Martin Defense Group
841 Bishop St., Suite 1110
Honolulu, Hawaii 96813

University of Michigan
Naval Architecture and Marine Engineering
Ann Arbor, Michigan 48109

A Generic Framework for Data Model Fusion

A report prepared for

The Office of Naval Research

as part of contract N00014-20-C-1099 titled

Data-Model Fusion for Naval Platforms and Systems

Corresponding Authors:
Jason Provancher, Martin Defense Group
jprovancher@mdefensegroup.com
Matt Collette, PhD, University of Michigan
mdcoll@umich.edu
Matt Reese, Martin Defense Group
mreese@mdefensegroup.com

March 2022

REVISION HISTORY

June 25, 2021	First Revision
October 15, 2021	Added “Survey of Fusion Approaches and Opportunities”. Some editing to enhance flow and remove redundancies.
March 10, 2022	Added “Survey of Data Persistence Approaches”. Moved Annotated references from Chapter 1 to Appendix.

CONTENTS

EXECUTIVE SUMMARY

This report was prepared for the Office of Naval Research in fulfillment of Task 1 of contract N00014-20-C-1099, titled "Data-Model Fusion for Naval Platforms and Systems." The objective of Task 1 is to develop a rigorous, generic framework for naval applications of Data-Model Fusion.

Data-Model Fusion (DMF) is a concept developed at Martin Defense Group (legacy Navatek), in conjunction with the University of Michigan, to describe:

1. Data: information from sensors, expert knowledge, reports, inspections, surveys, or other sources regarding physical components, systems, platforms or fleets within available operating conditions
2. Model: digital representations (e.g., empirical equations, physics-based models, networks, ontological characterizations, etc) of those components, systems, platforms or fleets within simulated operating conditions.
3. Fusion: the integration of said data and models to bring both into agreement. Our data-model fusion approach uses data science techniques and machine learning methods to improve state estimates, update model parameters, identify operational areas or anomalies, and inform decision-making.

Our integration approach utilizes and expands upon the state-of-the-art in data science and Artificial Intelligence (AI)-based decision support to provide real-time actionable diagnostic and/or prognostic information on the state of real-world physical platforms. When a digital model is operationally coupled via sensors to a specific real-world component, system, platform, or fleet, we refer it as a digital twin. Use cases for twins involve managing degrading systems, improving performance, updating design approaches, and optimal planning for a fleet of similar platforms. Digital twins are not a necessary component in data-model fusion, but they are frequently used as a basis for analysis and decision-making in our real-world system applications.

This report is a compilation of six separate reports, with an overall focus on defining a fundamental framework for data-model fusion in the naval domain. We start in Chapter 1 with a literature survey of approaches, gaps, and opportunities in data-model fusion. Next, we define in Chapter 2 a unified theory of digital twins, followed in Chapter 3 by a delineation of digital twin types in the naval domain. In Chapter 4 we discuss techniques for data persistence that enable storage of the geometry models, measurements, and environmental data needed by twins. In Chapter 5 we shift our focus back to data-model fusion, providing a survey of tools and techniques that can be used to inform naval systems design and operation. We develop methods for understanding and managing the implications, risks, and opportunities of digital naval engineering with respect to the design and operation of autonomous naval platforms and systems. Chapter 6 was written to serve as a primer on AI-based decision support methodologies, tools, and techniques for practicing naval research engineers and scientists. Finally, we conclude this report with a discussion on technology transfer, capability gaps, and opportunities for further research.

CHAPTER 1 SURVEY OF FUSION APPROACHES AND OPPORTUNITIES

Author: Matthew Collette[1]
1 - University of Michigan, Department of Naval Architecture and Marine Engineering
Date: July 2021
Marine Structures Design Laboratory Report Number: 2021-001

Abstract: Digital twins are becoming commonplace in discussions of future marine platforms. However, the community still lacks a standard definition and understanding of the state-of-the-art for such approaches. This report provides: a review of the historical development of digital twins, a formal definition of a digital twin that is contrasted to other definitions in the literature, a review of digital twin use cases, and a deep dive into the fusion step where a twin brings a numerical model and real-world observations into agreement. Digital twins are seen as an evolutionary development joining several related fields, with origins in the early 1990s. Use cases for twins involve managing degrading systems, improving performance, updating design approaches, and optimal planning for a fleet of similar platforms. Through a detailed review of 32 papers focused on the fusion step in twins, the state-of-the-art in fusion is documented. Fusion approaches appear to be in active development, with significant progress at the component level. Further work is needed for larger systems, integration into decision making, quantifying uncertainty, and to help select particular algorithmic approaches for differing applications.

SECTION 1.1 INTRODUCTION

The phrase "Digital Twin" has become commonplace over the last decade. Built upon a mixture of existing computational frameworks and an increased ability to sense the as-built world, twins have promised improved performance and safety for large engineering systems such as naval vessels. However, real-world success stories have been far fewer, with careful blending of data, simulation, and decision-making required to benefit from the twin approach. This report explores the current state-of-the-art of digital twins, focusing on two particular aspects of the twin system: (1) the potential applications of the twin for naval vessels and (2) the mechanics of fusing data with computational models, known as the fusion step. These two components are not comprehensive – the overall needs of the twin system are explored further in the next section. However, the needs and fusion approaches are linked and form the internal core of most twin systems and represent a logical starting point for exploring digital twins. The remainder of this document is divided into four sections. First, an overview of digital twins, fusion approaches, and the history of the concept are presented. Second, a review of potential applications for digital twins on naval vessels, drawing upon academic, commercial, and governmental literature. Third, a detailed academic literature search on fusion methods is presented, with the current state-of-the-art examined in detail. Fourth, the state-of-the-art in fusion methods is then contrasted with the desired applications for naval digital twins to determine areas needed for future research.

SECTION 1.2 TWINS AND FUSION

SECTION 1.2.1 WHAT IS A TWIN?

The term digital twin has become nebulous over time, as related concepts and commercial products have used the term "digital twin" to describe their products or procedures. To restore a stricter definition to this term, we examine the qualities that are unique to the digital twin in the marine setting. Wincott and Collette [1] proposed that a system must have each of the following characteristics to be considered a digital twin:

1. **A real-world system of interest:** Twins are specific to real-world systems. A simple twin may model one particular vessel or component on a vessel, while a more complex twin might span a fleet of similar vessels and integrate knowledge across the fleet. However, the ability to track discrete, real-world systems is key to the twin. While some talk about "design stage" digital twins, such a term only applies to an in-development twin or planned reuse of design-stage engineering products. Digital twins can interface or build upon model-based system engineering approaches, but the twin is not complete until there is a specific physical system in the real world.

2. **One or more digital representations of the system:** A true twin requires some sort of digital representation of the real-world object. Here, twins can differ dramatically in the complexity of this representation, which could be as simple as a regression equation or machine learning product, and as complex as a high-fidelity coupled CFD and FEA simulation. The digital representation may focus on engineering prediction (e.g., a simulation model) or may instead focus on data storage and integration (a geometric or product structure model). Twins may include more than one level of model fidelity or modeling approach and may switch between them as needed.

3. **Fusion to join the real-world system and the digital representation:** The twin must be able to relate events in the real world to the digital model. Making such a relationship can be as simple as inputting visual inspections (e.g., coating health or corrosion data to a structural twin) periodically, or it could comprise a multi-channel monitoring system in real-time. Many twins may fuse more than one type of information – e.g., visual inspections combined with weather records and recorded strains. This step involves both data acquisition and integration of the data with the digital representations.

4. **A decision that will depend on the output of the fusion step:** To differentiate between a twin and a monitoring or validation campaign, we also introduce the concept that a twin must influence a decision. This decision could be advisory (e.g., recommending optimal deployment scheduling of a fleet of assets for equal use), or it could be fully automated with final authority over some part of the vessel's operation.

In a deeper examination into the architecture of digital twins on this project, a revised definition was proposed that moved item 4, the decision-making requirement out of the formal twin, but into a twin system [2]. The idea behind this revision is that decision-making may require input from multiple, independent twins and other data sources. Additionally, different decision frameworks could be proposed for the same twin for differing decisions. A sketch of this definition is shown below. The image shows that the same first three requirements remain, with the decision moved to a different layer of the framework.

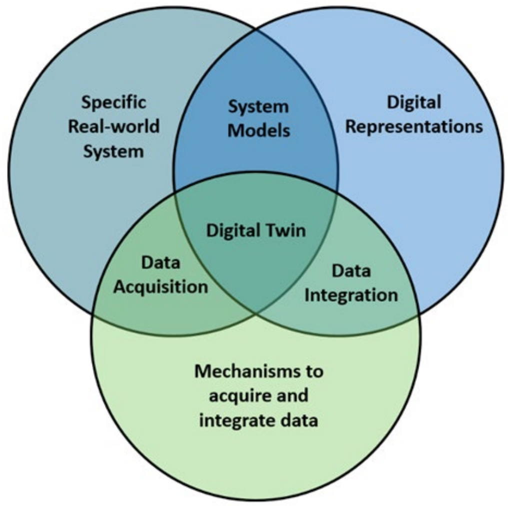


Figure 1: Revised Twin Scope Proposed with Decision Making System Complimentary – After [2]

Regardless of if the decision-making system is integrated into the formal twin, or considered as part of the twin system, a consistent vision of a twin emerges from this discussion. A twin must have a specific, real-world system or systems that it is trying to model. A twin must have one or more digital representations of the physical system and some sort of fusion approach to gather data from the physical world and integrate it with the digital model. A twin must also address decision-making in some form, either directly coupled to the twin or by providing input to a wider decision system. Few other specific marine definitions of twins have been proposed. Erikstad provides one in the context of wind turbine simulation, which is more detailed but broadly similar to the themes proposed here [3], and a broader recent literature review has also been published [4].

The terms digital thread and digital systems model have also become widely used over the last several years. Both of these are distinct from the definition of a digital twin. Kraft [5] provides a recent series of definitions from the U.S. military community's perspective on the differences between these, which is reproduced below directly from Kraft's paper [5]:

> **Digital System Model** - A digital representation of a weapon system, generated by all stakeholders, that integrates the authoritative data, information, algorithms, and systems engineering processes that define all aspects of the system for the specific activities throughout the system lifecycle.
>
> **Digital Thread** - An extensible, configurable, and Agency enterprise-level analytical framework that seamlessly expedites the controlled interplay of authoritative data, information, and knowledge in the enterprise data- information-knowledge systems, based on the Digital System Model template, to inform decision-makers throughout a system's life cycle by providing the capability to access, integrate and transform disparate data into actionable information.

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

The definition of twin proposed by Kraft is slightly less specific than the working definition proposed here but entirely compatible. It can be seen that the digital system model and digital thread refer to activities carried out early in the design process (before a physical system exists) and associated infrastructure and knowledge management activities. These definitions, and indeed the concept of the system model and the digital thread, show both the potential for reuse of design stage models, as well as the challenges in tracking and making information available to twins throughout the product lifecycle.

In a recent review of digital twin literature for manufacturing applications, Negri [6] summarized 16 twin definitions that have appeared in the literature over recent years. Few of the definitions are as specific as the definition proposed here. These 16 definition summaries by Negri were compared to the four components of our current definition and shown in the table below:

Table 1: Comparison of Twin Definitions

Component of Current Definition	The number of times this appears in other's definition summarized in [6]
A specific real-world system of interest	13/16
One of more numerical models	16/16
Fusion to join digital/physical	4/16
A decision that depends on the twin	4/16

From this table, it is clear that the relationship between a physical system and a numerical model(s) is well established and common. However, many previous authors do not include the fusion and decision steps in their definitions, even if the systems are envisioned to use these components. This could be a result of a modeling bias, where past research has focused on the ability to actually perform the computations required for the modeling side of the twin, or worried primarily about the data flow and infrastructure required for the twin to work. The lack of formal discussion on these two aspects of the twin to date has motivated the present work's focus on these aspects of the twin system, starting with fusion models in this report.

SECTION 1.2.2 BRIEF HISTORY OF THE TWIN APPROACH IN THE MARINE FIELD

Answering the question of who first came up with the concept of the digital twin is difficult. The first publication that proposed what would be recognized as the core of a modern twin is credited to work done at the University of Michigan by Grieves in 2002 [7], which specifically proposed a virtual computational space mirroring the physical world in the context of automotive systems. However, a similar system with virtual and real-world spaces had, in fact, been presented for seakeeping decision making in Sweden almost a decade earlier by Huss and Olander [8]. Erikstad noted that product lifecycle management (PLM) systems developed in Norway by the DNV maritime corporation integrated corrosion and thickness measurements for future performance prediction in the mid-1990s, basically the same time as Huss and Olander [9]. The work of Pegg and Gibson, which is functionally similar, dates back to 1993 [10]. While these structural modeling approaches were temporally slower (often integrating manual field observations), they were fundamentally the same architecture now adopted for twins and that proposed by Grieves. Glaessgen and Stargel [11] credit the twin to DARPA's DSO but provide no evidence of prior publication to that of Grieves. It appears that the concept of the digital twin emerged organically from advances in sensing and computation in several places around the world. While its origin cannot now be precisely determined, no publication with a prior date to that of Huss and Olander has been found. Since this time, the twin concept has exploded, with rapid growth in papers, software, and products related to digital twins. This review will focus on the marine field twins, with occasional references to review articles or other major publications in the related fields (primarily aerospace and manufacturing to date).

The broad origin is likely a reflection that the twin concept is best seen as a development of prior work instead of a new departure. Figure 2 shows a partial list of existing technology that has contributed to the modern digital twin. Several of these approaches, such as vibration-based machinery monitoring approaches, probabilistic crack inspection and repair, and PLM data systems predate twins, yet provide valuable functionality to the modern twin toolbox. Additionally, the recent growth in machine learning, accurate global weather models, as well as the decrease in the cost of computational power and sensing have helped make twins practical. Thus, we can see twins as emerging from the maturity and interaction of a number of engineering developments. Once these foundational technologies were mature enough and inexpensive enough, a twin becomes a logical approach for increasing platform performance.

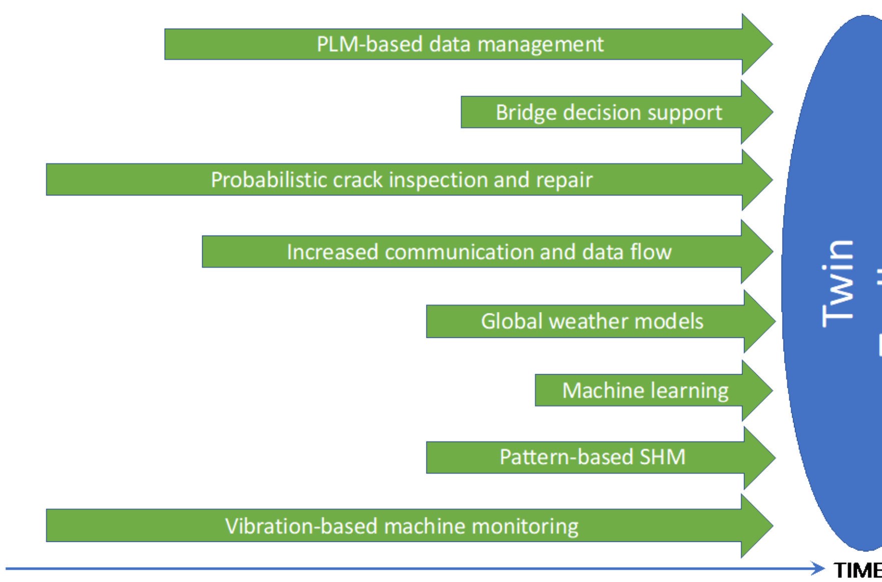


Figure 2: Historical Contributors to Current Twin Approach

The modern twin today then touches on several areas of related development, as shown in Figure 3. The concept of developing detailed simulation models underlies model-based systems engineering (MBSE). If such models are built during design, using them operationally becomes attractive, and in that way, MBSE can enable digital twins. Likewise, PLM and Digital Thread systems provide a backbone infrastructure capable of handling the data storage and retrieval necessary for twins. Structural health monitoring and condition-based maintenance also closely approximate twins, although in many cases, they do not have a detailed underlying numerical model, instead using pattern matching and thresholds to determine when intervention is required. Finally, the increasing desire to support autonomous vessels, where human crew members cannot monitor the vessels in real-time, and to support operability decisions even with a human crew has led to an interest in adapting digital twins for these use cases. Thus, the twin field today is broad and is expected to grow in conjunction with these related fields.

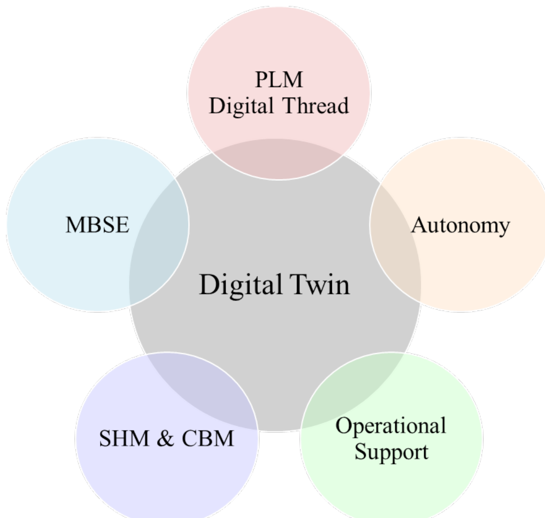


Figure 3: Overlap Between Twins and Related Disciplines

As shown in Figure 2, condition-based monitoring of machinery has been one of the primary forerunners of digital twin approaches. Vibration monitoring, electrical insulation testing, electrical power analysis, and other diagnostic approaches have been in use for decades. Most of these approaches started using simple degradation models or models learned from data without simulation of physics. However, physics-based models are now complementing data-driven models in some applications. Furthermore, hybrid models, which merge both approaches using fusion, are also growing in interest. Guo, Li, and Li [12] provide a recent review focused on both physics, data-driven, and hybrid models, while Kim, An, and Choi [13] provide a textbook introduction to these approaches with accessible code and examples. Alaswad and Xiang [14] provide a recent overview of non-physics degradation models used in this application. Marine-specific examples from the last three years include a recent review on corrosion modeling [15], deep-learning approaches for autonomous ship systems [16], wind-turbine mechanical systems [17][18], and propulsion systems [19].

Concurrently, product lifecycle management (PLM) approaches were also growing in popularity. Such approaches keep extensive product data and often engineering models with specific physical objects throughout the life of the object. In the marine field, an early area of application of such approaches was tracking corrosion and other structural defects over time. The emergence of the widespread use of FEA in the 1990s meant that many ships now had 3-D FEA models that could be updated with structural thickness gauging and other inspection data. Additionally, inspection data, photographs, and crack records could be easily stored and categorized. Such twins had limited numerical components beyond PLM – often, an FEA model was simply re-assessed against existing criteria, and rough forecasts of when corrosion would require the replacement of structural members could be made. Canada took the lead in this area, merging hydrodynamic loading, FEA models, and inspection results into a multi-level assessment system that could support operational deployment and repair decisions [10]. In the commercial world, ABS SafeShip [20] was one of the earlier examples of a similar approach, developed to leverage the 3-D FEA models created during design through the SafeHull application. While the inspection process and result input to the FEA model was still manual, and the fusion process could take several weeks, these approaches are among the first to have all aspects required to be considered a true twin.

The updatable structural model has proven popular beyond commercial vessels. A naval example is the Canadian experience with the Victoria/Upholder class diesel-electric submarines purchased from the United Kingdom. At some point in their service lives before purchase, these submarines suffered both denting and corrosion of their pressure hulls. This damage was significant enough that engineering analysis of each vessel was necessary to confirm if it was safe to return to service. Through a series of papers and reports, the development of a program known as SubSAS has been partially revealed [21][22]. This program appears to have built a forward-only twin where exact pressure hull measurements can be translated into non-linear finite element models and the collapse of the hull in its current state accurately modeled. Additional work covered modeling the impact of repairing some pressure hull areas with weld metal filler and experimental validation of the collapse behavior based upon scale experimental models.

The U.S. Navy proposed a system where inspection data and operational history would be merged to create ship-by-ship digital twins that could project future fatigue life and corrosion renewals as necessary [23]. Bureau Veritas has outlined an inspection approach for offshore structures that updates approval models, both hydrodynamic and FEA models, with in-service measurements. The unique model of each platform is then used to support service life extension decisions and future inspection requirements [24].

Canada has continued to lead in the development of marine digital twins. A broad scientometric study of intelligent maintenance approaches for military platforms examined twins among other approaches [25]. The scientometric study provides an extensive overview of the current literature around both digital twins and the more general condition-based maintenance world. Two figures from this report are reproduced below. Figure 4 shows a concept map generated from research article keywords around intelligent maintenance in the marine world. The map shows core concepts at the center, with branches dealing with detailed monitoring and data interpretation methods. Interestingly, the world of the composite structural health monitoring is completely separate from the other topics, and in the main group of the map, again, the structural monitoring approaches develop in a distinct region that is only loosely connected to the machinery monitoring topics. This further supports the idea that twins are building on existing approaches, approaches that have been developed with a degree of independence between disciplines.

Figure 5 expands upon Figure 4, presenting a cross-reference between different intelligence-based maintenance approaches and specific marine system components. The intensity of the connections supports the notion that structural and machinery approaches are the two largest active areas. Additionally, it appears that different keywords have developed in each area, with conditioned-based maintenance correlated more strongly with machinery systems and structural health monitoring corresponding to structural considerations. This table complements the conclusions drawn from Figure 4, providing more evidence that twins will be built upon existing discipline-specific approaches that have already achieved some degree of maturity. However, these approaches may not use the same language or underlying techniques. At the platform level, a twin may want to integrate concerns across these different domains. Such an approach would need to find a mapping between the different terminology and approaches used in each discipline.

