Ocean Engineering 223 (2021) 108629

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Subsystem selection for digital twin development: A case study on an unmanned underwater vehicle

Demetrious T. Kutzke [*], James B. Carter, Benjamin T. Hartman

Naval Surface Warfare Center Panama City Division (NSWC PCD), 110 Vernon Ave, Panama City, FL, 32407, USA

ARTICLE INFO

Keywords:
Digital twin
Multiobjective optimization
Pareto optimization
Reliability

ABSTRACT

Digital twins are virtual representations of subsystems within a system of systems. They can be utilized to model and predict performance and condition degradation throughout a system's life cycle. Condition based maintenance, or the performance of system maintenance based on the subsystem states, is often facilitated by the implementation of digital twins. An open challenge is selecting the subsystems that require digital twins. We establish a generic process for determining a set of priority-based system components requiring digital twin development for condition based maintenance purposes. The priority set, which we term the "triage" set, represents the set of components that when monitored through a digital twin lead to the greatest increase in total system reliability and simultaneously represent the minimal cost set of components for implementing a digital twin. While we focus our process on an unmanned underwater vehicle (UUV), where we frame the design problem as a multiobjective optimization problem utilizing experimentally determined data and metrics from the model of a real UUV system, the process is generic enough that it could be utilized by any system looking at cost and reliability estimates for leveraging digital twin technology.

1. Introduction

We explore the interplay between total system reliability and the development cost for building a digital twin for an SoS. Industries that depend on complex SoS for accomplishing hazardous or laborious tasks that would otherwise place humans in harm require reliable and robust SoS that function expectedly. Medical robotics, the airline and automotive industries, and smart manufacturing facilities are a few examples of SoS that perform tasks to mitigate the risk to the human. Often digital twins, a term first coined by Michael Grieves, facilitate this reliability and enhance the robustness by providing SoS operators and maintainers with additional system awareness (Grieves, 2005; Daneshmand et al., 2017; Kousi et al., 2019; Tao et al., 2019a). Digital twins are virtual representations that mimic the subsystems and dynamics of their real-time counterparts. Digital twins have proliferated in use because of their predictive capacity for the degradation of their real-time systems and are now a central part of product life cycle management (Grieves and Vickers, 2017). However, many legacy SoS still rely on traditional methods of reliability management to mitigate risks through a variety of maintenance strategies, including preventative, reactive, and condition based maintenance.

Preventative maintenance entails performing routine maintenance tasks on a fixed schedule (Swanson, 2001). This can lead to significant waste of labor, since component failures typically occur as a result of the condition of parts (Endrenyi et al., 2001). If the condition of the part to be replaced has not yet degraded to the point of failure, then the replacement part is wasted as is the labor to replace the part. Reactive maintenance means performing maintenance based on events, such as the failure or degradation of a mechanical part to the extent that it is non-functional (Karuppuswamy et al., 2006). While the typical maintenance schedule of an SoS includes a combination of both preventative and reactive maintenance, alternative methods are used throughout the airline, automotive, and other commercial industries that observe the "condition" of components to trigger maintenance tasks (Li and Nilkitsaranont, 2009; Prajapati et al., 2012; Zou et al., 2019). This condition based maintenance provides an opportunity to reduce labor and parts costs, since fewer parts are wasted and labor is optimized on an as-needed basis. Visual inspection can provide some degree of accuracy in assessing the condition of certain external components, for example the tread on a tire of an automobile (Horner et al., 1997). A newer approach offers more disciplined and less ambiguous condition assessments. This is condition based maintenance through the use of a digital twin. To minimize the need for visual inspection, digital twins are

[*] Corresponding author.
E-mail address: demetrious.kutzke1@navy.mil (D.T. Kutzke).

https://doi.org/10.1016/j.oceaneng.2021.108629
Received 29 October 2020; Received in revised form 7 January 2021; Accepted 10 January 2021
Available online 4 February 2021
0029-8018/Published by Elsevier Ltd.

Nomenclature

CBM	Condition based maintenance
DT	Digital twin
GANBI	Genetic-algorithm-based normal boundary intersection
MBSE	Model-based systems engineering
MOOP	Multiobjective optimization problem
SoS	System of systems
TDT	Triage digital twin
UUV	Unmanned underwater vehicle
$\bar{f}_{\min}$	Convex hull of individual minima for MOOP
$\delta_{R'R}$	Reliability improvement function
$\mathscr{C}$	Set of known design and development costs for digital twins of subsystems
$\mathscr{R}$	Set of known reliabilities of subsystem components
$\mathscr{R}'$	Known reliabilities for components with corresponding digital twin
C_{tot}	Total digital twin implementation cost
N_{gen}	Number of generations in evolutionary algorithm
N_{pop}	Size of population in evolutionary algorithm
N_{pts}	Number of points segmenting line connecting individual minima
L	Vector of component layer information
N	Total number of system components
r^*	Pareto optimal subset of component reliabilities
s^*	Triage digital twin components

utilized to monitor the condition of components in critical systems, like jet engines, fuel pumps, and robotics for example.

Digital twins have taken a leading role in advancing manufacturing and complex SoS throughout the Industry 4.0 movement and the proliferation of the Internet of Things (Kusiak, 2017; Fuller et al., 2020; Koulamas and Kalogeras, 2018). Given the widespread adaptation and integration of multiple data sources throughout a system's lifecycle, including sensor performance, testing and evaluation, and maintenance data for example, numerous technologies, among which the digital twin belongs, integrate and aggregate these data sources into a uniform picture of a system's state (Qi. and Tao, 2018). Tao et al. provide a detailed account and survey of digital twin technology, including numerous examples from industry that illuminate the utility of digital twins for predicting condition degradation within a SoS (Tao et al., 2019b). Notably, Tao et al. outline how General Electric (GE) has implemented digital twins for their wind turbine systems (Lund et al., 2018a, 2018b). The total digital twin system constitutes the sensor network by which the physical wind turbines are connected and the network connection to the cloud-based turbine model. The cloud models are constantly updated with performance data collected across the physical system network. The feedback from the cloud model can then be utilized in turn as an optimized controller for the wind turbines, creating an optimized feedback loop facilitated by the so-called smart connectivity. GE's digital twin model emphasizes a general standard that defines a digital twin system: There is a physical system, a virtual system, and connection between the virtual and physical system (Glaessgen and Stargel, 2012). Another interesting example is the development of a digital twin for a robotic surgery system (Laaki et al., 2019). While Laaki et al. utilize a robotic arm as a substitute for a real time system, they mention the da Vinci Surgical System, manufactured by Intuitive Surgical, Inc., as a suitable candidate that would benefit from the digital twin. Boeing has begun utilizing its in-house developed discrete event simulation model which incorporates material procurement data and manufacturing automation performance data. By using this data and the simulation tool, Boeing has an end-to-end supply chain model for its manufacturing businesses. Together these represent the digital twin (Hilton and Needham, 2019). Siemens, the German manufacturer, utilizes numerous digital twin models for its production processes. This underscores the key idea behind digital twins: There is not one digital twin that governs or represents an entire system; rather, a digital twin comprises numerous distinct and potentially different models based on the function of the individual subsystems. In Siemens' case, they utilize finite element models for stress and strain digital twins in their production line (Tao and Qi, 2019). For further examples taken from industry, see Liu et al. (2020).

For this paper, we choose the unmanned underwater vehicle for a case study, since there is a growing dependence on these systems from a variety of industries that directly remove humans from harm as a result of leveraging UUV technology. UUVs, or autonomous underwater vehicles (AUVs), have witnessed significant growth in their utility for a variety of industries, including sea-floor and oceanographic surveys, deep-water drilling platform inspections, ship hull inspections, bridge maintenance and surveys, and various academic uses (Hylton, 2020; Whitcomb, 2000; Fletcher, 2000; McFarlane, 2008; Fletcher and Wernli, 2003; Miller, 1996; Kukulya et al., 2010). The wide adaptation and continued use of UUVs indicates a high level of dependence on these platforms for critical tasking—tasking which would otherwise place humans in direct risk of harm (Thieme and Utne, 2017). This dependence largely requires reliable and available systems that can be expected to function according to specifications. Runtime mission failures present potentially catastrophic consequences for the operators, while presenting a challenging task to mitigate these risks for the UUV maintainers. A UUV presents a very challenging task for traditional maintenance strategies and CBM through visual inspection, since access to internal components is limited or requires significant effort to inspect. The internal components are housed inside either a free-flooded, the hull floods with water for buoyancy control, or a hermetically sealed hull, the hull is both air and water tight. To access the components housed inside the sealed hull requires breaking the seal and accessing the components in a confined space. External components can be inspected for wear and condition degradation, but these are also challenging since functionality can be degraded, without visual cues. These challenges can largely be subverted by the use of digital twins for CBM purposes.

The twin is a digital replica that maintains data and dynamics information in parallel with the real-time system. Building a twin typically involves selecting a data- or dynamics-driven model for every individual component (Liu et al., 2012; Coraddu et al., 2019; Kim et al., 2020). Data-driven models rely on historical system data for training machine learning models. Dynamics-driven models utilize first principles and physics-based modeling for creating models that are data independent. The particular methodology depends on the intended use. For example, a data-driven model which utilizes a machine learning process is particularly suited for predicting component degradation. Hong et al. demonstrate the use of one such technique, a special ordered map, which through unsupervised machine learning they train to extract degradation of ball bearings (Hong et al., 2014). Alternative methods have been used which provide performance estimations under unknown conditions. These dynamics-driven approaches offer higher degrees of predictability than their data driven counterparts, since they model time-based dynamics. These, too, update model parameters from the real-time system but do not depend on historical performance data for their predictive capacity. Hanachi et al. derive a physics-based model for capturing the performance of gas-turbine engines, with applications in the airline industry (Hanachi et al., 2015). Their work demonstrates one instance of the applicability of a model-based approach for predictive capacity. Others have developed novel techniques that give hybrid performance. Lei et al. use a model-based approach for remaining useful life (RUL) predictions, for which they provide experimental evidence demonstrating the method's efficacy on ball bearings. The challenge with leveraging data-driven models is the necessity for copious amounts of high quality data. For the airline and automotive industries, which typically have experimental apparatus in place, it is decidedly easier to produce the data for predictive maintenance models. However, for systems like UUVs, the presence of high quality data is generally lacking, since it is challenging to extract failure mode data from these systems, when failures occur at run-time and below the sea surface. There are two primary mechanisms for handling this case: machine learning algorithms that handle limited training data and better experimental apparatus.

Alternative machine learning models exist to leverage digital twin technology without the use of high-quality or properly labeled data. The work of Tan et al. demonstrates the use of one such technique to predict condition degradation of a ship's propulsion system without properly labeled data sets (Tan et al., 2019, 2020). Alternative test setups can be fashioned to run repeated experimental tests to extract relevant performance data under typical operational mission profiles. See Fig. 1 for an example experimental setup. In the figure, an oscilloscope is connected to the UUV, while the UUV's autonomy computer is running in order to collect sensory signal data on the workbench. User interface components are connected directly to the autonomy computer to manipulate simulation characteristics on the bench. The UUV shown in the figure is the system we consider for the case study in this paper. Section 3 provides a detailed overview of the system. Even these setups lack in realistic simulation since external failures can occur from the UUV hitting something in the water or during docking and deploying the UUV from a ship. The repeated insertion and extraction (deploying and docking) of the UUV can create significant wear on the system, because many of the external components are very sensitive, such as the fins and propeller. To accurately represent these mechanical failures is challenging in a benchtop environment. The apparatus shown in Fig. 1 is only capable of testing the electronic components and as such falls short at capturing the entirety of the potential failure modes.

Here we are not considering the particular model type, only that *some* model will be selected to serve as the twin. We do make the distinction that the digital twin for an entire system comprises any number of individual digital twins for the subsystems being monitored. Each subsystem has its own model requirements (data- or dynamics-driven) and has its own requirements for data acquisition and physical interfaces. Data acquisition establishes the means by which the twin acquires data from the real-time subsystem. This could be through client-server style resources, as in the Robot Operating System (ROS), or exchanging data through some other means, via Ethernet, for example (Stanford Artificial Intelligence Laboratory et al.). Physical interfaces indicate the required ports to facilitate the data acquisition process. These could be any physical connection such as Universal Serial Bus (USB), Ethernet, or other serial port connections.

