based on the in-coming and out-going connections of the nodes in the causal graph. The PageRank algorithm is flexible and it can be modified to handle the weights in the causal network. A rank order for the nodes is obtained from the PageRank algorithm based on the measured centrality score. The betweenness centrality score is also computed for the MDM to validate the results obtained from PageRank algorithm.

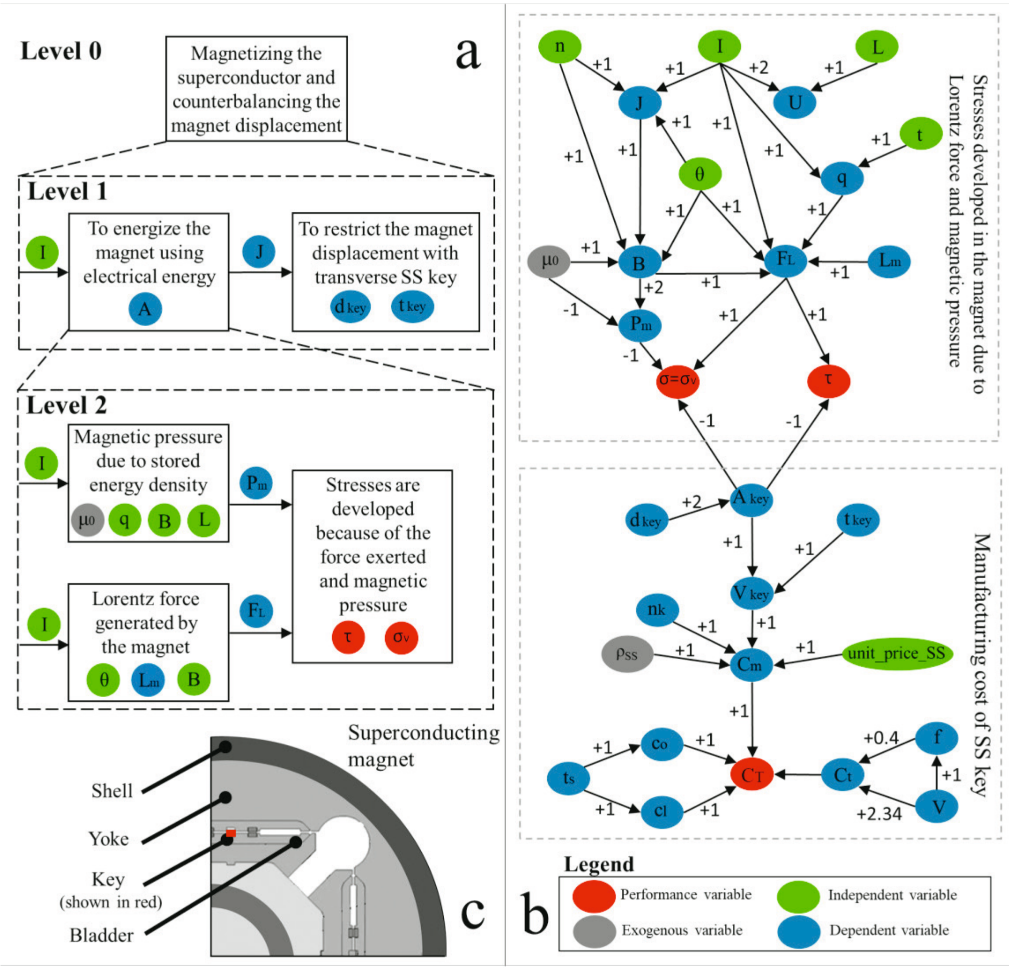

Level 0
Magnetizing the superconductor and counterbalancing the magnet displacement
a
Level 1
I
To energize the magnet using electrical energy
A
J
To restrict the magnet displacement with transverse SS key
d key
t key
Level 2
I
Magnetic pressure due to stored energy density
μ0
q
B
L
Pm
Stresses are developed because of the force exerted and magnetic pressure
τ
σv
I
Lorentz force generated by the magnet
θ
Lm
B
FL
Superconducting magnet
Shell
Yoke
Key
(shown in red)
Bladder
c
n
I
L
J
U
t
θ
q
μ0
B
FL
Lm
Pm
σ=σv
τ
+1
+2
-1
+0.4
+2.34
Stresses developed in the magnet due to Lorentz force and magnetic pressure
A key
d key
t key
V key
nk
ρss
Cm
unit_price_SS
co
ts
cl
CT
Ct
f
V
Manufacturing cost of SS key
b
Legend
Performance variable
Independent variable
Exogenous variable
Dependent variable

Figure 2: (a) Functional model describing expansion of magnet coil due to electromagnetic force; (b) Causal graph depicting magnet behaviour, and (c) Simplified cosine theta magnet design

Unlike PageRank, which is an Eigen vector centrality measure, the betweenness centrality is a measure of influence a node in a network has over the spread of information throughout the network [16]. The score measures the extent to which a node lies in the shortest path between the hub and the objective node. Higher the betweenness score implies the node is more central to the hub and objective nodes. Hence, betweenness score affirms the categorization of nodes as high importance and low importance nodes. The results of the PageRank algorithm and betweenness score are used to classify the variables into matrices X_H (high influence variables) and X_L (low influence variables). The results of weighted PageRank and betweenness scores of the variables are shown in Fig 3. A variable is considered to be of high importance if it has high score in both PageRank and betweenness centrality measures. The ranks are normalized based on the highest ranking node and a threshold of 0.3 is selected based on the distribution of ranks to categorize the variables. The high scoring variables are stored in X_H matrix and low scoring variables are stored in the X_L matrix. The X_L has low impact on the objective hence, the value of the variables are kept constant at its maximum or minimum depending on whether increase in the variable increases or reduces the performance objective (cost function). The variables in X_H matrix, i.e. $X_H = [B, F_L, \tau, \sigma_v, A_{key}, V_{key}, c_t, c_m, C_T]$, are considered for optimization of the total cost of manufacturing of the stainless steel key against the stress induced due to Lorentz force per magnet.

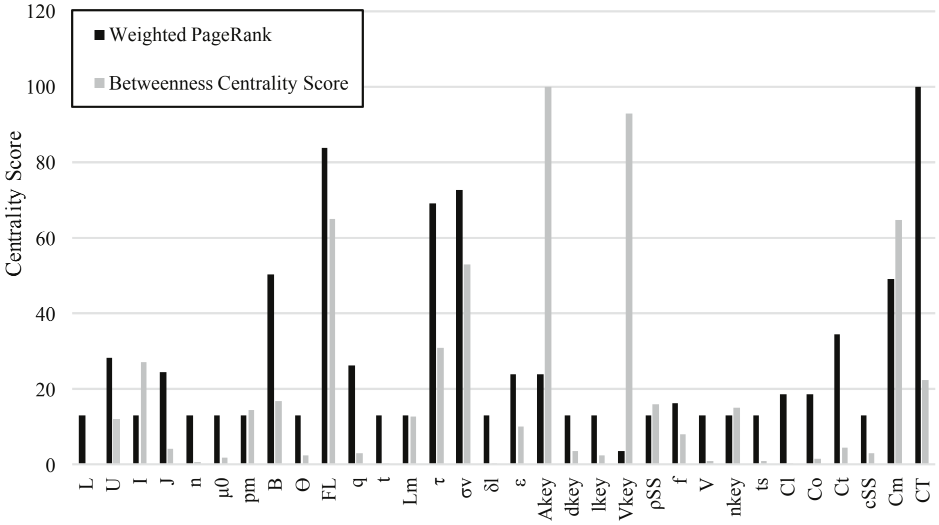


Figure 3: Centrality measure using PageRank algorithm and betweenness centrality score

The optimization problem is then formulated based on the mathematical function for cost of milling the key and the stress generated in the bladder and key mechanism due to the Lorentz force. The data used for optimization can be found in [14], [15]. The objective function for milling the key is as follows:

$$
\begin{aligned}
& f(X_H) \\
& \text{objective 1} \\
& \quad \min\ f(\sigma) \\
& \quad \text{where, } \sigma = \frac{4F_L}{\pi d_{key}{}^2} \\
& \text{objective 2} \\
& \quad \min\ f(C_T) \\
& \quad \text{where } C_T = c_m + c_t \\
& \mathrm{C}_T \approx \frac{1}{4}(\pi \mathrm{d}_{\mathrm{key}}{}^2).\mathrm{l}_{\mathrm{key}}.\rho_{\mathrm{SS}}.\mathrm{unit\_price\_SS} + \mathrm{k} \\
& \text{subjected to, } F_L \leq 12000 \\
& \qquad\qquad\quad l_{key} \leq 890 \\
& \qquad\qquad\quad d_{key} \leq 1.5
\end{aligned}
$$

The variables are; $x_1=d_{key}$, $x_2=l_{key}$, the material is considered as stainless steel, SS304 and having standard values of density, unit price, and standard CNC milling parameters with carbide tools. The optimization is run with 'gamultiobj' (a genetic algorithm based solver in Matlab) solver to find manufacturing cost estimate against various levels of stress during energizing of the magnet using the high influence variables in X_H and for fixed values of low influence variables in X_L. From the optimization, the lowest cost obtained from crossover for machining all keys of a magnet was found to be €13.225,51 for a stress value of 191.083 MPa.

4. Conclusions and Future Work

Dimension reduction enables faster and cheaper optimization in the field of engineering design and manufacturing performance measurement. In this research, a dimension reduction methodology is proposed using the DACM framework and graph centrality theory to estimate and optimize the manufacturing cost of new products in the conceptual design stage. The proposed method decomposes the optimization problem into two steps by categorizing variables into low importance variables whose values are prefixed during optimization step 1, and high importance variables, which are optimized using a solver (step 2). The methodology is demonstrated for optimizing the manufacturing cost of the bladder and key mechanism of a new high field superconducting magnet used by CERN. The two step optimization process reduces the effective number of variables that need to be optimized to get minimum manufacturing cost. The reduced variable list reduces the computation time during optimization and hence enables cheaper and faster optimization. Future research focusses on expanding the case study to include other components of 16 Tesla magnet and validation using real data from CERN's prototype magnet manufacturing.

References

[1] S. Robinson, "Conceptual modeling for simulation: issues and research requirements," presented at the Proceedings of the 2006 winter simulation conference, 2006, pp. 792–800.

[2] H. Mokhtarian *et al.*, "A Network Based Modelling Approach Using the Dimensional Analysis Conceptual Modeling (DACM) Framework for Additive Manufacturing Technologies," in *Proceedings of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Charlotte, NC, 2016, p. V01AT02A046 (9 p.).

[3] H. P. N. Nagarajan *et al.*, "Knowledge-Based Design of Artificial Neural Network Topology for Additive Manufacturing Process Modeling: A New Approach and Case Study for Fused Deposition Modeling," *Journal of Mechanical Design*, vol. 141, no. 2, pp. 021705-021705–12, Dec. 2018.

[4] D. Wu, E. Coatanea, and G. G. Wang, "Employing Knowledge on Causal Relationship to Assist Multidisciplinary Design Optimization," *Journal of Mechanical Design*, vol. 141, no. 4, p. 041402, 2019.

[5] E. Coatanéa, R. Roca, H. Mokhtarian, F. Mokammel, and K. Ikkala, "A Conceptual Modeling and Simulation Framework for System Design," *Computing in Science Engineering*, vol. 18, no. 4, pp. 42–52, Jul. 2016.

[6] E. Coatanéa, S. Nonsiri, T. Ritola, I. Y. Tumer, and D. C. Jensen, "A Framework for Building Dimensionless Behavioral Models to Aid in Function-Based Failure Propagation Analysis," *Journal of Mechanical Design*, vol. 133, no. 12, p. 121001, 2011.

[7] M. Maurer and U. Lindemann, "The application of the Multiple-Domain Matrix: Considering multiple domains and dependency types in complex product design," presented at the 2008 IEEE International Conference on Systems, Man and Cybernetics, 2008, pp. 2487–2493.

[8] S. Shan and G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Structural and Multidisciplinary Optimization*, vol. 41, no. 2, pp. 219–241, Mar. 2010.

[9] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, Jan. 1978.

[10] S. P. Borgatti and M. G. Everett, "A Graph-theoretic perspective on centrality," *Social Networks*, vol. 28, no. 4, pp. 466–484, Oct. 2006.

[11] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, Apr. 1998.

[12] F. Mokammel, E. Coatanéa, M. Bakhouya, F. Christophe, and S. Nonsiri, "Impact analysis of graph-based requirements models using PageRank algorithm," presented at the 2013 IEEE International Systems Conference (SysCon), 2013, pp. 731–736.

[13] K. Henni, N. Mezghani, and C. Gouin-Vallerand, "Unsupervised graph-based feature selection via subspace and page rank centrality," *Expert Systems with Applications*, vol. 114, pp. 46–53, 2018.

[14] M. Sorbi *et al.*, "The EuroCirCol 16T Cosine–Theta Dipole Option for the FCC," *IEEE Transactions on Applied Superconductivity*, vol. 27, no. 4, pp. 1–5, Jun. 2017.

[15] S. Caspi *et al.*, "The use of pressurized bladders for stress control of superconducting magnets," *IEEE Transactions on Applied Superconductivity*, vol. 11, no. 1, pp. 2272–2275, Mar. 2001.

[16] M. E. J. Newman, "A measure of betweenness centrality based on random walks," *Social Networks*, vol. 27, no. 1, pp. 39–54, Jan. 2005.

PUBLICATION
II

Digital Twin: Multi-dimensional Model Reduction Method for Performance Optimization of the Virtual Entity

Ananda Chakraborti, Arttu Heininen, Kari T. Koskinen, Ville Lämsä

53rd CIRP Conference on Manufacturing Systems.
https://doi.org/10.1016/j.procir.2020.04.050

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 93 (2020) 240–245

www.elsevier.com/locate/procedia

53rd CIRP Conference on Manufacturing Systems

Digital Twin: Multi-dimensional Model Reduction Method for Performance Optimization of the Virtual Entity

Ananda Chakraborti[a*], Arttu Heininen[a], Kari T. Koskinen[a], Ville Lämsä[b]

[a]*Automation Technology and Mechanical Engineering, Tampere University, Tampere FI-33720, Finland*
[b]*VTT Technical Research Centre of Finland, Tietotie 3, Espoo FI-02044 VTT, Finland*

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

Abstract

Digital Twin (DT) is an emerging technology that allows manufacturers to simulate and predict states of complex machine systems during operation. This requires that the physical machine state is integrated in a virtual entity, instantaneously. However, if the virtual entity uses computationally demanding models like physics-based finite element models or data driven prediction models, the virtual entity may become asynchronous with its physical entity. This creates an increasing lag between the twins, reducing the effectiveness of the virtual entity. Therefore, in this article, a model reduction method is described for a graph-based representation of multi-dimensional DT model based on spectral clustering and graph centrality metric. This method identifies and optimizes high-importance variables from computationally demanding models to minimize the total number of variables required for improving the performance of the DT.

Keywords: : Digital Twins, Multi-Physics Simulation, Model Fusion, Model Reduction, Spectral Clustering

1. Introduction

Currently Digital Twin (DT) is a leading research topic in manufacturing. The definition of a DT is flexible as this is a highly multidisciplinary approach and depends strongly on the application area. However, the common aspect of all DTs is the cyber-physical fusion. This cyber-physical fusion needs to be defined in the context of the application to produce functional and meaningful living models, which are precise copies of machines or systems based on advanced simulation and industrial internet of things (IIoT) data. In the context of smart manufacturing, DT is often defined as a virtual machine tool system, which aims to reflect the status of its physical object through the integration of manufacturing information. The integration of the physical and virtual entities is the key to a DT. However, in present times, it is not well understood, how to realize this physical and virtual fusion [1].

A large quantity of recent literature exists on DTs describing methods of building the twin and its fundamental components. In [2,3], the authors present a reference model of an industrial DT. The virtual entity (VE) representation of the DT is presented as a collection of various models such as geometric models (G_v), physics-based models (P_v), behavior models (B_v) and prediction or rule-based models (R_v). This is mathematically represented as; VE = $\{G_v, P_v, B_v, R_v\}$. These models do not operate in isolation in the virtual entity of the DT representation. Rather, these models exchange data and information amongst each other to enrich the prediction outcome and simulation results of a physical equipment or a manufacturing process. Such a concept for DT is used to predict the remaining useful life (RUL) of machine components in [4]. In this article, a method of fusing these virtual entity models is presented with graph-based system representation developed in [5]. This method transforms the system variables in to a causal network that models the cause-effect relationship between

10.1016/j.procir.2020.04.050

variable with the help of a conceptual modeling framework known as dimensional analysis conceptual modeling [6]. The fusion of models for virtual entity representation of a DT overcomes the problem of information silos created by various simulation models using different software packages.