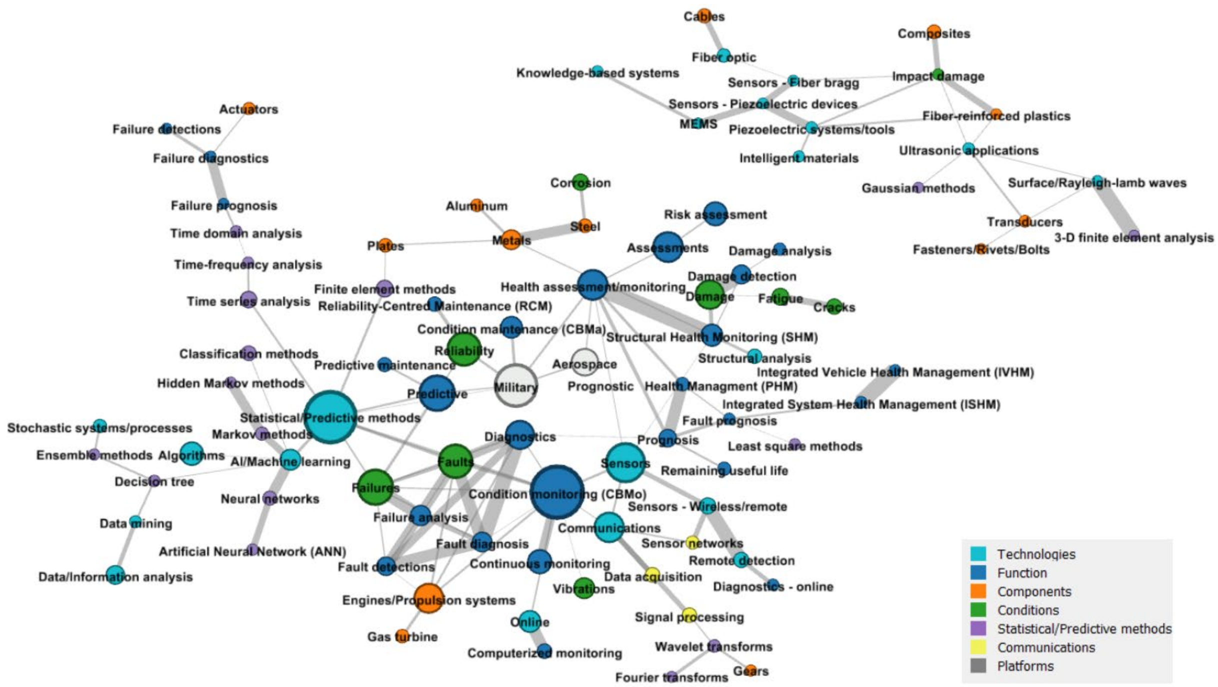


Figure 4: Concept Map for Marine Topics, Size of Nodes scales with Number of Publications, Size of Connector scales with Correlation. Developed by [25]

14

Topic	Engines/Propulsion systems	Machines	Metals	Oils	Fleet	Fuels	Hulls	Gases	Composites	Plates	Steel	Navigation	Gas turbine	Cables	Legacy assets	Critical components	Aluminum	Bearings	Actuators	Complex systems
Condition monitoring (CBMo)	61	41	19	33	6	14	9	11	6	4	8	11	6	11	3	5	3	10	2	4
Statistical/Predictive methods	45	26	21	16	6	13	18	9	6	13	4	8	7	4	9	8	6	3	3	2
Predictive	37	23	12	16	14	9	10	13	3	4	4	5	11	5	6	4	7	1	1	1
Sensors	30	20	19	20	8	8	4	8	7	7	7	10	5	10		4	3	4	7	2
Diagnostics	40	23	5	17	11	3		6	2	2	2	1	10	3	2	2		3	6	4
Assessments	15	12	16	12	8	6	12	5	5	6	9	2	2	3	11	2	4	2	2	4
Health assessment/monitoring	16	11	21	5	7	4	7	3	11	8	6	3	6	2	4	4	7	3	6	4
Communications	30	20	7	12	4	4	4	3	4	2	1	7	2	7	1	3	3	4	1	5
Preventative	22	16	7	10	4	9	4	7	1	1	2	2	7	2	4	3	2	2	1	2
Inspections	9	17	14	8	2	4	7	3	5	3	7	3	1	4	6		1		4	1
Simulation	16	8	10	6	1	5	4	5	4	8	4	2	3	3	5	4	3	2	3	2
Corrective	18	9	5	10	8	5	3	5	1		4	4	6	4	6	1	1	3		2
Structural Health Monitoring (SHM)	5	5	21	2		1	7	2	10	8	6	3	1	1	4	4	7	1	5	1
Integrated	13	14	9	6	8	3	4	3	4	2	2	4	1	3	3	4	2	2	2	2
Condition maintenance (CBMa)	22	12	1	6	10	8	3	7	2			1	4	1	4	2	1	1		1
Risk assessment	9	12	7	12	7	5	4	5	2	4	3	2	2	2		2	1			5
Accuracy	14	8	10	7	3	3	2	4	4	4	3	2	3	3	1	2		1	1	1
Online	17	16	4	12	3	2		4	3		1	2	1	4	2		1	2		1
Testing	11	9	14	3	1	2	4		4	6	7	3		1	2	1		4	1	
Real-time systems	10	12	4	7	2	3	3	5	4	3	1	5	1	3	1		1	3	3	1

Figure 5: Textual Keyword Publication Count for Intelligence Maintenance Papers in Marine Domain, Developed by [25]

Building on the concepts first introduced in the work of Pegg [10], Thompson [26] has proposed following a ship's service life with the twin, using weather hindcasts, recorded ship position, heading, and speed, with numerical seakeeping. Reasonable accuracy was achieved compared to full onboard measurements. Such an approach greatly reduces the monitoring cost, as no strain gauges need to be installed onboard the vessel. Thompson [27] has recently summarized the existing progress made in Canada into an overview report, outlining how the data gathered by the twin can be used for operational guidance, standards updating, and logistics support.

Most of the twins presented above would be categorized as forward-only, not reflective. This distinction is discussed in more detail in section 1.2.3, but briefly, a forward-only twin integrates real-world observations into an evolving digital model but does not try to correct the underlying digital model by comparing past predictions to current state information. Structural degradation updating is one area where underlying parameters (e.g., corrosion, fatigue crack growth parameters) are accessible. The current authors have worked to couple a variety of machine-learning techniques to make structural marine twins reflective, including work on updating wave loading and fatigue growth parameters [28]–[31]. Such reflective approaches have been extended by others to address inspection planning and long-term logistics [32], [33]. However, to date, most of these reflective twins are only using simple component-based failure models, not full FEA models of the vessel, and the inclusion of real-world structural data has not yet been attempted.

A related area where twin applications have been developed is that of course and speed guidance for operators. Typically, such guidance is provided on a much shorter timescale than structural twins, with guidance ranging from roughly ten days ahead to immediate support on the bridge for making course and heading changes. Perera and Guedes Soares [34] divide such approaches into two categories. The first category is weather routing, which they define as a pre-voyage plan for the vessel to minimize exposure to high sea states (to optimized time, fuel consumption, structural damage, or other variables of importance to the owner). Weather routing is typically done days in advance. The second category is what they refer to as safe ship handling, where the system provides near real-time guidance on speed and heading choices to the navigation team. Weather routing does not require a true twin – simply avoiding areas of rough weather can be accomplished without modeling the vessel itself, but both weather-routing and safe ship handling approaches now do include some methods that would be classified as twins as they use a ship-specific numerical model.

Huss and Olander [8] provided one of the earliest ideas for such a twin in 1994. Using the ship as a wave buoy, the vessel's current motions are used to approximate the sea state that the vessel is in. Then, hydrodynamic models are used to provide operator guidance and evaluate the probability of dangerous situations occurring. This is a complete twin, fusing motions with an inverse seakeeping model to determine wave parameters, then running those wave parameters through a forward model to predict motions and dangerous situations. Since then, this concept has been extensively extended. Nielsen has a notable body of work for large containerships, using this concept for operator guidance, structural fatigue damage rate prediction, and fault detection [35]–[38]. In a similar timeframe, submarine maneuvering digital twins were developed to provide feedback for autopilot systems and guidance on the impact of submarine maneuvering. These models make extensive use of online machine learning methods to update the underlying model as the submarine's condition change [39][40].

At a slightly longer timescale, work done by Dong et al. [41] and Nichols et al. [42] has explored using forward-only twins to provide optimal mission-level planning and routing decisions in a naval context. These twins use some sort of decision-making framework, forecast weather, and operational parameters such as speed and heading to give hours-to-days guidance on optimal routes. Perera and Guedes Soares review similar commercial concepts from the European research community [34].

SECTION 1.2.3 CATEGORIZING TWINS

Given the diversity of applications of digital twins encompassed by the twin definition, it is clear that a more descriptive way of categorizing subtypes of twins would aid in discussing and comparing twins. Several authors have proposed categorization systems for twins. Most notably in the marine field, Erikstad [9] has extended the concept of design patterns from object-oriented software programming to that of twins. Erikstad provides six primary patterns of twins based on the types of fusion logic, data, and purpose of the twin. Taylor et al. [43] use a variety of cyber-physical system architectures and comparisons to manufacturing-base twins to categorize twins and describe them. Fonseca and Gaspar [44] examine the twin primarily from a software and data storage perspective, proposing a layered architecture approach for understanding twins.

The above categorizations are useful at a high level to understand how twins are put together. However, in this work, a slightly more invasive categorization of twins is proposed. Based on the literature review, four primary axes emerge that separate twins into sub-categories. Each of these axes is a continuum, not a binary division between the respective endpoints. These provide another view for categorizing twins that will be used in reviewing the literature on fusion methods below.

1. **Volume of Data the Twin Uses:** Twins can be placed on an axis running from data-rich to data-poor. Data-rich twins typically get thousands or more data points directly related to their fusion approach and decision criteria. This opens up the ability to use a wide variety of machine learning approaches to build successful twins. Examples would include rotating machinery twins or seakeeping twins with many observations. Data-poor twins may not have a single observation of the target of their prediction. These twins must rely more heavily on the numerical models to predict un-observed physics, and their fusion will address the calibration of these models. Examples would include a twin guarding against structural collapse or capsize.

2. **Timescale of the Twin Fusion and Decision**: Twins can address problems on widely varying timescales. Seakeeping and stability-related twins may provide immediate guidance to the bridge watch team on heading and speed recommendations. Such twins must be able to integrate data, perform fusion, and make recommendations in near real-time. This implies very high autonomy and normally some sort of edge computing vs. cloud computing approach. On the other end of the spectrum, structural lifecycle twins may optimize fleet deployments and maintenance intervals over a decade-long time window. Such twins can fuse measurements, human inspection reports, and related data sources slowly, with extensive human involvement in the fusion process.

3. **Volume of State Information that Must Be Tracked**: Depending on their area of application, the physical state data that the twin needs to track over time can range widely. Detailed material failure models may be needed to understand the development of metallic microstructures, residual stresses, and other three-dimensional state variables throughout the structure. Battery models may need to understand the evolution of the chemistry in the battery over time. Long-term data management and validation is a key component of such twins, as the degradation of the system is usually path-dependent. However, other twins do not require such in-depth state vectors. Seakeeping and operability twins may only require the vessel's outer geometry and current mass properties, and prior operations need not be tracked at all to understand future motions.

4. **Forward or Reflective Twins**: Twins can be categorized based on their structure as well. Forward-only twins incorporate data from the physical object in the real world into a fixed numerical prediction model. This model then makes future predictions that are acted upon. An example of a forward-only twin would be a crack growth prediction model, which gets updated crack lengths from inspections, and then re-predicts future crack sizes using a crack propagation model. In this approach, the underlying numerical crack growth model is unchanging, though its input, starting crack size, varies. In a reflective twin, the twin compares the prediction it made to future observations and tries to reduce the error between prediction and observation by adjusting the numerical model. To return to the crack growth model, a dynamic Bayesian network modeling the same crack growth problem would study each observation and adjust the parameters of the crack growth model to give better agreement with all observations to date before predicting future crack size. This would be a reflective model. Another example would be the difference between using a machine learning model that is trained once and then

used to make future predictions and a machine learning model which continues to re-train itself with incoming observations. Forward or reflective remains a spectrum of possibilities, as the frequency of updating the underlying model can range from near-real-time to batch updates every few years.

Understanding where each twin application falls on these axes is critical to understanding what sort of technology challenges the twin will have to tackle to be successful. While these are not the only ways to differentiate twins, these four approaches directly address the type of fusion logic that would apply, the timescale of the computations and decisions, the data structures involved, and if the twin will refine itself over time. These four axes help compare twins – two twins with very different values on these axes would be expected to look very different internally, even if they both meet the twin definition discussed in section 1.2.1.

SECTION 1.2.4 SUMMARY

Digital twin is a broad concept. To focus the discussion on the marine community, we defined a canonical digital twin comprised of four different steps: a real-world physical object, one or more numerical models, fusion logic to link them, and input into a decision-making process. This definition is widely compatible with other definitions proposed to date. References to historical twins in the marine field were reviewed, and it was determined that research on systems meeting the modern definition of a twin was underway in the marine community a full decade before the term twin was first used. The marine community has a rich heritage in twins, with more than 25 significant papers reviewed above. While the definition of twin is helpful to scope what is or is not a twin, a four-category classification scheme was also proposed to further classify twins.

SECTION 1.3 POTENTIAL APPLICATIONS

We conducted a broad literature survey of the digital twin field, with a focus on marine applications. Scientific and popular articles were surveyed, as well as government publications. The survey looked beyond data-model fusion to capture all potential applications. A preference for either a direct marine application or a technology that could apply to a marine system was used to sort out twins from radically different fields (e.g., manufacturing real-time control). As these papers were found, a bottom-up affinity diagram approach was used to sort them into groups and themes. This approach was made in an iterative manner, with four top-level themes emerging, followed by 16 sub-themes under those higher-level themes. The first top-level theme was using a digital twin to monitor degradation for sustainment, repair, or to ensure safety. This was the most common theme. In the marine field, there were extensive examples of structural health monitoring and some machinery condition-based maintenance examples as well. Many condition-based maintenance systems for machinery are fairly robust (e.g., vibration measurement) and may not be showing up in recent publications as they are an established field of practice. Specific papers and topics within this subject are listed below:

1. Remaining life assessment of the structural system, accounting for corrosion, fatigue, and other deformations [3][27][45][46][29][47][48][10][49][50]

2. Inspection and repair scheduling and short-term integrity protection by detecting faults and modifying operations for both structural and mechanical systems [3][51]
3. Capturing true environmental load history of the platform to update inspection and maintenance [3]
4. Forecasting future mechanical and electrical systems condition [52] [53] [54][55][56] [57][58]
5. Manage spare parts, support additive manufacturing for some parts by assuring data availability [52]
6. Optimizing the timing of interventions/repairs for increased effectiveness [59] [55]

The second most frequent theme was using a digital twin to improve the operation and control of a vessel. Within this broad field, optimizing fuel consumption by controlling vessel trim, draft, and engine room parameters stood out as a particularly active field, especially in commercial service. Simulating future missions was also common, especially to handle damage onboard or to predict how the vessel will respond in conditions that it has not yet encountered. Finally, another cluster of papers described twins that provide real-time operational guidance to crew members around maneuvering the vessel in waves and safety hazards. These systems gave guidance about potential changes in heading, as well as providing operators with visibility into the rate of structural fatigue damage occurring on the hull. Such damage accumulation is not visible to human crews and hence is not always considered when mission planning or maneuvering a vessel. Specific papers within this theme are listed below:

7. Updating control systems based on in-service experience for system or overall vehicle dynamics [3][60][39]
8. Predicting future performance/mission simulation, including modifications for damage or degradation, or improved predictions in un-observed conditions [3][61][62][28][63]
9. Providing real-time feedback to the vessel operator on safety or cumulative damage (e.g., fatigue) [27] [54] [61]
10. Optimizing fuel consumption, system configuration, and validating system performance [59][64] [54][56][65][66][67][68]
11. Demonstrate and document performance for authorities [59]

One of the roots of digital twins is in product lifecycle management (PLM) systems, which integrate 3-D geometry models, part and component information, and other models into a single repository over the asset's lifetime. Such PLM approaches composed the third major theme of digital twin applications. In this theme, the majority of the papers dealt with updating design codes and standards based on in-service feedback from observations. Related papers dealt with design verification, planning alternations, and serving as a single model of the system through life. These areas mainly had one or two papers in them. However, much like condition-based maintenance in the first theme, a dedicated search for PLM systems would undoubtedly turn up many more papers. This is especially true given that Grieves [7] identified twins as the natural extension of PLM systems.

12. Updating design codes, uncertainty, and standards-based on actual performance in the ocean [3] [27][54][69]
13. Planning for alternations and modifications [27][70]
14. Checking I.T. systems before installation [71]

15. Providing a single, consistent representational model of the system for all stakeholders [5]

The fourth theme explored was that of fleet optimization, which goes beyond the work in optimizing a single vessel to look at optimal employments of fleets of vessels. This could be to match the characteristics of particular vessels with forecast conditions to optimize the ability to complete a mission. Alternatively, a longer-term twin application could look to balance fatigue exposure over a class of similar vessels to maximize asset lifetime. This theme had far fewer papers and tended to be more press-release-style papers about future system possibilities than deep academic papers. Overall, this theme appeared to be the least developed, and it could be argued that this theme is really another decision layer that could simply be layered on the outputs from vessel-level twins. Given the development still needed at the vessel level, this state is not surprising.

16. Optimize deployments and use of fleet of assets or maintenance [23][3][72] [54]

SECTION 1.4 LITERATURE SEARCH AND ASSESSMENT OF CURRENT FUSION TECHNIQUES

A more detailed literature review was made of 32 papers focused on the fusion step of the twin process. The fusion step is one of the more novel steps in the twin process and is essential to tie the virtual and real platforms together. Each paper was read, and a brief summary of the fusion approach was noted in an annotated bibliography format. These entries are contained in Appendix A at the end of the document. A matrix of criteria was used to rank the fusion method in each paper. These criteria consisted of the following terms:

1. The year the paper was published
2. The four criteria introduce in section 1.2.3:
    - Whether the twin was data-rich or data-poor. Three broad division points were used here, data-rich twins were those that used more than 1,000 data points (not 1,000 individual data streams, just more than 1,000 observations across all data streams), data-poor twins used ten or fewer, and all other twins were ranked as intermediate
    - The timescale of the decision, divided into immediate decisions, decisions over a days-to-weeks timeframe, and decisions over a months-to-years timeframe
    - State information used, which was ranked by stateless systems (e.g., no history kept), recent history used for comparison or updating, or complete history through manufacturing
    - Whether the twin was forward or reflective
3. An additional list of characteristics that evolved as the papers were reviewed:
    - Which types of data sources were used?
    - The number of individual data streams
    - Which types of numerical models were used?
    - The field of application
    - The level of development
    - Which mathematical fusion approach was used?
    - What type of system was the fusion method applied to?

4. A survey of which components was included in each twin:
   - Did it discuss the sensing step in detail?
   - Did it discuss the fusion step in detail?
   - Did it discuss the decision step in detail?
   - Did it include a hierarchy of one or more systems or models?
   - Did it include uncertainty in the approach?

The publication count by year is given in Figure 6, which shows increasing research intensity in the last five years, with a sudden jump in 2020. This tracks closely with results from Collette and Wincott [1], which showed a growing interest in twins in the popular literature in the early 2000s, followed by a much more recent and exponential uptick in scholarly publications in this area. This again shows the relative novelty of the field and the high rate of current research in this area.

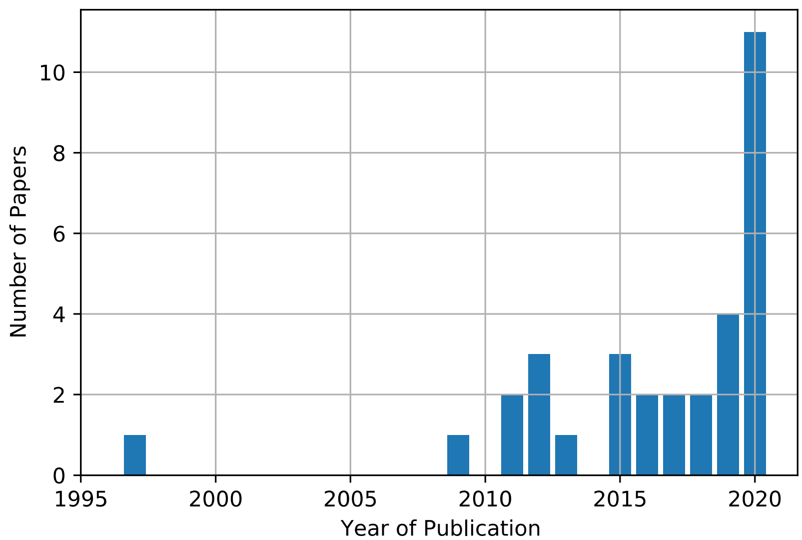


Figure 6: Publication Count by Year

In terms of the four twin criteria introduced in this paper, the results showed that many fusion-oriented twins straddled boundaries between the categories proposed, suggesting that these boundaries may be relatively flexible. This means that the total of the observations frequently exceeded 32 -the numbers of papers reviewed, as some methods applied across categories. In terms of the data-rich and data-poor twins, there was a clear bias towards data-rich fusion methods, with 20 papers addressing approaches where the data points available exceeded 1,000, with eight papers in the intermediate state and only four papers in the data-poor category with fewer than 10 data points. The current state of the field reflects perhaps related developments in machine learning, which has largely prioritized developments of data-rich black-box modeling approaches, such as deep learning via neural networks. It may also reflect the spillover from condition-based maintenance of rotating machinery systems, which generally feature continuous measurements.

The timescale of the decision was more varied, with 16 addressing immediate decisions, 22 addressing days to weeks decisions, and 14 addressing months to years. This last category had the fewest "unique" entries – the overwhelming majority of systems that provided long-term guidance also were able to provide the days-to-weeks guidance as well.

The state information was much more clustered. The majority of the papers (20) used no state information. This was representative of trained machine learning models that simply looked for differences in the input signal momentarily, without making time comparisons. Ten methods used recent history, while only two integrated data from manufacturing. This indicates that full integration between manufacturing and design-oriented digital thread approaches and fusion in digital twins is still somewhat lacking. In part, this may be a result of the difficulty in replicating a full digital thread in a research environment, making exploration of this aspect of fusion difficult.

Surprisingly, of the 32 papers reviewed, 25 used the more complex reflective twin approach, where fusion attempts to identify and reduce errors in future predictions, not simply update the state information. Seven approaches used fusion in a forward-only sense, focusing on state updates. Overall, the analysis of the four criteria points to a varied landscape for digital twins. This both emphasizes the importance of having a good taxonomy to understand twins and the amount that the underlying physics of the problem to be solved influence the form and functional design of the fusion method and wider twin. The analysis of the fusion papers shows that reflective twins of varying decision timescales are now widely being explored. The analysis also shows that data-poor and state-rich twins are perhaps underexplored corners of the fusion and twin landscape.

The broader review of twin characteristics was also illuminating. Data stream diversity was very low; 31 papers used simple time-histories of numerical readings (e.g., strain, motion, acceleration). Three papers used human-developed soft input, and three used image-based inputs. The number of individual input channels (e.g., a single sensor that may produce many readings) was also tracked and is shown in Figure 7. The majority of fusion approaches today are focused on a relatively low number of inputs, with five or fewer the most common and the majority of systems having fewer than ten input channels. However, there is clearly interest in larger systems, as the largest paper looked at more than fifty channels, and seven looked at using more than 20 channels. From this analysis, it is clear that fusion systems today are dominated by low numbers of simple data sources, but there is interest in broadening both the types of inputs and the number of inputs.