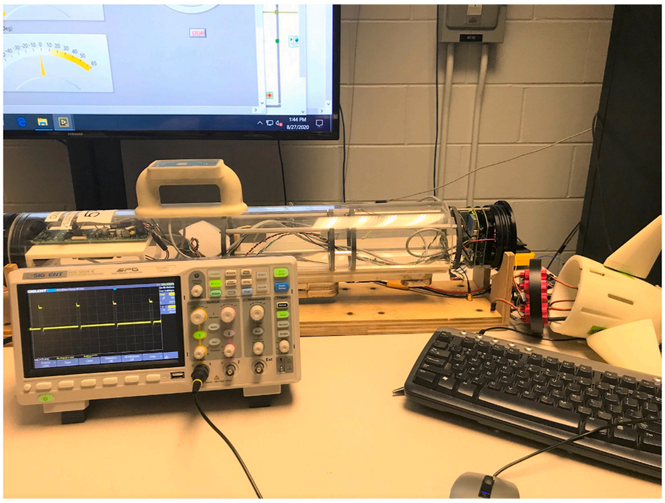


Fig. 1. (Color online) The UUV system under test with experimental apparatus. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

There are two pathways to leveraging digital twin technology for CBM purposes: First, the system designer can build a twin for each component, incorporating the necessary data and physical interfaces into the system design. Second, the system can be retrofitted with the information acquisition resources (data and physical interfaces) to extract the relevant information to support the twin. The former requires a significant amount of design effort, since the system designer must meet the standard system specifications and requirements, as well as incorporate the acquisition resources for the digital twin for the relevant components. The second pathway provides a means for retrofitting the system with digital twins. It also provides attachability, meaning legacy systems that are critical for removing humans from direct harm can still benefit from digital twin technology. The second also has the advantage of an established (perhaps inefficient) maintenance schedule that can be optimized from the incorporation of digital twins for CBM purposes. However, the questions unanswered from the perspective of decision makers and developers alike remain.

- What components should have digital twins?
- How should the system be retrofitted with the data acquisition resources for the digital twins?

We address the cost issue associated with incorporating a digital twin by developing a novel approach that incorporates both MBSE and multiobjective optimization to enable the selection of appropriate components for digital twin representation. These "triage" components produce the largest increase in total system reliability and minimize the cost-entrance hurdle for leveraging digital twin technology. We utilize the term triage to reflect the fact that some components are prioritized over others, much like the triage process utilized during disaster relief by first responders. Prioritization of limited medical treatment resources optimizes the care given, ensuring that those who qualify and meet the selection criteria receive treatment first (Koenig and Schultz, 2010). Though the criteria change based on the triage system used, the idea remains the same. Those who benefit most from treatment receive priority over those who would benefit the least, such as someone with significant blood loss over someone with a minor cut. Likewise, certain subsystems yield larger and more significant benefits to total system reliability than other subsystems.

The paper is organized as follows: In Section 2 we explore established research for reliability predictions, redundancy allocation, and reliability optimization. In Section 3 we introduce the platform used for modeling a minimalist UUV system, as well as a basic description of the vehicle and additional reference data. In Section 4 we introduce our MBSE approach, including the language and modeling used to develop the model for the UUV system, as well as challenges in implementing MBSE retroactively. In Section 5 we explain in detail the reliability analysis and demonstrate results from applying our process to the UUV under test. We conclude the case study in Section 6 by commenting on the implications of the case study for the field of UUVs and how the work can be used to inform the TDT design process. We address the implications on the work for the field of reliability studies and how the process generalizes for arbitrary SoS looking at implementing a TDT.

2. Related work

Much of the existing literature in the fields of redundancy allocation and reliability predictions offer technically rigorous mechanisms for improving SoS reliabilities through the introduction of redundant components and sophisticated methods for predicting and optimizing SoS reliability. As a result, these methods provide a breadth of established research that can be used for a variety of SoS. Our contribution is a coupling of an MBSE approach and an MOOP, providing a simplified process that incorporates data from a variety of stakeholders, system maintainers and operators, to guide the reliability improvement analysis, and frame the discussion with decision makers by providing candidate subsets for TDT implementation. Our specific contributions are as follows:

- A novel methodology to prioritize and select a triage set of subsystems that comprise the digital twin of the total system
- An MBSE approach that elucidates the necessary data acquisition and physical interfaces that must be established for future digital twin development
- A reliability analysis based on the application of our methodology to a case study on a UUV
- Qualitative analysis of the implications of our methodology on the development of next-generation unmanned systems on the TDT design process

We will now discuss the state of the art in digital twin development and reliability improvements for SoS, and how they are augmented by our methodology. Biggs et al. introduce the contemporary challenges in utilizing SysML as the primary MBSE language to include reliability considerations in an MBSE effort, early in a project's life cycle (Biggs et al., 2018). They advocate for the inclusion of new language concepts that extend SysML to better model reliability requirements than are currently possible. At the time of this authorship, we are not aware of the inclusion of these concepts in the SysML standard. Their work is notable because they address several issues that we have encountered, including the tracking of reliability and safety data across different documents and spreadsheets. These are challenging to maintain and often the critical information is lost. To mitigate this challenge, we capture reliability data directly in the MBSE model. As we conduct discussions with the system maintainers and architects, this information is tracked and can be updated as the model is updated throughout its life cycle. Additional techniques are utilizing the central structure that MBSE provides for conducting failure modes and effects analysis (FMEA) (Huang et al., 2017). Our choice in using MBSE for modeling complex SoS is supported by the use and acceptance of several organizations (D'Ambrosio and Soremekun, 2017; Mitchell, 2010). Given the complexity of many SoS, MBSE is gaining traction across organizations that would like to simplify system life cycle maintenance, since it unifies many traditionally disparate data sources and documents into one accessible and traceable location.

Several strategies are used to improve system reliability. One such strategy is the introduction of redundant components. By introducing redundant components, or backups, the failure of the component is mitigated by these alternates. Indeed this is an integral part of robust system design and the topic of numerous papers on designing for reliability (Rochlin et al., 1987; Kim and Kim, 2017; Muhuri et al., 2018). An example is the electronic flight data recorder or black box used in commercial airplanes. The electronic flight data recorder tracks and records flight data and system diagnostics. It is recovered after airplane crashes and used to provide investigators with vital system information leading to the failure and crash. The data recorder is often installed with a redundant data recorder to improve the chances of finding the device in a crash and to improve the reliability (Downer, 2011). A challenging problem is determining how many redundant components should be used to achieve a desired level of reliability. This is the redundancy allocation problem for parallel subsystems. Coit and Smith introduce a genetic algorithm for the redundancy allocation problem in parallel subsystems (Coit and Smith, 1996). The genetic algorithm of Coit and Smith scales well for large systems and offers a unique formulation for determining redundancy levels and searching the component trade-space. As a consequence, they can quickly identify the level of redundancy required to achieve a stated level of reliability within cost constraints. Our work seeks a similar calculation, except that we are not considering parallel subsystems exclusively, and we do not have reliability requirements. Muhuri et al. provide a detailed approach to the redundancy allocation problem. They go beyond the work of Coit and Smith by accommodating traditionally challenging trade space parameters such as physical dimensions and weight of redundant components through fuzzy parameters in their optimization routine. The challenge with such an approach is utilizing this retroactively. While this is an excellent approach when utilized early in the design phase of a system, existing components are largely fixed and the system has been optimized to accommodate those components. Fundamentally, the reliability can only be changed through a more intelligent maintenance routine such as a CBM approach. Yeh and Fiondella expound in great detail on a redundancy allocation application in computer networks (Yeh and Fiondella, 2017). They utilize a simulated annealing algorithm to solve the allocation problem. Garg and Sharma perform a strikingly similar analysis as our work. They utilize a particle swarm optimization algorithm to solve their MOOP, which is a bi-objective problem maximizing reliability improvement from the introduction of redundant components. They also minimize system design cost (Garg and Sharma, 2013). We also optimize for a reliability difference, which is the difference between the base system and the new proposed system with digital twins for CBM purposes. The work is distinct in that they minimize the cost for adding the redundant components, whereas our work considers selecting candidate subsets that minimize TDT implementation costs.

There have been significant advances in reliability predictions for complex systems in recent years. As of 2018, Yuan et al. introduced a statistical mechanism for reliability evaluation (Yuan et al., 2018). Using Bayesian networks (BN) they construct object-oriented representations of subsystems which they utilize to construct a BN for propagating uncertainties about component conditions. These dynamic object-oriented bayesian networks (DOOBNs) facilitate ease of modeling since the connections between subsystems are concise and built from conversations between the modelers and the domain experts. Kumar et al. show using fuzzy set logic a method for computing reliability of series parallel systems (Kumar et al., 2020). In their survey, Pérez-Rosés outlines numerous other mechanisms that exist for reliability optimization and redundancy allocations on complex systems and over series parallel networks (Pérez-Rosés, 2018).

3. Unmanned underwater vehicle for case study

In Fig. 2 we show the low-cost vehicle with subsystem labels and components. We have chosen this vehicle as our case study system for two reasons: First, we have direct access to design specifications, bill of materials, and the system architects. Second, the system provides a baseline representation of a UUV. Without the added components and complexity of legacy commercial off-the-shelf systems, the experimental vehicle offers the minimal set of functioning components. As a result, the systems engineering process is simplified to complete retroactively. We acknowledge that for a more complex system this process would not be as straightforward, particularly since access to the original architects or maintainers may be severely limited. We argue those who perform routine maintenance can provide insight into the design and the components that are the most challenging to maintain. The system operators are also an excellent resource, since they have a better idea of how the system functions under normal or expected operating conditions, which helps inform the triage component selection process. The output of the process in this paper guarantees subsets of components maximizing reliability and minimizing digital twin development cost, but it does not say whether these components are the most challenging to maintain. Cost does not necessarily imply difficulty of maintenance. If the

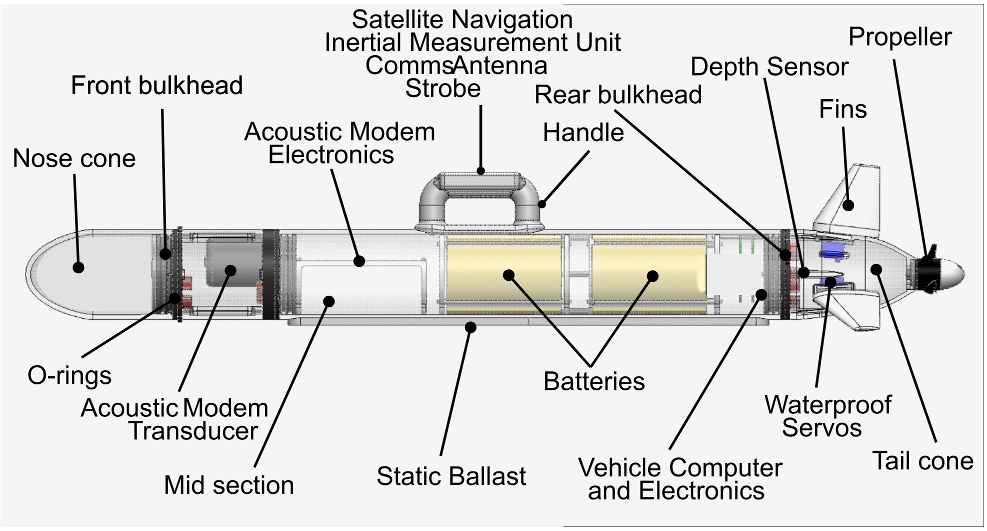


Fig. 2. (Color online) The UUV system under test with detailed subsystems. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

component is challenging to reach, such as an electronics component in the interior of the hull, but is low cost like a resistor, this would be a component that is hard to maintain but low unit cost. While we concede that there is generally a labor cost associated with performing maintenance, here we do not consider this. We do not address maintainability directly in this work, but it should be considered during the discussion phase with decision makers. Future work will address maintainability in a more rigorous fashion by including a maintainability metric in the objective set. In Section 6, we discuss the use of this and other higher-level information which can be used to scale the optimal subsets of components provided by the MOOP in our process.