A graph-based representation is chosen to construct the fusion model of the virtual entity because graphs have emerged as powerful data analytics and representation tool in the recent years. This is because graphs are simple to understand, and several types of analyses become possible for data represented in graphical format. The fusion graph of the virtual entity contains the system variables on its vertices and the relationship between the variables on the edges. This graph is a weighted or unweighted directed graph in a tuple G = (V, E, A) format, where V = {1, 2, ..., n} are the vertices set of the graph, E (i, j) are the edge set of the directed graph where $\{i,j\} \epsilon E$ such that i is the tail of the edge and j is the head. $A = [a_{ij}]$ is the adjacency matrix with non-negative values for weighted graph and ‘one’ for unweighted graph. This graph G serves as an input to several machine learning algorithms for clustering or classification of data. However, the size of such graph-based representation of the VE models will be high dimensional. That means knowledge of many system variables and their relationship is needed, in order to obtain accurate representation and predictions by the DT. This high-dimensional representation makes the fusion model of the virtual entity asynchronous with the physical entity and makes virtual entity computationally expensive. Therefore, a graph-based model reduction (dimensionality reduction) method is proposed to obtain a reduced set of system variables, which are responsible for maximum information flow in the system and are needed to obtain fast and accurate simulation of the virtual entity.

This article is organized as follows; section 2 introduces the reference model of the VE with the help of a surface grinding case study. After that, a model fusion methodology is proposed for the case study in section 3 with graph-based system representation. Section 4 proposes a model reduction (dimensionality reduction) method for the high dimensional fusion model and section 5 concludes the article.

Nomenclature

acc	accuracy
a_e	Actual depth of cut (mm)
AE	Acoustic emission intensity (V)
a_{gmax}	maximum un-deformed chip thickness (mm)
a_p	programmed or set depth of cut (mm)
A_r	Real area of contact (mm^2)
b_w	Width of grind (mm)
c	workpiece heat capacity (J/K)
C	Active grit density (mm^{-2})
d_e	equivalent wheel diameter (mm)
d_g	mean grain size (mm)
d_s	diameter of grinding wheel (mm)
d_w	diameter of work piece (mm)
E*	Combined elastic Modulus (MPa)
E_1	Elastic Modulus of Wheel (MPa)
E_2	Elastic Modulus of workpiece (MPa)
e_c	Specific grinding energy (J/mm^3)
f_g	force per grit (N)
F_n	Normal G. Force (N)
F'_n	Normal G. Force per unit width (N/mm)
F_t	Tangential G. Force (N)
F_x	grinding force in x-direction (N)
F_y	grinding force in y-direction (N)
F_z	grinding force in z-direction (N)
G_{Ratio}	Volume of material ground per unit wheel width by volume of wheel worn per unit wheel width (mm^3)
H	Hardness of grinding grain (Rockwell hardness)
h_{cu}	Uncut chip thickness (mm)
K	Archard's constant
k	workpiece thermal conductivity (WK^{-1})
k_g	Grain thermal conductivity (WK^{-1})
L	Sliding distance (mm)
l_c	real contact length (mm)
l_f	deflection contact length (mm)
l_g	geometrical contact length (mm)
L_s	variation of the contact length (mm)
L_w	Grinding distance (mm)
n_b	number of parts
N_s	Rotational wheel speed (m/min)
P	Grinding power (KW)
P_{Ratio}	Volume of metal ground per unit area of wheel surface (mm^3)
Q'	Specific removal rate
r	Grit cutting point shape factor
r_0	grain contact radius (mm)
R_r	Roughness factor
R_t	Maximum Surface Roughness (μm)
R_{ws}	Workpiece partition ratio
t	total grinding contact time (min)
t_b	total time (min)
t_c	Cycle time (min)
t_g	grinding time (min)
T_{max}	Maximum temperature (°C)
T_{mp}	Temperature approaching the melting point (°C)
V	Volume of wear of the wheel (mm^3)
v_s	Wheel speed (m/min)
V_s	Volume of wheel worn per unit wheel width (mm^3)
v_w	Workpiece speed (m/min)
V_w	volume of material ground per unit wheel width (mm^3)
V_x	velocity in x-direction (mm/min)
V_y	velocity in y-direction (mm/min)
V_z	velocity in z-direction (mm/min)
w_r	Wear life cycle
Λ	Stock removal parameter
μ	Coefficient of Grinding
ν	Poisson's Ratio
ρ	density (kg/m^3)
φ	abrasive angle (°)

2. Reference Model

A reference model for DT is proposed by the authors in [7]. This reference model describes a DT as a fusion of five components working synchronously with each other. These five components are; (1) the physical equipment (PE)

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

consisting of the actual physical device at the shop floor, (2) the virtual entity consisting of the entire virtual representation of the PE, (3) DT data, which is operational data collected from the PE, (4) connection and networking (CN), and (5) Services provided by the DT at the shop floor (Ss). The authors describe these five components as five dimensions of the DT. Each of the five dimensions can be individually modelled for every complex equipment on the shop floor.

In this article, the reference model is implemented on a cylindrical surface grinding case study. In grinding operation, an important concern is the life of the grinding wheel. Efforts to develop a low cost digital twin for grinding operation with RFID tags and single board computers is demonstrated in [8].

The PE consists of various subsystems of the grinding machine, several sensors and IIoT devices are integrated with the machine. Acoustic emission sensors, accelerometers, temperature sensors and sensors for current and voltage measurement are used to capture data from the PE. The VE is the entire virtual representation of the grinding system unifying several models such as the G_v, P_v, B_v and R_v.

G_v is the 3D model of the grinding wheel geometry. Information regarding the geometry of the grinding wheel can be obtained from these models. The P_v describes the physical phenomena associated with the operations of the PE. Finite element method (FEM) are used to obtain the physics-based modeling and simulation of the grinding wheel parameters during operation. The P_v is a complex representation of all the physical parameters effecting the grinding process and its effect on the change of grinding wheel geometry due to wheel wear. B_v describes the behavior of the grinding machine or the grinding process under certain conditions. For example, what happens when the maximum temperature of grinding exceeds predefined limits or the power consumption increase? B_v is a comprehensive model, which provides an exhaustive explanation on the events experienced by the DT. R_v is the highest-level model, which facilitates prediction and decision-making by the virtual entity with the help of machine learning algorithms. For example, tool wear prediction in manufacturing by application of deep learning methods like Recurrent Neural Networks (RNN) [9] or ensemble classifiers like Random Forest [10].

The other components i.e. DT data, CN and Ss are important auxiliary components of the DT, which are dependent on PE and VE to give context to the definition of a DT. This article focuses categorically on the high dimensional fusion model of the VE to develop a methodology for model reduction that enhances the performance of the twinning process.

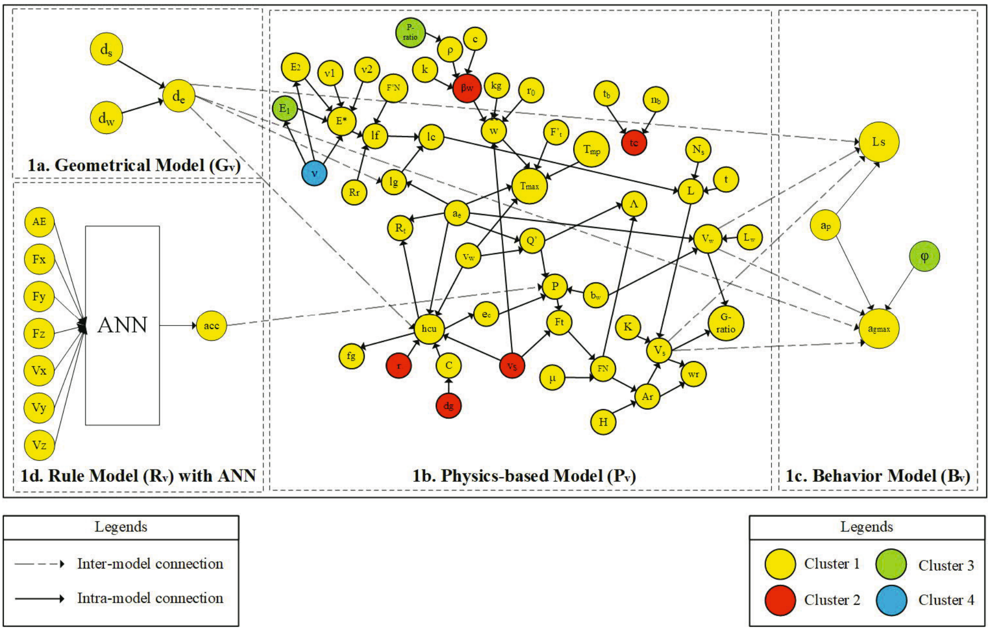


Fig. 1. The fusion model G_F of the VE; (a) Geometry model G_v; (b) Physics-based model P_v; (c) Behavior model B_v; (d) Rule model R_v.

3. Model Fusion

Fusion of the above mentioned multi-dimensional models are an important step in construction and operation of the VE. These models embed large amount of information such as product geometry, process physics, functional behavior or failure rules in different formats. Hence, realization of the fusion process is a challenging but essential task. Different software are used to build different models, which might not be compatible with each other. For example, the geometry models are constructed by 3D computer aided design (CAD) software whereas the physics-based model are built with finite element modeling (FEM) software and the rule-based model are constructed with machine learning algorithms. For this reason, a graph-based approach is used for the fusion model shown in Figure 1.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

3.1. The Fusion Model based on Reference Model

Firstly, directed graphs are generated to construct individual models with the help of dimensional analysis conceptual modeling framework. Thereafter, these models are interlinked to form the unified model of the VE, and it is denoted as a multi-dimensional graph, G_F. Figure 1 shows the individual models as well as the inter model connections to obtain the larger fusion model for building a DT for grinding system.

- G_v is the simplest graphical model. It contains geometric information such as dimensions. G_v contains information like diameter of the grinding wheel (d_s), bore to hole ratio, width of the bore and work piece diameter (d_w). Another parameter that can be obtained from the geometry models of the grinding wheel and the work piece is equivalent diameter (d_{eq}). This measure allows the comparison between two grinding applications. From the relationship of the equivalent grinding diameter, the graphical model G_v is constructed. G_v is connected to P_v through the geometry information it contains regarding the equivalent diameter.
- P_v is the physics-based model generated based on dimensional analysis conceptual modeling framework along with well understood grinding physics available in literature. P_v is the graphical representation of the multi-physical phenomena in the grinding process and it is solved with finite element method to obtain wheel wear and volume of material ground.
- B_v represents the dynamic behavior model of the grinding process. B_v determines the variation of dynamic response such as the grinding force, vibration, and grinding power. This dynamic behavior model effects grinding quality parameters such as roughness or residual stresses. In [9], the dynamic behavior of grinding process is defined mathematically as a function of chip thickness and contact length between wheel and workpiece. According to the authors, in high speed grinding, variation in vibration or work piece runout are dynamic behavior patterns that influences the grinding quality. B_v can be built as a function of maximum uncut chip thickness and contact length between the wheel and workpiece. B_v is connected to P_v through common parameters governing process physics and dynamic behavior such as V_s, V_w and d_{eq}.
- R_v is a data-driven model which defines the deduction rule between the grinding wheel wear and the input parameters. Data collection is done from the grinding process with the help of acoustic emission sensors and accelerometers. The input parameters are rotational speed, tangential force and vibration along X, Y and Z. An artificial neural network (ANN) is implemented similar to [10] to associate specific levels of the sensor signals with the grinding wheel wear. The feed forward propagation of the ANN is a logistical regression method, but the back propagation is the actual learning event when the weights of the ANN are learnt to predict grinding wheel wear based on sensor data. In order to connect R_v to the rest of the models, accuracy of the ANN is chosen. When the accuracy of the ANN is sufficiently high, R_v can be connected to P_v through grinding power [11].

4. Model Reduction

The fusion model G_F of the VE contains of large number of variables whose states and values must be known to obtain correct prediction and simulation results from the VE. In this section, a novel method of model reduction (or dimensionality reduction) is introduced to obtain a reduced model of the graph G_F so that knowledge of a smaller number of variables is required to compute and optimize the target variables. This improves the performance of the simulation process of the VE making it fast and computationally less expensive. This method uses spectral clustering techniques to group similar variables of graph G_F together. Spectral clustering is an unsupervised machine learning technique where a graph is partitioned into clusters based on the topology of the graph. Spectral clustering uses graph Laplacian to construct the clusters. The following sections describe the subcomponents of the model reduction method.

4.1. Spectral Clustering

A spectral clustering algorithm is implemented to obtain the graph partition of G_F. G_F consists of 62 variables in total. The first step in the algorithm is to define a similarity matrix. The similarity matrix is chosen as the adjacency of G_F. From this adjacency matrix, the graph Laplacian is computed, and the eigenvalues and eigenvectors are computed from the Laplacian. As G_F is a directed graph, the Laplacian is normalized to obtain the graph cuts. The number of clusters are decided based on the eigenvectors. Presence of sharp peaks in the eigenvector graph indicate the presence of a partition. The cluster is reconstructed with the graph along with *k*-means clustering where *k* is decided by the eigenvector values. For G_F, *k*=4 was obtained. Figure 1 shows the clusters from spectral clustering algorithm for G_F. The clusters are colour coded. The cluster marked in yellow is the first and biggest cluster with 53 variables in it, which cannot be subdivided further when the value of *k* is 4. The variables marked in red belong to the second cluster that contains many variables whose values depend upon the variables in the first cluster. The green and blue clusters contain other less important variables that do not contribute significantly to the prediction and simulation results of the VE. The spectral clustering already sets a hierarchy of variables. To find the relative importance of these clusters to each other, page rank algorithm is used to find the cluster hierarchy. The cluster importance is defined in the next section.

4.2. Cluster Importance

Page rank algorithm is used to find the relative importance of the clusters. This algorithm has shown stability in node ranking and has been popularly used for variable screening in physical networks such as chemical networks, protein networks and power grids. Page rank algorithm is a modified form of eigenvector centrality measure of vertices in a graph. Page rank computes the principal eigenvector of the matrix describing the edges in the graph using the power method. It is a probabilistic ranking technique to obtain which variables in the graph could be more important based on the degree distribution and fitness function. The nodes with higher Page rank are more central, meaning they tend to have a higher influence on the objective function, or they are connected to the variables which have higher influence on the objective. The principle of Page rank algorithm holds for directed unipartite networks, complex or small scale, as it generates a hierarchy of nodes in the network based on the probabilities of the nodes at a given state of the system obtained from the degree distribution. Page rank is computed for G_F. The Page rank scores with power iteration method are shown in Figure 2. It is found that higher page rank scores are obtained for variables lying in the first cluster. Whereas the page rank scores of variables lying in other clusters are low and fail to qualify as important variables when a threshold of 0.025 is selected. This threshold of 0.025 is selected based on the percentile Page rank score of all the variables and applying classical Pareto 80/20 principle on the percentile scores.

It can be observed from the spectral clustering pattern that cluster 1 contains V_s and V_w, which are the target variables of the physics-based model P_v. At this point, it is desirable to reformulate the VE model based only on the variables belonging to cluster 1. Variables in other clusters can be neglected as they contain dependent variables and their knowledge itself does not alter the computing or simulation time of the VE by a large amount. Clusters 2, 3 and 4 contain variables, which have a low page rank score, and it could be inferred that they do not contribute significantly to the target variables V_s and V_w. Cluster 1 is therefore the most important cluster containing all the important variables that should be optimized to obtain higher performance of the twin. In other words, Cluster 1 is the reduced model of G_F that contains the knowledge of all the variables, which are critical to represent the VE with a reasonable accuracy.

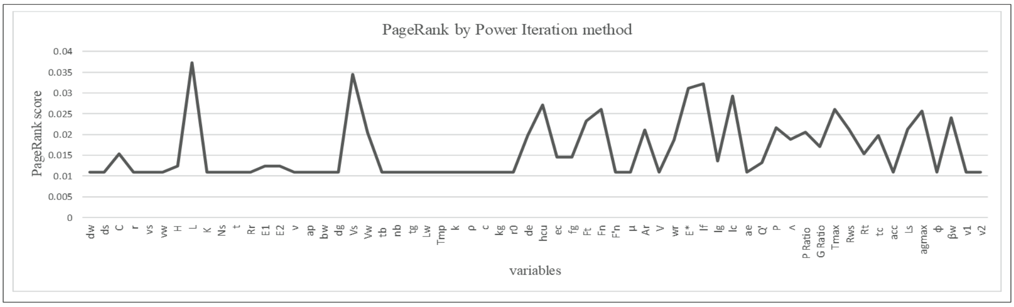


Fig. 2. Page Rank score of G_F by power iteration method

To validate this approach, performance variables V_s and V_w are optimized based only on the variables from cluster 1 and the values are compared with other grinding case studies from the literature. There are conflicting objectives in the VE as the variable V_s should be minimized and V_w should be maximized in the grinding operation. Hence, in the next section, a multi-objective optimization problem is formulated and solved to demonstrate the significance of cluster 1 variables on the performance of the VE.