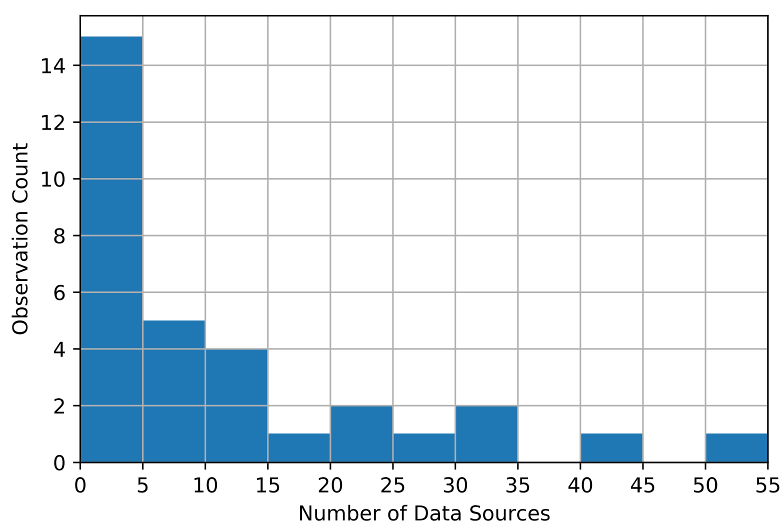


Figure 7: Count of Number of Papers vs. Number of Individual Data Sources Used in Fusion

The types of models updated by the fusion method also varied. By far, the most common approach was to have a physics-based model of a single component, 22 of the papers used this approach. The next most common approach was constructing an abstraction – which could be a machine learning model, or a network-based model, to handle the fusion tasks. This is a different approach as it does not try to replicate the physics of the problem directly during fusion but instead functions as an independent predictor of the future responses or adds a correcting step to the output of the physics-based model. Only four papers used 3-D CAD models of the system that could attempt to act as a PLM or central repository of geometry, models, and data through-life. This again shows that while tying twins and fusion approaches into digital threads from manufacturing onwards has been discussed, it is still relatively rare to see this occur in the research field to date.

In terms of fields of application, the naval domain was the best represented, with thirteen papers addressing marine-focused systems. Eight addressed aerospace systems, while civil engineering structures had six systems, with no other domain having more than one system. This indicates that at the moment, mobile platforms are dominating the twin discussion, and this focus does line up well with the historical interest in this approach from the marine domain as well as both the U.S. Air Force and NASA. The underlying physics involved in each model also varied. Structural concerns were the most commonly referenced with 20 papers on this topic, followed by six addressing fluid-related performance, three for mechanical components, and two dealing with general dynamic responses, and one focused on an electrical system. This structural and fluid focus also tracks with the marine, aero, and civil fields being the most common fields seen.

The fusion technologies used were also diverse. Regression-type approaches were the most common, with ten methods overall using these, including six using artificial neural networks and four using more basic fitting and regression approaches. After this, Bayesian networks were the next most common, with eight papers using them and another three using simpler Bayesian updating of variables. The large amount of Bayesian networks seen were surprising, given that these approaches can struggle with large amounts of data. Two methods used simple, direct updating from observations. There were another eight methods proposed, including bond graphs, decision trees, and methods that combined multiple approaches in their fusion technique, but not more than one paper used each of these. The range of methods encountered is likely a function of the diversity in application domains as well as the difference across the four categories identified in section 1.2.3 The range of methods underlines that a toolbox of techniques will be required to span the physics and twin requirements of modern military platforms.

The maturity of the methods was skewed towards early-stage research. Twenty-three papers were from purely academic explorations, including thirteen that used experiments and ten that only used synthetic data. A further six had a prototype in the real world, and three described fully operational systems. This tracks with the idea that digital twins are relatively new, and hence fusion methods are currently being matured. However, the increased indexing and availability of academic papers could also have influenced this result, with commercial or governmental agencies less likely to publish descriptions of their twins as frequently.

Finally, while the focus of this literature review was on the fusion step in the twin process, the papers were reviewed to see how the current fusion literature interacts with the rest of the proposed twin systems. From this analysis, the dominant conclusion is that the fusion literature is not particularly well integrated with these system-level concerns. Of the 32 papers, only six dealt with decision-making from the twin, another six dealt with uncertainty, and two dealt with a hierarchy of multiple components in the fusion step. This is compatible with the academic focus of the existing literature, which is more likely to explore distinct components of the fusion process in isolation for each paper. However, similar to the lack of integration of digital thread approaches, these areas will need to be explored to fully realize twin systems for real-world applications.

The literature review confirmed several aspects of the twin model. First, the four different ways of categorizing twins proved useful in practice when looking at fusion methods. Second, there is a clear need for a variety of fusion approaches. The need for variety is driven by the variety in applications, important physics, categorization, and numerical techniques fusion methods must address. Third, the field is relatively immature, with an academic focus on smaller problems and lower numbers of data sources, and without evidence of extensive integration with digital thread approaches, large system problems, or the wider issues of decision making and uncertainty.

SECTION 1.5 ANALYSIS OF GAPS AND NEEDS

Based on the analysis in the preceding three sections, there is a clear need for future developments in both twin systems and, more specifically, in the fusion step. Four major need areas were identified: monitoring degradation for maintenance and logistics, improving operational capability, supporting design, and optimizing fleet utilization. All of these twin areas are active in the research community. Additionally, these areas are expected to become even more important as crewless platforms become more common, and automated computational systems have to take the place of human crews for mission performance and maintenance concerns. Furthermore, the review of the types of twins and fusion techniques published to date shows that a variety of internal calculations and algorithmic approaches are needed as the characteristics of the physical systems change. This was shown in the differences between data-rich and data-poor twins, differing levels of state information, differing decision timescales, and the growing desire to take advantage of the more complex reflective fusion approaches. Overall, the current situation indicates that further research into fusion algorithms is important so that future twins can support the community's needs.

Perhaps the most obvious gap is the need for a deeper understanding of the potential algorithmic approaches to fusion. As noted in section 1.4 the reviewed fusion techniques used a wide variety of algorithmic formulations. While neural networks and Bayesian nets are the most popular, the best way of applying these techniques to marine fusion problems is not yet clear. Internal research at Michigan is currently looking into basic formulation questions for both of these techniques. However, it is clear that the understanding around each of these methods is not mature enough to give near-optimal performance when applied to new problems. Additionally, a range of different techniques beyond neural networks and Bayesian networks were identified in section 1.4. These methods have appeal for specific fusion problems but are even less understood in the context of fusion than the neural networks and Bayesian approaches.

A second clear gap is in the ability of current fusion approaches to handle larger systems. Here, there are several clear areas where "larger" systems are relatively unexplored. The raw number of data inputs explored to date tends lower to fewer than ten input streams. While such data density may be able to capture component responses, it is likely to be insufficient to capture all of the relevant variables for even sub-systems on larger platforms. A second approach, that of building a higher-level fusion approach to integrate many component-level twins, could also be used to solve this problem. However, this hierarchal approach was even rarer in the literature reviewed to date. Integration with decision-making was also shown to still be lacking in the literature, and applications of manufacture-to-operation digital thread approaches are also scarce. Overall, as fusion approaches move from individual components, it is clear that their maturity decreases rapidly, and more research is needed in these areas.

Uncertainty and reliability aspects of fusion were also sparsely discussed in the literature. As twins start to support decision-making, the level of confidence in the fused result is an important output of the fusion method. This becomes even more important in the context of crewless systems, where the decision loop maybe fully automated without a human crew member evaluating the recommended course of action. Short timescale decisions are especially important in this regard, as relying on off-vessel help may not be possible. While such concepts have been discussed, few fusion approaches proposed algorithmic solutions for this part of the twin problem. Related to integrating uncertainty and reliability is the need for decisions where multiple twins could be used – e.g., a higher fidelity model and a lower fidelity model. Having the decision process switch between models based on uncertainty and accuracy needs or when data is not available for certain models is also unexplored.

While the focus in this report was on fusion methods, it is also clear that there is an overall low level of integration of fusion methods into end-to-end digital twins. The literature search revealed a fairly low crossover between fusion research and decision-making in general. Likewise, integration into digital thread approaches or other whole-of-life data models were lacking. More work in demonstrating end-to-end twins would be helpful for this aspect of the work.

SECTION 1.6 CONCLUSION

Digital twins represent a new capability to use numerical engineering tools in operational support. Twins today are built on several existing areas of research, such as condition-based maintenance and structural health monitoring. The development of digital twins from these underlying technologies has spanned nearly thirty years at this point, with the initial concept of what would be recognized as a twin today being developed in the early 1990s. However, twins have their own unique structure, and a formal definition for such twins was proposed that integrates much of the existing literature and previous definitions.

There is a strong pull for the capabilities of twins today. Four major areas of need and development were documented in the community. The first is to use the twin to improve maintenance, safety, and logistics of degrading systems such as mechanical or structural systems. The second area was to improve in-service performance through enhanced modeling and control of the platform. The third was to serve as a unified lifecycle model of the platform and provide feedback from the platform to update design standards and approaches. The fourth was to perform optimization over a fleet of similar vessels. The first two of these areas have seen more publications and interest to date than the second two.

A key challenge in digital twins is the need to fuse the numerical models with real-world measurement and data. A comprehensive review of fusion approaches through 32 existing papers on fusion methods revealed that the existing fusion methods are well suited to component-level problems, but extensions to larger problems, system-level hierarchies, or considerations around uncertainty and reliability are largely unexplored. Additionally, the algorithmic formulation of the fusion steps varies widely depending on the physics involved and the types of data available. A broad understanding of optimal or general formulations for this problem is not currently available. Overall, while twins hold promise for improving operational capability, reliability, and safety, more research work is needed in the above-mentioned areas before twins can reach their full potential.

SECTION 1.7 REFERENCES

[1] C. Wincott and M. D. Collette, "Digital Twins: An Assessment of the State-of-the-Art," presented at the ASNE TSS, Washington D.C., Jun. 2019.

[2] "A standardized definition and preliminary taxonomy for digital twins (DRAFT)," Martin Defense Group, Honolulu, Hawaii, USA, Oct. 2020.

[3] S. Erikstad, "Merging Physics, Big Data Analytics and Simulation for the Next-Generation Digital Twins," Zevenwacht, South Africa, Sep. 2017, pp. 139–149.

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

[5] E. M. Kraft, "The Air Force Digital Thread/Digital Twin - Life Cycle Integration and Use of Computational and Experimental Knowledge," presented at the 54th AIAA Aerospace Sciences Meeting, San Diego, California, USA, Jan. 2016. doi: 10.2514/6.2016-0897.

[6] E. Negri, L. Fumagalli, and M. Macchi, "A Review of the Roles of Digital Twin in CPS-based Production Systems," *Procedia Manufacturing*, vol. 11, pp. 939–948, 2017, doi: 10.1016/j.promfg.2017.07.198.

[7] M. Grieves, "Virtually Perfect: Driving Innovative and Lean Prod- ucts through Product Lifecycle Management." Space Coast Press, 2011.

[8] M. Huss and A. Olander, "Theoretical Seakeeping Predictions On Board Ships – A System for Operational Guidance and Real Time Surveillance," Naval Architecture, Royal Institute of Technology, Stockholm, Sweden, ISRN KTH/FKT/SKP/FR--94/50--S.E., 1994.

[9] S. Erikstad, "Design Patterns for Digital Twin Solutions in Marine Systems Design and Operations," May 2018.

[10] N. G. Pegg and S. Gibson, "Application of Advanced Analysis Methods to the Life Cycle Management of Ship Structures," Defence Research Establishment Atlantic, Dartmouth, Nova Scotia, Canada, May 1997.

[11] E. Glaessgen and D. Stargel, "The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2012-1818.

[12] J. Guo, Z. Li, and M. Li, "A Review on Prognostics Methods for Engineering Systems," *IEEE Transactions on Reliability*, vol. 69, no. 3, pp. 1110–1129, Sep. 2020, doi: 10.1109/TR.2019.2957965.

[13] N.-H. Kim, D. An, and J.-H. Choi, *Prognostics and Health Management of Engineering Systems*. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-44742-1.

[14] S. Alaswad and Y. Xiang, "A review on condition-based maintenance optimization models for stochastically deteriorating system," *Reliability Engineering & System Safety*, vol. 157, pp. 54–63, Jan. 2017, doi: 10.1016/j.ress.2016.08.009.

[15] M. Abbas and M. Shafiee, "An overview of maintenance management strategies for corroded steel structures in extreme marine environments," *Marine Structures*, vol. 71, p. 102718, May 2020, doi: 10.1016/j.marstruc.2020.102718.

[16] A. L. Ellefsen, V. Æsøy, S. Ushakov, and H. Zhang, "A Comprehensive Survey of Prognostics and Health Management Based on Deep Learning for Autonomous Ships," *IEEE Transactions on Reliability*, vol. 68, no. 2, pp. 720–740, Jun. 2019, doi: 10.1109/TR.2019.2907402.

[17] E. Artigao, S. Martín-Martínez, A. Honrubia-Escribano, and E. Gómez-Lázaro, "Wind turbine reliability: A comprehensive review towards effective condition monitoring development," *Applied Energy*, vol. 228, pp. 1569–1583, Oct. 2018, doi: 10.1016/j.apenergy.2018.07.037.

[18] G. de N. P. Leite, A. M. Araújo, and P. A. C. Rosas, "Prognostic techniques applied to maintenance of wind turbines: a concise and specific review," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1917–1925, Jan. 2018, doi: 10.1016/j.rser.2017.06.002.

[19] F. Cipollini, L. Oneto, A. Coraddu, A. J. Murphy, and D. Anguita, "Condition-Based Maintenance of Naval Propulsion Systems with supervised Data Analysis," *Ocean Engineering*, vol. 149, pp. 268–278, Feb. 2018, doi: 10.1016/j.oceaneng.2017.12.002.

[20] Anon, "ABS Introduces SafeShip," *Maritime Reporter and Engineering News*, p. 12, Jul. 2000.

[21] M. J. Smith, T. Macadam, and J. R. MacKay, "Integrated modelling, design and analysis of submarine structures," *Ships and Offshore Structures*, vol. 10, no. 4, pp. 349–366, Jul. 2015, doi: 10.1080/17445302.2014.937058.

[22] J. R. MacKay, M. J. Smith, and N. G. Pegg, "Design of Pressure Hulls Using Nonlinear Finite Element Analysis," Oct. 2008, pp. 843–851. doi: 10.1115/OMAE2006-92591.

[23] T. J. Eccles, G. Ashe, and S. Albrecht, "The Achieving Service Life Program," *Naval Engineers Journal*, vol. 122, no. 3, pp. 103–112, Sep. 2010, doi: 10.1111/j.1559-3584.2010.00275.x.

[24] J. Boutrot, Y. Giorgiutti, F. Rezende, and S. Barras, "Reliable and Accurate Determination of Life Extension for Offshore Units," presented at the Offshore Technology Conference, May 2017. doi: 10.4043/27547-MS.

[25] M. Lethiecq-Normand and R. Jansen, "Scientometric Study on Intelligent Maintenance Systems for Military Platforms," DRDC – Atlantic Research Centre, DRDC-RDDC-2018-C122, Jun. 2018.

[26] I. Thompson, "Virtual hull monitoring of a naval vessel using hindcast data and reconstructed 2-D wave spectra," *Marine Structures*, vol. 71, p. 102730, May 2020, doi: 10.1016/j.marstruc.2020.102730.

[27] I. Thompson, "Digital twinning of ship structural fatigue: state of the art review and strategic research agenda," DRDC - Atlantic Research Center, DRDC-RDDC-2019-R099, Jul. 2019.

[28] J. Zhu and M. Collette, "A Bayesian approach for shipboard lifetime wave load spectrum updating," *Structure and Infrastructure Engineering*, vol. 13, no. 2, pp. 298–312, Feb. 2017, doi: 10.1080/15732479.2016.1165709.

[29] M. Groden and M. Collette, "Fusing fleet in-service measurements using Bayesian networks," *Marine Structures*, vol. 54, no. Supplement C, pp. 38–49, Jul. 2017, doi: 10.1016/j.marstruc.2017.03.001.

[30] J. Zhu and M. Collette, "Updating Structural Engineering Models with In-service Data: Approaches and Implications for the Naval Community," *Naval Engineers Journal*, vol. 127, no. 1, pp. 63–74, Mar. 2015.

[31] M. Schirmann, M. Collette, and J. Gose, "Improved Vessel Motion Predictions using Full-Scale Measurements and Data-Driven Models," presented at the 33rd Symposium on Naval Hydrodynamics, Osaka, Japan, Oct. 2020.

[32] D. Y. Yang and D. M. Frangopol, "Evidence-based framework for real-time lifecycle management of fatigue-critical details of structures," *Structure and Infrastructure Engineering*, vol. 14, no. 5, pp. 509–522, May 2018, doi: 10.1080/15732479.2017.1399150.

[33] D. Y. Yang and D. M. Frangopol, "Probabilistic optimization framework for inspection/repair planning of fatigue-critical details using dynamic Bayesian networks," *Computers & Structures*, vol. 198, pp. 40–50, Mar. 2018, doi: 10.1016/j.compstruc.2018.01.006.

[34] L. P. Perera and C. G. Soares, "Weather routing and safe ship handling in the future of shipping," *Ocean Engineering*, vol. 130, pp. 684–695, Jan. 2017, doi: 10.1016/j.oceaneng.2016.09.007.

[35] U. D. Nielsen, "A concise account of techniques available for shipboard sea state estimation," *Ocean Engineering*, vol. 129, pp. 352–362, Jan. 2017, doi: 10.1016/j.oceaneng.2016.11.035.

[36] U. D. Nielsen and J. J. Jensen, "A novel approach for navigational guidance of ships using onboard monitoring systems," *Ocean Engineering*, vol. 38, no. 2, pp. 444–455, Feb. 2011, doi: 10.1016/j.oceaneng.2010.11.024.

[37] U. D. Nielsen, J. J. Jensen, P. T. Pedersen, and Y. Ito, "Onboard monitoring of fatigue damage rates in the hull girder," *Marine Structures*, vol. 24, no. 2, pp. 182–206, Jun. 2011, doi: 10.1016/j.marstruc.2011.03.003.

[38] U. D. Nielsen, Z. Lajic, and J. J. Jensen, "Towards fault-tolerant decision support systems for ship operator guidance," *Reliability Engineering & System Safety*, vol. 104, pp. 1–14, Aug. 2012, doi: 10.1016/j.ress.2012.04.009.

[39] L. Jiang *et al.*, "A Hydrodynamic Digital Twin Concept for Underwater Vehicles," presented at the 33rd Symposium on Naval Hydrodynamics, Osaka, Japan, Oct. 2020.

[40] W. Faller, D. Hess, T. Fu, and E. Ammeen, "Analytic Redundancy for Automatic Control Systems: Recursive Neural Network Based Virtual Sensors," in *45th AIAA Aerospace Sciences*

Meeting and Exhibit, American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2007-156.

[41] Y. Dong, D. M. Frangopol, and S. Sabatino, "A decision support system for mission-based ship routing considering multiple performance criteria," *Reliability Engineering & System Safety*, vol. 150, pp. 190–201, Jun. 2016, doi: 10.1016/j.ress.2016.02.002.

[42] J. M. Nichols, P. L. Fackler, K. Pacifici, K. D. Murphy, and J. D. Nichols, "Reducing fatigue damage for ships in transit through structured decision making," *Marine Structures*, vol. 38, pp. 18–43, Oct. 2014, doi: 10.1016/j.marstruc.2014.04.002.

[43] N. Taylor, C. Human, K. Kruger, A. Bekker, and A. Basson, "Comparison of Digital Twin Development in Manufacturing and Maritime Domains," in *Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future*, Cham, 2020, pp. 158–170. doi: 10.1007/978-3-030-27477-1_12.

[44] Í. A. Fonseca and H. M. Gaspar, "Challenges when creating a cohesive digital twin ship: a data modelling perspective," *Ship Technology Research*, vol. 0, no. 0, pp. 1–14, Sep. 2020, doi: 10.1080/09377255.2020.1815140.

[45] Y. Bazilevs, X. Deng, A. Korobenko, F. Lanza di Scalea, M. D. Todd, and S. G. Taylor, "Isogeometric Fatigue Damage Prediction in Large-Scale Composite Structures Driven by Dynamic Sensor Data," *Journal of Applied Mechanics*, vol. 82, no. 9, p. 091008, Sep. 2015, doi: 10.1115/1.4030795.

[46] A. Dourado and F. A. C. Viana, "Physics-Informed Neural Networks for Missing Physics Estimation in Cumulative Damage Models: A Case Study in Corrosion Fatigue," *Journal of Computing and Information Science in Engineering*, vol. 20, no. 061007, Jun. 2020, doi: 10.1115/1.4047173.

[47] J. Luque and D. Straub, "Reliability analysis and updating of deteriorating systems with dynamic Bayesian networks," *Structural Safety*, vol. 62, pp. 34–46, Sep. 2016, doi: 10.1016/j.strusafe.2016.03.004.

[48] T. Magoga, S. Aksu, S. Cannon, R. Ojeda, and G. Thomas, "Through-life hybrid fatigue assessment of naval ships," *Ships and Offshore Structures*, vol. 0, no. 0, pp. 1–11, Jan. 2019, doi: 10.1080/17445302.2018.1550900.

[49] U. T. Tygesen, K. Worden, T. Rogers, G. Manson, and E. J. Cross, "State-of-the-Art and Future Directions for Predictive Modelling of Offshore Structure Dynamics Using Machine Learning," in *Dynamics of Civil Structures, Volume 2*, Cham, 2019, pp. 223–233. doi: 10.1007/978-3-319-74421-6_30.

[50] M. A. Vega, Z. Hu, and M. D. Todd, "Optimal maintenance decisions for deteriorating quoin blocks in miter gates subject to uncertainty in the condition rating protocol," *Reliability Engineering & System Safety*, vol. 204, p. 107147, Dec. 2020, doi: 10.1016/j.ress.2020.107147.

[51] H. Zhang and Y. Deng, "Weighted belief function of sensor data fusion in engine fault diagnosis," *Soft Comput*, vol. 24, no. 3, pp. 2329–2339, Feb. 2020, doi: 10.1007/s00500-019-04063-7.

[52] M. Debbink and C. Coleman, "Strategy for an Intelligent Digital Twin (IDT)," presented at the NIST Model Based Enterprise (MBE) Summit 2019, 2019.

[53] "ABS Recognizes G.E.'s SeaStream* Insight Data-Driven Techniques for Marine and Drilling Equipment Maintenance and Class Surveys." https://ww2.eagle.org/en/news/press-room/abs-recognizes-ge-seastream-insight.html (accessed Dec. 23, 2020).

[54] "Does your ship need a digital double?," *Wartsila.com*. https://www.wartsila.com/insights/article/does-your-ship-need-a-digital-double (accessed Dec. 23, 2020).

[55] "G.E. and Maersk Drilling to Pilot Marine Digital Transformation | G.E. News." http://www.ge.com/news/press-releases/ge-and-maersk-drilling-pilot-marine-digital-transformation (accessed Dec. 23, 2020).

[56] "Your systems may be optimized but digital twins could learn to do it better." https://new.abb.com/news/detail/24663/your-systems-may-be-optimized-but-digital-twins-could-learn-to-do-it-better (accessed Dec. 23, 2020).

[57] J. Liu, W. Wang, F. Ma, Y. B. Yang, and C. S. Yang, "A data-model-fusion prognostic framework for dynamic system state forecasting," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 4, pp. 814–823, Jun. 2012, doi: 10.1016/j.engappai.2012.02.015.

[58] S. Burmaster, R. Brown, D. Maraini, and M. Simpson, "A Renewed Digital, Data-Driven Approach to Condition-Based Maintenance for United States Navy Hull, Mechanical, and Electrical Equipment," presented at the ASNE Intelligent Ship Symposium, Philadelphia, PA, 2019.