The vehicle comprises 17 main subsystems. Here we do not make a distinction between a subsystem and a component. For a minimalist system like the vehicle, these can be used interchangeably. The components listed in Fig. 2 represent subsystems supporting communication between the vehicle and the operators, navigation, and intelligence. The communication subsystems are the acoustic modem transducer, acoustic modem electronics, and comms antenna contained in the handle. The navigation components are the passive sonar array, inertial measurement unit, satellite navigation, velocity sensor, and depth sensor. The intelligence subsystems support the higher level vehicle control, including path planning, ascending, descending, and waypoint navigation. The primary intelligence is provided by the vehicle computer and electronics. All of these subsystems depend on the energy source provided by the batteries. The components are assembled in the external housing, which is composed of the nose cone, front bulkhead, mid section, rear bulkhead, O-rings, and tail cone. The propeller and fins provide propulsion and control, respectively. The only subsystem we refer to that is not shown in Fig. 2 is the propeller shroud, which is concentric with the propeller, and shields the propeller from any damage. Together, these subsystems constitute a minimalist UUV and provide the ideal experimental apparatus to test our decision process on selecting triage component sets for implementing digital twins. However, to understand the process for determining the triage components, we must first understand the connections of the subsystems. We utilize an MBSE process to facilitate this understanding.

4. Model-based systems engineering approach

Ideally, MBSE models associated with physical systems are designed in the early stages of a project life cycle. This method captures full and accurate physical system information, capabilities, and design decisions in the model as the project evolves. It also permits MBSE modeling products to influence the evolving project. Retrofitting an MBSE model to a system that has already been designed and manufactured can be challenging. One of the challenges is collecting accurate data about the system requirements and physical components from documented resources. Those documents may be out of date or located in various locations that may be difficult to access. Another challenge is attempting to develop an understanding of how and why the system's components interact with each other based on analyzing the final design. To counteract these challenges, we initiated routine meetings with the vehicle engineers to collect the most current system information to help develop a vehicle CBM-based MBSE model. These discussions are an integral part in the overall process demonstrated in Fig. 3. Indeed, they permit the collection and organization of relevant reliability and cost estimating data required for implementing the triage component selection algorithms.

Regardless of whether an MBSE model is generated at the start or the finish of its associated physical system's creation, the design of an MBSE model mainly depends on the nature of the project, what the project is trying to achieve, what are the expected outputs of the project, and the resources available to the project during its life cycle. Another key aspect of an MBSE methodology is ensuring that the method provides a framework for specifying both the structure and operation of the project system and for analyzing its performance. Additionally, the steps taken in designing an MBSE model are often written in sequence, but many of them are executed in parallel. The same is true for the MBSE methodology of this project. The MBSE method chosen for the vehicle digital twin project is a modified version of the methodology described throughout Friedenthal et al.'s well-known text on MBSE in SysML (Friedenthal et al., 2015). The first step in retroactively applying an MBSE methodology is the identification of the core components of the system. As a result, the core components also constitute the core components of a digital twin, if one is implemented. Enterprise Architect (E. A.) is the MBSE software tool utilized to develop the model (Sparx Systems). We chose E.A. after a detailed trade study on the utility of various SysML implementations. We selected E.A. after determining that it aligns with industry use and supports the infrastructure for MBSE model life-cycle maintenance. An additional feature that assists the automation of our proposed process is E.A.'s support for the XML Metadata Interchange (XMI) format. This allows us to exchange the model with different MBSE software tools. It also allows us to easily parse the XMI formatted output for connectivity of the subsystems, which for a

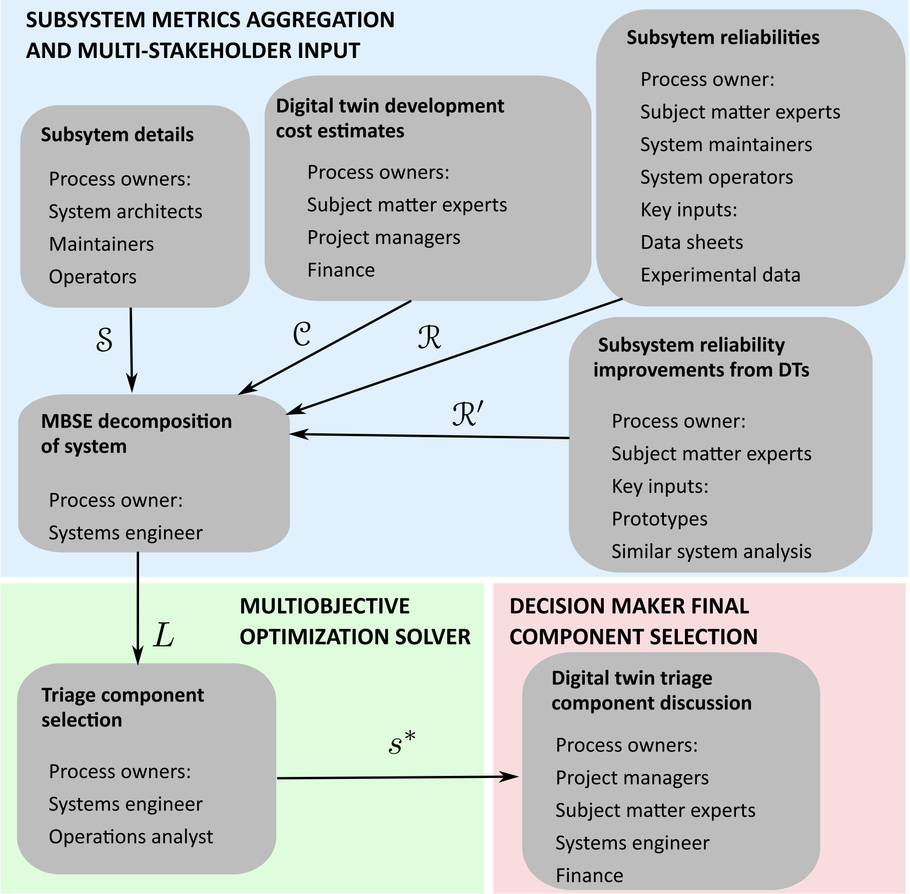


Fig. 3. (Color online) Process flow for determining the triage components for implementation of digital twins. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

very large system proves formidable if undertaken by hand.

The model's package structure defines the organization hierarchy. In Fig. 4 we show this hierarchy. The packages in the model are organized based on the artifacts and elements they contain. Since we are informing the development of a digital twin for a UUV, our first step is to develop the requirements for the digital twin. These requirements closely track those of the actual system, since the twin is a virtual representation of the vehicle real-time system. According to Friedenthal et al., the initial phase of establishing requirements is implemented by generating requirements from a project's mission statement or original "problem" concept that the project is trying to solve or answer. The modeling team facilitated the following efforts to obtain the most relevant current requirements: communicated with the vehicle's initial designers and stakeholders to retrieve initial project requirements, met with the vehicle engineering team to refine the requirements to their current status, and generated requirement elements in the vehicle digital twin MBSE model in requirements diagrams.

Table 1 shows the requirements gathered retroactively. These are shown here as tabular data but are extracted directly from the MBSE requirements diagram. We do not show the diagram for the sake of clarity. By including a requirements diagram in the MBSE model, we can trace the functionality of the subsystems to the original project requirements and also establish a working set of initial requirements for future DT development. The DTs constructed will have requirements that closely track those of the original subsystems, since for CBM purposes, any subsystem that stops functioning as expected will manifest the failure in the requirement. An example is a depth sensor, shown in Fig. 5. The depth sensor participates in localization and waypoint navigation. If the depth sensor malfunctions, then requirements 6 and 11 are no longer met. Consequently, the sensor will no longer deliver depth data to the main vehicle computer, which is responsible for higher level planning and waypoint navigation. This depth data, shown as a red dotted item flow in Fig. 5, defines an interface that would be monitored through the data-driven DT, updating the DT's learning model based on expected and actual functionality.

The structural vehicle block elements in the MBSE model represent the physical components of the vehicle. We show the entire system composition from the MBSE process in Fig. 5. Block Definition Diagrams (BDDs) represent the physical components of the system. Next, the digital twin CBM physical components are specified in the model. This step involved two main phases: generating block element representations of the physical components based on vehicle design documentation and adding data and energy flow connectors between appropriate components. For both phases, the UUV Digital Twin MBSE modeling team communicated with the vehicle engineers to verify and update designated CBM components and connections.

According to Friedenthal et al., a key effort in designing systems involves performing various engineering analyses such as trade studies, sensitivity analysis, and design optimization (Friedenthal et al., 2015). It may include the analysis of performance, reliability, cost, and physical properties of the project system. MBSE modeling supports these types of analyses via SysML parametric modeling. Parametric models constrain the properties of a system, which can then be evaluated using available analysis tools. Constraints are expressed as equations, with parameters of the equations being bound to the properties of the project's system. A parametric diagram within an MBSE model can be used to represent one

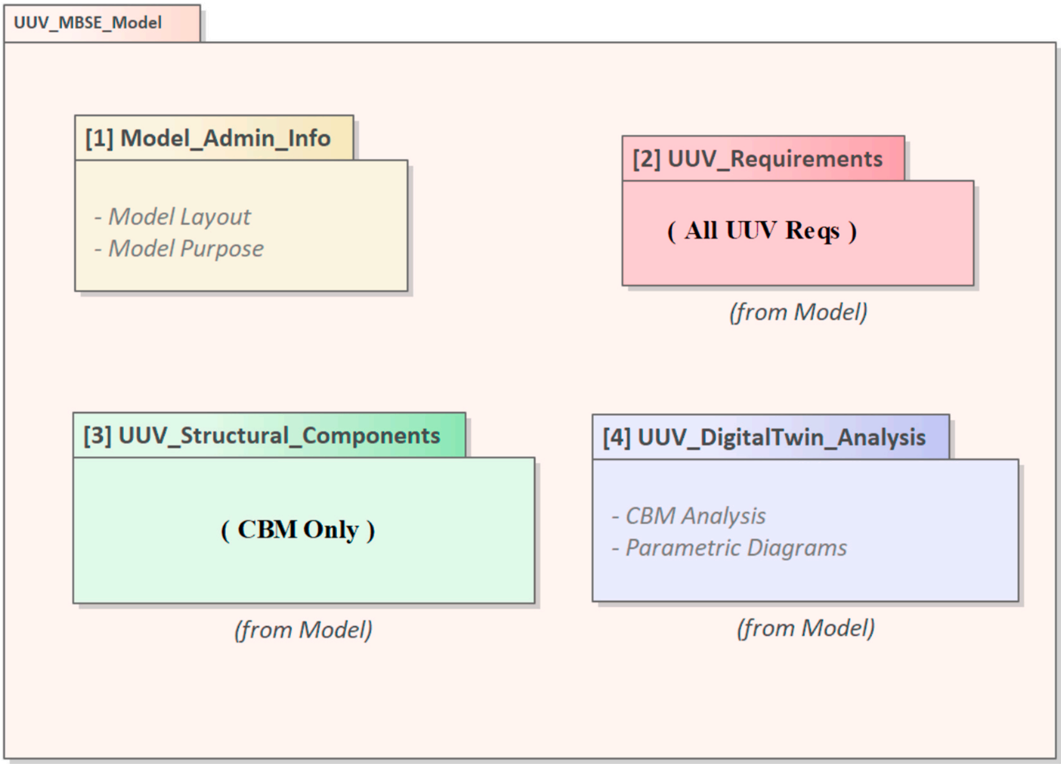


Fig. 4. (Color online) Model-based systems engineering structure of the UUV under test system model in Systems Modeling Language. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Table 1
Generic requirements taken from the model-based systems engineering requirements diagram for the UUV.