4.3. Multi objective optimization

Spectral clustering and graph centrality metric define variables in cluster 1 as the variables, which have the highest impact on the target variables i.e. V_s and V_w. R_t is also considered in the optimization problem as maximum roughness should be minimized. But, R_t contains v_s in its equation, which is a variable from cluster 2. While formulating the R_t optimization equation, v_s is replaced by 1 as it is in the denominator. This replacement is done for all the variables belonging to cluster 2, 3 and 4. Due to the presence of conflicting objectives in the VE, a multi-objective optimization problem is formulated based on the mathematical equations for grinding defined in [12,13]. The optimization problem contains three objectives however, several other objectives can be selected depending on the application of the VE. The optimization formulation is as follows:

f(cluster 1)

objective 1

min : $f(V_s)$

where, $V_s = b_w\, a_s\, \pi\, d_s$

objective 2

max $f(V_w)$

where, $V_w = b_w\, a_e\, L_w$

objective 3

min $f(R_t)$

where, $R_t \approx \left(\dfrac{v_w}{v_s} \cdot \dfrac{1}{C\,r\sqrt{d_e}} \right)^{\frac{2}{3}}$ and $d_e = \dfrac{d_s d_w}{d_s \pm d_w}$

with model reduction ($v_s \approx 1$); $R_t \approx \left(\dfrac{v_w}{C\,r\sqrt{d_e}} \right)^{\frac{2}{3}}$

subjected to;

$200 \le d_s \le 355\,\text{mm}$

$150 \le L_w \le 300\,\text{mm}$

$0.0505 \le a_e \le 0.1\,\text{mm}$

In objective 1, V_s is defined as a function of a_s, which is the wear depth of the grinding wheel. As no standard value for a_s

was found in literature, it is assumed to be the lowest value of the variable a_e. In objective 3, the value of C is chosen as 32 μ/mm^2 and grit size r is chosen as 500 based on ANSI B74.16, 1995. Also, for surface grinding, $d_e \approx d_s$. The optimization problem is solved with the help of a MATLAB based genetic algorithm solver known as 'gamultiobj' and data collected from the digital environment for grinding with IIoT devices. The values of the variables, whose data are not collected from the digital grinding environment, are derived from previous work on grinding parameter optimization such as the case study presented in [14]. Acoustic emission data is collected from the grinding machine with acoustic emission sensors. The acoustic emission intensity is related to the grinding power P to compute volume of wheel wear V_w, if a significant accuracy is obtained from the rule-based model R_v. The fitness function for multi-objective optimization is defined according to the mathematical expression of the performance variables. The optimization result is shown in Table 1, for n_b = 20.

Table 1. Optimization results.

Variables	Optimized value
V_s	24.485 mm^3
V_w	1419.4 mm^3
R_t	0.8 μm

To compare the optimization results with the case study from literature, G_{Ratio} is used. G_{Ratio} is a measure of the ability of a grinding wheel to remove materials and it is given by the ratio of V_w/V_s. G_{Ratio} obtained from Table 1 is 57.95 and G_{Ratio} obtained from the case study is 60. Hence, less than 10% error is obtained. This indicates that the variables in cluster 1 is sufficient to provide meaningful information on performance optimization of the VE. Thus, cluster 1 containing 53 variables provides an equivalent representation of the high dimensional model G_F.

5. Conclusion

In this article, a model reduction method is presented for multidimensional graphical representation of the variables in the VE models. This model reduction method uses spectral clustering to group similar variables together and analyses their importance in optimization of the performance variables that governs the overall performance of the DT. An important performance indicator of the DT is the time required by the simulation models to update itself based on industrial internet of things (IIoT) data from the shop floor. This model reduction method can identify the less sensitive variables in the VE so that data collection or complicated analytics of these variables can be eliminated. This makes the VE representation of the DT faster and more efficient.

6. Future Work

In the future, a detailed graphical representation of the grinding system will be built, and the model reduction method will be integrated with the functional DT for daily operations of the grinding machine to make the simulation models faster and more sensitive to the working environment. Collecting and processing all information needed to update simulation runs of the PE remains a practical challenge for high-fidelity simulation models, which try to mimic the reality in greater detail. Model reduction methods, such as the method described in this article, will facilitate in understanding which parameters have the most influence on the process and selectively optimize those parameters in the virtual entity to enhance the performance of the DT.

Acknowledgements

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

References

[1] Schleich B, Anwer N, Mathieu L, Wartzack S. Shaping the digital twin for design and production engineering. CIRP Annals 2017;66:141–4. https://doi.org/10.1016/j.cirp.2017.04.040.

[2] Tao F, Zhang M, Liu Y, Nee AYC. Digital twin driven prognostics and health management for complex equipment. CIRP Annals 2018;67:169–72. https://doi.org/10.1016/j.cirp.2018.04.055.

[3] Tao F, Zhang H, Liu A, Nee AYC. Digital Twin in Industry: State-of-the-Art. IEEE Transactions on Industrial Informatics 2019;15:2405–15. https://doi.org/10.1109/TII.2018.2873186.

[4] Aivaliotis P, Georgoulias K, Chryssolouris G. The use of Digital Twin for predictive maintenance in manufacturing. International Journal of Computer Integrated Manufacturing 2019;32:1067–80. https://doi.org/10.1080/0951192X.2019.1686173.

[5] 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.

[6] Coatanéa E, Roca R, Mokhtarian H, Mokammel F, Ikkala K. A Conceptual Modeling and Simulation Framework for System Design. Computing in Science Engineering 2016;18:42–52. https://doi.org/10.1109/MCSE.2016.75.

[7] Tao F, Zhang M, Nee AYC. Chapter 6 - Cyber–Physical Fusion in Digital Twin Shop-Floor. In: Tao F, Zhang M, Nee AYC, editors. Digital Twin Driven Smart Manufacturing, Academic Press; 2019, p. 125–39. https://doi.org/10.1016/B978-0-12-817630-6.00006-0.

[8] Kannan K, Arunachalam N. A Digital Twin for Grinding Wheel: An Information Sharing Platform for Sustainable Grinding Process. Journal of Manufacturing Science and Engineering 2019;141:021015. https://doi.org/10.1115/1.4042076.

[9] ma Y, yang J, li B, lu J. An analytical model of grinding force based on time-varying dynamic behavior. Int J Adv Manuf Technol 2017;89:2883–91. https://doi.org/10.1007/s00170-016-9751-5.

[10] Wu D, Jennings C, Terpenny J, Gao RX, Kumara S. A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests. Journal of Manufacturing Science and Engineering 2017;139:071018. https://doi.org/10.1115/1.4036350.

[11] Badger J, Murphy S, O'Donnell GE. Acoustic emission in dressing of grinding wheels: AE intensity, dressing energy, and quantification of dressing sharpness and increase in diamond wear-flat size. International Journal of Machine Tools and Manufacture 2018;125:11–9. https://doi.org/10.1016/j.ijmachtools.2017.11.007.

[12] Marinescu, Ioan D., et al. Handbook of Machining with Grinding Wheels. CRC Press LLC, 2016; n.d.

[13] Rowe WB. Principles of Modern Grinding Technology. Principles of Modern Grinding Technology, William Andrew Publishing; 2009, p. 79–93.

[14] Slowik A, Slowik J. Multi-objective optimization of surface grinding process with the use of evolutionary algorithm with remembered Pareto set. Int J Adv Manuf Technol 2008;37:657–69. https://doi.org/10.1007/s00170-007-1013-0.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

PUBLICATION III

Evidential Reasoning based Digital Twins for Performance Optimization of Complex Systems

Ananda Chakraborti, Arttu Heininen, Saara Väänänen, Kari T. Koskinen, Henri Vainio

54th CIRP Conference on Manufacturing Systems.
https://doi.org/10.1016/j.procir.2021.11.104

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

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

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.

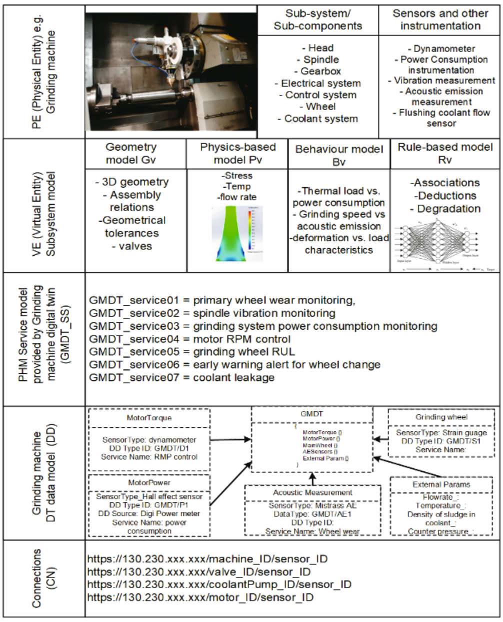

PE (Physical Entity) e.g. Grinding machine
Sub-system/ Sub-components
- Head
- Spindle
- Gearbox
- Electrical system
- Control system
- Wheel
- Coolant system
Sensors and other instrumentation
- Dynamometer
- Power Consumption instrumentation
- Vibration measurement
- Acoustic emission measurement
- Flushing coolant flow sensor
VE (Virtual Entity) Subsystem model
Geometry model Gv
- 3D geometry
- Assembly relations
-Geometrical tolerances
- valves
Physics-based model Pv
-Stress
-Temp
-flow rate
Behaviour model Bv
-Thermal load vs. power consumption
- Grinding speed vs acoustic emission
-deformation vs. load characteristics
Rule-based model Rv
-Associations
-Deductions
- Degradation
PHM Service model provided by Grinding machine digital twin (GMDT_SS)
GMDT_service01 = primary wheel wear monitoring,
GMDT_service02 = spindle vibration monitoring
GMDT_service03 = grinding system power consumption monitoring
GMDT_service04 = motor RPM control
GMDT_service05 = grinding wheel RUL
GMDT_service06 = early warning alert for wheel change
GMDT_service07 = coolant leakage
Grinding machine DT data model (DD)
MotorTorque
SensorType: dynamometer
DD Type ID: GMDT/D1
Service Name: RMP control
GMDT
{
MotorTorque {}
MotorPower {}
MainWheel {}
AESensors {}
External Param {}
}
Grinding wheel
SensorType: Strain guage
DD Type ID: GMDT/S1
Service Name:
MotorPower
SensorType_Hall effect sensor
DD Type ID: GMDT/P1
DD Source: Digi Power meter
Service Name: power consumption
Acoustic Measurement
SensorType: Mistrass AE
DataType: GMDT/AE1
DD Type ID:
Service Name: Wheel wear
External Params
Flowrate_:
Temperature_:
Density of sludge in coolant_:
Counter pressure_:
Connections (CN)
https://130.230.xxx.xxx/machine_ID/sensor_ID
https://130.230.xxx.xxx/valve_ID/sensor_ID
https://130.230.xxx.xxx/coolantPump_ID/sensor_ID
https://130.230.xxx.xxx/motor_ID/sensor_ID

Fig. 1. DT reference model for grinding wheel wear

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

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.

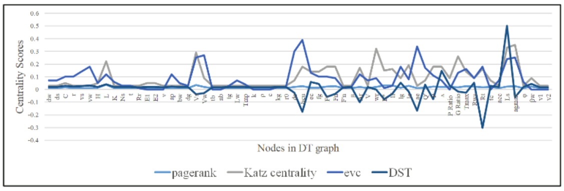


Fig. 2. Comparison of node importance by centrality methods and DST

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:

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

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

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

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

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.

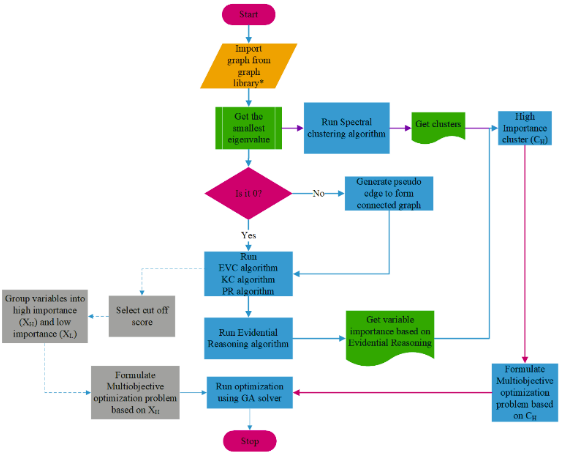

Start
Import graph from graph library*
Get the smallest eigenvalue
Run Spectral clustering algorithm
Get clusters
High Importance cluster (C_H)
Is it 0?
No
Generate pseudo edge to form connected graph
Yes
Run EVC algorithm KC algorithm PR algorithm
Select cut off score
Group variables into high importance (X_H) and low importance (X_L)
Run Evidential Reasoning algorithm
Get variable importance based on Evidential Reasoning
Formulate Multiobjective optimization problem based on X_H
Run optimization using GA solver
Formulate Multiobjective optimization problem based on C_H
Stop

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.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

PUBLICATION
IV

A Graph-Based Model Reduction Method for Digital Twins

Ananda Chakraborti, Henri Vainio, Kari T. Koskinen, Juha Lammi

Chakraborti A, Vainio H, Koskinen KT, Lammi J. A Graph-Based Model Reduction Method for Digital Twins. Machines 2023;11:733. https://doi.org/10.3390/machines11070733.

machines

Article

A Graph-Based Model Reduction Method for Digital Twins

Ananda Chakraborti [1,*], Henri Vainio [1], Kari T. Koskinen [1] and Juha Lammi [2]

[1] Automation Technology and Mechanical Engineering Department, Tampere University, 33720 Tampere, Finland; henri.vainio@tuni.fi (H.V.); kari.koskinen@tuni.fi (K.T.K.)
[2] Tamturbo Oy, 33100 Tampere, Finland
* Correspondence: anandaschakraborti@gmail.com; Tel.: +358-413692130

Abstract: Digital twin technology is the talking point of academia and industry. When defining a digital twin, new modeling paradigms and computational methods are needed. Developments in the Internet of Things and advanced simulation and modeling techniques have provided new strategies for building complex digital twins. The digital twin is a virtual entity representation of the physical entity, such as a product or a process. This virtual entity is a collection of computationally complex knowledge models that embeds all the information of the physical world. To that end, this article proposes a graph-based representation of the virtual entity. This graph-based representation provides a method to visualize the parameter and their interactions across different modeling domains. However, the virtual entity graph becomes inherently complex with multiple parameters for a complex multidimensional physical system. This research contributes to the body of knowledge with a novel graph-based model reduction method that simplifies the virtual entity analysis. The graph-based model reduction method uses graph structure preserving algorithms and Dempster–Shaffer Theory to provide the importance of the parameters in the virtual entity. The graph-based model reduction method is validated by benchmarking it against the random forest regressor method. The method is tested on a turbo compressor case study. In the future, a method such as graph-based model reduction needs to be integrated with digital twin frameworks to provide digital services by the twin efficiently.

Keywords: digital twin; graph-based knowledge representation; model fusion; model reduction; importance measurement

check for updates

Citation: Chakraborti, A.; Vainio, H.; Koskinen, K.T.; Lammi, J. A Graph-Based Model Reduction Method for Digital Twins. *Machines* **2023**, *11*, 733. https://doi.org/10.3390/machines11070733

Academic Editor: Aydin Nassehi

Received: 4 June 2023
Revised: 4 July 2023
Accepted: 9 July 2023
Published: 12 July 2023

1. Introduction

Digital twins (DTs) have been perceived in multiple ways. Several descriptions of DTs exist in the scientific literature, many of which have gone beyond the three-dimensional DT proposed by Michael Grieves [1]. Digital twins are described as virtual substitutes of real-world objects consisting of virtual representations and communication capabilities making up smart objects and acting as intelligent nodes inside the Internet of Things context. Digital twins have reached beyond the field of product lifecycle management, where it was first conceived, into manufacturing processes [2], communication and networking [3], construction [4] and smart grids [5]. However, the underlying research questions remain; how to best represent a complex multidimensional DT system and how to simplify that representation to reduce twin's computational complexity and interpretability?