[59] L. Kristine, "Digital Twins for Blue Denmark," DNG GL Maritime, 2018–0006, Rev. A, Jan. 2018.

[60] A. Behjat, C. Zeng, R. Rai, I. Matei, D. Doermann, and S. Chowdhury, "A physics-aware learning architecture with input transfer networks for predictive modeling," *Applied Soft Computing*, vol. 96, p. 106665, Nov. 2020, doi: 10.1016/j.asoc.2020.106665.

[61] M. G. Kapteyn, D. J. Knezevic, D. B. P. Huynh, M. Tran, and K. E. Willcox, "Data-driven physics-based digital twins via a library of component-based reduced-order models," *International Journal for Numerical Methods in Engineering*, vol. n/a, no. n/a, 2020, doi: https://doi.org/10.1002/nme.6423.

[62] A. Mondoro, M. Soliman, and D. M. Frangopol, "Prediction of structural response of naval vessels based on available structural health monitoring data," *Ocean Engineering*, vol. 125, pp. 295–307, Oct. 2016, doi: 10.1016/j.oceaneng.2016.08.012.

[63] G. D. Weymouth and D. K. P. Yue, "Physics-Based Learning Models for Ship Hydrodynamics," *Journal of Ship Research*, vol. 57, no. 01, pp. 1–12, Mar. 2013.

[64] "Digital twin developed to model green ship technology," *Riviera*. https://www.rivieramm.com/news-content-hub/news-content-hub/digital-twin-developed-to-model-green-ship-technologynbsp-59419 (accessed Dec. 23, 2020).

[65] I. Brahma, "Extending the Range of Data-Based Empirical Models Used for Diesel Engine Calibration by Using Physics to Transform Feature Space," *SAE Int. J. Engines*, vol. 12, no. 2, pp. 03-12-02–0014, Mar. 2019, doi: 10.4271/03-12-02-0014.

30

[66] A. Coraddu, L. Oneto, F. Baldi, F. Cipollini, M. Atlar, and S. Savio, "Data-driven ship digital twin for estimating the speed loss caused by the marine fouling," *Ocean Engineering*, vol. 186, p. 106063, Aug. 2019, doi: 10.1016/j.oceaneng.2019.05.045.

[67] S. V. Hansen, "Performance Monitoring of Ships," PhD Thesis, DTU, Lyngby, Denmark, 2011.

[68] “Miros Mocean,” *Miros Mocean*. https://www.mirosmocean.com/ (accessed Mar. 06, 2021).

[69] I. Thompson, "Validation of naval vessel spectral fatigue analysis using full-scale measurements," *Marine Structures*, vol. 49, pp. 256–268, Sep. 2016, doi: 10.1016/j.marstruc.2016.05.006.

[70] "'Digital Twin' Modeling Guides U.S. Navy's $21B Shipyard Plan," *The Maritime Executive*. https://www.maritime-executive.com/article/digital-twin-modeling-guides-u-s-navy-s-21b-shipyard-plan (accessed Dec. 23, 2020).

[71] "Navy using 'digital twins' to speed innovation to the fleet.," *Federal News Network*, May 14, 2020. https://federalnewsnetwork.com/federal-insights/2020/05/navy-using-digital-twins-to-speed-innovation-to-the-fleet/ (accessed Dec. 23, 2020).

[72] "Digital Twins for the Maritime Sector," *The Maritime Executive*. https://www.maritime-executive.com/editorials/digital-twins-for-the-maritime-sector (accessed Dec. 23, 2020).

CHAPTER 2 A STANDARDIZED DEFINITION AND PRELIMINARY TAXONOMY FOR DIGITAL TWINS

Authors: Taylor Smith[1], Matthew Collette[2], Conner Goodrum[1], Mike Sypniewski[1]
1 – Martin Defense Group
2 – University of Michigan, Department of Naval Architecture and Marine Engineering
Date: July 2021

SECTION 2.1 MOTIVATION & SCOPE

For the purposes of maintaining consistency within this study, we established a standardized definition of digital twins. We use this definition as the basis for our preliminary taxonomy, which describes the components (systems, sensors, simulations, inferences, etc.) and relationships that comprise digital twins. Within this taxonomy we explored three vital areas of practice: data acquisition, data integration, and systems models.

SECTION 2.2 STANDARDIZED DEFINITION AND PROPOSED TAXONOMY

In formalizing a definition for digital twins, we begin with the following from Collette:

> *"For naval applications, we define a twin as a system that includes* [the] *following four components: (1) a real-world system of interest, (2) one or more digital models of this system, (3) sensor data plus fusion logic to join 1 & 2, and (4) a decision that will change based on twin output." [25]*

This was used, alongside observations from the literature review as discussed in the Chapter 1 Survey of Fusion Approaches and Opportunities, to develop the following set of conditions. A digital twin requires:

Condition 1 (C1): A specific, real-world component, system, or process.

Condition 2 (C2): One or more digital representations of **C1.**

Condition 3 (C3): Mechanisms for acquiring and integrating data such that **C2** reflects **C1** over time.

These conditions provide a fair mapping to the three common elements of digital twins that were identified in the survey, those being: a system of interest, a virtual representation of that system, and a continuum of data being shared between them. A Venn Diagram, depicted in Figure 8, was developed as a simple visualization to showcase the overlapping elements that comprise a digital twin. The diagram will be discussed more throughout the remainder of this paper.

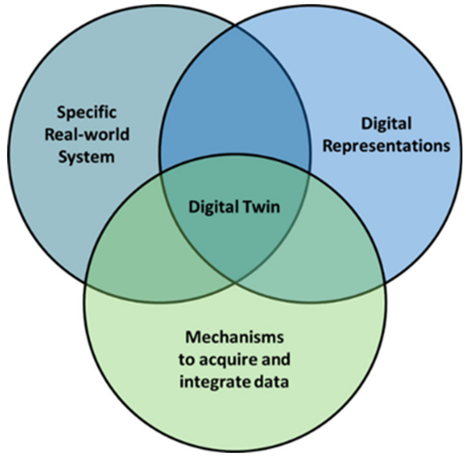


Figure 8. Elements of a digital twin

Next, we explain each condition in more detail and expound on the rationale behind the terminology used.

"**Condition 1:** A specific, real-world component, system, or process."

Traditionally, digital twin applications have primarily been of physical products, which are captured in the terms "component" and "system", where a component represents the most foundational base element of an entity, and a system represents a composition of multiple interdependent parts. For example, a digital twin of a system could be a vessel's propulsion system, and that of a component could be the shaft of that propulsion system. While digital twins have not been applied extensively to processes, they do have potential in this space, and the term "process" is adopted to include such applications. For example, one could imagine creating a digital twin of a design activity, which corresponds to a process. The phrase "real-world" is taken directly from Collette and is meant to extend beyond other definitions' use of the term "physical". Lastly, the term "specific" indicates that a digital twin maintains a one-to-one correspondence with a single, unique instance and does not consider a class of instances. For example, one could create a digital twin of a car, but not of all Jeep Renegades. Future work will explore different types of digital twins and their delineating characteristics.

"**Condition 2:** One or more digital representations of **C1.**"

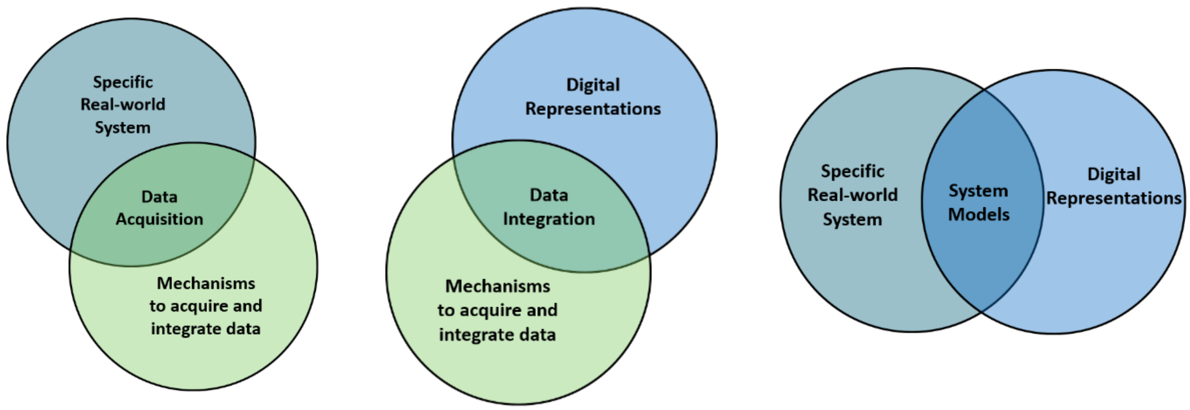
The phrase "digital representations" was adopted over the more commonly used term "model", which is generally associated with physics-based models or geometrically driven CAD models. This association is further reinforced by the fact that digital twins are often depicted as virtual, geometrical skeletons of their corresponding real-world system. "Representations" is meant to expand the potential and to be more inclusive toward using empirical equations, networks, ontological characterizations, and other such methods. For example, an empirical representation could facilitate evaluating the wear of a bearing, while an ontological representation could facilitate diagnosing faults in a propulsion system. The phrase "one or more" is meant to detract from the idea that a digital twin must be comprised of a single, exhaustive model. As demonstrated in the empirical-ontological example above, digital twins may often be used to conduct distinctly different types of analysis (diagnostics, health assessments, etc.), each with corresponding types of representations. The merging of multiple representations into one generally has no immediate benefits and often has negative consequences that will detract from the twin's flexibility and efficiency. Our definition is meant to support and inspire digital twins to contain multiple different representations based on the inferences that will be required. Lastly, the phrase "of Condition 1" better scopes the digital twin and more importantly delineates the twin from its environment. While certain inferences may require or benefit from observations of the environment, our definition does not consider its environment to be included within the digital twin.

"**Condition 3:** Mechanisms for acquiring and integrating data such that **C2** reflects **C1** over time."

The third condition establishes two processes essential to creating a digital twin by using the terminology "mechanisms to acquire and integrate data." This phrase signifies that there is a connection between the real-world system and the digital representation. The phrase "mechanisms to acquire … data" is used to generalize the source of data. Sensors are often prescribed as the method of automatic data collection; however, this is not always be feasible or cost-effective. For example, a sensor may not be the most accurate or effective means of updating a digital representation informing that a propulsion system has been repaired - it may be simpler to update the digital representation using manual input from the mechanic who worked on the system. The proposed terminology is meant to generalize the sources of collection to include reports, inspections, surveys, and other such sources. On the other hand, "mechanisms for…integrating data" is meant to update the digital representation "over time", which signifies its dynamic nature. While acquiring data relates to the real-world system, integrating data relates to the digital representations.

There are certain elements not included in these conditions, namely features and decision-making. While certain features, such as artificial intelligence and machine learning techniques, may improve twin performance, they should not be considered requisite components since a digital twin could exist and function without them. In regard to decision-making, our definition views digital twins as supporting, but not encompassing, decision-making processes. While this makes sense for multiple reasons, the primary one is that the community has created or posed digital twins for manned, unmanned, and autonomous applications. In other words, there are digital twins made to inform human decision makers, and there are those made to operate independently of human influence. Both applications cannot be digital twins unless decision-making is seen as a separate entity. While Collette's definition does not encompass decision-making, it does highlight the importance of considering how the digital twin will be used regarding the decisions that must be informed. Future work will extensively discuss decision-making and its role associated with digital twins.

The digital twin environment which arises through the three conditions provides a good first step towards defining a holistic and generic taxonomy. However, the Venn Diagram shown in Figure 8 reveals overlap regions between these three conditions which necessitates further exploration and discussion. These overlap regions, depicted separately in Figure 9, provide a perspective into capabilities of digital twins.



"Data Acquisition" "Data Integration" "System Models"

Figure 9. Overlap regions

Data acquisition is the intersection of "real-world system" and "mechanisms to acquire and integrate data". In addition to identifying what needs to be acquired about the real-world system, we also identify how that information is acquired. An example of data acquisition is via sensor strain gauge on a shaft in specified propulsion system. Human observations are also classified as data acquisition. For example, if a bearing is replaced on the shaft, the mechanic has that information.

Data integration is the intersection of "mechanisms to acquire and integrate data" and "digital representations." The technique to integrate data is dependent on the method of presenting information. Suppose a free body diagram of a shaft in a propulsion system is the digital representation, and the information available is strain gauge sensor data. An application programming interface (API) is a technique that could foster the available data to the free body diagram. Data integration can also be as simple as human input, such as manually inputting strain gauge data into a free body diagram.

The last intersection of two conditions is between a "real-world system" and "digital representations," referred to as system models, which is where inferences are made that will be valuable towards informing a decision. For example, a free body diagram of a specified shaft in a propulsion system that can represent loads is a system model. When the system model is analyzed such that information is inferred about pump that was not transparent before.

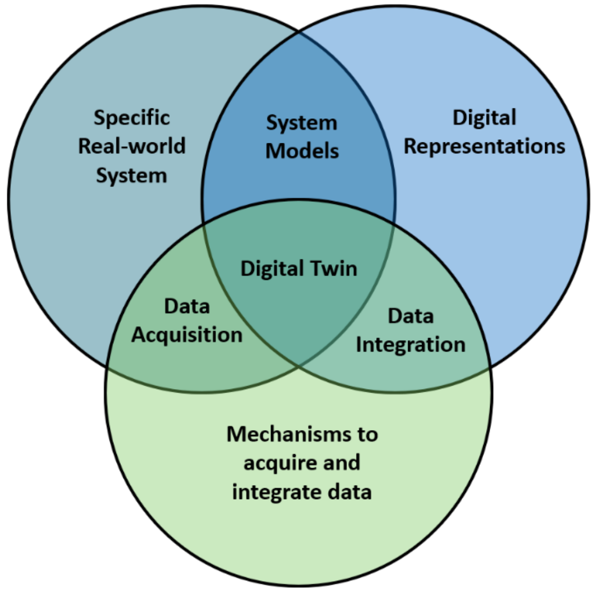


Figure 10. Elements of a digital twin with overlap regions defined

While the overlap regions provide insight into capabilities of digital twins. Overlap regions themselves are not digital twins as they lack the intersection of a third condition. All overlap regions can be observed independently, but to view a digital twin, they must be interrelated. Consider data acquisition; it is the capability to acquire information about the real-world system. However, identifying mechanisms to acquire data is determined by information or inferences required about the real-world system. System models allows this through its ability to represent the real-world system in digital representations. A digital twin is then created when data that has been acquired about the real-world system is being integrated into the digital representation. The development of a digital twin is the intersection of the three overlap regions, that being the center region in Figure 10.

SECTION 2.3 CONCLUSIONS

This report recognized a lack of unity in defining and understanding digital twins. This led to a unified definition and taxonomy of digital twins. While this preliminary taxonomy for digital twins is part of a larger research goal, it is our hope to provide a basis of discussion and collaboration, inspiring future evolution of this work and ultimate adoption across the Department of Defense.

The following chapters of this report discuss the different types of digital twins as well as their delineating characteristics. Decision-making and its relation to digital twin is also discussed more extensively.

SECTION 2.4 REFERENCES

[1] M. Grieves and J. Vickers, “Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems,” in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, Springer International Publishing, 2016, pp. 85–113.

[1] C. Parris, “What is a Digital Twin? | GE Digital.” https://www.ge.com/digital/blog/what-digital-twin (accessed Oct. 13, 2020).

[2] “The digital twin of the product | Automotive Manufacturing | Global.” https://new.siemens.com/global/en/markets/automotive-manufacturing/digital-twin-product.html (accessed Oct. 13, 2020).

[3] R. Rosen, J. Fischer, and S. Boschert, “Next generation digital twin: An ecosystem for mechatronic systems?,” in *IFAC-PapersOnLine*, Sep. 2019, vol. 52, no. 15, pp. 265–270, doi: 10.1016/j.ifacol.2019.11.685.

[4] P. Scully, “The 250 classifications of Digital Twin technology,” 2020.

[5] S. O. Erikstad, “Merging Physics, Big Data Analytics and Simulation for the Next-Generation Digital Twins,” *HIPER 2017, High-Performance Mar. Veh. Zevenwacht, South-Africa, 11-13 Sept. 2017*, no. September, pp. 139–149, 2017.

[6] B. Gesing and M. Kückelhaus, “A DHL perspective on the impact of digital twins on the logistics industry DHL Trend Research Digital Twins in Logistics,” 2019.

[7] A. Mussomeli, B. Meeker, S. Shepley, and D. Schatsky, “Signals for Strategists Expecting digital twins Adoption of these versatile avatars is spreading across industries,” 2018.

[8] D. E. Jones, C. Snider, L. Kent, and B. Hicks, “Early stage digital twins for early stage engineering design,” in *Proceedings of the International Conference on Engineering Design, ICED*, 2019, vol. 2019-Augus, pp. 2557–2566, doi: 10.1017/dsi.2019.262.

[9] M. Colledani, W. Terkaj, T. Tolio, and M. Tomasella, “Development of a conceptual reference framework to manage manufacturing knowledge related to products, processes and production systems,” in *Methods and Tools for Effective Knowledge Life-Cycle-Management*, Springer Berlin Heidelberg, 2008, pp. 259–284.

[10] “The digital twin of the production | Automotive Manufacturing | Global.” https://new.siemens.com/global/en/markets/automotive-manufacturing/digital-twin-production.html (accessed Oct. 13, 2020).

[11] J. Um, S. Weyer, and F. Quint, “Plug-and-Simulate within Modular Assembly Line enabled by Digital Twins and the use of AutomationML,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15904–15909, Jul. 2017, doi: 10.1016/j.ifacol.2017.08.2360.

[12] “The digital twin of performance | Automotive Manufacturing | Global.” https://new.siemens.com/global/en/markets/automotive-manufacturing/digital-twin-performance.html (accessed Oct. 13, 2020).

[13] C. Dufour, Z. Soghomonian, and W. Li, “Hardware-in-the-Loop Testing of Modern On-Board Power Systems Using Digital Twins,” in *SPEEDAM 2018 - Proceedings:*

International Symposium on Power Electronics, Electrical Drives, Automation and Motion, Aug. 2018, pp. 118–123, doi: 10.1109/SPEEDAM.2018.8445302.

[14] A. Coraddu, L. Oneto, F. Baldi, F. Cipollini, M. Atlar, and S. Savio, “Data-driven ship digital twin for estimating the speed loss caused by the marine fouling,” *Ocean Eng.*, vol. 186, no. March, p. 106063, 2019, doi: 10.1016/j.oceaneng.2019.05.045.

[15] M. Schirmann, M. Collette, and J. Gose, “Ship motion and fatigue damage estimation via a digital twin,” *Life-Cycle Anal. Assess. Civ. Eng. Towar. an Integr. Vis. - Proc. 6th Int. Symp. Life-Cycle Civ. Eng. IALCCE 2018*, pp. 2075–2082, 2019.

[16] A. Danielsen-Haces, “Digital Twin Development,” *Nor. Univ. Sci. Technol. Master Thesis*, no. June, 2018.

[17] A. Bekker, “Exploring the blue skies potential of digital twin technology for a polar supply and research vessel,” *Mar. Des. XIII*, vol. 1, no. June, pp. 135–146, 2018.

[18] A. Bekker, M. Suominen, P. Kujala, R. J. O. De Waal, and K. I. Soal, “From data to insight for a polar supply and research vessel,” *Sh. Technol. Res.*, vol. 66, no. 1, pp. 57–73, 2019, doi: 10.1080/09377255.2018.1464241.

[19] D. Knezevic, “Enabling High-Fidelity Digital Twins of Critical Assets via Reduced Order Modeling.” 2020.

[20] W. C. Baldwin and W. N. Felder, “Mathematical characterization of system-of-systems attributes,” in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, Springer International Publishing, 2016, pp. 1–24.

[21] A. N. Steinberg and C. L. Bowman, “Rethinking the JDL Data Fusion Levels,” 1998.

[22] A. Canedo, “Industrial IoT lifecycle via digital twins,” *2016 Int. Conf. Hardware/Software Codesign Syst. Synth. CODES+ISSS 2016*, p. 2974008, 2016, doi: 10.1145/2968456.2974007.

CHAPTER 3 DELINEATION OF DIGITAL TWIN TYPES IN THE NAVAL DOMAIN

Authors: Brendan Zauel[1], Matthew Collette[2], Michael Sypniewski[1]
1 – Martin Defense Group
2 – University of Michigan, Department of Naval Architecture and Marine Engineering
Date: February 2021

SECTION 3.1 MOTIVATION & SCOPE

This section will address twin definitions, methods, and relationships to other twins and decision-makers. Per our defined taxonomy, a digital twin requires a "specific, real-world component, system, or process". We briefly discussed the rationale behind the terms "component, system, and process" but did not elaborate on either the nuances that delineate these different types or the proposed utility associated with making such distinctions, this is the focus of the research presented here.

As potential applications of digital twins continue to expand, twin intents and objectives evolve to solve new problems. By constraining the scopes of digital twins, various types of digital twins can be identified to help focus on the evolving missions and goals for applied twin technologies.

This report begins by investigating how industry and academia have approached the concept of delineating types of digital twins, presents a purpose-driven classification for digital twins in the naval domain, and identifies areas for further research into types of digital twins.

SECTION 3.2 SURVEY OF DIGITAL TWIN TYPES

Sources across academia and industry have proposed that there are different types of digital twins that can be distinguished based on characteristics about their system of interest. This section explores these sources to detail key considerations for classifying digital twins and to provide clarity and mutual understanding among the developers, suppliers, and users of future digital twin technology. Due to different baseline definitions of digital twins across industry and academia discussed Chapter 2 A Standardized Definition and Preliminary Taxonomy for Digital Twins, digital twins have been posed as ranging from simple data collection tools to uber-models of systems that provide a base to solve any type of problem. A standardized delineation of digital twin types can be used to better prescribe the expectations and responsibilities for digital twins between these two extremes.

Sources that expound upon types of digital twins [1-13] typically differentiate them with respect to the system characteristics (i.e. what the digital twin aims to represent) using three main system qualities: quantity, hierarchical level, and abstraction. A digital twin's *quantity* refers to the number of objects the digital twin represents and whether they are combined into a system or relatively unconnected. The numerical distinction is primarily between "single" or "many" [1]–[6]. For example, a digital twin may be an asset or a system of connected assets [1], [2], [4]. A digital twin's *hierarchical* level refers to how the interrelationships among digital twins create an embedded or implicit hierarchy of digital twins, i.e. where digital twins envelop and roll up into larger digital twins, and where the digital twin in question may fall within that hierarchy. The hierarchical level of a digital twin is less strictly identified, but the distinction is made by characterizing digital twin relationships. Usage of the hierarchical distinction in digital twins is more directly applicable in manufacturing [4] and shipping [7]. For example, DHL describes a specific application for interrelated hierarchies of digital twins in logistical supply chain setting and how the collected data and relationships can be leveraged to optimize the operations, such as the relationships between the products themselves, the warehouse distribution systems, and the global scale logistics networks [7]. Scully designates six different classified types of digital twins based on their "hierarchical level" ranging from information models to multi-system digital twins [5]. The hierarchical levels exist within systems that have distinct hierarchical characteristics, like the relationship between a cog and the machine that the cog is a part of. Embedded digital twins can contain these relationships in reality or perception according to the characteristics of the hierarchy itself [7]. A digital twin's *abstraction*, in this case, refers to the state of the system's existence, e.g. physical object, class, or process. The term "abstraction" is used for classifying digital twins by the level of physical substance of the system they represent. Various literature describe a digital twin's level of abstraction as: physical product [1], [2], [5], [7]–[10], pre-production or design [1], [5], [9], process [2], [4], [5], [8], [11], [12], or performance [4], [5], [13]. The difference in abstraction is that the digital twin for a physical product is linked to a product that both exists and sends data to its virtual counterparts, whereas more abstract subjects like processes or classes link the digital twin to a process or a class of objects for evaluation or design analysis, respectively.