Requirement number	Requirement text
1	The UUV shall have modifiable behavior.
2	The UUV shall be able to communicate subsurface with either acoustic communication or radio frequency.
3	The UUV shall be able to communicate on the surface with either acoustic communication or radio frequency.
4	The UUV shall have low cost reconfigurability.
5	The UUV shall have a modular payload design.
6	The UUV shall be capable of waypoint navigation.
7	The UUV shall be able to operate in very shallow water.
8	The UUV shall provide sensor payload options for end-users.
9	The UUV shall be able to communicate subsurface.
10	The UUV shall be able to communicate on the surface.
11	The UUV shall be capable of subsurface waypoint navigation and GPS fixes.
12	The UUV shall have open and modular data interfaces.
13	The UUV shall be capable of specific runtimes.
14	The UUV shall have a specific unit cost.
15	The UUV shall have a specific range.

or more engineering analyses of a system's design. To incorporate this capability, we performed the following: designed Internal Block Definition (IBDs) diagrams for vehicle components in the MBSE model, integrated parametric diagrams to support the IBDs and generate SysML relational connectors between them and the IBDs using ports, implemented constraint equations to govern CBM design of the digital twin version of the vehicle, and executed computational evaluations of the CBM design. We incorporated parametrics as an auxiliary task to facilitate future reliability computations that can be taken directly from the MBSE model.

4.1. Reliability parametric analysis

As a part of the MBSE effort, we have included parametrics in the vehicle model. These parameters constrain the values of model parameters, permitting the use of simulation tools to extract relevant model information. We utilize parametric analysis to produce a baseline value of the total system reliability, to which we compare our extracted reliability taken from connectivity assumptions. We utilize the National Institute of Standards (NIST) definition of *reliability*, which defines reliability as "The ability of a system or component to function under stated conditions for a specified period of time" (Ross et al., 2019). In the case of a UUV system, we apply this definition and define reliability as the ability for the vehicle to operate according to mission requirements for the duration of the mission's operational time. The specific tasks enumerated under mission requirements include basic navigational capabilities, such as navigating between waypoints, ascending and descending, and performing routine surface behaviors to establish communications. To capture the reliability of the vehicle, we utilize input from the system architects and data sheets about estimated subsystem reliabilities. Unfortunately, since the development and expanded use of the vehicle is in its nascency, we do not have experimental data from operational missions which is an alternative source of reliability determinations. Utilizing these values and analyzing subsystem connections we can calculate total system reliability utilizing the well known reliability functions defined by NIST (Tobias, 2013). Because we are modeling reliability in the normal operating period or Intrinsic Failure Period of the Bathtub Curve, we assume constant failure rates and an exponential distribution of subsystem reliabilities. The equation for component i's reliability R_i is given as

$$\mathrm{R}_i(t) = \mathrm{e}^{-\lambda_i t}, \tag{1}$$

where t is the time in hours of operation, and λ_i is the failure rate of

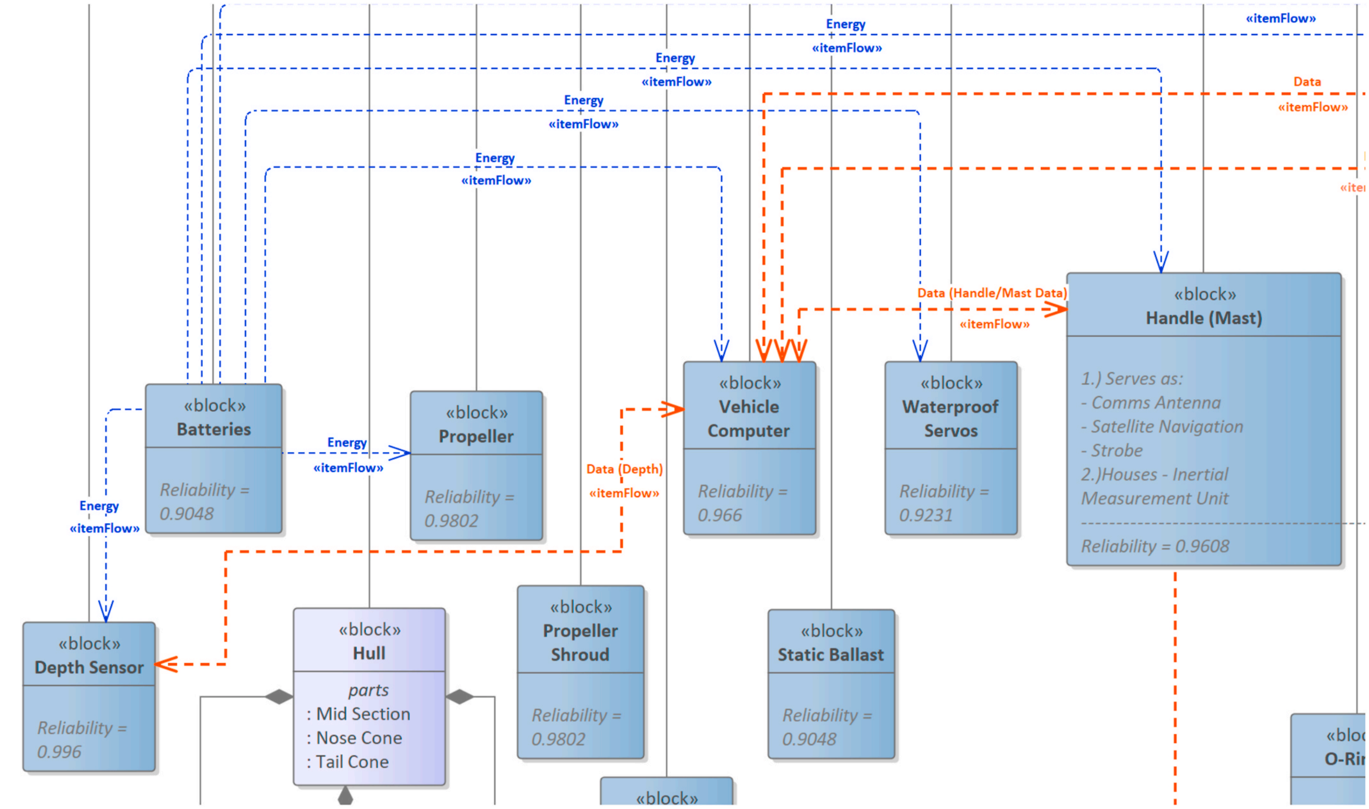


Fig. 5. (Color online) Subset of model-based systems engineering structural components as block definition diagrams. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

component *i*. The explicit dependence on time indicates that the reliability of the system is not fixed. However, for the purposes of our work, we assume that all reliability estimates are calculated with respect to a constant operation completion time t_{op}. To make the conversation more informative, we cast (1) in terms of the number of missions *n*. Rewriting (1) gives

$$\mathrm{R}_i(n) = \mathrm{e}^{-\lambda_i t_{\text{op}} n}. \tag{2}$$

We then estimate the mean time between failures (MTBF) in units of number of missions, to arrive at reliability estimates for operational mission duration. This makes the conversation with operators, who aren’t necessarily familiar with the exact numbers of failure rates and reliabilities, more informative by means of placing failures in context. An example would be, “After how many missions do you (the operator) need to realign or adjust the waterproof servos on the vehicle?” If the answer is every 25 missions, then if $t_{\text{op}} = 2$ hr, the failure rate is $1/50\ \text{hr}^{-1}$, yielding a reliability of

$$\mathrm{R}\left(t_{\text{op}} = 2\ \text{hr}\right) = \mathrm{e}^{-\frac{1}{50\ \text{hr}} 2\ \text{hr}} \approx 0.9231. \tag{3}$$

This simple calculation is a helpful approximation for complex analysis. By casting the reliability in units of number of missions, the operators and maintainers can inform the discussion much more easily, since the burden of the calculation falls on the operations analyst or system analyst.

The overall reliability of a system depends on the network connections. For *N* components attached in series, then the reliability of the subsystem is

$$\mathrm{R}_{\text{series}}(t) = \prod_i^N \mathrm{R}_i(t). \tag{4}$$

The reliability of the subsystem of components attached in parallel is

$$\mathrm{R}_{\text{parallel}}(t) = 1 - \prod_i^N (1 - \mathrm{R}_i(t)). \tag{5}$$

The equations assume that the subsystems are statistically independent. Parallel subsystems typically refers to the same subsystem and indicates redundancy, where the failure of the subsystem is mitigated by the redundancy. Series subsystems define the connection between two different subsystems that are connected via single or multiple-data connections. The functionality of the different components results in the processing or utilization of the data for different purposes. One can think of these connections as an analogy to parallel and series resistors from electronics circuits. A series connections permits a current flow along a single path, whereas a parallel connection permits the division of current along multiple paths. Whereas in circuit analysis these connections result from a transmitted current through physical connections, the connections of subsystems can be quite different and include any connection and the transfer of any data type. In the next section, we introduce how we define layered connectivity and how we automatically extract these connections from the output of the MBSE process.

4.2. Extraction of network connectivity

A natural result of performing an MBSE effort retroactively is the determination of how subsystems are connected to each other. Necessarily, connections can be abstract, for example the passage of energy, between a battery and a dependent electronic device. See Fig. 5 for an example of energy (shown as a blue dotted line) passing between the battery and multiple subsystems. However abstract, if the model is detailed enough to capture these connections, then utilizing the output of the MBSE we can extract the total system reliability, which we use as a

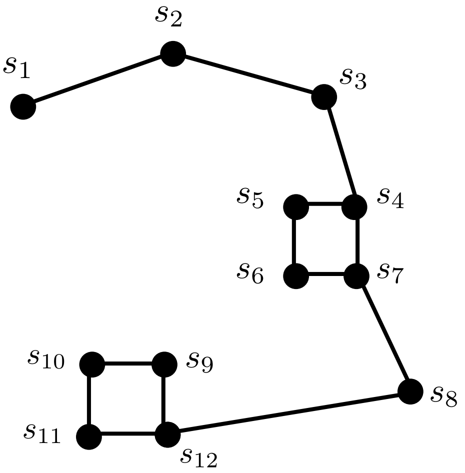


Fig. 6. (Color online) Layered network topology.

primary input in the next section (Section 5) for performing our optimization of subsystem selection. We assume these connections establish the basis for calculating the total reliability as a product over (4) and (5). To extract those connections from the model, we assume a layered network topology. Each layer l_i defines how many subsystems are redundant within the layer. Within a redundant layer, we use (5) to calculate the reliability. Individual layers are assumed to be connected in series. This permits fast iteration over the network subsystems and allows us to construct a one dimensional representation of the network. We show the layers in Fig. 6. Notice that this permits an arbitrarily complex system to be captured in the one-dimensional data structure.

In Fig. 6, subsystems are represented by black dots and labeled by s_i. Components $s_4 \ldots s_7$ and $s_9 \ldots s_{12}$ are in parallel and are redundant subsystems. The remaining components are attached in series with the parallel subsystems. From this graphical representation, which closely mimics the representation one might see within a SysML model, we can extract the layer vector L. The layer vector is constructed from the cardinality of the subsets attached in series. Because $s_{4\ldots7}$ and $s_{9\ldots12}$ are attached in parallel, then the L_4 and L_6 vector elements equal 4. Written out,

$$L = [1, 1, 1, 4, 1, 4], \tag{6}$$

where the number indicates the total number of redundant subsystems. The utility of this layer data structure will become apparent when we calculate reliability improvement expectations.

5. Reliability improvement expectations

We have argued thus far that one can utilize the output of the MBSE process to represent the underlying subsystem connections as a one dimensional data structure L. This in turn can be used to calculate the reliability of the system as a whole, since it is a discrete representation of the total system reliability. To show how L can be used to calculate the expected reliability improvements as a result of utilizing a digital twin to monitor subsystems, we introduce an MOOP to determine expected reliabilities and cost estimates for implementing digital twins for the UUV subsystems. The main objectives we consider are the reliability difference $\delta_{\mathrm{R'R}}$, which is the difference in reliabilities between the base system and the resulting system after implementing digital twins, and the total cost C_{tot}, which is the cost of implementing all digital twins within the triage subset.

Problem 1. (Multi-objective optimization problem statement).