The DT is a multidimensional entity. In [6], the DT is realized as a five-dimensional living model. It is a collection of simulation models, information models, and IoT data acquisition and processing. Plenty of research is available on the development of these models and data-driven methods [7]. However, the important area that has been overlooked by the digital twin research community is model reduction. In this study, graph-based methods are proposed for conceptualizing the complex DT representation. To address the computational complexity of the DT, a graph-based model reduction (GBMR) method is proposed. The GBMR method was first conceived as a dimensionality reduction method [8] but evolved into the virtual entity representation and optimization tool for the DT [9]. The GBMR method addresses the computational complexity of the DT with a two-step approach: (1) providing a graph-based conceptual model representation of the DT by utilizing a casual graph extraction method known as dimensional analysis and conceptual modeling, and (2) reducing the DT graph model by spectral decomposition and identifying the important parameters in it. The novelty of GBMR lies in representing the physical system as a graph-based model and reducing that graph by finding important parameters dynamically by the DT. The model reduction process helps to optimize the virtual entity performance of the DT as the reduced model uses a subset of important parameters to predict the target parameter of the physical entity.

To that end, the primary contributions of this research work are: (1) provide the development of the GBMR method for fast identification of the important parameters for reducing the computational complexity of the DT, and (2) test the GBMR method with the help of a turbo compressor case study and analyze the results. This paper is structured as follows: Section 2 provides the state of the art in DT development with a focus on conceptual graph-based methods. Section 3 introduces the GBMR method. In Section 4, a case study of the GBMR method is presented for a turbo compressor system including the results and future research directions of the method, and Section 5 concludes the article.

2. State of the Art

2.1. The Complexity of DT Development

In [10], the authors propose the technologies enabling DT development, which is the combination of the digital world such as simulation and modeling (S&M) and the physical world such as the IoT. The S&M approach for DTs creates the possibility of modularizing the development of DTs and focusing on the essentials. However, it also introduces large amount of complexity in building and operating such DTs. These models can arise from different domains of the physical object. They are complex, real-time, "living" entities. These models can be organizational information models, engineering models (thermal, fluid, dynamic, electrical or systems-level), rule-based models (associative, deductive or degradation) or purely data-driven models (ANN or deep-learning-based models). The Internet of Things (IoT) or Industrial Internet of Things (IIoT) facilitate building real-time models based on the data. Figure 1 demonstrates the system-level architecture that combines the IoT/IIoT and S&M to provide the foundation for such DT development. A similar approach was also adopted in [11] for a DT development.

The right side of Figure 1 focuses on the IoT part. The IoT provides the following components as building blocks of a digital twin:

Data-driven Models: Data-driven predictive models form the basis of many digital twins. Many such data-driven digital twins can be found in the literature [12,13]. These predictive models are built for state estimation, behavior prediction or causal analysis. Machine learning methods such as Bayesian networks and evidential reasoning are used for building these models [14]. The future state estimation by these methods could serve as the input to many simulation models. Environmental data and other web-based data such as metrological data are also used in building such estimation models, which form an essential part of the virtual entity.

Model Fusion and Model Reduction: The curse of dimensionality is often experienced in building data-driven models. State prediction models or behavior estimation models typically contain several parameters that should be monitored, and data should be collected with sensors and a proper connectivity mechanism. Building high-dimensional and high-fidelity models that replicate the reality with a high degree of accuracy are extremely challenging. These models are computationally extensive. It is also resource consuming to train these models with data from the physical device and develop methods for validating the results. Model fusion provides a mean to generate a hybrid model by combining physics-based and data-driven models and model reduction provides a means to reduce or combine the number of parameters in that hybrid model. In doing so, the hybrid model becomes computationally simplified and takes less time to provide prediction result.

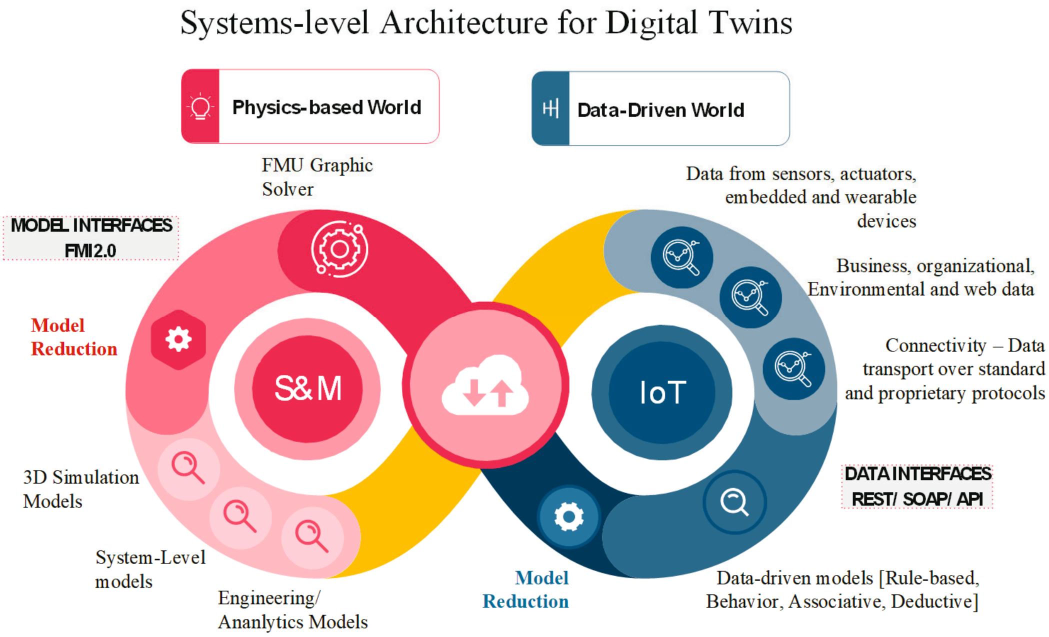

Systems-level Architecture for Digital Twins
Physics-based World
Data-Driven World
FMU Graphic Solver
MODEL INTERFACES FMI2.0
Model Reduction
S&M
3D Simulation Models
System-Level models
Engineering/ Ananlytics Models
Data from sensors, actuators, embedded and wearable devices
Business, organizational, Environmental and web data
Connectivity – Data transport over standard and proprietary protocols
IoT
DATA INTERFACES REST/ SOAP/ API
Model Reduction
Data-driven models [Rule-based, Behavior, Associative, Deductive]

Figure 1. A system-level architecture for DTs.

The left side of Figure 1 focuses on S&M. This part of the figure highlights the advanced simulation models that need to be built to capture the physics of the system. S&M creates the virtual entity of the five-dimensional representation of the digital twin [15]. The simulation models could be from one or several domains such as analytical models, geometrical models or system-level models. Analytical models such as finite element or computational fluid flow models are necessary to predict the state of the physical entity through software-defined methods such as thermal analysis or stress analysis. Geometrical models purely represent the physical phenomena of the physical entity such as deformation and buckling. The system-level model combines other types of models to provide system-level information such as efficiency and performance. By combining these advanced simulation models, it is possible to represent the complete state of the physical entity. The DT will demand that these advanced simulation models work in unison and possess the capability to provide the updated state of the system based on IoT data. This makes the DT computationally extensive and requires the development of methods, tools and techniques for understanding and reducing the DT's computational complexity.

Compressing information from these bulky analytical models and making them predict the system state based on real-time data need further advancement of technology. These simulation models are designed to provide a high-fidelity representation of the system without the consideration of a faster prediction of model output. Model reduction is needed for building compressed digital representations from these simulation models [16]. Model reduction methods are already applied in these advanced simulation environments at an individual component level. This could be traditional methods such as reduced order modeling by proper orthogonal decomposition [17] or by applying newer deep learning methods [18,19]. Metamodeling has been used to reduce the dimensionality of complex systems and is propagated as a class of model reduction as well [20]. However, there is a lack of a unified method that combines the reduced model from the component to system level.

2.2. DT Reference Model

To realize the complexity of the DT development process, a reference model is crucial. The five-dimensional representation of the DT provides such a reference model [21,22]. In this section, the five-dimensional representation from the literature is utilized to build a reference model for DTs. Figure 2 presents the reference model based on a grinding machine case study [9].

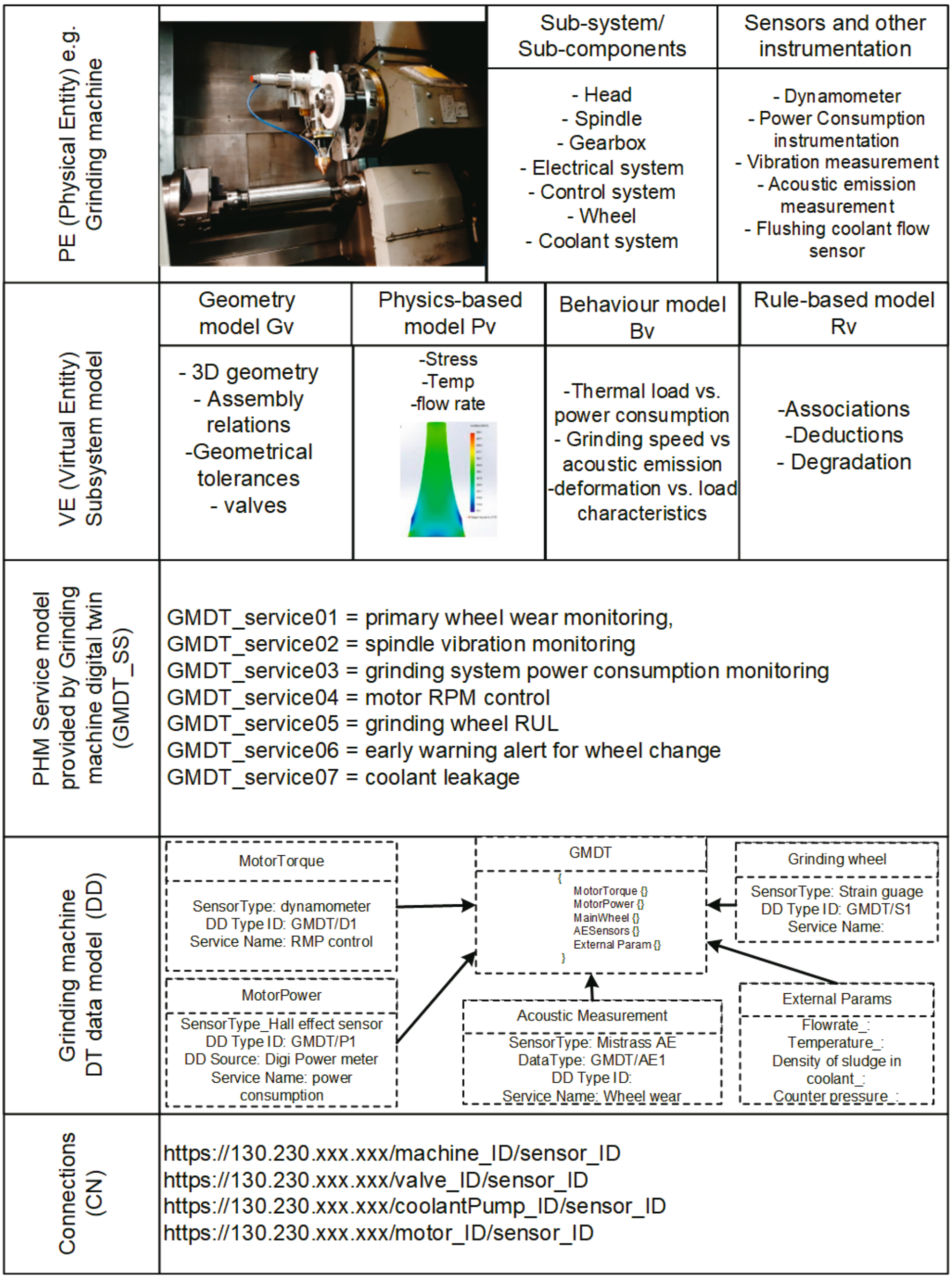

PE (Physical Entity) e.g. Grinding machine
Sub-system/ Sub-components
- Head
- Spindle
- Gearbox
- Electrical system
- Control system
- Wheel
- Coolant system
Sensors and other instrumentation
- Dynamometer
- Power Consumption instrumentation
- Vibration measurement
- Acoustic emission measurement
- Flushing coolant flow sensor
VE (Virtual Entity) Subsystem model
Geometry model Gv
- 3D geometry
- Assembly relations
-Geometrical tolerances
- valves
Physics-based model Pv
-Stress
-Temp
-flow rate
Behaviour model Bv
-Thermal load vs. power consumption
- Grinding speed vs acoustic emission
-deformation vs. load characteristics
Rule-based model Rv
-Associations
-Deductions
- Degradation
PHM Service model provided by Grinding machine digital twin (GMDT_SS)
GMDT_service01 = primary wheel wear monitoring,
GMDT_service02 = spindle vibration monitoring
GMDT_service03 = grinding system power consumption monitoring
GMDT_service04 = motor RPM control
GMDT_service05 = grinding wheel RUL
GMDT_service06 = early warning alert for wheel change
GMDT_service07 = coolant leakage
Grinding machine DT data model (DD)
MotorTorque
SensorType: dynamometer
DD Type ID: GMDT/D1
Service Name: RMP control
MotorPower
SensorType_Hall effect sensor
DD Type ID: GMDT/P1
DD Source: Digi Power meter
Service Name: power consumption
GMDT
{
MotorTorque {}
MotorPower {}
MainWheel {}
AESensors {}
External Param {}
}
Acoustic Measurement
SensorType: Mistrass AE
DataType: GMDT/AE1
DD Type ID:
Service Name: Wheel wear
Grinding wheel
SensorType: Strain guage
DD Type ID: GMDT/S1
Service Name:
External Params
Flowrate_:
Temperature_:
Density of sludge in coolant_:
Counter pressure_:
Connections (CN)
https://130.230.xxx.xxx/machine_ID/sensor_ID
https://130.230.xxx.xxx/valve_ID/sensor_ID
https://130.230.xxx.xxx/coolantPump_ID/sensor_ID
https://130.230.xxx.xxx/motor_ID/sensor_ID

Figure 2. DT reference model.

The DT reference model consists of five dimensions:

(1) Physical Entity (PE): This consists of the sub-systems and sensory devices. This could range from sensors, actuators and control systems to the whole sub-system such as the motor drives, spindles or transmission of the machine. PE guides the process of DT development by providing IoT data from these sub-systems. So, the PE also provides communication interfaces, RFID tags or distributed sensor networks.

(2) Virtual Entity (VE): This is the complex virtual representation of the PE. The VE consists of geometric models, analytical- or physics-based models, behavioral models and rule-based models [23,24]. The VE may contain detailed geometric models such as 3D CAD models or physics-based models such as finite element models. It may contain various behavior modeling methods such as Markov-chains or ontology-based models. Historical data from the PE are used to create rule-based models. The rule-based models provide the VE with the capacity for judgement, optimization and prediction.

(3) Service: This provides the reason for building a DT that is the digital services. In Figure 2, it could be services such as grinding wheel wear monitoring or early warning for a wheel change based on the remaining useful life of the wheel. These services fall under the category of prognostics and health management (PHM) services for the grinding machine.

(4) Data model: The data model creates the schema for data exchange between the PE and the VE. In Figure 2, an example is provided. The grinding machine digital twin requires sensor data to exchange between PE and VE, such as motor torque, motor power, acoustic emission and wheel wear.

(5) Connections: The connections bind the PE to the VE with the help of the data dimension. PE to VE binding defines acquiring data from the sensors on the grinding wheel with API endpoints. Similarly, VE to PE binding provides the output of analytical results to the physical device to perform an action such as grinding wheel speed control.

The VE embeds information from multi-domain models. A model fusion approach is taken to build the VE graph to model the interaction of the parameters with the help of methods such as DACM (Section 2.3.1) and heuristic search (Section 2.3.2). The VE graph is denoted as $VE_g = \{G_v, P_v, B_v, R_v\}$, where G_v, P_v, B_v and R_v are the graph representation of parameters in geometric, analytical, behavioral and rule-based domains, respectively. The fused model or VE_g then represents the complete knowledge model of the DT. The VE_g is reduced with graph-based methods such as node importance (Section 2.4) and evidential reasoning (Section 2.5).

2.3. Graph-Based Modeling of Complex Systems

2.3.1. Dimensional Analysis and Conceptual Modeling

Dimensional analysis and conceptual modeling (DACM) is a conceptual modeling mechanism used to extract the causal relationship between variables in a physics-based simulation environment [25]. This method uses the dimensional homogeneity principle to extract the causal relationship between the parameters. DACM is a matured framework and is already applied to use cases in the field of additive manufacturing [26] and multi-disciplinary design optimization [27]. The DACM framework starts with functional modeling of the system and the assigning of fundamental variables to the different functions of the model. The functions, associated variables and representative equations are characterized in the causal graph in the form of the cause–effect relationship between the fundamental variables of the functional model. The mathematical machinery to check the propagation of an objective in a causal graph is based on the Vashy–Buckingham pi (π) theorem and the dimensional analysis (DA) theory. DACM encodes the domain knowledge of the system in the form of the directed causal graph. Specific checks are run to identify and remove any loops or contradictions in the graph. This ensures a target-driven directed model. This source of the domain knowledge could be from the literature, empirical relationship or analytical models. DACM is combined with machine learning methods such as the Bayesian network for causal inference. The objective of the causal graph provided by

DACM is to arrive at the target variable in the directed acyclical graph or DAG with the help of a set of intermediate dependent and independent variables. Apart from extracting the causal graph, DACM also provides the following checks: (1) it generates sets of behavioral equations associated with the causal graphs, (2) it simulates qualitatively behaviors, (3) it detects contradictions in systems, and (4) it provides a set of analytical concepts for analyzing complex systems.