In the naval domain, shipboard digital twins have been created or proposed throughout academia with respect to the distributed power system [14], fouling-speed loss relationships [15], structural fatigue and ship motions [16], propulsion system monitoring and calibration [17], and entire vessel platforms [18], [19]. An industry application of digital twin work in the marine context is Akselos [20], who introduced high-fidelity reduced order modeling of offshore marine structures, primarily in the energy sector. As it pertains to the level of abstraction, the current digital twins in the field of naval architecture and marine engineering are highly specific; these twins inform decisions using collected data from real-world products and systems.

The discussion of digital twin classifications is dominated by distinctions based on characteristics of the twin's real-world system (i.e. what that system physically consists of), such as the classification shown in Table 2 given by Baldwin [21]. Baldwin's original system classification framework includes mathematical representations for each system along with biological examples, each of which are omitted here.

Table 2. System Attributes and Taxonomy [21]

System Type	Characteristic Attribute(s)	Description
Component	Existence	Physical entity
Sub-System	Process	Transforms inputs into outputs via a process
Simple system	Autonomy	Able to meet a stated system goal without outside help
Composite SoS	Diversity, Connectivity, Belonging	Able to exchange information to provide mutual support; outputs predictable
Complex SoS	Emergence	Unpredictable outputs
Adaptive SoS	Adaptability	Able to change architecture of system in response to outside pressures

Baldwin's systems framework uses both the characteristics and the decision-making processes of the system to delineate the different systems; however, as explained in the Task 1.2 report, our framework for types of digital twins is independent of the means for decision-making for the real-world system. It is apparent that a framework for types of digital twins can be based on the real-world system characteristics, but distinction is limited by subjectivity and relativity of the systems themselves. When separating systems characteristically, Steinberg et. al argues that the differentiation of assemblies and parts is specific to perspective, and partitioning is based on which characterizations of the situation, or system, are of interest to the user [22]. Distinguishing digital twins solely by the characteristics of the real-world system is not an entirely objective process. Instead, it may be more useful to classify twins based on their application or purpose, with the intent of maximizing the utility in making such distinctions.

The purpose of a digital twin is inherently tied to the goals of the system operator and the decisions the twin is being used to inform. Using the purpose of the twin to differentiate between twin types creates opportunity to tailor digital twins more closely to what they aim to do and eliminates the implied necessity for the digital twin to perfectly model all aspects of the real-world system of interests. By keeping the objectives of the creator and user at the forefront, a purpose-driven classification helps prevent any subjective or arbitrary delineation that can arise when differentiating by system characteristics.

SECTION 3.3 CLASSIFICATION OF DIGITAL TWIN TYPES IN THE NAVAL DOMAIN

While most digital twin classification schemas have been based on the characteristics of their corresponding systems, it is more useful in a naval context to differentiate twins by how they are used (i.e., the types of questions they are used to inform). This is due to the dynamic, diverse, and highly interconnected problem landscape the naval domain presents. The flexibility and subjectivity in classifying digital twin types can help provide modularity within the naval and marine industry by relating the twin purpose to their context. Accounting for the objectives of developers and users, we pose the following types of twins: **component**, **system**, **platform**, and **fleet**. These posed types aim to highlight differences in twins based on the decisions they are meant to inform in addition to encapsulating the key characteristics of each specific system of interest.

SECTION 3.3.1 COMPONENT

The component digital twin is the "atom" of our classification scheme and represents the most basic twin from which a system operator would collect useful information for decision-making. While a "component" may be composed of subsystems or interconnected sub-components, there may be little to no utility in representing those parts in additional granularity if they do not alter a decision being made. The value of the component twin comes from modeling a single part from which inherent, observable, and pertinent information can be gleaned that satisfies the goals of the digital twin and, in turn, the decision-maker. When it comes to the hierarchical levels characterized throughout the literature survey, the component digital twin represents the lowest hierarchical level that warrants a digital twin.

For example, consider a shaft bearing (Figure 11) used as one component in a representative shafting system to monitor and predict shaft vibrations. We can outfit the bearing with force and

41

displacement sensors, ensuring this data is collected and integrated with an appropriate dynamic digital representation, thereby creating a component digital twin of the shaft bearing. With such sensors and an appropriate model (or models), it would be possible to monitor parameters specific to the bearing and infer shaft vibrations considering this bearing as a component.

While the bearing is made up of smaller individual parts (such as its case, skirts, or balls), creating digital twins of these sub-components may not provide utility in the context of modeling and predicting shaft vibrations. If the context of the decision were to change, such as in the case of predictive maintenance of the bearing itself, this may require the bearing subcomponents to be reconsidered as components themselves, so long as they yield relevant information to the decision being made, and the parameters of these components are observable (and/or inferable) and useful.

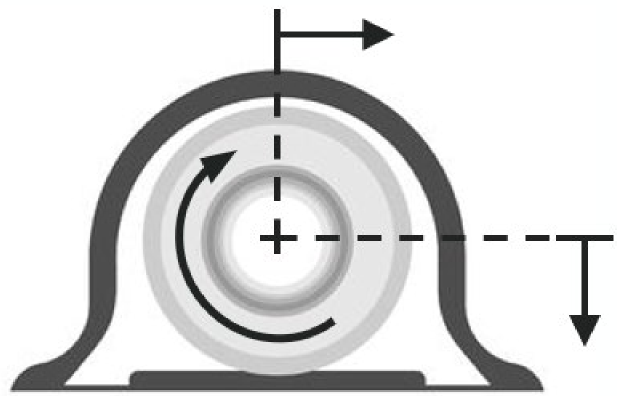


Figure 11. Component digital twin: shaft bearing displacements

As such, a component digital twin represents the most fundamental element required to answer a question within a given context. While this delineation remains somewhat subjective, it is more apparent when considered contextually. What is deemed an effective component twin should be considered relative to the question being posed. Is there additional value in considering finer-grained components? Are parameters of these fine-grained components observable or inferable? Does the information provided by measuring and modeling these components impact decision-making? These questions are critical considerations in developing a useful digital twin, and must be carefully considered in the full context of the digital twin taxonomy (See Chapter 2 Standardized Definition and Proposed Taxonomy).

SECTION 3.3.2 SYSTEM

What we call a "system" digital twin is comprised of complementary components and/or subsystems whose interrelationships demonstrate emergence, where the whole is greater than the sum of its parts. Components serve specific functions and contain specific goals, while systems achieve a goal through the coordination of component goals states and their associated interdependencies. Components within a system may be coupled directly, such as through physical connections or through logical dependencies to attain a goal but may also be indirectly coupled through their spatial locations or how they are used through time. Systems also exist on a spectrum of complexity which is characterized by the number and nature of these interdependencies. More complex systems are likely to exhibit more emergent behaviors and pose more of a challenge to creating effective system digital twins.

As an example, consider the simple propulsion system shown in Figure 12. This representative system is comprised of four components: the engine, the reduction gear, the shaft, and the propeller. The goal of this system is to create and provide thrust. Each of these components has a specific function and role in the system – the engine generates rotation, the gear and the shaft transmit rotation to the propeller, and the propeller converts rotation to thrust. Data from each component can help the digital twin model and determine the system's efficiencies or faults.

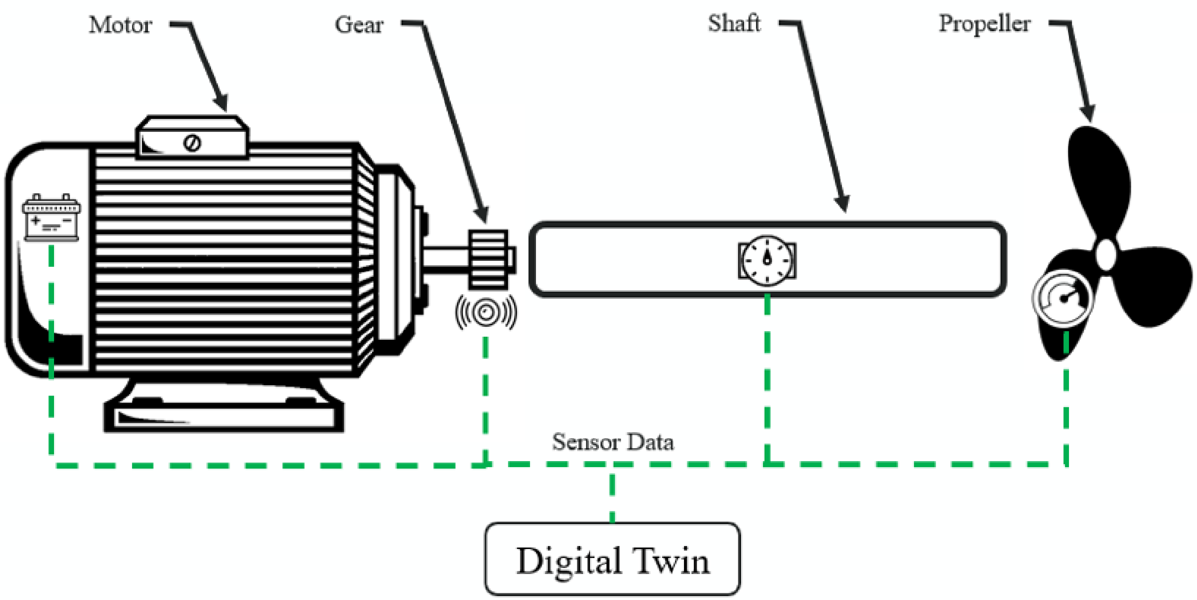


Figure 12. System digital twin: propulsion system efficiency

Each component of this example system has a different relative goal, and the goal of the overall system is not directly tied to any specific individual component. Sensors are available to provide component-centric feedback, and the combination of sensor data must be aggregated to provide system-level information. For example, flowmeters and RPM sensors on the engine might be used to determine fuel efficiency, strain gauges on the shaft may be used to inform the condition of the shaft, and pressure sensors and distribution models about the actual thrust output from the propeller. An effective system digital twin would integrate this information with an appropriate system model to yield information about the propulsion system's performance.

An important aspect of a system level twin is that it requires the aggregation of component information and interdependencies to achieve a higher, system-level goal which is still well-defined. In this case, a goal may be to produce the maximum amount of thrust using the minimum amount of fuel. This goal is still quantifiable and measurable but exists at a higher level than any single component, and the twin requires a model that carefully coordinates and integrates its components to achieve its purpose.

While this simple example is included to succinctly illustrate how a system twin differs from a component twin, systems can be far more complex. The system digital twin can correspond to systems with greater numbers of components and complex interdependencies and model them accordingly. While this added complexity serves as a challenge in developing useful and effective system twins, it does not impact the fundamental nature which delineates such systems from components.

SECTION 3.3.3 PLATFORM

The third type of digital twin we pose is the “platform” digital twin, which consists of multiple *diverse* systems. A naval vessel is comprised of numerous complex systems which work in tandem during operations with very different purposes and goal states. Similar to the component-system twin interaction, the platform level twin considers also encapsulates emergence. However, the goals and decisions made at a platform level are fundamentally different from those that exist at component or system levels, due to the presence of coordinating diverse system goals. These platform level goals are broader and more abstract than for a specific component or system. Due to their level of abstraction, the number and diversity of possible goals are much larger, and do not directly translate to a finite set of system or component goal states.

Consider a specific naval vessel as a real world system of interest for which a platform digital twin is developed that collects data from the various critical subsystems to assess the vessel’s condition (Figure 13). In this case, consider these critical subsystems as the propulsion, armament, navigation, and communications systems. The process and scope of assessing the condition of the entire vessel is different than that of assessing, for example, the state of one of its engines, or of the radar. Moreover, the vessel’s overall status extends far beyond the condition of any individual system. The scope required to assess these components and systems are narrower, and the goals and associated decisions to be made for each are more well defined. Determining a holistic platform perspective would require the careful consideration of components, systems, and their associated interdependencies in tandem. However, the combination of potential component and system states do not directly translate to a single platform state, especially when considering exogenous factors to the platform such as the environment.

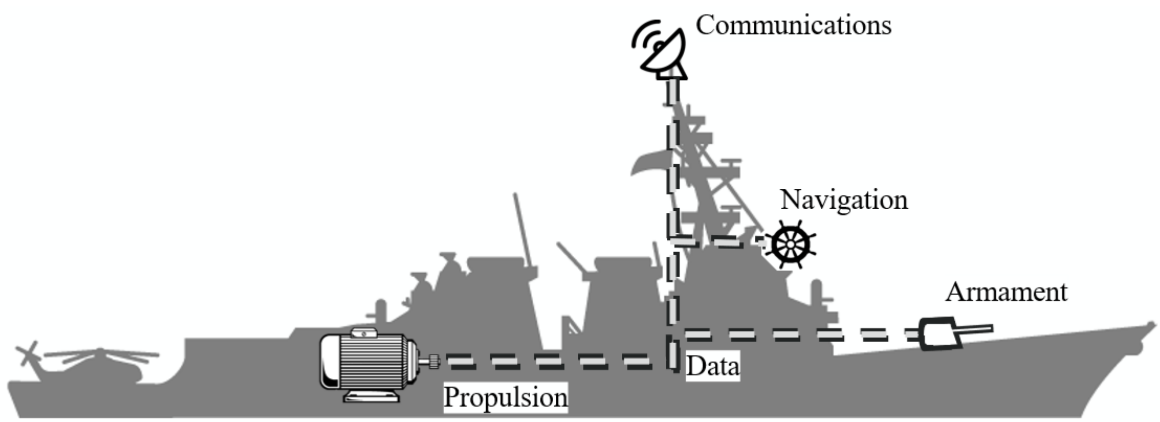


Figure 13. Platform digital twin: naval vessel holistic data modeling

In addition to the combinatorial difficulties, ‘rolling up’ component and system parameters suggests the behavior of each system is quantifiable in a similar manner. In this example, the platform-level assessment requires an assessment of each system’s condition and weighs condition against criticality. However, the criticality of each system is inherently dynamic – changing over time with the vessel’s current state, mission, and environment. For example, in the presence of an immediate threat, the status of the weapons system may be more critical than that of the vessel’s structural health. However, this is highly dependent on the vessel mission and objectives, encompasses largely varying timescales, and may shift rapidly (e.g. in the presence of a damaging event to the vessel, the hull structure could become the most critical system). As such, it is not straightforward (nor in some cases possible) to establish a single function which combines the statuses of its diverse subsystems into a holistic and useful ship-wide assessment for decision-making.

These difficulties suggest that a platform level digital twin is not simply a synthesized perspective of the associated component and system level twins. The diversity of onboard systems distinguishes the platform digital twin and makes it infeasible and insufficient to consider the platform as a large scale "system twin". While sensors associated with components and systems will certainly be useful in informing such a platform level twin, unique models and perspectives are required to translate this data into the more abstract platform level models for decision-making.

SECTION 3.3.4 FLEET

Beyond the platform level digital twin, here we pose an additional digital twin which considers multiple platforms. Each platform within a fleet twin can operate independently but is linked with other platforms spatially and temporally. Fleets are characterized by the coordinated efforts of platforms to attain higher-level objectives. Relationships between platforms may be organized and explicit, such as entities within a supply chain, or random and disjointed, such as cars on a city street [23]. Platforms within a fleet are often diverse and are developed for specific functions and are developed (perhaps not explicitly) for the purpose of exhibiting emergence. The naval fleet is an intuitive example of a maritime fleet digital twin application, whereby vessels of various classes and roles are monitored and managed to complete strategic objectives.

Digital twins of fleets are valuable in capturing the logistical nature of platform interdependencies. Such twins are more focused on the macro-behaviors of vessels and can be queried to inform more strategic decisions. For example, in the context of Figure 14, a fleet level twin would be useful in determining the most effective allocation of vessels to survey a given area.

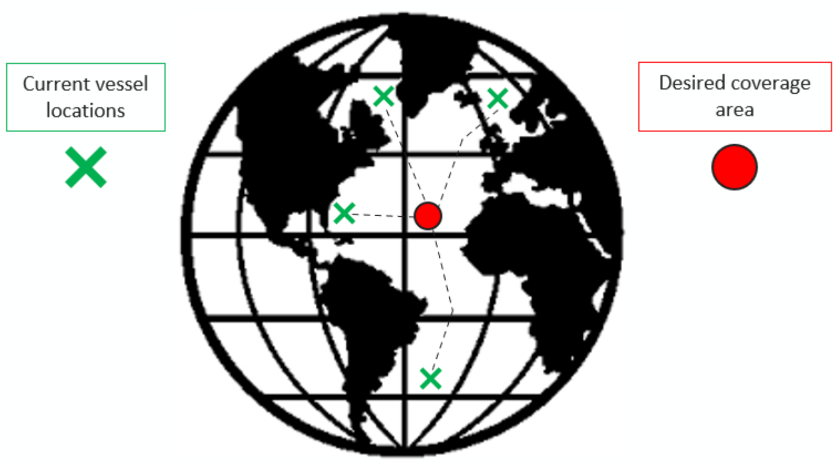


Figure 14. Fleet digital twin: naval fleet logistics question

In this example, the fleet digital twin must consider the distances of vessels from the desired area and determine the time frames in which each vessel can converge on that area. Additionally, such a twin may consider each platform's capabilities and the relevance to accomplish the mission. While this example is simple, there is the high potential for such a twin to become extremely complex for useful application in naval fleets. Regardless, these high-level queries are unable to be answered using the other types of twin and warrant a delineation as it pertains to the fleet use case and applicability for decision makers.

SECTION 3.3.5 SUMMARY

In this classification scheme, four different types of digital twins were proposed to delineate digital twin types based upon their purpose and intended value rather than on the characteristics of the real-world system of interest. While the presented digital twin types have been framed in the context of the naval domain, the types remain extensible to the various hierarchies and levels associated with other domains. Table 3 presents a summary of the presented digital twin types, and the associated characteristics of each.

Table 3. Summary of proposed digital twin types

Digital Twin Type	**Digital Twin Purpose**	**Naval Domain Examples**
Component	Provide relevant elemental information for system inferences.	Gear, shaft, bearing, propeller…
System	Capture interdependencies and relationships between components.	Pump, engine, propulsion system…
Platform	Capture interdependencies between disparate diverse systems.	Naval vessel
Fleet	Capture coordinated macro-behaviors between platforms.	Naval fleet

The types presented above increase in abstraction, scope, and complexity moving down the table. While there exists a natural progression in goals from component to system level twins in terms of the capabilities and values, the jump gets larger in progressing into the platform and fleet level twins due to more abstract and diverse goals and decisions. Figure 15 shows the progression of the types of digital twins as it relates to two axes: the diversity of the real-world system and the abstraction of the question to pose to the digital twin. This graphic is simplified, but these differentiating characteristics show the pattern of how the type of the digital twin is affected by both its purpose and the system it represents. The types require different perspectives in development and use.

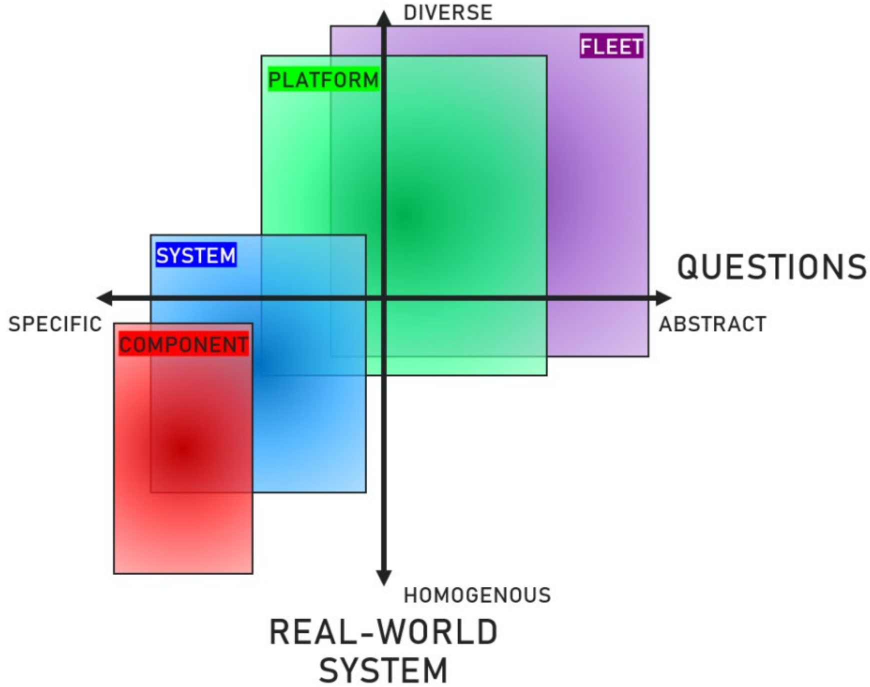


Figure 15. Types: system characteristics vs. purpose

Figure 15 shows the relationship between a system's characteristics, the purpose of the system's digital twin, and the type of the digital twin in question, as well as highlights the subjectivity and overlap of the typing classification framework outlined in this report. The more specific questions for a digital twin represent those that have more tangible answers; for example, quantifying a pump's flow rate or the efficiency of an engine considering data from the real-world systems are specific questions to pose to a digital twin. The more abstract questions are higher order, take more information into account, and don't necessarily have correct answers, such as logistics problems and coordination efforts. The higher order questions should be asked of the platform and the fleet digital twins in this framework. With respect to the characteristics of the real-world system, represented here by the diversity of the system and its subsystems, the more homogenous systems correspond to components and systems with closely related and alike subsystems, such as pumps or electrical systems. For a pump, the homogeneity of components within it plays a role in narrowing the type of the digital twin to either component or system, but the types of questions to ask the pump digital twin may further delineate; for example, if a component digital twin for a pump would answer questions that have specific answers and require little or no deeper analysis, then a system digital twin for the same pump could answer more in-depth questions concerning each component along with the entire pump holistically. The more diverse systems utilize information from different types of components and subsystems, such as propulsion systems, vessels, or fleets. The overlapping sections of the different types of digital twins represent the scenarios where more metrics like its exact applications, user preferences, and design processes to determine the best classification.

While this report presents the purpose-driven delineation, strategies to develop these twins to be useful and accurate will be left to future work. It should also be noted that the examples presented above are not exhaustive and are included as potential references to applicable systems of interest.