For a system comprising N subsystems $\mathscr{S} = \{s_1, s_2, \cdots, s_N\}$ with reliabilities $\mathscr{R} = \{r_1, r_2, \cdots, r_N\}$, digital twin development costs $\mathscr{C} = \{c_1, c_2, \cdots, c_N\}$, and known reliabilities for components with corresponding digital twins $\mathscr{R}' = \{r'_1, r'_2, \cdots, r'_N\}$, find binary indicator vectors x_r and x_c that maximize the reliability difference $\delta_{R'R} := (R' - R)$ and minimize total development cost C_{tot} subject to the constraints that the total number of nonzero elements in both x_r and x_c does not exceed N. The sets which satisfy both objective functions are the Pareto optimal sets. These provide the corresponding TDT set of components s^*.

Stated as a bi-objective optimization problem

$$\underset{x_r \in \mathscr{X}}{\text{minimize}} \quad \delta_{\mathrm{R'R}} = -(\mathrm{R}'(x_r) - \mathrm{R}) \tag{7}$$

$$\underset{x_c \in \mathscr{X}}{\text{minimize}} \quad \mathrm{C}_{\mathrm{tot}} = x_c^{\mathrm{T}} c \tag{8}$$

$$\text{subject to } \mathscr{X} = \{0, 1\}^N \tag{9}$$

$$x_r^{\mathrm{T}} x_r < N \tag{10}$$

$$x_c^{\mathrm{T}} x_c < N. \tag{11}$$

Note that to facilitate ease of computation we have utilized the duality principle in optimization to convert the maximization of reliability improvement function to a minimization by multiplying by -1. The objective function given by (7) represents the difference in reliability improvements, where the decision variable is given by x_r. The function R' yields the system reliability as a result of switching out the candidate set of components x_r^{T}. The value R represents the total system reliability before improvements. This is calculated either from the MBSE model, using the parametric diagrams, or from the connectivity vector L. In theory, (7) should be negative for every subset x_r, assuming that the digital twins are designed properly and function no worse than the real-time system. This is a reasonable assumption since the digital twins are engineered by a team of experts, typically those with data science or machine learning backgrounds and those with hardware and software expertise in the component area. The variable c is a vector equivalent of the set of implementation costs $\mathscr{C}$.

5.1. Numerical approximation methods

We compare two optimization methods. These are chosen partially out of convenience and for illustrative purposes. We emphasize that the methods introduced here are not novel or in anyway prescriptive. Any method that solves Problem 1 suffices for producing the TDT subsets. The first is a direct computation of the subsets, since the number of subsystem components is reasonably small that employing a direct method of computation is feasible, similar to the method employed by Custódio et al. (2011). This direct calculation iterates over the entire component space, which for a practical standpoint for a UUV is not unreasonable. However, if the number of components grows sufficiently large ($N \approx 20$) then more efficient methods must be considered, such as Gray code methods (Savage, 1997) or evolutionary algorithms. The direct computation method is shown in the procedure BRUTEFORCEOPTIMIZATION (). In Algorithm 1, we show this method and its auxiliary procedures to calculate all candidate subsets of reliabilities and costs. Though we do not provide the pseudocode for the procedure ALLCOMBINATIONS (), it is a relatively simple combinatorial task to generate all possible combinations, since one can use the binomial coefficient $\binom{n}{k}$ as a starting point, producing up to $k = n - 1$ combinations to satisfy (10) and (11). Once all possible combinations of subsystem reliabilities and costs are generated, then we use OBJECTIVE to return the total system reliability as a result of utilizing digital twins R'. The OBJECTIVE () procedure closely follows (7), in that it utilizes the connectivity information, provided by L, to determine how subsystems and components should be treated as in series or parallel. Once all possible combinations of reliability improvements and costs are calculated, they are stored in δ and C_{tot}, respectively. The keen observer will note that this optimization routine has generated the entire feasible region of the problem. To identify only those non-dominated or Pareto solutions, we utilize an additional procedure GeneratePatetoFront () shown in Algorithm 3. The algorithm works by sorting in ascending order the C_{tot} and δ vectors. From these, we iterate in ascending order, recording at each step if the reliability difference $\delta(i)$ is less than at least one point already sampled. This produces the non-dominated point set.

The second method we use is the genetic-algorithm-based normal boundary intersection (GANBI) method. The GANBI method is particularly well-suited for engineering design problems, because unlike a standard evolutionary algorithm, which produces a solution set at each iteration, the GANBI algorithm produces a solution that approximates the Pareto front at each iteration (Wettergren, 2006). The GANBI algorithm serves as a preprocessor for existing genetic algorithms. We utilize the non-dominated sorting genetic algorithm (NSGA) as the baseline algorithm, for which we use the GANBI preprocessor. The interested reader should consult the work of Wettergren for a detailed account of the GANBI algorithm, including a formal algorithm description of the approximation step (Wettergren, 2006). Note that we are comparing the GANBI algorithm to the direct computation to show that for a small system with $N \approx 10$ components, the methods (direct and evolutionary) are equivalent. We also introduce the GANBI algorithm to allow an extension of this work for very large systems, with complex connection layers. Indeed we rely on this method for our real-world case study, which we compare to the well known NSGA. We show the pseudocode of the GANBI algorithm in GANBI() of Algorithm 4. The key distinction between the GANBI algorithm and traditional evolutionary approaches is shown at Line 2 in the procedure IndividualMaxima (). The GANBI algorithm relies on the calculation of the minima for the individual objective functions, without regard to the other objectives. If x_1 denotes the minimum of δ and x_2 denotes the minimum of C_{tot}, then

$$\mathbf{X}=[x_1,x_2], \tag{12}$$

is the vector that contains the minima for the reliability improvement and total cost objectives. Correspondingly, we can write the matrix $\bar{f}_{\text{min}}$ as

$$\bar{f}_{\text{min}}=\begin{bmatrix} 0 & \delta(x_2)-\delta(x_1) \\ C_{\text{tot}}(x_1)-C_{\text{tot}}(x_2) & 0 \end{bmatrix}. \tag{13}$$

This connects the individual minima through the line segment defined by the difference of points. The convex hull of individual minima (or "CHIM" from Wettergren's work) is given by convex combinations of (13). These are given by

$$\text{CHIM}:=\left\{\bar{f}_{\text{min}}B+X \ : B=[b_1,b_2]^{\text{T}},\right. \tag{14}$$

$$\left.\sum_i b_i=1, b_i\geq 0\right\} \tag{15}$$

The CHIM provides a coarse approximation to the Pareto front produced by the reliability difference and total cost objectives. The purpose of this is the creation of supporting objectives $\bar{f}$ given by NBIObjective (). The supporting objectives divide the hull into N_{pts}, which approximate the vector set normal to the points along the line segment connecting the two minima. Effectively this creates a point set that can be "pushed out" normal to the segment and eventually meet the Pareto front of the bi-objective problem. Creating the supporting objective functions actually increases computational time, but it provides a better approximation to the Pareto front than evolutionary approaches without the preprocessing. These modified objective functions are then passed into the traditional NSGA procedure for which we have declared the procedures but not the implementation of the code. These are shown as NextGen() and EvalPop() in Algorithm 4. We once again call GeneratePatetoFront () (Line 8) to do a final sort on the Pareto frontier.

Algorithm 1. Direct computation of reliability difference

```
Algorithm 1 Direct computation of reliability difference
procedure BruteForceOptimization($\mathcal{R}, \mathcal{R}', \mathcal{C}, L$)
    $N = |\mathcal{R}|$
    $\text{R} = $ Objective($\mathcal{R}, L$)
    $\mathbf{r} = $ AllCombinations($\mathcal{R}'$)
    $\mathbf{c} = $ AllCombinations($\mathcal{C}$)
    for $i = 1 \ldots |\mathbf{r}|$ do
        $r' \leftarrow$ Objective($\mathbf{r}(i), L$)
        $\delta(i) = r' - \text{R}$
        for $j = 1 \ldots |\mathbf{c}|$ do
            $C_{\text{tot}}(i) \leftarrow C_{\text{tot}}(i) + \mathbf{c}(i, j)$
        end for
    end for
    return GenerateParetoFront($\delta, C_{\text{tot}}$)
end procedure
```

Algorithm 2. Reliability from layer connectivity

```
Algorithm 2 Reliability from layer connectivity
procedure Objective($\mathcal{R}, L$)
    $\text{N}_{\text{layers}} = |L|$
    $\text{R} = 1$
    for $i = 1 \ldots \text{N}_{\text{layers}}$ do
        $\text{R}_{\text{sublayer}} \leftarrow \emptyset$
        for $l \in$ sublayer do
            $\text{R}_{\text{sublayer}} \leftarrow \text{R}_{\text{sublayer}} \cup \mathcal{R}(l)$
        end for
        if $|L(i)| > 1$ then
            $\text{R} \leftarrow \text{R} \cdot$ AtLeastOneSucceeds($\text{R}_{\text{sublayer}}$)
        end if
    end for
    return R
end procedure
procedure AtLeastOneSucceeds($\mathcal{R}$)
    $\text{R} = 1$
    for $r \in \mathcal{R}$ do
        $\text{R} \leftarrow \text{R} \cdot (1 - r)$
    end for
    return $1 - \text{R}$
end procedure
```

Algorithm 3. Producing Pareto front from objectives

```
Algorithm 3 Producing Pareto front from objectives
procedure GenerateParetoFront($\delta, C_{\text{tot}}$)
    $[\delta, C_{\text{tot}}] \leftarrow$ SortAscend($\delta, C_{\text{tot}}$)
    $\mathcal{R}^* \leftarrow \delta(1)$
    $\mathcal{C}^* \leftarrow C_{\text{tot}}(1)$
    for $i = 2 \ldots |C_{\text{tot}}|$ do
        if $\exists r \in \mathcal{R}^*$ s.t. $\delta(i) < r$ then
            $\mathcal{R}^* \leftarrow \mathcal{R}^* \cup \delta(i)$
            $\mathcal{C}^* \leftarrow \mathcal{C}^* \cup C_{\text{tot}}(i)$
        end if
    end for
    return $\delta^*, C^*$
end procedure
```

Algorithm 4. GANBI algorithm

```
Algorithm 4 GANBI algorithm
procedure GANBI($\mathcal{R}, \mathcal{R}', \mathcal{C}, L, \mathrm{N_{pts}}, \mathrm{N_{gen}}, \mathrm{N_{pop}}$)
    $\bar{f}_{\min}$ = IndividualMaxima($\mathcal{R}, \mathcal{R}', \mathcal{C}, L$)
    $\bar{f}$ = NBIObjective($\bar{f}_{\min}, \mathrm{N_{pts}}$)
    for $i = 1 \ldots \mathrm{N_{gen}}$ do
        pop $\leftarrow$ NextGen(pop, aux, $\mathrm{N_{pop}}$)
        $[\delta(i), C_{\mathrm{tot}}(i), \mathrm{aux}] \leftarrow$ EvalPop(pop, $\mathcal{R}, \mathcal{R}', \mathcal{C}, \bar{f}$)
    end for
    return GenerateParetoFront($\delta, C_{\mathrm{tot}}$)
end procedure
```

5.2. Approximation results for example system

Before we consider the vehicle in our case study, it is instructive to utilize our notional system of $N = 12$ components to demonstrate how the algorithms perform, understanding the results of the algorithms, and placing into context the results as they apply to selecting the TDTs and informing decision makers about the trade offs.