2.3.2. Greedy Equivalence Search

Chickering, in [28], provided a method for graph structure learning with a two-phase greedy equivalence search (GES) algorithm from data. Graph structure learning is a sequential process that learns the relations between the random variables (nodes of a graph) that are embedded in the edges simulating a causal influence. The GES algorithm provides a mechanism to obtain such a distribution and represent it in the form of a DAG. The GES approach has an important influence in machine learning methods such as the Bayesian network for graph structure learning. Another experimental GES was proposed by [29], it was called the greedy interventional equivalence search (GIES) and generalizes the GES algorithm. Interventions distort the value of random variables to throw the graph out of its original causal dependencies and make it find the original DAG. In this article, the GES algorithm is used to discover the accurate causal reasoning of the DT graph.

It was proved that, for two DAGs δ and λ, where δ is an I-map of λ, there are a finite sequence of edge additions and reversals in λ, such that: (1) after each edge modification, δ remains I-Map of λ, and (2) after all modification, λ is a perfect map of δ. The two-phase algorithm starts with a graph assuming that there are no dependencies. This is indicated as the zero-edge model. Then, all possible single edges are added till the algorithm reaches a local maximum. The phase of progressively adding single edges in the DAG is known as the forward equivalence search (FES); the corresponding local maxima is known as the FES local maxima. Once the FES algorithm stops at a local maximum, a second-phase greedy algorithm is applied that considers at each step all possible single-edge deletions that can be made to the DAG. This phase is known as the backward equivalence search (BES). The algorithm terminates when the BES local maxima is identified. The concept is demonstrated in Figure 3.

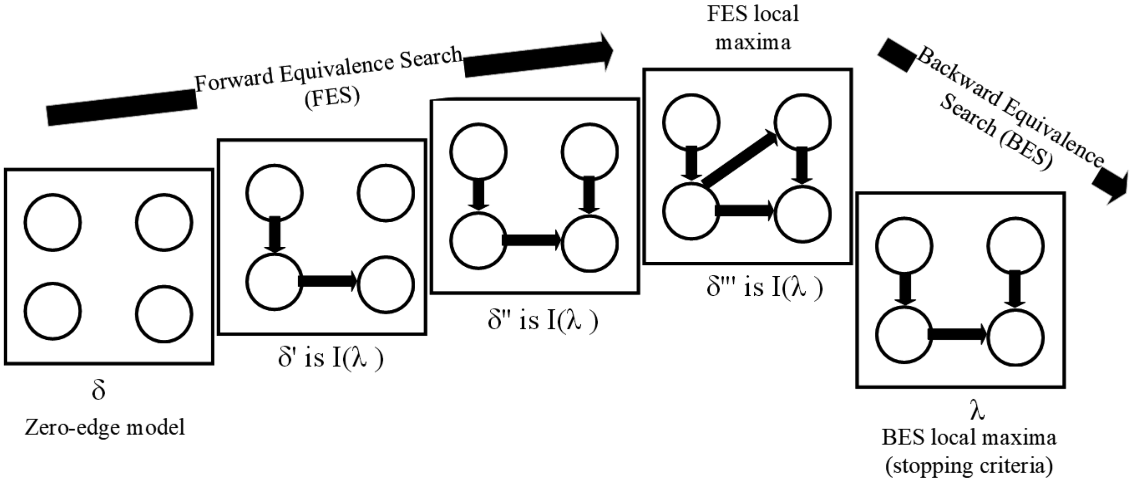


Figure 3. Greedy Equivalence Search Algorithm.

2.4. Node Importance Measurement

Identifying the node importance in a complex graph is an active field of research in artificial intelligence. Several studies and algorithms have been published to estimate the importance of nodes in a graph [30,31]. 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 given node to the target node, predicting the links between the nodes, understanding the relative importance of the nodes in a graph and finding the bridge nodes to detect communities or clusters. Experimental studies have proven the validity of such systems applied to complex networks [32]. The PageRank algorithm is a popular algorithm in graph centrality measurement in directed graphs. It is a network ranking method developed to compute the ranks of webpages in Google's search engine results. The PageRank algorithm iteratively converges to a point for the most influential nodes. Hence, it creates a hierarchical node importance ranking system in a DAG. An improved version of this algorithm is used in applications that go beyond search engine ranking, which include impact analysis of graph-based system requirements and graph-based feature selection [33]. The PageRank $P(i)$ of a node i can be calculated as follows:

$$P(i)^n = \sum_{j=1}^{q} a_{ij} \frac{P(j)^{n-1}}{k_j} \tag{1}$$

The influence of node i in n steps is denoted as $P(i)^n$. The higher the value of $P(i)$, the higher are the chances that it is an important node. The $P(j)^{n-1}$ indicates that the node also depends on the importance of the $n-1^{th}$ node. So, if a high importance node is pointing towards the node, it is considered important. Weighted PageRank (WPR) is a modified form of the PageRank algorithm that is used to rank real system parameters. In WPR, the influence of other nodes can be controlled by selecting appropriate weights [34].

Eigenvector centrality [35] is another graph centrality measuring algorithm that is used by social scientists to measure prestige in large connected graphs. EVC identifies nodes by the number of their neighbors and their importance. EVC is calculated with the following formula:

$$EVC = \frac{1}{\lambda} \sum_{j=1}^{n} (a_{ij} x_j) \tag{2}$$

where the largest EVC value is represented by λ.

2.5. Evidential Reasoning

The graph centrality and node importance approaches, though mathematically accurate, are often general and yield contradicting results. This results in disparity in the ranking system, introducing uncertainty and incompleteness in the ranking system. The Dempster–Shafer theory (DST) deals with this uncertainty and the incomplete behavior of any ranking system. Having its roots in probability theory, the DST uses data fusion and combinatorial rules to provide a belief function to a set of elements in the domain. The DST is used to combine the information available regarding the nodes in the DT graph and their relative importance obtained from the node importance scores. 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 [36] for directed networks. The frame of discernment contains all the possible combinations where the combination lies. If there are three hypotheses possible ($\varnothing = \{\theta_1, \theta_2, \theta_3\}$), the set of all combinations where the solution lies are:

$$2^{\varnothing} = \{\phi\{\theta_1\}, \{\theta_2\}, \{\theta_3\}, \{\theta_1\theta_2\}, \{\theta_2\theta_3\}, \{\theta_1\theta_3\}, \{\theta_1\theta_2\theta_3\}\} \tag{3}$$

The maximum and minimum values of the corresponding ranking is used to compute the mass functions with the following formulae:

$$m_{C(i)}(h) = \frac{C_i - C_m}{C_M - C_m + \omega} \tag{4}$$

$$m_{C(i)}(l) = \frac{C_i - C_M}{C_M - C_m + \omega} \tag{5}$$

$$m_{C(i)}(\varnothing) = 1 - m_{C(i)}(h) - m_{C(i)}(l) \tag{6}$$

where ω is a tunable parameter that is chosen to avoid the denominator becoming zero. Repeating the steps in Equations (1)–(3) creates a basic probability assignment (BPA) for each node in the form:

$$M_{C(i)} = \left\{ m_{C(i)}(h), m_{C(i)}(l), m_{C(i)}(\varnothing) \right\} \tag{7}$$

There will be the same number of BPAs created as there are nodes in the sets. Now, all the node importance scores obtained from different centrality metrics can be combined with the help of Dempster's combination rule to generate a new combined ranking for the nodes. Dempster's combination rule, used in the field of IoT sensor fusion [37], is modified to obtain the new metric for nodes based on the evidence of whether the node is high importance or low importance:

$$m_i(h) = \frac{1}{1-k} \sum_{C(i)=h}^{n} m_{C(i)}(h) \tag{8}$$

$$m_i(l) = \frac{1}{1-k} \sum_{C(i)=l}^{n} m_{C(i)}(l) \tag{9}$$

where

$$k = \sum_{C(i)=\varnothing}^{n} m_{C(i)}(\varnothing) \tag{10}$$

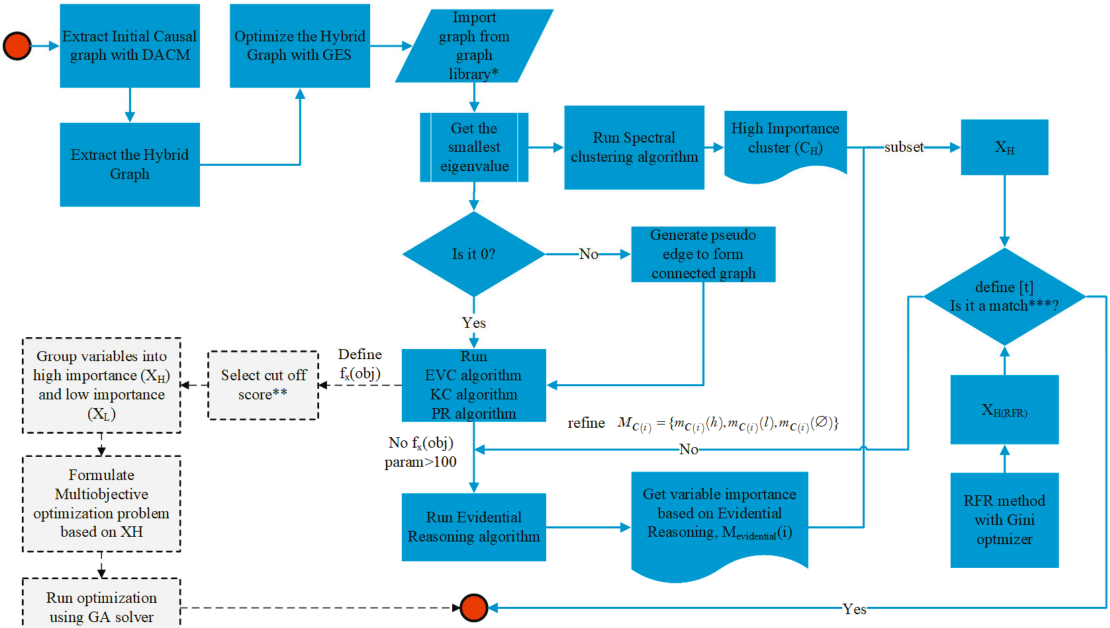
The factor k is a normalization constant known as the conflict coefficient of two BPAs. The higher the value of k, the more conflicting the sources of evidence are and the less information they will combine. Finally, the combined scores of each node based on evidential reasoning is obtained as:

$$M_{evidential}(i) = m_i(h) - m_i(l) \tag{11}$$

In Section 2, the complexity involved in building the DT is presented. A reference model from the literature is presented to provide the context to the VE of the DT. To understand the full potential of this VE, a conceptual modeling approach is presented. A correct balance should be struck between the graph–model complexity, speed and accuracy. New methods are needed that quickly capture the underlying structure of the graph–model and generate a reduced representation of that model for faster and less resource intensive computation of the target quantity. This is achieved with the GBMR method.

3. Graph-Based Model Reduction Method

The DT is a living hybrid cross-disciplinary model of a real entity. For this, many cross-platform parameters need to be coupled together. Model fusion facilitates this coupling process by bringing several high-fidelity models from domains described in the DT reference model in Figure 2 to create a unified model that provides specific digital services. Model fusion is a two-stage process where the first stage is building the multi-disciplinary model by coupling model parameters. The second stage is simplifying that model for faster interpretability and the prediction capacity of target parameters with model reduction. The GBMR method was first presented in [38]. This section describes the GBMR method as a model fusion strategy for building faster and more accurate DTs. The GBMR method is shown in Figure 4.


Extract Initial Causal graph with DACM
Optimize the Hybrid Graph with GES
Import graph from graph library*
Extract the Hybrid Graph
Get the smallest eigenvalue
Run Spectral clustering algorithm
High Importance cluster (C_H)
subset
X_H
Is it 0?
No
Generate pseudo edge to form connected graph
define [t]
Is it a match***?
Yes
Group variables into high importance (X_H) and low importance (X_L)
Select cut off score**
Define f_x(obj)
Run EVC algorithm KC algorithm PR algorithm
refine $M_{C(i)} = \{m_{C(i)}(h), m_{C(i)}(l), m_{C(i)}(\varnothing)\}$
$X_{H(RFR)}$
No
No f_x(obj) param>100
Formulate Multiobjective optimization problem based on XH
Run Evidential Reasoning algorithm
Get variable importance based on Evidential Reasoning, $M_{evidential}$(i)
RFR method with Gini optmizer
Run optimization using GA solver
Yes

* Import all graph components of the machine system
** Earlier approach of selecting cutoff on node importance score
*** The benchmarking process takes place here

Figure 4. The Graph-based Model Reduction Method.

3.1. Initial Graph

The initial graph is the causal graph extracted from the physics defined by the analytical models of the PE. The parameters from FEM models or system-level models are used as inputs to the DACM method. The output of the DACM method is the causal graph containing all nodes and edges from physics-based equations. An alternative to DACM for building the initial network are graph structure learning algorithms from data such as Bayesian networks [39]. The GBMR was combined with a Bayesian network to obtain the variable importance in the causal graph [32]. In the Bayesian approach, the conditional probability of each node on the causal graph is calculated based on the available data. Sensitivity analyses were performed to find out the most responsive parameters using a Bayesian inference engine. The initial graph contains information about the relationship between the node and the weighted edges. It also contains information about the target variables that the DT needs to optimize.

3.2. Hybrid Graph

The GBMR process starts with building the initial graph. However, this initial graph is a physics-based representation. The DT is a hybrid representation. There is a need to inject process data into the initial graph generated in the previous step to build such a hybrid graph. Hence, process parameter data is collected with IoT platforms. When the parameter graph and data are available, an analogy modeling technique is followed to generate the relationship between the parameters. For example, the relationship between the power consumption (P), voltage (V) and current (I) of a machine could be established based on the popular relationship $P = V \times I$ with the initial graph, as shown in Figure 5A. If the data obtained from the machine contains datasets such as active power (A(P)), active voltage (A(V)) and active current (A(I)), the datapoints could be appended to the initial graph and a hybrid graph could be established, as shown in 5b.

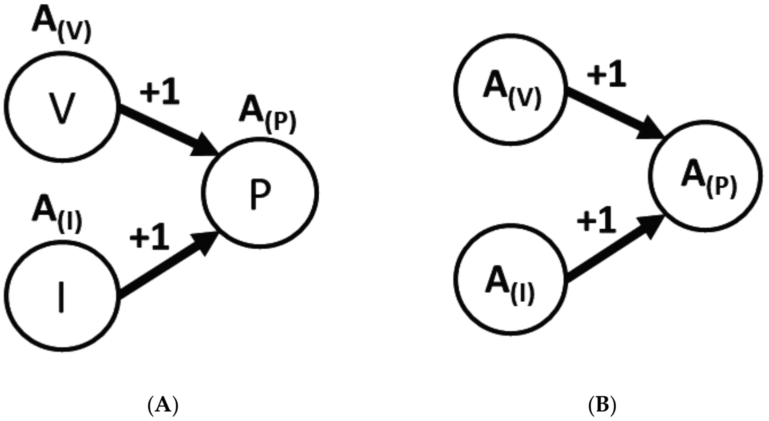


Figure 5. (**A**) DAG representation from initial graph. (**B**) Hybrid graph representation.

3.3. Heuristic Method for Hybrid Graph

The hybrid graph consists of several parameters. Though the relationship is defined by the underlying physics, it is not necessary that a relationship exists between the two entities. Hence, a heuristic approach is taken in this article. A greedy search algorithm, such as the GES, is applied to construct the final graph. As mentioned in the Section 2.3.2, the GES progressively generates and removes edges in the graph with FES. It finally stops at the point where the BES hits the local maxima. This process is computationally extensive. For graphs with a high number of nodes, this process is performed in stages. The GES algorithm is suited for high-dimensional datasets. GES was applied in for causal model discovery in directed graphs such as the hybrid network in [40]. The GES provides the final graph, which serves as an input to the model reduction method.