47

SECTION 3.4 CONCLUSIONS & FURTHER CONSIDERATIONS

The classification scheme in this report is a result of a survey into the different types of digital twins in industry and academia. This survey uncovered that many of these definitions are focused on system-specific delineations of digital twin types, rather than the purpose for which they were created, or the decisions they are meant to inform. The four types of digital twins defined in the previous section include component, system, platform, and fleet digital twins, differentiated based on what they do and purpose they serve, rather than what they consist of or what they model. The presented digital twin types are not meant to form a strict hierarchy, but instead are meant to motivate a careful consideration of different digital twin types based on the context and their intended use. While the presented types have been framed in the naval context, these definitions remain applicable to other domains as well.

It is our hope that the presented digital twin types will begin to lay the groundwork for systems of digital twins, both within and beyond the naval domain. Doing so will require significant additional research, including strategies to manage digital twin interfacing, capturing appropriate information flows, and how to implement and validate such twins in practice. Although this is a lofty goal, focusing on purpose-driven delineations of digital twins should help address these fundamental concerns.

SECTION 3.5 REFERENCES

[1] M. Grieves and J. Vickers, “Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems,” in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, Springer International Publishing, 2016, pp. 85–113.

[2] C. Parris, “What is a Digital Twin? | GE Digital.” https://www.ge.com/digital/blog/what-digital-twin (accessed Oct. 13, 2020).

[3] “The digital twin of the product | Automotive Manufacturing | Global.” https://new.siemens.com/global/en/markets/automotive-manufacturing/digital-twin-product.html (accessed Oct. 13, 2020).

[4] R. Rosen, J. Fischer, and S. Boschert, “Next generation digital twin: An ecosystem for mechatronic systems?,” in *IFAC-PapersOnLine*, Sep. 2019, vol. 52, no. 15, pp. 265–270, doi: 10.1016/j.ifacol.2019.11.685.

[5] P. Scully, “The 250 classifications of Digital Twin technology,” 2020.

[6] S. O. Erikstad, “Merging Physics, Big Data Analytics and Simulation for the Next-Generation Digital Twins,” *HIPER 2017, High-Performance Mar. Veh. Zevenwacht, South-Africa, 11-13 Sept. 2017*, no. September, pp. 139–149, 2017.

[7] B. Gesing and M. Kückelhaus, “A DHL perspective on the impact of digital twins on the logistics industry DHL Trend Research Digital Twins in Logistics,” 2019.

[8] A. Mussomeli, B. Meeker, S. Shepley, and D. Schatsky, “Signals for Strategists Expecting digital twins Adoption of these versatile avatars is spreading across industries,” 2018.

[9] D. E. Jones, C. Snider, L. Kent, and B. Hicks, "Early stage digital twins for early stage engineering design," in *Proceedings of the International Conference on Engineering Design, ICED*, 2019, vol. 2019-Augus, pp. 2557–2566, doi: 10.1017/dsi.2019.262.

[10] M. Colledani, W. Terkaj, T. Tolio, and M. Tomasella, "Development of a conceptual reference framework to manage manufacturing knowledge related to products, processes and production systems," in *Methods and Tools for Effective Knowledge Life-Cycle-Management*, Springer Berlin Heidelberg, 2008, pp. 259–284.

[11] "The digital twin of the production | Automotive Manufacturing | Global." https://new.siemens.com/global/en/markets/automotive-manufacturing/digital-twin-production.html (accessed Oct. 13, 2020).

[12] J. Um, S. Weyer, and F. Quint, "Plug-and-Simulate within Modular Assembly Line enabled by Digital Twins and the use of AutomationML," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15904–15909, Jul. 2017, doi: 10.1016/j.ifacol.2017.08.2360.

[13] "The digital twin of performance | Automotive Manufacturing | Global." https://new.siemens.com/global/en/markets/automotive-manufacturing/digital-twin-performance.html (accessed Oct. 13, 2020).

[14] C. Dufour, Z. Soghomonian, and W. Li, "Hardware-in-the-Loop Testing of Modern On-Board Power Systems Using Digital Twins," in *SPEEDAM 2018 - Proceedings: International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, Aug. 2018, pp. 118–123, doi: 10.1109/SPEEDAM.2018.8445302.

[15] A. Coraddu, L. Oneto, F. Baldi, F. Cipollini, M. Atlar, and S. Savio, "Data-driven ship digital twin for estimating the speed loss caused by the marine fouling," *Ocean Eng.*, vol. 186, no. March, p. 106063, 2019, doi: 10.1016/j.oceaneng.2019.05.045.

[16] M. Schirmann, M. Collette, and J. Gose, "Ship motion and fatigue damage estimation via a digital twin," *Life-Cycle Anal. Assess. Civ. Eng. Towar. an Integr. Vis. - Proc. 6th Int. Symp. Life-Cycle Civ. Eng. IALCCE 2018*, pp. 2075–2082, 2019.

[17] A. Danielsen-Haces, "Digital Twin Development," *Nor. Univ. Sci. Technol. Master Thesis*, no. June, 2018.

[18] A. Bekker, "Exploring the blue skies potential of digital twin technology for a polar supply and research vessel," *Mar. Des. XIII*, vol. 1, no. June, pp. 135–146, 2018.

[19] A. Bekker, M. Suominen, P. Kujala, R. J. O. De Waal, and K. I. Soal, "From data to insight for a polar supply and research vessel," *Sh. Technol. Res.*, vol. 66, no. 1, pp. 57–73, 2019, doi: 10.1080/09377255.2018.1464241.

[20] D. Knezevic, "Enabling High-Fidelity Digital Twins of Critical Assets via Reduced Order Modeling." 2020.

[21] W. C. Baldwin and W. N. Felder, "Mathematical characterization of system-of-systems attributes," in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, Springer International Publishing, 2016, pp. 1–24.

[22] A. N. Steinberg and C. L. Bowman, "Rethinking the JDL Data Fusion Levels," 1998.

[23] A. Canedo, “Industrial IoT lifecycle via digital twins,” *2016 Int. Conf. Hardware/Software Codesign Syst. Synth. CODES+ISSS 2016*, p. 2974008, 2016, doi: 10.1145/2968456.2974007.

CHAPTER 4 SURVEY OF DATA PERSISTENCE APPROACHES

Author: Matthew Collette[1]
1 - University of Michigan, Department of Naval Architecture and Marine Engineering
Date: March 2022
Marine Structures Design Laboratory Report Number: 2022-001

Abstract: To perform data-model fusion, future digital twins require robust data persistence approaches to store geometry, models, measurements, and environmental factors necessary for twin computations. However, surveys and standards for data persistence approaches appear lacking in this field. This report provides a brief introduction to this topic by reviewing over 30 papers in five different communities that have similarities to the naval digital twin community: the structural updating community, the machinery monitoring community, the lifecycle assessment community, the offshore wind turbine community, and the digital thread community. The survey reveals that data labeling, interchange, cleansing, and transfer seem to be the primary challenge with raw storage capabilities not appearing frequently. Data sources are notably heterogeneous, with unstructured data such as human inspection reports representing an extra level of challenge. While specific technical areas have developed some standards addressing these challenges, there are no well-established approaches that can be used off the shelf for data-model fusion. Instead, using the experiences of these communities, a data-model fusion project would need to develop its own persistence approach. In doing so, the offshore wind turbine community looks especially attractive owing to high-level similarities between their challenges and that of data-model fusion.

SECTION 4.1 INTRODUCTION

As digital twin applications proliferate, the need to record geometry, parameters, and sensed data has grown. Most twins rely on keeping a digital model in sync with a real-world platform so that they can perform prognoses and update underlying computational models. As part of this process, twins may have a requirement to store extensive data. Such information may include manufacturing and as-built information on the platform, operational histories, weather exposure, inspection reports, and sensed data. Especially for rapid sensing systems, including machinery control systems and strain gauge measurements, raw data storage for these signals may be a daunting challenge. Indeed, a completely high-fidelity digital twin has been shown to be most likely computationally infeasible [1]. Thus, a background literature review on how different twin industries have approached the data persistence problem will be helpful in moving forward with marine digital twins and data-model fusion approaches.

SECTION 4.1.1 DIGITAL TWIN DEFINITIONS

Before proceeding with a study of persistence, a basic overview of twins is required. An overview is especially important, as many other product lifecycle management data solutions have been proposed over the years. With the emergence of software platforms targeting sales in this domain, many terms have lost their specific meaning and become blended. In this regard, we will adapt the definitions present by Kraft [2] that helps track the U.S. military community's perspective on the differences between several related terms. The following is reproduced below directly from Kraft's paper [2]:

> **Digital System Model** - A digital representation of a weapon system, generated by all stakeholders, that integrates the authoritative data, information, algorithms, and systems engineering processes that define all aspects of the system for the specific activities throughout the system lifecycle.
>
> **Digital Thread** - An extensible, configurable, and Agency enterprise-level analytical framework that seamlessly expedites the controlled interplay of authoritative data, information, and knowledge in the enterprise data- information-knowledge systems, based on the Digital System Model template, to inform decision-makers throughout a system's life cycle by providing the capability to access, integrate and transform disparate data into actionable information.
>
> **Digital Twin** - An integrated multi-physics, multi-scale, probabilistic simulation of an as-built system, enabled by Digital Thread, that uses the best available models, sensor information, and input data to mirror and predict activities/performance over the life of its corresponding physical twin.

Wincott and Collette [3] extend this further by developing a twin's unique aspects. To count as a twin, a system must have each of the following characteristics:

1. **A real-world system of interest:** Twins are specific to one real-world platform. This separates them from digital systems models and digital threads, though they may call on models, simulations, or data stored in either of those two systems to function.

2. **One or more digital representations of the system:** A true twin requires some sort of digital representation of the real-world object. This could be as simple as 3-D CAD files from the digital system model or as complex as multi-discipline simulations spanning millions of degrees of freedom.

3. **Fusion to join the real-world system and the digital representation:** The twin must be able to relate events in the real world to the digital model. Again, a range of approaches fit under this concept of fusion, from human inspection data to complex machine learning fusing multiple sensor streams into a computational model.

4. **A decision that will depend on the output of the fusion step:** To differentiate between a twin and a monitoring or validation campaign, we also introduce the concept that a twin must influence a decision. However, recent work within the current contract has removed item 4 from the strict definition of a twin, as some decisions may result from consulting multiple twins (e.g., a mission go/no go decision depending on machinery health and expected platform motions.) Two twins may only deliver information on their part of the problem to a supervisory decision system.

From these definitions, we can see that the data domain that may need to be stored in a persistence system is large and complex. A twin may need to:

- Access design-stage CAD models of the system and as-built manufacturing data
- Access design-stage computational models that may need to be updated
- Track the history of individual platforms, including weather, operational parameters, and other relevant "background" information necessary to run the computational models
- Track sensors and health metrics on the platform, including control systems, vibration, acceleration, pressure, temperature, voltage, current, strain, and a host of other modern sensor data streams
- Track periodic inspections or human feedback, including natural language, images, and other less-precisely defined information on the platform.

How to best organize, document, and store this data is not immediately apparent. Fonseca and Gaspar [4] provide a good high-level summary of the challenges around creating twins, looking at data persistence issues as well as broader data exchange, standardization, and business case demands. Based on the increasing number of publications in this field, a literature search approach was used to compare and contrast systems for data persistence.

SECTION 4.2 LITERATURE SEARCH PROCESS AND SOURCE

A broad literature search, using a variety of keywords involving digital twin and lifecycle data, was run on several different search platforms for academic literature. In this process, digital twin was not used as a fixed criterion – e.g., papers on related topics that did not directly address twins were considered. However, as the literature review took place, pure product lifecycle management (PLM) systems were dropped from the literature review. The functionality of these systems was largely subsumed by the more complex systems that had been proposed. As most of these did not deal with the related issues of data fusion and model updating, they were seen as of secondary importance. From the remaining papers, five significant areas of development were identified:

1. **Structural Model Updating:** One of the oldest and simplest forms of a digital twin. In this approach, 3-D finite element models created at the design stage are kept up-to-date with corrosion and other structural damage or modification through life. By reassessing the vessel periodically, structural safety is assured.
2. **Machinery Monitoring:** Monitoring of machinery systems, especially rotating machinery, is also a well-established discipline. Early systems in this area did not meet the full definition of a twin. Typically, statistical pattern matching on vibration signals was used to diagnose possible deterioration of the system. Still, no consistent digital model of the system was developed or updated as part of this process. However, these systems are now both more advanced in their modeling approaches as well as requiring extensive data persistence.
3. **Lifecycle Assessment (LCA)/ Lifecycle Costing Analysis (LCCA):** These methodologies extend PLM to look at cost and environmental impacts. The models used are often simple, but detailed product data is tracked from production through manufacturing, operation, and disposal.

4. **Wind Turbines:** Offshore wind turbines are typically remotely monitored for the health of the machinery and blade structure of the turbine. Given the strong interactions between the weather, control system, and resulting structural and mechanical loads, many of these monitoring approaches have adopted an integrated, system-level view of the turbine. This view necessitates archiving a large amount of data.
5. **U.S. Military Digital Thread Approaches:** The concept of a digital thread is primarily discussed within the U.S. defense enterprise. While some commercial PLM approaches attempt to come close in scope, the digital thread is likely the most comprehensive data persistence strategy attempted to date. Thus, reviewing progress on this topic is essential to capturing persistence approaches.

Each of these development areas will be reviewed in turn.

SECTION 4.2.1 STRUCTURAL MODEL UPDATING

During the 1980s and 1990s, the emergence of finite element analysis as part of structural design and approval significantly changed the approach to designing marine structures. Complex computer analysis could now link load estimations to structural response and approval criteria. The desire to apply these tools for in-service assessment led to the realization that these design tools and data structures should be archived through life. One of the first major efforts in this area took place in Canada, where a project called ISSMM [5], "Improved Ship Structural Maintenance" proposed to develop such a data persistence structure, noting:

> One of the main advantages to be realized in creating a special purpose comprehensive analysis tool such as ISSMM, is that it can readily contain most, if not all, of the required load and structural data for analysis of the various failure limit states. The most time consuming task in current ship structural analysis is acquiring and implementing the necessary data. Often an analysis is limited by what data is available and what can be put into the necessary format in the time allotted. The database will contain all information necessary for the various modules of ISSMM: weight distribution and lines plans for the seakeeping (loads) analysis; world wide wave statistics, results for the Halifax Class of the linear 3-D sea loads analysis, predefined operational profiles including global structural responses, the basic hull girder structure finite element model and detail finite element meshes of critical structure; materials data including information for nonlinear behaviour and fatigue crack initiation and growth; and, second order (means, COV and distributions) statistics of the structural parameters for reliability analysis. Accurate weight distributions are being developed for the Halifax class with consistency between the sea loads and structural models.

ISSMM was proposed to allow wide-ranging analysis types to be performed, capturing structural corrosion, denting, cracking and allowing the current state of the structure to be assessed against criteria quickly throughout the vessel's life. Such comprehensive through-life support approaches have continued to be developed, including the "Achieving Service Life Program" for the U.S. Navy

54

[6] and a similar program for submarines in Canada referred to as subSAS, of which only minor details have appeared in the public domain [7].

The commercial world has readily adopted a related but different system to monitor and maintain commercial vessel structures. Here, the focus is corrosion, coating condition, and other defects that might need structural repair. These systems capture designs from the construction stage, using this data as the baseline for updating structural component health through-life while often sharing data with classification societies for approval. Li et al. have recently published an overview of the data processing needs of such a system [8]. Given the central role many classification societies play in this process, many have evolved similar systems. An example is DNV's "Hull Insight" service, which also adds accurate weather hindcasting to estimate the structural loading a ship has seen and forecast future repairs. Intellectual property concerns and business value for the shipyards remain obstacles in fully implementing these systems, as reviewed by Thomson and Renard [9]. However, the use of such 3-D models, instead of PDF or paper drawings, throughout the approval process may streamline some of these problems.

SECTION 4.2.2 MACHINERY MONITORING

Machinery monitoring also has a long history as a condition-based approach to maintenance, similar to structural systems. Initial marine systems in this area were often vibration-based, using portable sensors to periodically record vibrations near key bearings, pumps, or motors in shipboard systems. Early systems simply compared vibratory signals from one reading to another without attempting to make an integrated model of system health, and as such, the data storage requirements were small. More recently, advanced machine learning methods are attempting to use both edge computing and existing monitoring signals to infer machinery health. Recent examples include [10]–[12], but much of the work in this regard has been for researching machine learning methods applied to relatively isolated data sets.

More comprehensive machinery monitoring proposals have been made, but the details of their data persistence are not widely available in the public domain. In the commercial world, optimizing a vessel's fuel consumption by careful monitoring of the vessel's draft, trim, machinery parameters, and control settings is now widely viewed as possible. Maersk is known to perform such optimizations on vessels in its fleet, using the "Maersk Ships Performance System," but little has been published about the system’s internal design. Classification societies are also moving into this area, hoping to offer this as a service for owners who are not interested in developing their own complex integration and machine learning expertise. Public documents on such systems do not indicate a significant tie to the machinery health predictions to date. Systems appear to integrate vessel position in the water, weather, fouling, and models for propellers, shafts, generators, engines but without adaptive models for damaged or degraded machinery systems - e.g. [13]–[15]. Lazakis used FEMCA and fault trees to find efficient groups of sensors to predicting future failures vs. wide-scale monitoring of all possible data [16]. For data persistence, the published work implies a combination of time-series data from machinery (vibration, temperature, pressures, flows) depending on the model's aim, as well as macro-level platform parameters (draft, trim, weather, speed) are necessary to build such models. Few real-world examples with data are available in the literature. Abbasian et al. [17] present a "big data" warehouse approach applied to a real-world offshore support vessel. Offboard weather, machinery parameters, electrical bus loads, draft, and position are all recorded. Three thousand signals (features) were sampled every 5 seconds for 42 days of operation, resulting in over 7 million records stored in a database format. However, of the 3,000 possible signals, only 80 were used for further analysis in the paper.

SECTION 4.2.3 LIFECYCLE ASSESSMENT (LCA)/ LIFECYCLE COSTING ANALYSIS (LCCA)

Lifecycle assessment and lifecycle costing methods are attempts to track both the environmental impact and the overall costs of engineering projects. Such tracking requires decomposition of the engineered structure into many components, tracking the manufacturing, operation, and disposal of each while assigning multiple environmental impact measures and cost measures to each step. In such approaches, gathering and tracking the relevant data is a key challenge in implementing the method. Jeong et al. give a recent overview of this process for selecting propulsion systems on a series of sample vessels [18]. From this work and others, it is clear that for LCA/LCCA, efficient access and integration of data is a central challenge to conducting these methods. Indeed, unlike more complex engineering simulations, in LCA/LCCA, the final calculation is usually an additive assembly of impacts or costs, but one that must be done correctly for a large number of components and operations onboard the vessel. Commercial software, with databases of lifecycle impact, such as GaBi, appear widely used. The data challenge is primarily connecting the description of the physical system and its operation to these databases.

Favi et al. [19] directly address these challenges, proposing a system that would link databases of impacts into engineering CAD models, allowing rapid development of LCA/LCCA estimates during ship design without extensive manual data entry. A proposed database architecture and examples for three motor yachts are given. For land-side buildings, Lu et al. [20] present a review of LCA and conventional building information modeling. The review indicated that three major approaches were in common use, and approaches that integrate across all phases of the lifecycle are still rare. In the context of ceramic tile manufacturing, Ferrari et al. [21] extended this approach to capture real-world manufacturing performance using feedback from internet-of-things readings in a factory. Commercial ERP, including SAP BusinessObjects were used as data translators and databases.

SECTION 4.2.4 OFFSHORE WIND TURBINES

Offshore wind turbines are one of the strongest system-level examples of data persistence to date. Offshore wind turbines have several simple sub-systems that are coupled and interact during operation, including large and flexible structures, generators, and gearboxes. Additionally, many of the loads on the structure can be strongly influenced by the blade pitch control strategy. To monitor and analyze these turbines, it is necessary to integrate several different measurements, including weather, machinery, control commands, and structural responses. Many countries require active monitoring of a certain percentage of these turbines, which has resulted in several recent studies exploring data persistence approaches for these problems. With the remote location of offshore wind turbines, there is a strong desire to predict machinery failures ahead of time to avoid costly downtime or unplanned visits to the turbines.

Papatzimos et al. [22] provide an overview of a recent system designed to reduce operational and maintenance costs of offshore wind farms. A relational database approach was taken for persistence, with structured data directly manipulated in the database (including local weather, wave buoy data, machinery, and structural health readings) and a distributed database approach for reports, images, and other unstructured data. The authors note that the key problem is data collection and processing, with data streams ranging from database queries to emails. Helsen et al. [23] provide a very similar system, but instead of using a relational database, they choose a NoSQL database approach. The differences between these two papers indicate that the persistence approach may vary between applications. Martinez-Luengo et al. [24] addressed the challenge between strain gauge data, which may have drop-outs and is sampled at a very high frequency, and 10-minute weather observations for health monitoring with data compaction, cleansing, and missing data imputation strategy. Indeed, the topic of making these data persistence schemes adaptable over time has also been studied, though the results to date are primarily desk studies [25]. The wind turbine industry appears to be actively researching the broader issue of data persistence and is collecting a large amount of data from nearly-identical turbines worldwide.

SECTION 4.2.5 DIGITAL THREAD

Kraft laid out the U.S. Air Force's vision of a digital thread in 2016 [2], standardizing the interrelationship between the digital systems models, thread, and twins as covered in the introduction. However, the main bulk of Kraft's paper focused on design-stage data exchange and re-use and its impact on acquisition programs. The broad issues of data persistence to support data-model fusion were not explored in depth. Over the last five years, digital thread has become a focal point for acquisition programs or early-stage design (e.g. [26]). Still, the amount of actual system experience with the concept and the architectures for data persistence is less clear. This slower pace of development is potentially a result of the longer lead time on military acquisition compared to offshore wind turbines applications, where the focus on supporting operations and maintenance is already well-established.

Despite the design focus at the outset of the digital thread definition, work since this time has focused on implementing the broader concept throughout the engineering lifecycle. Kwon et al. [27] present a detailed study of extending design-stage STEP data standards to include human quality inspection in QIF format, using a knowledge graph approach. At the same time, Sousa [28] explored further standards around integrating quality assurance into digital-thread-like applications. In both works, the focus is primarily on data manipulation, and labeling, the storage aspect of data persistence appears less critical. Gopalakrishnan et al. [29] provide a complete example of this type of approach, using STEP and QIF approaches to store material microstructure information for a gas turbine component through-life to allow downstream damage-tolerant safety assessments to be made during the turbine’s operational lifecycle. This paper is one of the few that ties together data exchange and labeling methods with a complete example.

At a higher level, Singh and Willcox [30] worked towards a mathematical definition of a digital thread and developed a composite wing-box example. In this formulation, the digital thread's primary focus was choosing sensor locations for model updating and design step sequencing to reduce uncertainty in the design problem efficiently. While many existing data exchange standards are discussed in the introduction, it is not clear that the final example used any of them. Pang et al. [31] provide an even higher-level description focusing on a shipyard. Pang et al. identify similar data types to be stored as was highlighted in the offshore wind turbine section above, and also note the desire to integrate with existing model-based engineering and product lifecycle management commercial software for efficiency of implementation. Jagusch et al. [32] provide a high-level discussion of the digital thread in shipbuilding but do not discuss specifics of the data persistence necessary for their vision.