The first step is the enumeration of existing subsystem reliabilities $\mathscr{R}$, reliabilities after implementing digital twins $\mathscr{R}^{'}$, and the estimated cost to implement a digital twin $\mathscr{C}$. These are summarized in Table 2. To simulate a realistic scenario, we have created a randomized set of $\mathscr{R}$, where some components are noticeably "worse," or unreliable in comparison with others, as much as an order of magnitude worse. The upgraded reliabilities are also randomized. These, however, are randomized within $[r_i, 1.0]$, where $r_i \in \mathscr{R}$. This enforces the assumption that the reliability after implementing a digital twin can be no worse than the original. The implementation costs are also randomized within the range $[500, 5000]$, with arbitrary units. The connectivity of the subsystems is given by (6). Using these inputs to the BruteForceOptimization () we generate the Pareto front in Fig. 7a as well as show the entire feasible solution space. The black star markers indicate the Pareto solutions, and the black dots indicate a dominated feasible solution. The vertical axis represents the percentage increase in reliability as a result of implementing the TDTs. The abnormally large increases result from markedly improved reliabilities for the exceptionally unreliable subsystems. The total system reliability before upgrades is $\mathrm{R} = 0.029$. The vertical axis is the scaled version of the objective function (7) or

$$\frac{\delta_{\mathrm{R'R}}}{\mathrm{R}} \times 100. \tag{16}$$

We have also plotted on the same axes the Pareto front and the feasible solution space for the GANBI and NSGA methods, which are not required for this simplistic example of $N = 12$ components. The feasible space for the GANBI algorithm is shown by the blue crosses. The Pareto front for the GANBI algorithm is shown by the blue squares. The NSGA feasible space is shown by red diamonds, while the Pareto front is shown by the red squares. We have used $\mathrm{N_{gen}} = 500$ generations in both the GANBI and NSGA methods and a population size of $\mathrm{N_{pop}} = 50$ for the example system. After several numerical runs, we determined that neither larger populations nor more generations improved the Pareto frontier approximation. However, by plotting the GANBI algorithm it illustrates that we can approximate the Pareto front reasonably well. The benefit of using the evolutionary method comes when one considers complex systems with $N \approx 20$, where it is computationally challenging (if not impossible) to iterate over all possible combinations of subsystems. Therefore, the GANBI algorithm can be used for these larger complex systems. In Fig. 7a–c we have plotted the Pareto fronts from the direct, GANBI, and NSGA methods. In Fig. 7d we have also extracted two example subsets s_A and s_B from the Pareto front to show the trade off between reliability improvement and cost. Note that $s_A = \{1, 3, 8\}$ has a total cost of $C_{\mathrm{tot}} = 6691$ arb. unit, while $s_B = \{1, 2, 3, 4, 8, 9, 10, 11, 12\}$ has a total cost of $C_{\mathrm{tot}} = 26862$ arb. unit. As expected, the more components that are selected for digital twin monitoring the higher the reliability improvement and the higher the cost. As a result, a large part of the process is prioritizing reliability vs cost. For critical projects that remove humans from harm, no expense should be spared when considering the reliability improvement, whereas a project that does rely on the system for critical tasking but does not depend on the system for delivering humans from harm, should expect to have reduced development costs while achieving significant reliability improvements.

Table 2
Parameters for system with $N = 12$ components.

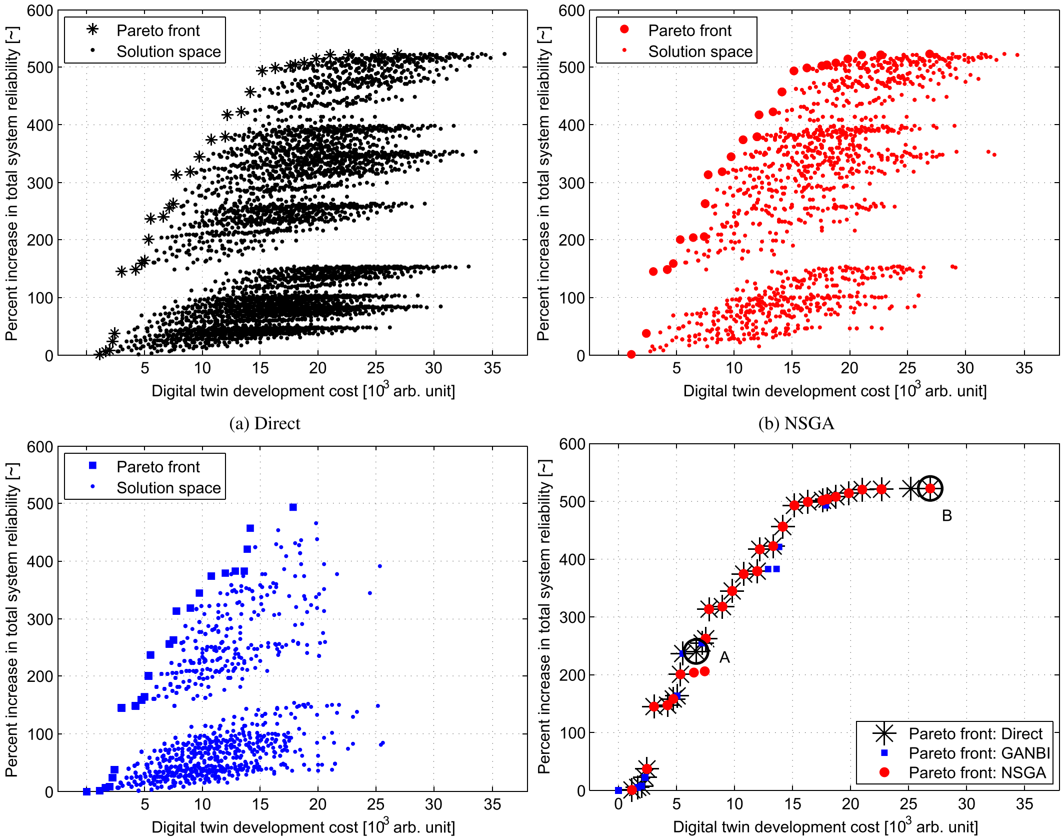
$\mathscr{S}$	$\mathscr{R}$[~]	$\mathscr{R}'$ [~]	$\mathscr{C}$[arb. unit]
1	0.95	0.96	1159
2	0.72	0.90	4367
3	0.40	0.98	3069
4	0.83	1.00	2991
5	0.13	0.58	1988
6	0.06	0.46	2713
7	0.08	0.15	4495
8	0.16	0.22	2463
9	0.32	0.88	2275
10	0.30	0.73	4184
11	0.01	0.89	4690
12	0.54	0.63	1664

5.3. Approximation results for unmanned underwater vehicle

Now we consider the vehicle in our case study. We show the initial reliabilities and improvements based on discussions from the UUV maintainers and operators. These values are shown in Table 3 below.

We show the resulting optimization and extraction of candidate TDTs s^* in Fig. 8. In Fig. 8a, we have plotted the Pareto frontier from the GANBI algorithm, shown as blue squares, as well as the NSGA method, shown as red circles. We have not utilized the direct computation, since the system of $N = 17$ components is sufficiently complex that the feasible space is too large to calculate directly. We have also extracted three candidate TDT subsets, shown as points "A," "B," and "C." These are summarized in Table 4. The initial system reliability calculated from R is 0.43. This comes from using a series connected L vector with 17 components. Since the vehicle is a minimalist UUV, there are no redundant components, and therefore the total system reliability depends on the individual subsystems, as in (4). This initial reliability indicates a failure rate of $\lambda = 0.42\ \mathrm{hr}^{-1}$ (calculated using $t_{op} = 2$ hr). This gives a MTBF of 2.37 hr or roughly after every mission some component stops functioning as expected. Using this number we are able to calculate expected system reliabilities for the subsets s_A, s_B, and s_C shown in Table 4.

While these subsets are selected at random, they raise key questions that need to be answered by continued discussions with decision makers. First, there is a 750 000 arb. unit cost difference between options A and B, with a 20% difference in total system reliability. Does the reliability improvement merit the additional cost burden? Should option C be prioritized for its simplicity and low cost? At this point it is natural to address these questions using higher-level information provided by discussions with decision makers. Without the added information, strictly, these options are all equivalent. Something to consider is the aforementioned maintainability of these components. Which of these is far more challenging to maintain, requiring significant downtime? The components that present the longest downtimes and appear in the TDT candidate subsets could be prioritized, since the maintenance could be scheduled when the vehicle is not required for operations. If the UUV is selected for deep water drilling platform inspections, then the reliability improvement is likely a significant factor due to the inherent risk of



(a) Direct

(b) NSGA

(c) GANBI

(d) Pareto fronts for the direct, NSGA, and GANBI methods

Fig. 7. (Color online) Percent increase in total system reliability from the use of digital twins for the triage components vs total implementation cost for $N = 12$ components. In (d) the circles labeled "A" and "B" indicate Pareto front solutions. The components to be upgraded are $s_A = \{1,3,8\}$ and $s_B = \{1,2,3,4,8,9,10,11,12\}$, where "A" has a cost of $C_{\text{tot}} = 6691$ arb. unit and "B" has a cost of $C_{\text{tot}} = 26\,862$ arb. unit. The number of generations used is $\text{N}_{\text{gen}} = 500$, the population size is $\text{N}_{\text{pop}} = 50$, and the number of points used in the GANBI method is $\text{N}_{\text{pts}} = 5$. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Table 3
Parameters for unmanned underwater vehicle.

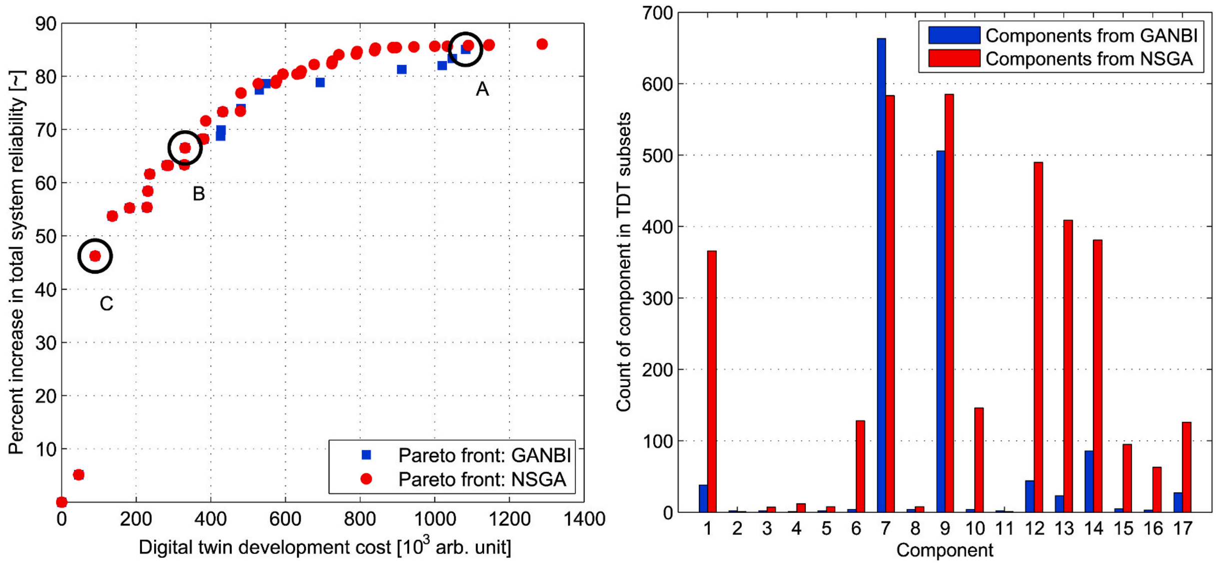
$\mathscr{S}$	Name	$\mathscr{R}$[~]	$\mathscr{R}'$ [~]	$\mathscr{C}$[10^3 arb. unit]
1	Acoustic modem electronics	0.9608	0.99	95
2	Acoustic modem transducer	0.996	0.9967	142
3	Nose cone	0.996	0.9967	56
4	Tail cone	0.996	0.9967	56
5	Mid section	0.996	0.9967	56
6	Handle (mast)	0.9608	0.9802	150
7	Batteries	0.9048	0.9512	46
8	Depth sensor	0.996	0.9967	90
9	Fins	0.6703	0.9802	90
10	Propeller	0.9802	0.99	66
11	Vehicle computer	0.996	0.9967	145
12	Static ballast	0.9048	0.9512	100
13	Waterproof servos	0.9231	0.99	150
14	Propeller shroud	0.9802	0.9967	46
15	Front bulkhead	0.9934	0.9967	49
16	Rear bulkhead	0.9934	0.9967	49
17	O-Rings	0.996	0.9967	47

having people perform this job. If the UUV is used for shallow water oceanographic surveys, then the reliability is likely not as an important factor as the implementation cost and option C will prevail.