3.4. Graph Spectral Cluster

The evolution of the GBMR method continued when more fundamental questions were raised about the causal graph structure. The hybrid graph input to the GBMR method assumes that the causal graph structure is complete when the node-importance measuring algorithms are run on it. This is indicated by a * sign in Figure 4 when importing the graph from the graph library. That means that the graph structure does not change at runtime (no nodes or edges are added or removed). Hence, it becomes possible to segment the bigger graph into structurally similar chunks. A spectral clustering-based graph-cut method for DAGs was used for this purpose [41]. The spectral clustering method learns the graph structure and provides the hierarchical clusters based on the graph Laplacian [42]. The spectral clustering algorithm classifies the parameters into cluster membership based on the adjacency matrix. The spectral clustering algorithm computes the normalized graph Laplacian such that:

$$L_n = D^{-1}L \tag{12}$$

where L is the un-normalized graph Laplacian and D is the degree matrix. The algorithm defines three parameters: the number of clusters in which the graph should be split, the affinity or adjacency matrix of the graph and the random state used to initialize the graph decomposition method. Because of this graph decomposition, the nodes that belong to similar clusters can be identified that might have similar behavior in the graph.

3.5. Importance Measurement

In GBMR, graph centrality methods such as WPR and EVC are used for the hybrid graph to identify the important nodes. If the smallest eigenvalue is 0, the node importance can be measured. When these nodes are identified according to the WPR score, they are compared with the spectral clustering algorithm result. When both the node centrality algorithms and spectral clustering rank the node as high, the node is declared as an important node. If the eigenvalues are not zero, the hierarchical node rankings could be applied. This check is made in the GBMR method soon after the hybrid graph is obtained. If the hybrid graph provides zero or negative eigenvalues, that means the graph is not complete. There are some edges that are connected in the wrong locations or some nodes that are not connected at all. In either case, the input graph is invalid. If there are nodes that cannot be connected in a legitimate graph with all other nodes and edges in place, a pseudo-node is generated to the nearest neighbor to make the graph valid.

The WPR algorithm uses three important parameters to reach the final score. The maximum iterations allow users to define how many iterations of the PageRank should be performed. In most cases, the WPR score stabilizes after 100 iterations. The damping factor α is defined as 0.85, which indicates a gradual convergence towards the final score. Finally, the weights on the directed graph are added as per the power law defined by DACM. A threshold is selected by combining the domain-specific expert knowledge and piecewise linear regression of the centrality score. The development of a generic method of threshold measurement is currently being researched and should be treated as a future direction of this article. Hence, the variables can be moved into a high importance matrix [X_H] and a low importance matrix [X_L].

3.6. Consolidation of Importance

The GBMR method further evolved when the eigenvector centrality algorithms such as EVC and Katz were compared with the output obtained from the modified PageRank algorithm; the match of important parameters was less than 60%. It was discovered that the accuracy of the node importance depends on the ranking method selected. To address this issue, evidential reasoning techniques such as the Dempster–Shaffer theory were used for consolidating the node importance scores. The DST complements the GBMR method by providing a belief structure to decide when a node is considered of high importance. The mass functions are calculated and $M_{evidential}(i)$ was computed. Even though the node is differentially ranked by the node ranking algorithm, the DST helps to take a decision by fusing the available information of the nodes that are contradictory in their ranking. An aggregated node ranking is achieved with the application of the DST, which is used to select the parameters for DT the optimization problem [38].

3.7. Validation

The GBMR method is validated by benchmarking it against a machine learning method known as the random forest regressor (RFR) with its associated Gini importance. RFR with Gini importance is a fast and accurate way of analyzing the feature importance in high dimensional data. The benchmarking process provides an estimate of the relevance of the important parameters. Gini provides a superior method of feature importance measurement than other methods such as PCA. The RFR and Gini optimizer provide a parameter ranking stored in X_H(RFR). X_H(RFR) is compared with X_H to check the number of parameters declared important by both methods. It may not be a one hundred percent match. A threshold [t] is defined at this stage. If the match percentage is below [t], the mass functions and BPA from Equation (6) is reevaluated and the process is run again.

An alternative approach to the RFR method was to analyze the GBMR method by the formulation of an optimization problem. This is shown in Figure 4 by the dotted boxes. The problem was formulated as a multi-objective optimization case. The objective function is setup with the target variables of the DAG and the parameters from X_H used to attain the target variables. An empirical approach was taken to determine a Pareto efficient solution. However, formulating such a multi-objective optimization problem is challenging in situations containing a high number of parameters. Multi-objective problem formulation cannot be meaningfully developed when the number of parameters is high.

At this stage, the major component of the GBMR method is explained. Now, the validity of this method is demonstrated with the help of a turbo compressor case study in the next section.

4. GBMR Case Study with a Turbo Compressor System

This case study demonstrates the GBMR method for building the DT of a turbo compressor system. The purpose of this case study is to showcase the following:

1. DT as a graphical representation of a system, built to provide specific outcomes;
2. Utilize the GBMR framework to demonstrate how this conceptual DT is useable in a real scenario providing a digital service;
3. Validation by benchmarking the GBMR method against machine-learning-based methods.

4.1. The Graph-Based Model of the Turbo Compressor System

The GBMR method is applied on the graph-based representation of the system. The initial graph is constructed with DACM, as described in Figure 4. Conditional restrictions are applied on the hybrid graph by the turbo compressor system such as (1) the graph has to be completed. All nodes should have at least 1 edge connecting it to the whole graph, and (2) the target parameters are identified by propagation of strategic objectives in the causal graph. The causal graph is at least checked for loops and contradictions and they are removed. The parameters from the initial graph are appended to the system level model and the hybrid graph is established. The hybrid graph is treated as the input to a GES algorithm. GES algorithms use a sequence of forward and backward searches to create the final hybrid graph. The final graph is the knowledge model that consists of the details from both physics-based and data domains. The final graph is used as an input to the model reduction process. The need of a DT is to understand and mitigate the effects of dynamic instabilities in the turbo compressor system.

There are two types of dynamic instabilities known as stall and surge [43]. Stalling is a complex flow instability originating from regions of flow stagnation that are created near the impeller blade confinements of the centrifugal compressor known as stall cells. Turbo compressors are at risk of developing stall cells that result in eventual impeller failure. Stalling can be progressive or abrupt in nature. A progressive stall is the more common and riskier form of stall. Stalling is a precursor to surge, which is the principal destabilizing phenomenon in turbo compressors. Due to stalling, the compressor cannot generate enough pressure at the outlet to match the pressure built up inside it. This forces the compressed air to flow back towards the inlet, resulting in a rapid asymmetrical oscillation known as surge oscillation. Hence, surge mitigation requires precise and accurate modeling and control methods. For that, the Greitzer compression system model is adopted as a foundation for building the graph-based VE [44].

4.1.1. The Initial Graph Development

Surge is a gradual build-up phenomenon whose occurrence can be understood by monitoring the pressure rise in the compressor (ψ_C), the plenum pressure rise (ψ_P), mass flow rate of compressed air (ϕ_P) and mass flow rate at the throttle (ϕ_{th}). These variables and their interrelationship are defined by the Greitzer compression system model, which provides a lumped parameter model of the compression system. The mathematical equation needed to build the Greitzer compression system model is described in equations A.1 through A.9 in Appendix A. This model provides the basic physics-based causal formulation of turbo compressor system.

The causal model is used to extract and represent the causal relationship between the variables in the Greitzer compression system model. The result graph obtained is shown in Figure 6. This network is built to optimize the stability of the compression system. The stability is quantified with ψ_C, ψ_P and ϕ_P from the Greitzer compression system model. This is indicated with red color variables. These are called performance variables in the conceptual modeling nomenclature. The variables in red color are the target variables defined by the system. The blue variables are the dependent variables that are dependent on the green variables known as independent variables. The black variables are exogenous variables that are not affected by external or internal change. The weights on the edge of the network are assigned as per the power law and utilized in the GBMR approach.

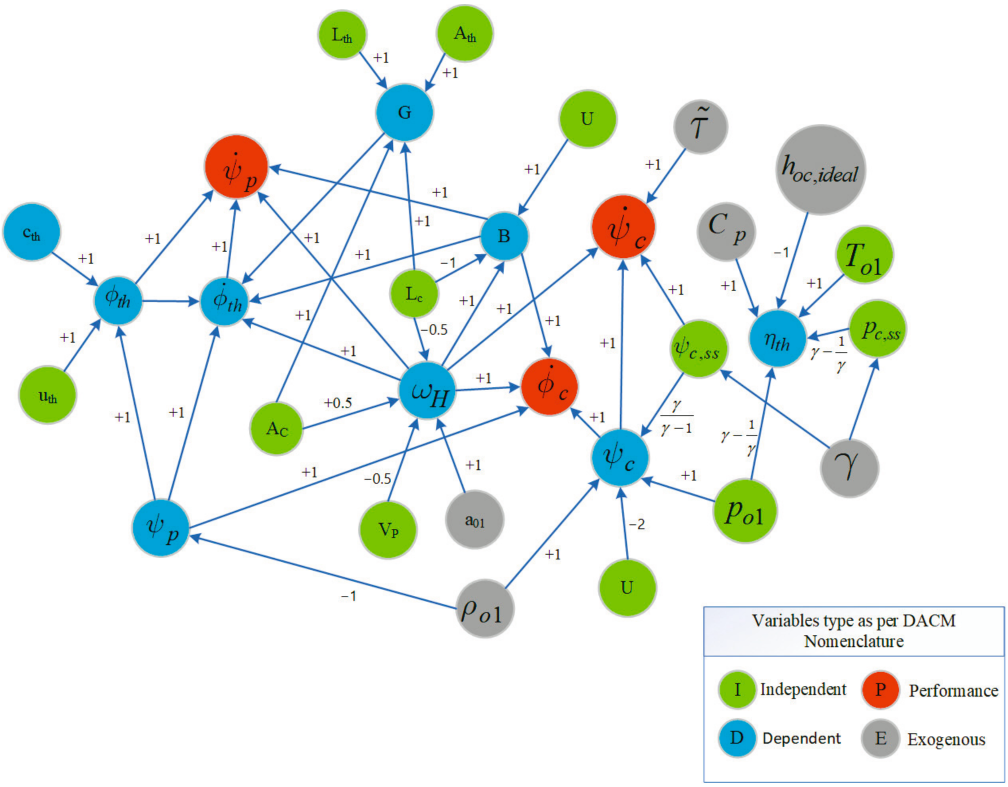


Figure 6. The physics-based initial graph.

4.1.2. The Hybrid Graph Development

The initial graph captures the physics-based representation of the system; however, it is bound to the steady state of the system and its implication is theoretical. In earlier studies, efforts were made to define the practical usage of this causal graph entity using theories from artificial intelligence. In this article, the utility of the physics-based initial graph is set to guide the development of a graph-based representation of the system-level model. System-level models, such as those in the Figure 7C, represent the dynamic state of the system that is related to the physical phenomena guiding it. In the VSD unit of turbo fans and motor systems shown in Figure 5B, the variables from the dynamic system model could be organized and the causality between these variables can be extracted guided by the initial graph (i.e., the Greitzer compression system model). Therefore, embedding the parameters or the network from Figure 4 into the relevant sections of Figure 7C, a hybrid of the physics-based and the systems-level models is created. This combination of the physics-based and systems-level models is known as the hybrid graph. The resulting hybrid graph is the dynamic system-level model in DAG form.

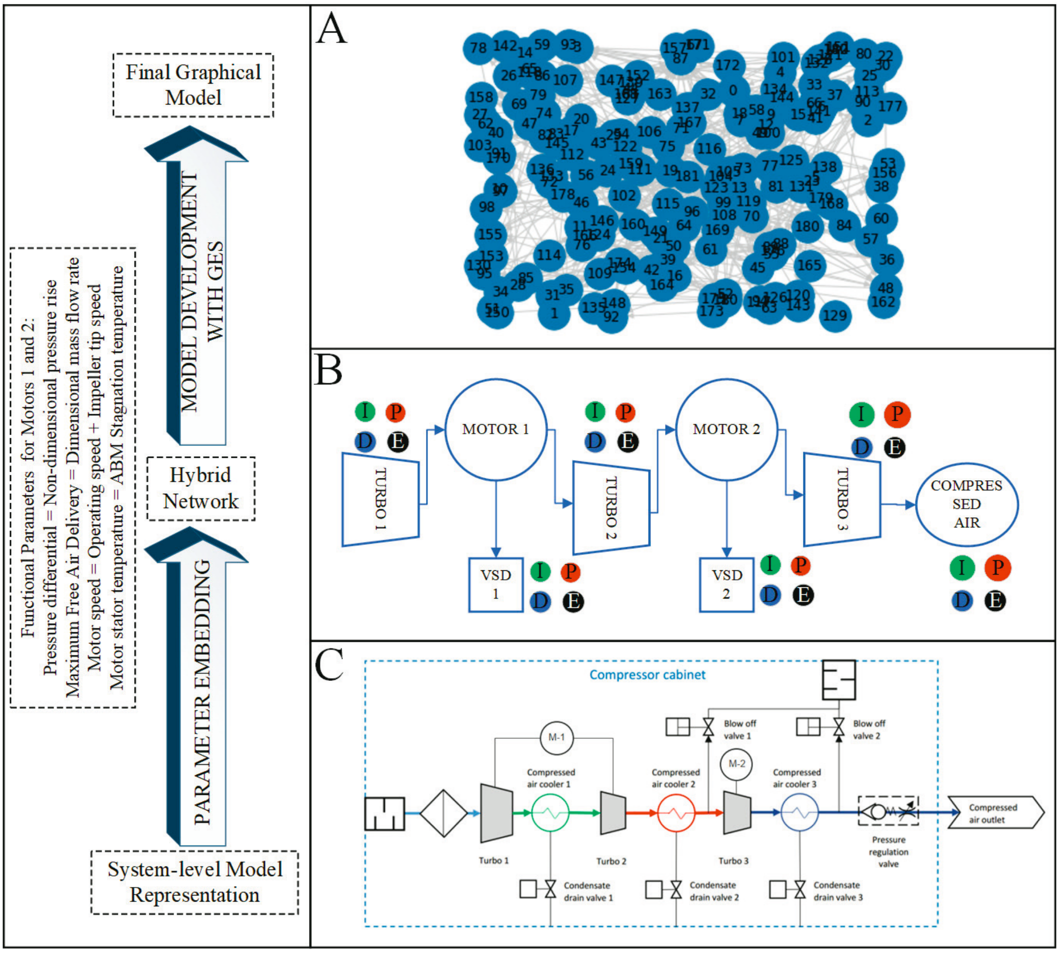


Figure 7. Parameter assignment and causality extraction of hybrid model. (**A**) The hybrid model; (**B**) Parameter extraction; (**C**) The system-level model.

The methodology described in Section 3.2 is used to build the hybrid graph. The factors affecting the pressure differential parameters in the three turbo motors and compressor system in the systems-level model: pressure differential, maximum free air delivery, motor speed and motor stator temperature, could be associated with a non-dimensional pressure rise, dimensional mass flow rate, impeller tip speed and stagnation temperature, respectively, from the initial graph. Similarly, the surge limit could be connected to the set pressure and the motor rpm could be connected to the active power based on the physics of the system. The same methodology is repeated to create the causal model of the whole system. Additional parameters where this methodology cannot be applied such as state variables or ON/OFF signals are considered independent if they could be connected to a dependent variable. If the variable has no association with the physics of the system, the assumption is made to consider it as exogenous.

4.1.3. GES Algorithm

The hybrid graph serves as the input to the GES algorithm as described in Section 2.3.2. The graph is transformed into an adjacency matrix embedding the relationship on the edges with $\pm n$ weight signifying the presence of a relationship. Here, n is the weight on the edges of the initial graph. FES and BES maxima are computed with Gaussian likelihood score with the python library known as sempler [45]. The result of the GES algorithm is the hybrid graph that serves as an input to the GBMR method.

5. Results and Discussion

5.1. Node Importance

The hybrid network is a complete knowledge model in line with the VE representation of the DT, embedding the relationship obtained from the underlying physical phenomena, the system-level information model and the data collected from the turbo compressor system. The final network contains 183 nodes. This provides a good use case for demonstrating the GBMR method. The result of the ranking algorithms is presented in Figure 8A, and a spectral decomposition of the hybrid graph is shown in Figure 8B.