SECTION 4.3 COMPARISON OF APPROACHES AND FINDINGS

The literature search above revealed five major areas of development that could inform data persistence approaches for data-model fusion projects: the structural updating community, the machinery monitoring community, the lifecycle assessment community, the offshore wind turbine community, and the digital thread community. It was clear that the maturity of these communities varied widely – the structural updating and machinery community have decades of increasingly-complex experience. In contrast, others, such as the digital thread approach, have less than a decade of experience.

Despite this variety of maturity, several common themes could be seen across the different disciplines investigated. Very few papers mentioned challenges in data scale – e.g., the ability to store a large amount of data was not a critical factor. However, many papers also addressed academic-scale proof-of-concept applications where gathering data from multiple assets for long periods of time was not attempted. Thus, while this seems like a secondary challenge at the moment, more extensive systems may struggle here in the future.

Many papers did discuss data labeling, ontologies, and transfer standards. It would appear from the published record that such standardization and labeling of data is a central challenge. Cleaning and processing data before analysis was also a commonly-mentioned challenge. Additionally, it is also clear that we can collect data easier than we can fully understand it. Abbasian et al. [17] experience of sampling 3,000 channels of data, but only investigating 80 in detail is the latest example of the challenge being more making sense of the data than gathering the data. Similar complaints have been published for decades now, including the challenge of "data to decisions" and experience with marine monitoring systems that were not maintained after the data gathered could not improve decision-making.

Any marine data-model fusion data persistence approach is likely to have to work with different heterogeneous data sources. Table 4 below shows the different data sources explicitly mentioned in the papers reviewed. As such, this must be seen as a lower bound – and likely far lower – to what is being used in practice today. Data types include design-stage CAD models, weather and operational data, time-history signals, and "unstructured" data such as images or human inspection reports. While the literature presented several different data storage architectures for such data, a larger challenge appears to be able to search and interpret such diverse data effectively.

58

Table 4: Comparison of Mentioned Data Types by Disciplines

Data type	**Structural Updating**	**Machinery Monitoring**	**LCA/ LCCA**	**Wind Turbines**	**Digital Thread**
3-D CAD Models	X		X		X
Finite element models	X				X
Hydrodynamic models	X	X			X
Weather models	X	X		X	
Model updating from field	X	X	X	X	
Vessel position for weather	X	X			
Ship position in the water		X			
Time signals from machinery monitoring systems		X		X	
Commercial databases/business process software			X	X	X
Wave buoys and local weather readings				X	
Human inspection reports	X			X	X
Images and other non-structured data	X			X	X
Control system logging				X	

Based on the analysis of the written literature, it is likely that any data-model fusion system will need to develop its own unique data persistence approach. The level of standardization appears low at the current time, with only a handful of papers documenting the successful application of existing data transfer standards to engineering problems. Selecting an applicable standard, or at least a data labeling scheme (such as XML schema or UML/SysML defined data path), would help formalize such a system. Data storage requirements seem likely to be met by existing database systems; though relational databases seemed most common in the literature, it was clear that the final choice of database architecture is not yet standardized.

Of the five fields explored, the wind turbine industry experience is perhaps the best jumping-off point for developing such data persistence approaches for marine data-model fusion problems. Wind turbines share many similar sub-systems to marine vessels, having structural systems, machinery systems, and electrical systems. They have periodic inspection reports from humans and need to understand their environmental exposure history when performing analysis of the recorded data. Additionally, there are many wind turbines in service offshore, with frequent inspection and a growing research community focused on these data sets. Further analysis and discussion of their approaches seems the most valuable way forward.

SECTION 4.4 CONCLUSIONS

A picture of current data persistence approaches has emerged by reviewing recent developments in 32 papers covering the structural updating community, the machinery monitoring community, the lifecycle assessment community, the offshore wind turbine community, and the digital thread community. The literature does not yet have a standardized approach for such persistence, but challenges around data cleansing, transfer, labeling, and interchange seem to be common. Information exchange standards have emerged in specific industrial sectors, but no common standard approach appears to have become established across multiple industries. Storing large amounts of data appears to be a secondary challenge at the moment and was not the focus of most of the papers. Future data-model fusion approaches will likely have to develop their own data persistence approach, picking and choosing from successful examples in related industries. In doing so, the offshore wind turbine industry appears to be a strong starting point, owing to high-level similarities between their systems and naval vessels.

SECTION 4.5 REFERENCES

[1] T. D. West and M. Blackburn, “Is Digital Thread/Digital Twin Affordable? A Systemic Assessment of the Cost of DoD’s Latest Manhattan Project,” *Procedia Computer Science*, vol. 114, pp. 47–56, 2017, doi: 10.1016/j.procs.2017.09.003.

[2] E. M. Kraft, “The Air Force Digital Thread/Digital Twin - Life Cycle Integration and Use of Computational and Experimental Knowledge,” presented at the 54th AIAA Aerospace Sciences Meeting, San Diego, California, USA, Jan. 2016. doi: 10.2514/6.2016-0897.

[3] C. Wincott and M. D. Collette, “Digital Twins: An Assessment of the State-of-the-Art,” presented at the ASNE TSS, Washington D.C., Jun. 2019.

[4] Í. A. Fonseca and H. M. Gaspar, “Challenges when creating a cohesive digital twin ship: a data modelling perspective,” *Ship Technology Research*, vol. 68, no. 2, pp. 70–83, May 2021, doi: 10.1080/09377255.2020.1815140.

[5] N. G. Pegg and S. Gibson, “Application of Advanced Analysis Methods to the Life Cycle Management of Ship Structures,” Defence Research Establishment Atlantic, Dartmouth, Nova Scotia, Canada, May 1997.

[6] T. J. Eccles, G. Ashe, and S. Albrecht, “The Achieving Service Life Program,” *Naval Engineers Journal*, vol. 122, no. 3, pp. 103–112, Sep. 2010, doi: 10.1111/j.1559-3584.2010.00275.x.

[7] M. J. Smith, T. Macadam, and J. R. MacKay, “Integrated modelling, design and analysis of submarine structures,” *Ships and Offshore Structures*, vol. 10, no. 4, pp. 349–366, Jul. 2015, doi: 10.1080/17445302.2014.937058.

[8] K. Li, Z.-Y. Yi, Y.-Y. Yu, and M. Chen, “A FRAMEWORK FOR INTEGRATED SHIP STRUCTURE LIFECYCLE MANAGEMENT,” *Journal of Marine Science and Technology*, vol. 26, no. 4, Aug. 2018, doi: 10.6119/JMST.201808_26(4).0005.

[9] D. Thomson and P. Renard, “The Digital Handover Shipyards as Producers of Life-Cycle Maintenance Models,” presented at the Computer and IT Applications in the Maritime Industries, Cortona, Italy, 2013. [Online]. Available: http://compit.hiper-conf.info/?page_id=9

[10] M. Cheliotis, I. Lazakis, and G. Theotokatos, “Machine learning and data-driven fault detection for ship systems operations,” *Ocean Engineering*, vol. 216, p. 107968, Nov. 2020, doi: 10.1016/j.oceaneng.2020.107968.

[11] P. Lu, H. Liu, C. Serratella, and X. Wang, “Assessment of Data-Driven, Machine Learning Techniques for Machinery Prognostics of Offshore Assets,” presented at the Offshore Technology Conference, May 2017. doi: 10.4043/27577-MS.

[12] V. Asalapuram, I. Khan, and K. Rao, “A Novel Architecture for Condition Based Machinery Health Monitoring on Marine Vessels Using Deep Learning and Edge Computing,” in *2019 IEEE International Symposium on Measurement and Control in Robotics (ISMCR)*, Sep. 2019, pp. C1-3-1-C1-3–6. doi: 10.1109/ISMCR47492.2019.8955729.

[13] F. Zhao, W. Yang, W. W. Tan, S. K. Chou, and W. Yu, “An Overall Ship Propulsion Model for Fuel Efficiency Study,” *Energy Procedia*, vol. 75, pp. 813–818, Aug. 2015, doi: 10.1016/j.egypro.2015.07.139.

[14] C. Capezza, S. Coleman, A. Lepore, B. Palumbo, and L. Vitiello, “Ship fuel consumption monitoring and fault detection via partial least squares and control charts of navigation data,” *Transportation Research Part D: Transport and Environment*, vol. 67, pp. 375–387, Feb. 2019, doi: 10.1016/j.trd.2018.11.009.

[15] F. Tillig, J. W. Ringsberg, W. Mao, and B. Ramne, “Analysis of uncertainties in the prediction of ships’ fuel consumption – from early design to operation conditions,” *Ships and Offshore Structures*, vol. 13, no. sup1, pp. 13–24, Apr. 2018, doi: 10.1080/17445302.2018.1425519.

[16] I. Lazakis, Y. Raptodimos, and T. Varelas, “Predicting ship machinery system condition through analytical reliability tools and artificial neural networks,” *Ocean Engineering*, vol. 152, pp. 404–415, Mar. 2018, doi: 10.1016/j.oceaneng.2017.11.017.

[17] N. S. Abbasian, A. Salajegheh, H. Gaspar, and P. O. Brett, “Improving early OSV design robustness by applying ‘Multivariate Big Data Analytics’ on a ship’s life cycle,” *Journal of Industrial Information Integration*, vol. 10, pp. 29–38, Jun. 2018, doi: 10.1016/j.jii.2018.02.002.

[18] B. Jeong, H. Wang, E. Oguz, and P. Zhou, “An effective framework for life cycle and cost assessment for marine vessels aiming to select optimal propulsion systems,” *Journal of Cleaner Production*, vol. 187, pp. 111–130, Jun. 2018, doi: 10.1016/j.jclepro.2018.03.184.

[19] C. Favi, F. Campi, M. Germani, and S. Manieri, “Using design information to create a data framework and tool for life cycle analysis of complex maritime vessels,” *Journal of Cleaner Production*, vol. 192, pp. 887–905, Aug. 2018, doi: 10.1016/j.jclepro.2018.04.263.

[20] K. Lu, X. Jiang, J. Yu, V. W. Y. Tam, and M. Skitmore, “Integration of life cycle assessment and life cycle cost using building information modeling: A critical review,” *Journal of Cleaner Production*, vol. 285, p. 125438, Feb. 2021, doi: 10.1016/j.jclepro.2020.125438.

[21] A. M. Ferrari, L. Volpi, D. Settembre-Blundo, and F. E. García-Muiña, “Dynamic life cycle assessment (LCA) integrating life cycle inventory (LCI) and Enterprise resource planning (ERP) in an industry 4.0 environment,” *Journal of Cleaner Production*, vol. 286, p. 125314, Mar. 2021, doi: 10.1016/j.jclepro.2020.125314.

[22] A. Koltsidopoulos Papatzimos, T. Dawood, and P. R. Thies, “An Integrated Data Management Approach for Offshore Wind Turbine Failure Root Cause Analysis,” presented at the ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering, Sep. 2017. doi: 10.1115/OMAE2017-62279.

[23] J. Helsen, G. De Sitter, and P. J. Jordaens, “Long-Term Monitoring of Wind Farms Using Big Data Approach,” in *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*, Mar. 2016, pp. 265–268. doi: 10.1109/BigDataService.2016.49.

[24] M. Martinez-Luengo, M. Shafiee, and A. Kolios, “Data management for structural integrity assessment of offshore wind turbine support structures: data cleansing and missing data imputation,” *Ocean Engineering*, vol. 173, pp. 867–883, Feb. 2019, doi: 10.1016/j.oceaneng.2019.01.003.

[25] D. van der Linden, G. De Sitter, T. Verbelen, C. Devriendt, and J. Helsen, “Towards an evolvable data management system for wind turbines,” *Computer Standards & Interfaces*, vol. 51, pp. 87–94, Mar. 2017, doi: 10.1016/j.csi.2016.12.001.

[26] R. Stevens, “Weaving a Digital Thread into Concept Design,” in *2020 IEEE Aerospace Conference*, Mar. 2020, pp. 1–7. doi: 10.1109/AERO47225.2020.9172812.

[27] S. Kwon, L. V. Monnier, R. Barbau, and W. Z. Bernstein, “Enriching standards-based digital thread by fusing as-designed and as-inspected data using knowledge graphs,” *Advanced Engineering Informatics*, vol. 46, p. 101102, Oct. 2020, doi: 10.1016/j.aei.2020.101102.

[28] J. Sousa, J. P. Mendonça, and J. Machado, “A generic interface and a framework designed for industrial metrology integration for the Internet of Things,” *Computers in Industry*, vol. 138, p. 103632, Jun. 2022, doi: 10.1016/j.compind.2022.103632.

[29] S. Gopalakrishnan, N. W. Hartman, and M. D. Sangid, “Model-Based Feature Information Network (MFIN): A Digital Twin Framework to Integrate Location-Specific Material Behavior Within Component Design, Manufacturing, and Performance Analysis,” *Integr Mater Manuf Innov*, vol. 9, no. 4, pp. 394–409, Dec. 2020, doi: 10.1007/s40192-020-00190-4.

[30] V. Singh and K. E. Willcox, “Engineering Design with Digital Thread,” *AIAA Journal*, vol. 56, no. 11, pp. 4515–4528, 2018, doi: 10.2514/1.J057255.

[31] T. Y. Pang, J. D. Pelaez Restrepo, C.-T. Cheng, A. Yasin, H. Lim, and M. Miletic, “Developing a Digital Twin and Digital Thread Framework for an ‘Industry 4.0’ Shipyard,” *Applied Sciences*, vol. 11, no. 3, Art. no. 3, Jan. 2021, doi: 10.3390/app11031097.

[32] K. Jagusch, J. Sender, D. Jericho, and W. Flügge, “Digital thread in shipbuilding as a prerequisite for the digital twin,” *Procedia CIRP*, vol. 104, pp. 318–323, Jan. 2021, doi: 10.1016/j.procir.2021.11.054.

CHAPTER 5 SURVEY OF DATA-MODEL FUSION TECHNIQUES

Authors: Rachel Bielski[1], Matthew Collette[2], Conner Goodrum[1], Michael Sypniewski[1]
1 – Martin Defense Group
2 – University of Michigan, Department of Naval Architecture and Marine Engineering

Date: December 2020

At a high-level, data-model fusion is an area of study concerned with integrating (or "fusing") data science and/or machine learning techniques with engineering modelling methods, while leveraging the advantages of both. Digital twins provide numerous opportunities for utilizing system-specific data in concert with engineering models, and as such, DMF presents a promising area that is both complementary and significantly relevant toward furthering the capabilities of digital twin technology.

We begin by defining DMF and presenting motivation for its study and application. Next, we detail a survey of existing data-model fusion techniques, where each method is described alongside a brief assessment of its appropriateness and applicability to different problem types. Finally, we discuss the application of DMF within the context of digital twins, both for current constructs as well as extensions to platform-level twins.

SECTION 5.1 BACKGROUND & MOTIVATION

Recent advances in computational capabilities have led to improvements in both physics-based models and data science techniques. However, these advancements have generally been studied and applied separately, limiting their potential positive impacts. DMF is a discipline concerned with the integration of governing equations[1] and data science or machine learning techniques, with the intent of leveraging the advantages of both. Before discussing DMF in more detail, let's first review how these two different approaches (model-based and data-driven) have been used traditionally.

Model-based approaches are familiar to the realm of engineering analysis. These methods typically involve implementing models in the form of mathematical functions to describe the physics of the system states and failure modes. These governing equations incorporate physical understanding of the system into the estimation of state and how it will behave based on any given input. While powerful tools, model-based approaches also have computational drawbacks, which have been addressed traditionally by trading fidelity for efficiency. Low-fidelity representations strategically ignore behaviors found in the real-world context of a system. Examples include the simplifications used in Hooke's law, linear wave theory, and the exclusion of air resistance in many applications of the equations of motion. On the other hand, high-fidelity models offer more detail at increased cost, either monetary or computational. For example, Computational Fluid Dynamics (CFD) analysis offers a very robust simulation of fluid flows but quickly becomes computationally infeasible for large search spaces. Regardless of fidelity, model-based approaches are generally applied during the system design phase and not during post-production operation. As such, they are rarely tailored to the specific instance of the system being analyzed.

[1] The term "governing equations" was chosen over the more commonly used "physics-based models" to remain inclusive of behaviors that are not strictly physics-based, such as the laws of economics [1,2,3].

While data-driven approaches have long been the study of statisticians and data scientists, they have also become increasing popular in the engineering world, at least in part, for their advantages in terms of efficiency and flexibility. These methods encompass a wide array of regression and classification techniques, from simple polynomial curve fitting to deep convolutional neural networks. Simply put, they leverage statistical correlations within data to make predictions, which may be in the form of state estimates or anomaly detection. Although the training process may require significant computational resources, the trained models often have much less computational overhead than their model-based counterparts. Since they only rely on statistical differences within data and not on the principals of physics or engineering, data-driven approaches are extremely attractive as flexible, off-the-shelf solutions in many cases. While it comes with many advantages, the sole reliance of machine learned models on observed data has notable drawbacks. For one, the models do not typically generalize well, meaning that when presented with samples beyond those experienced during training, they do not offer reliably accurate predictions of the system's behavior in that space. A similar issue is also experienced in data sparse environments, which are commonly found in engineering applications. Data sparsity is not caused solely by a lack of data but also by high dimensional state spaces (i.e., large input or output vectors) and unbalanced targets (e.g., failure caused by a rare event). In engineering settings, it may not be effective or even feasible to acquire the amount of data needed to fuel a purely data-driven model, which necessitates new innovative methods for handling these scenarios.

As stated above, data-model fusion techniques integrate these two approaches to leverage their advantages and offset their drawbacks, but how this is handled varies across different DMF techniques. Raissi uses governing equations to constrain the space of admissible solutions to a manageable size [4]. Others have based their machine learned model on physical laws (i.e. creating a relationship between the model, the training set data, and a physical governing equation) and have demonstrated accurate estimates with sparse datasets and low relative cost [14]. DMF techniques, which are discussed in more detail in the following section, have been shown to provide more accurate state forecasting than singular models and have been successfully applied for predictive maintenance and health diagnostics. Many areas of autonomy could also benefit from improved forecasting capabilities. In a statement to a defense writer's group in July 2020, DARPA Acting Director Peter Highnam conceded that there is still a major issue with the robustness of AI-based systems, neural nets, and reinforcement learning [7].

Before moving onto current techniques, note that DMF is commonly confused with the concept of sensor fusion or data fusion. Sensor fusion is the integration of data from multiple sources or multiple types of sensors to reduce the uncertainty of the parameter being measured [8]. A simple example of this is the coordination of cameras and LIDAR sensors on semi-autonomous vehicles that create more accurate awareness of the surrounding environment than would be presented from just one data source. While fusing sensor data in this way is of great interest in digital twin research, it is different from the concept of DMF discussed throughout this report.

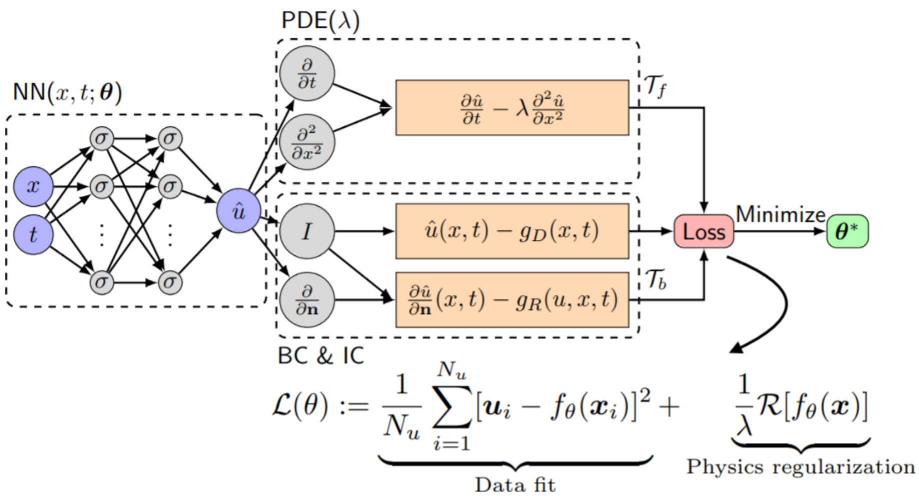
SECTION 5.2 SURVEY OF EXISTING TECHNIQUES

A search was conducted to identify current efforts that merge data-driven methods and model-based approaches. The explored techniques are summarized in this section, and while not an exhaustive list, they provide an adequate reflection of the current work in the space. Methodologies that fit within the definition of data-model fusion were included even without the explicit mention of the term.

SECTION 5.2.1 PHYSICS-INFORMED NEURAL NETWORKS

Definition: Neural networks that are trained to solve supervised learning tasks while respecting any given laws of physics described by general nonlinear partial differential equations.

An artificial neural network is a computational modeling tool that mimics the ability of neural systems to capture and represent complicated, multi-dimensional, linear and nonlinear relationships through a layered structure of units [9]. Physics-informed neural networks utilize built-in governing laws to aid decision-making with incomplete information and generate a space of admissible solutions when data is sparse or high cost. In comparison, traditional neural networks are often limited by their need for large amounts of training data. In the proposed methodology in [10], a problem defined by a non-linear partial differential equation has its complex-valued solution approximated by a deep neural network. The scientific computing technique of automatic differentiation is used to evaluate the derivative of the neural network with respect to both the input coordinates and physical model parameters. Since the physics of a given problem can be described by differential equations and these constraints help avoid over-fitting, this technique can be trained with smaller datasets. However, for high dimensional problems (as opposed to those with just one or two spatial dimensions) the requirement for many collocation points to enforce the physical constraints may introduce a severe computational bottleneck.



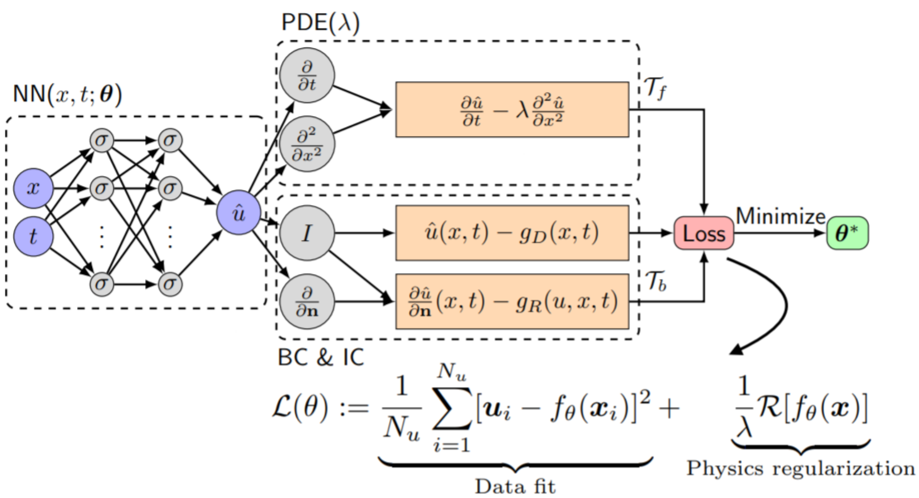


Figure 16. Physics is explicitly imposed by constraining the output of conventional neural architectures with weak inductive biases. Governing equations represented by Partial Differential Equations (PDE) add an element of regularization to the output of the analysis [32]

SECTION 5.2.2 PHYSICS-BASED MACHINE LEARNING

Definition: The implementation of physics-based models to ensure that predictions made via machine-learned methods enforce physical constraints of a system or environment.