We also show how often each component appears in a TDT candidate subset in Fig. 8b. The histogram shows that there are some differences between the GANBI and the NSGA methods used here. Nevertheless, there are standout components that appear consistently more often. Note that $s = 7, 9,$ and 12 appear more often. This is expected since these have the lower initial reliabilities. These are also the components that require maintenance most often, according to the vehicle maintainers. As a result, these components are natural candidates for TDT monitoring. Equipped with these subsets, the conversation with decision makers will be more directed and productive, since the selection of the TDT subsets is bolstered by inputs from multiple stakeholders and team members, all of whom contribute in a meaningful way, guiding the selection process with relevant information. Moreover, the MBSE effort establishes the necessary data acquisition interfaces and physical interfaces that directly influence the implementation of DTs once a TDT subset is downselected from the candidates. This shortens the design and development timeline since at a minimum subject matter experts or those responsible for DT development will have requirements established.

5.4. Numerical performance comparison

For both the example system and the UUV system data analyzed in Subsections 5.2 and 5.3 the direct computation, NSGA, and GANBI methods were performed using custom code developed in MATLAB® R2013a on a computer running Windows 10, with 8 GB RAM, and two Intel® Core™ i5-4200M CPUs @ 2.50 GHz (MATLAB, 2013). Using a $N = 12$ and $N = 17$ system as a benchmark case with randomly generated reliabilities $r \in [0.50, 1]$ and costs $c \in [100, 10 \times 10^3]$ arb. unit we show averaged computation times for the three methods in Table 5 based on 10 individual runs of each algorithm. The table also contains computational complexities for the three algorithms based on N subsystems. According to Curry and Dagli, the NSGA algorithm is



(a) Percent increase in total system reliability vs. digital twin implementation cost

(b) Component count in TDT subsets

Fig. 8. (Color online) We show the Pareto frontiers and candidate subsets in (a) and the resulting frequency of appearance of individual components in candidate subsets in (b). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Table 4
Candidate TDT subsets s_A, s_B, and.s_C

s_A	Name	s_B	Name	s_C	Name
1	Acoustic modem electronics	1	Acoustic modem electronics	9	Fins
5	Mid section	7	Batteries		
6	Handle (mast)	9	Fins		
7	Batteries	12	Static ballast		
8	Depth sensor				
9	Fins				
10	Propeller				
11	Vehicle computer				
12	Static ballast				
13	Waterproof servos				
14	Propeller shroud				
16	Rear bulkhead				
$\mathrm{R}'(r)$	0.80		0.72		0.63
$\frac{\delta_{\mathrm{R}'\mathrm{R}}}{\mathrm{R}}100$	85%		66%		46%
C_{tot}	1 083 000		331 000		90 000

Table 5
Computation time and complexities for direct, NSGA, and GANBI computation methods using $N = 7, N = 12$, and $N = 17$ subsystems with $\mathrm{N_o} = 2, \mathrm{N_{pop}} = 50$, $\mathrm{N_{gen}} = 500$, and $\mathrm{N_{pts}} = 5$. The computation times were calculated by averaging over 10 runs of randomly configured subsystems.

Algorithm	Computation time [s]			Complexity
	$N = 7$	$N = 12$	$N = 17$	
Direct	0.01	2.29	2082.35	$\mathscr{O}(\mathrm{N_o} \cdot (2^N)^{\mathrm{N_o}+1})$
NSGA	48.76	46.83	49.19	$\mathscr{O}(\mathrm{N_o} \cdot \mathrm{N_{pop}^3})$
GANBI	57.24	63.68	63.87	$\mathscr{O}(\mathrm{N_{pts}} \cdot \mathrm{N_{pop}^3})$

$\mathscr{O}(\mathrm{N_o N_{pop}^3})$, where $\mathrm{N_o}$ is the number of objective functions. In this case, there are two objective functions, indicating that the complexity is $\mathscr{O}(2 \cdot \mathrm{N_{pop}^3})$. The GANBI algorithm is a preprocessor for the NSGA, so its complexity is that of the NSGA plus a term that accounts for the additional helper objective functions $(\mathrm{N_{pts}} + 2) \cdot \mathrm{N_{pop}^3}$. The leading order in the large population size limit is $\mathrm{N_{pts}} \cdot \mathrm{N_{pop}^3}$. The distinction must be made that although the GANBI method requires additional steps to compute the supporting objectives, it gives a better approximation to the true pareto front of the MOOP. The direct method requires the computation of all possible combinations of subsystems, which strictly is $2^N - 1$, accounting for the constraint that not all subsystems can be selected. This term is dominated by 2^N, so we can neglect the -1. This is done for every objective. Every objective is then compared with every other objective. For small values of $N \ll \mathrm{N_{pop}}$, it is clear that the direct method is far better than the evolutionary algorithms, since the complexity scales with cubic population size.

The values shown in Table 5 are for a notional subsystem with $N = 7$, $N = 12$, and $N = 17$ subsystems. We have argued thus far that for a small number of subsystems $N \approx 10$ the direct method is the most efficient, whereas for a large number of subsystems $N \approx 20$ evolutionary algorithms such as the NSGA and the NSGA with GANBI are the most efficient. The keen observer will note that the computational times shown in the table suggest that the direct method is fast, even as the number of subsystems exceeds the lower limit. In fact it would seem that the direct method is an order of magnitude faster than the times of the evolutionary algorithms for the $N = 12$ case. This directly contradicts the complexities shown in the table as well, since a simple calculation for the $N = 12$ case yields $\mathscr{O}(10^{11})$ for the direct method, and $\mathscr{O}(10^5)$ and $\mathscr{O}(10^6)$ for NSGA and GANBI, respectively. This is because of two primary factors: First, the direct method computes all combinations of costs and reliabilities, which for $N = 12$ is 4095 different combinations, accounting for the constraints. This is it, essentially, since a simple comparison is performed to determine the non-dominated solutions. Second, the evolutionary algorithms on the other hand must initialize a random population of chromosomes using $\mathrm{N_{pop}} = 50$ with length N, evaluate fitness, reproduce, crossover, mutate, and reevaluate for $\mathrm{N_{gen}} = 500$ generations. Assuming the three algorithms have been programmed to a similar level of efficiency, then the evolutionary algorithms take longer from a computational perspective simply because of the number of steps involved. It is therefore more appropriate to utilize computational complexity in the large N limit and recognize that for a small number of subsystems, the direct method is more efficient because of how few steps there are programmatically.

6. Discussion

We have introduced a generalizable process for determining the subset of components from a SoS that when monitored through digital twins yield the maximal increase in total system reliability and minimize the cost entrance hurdle for implementing the digital twins. We have utilized our process on a real-life application for an unmanned underwater vehicle. Through a combination of experimentally derived system reliabilities and an MBSE effort, where numerous discussions were held with the vehicle development team, we have been able to demonstrate the applicability of this relatively simple approach to TDT selection. The implications for the industry of unmanned underwater vehicles are that through the process developed in the paper, groups that rely on UUVs for critical tasking which would otherwise place humans in direct harm can use the process for retroactively developing digital twins to increase system reliability, while staying within budgetary constraints. Our process considers input from a variety of stakeholders and system users to ensure that the most accurate information is solicited for inclusion in the process. We have not introduced variables, data, or parameters that are not readily available or those that would not be determined regardless of the TDT selection process. The system reliabilities $\mathscr{R}$ are usually well known from experimental test data, component data sheets, or can be determined from like-system analysis. The reliabilities as a result of utilizing a twin to monitor the corresponding component $\mathscr{R}'$ can be estimated based on a decrease in failure rates and discussions with subject matter experts responsible for implementing the digital twin. The project management team in collaboration with the subject matter experts can provide cost estimates $\mathscr{C}$ for developing digital twins for the subsystems based on known parameters such as labor rates, auxiliary software and IT costs, and historical development and testing timelines. These parameters and the outputs of the connectivity analysis from the MBSE suffice to perform a selection process on the subsystems to be monitored. As we have shown, higher level information is required to prioritize the subset of TDTs s^* that are selected as the requisite systems for implementing digital twins, such as budgetary constraints or reliability requirements. The implications for the field of reliability studies are a generalizable process that is straightforward to utilize for a variety of systems that would like to leverage digital twin technology for enhancing reliability of legacy systems. Our process is particularly well-suited for those who do not know which components to select for digital twin monitoring, must rely on legacy systems for critical tasking, and do not have the budget to design, develop, test, and deploy complete digital twins for their systems.

CRediT authorship contribution statement

Demetrious T. Kutzke: Conceptualization, Methodology, Investigation, Writing - original draft, Writing - review & editing. **James B. Carter:** Methodology, Investigation, Writing - original draft. **Benjamin T. Hartman:** Software, Formal analysis, Writing - review & editing.

Declaration of competing interest

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

Acknowledgments

The authors would like to thank the entire vehicle development team at NSWC PCD for informing the modeling process and the continued support they have provided through numerous discussions and meetings. The authors would like especially to thank M. McBain who was instrumental in determining the appropriate reliability estimates based on her experience as both an operator and maintainer of the vehicles. Finally, the authors are indebted to Dr. Patrick Walters, the system architect who designed the vehicle and provided the experimental system on which the paper is based. This work has been supported by the NSWC PCD Naval Innovative Science and Engineering (NISE) Program.

References

Biggs, G., Juknevicius, T., Armonas, A., Post, K., 2018. Integrating safety and reliability analysis into mbse: overview of the new proposed omg standard. INCOSE Int. Symp. 28, 1322–1336. https://doi.org/10.1002/j.2334-5837.2018.00551.x.

Coit, D.W., Smith, A.E., 1996. Reliability optimization of series-parallel systems using a genetic algorithm. IEEE Trans. Reliab. 45, 254–260. https://doi.org/10.1109/24.510811.

Coraddu, A., Oneto, L., Baldi, F., Cipollini, F., Atlar, M., Savio, S., 2019. Data-driven ship digital twin for estimating the speed loss caused by the marine fouling. Ocean Eng. 186, 106063. https://doi.org/10.1016/j.oceaneng.2019.05.045.

Custódio, A.L., Madeira, J.F.A., Vaz, A.I.F., Vicente, L.N., 2011. Direct multisearch for multiobjective optimization. SIAM J. Optim. 21, 1109–1140. https://doi.org/10.1137/10079731X.

D'Ambrosio, J., Soremekun, G., 2017. Systems engineering challenges and mbse opportunities for automotive system design. In: 2017 IEEE Int. Conf. SMC, pp. 2075–2080. https://doi.org/10.1109/SMC.2017.8122925.

Daneshmand, M., Bilici, O., Bolotnikova, A., Anbarjafari, G., 2017. Medical robots with potential applications in participatory and opportunistic remote sensing: a review. Robot. Autonom. Syst. 95, 160–180. https://doi.org/10.1016/j.robot.2017.06.009.

Downer, J., 2011. On audits and airplanes: redundancy and reliability-assessment in high technologies. Account. Org. Soc. 36, 269–283. https://doi.org/10.1016/j.aos.2011.05.001.

Endrenyi, J., Aboresheid, S., Allan, R.N., Anders, G.J., Asgarpoor, S., Billinton, R., Chowdhury, N., Dialynas, E.N., Fipper, M., Fletcher, R.H., Grigg, C., McCalley, J., Meliopoulos, S., Mielnik, T.C., Nitu, P., Rau, N., Reppen, N.D., Salvaderi, L., Schneider, A., Singh, C., 2001. The present status of maintenance strategies and the impact of maintenance on reliability. IEEE Trans. Power Syst. 16, 638–646. https://doi.org/10.1109/59.962408.

Fletcher, B., 2000. UUV master plan: a vision for navy UUV development. In: Proc. 2000 MTS/IEEE OCEANS Conf. Exhib. IEEE, Providence, RI, USA, pp. 65–71. https://doi.org/10.1109/OCEANS.2000.881235.

Fletcher, B., Wernli, R.L., 2003. Expanding missions for small unmanned undersea vehicles (UUVs). In: Proc. ASME 2003 22nd Int. Conf. OMAE. ASME, Cancún, Mexico, pp. 599–602. https://doi.org/10.1115/OMAE2003-37254.