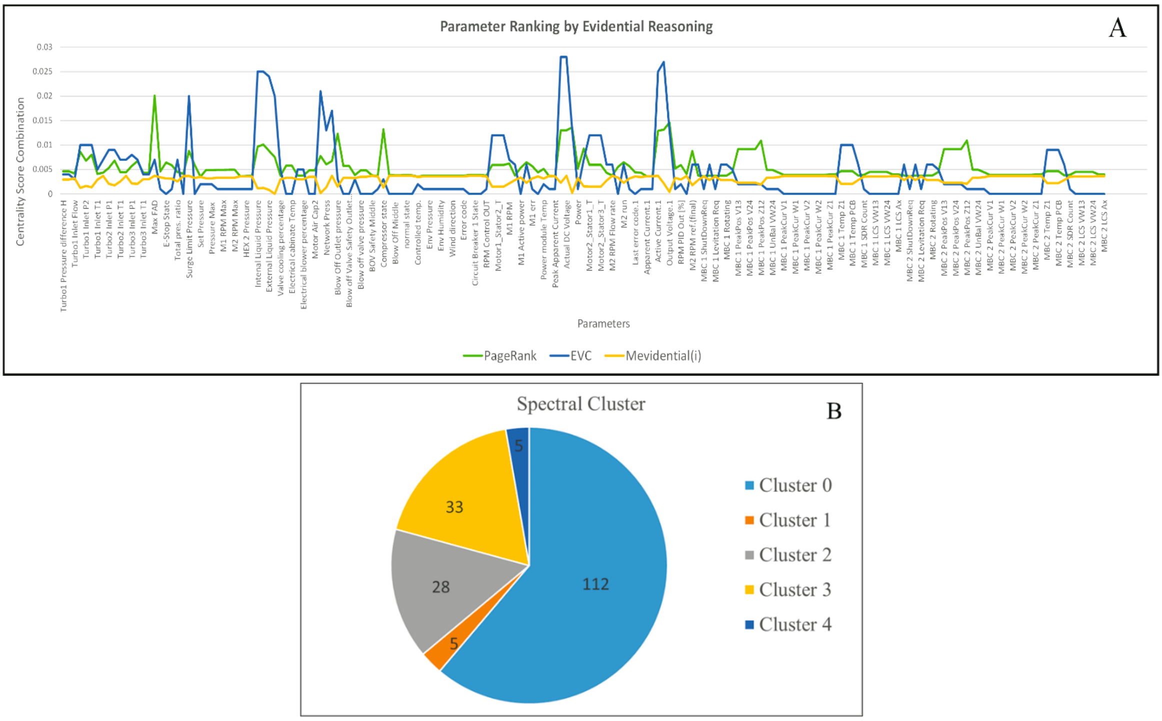


Figure 8. (**A**) Parameter ranking by evidential reasoning algorithm. (**B**) Cluster membership of nodes.

The spectral method and the node ranking methods are conducted parallelly. The spectral clustering algorithm is applied to obtain the structure preserving graph cuts. The spectral clustering algorithm is developed with python API that uses the clustering of normalized Laplacian. The eigenvector plot clearly indicates that there are five peaks with zero eigenvalues, indicating the presence of five ways that the graph could be partitioned. The cluster membership result of the parameters is indicated in Figure 8B. Cluster0 contains 112 parameters; the second biggest cluster, cluster3, contains 33 parameters; the third biggest cluster, cluster2, contains 28 parameters. The two remaining small clusters contain five parameters each. If the clusters are reconstructed with the number of clusters as six, no change was observed in cluster0, cluster2 or cluster3. The smaller clusters start fragmenting as number of clusters increases. It is interesting to note that the small clusters contain nodes that are farther from the center of the graph, where the target variables are located. The target variables such as FAD, motor voltage and active power consumption all belong to the cluster0.

The WPR and EVC scores are shown in Figure 8A. The WPR scores of nodes in cluster0 are consistent. The intersection of the high WPR scores with cluster0 are agreeable, except the environmental parameters that were added to the dataset externally. The highest WPR score was given to maximum FAD. This is consistent with the hybrid graph, as FAD was considered as a target variable. The WPR also nominated motor speed as important, which was associated with FAD in the hybrid graph. EVC agrees with the WPR in many cases; however, EVC differs in the ranking scheme given by WPR. State variables such as the motor state and turbo state differ considerably between EVC and WPR. Hence, the consolidated ranking results are obtained from the node importance algorithms with DST. This is indicated by the difference between the blue and green lines in Figure 8A. DST was applied to this hierarchical ranking system to find a combined ranking. This is indicated by the yellow line and marked as the final evidential mass or $M_{evidential}(i)$. DST is able to consider all the available pieces of evidence in deciding whether a node is important or not and can give a trusted final decision. After application of DST, the nodes were rearranged, and the nodes lying in the intersection of cluster0, cluster2 and cluster3 with $M_{evidential}(i)$ were considered as important nodes. These nodes were stored in a high importance parameter matrix [$\mathrm{X_H}$] and the other variables were kept in a low importance parameter matrix [$\mathrm{X_L}$]. The GBMR process is documented in [46].

Because so many variables were declared as important (173 params), a cutoff was set to the $M_{evidential}(i)$ score. When a threshold of lower limit of 25% of the $M_{evidential}(i)$ score was selected, the number of important parameters obtained was 145. So, a 22.4% reduction in parameters was obtained. That is, with 22.4% fewer parameters, the hybrid graph will be able to compute the target variables. The values of the target variables were measured from the reduced model and computed and compared with the values from the literature presented in [43]. These values are free air delivery and pressure differential for all turbo compressors ([Turbox_OUT_P]-Turbox_IN_P), where x is 1, . . . ,3. The error in the free air delivery is ($e = FAD - \dot{\phi}_C$) and is less than 4%. The error in pressure differential is e = (Turbox_OUT_P-Turbox_IN_P)-$\dot{\psi}_C$ and is less than 6%. The maximum value from the DST scores is 0.0037. Considering a lower threshold than 25% results in the selection of 178 parameters (or a 2.73% reduction). This makes the GBMR very inefficient. On the other hand, a higher threshold eliminates important parameters ranked highly by both EVC and WPR. This result is valid for the selected threshold, which is obtained by a numerical analysis. For a different threshold, the percentage reduction will vary. It should be mentioned here that an accurate method for obtaining the threshold is under consideration and should be realized as a future direction of this research. These parameters are used to construct a benchmarking procedure for the turbo compressor case study.

5.2. Validation of GBMR Method through Benchmarking

A validation by benchmarking approach was designed for the GBMR method when applied to the VE representation of the DT. The benchmarking method compares the GBMR method with machine-learning-based approaches applied to the hybrid model of turbo compressors. The validity of the method can be proven in case more complex models are used. This is because GBMR uses graph-based methods and algorithms to build and reduce the complex model. A complex model will generate a complex graph. The spectral decomposition and node importance algorithms used in GBMR can be applied to any type of complex graph. Classical methods such as the RFR [47] or a mixture of the RFR and a deep learning method such as CRNN [48] are used to find the statistically significant parameters that contribute most to explaining the target outcomes when a large number of feature sets are present in the data. Such approaches become necessary because, unlike other domains, large training datasets are not readily available for identifying important variables for compressor surge prediction.

The RFR is a type of ensemble method machine learning algorithm. It consists of several decision trees, each constructed based on a randomly selected subset of the training data set. CART, a popular training algorithm, is used for this. The training indicated that the individual trees learn some features in the training data and the ensemble of all trees learn the features present in the turbo compressor dataset. The estimation happens by voting or sampling some statistical value for the estimations of the individual trees [49]. A by-product of the training of the RFR is combined feature importance. Feature importance is a method of ranking features based on how much each feature reduces the impurity of the estimation through the nodes of the decision trees. The importance of each feature is calculated by normalizing the total reduction in impurity that the feature causes. The feature importance identification can be performed by two methods: Gini importance [50] and permutation importance [51]. It was experimentally observed that the importance distribution in Gini could be skewed by ranking some features much higher than others. Hence, the permutation importance was used to estimate the importance of the features. The permutation importance is calculated by using a trained estimator and dataset, such as the RFR. The algorithm first uses the estimator to calculate a baseline output using some metric on the dataset, then permutes a feature of the dataset and re-calculates the output metric. This is repeated with the other features. The difference of these metrics then indicates the importance of each feature in the dataset with respect to the estimator output.

The dataset used to build GBMR is used for validation with the RFR method. The dataset consists of all parameters from the turbo compressor system shown in Figure 7C. The data is split into three zones for three turbo compressor fans. It contains physical parameters such as the inlet and outlet pressure, inlet and outlet temperature of rotor and stator, pressure ratios, maximum free air delivery (MAX. FAD), RPM of fans, active power consumption of motors, peak currents, voltages and positions of the motors, internal liquid flow temperature and levels and cooling valve temperatures. It also contains state parameters such as rotor stop state, fluid rise state, speed control state, error states and compressor states. Finally, it consists of operational data such as surge limit and environmental data such as atmospheric temperature, pressure, humidity, and dew points.

Experiment Design for Validation

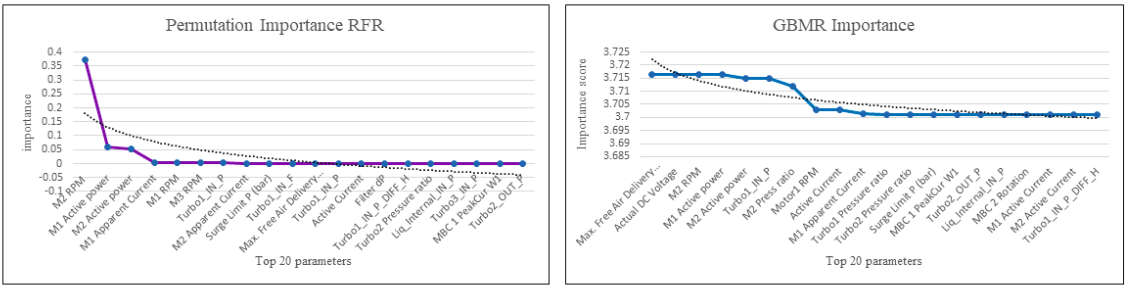
The RFR method is chosen in this study to benchmark the results obtained from the GBMR method. The RFR algorithm was built with a python-based machine learning library scikit-learn. An experiment was designed to achieved that. These data were obtained from an industrial turbo compressor system. The RFR was trained with 517,902 samples using the dataset described above with 183 variables as inputs. The target of the training was the power variable calculated as a sum of power variables from the two-motor assembly in the VSD compressor line. A hold-out test data set of 129,475 was left out of the training data. The hyperparameters of the regressor training were as follows:

Number of trees = 200;
Maximum depth of a tree = 10;
The minimum samples required for a split = 2;
The accuracy of the trained model on the training data = 0.9998;
Holdout test dataset accuracy = 0.9997.

The top 20 important variables from permutation importance and GBMR are listed in Table 1. The parameters are presented in order from high to low. Figure 9 presents the top 20 parameters obtained from the two methods. The common parameters selected by the two methods are more important than the order of the parameters. The parameters with higher importance score are the most important parameters obtained from both of the methods. In the benchmarking process, the parameter set obtained from the permutation importance serves as a control.

Table 1. Parameter benchmarking between the GBMR and RFR methods.

Node Importance Rank	GBMR	Gini Permutation Importance Rank of Nodes	RFR
1	Max. Free Air Delivery (cal.)	1	M2 RPM
2	Actual DC Voltage	2	M1 Active Power
3	M2 RPM	3	M2 Active Power
4	M1 Active Power	4	M1 Apparent Current
5	M2 Active Power	5	M1 RPM
6	Turbo1_IN_P	6	M3 RPM
7	M2 Press Ratio	7	Turbo1_IN_P
8	Motor1 RPM	8	M2 Apparent Current
9	Active Current	9	Surge Limit P (bar)
10	M1 Apparent Current	10	Turbo1_IN_F
11	Turbo1 Pressure Ratio	11	Max. Free Air Delivery (cal.)
12	Turbo2 Pressure Ratio	12	Turbo1_IN_P
13	Surge Limit P (bar)	13	Active Current
14	MBC 1 PeakCur W1	14	Filter dP
15	Turbo2_OUT_P	15	Turbo1_IN_P_DIFF_H
16	Liq_Internal_IN_P	16	Turbo2 Pressure Ratio
17	MBC 2 Rotation	17	Liq_Internal_IN_P
18	M1 Active Current	18	Turbo3_IN_P
19	M2 Active Current	19	MBC 1 PeakCur W1
20	Turbo1_IN_P_DIFF_H	20	Turbo2_OUT_P


Permutation Importance RFR

GBMR Importance

Figure 9. Permutation importance vs. GBMR importance.

5.3. Discussion

A sample of the top twenty parameters is taken to demonstrate the accuracy of the GBMR method compared with the RFR method. Both methods capture the significant parameters from the three motors and main compression parameters. Both methods return active current and active power as important parameters. This is in line with the hybrid graph, where the current and voltage were considered as inputs to several parameters including the turbo motor pressure and temperature. Both methods indicate the motor RPM as an important parameter; however, the RFR indicates it is the most significant variable by a large margin, with a mix of power and current variables also indicated. This is because of the permutation importance; the peak is distinctly noticeable in Figure 9. The significance of Motor2 RPM is likely because it explains the operational condition of the motor that runs the last compressor stage and therefore works with the highest pressures. This is clearly linked to the highest energy consumption rate, making the apparent current, active current, output voltage and active power crucial parameters in surge control. The compressor state is correlated with the maximum free air delivery, which in turn is related to the active power of the system. The air pressure value at the intake and compressor-control-related variables are also seen as important parameters. A calculated parameter, surge limit, is marked as an important parameter by both RFR and MR. The surge limit is computed from the pressure rise in the compressor and the maximum free air delivery. This is because of using node importance methods to rank order the parameters. The parameter pointing to important parameters is also considered important.

In the sample of the top 20 parameters, 15 parameters match from both methods. This amounts to a 75% match between the two methods. The RFR is a data-driven method and does not contain any knowledge of the physics of the parameters involved. The GBMR, on the other hand, works on the hybrid representation. The 75% match indicates that both methods attempt to identify the common important features in the parameters. The percentage match between the parameters obtained from the two methods increases up to 78% when top 100 parameters are considered. The motor currents and voltages are identified as important parameters in the [X_H] set, as they point to the high importance parameter active power. The parameters in [X_L] are ignored for benchmarking purposes. When the values of [X_L] are considered, the benchmark percentage will vary. GBMR still provides a fast method to quickly capture the influential system parameters. The model for the RFR is retrained, but the basic result of M2 RPM importance remains the same with the existing data from the turbo compressor. The GBMR results seem similar because of the application of evidence theory. Evidence theory tends to remove any disagreement between different ranking score and provides the relative importance for the parameter. This is indicated by the yellow curve in Figure 8A.

To understand the simplification of the computational complexity obtained by GBMR, it is possible to compare the training time of the RFR method with the total execution time of the GBMR method. The training time of the RFR method is 81.3471 min. The total execution time of the GBMR method, which is a summation of graph structure learning, spectral decomposition and importance measurement, is 72.2318 min. Hence, GBMR is 9.1153 min faster in identifying the reduced model.

6. Conclusions

This work presents the research activities on the VE optimization of the DT with the help of a model fusion technology. The VE is a computationally complex entity, comprising of models from different domains, that tries to faithfully replicate the state of the PE. The VE combines advanced simulation models such as system-level models with data-driven prediction models to predict the state of the PE. This article describes the GBMR method for optimizing the performance of the VE with a two-step approach: (1) providing a graph-based conceptual model representation of the VE, and (2) reducing the VE graph model by identifying the important parameters in it. The GBMR embeds all the parameters and their relationships in a graph model and facilitates application of graph algorithms for measuring node importance. Therefore, GBMR facilitates modeling of physical systems in the form of graphs and the reduction of such models based on graph algorithms. The GBMR makes the VE more efficient, as the reduced model uses a subset of parameters to predict the target parameters in the PE.

The GBMR method is tested with the help of a turbo compressor case study. The GBMR method is benchmarked against a machine-learning-based approach known as the random forest regressor, which also estimates the important parameters in a given dataset with the help of permutation importance. Both the GBMR and RFR methods were applied on the turbo compressor dataset, and it was found that both methods find 75% common parameters in no fixed order. These important parameters bear maximum contribution towards the performance of the VE.

Methods such as GBMR become important in the context of DTs because they help to simplify the VE development but still capture both the physics-based and data-driven aspects of the twin. With the help of the GBMR, the DT becomes more context aware by knowing the important parameter that it needs to monitor and optimize to obtain the fastest result. The GBMR method aids in the fast computation of the target parameters that the

DT is trying to replicate by capturing the intricacies of a multi-domain system. It will also become imperative to integrate methods such GBMR with the DT frameworks and perform computational tests on entire complex DT systems such as the turbo compressor system described in this article. It is possible to convince PE owners about what is essential in their system with GBMR and how resource allocation and optimization can be performed effectively with DTs. The GBMR method is a python-based software package that can be used as a virtual sensor or a prototyping tool where quick estimation regarding the system and the effort needed to build a VE representation can be analyzed effectively.

Author Contributions: Conceptualization, A.C. and H.V.; methodology, A.C.; software, A.C.; validation, A.C., H.V. and K.T.K.; formal analysis, A.C.; investigation, A.C.; resources, J.L.; writing—review and editing, A.C., H.V. and K.K; visualization, A.C.; supervision, K.T.K.; project administration, K.T.K.; funding acquisition, K.T.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Business Finland under the project named SNOBI, and the project number 545/31/2020.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the SNOBI project funded by Business Finland for making this research possible. The authors would also like to acknowledge the contribution of Tamturbo Oy for their support towards this research work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in the manuscript:

AMB	Active Magnetic Bearing
AI	Artificial Intelligence
ANN	Artificial Neural Network
CERN	European Organization for Nuclear Research
CRNN	Convolutional Recurrent Neural Network
DAG	Directed Acyclical Graph
DST	Dempster–Shafer Theory
DT	Digital Twin
EVC	Eigenvector Centrality
DACM	Dimension Analysis Conceptual Modeling
FEM	Finite Element Model
FMI	Functional Mockup Interface
FMU	Functional Mockup Unit
GES	Greedy Equivalent Search
GBMR	Graph-based Model Reduction
IP	Internet Protocol
IPv6	Internet Protocol version 6
IoT	Internet of Things
ML	Machine Learning
M2M	Machine-to-Machine
M2S	Machine-to-System
PCA	Principal Component Analysis
PE	Physical Entity
RFR	Random Forest Regressor
S&M	Simulation and Modeling
VE	Virtual Entity
VSD	Variable Speed Drive
WPR	Weighted PageRank

Appendix A

List of equations for surge modeling

Appendix A.1. Greitzer Compression System Model

$$\psi = \frac{\Delta p}{\frac{1}{2}\rho_{o1}U^2},$$

$$\phi = \frac{m}{\rho_{o1}UA_c}$$

ψ: non-dimensional pressure rise

ϕ: non-dimensional mass flow rate

Δp: dimensional pressure rise

m: dimensional mass flow rate

ρ_{o1}: density @ inlet condition (ambient)

U: impeller tip speed

A_c: Area of the compressor duct

Appendix A.2. Helmholtz Frequency (ω_H)

$$\omega_H = a_{o1}\sqrt{\frac{A_c}{V_pL_c}}$$

L_c: length of compressor duct

V_p: volume of plenum

a_{o1}: speed of sound in @ inlet (ambient) condition

Appendix A.3. Original Greitzer Compression System Model with Non-Dimensional Variables

$$\left.\begin{array}{l}\frac{d\phi_c}{dt} = B\omega_H(\psi_C - \psi_P) \\ \frac{d\phi_{th}}{dt} = \frac{B\omega_H}{G}(\psi_C - \psi_{th})\end{array}\right\}\text{conservation of momentum of fluid in compressor and throttle duct}$$
$$\left.\frac{d\psi_p}{dt} = \frac{\omega_H}{B}(\phi_C - \phi_{th})\right\}\text{ conservation of mass in plenum volume}$$
$$\left.\frac{d\psi_c}{dt} = \frac{\omega_H}{\tilde{\tau}}(\psi_{c,SS} - \psi_c)\right\}\text{ behavior of dynamic compressor settling}$$

ϕ_c: non-dimensional mass flow rate

ϕ_{th}: throttle mass flow rate

ψ_p: plenum pressure rise

ψ_c: *compressor pressure rise*

$\overset{\sim}{\tau}$: *time constant of the compressor*

Appendix A.4. Greitzer Stability Parameter (B) Governs the Intensity of Surge Instability in the Greitzer Model

$$B = \frac{U}{2\omega_H L_C}$$
$$G = \frac{L_{th} A_c}{L_c A_{th}}$$

L_{th}: *length of throttle duct*

A_{th}: *cross-sectional area of throttle duct*

Appendix A.5. For Subsonic Flows

$$\phi_{th} = c_{th} u_{th} \sqrt{\psi_p}$$

u_{th}: throttle percentage opening

c_{th}:*Constant¯determined experimentally and depends on valve geometry and properties of the fluid*

Appendix A.6. Curve Fitting to Determine the Steady-State Pressure and Flow Rate Measurements

$$\psi_c(\phi) = \psi_{c0} + H(1 + \frac{3}{2}(\frac{\phi}{W} - 1) - \frac{1}{2}(\frac{\phi}{W} - 1)^3)$$

ψ_{c0}: pressure @ 0 flow

H, W: constant computed from pressure rise and flow rate corresponding to surge point (predicted params from curve fitting in the stable region and used to correct ψ_{c0}).

Appendix A.7. Variation of Impeller Tip Clearance with AMB. Isentropic Efficiency of Compressor

$$\eta_{th} = \frac{T_{o1} C_p \left(\frac{p_{c,SS}}{p_{o1}}\right)^{\frac{\gamma-1}{\gamma}} - 1}{\Delta h_{oc,ideal}}$$

T_{o1}: stagnation temperature

p_{o1}: stagnation pressure

C_p: specific heat at constant pressure

$\Delta h_{oc,ideal}$: total specific enthalpy delivered to the fluid

γ: specific heat ratio

$p_{c,SS}$: compressor output pressure at nominal tip clearance $\delta_{cl} = 0$

Appendix A.8. ψ_c (Non-Dimensional Compressor Pressure Rise) as a Function of $\psi_{c,SS}$ and δ_{cl}

$$\psi_c = \frac{p_{o1}}{\frac{1}{2}\rho_{o1}U^2}\left(\left(1+\frac{\left(\frac{0.5\rho_{o1}U^2}{p_{o1}}\psi_{c,SS}+1\right)-1}{1-k_0\frac{\delta_{cl}}{b_2}}\right)^{\frac{\gamma-1}{\gamma}}-1\right)$$

Appendix A.9. Level I Stability Analysis (Initial Screening to Identify Safe Compressor Operations)

$$q_A = HP\frac{B_C C\rho_d}{D_C H_C N\rho_s}$$

q_A: predicted cross-coupling stiffness

HP: rated horsepower

B_C: constant for centrifugal compressors determined experimentally

C: constant for centrifugal compressors experimentally

ρ_d: discharge gas density per impeller/stage

ρ_s: suction gas density per impeller/stage

D_C: impeller diameter

H_C: minimum of diffuser or impeller discharge width

N: operating speed

References

1. Grieves, M.; Vickers, J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*; Kahlen, F.-J., Flumerfelt, S., Alves, A., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 85–113. [CrossRef]
2. Kannan, K.; Arunachalam, N. A Digital Twin for Grinding Wheel: An Information Sharing Platform for Sustainable Grinding Process. *J. Manuf. Sci. Eng.* **2019**, *141*, 021015. [CrossRef]
3. Khan, L.U.; Saad, W.; Niyato, D.; Han, Z.; Hong, C.S. Digital-Twin-Enabled 6G: Vision, Architectural Trends, and Future Directions. *IEEE Commun. Mag.* **2022**, *60*, 74–80. [CrossRef]
4. Gürdür Broo, D.; Bravo-Haro, M.; Schooling, J. Design and implementation of a smart infrastructure digital twin. *Autom. Constr.* **2022**, *136*, 104171. [CrossRef]
5. Jiang, Z.; Lv, H.; Li, Y.; Guo, Y. A novel application architecture of digital twin in smart grid. *J. Ambient Intell. Hum. Comput.* **2022**, *13*, 3819–3835. [CrossRef]
6. Tao, F.; Cheng, J.; Qi, Q.; Zhang, M.; Zhang, H.; Sui, F. Digital twin-driven product design, manufacturing and service with big data. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 3563–3576. [CrossRef]
7. Yang, X.; Ran, Y.; Zhang, G.; Wang, H.; Mu, Z.; Zhi, S. A digital twin-driven hybrid approach for the prediction of performance degradation in transmission unit of CNC machine tool. *Robot. Comput.-Integr. Manuf.* **2022**, *73*, 102230. [CrossRef]

8. Chakraborti, A.; Nagarajan, H.P.N.; Panicker, S.; Mokhtarian, H.; Coatanéa, E.; Koskinen, K.T. A Dimension Reduction Method for Efficient Optimization of Manufacturing Performance. *Procedia Manuf.* **2019**, *38*, 556–563. [CrossRef]
9. Chakraborti, A.; Heininen, A.; Koskinen, K.T.; Lämsä, V. Digital Twin: Multi-dimensional Model Reduction Method for Performance Optimization of the Virtual Entity. *Procedia CIRP* **2020**, *93*, 240–245. [CrossRef]
10. Qi, Q.; Tao, F.; Hu, T.; Anwer, N.; Liu, A.; Wei, Y.; Wang, L.; Nee, A.Y.C. Enabling technologies and tools for digital twin. *J. Manuf. Syst.* **2021**, *58*, 3–21. [CrossRef]
11. Liu, Z.; Meyendorf, N.; Mrad, N. The role of data fusion in predictive maintenance using digital twin. *AIP Conf. Proc.* **2018**, *1949*, 020023. [CrossRef]
12. Darvishi, H.; Ciuonzo, D.; Eide, E.R.; Rossi, P.S. Sensor-Fault Detection, Isolation and Accommodation for Digital Twins via Modular Data-Driven Architecture. *IEEE Sens. J.* **2021**, *21*, 4827–4838. [CrossRef]
13. Selvaraj, P.; Radhakrishnan, P.; Adithan, M. An integrated approach to design for manufacturing and assembly based on reduction of product development time and cost. *Int. J. Adv. Manuf. Technol.* **2009**, *42*, 13–29. [CrossRef]
14. Verbert, K.; Babuška, R.; De Schutter, B. Bayesian and Dempster–Shafer reasoning for knowledge-based fault diagnosis–A comparative study. *Eng. Appl. Artif. Intell.* **2017**, *60*, 136–150. [CrossRef]
15. Tao, F.; Zhang, M.; Nee, A.Y.C. Chapter 6—Cyber–Physical Fusion in Digital Twin Shop-Floor. In *Digital Twin Driven Smart Manufacturing*; Tao, F., Zhang, M., Nee, A.Y.C., Eds.; Academic Press: Cambridge, MA, USA, 2019; pp. 125–139. [CrossRef]
16. Kapteyn, M.G.; Knezevic, D.J.; Huynh, D.B.P.; Tran, M.; Willcox, K.E. Data-driven physics-based digital twins via a library of component-based reduced-order models. *Int. J. Numer. Methods Eng.* **2022**, *123*, 2986–3003. [CrossRef]
17. Fresca, S.; Manzoni, A. POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. *Comput. Methods Appl. Mech. Eng.* **2022**, *388*, 114181. [CrossRef]
18. Wang, M.; Li, H.-X.; Chen, X.; Chen, Y. Deep Learning-Based Model Reduction for Distributed Parameter Systems. *IEEE Trans. Syst. Man Cybern.: Syst.* **2016**, *46*, 1664–1674. [CrossRef]
19. Morimoto, M.; Fukami, K.; Zhang, K.; Nair, A.G.; Fukagata, K. Convolutional neural networks for fluid flow analysis: Toward effective metamodeling and low dimensionalization. *Theor. Comput. Fluid Dyn.* **2021**, *35*, 633–658. [CrossRef]
20. Cui, C.; Hu, M.; Weir, J.D.; Wu, T. A recommendation system for meta-modeling: A meta-learning based approach. *Expert Syst. Appl.* **2016**, *46*, 33–44. [CrossRef]
21. Tao, F.; Zhang, H.; Liu, A.; Nee, A.Y.C. Digital Twin in Industry: State-of-the-Art. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2405–2415. [CrossRef]
22. Tao, F.; Xiao, B.; Qi, Q.; Cheng, J.; Ji, P. Digital twin modeling. *J. Manuf. Syst.* **2022**, *64*, 372–389. [CrossRef]
23. Tao, F.; Zhang, M.; Liu, Y.; Nee, A.Y.C. Digital twin driven prognostics and health management for complex equipment. *CIRP Ann.* **2018**, *67*, 169–172. [CrossRef]
24. Tao, F.; Zhang, M. Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing. *IEEE Access* **2017**, *5*, 20418–20427. [CrossRef]
25. 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. [CrossRef]
26. Mokhtarian, H.; Coatanéa, E.; Paris, H.; Mbow, M.M.; Pourroy, F.; Marin, P.R.; Vihinen, J.; Ellman, A. A Conceptual Design and Modeling Framework for Integrated Additive Manufacturing. *J. Mech. Des.* **2018**, *140*, 081101. [CrossRef]
27. Wu, D.; Coatanéa, E.; Wang, G. Employing knowledge on causal relationship to assist multidisciplinary design optimization. *J. Mech. Des.* **2019**, *141*, 041402. [CrossRef]
28. Chickering, D.M. Optimal Structure Identification With Greedy Search. *J. Mach. Learn. Res.* **2002**, *3*, 507–554.
29. Hauser, A. Characterization and Greedy Learning of Interventional Markov Equivalence Classes of Directed Acyclic Graphs. *J. Mach. Learn. Res.* **2012**, *13*, 2409–2464.
30. Zhang, W.Y.; Zhang, S.; Guo, S.S. A PageRank-based reputation model for personalised manufacturing service recommendation. *Enterp. Inf. Syst.* **2017**, *11*, 672–693. [CrossRef]
31. Chen, D.; Lü, L.; Shang, M.-S.; Zhang, Y.-C.; Zhou, T. Identifying influential nodes in complex networks. *Phys. A: Stat. Mech. Its Appl.* **2012**, *391*, 1777–1787. [CrossRef]
32. Hu, P.; Fan, W.; Mei, S. Identifying node importance in complex networks. *Phys. A: Stat. Mech. Its Appl.* **2015**, *429*, 169–176. [CrossRef]
33. Henni, K.; Mezghani, N.; Gouin-Vallerand, C. Unsupervised graph-based feature selection via subspace and pagerank centrality. *Expert Syst. Appl.* **2018**, *114*, 46–53. [CrossRef]
34. Shang, Q.; Deng, Y.; Cheong, K.H. Identifying influential nodes in complex networks: Effective distance gravity model. *Inf. Sci.* **2021**, *577*, 162–179. [CrossRef]
35. Bonacich, P.; Lloyd, P. Eigenvector-like measures of centrality for asymmetric relations. *Soc. Netw.* **2001**, *23*, 191–201. [CrossRef]
36. Mo, H.; Deng, Y. Identifying node importance based on evidence theory in complex networks. *Phys. A: Stat. Mech. Its Appl.* **2019**, *529*, 121538. [CrossRef]
37. Ghosh, N.; Paul, R.; Maity, S.; Maity, K.; Saha, S. Fault Matters: Sensor data fusion for detection of faults using Dempster–Shafer theory of evidence in IoT-based applications. *Expert Syst. Appl.* **2020**, *162*, 113887. [CrossRef]
38. Chakraborti, A.; Heininen, A.; Väänänen, S.; Koskinen, K.T.; Vainio, H. Evidential Reasoning based Digital Twins for Performance Optimization of Complex Systems. *Procedia CIRP* **2021**, *104*, 618–623. [CrossRef]

39. Scanagatta, M.; Salmerón, A.; Stella, F. A survey on Bayesian network structure learning from data. *Prog. Artif. Intell.* **2019**, *8*, 425–439. [CrossRef]
40. Ramsey, J.; Glymour, M.; Sanchez-Romero, R.; Glymour, C. A million variables and more: The Fast Greedy Equivalence Search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *Int. J. Data Sci. Anal.* **2017**, *3*, 121–129. [CrossRef]
41. von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416. [CrossRef]
42. Kang, Z.; Shi, G.; Huang, S.; Chen, W.; Pu, X.; Zhou, J.T.; Xu, Z. Multi-graph fusion for multi-view spectral clustering. *Knowl.-Based Syst.* **2020**, *189*, 105102. [CrossRef]
43. Yoon, S.Y.; Lin, Z.; Allaire, P.E. *Control of Surge in Centrifugal Compressors by Active Magnetic Bearings*; Springer: London, UK, 2013. [CrossRef]
44. Giarré, L.; Bauso, D.; Falugi, P.; Bamieh, B. LPV model identification for gain scheduling control: An application to rotating stall and surge control problem. *Control Eng. Pract.* **2006**, *14*, 351–361. [CrossRef]
45. Sempler Library. Available online: https://sempler.readthedocs.io/en/latest/ (accessed on 5 September 2022).
46. GBMR. 2022. Available online: https://github.com/anandashankar/gbmr (accessed on 15 April 2023).
47. Grömping, U. Variable Importance Assessment in Regression: Linear Regression versus Random Forest. *Am. Stat.* **2009**, *63*, 308–319. [CrossRef]
48. Choi, K.; Fazekas, G.; Sandler, M.; Cho, K. Convolutional recurrent neural networks for music classification. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017. [CrossRef]
49. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
50. Menze, B.H.; Kelm, B.M.; Masuch, R.; Himmelreich, U.; Bachert, P.; Petrich, W.; Hamprecht, F.A. A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC Bioinform.* **2009**, *10*, 213. [CrossRef] [PubMed]
51. Altmann, A.; Toloşi, L.; Sander, O.; Lengauer, T. Permutation importance: A corrected feature importance measure. *Bioinformatics* **2010**, *26*, 1340–1347. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.