Friedenthal, S., Moore, A., Steiner, R., 2015. A Practical Guide to SysML: the Systems Modeling Language, third ed. Elsevier. https://doi.org/10.1016/C2013-0-14457-1.

Fuller, A., Fan, Z., Day, C., Barlow, C., 2020. Digital twin: enabling technologies, challenges and open research. IEEE Access 8, 108952–108971. https://doi.org/10.1109/ACCESS.2020.2998358.

Garg, H., Sharma, S., 2013. Multi-objective reliability-redundancy allocation problem using particle swarm optimization. Comput. Ind. Eng. 64, 247–255. https://doi.org/10.1016/j.cie.2012.09.015.

Glaessgen, E., Stargel, D., 2012. The digital twin paradigm for future NASA and U.S. Air Force vehicles. In: Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf. https://doi.org/10.2514/6.2012-1818.

Grieves, M., 2005. Product Lifecycle Management: Driving the Next Generation of Lean Thinking: Driving the Next Generation of Lean Thinking: Driving the Next Generation of Lean Thinking. McGraw Hill Professional.

Grieves, M., Vickers, J., 2017. Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems. In: Transdisciplinary Perspectives on Complex Systems. Springer, Cham, pp. 85–113. https://doi.org/10.1007/978-3-319-38756-7_4.

Hanachi, H., Liu, J., Banerjee, A., Chen, Y., Koul, A., 2015. A physics-based modeling approach for performance monitoring in gas turbine engines. IEEE Trans. Reliab. 64, 197–205. https://doi.org/10.1109/TR.2014.2368872.

Hilton, G., Needham, J., 2019. Welcome to Boeing's Factory of the Future: Manufacturing Revolution Comes to the U.K. Innov. Q.

Hong, S., Zhou, Z., Zio, E., Hong, K., 2014. Condition assessment for the performance degradation of bearing based on a combinatorial feature extraction method. Digit. Signal Process. 27, 159–166. https://doi.org/10.1016/j.dsp.2013.12.010.

Horner, R., El-Haram, M., Munns, A., 1997. Building maintenance strategy: a new management approach. J. Qual. Mainten. Eng. 3, 273–280. https://doi.org/10.1108/13552519710176881.

Huang, Z., Swalgen, S., Davidz, H., Murray, J., 2017. MBSE-assisted FMEA approach — challenges and opportunities. In: 2017 Annu. RAMS, pp. 1–8. https://doi.org/10.1109/RAM.2017.7889722.

Hylton, W.S., 2020. History's largest mining operation is about to begin: it's underwater—and the consequences are unimaginable. The Atlantic.

Karuppuswamy, P., Sundararaj, G., Devadasan, S., Elangovan, D., Savadamuthu, L., 2006. Failure reduction in manufacturing systems through the risk management approach and the development of a reactive maintenance model. Int. J. Risk Assess. Manag. 6, 545–564. https://doi.org/10.1504/IJRAM.2006.009553.

Kim, H., Jin, C., Kim, M., Kim, K., 2020. Damage detection of bottom-set gillnet using artificial neural network. Ocean Eng. 208, 107423. https://doi.org/10.1016/j.oceaneng.2020.107423.

Kim, H., Kim, P., 2017. Reliability–redundancy allocation problem considering optimal redundancy strategy using parallel genetic algorithm. Reliab. Eng. Syst. Saf. 159, 153–160. https://doi.org/10.1016/j.ress.2016.10.033.

Koenig, K.L., Schultz, C.H., 2010. Koenig and Schultz's Disaster Medicine: Comprehensive Principles and Practices. Cambridge University Press.

Koulamas, C., Kalogeras, A., 2018. Cyber-physical systems and digital twins in the industrial internet of things [cyber-physical systems]. Comput. Times 51, 95–98. https://doi.org/10.1109/MC.2018.2876181.

Kousi, N., Gkournelos, C., Aivaliotis, S., Giannoulis, C., Michalos, G., Makris, S., 2019. Digital twin for adaptation of robots' behavior in flexible robotic assembly lines. Procedia Manuf 28, 121–126. https://doi.org/10.1016/j.promfg.2018.12.020.

Kukulya, A., Plueddemann, A., Austin, T., Stokey, R., Purcell, M., Allen, B., Littlefield, R., Freitag, L., Koski, P., Gallimore, E., Kemp, J., Newhall, K., Pietro, J., 2010. Under-ice operations with a REMUS-100 AUV in the arctic. In: Proc. 2010 IEEE/OES AUV. IEEE, Monterey, CA, USA, pp. 1–8. https://doi.org/10.1109/AUV.2010.5779661.

Kumar, A., Singh, S., Ram, M., 2020. Systems reliability assessment using hesitant fuzzy set. Int. J. Oper. Res. 38, 1–18. https://doi.org/10.1504/IJOR.2020.106357.

Kusiak, A., 2017. Smart manufacturing must embrace big data. Nat 544, 23–25. https://doi.org/10.1038/544023a.

Laaki, H., Miche, Y., Tammi, K., 2019. Prototyping a digital twin for real time remote control over mobile networks: application of remote surgery. IEEE Access 7, 20325–20336. https://doi.org/10.1109/ACCESS.2019.2897018.

Li, Y., Nilkitsaranont, P., 2009. Gas turbine performance prognostic for condition-based maintenance. Appl. Energy 86, 2152–2161. https://doi.org/10.1016/j.apenergy.2009.02.011.

Liu, J., Wang, W., Ma, F., Yang, Y., Yang, C., 2012. A data-model-fusion prognostic framework for dynamic system state forecasting. Eng. Appl. Artif. Intell. 25, 814–823. https://doi.org/10.1016/j.engappai.2012.02.015.

Liu, M., Fang, S., Dong, H., Xu, C., 2020. Review of digital twin about concepts, technologies, and industrial applications. J. Manuf. Syst. https://doi.org/10.1016/j.jmsy.2020.06.017.

Lund, A.M., Mochel, K., Lin, J.W., Onetto, R., Srinivasan, J., Gregg, P., Eric Bergman, J., Hartling, Jr, K.D., Ahmed, A., Chotai, S., U.S. Patent 15/075,231, Nov. 2018. Digital Wind Farm System. General Electric.

Lund, A.M., Mochel, K., Lin, J.W., Onetto, R., Srinivasan, J., Gregg, P., Eric Bergman, J., Hartling, Jr, K.D., Ahmed, A., Chotai, S., U.S. Patent 9 995 278 B2, Jun. 2018. Digital Twin Interface for Operating Wind Farms. General Electric.

Matlab, 2013. MATLAB Version 8.1.0.604 (R2013a). The Mathworks, Inc., Natick, Massachusetts. URL. https://www.mathworks.com/products/matlab.html.

McFarlane, J.R., 2008. Tethered and untethered vehicles: the future is in the past. In: Proc. 2008 MTS/IEEE OCEANS. IEEE, Quebec City, QC, Canada, pp. 1–4. https://doi.org/10.1109/OCEANS.2008.5151918.

Miller, D.P., 1996. Design of a small, cheap UUV for under-ship inspection and salvage. In: Proc. OES Symp. AUV Tech. IEEE, Monterey, CA, USA, pp. 18–20. https://doi.org/10.1109/AUV.1996.532395.

Mitchell, S.W., 2010. Complex product family modeling for common submarine combat system mbse. In: 3rd Int. Conf. MBSE. Citeseer, Fairfax, VA.

Muhuri, P.K., Ashraf, Z., Lohani, Q.M.D., 2018. Multiobjective reliability redundancy allocation problem with interval type-2 fuzzy uncertainty. IEEE Trans. Fuzzy Syst. 26, 1339–1355. https://doi.org/10.1109/TFUZZ.2017.2722422.

Pérez-Rosés, H., 2018. Sixty years of network reliability. Math. Comput. Sci. 12, 275–293. https://doi.org/10.1007/s11786-018-0345-5.

Prajapati, A., Bechtel, J., Ganesan, S., 2012. Condition based maintenance: a survey. J. Qual. Mainten. Eng. 18, 384–400. https://doi.org/10.1108/13552511211281552.

Qi, Q.L., Tao, F., 2018. Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. IEEE Access 6, 3585–3593. https://doi.org/10.1109/ACCESS.2018.2793265.

Rochlin, G.I., La Porte, T.R., Roberts, K.H., 1987. The self-designing high-reliability organization: aircraft carrier flight operations at sea. Nav. War Coll. Rev. 40, 76–92.

Ross, R., Pillitteri, V., Graubart, R., Bodeau, D., McQuaid, R., 2019. Developing Cyber Resilient Systems: A Systems Security Engineering Approach. Special Publication (SP) 800-160. National Institute of Standards, Gaithersburg, MD. https://doi.org/10.6028/NIST.SP.800-160v2.

Savage, C., 1997. A survey of combinatorial gray codes. SIAM Rev. 39, 605–629. https://doi.org/10.1137/S0036144595295272.

Sparx Systems. Enterprise architect. URL. https://sparxsystems.com/products/ea/index.html.

Stanford Artificial Intelligence Laboratory, et al.. Robotic operating system (ROS). URL. https://www.ros.org.

Swanson, L., 2001. Linking maintenance strategies to performance. Int. J. Prod. Econ. 70, 237–244. https://doi.org/10.1016/S0925-5273(00)00067-0.

Tan, Y., Niu, C., Tian, H., Hou, L., Zhang, J., 2019. A one-class svm based approach for condition-based maintenance of a naval propulsion plant with limited labeled data. Ocean Eng. 193, 106592. https://doi.org/10.1016/j.oceaneng.2019.106592.

Tan, Y., Tian, H., Jiang, R., Lin, Y., Zhang, J., 2020. A comparative investigation of data-driven approaches based on one-class classifiers for condition monitoring of marine machinery system. Ocean Eng. 201, 107174. https://doi.org/10.1016/j.oceaneng.2020.107174.

Tao, F., Qi, Q., 2019. Make more digital twins. Nat 573, 490–491. https://doi.org/10.1038/d41586-019-02849-1.

Tao, F., Qi, Q., Wang, L., Nee, A., 2019a. Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: correlation and comparison. Eng. Times 5, 653–661. https://doi.org/10.1016/j.eng.2019.01.014.

Tao, F., Zhang, H., Liu, A., Nee, A.Y.C., 2019b. Digital twin in industry: state-of-the-art. IEEE Tran. Ind. Inform. 15, 2405–2415. https://doi.org/10.1109/TII.2018.2873186.

Thieme, C.A., Utne, I.B., 2017. Safety performance monitoring of autonomous marine systems. Reliab. Eng. Syst. Saf. 159, 264–275. https://doi.org/10.1016/j.ress.2016.11.024.

Tobias, P., 2013. Assessing product reliability. In: NIST/SEMATECH E-Handbook of Statistical Methods. National Institute of Standards. https://doi.org/10.18434/M32189.

Wettergren, T.A., 2006. The Genetic-Algorithm-Based Normal Boundary Intersection (GANBI) Method: an Efficient Approach to Pareto Multiobjective Optimization for Engineering Design. Naval Undersea Warfare Center Newport Division. Newport, RI. NUWC-NPT Technical Report 11,741.

Whitcomb, L.L., 2000. Underwater robotics: out of the research laboratory and into the field. In: Proc. 2000 IEEE ICRA. IEEE, San Francisco, CA, USA, pp. 709–716. https://doi.org/10.1109/ROBOT.2000.844135.

Yeh, C.T., Fiondella, L., 2017. Optimal redundancy allocation to maximize multi-state computer network reliability subject to correlated failures. Reliab. Eng. Syst. Saf. 166, 138–150. https://doi.org/10.1016/j.ress.2016.08.026.

Yuan, X., Cai, B., Ma, Y., Zhang, J., Mulenga, K., Liu, Y., Chen, G., 2018. Reliability evaluation methodology of complex systems based on dynamic object-oriented bayesian networks. IEEE Access 6, 11289–11300. https://doi.org/10.1109/ACCESS.2018.2810386.

Zou, G., Banisoleiman, K., González, A., Faber, M.H., 2019. Probabilistic investigations into the value of information: a comparison of condition-based and time-based maintenance strategies. Ocean Eng. 188, 106181. https://doi.org/10.1016/j.oceaneng.2019.106181.