Physics-Based Machine Learning is a technique used for applications where the goal is to predict high dimensional output quantities of interest. Its development was motivated by the need for strong predictions grounded by physical constraints from relatively sparse data. In [11], low-dimensional approximations of a high dimensional model are created using proper orthogonal decomposition. Then machine learning methods use these low-dimensional snapshots as training data to build a new, reduced-order model that maps inputs to the outputs of the original high-fidelity model. The fusion of physical parameters come from data-driven models learning the *operators* of the reduced models, which contain information about the dynamics of the system of interest. Systems in science and engineering that respond to inputs with physical fields or quantities are considered in the work. Importantly, the type of machine learning model used can be varied and the limitations of each should be considered with choosing an appropriate strategy for a problem. Later work by Swischuk in [12] demonstrates that the physics-based machine learning is able to accurately make predictions and enforce important physical constraints where the operators of discretized governing equations are unknown or too complex. Svrivastava affirmed that fusion techniques effectiveness at making predictions even with little or incomplete data [13].

SECTION 5.2.3 PHYSICS-BASED LEARNING MODELS

Definition: A Generic Learning Model (GLM) and training data set that is complemented with physical behaviors and constraints inherent in physics-based intermediate models.

The Physics-Based Learning Models (PBLM) technique developed by Weymouth et. al requires the use of a generic learning model (GLM), a fast physics-based intermediate model, and a small set of high quality experimental or computational training data. The method expands traditional GLMs that may require prohibitively large sets of data for complex problems and contain no knowledge of the system outside of the data by cross-referencing predictions from semi-empirical models (like physical governing equations) in order to make predictions. This technique expands upon previous work for maneuvering predictions in a variety of ship hydrodynamics and ocean engineering problems. PBLM obtains significantly improved prediction accuracy as compared with traditional non-fusion methods while reducing data dependence and over-fitting. Recent studies on seakeeping prediction also suggest that physics-based model predictions improve results even in data rich contexts as compared to a pure machine learning model [15].

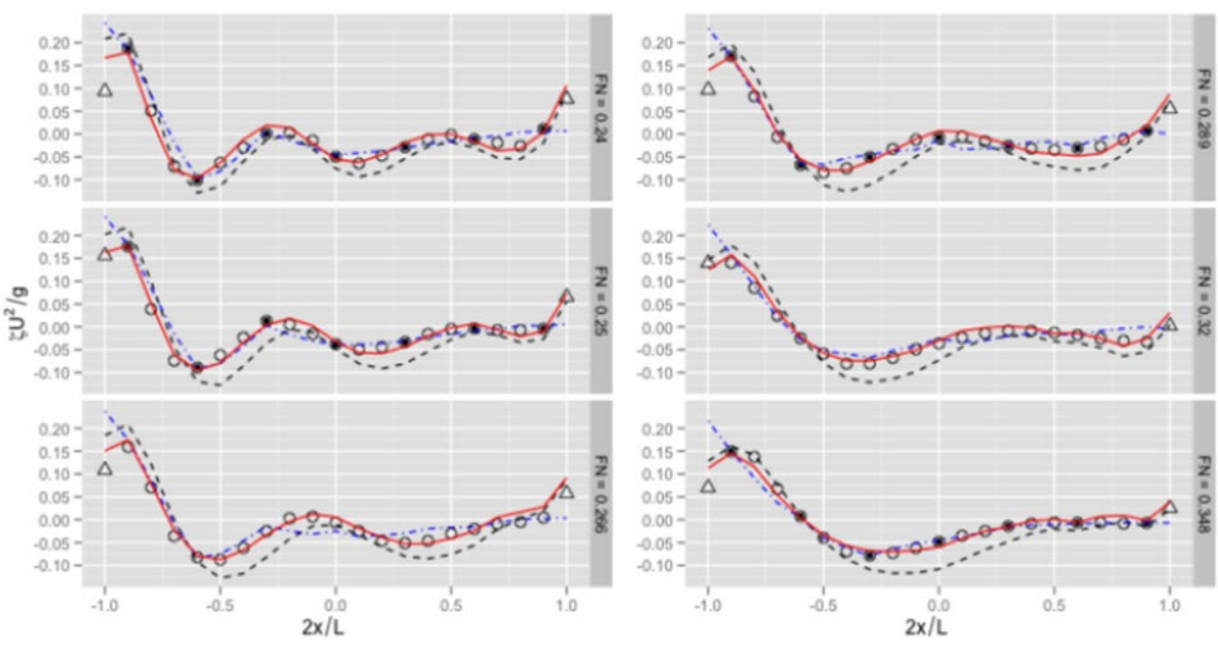


Figure 17. Waterline elevation profile data and predictions for a ship hull over six Froude numbers. PBLM predictions (red line) obtained significantly increased accuracy to modeling test data (hollow circles) as compared to GLM predictions (blue)[14]

SECTION 5.2.4 FUSION PROGNOSTIC FRAMEWORK

Definition: A prognostic framework that incorporates data-driven predictions into a particle filtering (PF) learning structure, such that predicted future measurements from the data method have continuously updated system parameters from the PF approach which results in reliable state forecasting.

This framework was developed to aid in applications that require system state estimation and forecasting, and whose forecasts must be reliable and accurate to schedule maintenance and avoid critical failure. Data-driven methods use pattern recognition to detect changes in systems but rely on accurate and large historical datasets, while model-based methods use models or functions to describe a system and techniques like PF to infer state variables where data is unavailable. This fusion technique integrates both methods for a comparatively more effective prognostic framework. First, a data-driven predictor is trained from historical data from similar systems, and then tuned using available data from the system of interest. This data-driven predictor can be trained using recursive learning algorithms and then fed into a PF learning model, which updates the predictor's model parameters as new information becomes available. Thus, physical understanding of the system and the current conditions are embedded into state estimates and forecasting. When implemented in a state prediction study, this DMF technique outperformed traditional data-driven and particle-filtering based approaches. This technique also improved upon the transparency of the pure data-driven method, meaning it may be more suitable for applications where forecast reasoning transparency is required (e.g. earthquake prediction or stock market forecasts). Depending on the application, the differences between the two specific components the data-driven and degradation model may be large and, in that case, must be determined and reconciled [16,17].

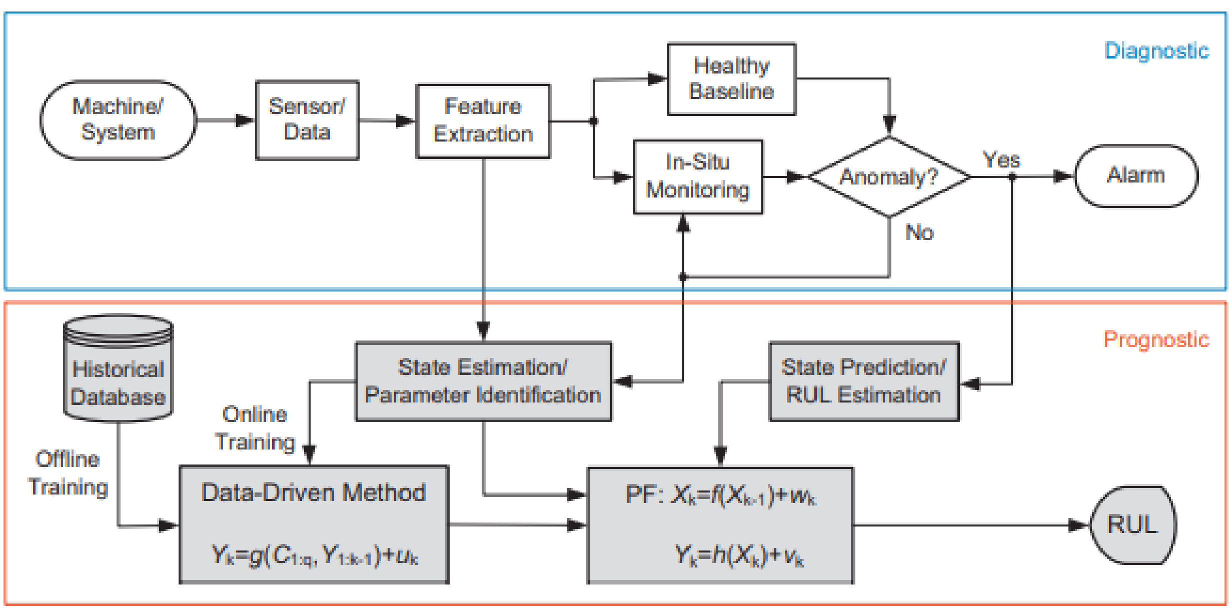


Figure 18. A schematic diagram of system conditioning monitoring and prognosis. The DMF technique is utilized once the process enters the prognosis stage

SECTION 5.2.5 BAYESIAN MODEL LEARNING

Definition: Framework in which Bayesian networks synthesize real world data with common underlying structural models, fusing multiple types of evidence and improving underlying models for better future forecasts.

Bayesian inference is a method of statistical inference in which Bayes' theorem is used to update the probability for a hypothesis as more evidence or information becomes available. A Bayesian network is a probabilistic graphical model that represents a set of parameters and their conditional dependencies, which makes them able to incorporate different types of interdependent information into a predicted outcome. It has proven to be a useful tool in data-sparse problems, where the parameters of a model may be unknown. Physics-based equations and prediction models related to observable outcomes can be integrated into Bayesian networks, which can then update beliefs in the underlying parameters in the models to better reflect a real-world system. Bayesian networks have been studied extensively in literature which applies learning methods to several marine loading and fatigue applications. Collette et. al (*A Bayesian approach for shipboard lifetime wave loading spectrum*) reveals a major advantage of the Bayesian method "is that it enables the inclusion of already-generated design knowledge as prior information. Therefore, even with limited life cycle data, the Bayesian approach can still provide a reasonable prediction of future performance." In a preceding study of marine structures subject to fatigue, networks have been used to update crack occurrence and length prediction for decision support [18,19,20,21].

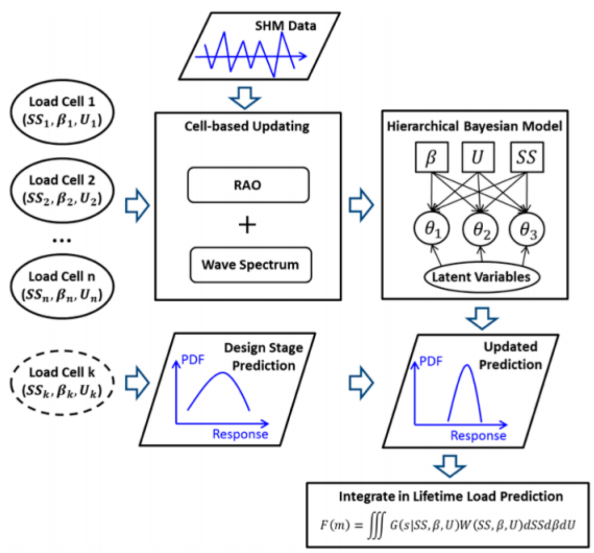


Figure 19. A proposed lifetime load uploading procedure using Bayesian models [19]

SECTION 5.3 APPLICATIONS

As demonstrated by the survey, there is already great interest in developing methods for integrating data-driven and physics-driven techniques for engineering applications. This section describes how DMF can be implemented in real-world applications and bring added value to a system. First, an overview of the value of DMF techniques is given, followed by real-world applications that already utilize fusion or present good use cases for the future. Examples from the maritime industry are described, followed by examples of fusion in other sectors.

The studies which looked to verify the effectiveness of a data-model technique often look to small-scale experiments that modeled a real-world system or structure and measured how well the new technique could make state estimates or remaining useful life (RUL) estimates and compared results to traditional models. The positive results of these studies have important real-world implications. The ability to accurately predict the RUL of a lithium-ion battery can be extended to the RUL of another simple component of a more complex system. Improved prediction of crack propagation would be hugely valuable for any civil engineering or naval architecture structure subject to regular fatigue forces.

For digital twins – whose useful applications include RUL assessments, maintenance planning based on load history, early damage detection and shutdown prevention, inspection support, and design feedback and improvements, as identified by Report 1.2 – the value of fusion techniques are very attractive. For high value, high complexity assets, the accuracy of the predictions made using their twins is extremely important. However, the computational cost of making the assessments or predictions must remain low enough to justify the benefits. Results of DMF studies have demonstrated that fusion techniques may be a key component of improving twin output. In a discussion of successful fusion techniques, Raissi states that "specific applications that can readily enjoy these benefits include, but are not limited to, data-driven forecasting of physical processes, model predictive control, multi-physics/multi-scale modeling and simulation" [1].

Within the scope of maritime technology, there are multiple ongoing efforts that could benefit from the implementation of DMF techniques. In a bid to develop advanced condition-based maintenance management processes for ship machinery, Japan's NYK Group moved forward with research plans that would install sensors on ship engines and steam turbines and share data with the classification society and machinery manufacturers in real time. The goal is to collect detailed operational data such as vibration and temperature of bearings, which will inform operators of the condition of the engine. Information on engine condition will be used in turn to inform predictions of machinery failure and remaining useful life (RUL) assessments. Moving towards a more autonomous future, the U.S. Navy is funding $2.7 billion to develop and field unmanned platforms over the next five years [22], and the establishment of condition-based maintenance systems will support the development of autonomous vessels [23]. DMF may be a critical component of realizing those goals.

Another representative maritime example is Askelos' digital twin of Shell's Bongo Main FPSO. The twin is a structural, physics-based model of the offshore facility located 120 km southwest of the Niger Delta in Nigeria at a depth of more than 1000m. It represents the entire physical counterpart in exact detail and is updated with loading conditions and inspection data on a regular basis, providing the ability to carry out structural assessments based on its current condition. Akselos was selected as a partner by Shell Nigeria Exploration and Production Company in pursuit of key operational objectives, such as identifying areas for priority inspection, reducing personnel onboard the asset, and reducing need for physical inspection where possible. DMF techniques provide new tools for industry to protect their high value assets and extend their useful lives [24].

The benefits of DMF can also be extended to different types of digital twins. Its strength in forecasting is highly applicable when considering fleet operations, maintenance, and logistics. There is an entire field of study dedicated to the understanding and optimization of production and transportation systems, whose increased efficiency can represent significant economic benefits. Several literature reviews of design and planning methods of platform problems have been conducted in pursuit of identifying current shortcomings in these methods. Colledani et. al studied various modeling techniques for products, process chains, and entire manufacturing systems and noted that "the main drawback is that existing models generally consider products, processes and production systems separated from one another… none of the existing works seems to be able to jointly represent products, processes and production systems data, information and knowledge, satisfying all the requirements" of each of those system levels [28]. In his dissertation on system-of-systems design, Frommer [29] identifies the platform problem as one that is large in scope, involving many entities with varying capabilities whose resources are spread out but connected through information, and whose individual parts are capable of independent operation. His work is focused on how individual assets can be designed for high performance levels but that will also form an "optimal fleet whose capability is greater than the sum of its parts" [29]. Given their dynamic and complex nature, fleets are difficult to characterize and model, especially given the various and changing relationships between their subsystems. Baykasoglu goes a step further and breaks intermodal transportation fleet planning problems down not just into separate components, but also into strategic, tactical, and operational decision-making levels [30]. The interdependencies among the sub-problems must be addressed within a fleet architecture, and, therefore, the sub-problems cannot be considered as lone entities. The advantages of DMF techniques could help address the complexity of the problem of fleet modeling and decision-making, as well as the underlying need to keep the solution computationally affordable.

An example of how model fusion and utilization of various learning data-driven methods could inform maintenance planning at a platform level comes from work done by the University of Michigan Data Science Team with the City of Detroit's Operations and Infrastructure Group [32]. The research sought to uncover and understand the existence of patterns and trends in the significant and highly complex maintenance data of the fleet of vehicles owned and operated by the city of Detroit. The complexity arose "from inter-relationships between vehicle type, system repair type, and time (both absolute time and vehicle lifetime)". Tensor decomposition techniques were used to discover temporal patterns in vehicle maintenance, and then differential sequence mining and neural network models were used to predict maintenance sequences with demonstrated success. The research recognized the unique problem planning and decision-making for a fleet of vehicles and, more generally, sub-systems.

SECTION 5.4 FUTURE CONSIDERATIONS

Data-model fusion is a promising concept that deserves further study and investment. This report described DMF and presented a literature review of existing fusion techniques, demonstrating how the integration of different modeling methods can be used to make more accurate, robust, or efficient predictions than either approach used in isolation. Future research into DMF may be focused on creating novel techniques from different types of models or studying how it could be utilized in new digital twins. Increased interest in optimizing platform and fleet-level systems, enabled by improved computational capability of the last decade, also represents a promising future area of application for DMF. For naval efforts, future work can utilize current techniques and system information to build more robust predictive models to predict demand, maintenance costs, and vehicle downtime (repair duration), and to assess maintenance effectiveness.

SECTION 5.5 REFERENCES

[1] Gogas, P., Papadimitriou, T., Matthaiou, M. *et al*. Yield Curve and Recession Forecasting in a Machine Learning Framework. *Comput Econ* **45,** 635–645 (2015)

[2] Mullainathan, Sendhil, and Jann Spiess. 2017. "Machine Learning: An Applied Econometric Approach." *Journal of Economic Perspectives*, 31 (2)

[3] Yu, Lean & Huang, Wei & Lai, Kin Keung & NAKAMORI, YOSHITERU & Wang, Shouyang. (2007). Neural Networks in Finance and Economics Forecasting. International Journal of Information Technology & Decision-making. 06. 113-140.

[4] Raissi, M., Perdikaris, P., Karniadakis, G. 'Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations'. *Cornell University*. (2017)

[5] "Pros and Cons of Predictive Analysis." *Georgetown University Online*, 28 Sept. 2018, scsonline.georgetown.edu

[6] Gaul, Vishwa, and Vineet Kumar. "Predictive Analytics Market Share, Trends & Growth: Forecast - 2027." *Allied Market Research*, Allied Market Research, July 2020, www.alliedmarketresearch.com

[7] Hitchens, Theresa. "DARPA Eyes More 2022 Funds To Improve AI Reliability." *Breaking Defense*, 30 July 2020, www.breakingdefense.com

[8] Klein, Lawrence A. *Sensor and Data Fusion: A Tool for Information Assessment and Decision-making*. SPIE Press, 2012

[9] Zou, Jinming & Han, Yi & So, Sung-Sau. (2009). Overview of Artificial Neural Networks. Methods in molecular biology (Clifton, N.J.). 458. 14-22. 10.1007/978-1-60327-101-1_2.

[10] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686-707.

[11] R. Swischuk et al., Projection-based model reduction: Formulations for physics-based machine learning, Computers and Fluids (2018)

[12] Swischuk, R 2019, 'Physics-based machine learning and data-driven reduced-order modeling', Masters Thesis, Massachusetts Institute of Technology, Cambridge.

[13] Srivastava, Ankur, et al. "A Hybrid DMF Approach to Calibrate a Flush Air Data Sensing System." AIAA Infotech@Aerospace 2010, 2010, doi:10.2514/6.2010-3347.

[14] Weymouth, Gabriel & Yue, Dick. (2014). Physics-Based Learning Models for Ship Hydrodynamics. Journal of Ship Research. 57. 10.5957/JOSR.57.1.120005.

[15] Collette, Matthew. "Data-Model Fusion For Digital Twins on Crewed and Crewless Vessels." Business Finland Lecture Series, 2 October 2020, Helsinki FI. Lecture.

[16] A. Castelletti, S. Galelli, M. Restelli, R. Soncini-Sessa, 'Data-driven dynamic emulation modelling for the optimal management of environmental systems', Environmental Modelling & Software, Volume 34, 2012, Pages 30-43, ISSN 1364-8152.

[17] Liu, J., Wang, W., Ma, F., Yang, YB, & Yang, CS. (2012). A data-model-fusion prognostic framework for dynamic system state forecasting. Engineering Applications of Artificial Intelligence, 25(4), 814-823.

[18] Groden, Mark D. and M. Collette. "Bayesian networks for model updating inspection support of marine structures subject to fatigue." (2015).

[19] Jiandao Zhu & Matthew Collette (2017) A Bayesian approach for shipboard lifetime wave load spectrum updating, Structure and Infrastructure Engineering, 13:2, 298-312, DOI: 10.1080/15732479.2016.1165709

[20] Mark Groden and Matt Collette. Fusing fleet inservice measurements using Bayesian networks. Marine Structures, 54:38–49, July 2017.

[21] J. Zhu and M.D. Collette. Updating Structural Engineering Models with In-Service Data: Approaches and Implications for the Naval Community. Naval Engineers Journal, 127(1):63–74, March 2015.

[22] Larter, David B. "5 Things You Should Know about the US Navy's Plans for Autonomous Missile Boats." *Defense News*, 14 Jan. 2020, www.defensenews.com

[23] "NYK Starts Verification of Advanced Condition-Based Maintenance for Autonomous Ships." *NYK*, 19 Nov. 2019, www.nyk.com

[24] Adelman, David. "Akselos Deploys Digital Twin of Shell's Bonga FPSO." *Business Wire*, Berkshire Hathaway, 1 Sept. 2020, www.businesswire.com.

[25] Chen, M., Shi, L., Kelly, R. *et al.* Selecting a single model or combining multiple models for microarray-based classifier development? – A comparative analysis based on large and diverse datasets generated from the MAQC-II project. *BMC Bioinformatics* **12,** S3 (2011)

[26] Das, Shiva K et al. "Combining multiple models to generate consensus: application to radiation-induced pneumonitis prediction." *Medical physics* vol. 35,11 (2008): 5098-109. doi:10.1118/1.2996012

[27] Vettoretti, Martina et al. "Addressing practical issues of predictive models translation into everyday practice and public health management: a combined model to predict the risk of type 2 diabetes improves incidence prediction and reduces the prevalence of missing risk predictions." *BMJ open diabetes research & care* vol. 8,1 (2020): e001223. doi:10.1136/bmjdrc-2020-001223

[28] Colledani M., Terkaj W., Tolio T., Tomasella M. (2008) Development of a Conceptual Reference Framework to Manage Manufacturing Knowledge Related to Products, Processes and Production Systems. In: Bernard A., Tichkiewitch S. (eds) Methods and Tools for Effective Knowledge Life-Cycle-Management. Springer, Berlin, Heidelberg.

[29] Frommer, Joshua B. "System of Systems Design: Evaluating Aircraft in a Fleet Context Using Reliability and Non-Deterministic Approaches." *Purdue University* , PURDUE UNIVERSITY GRADUATE SCHOOL, 2008.

[30] Adil Baykasoğlu, Kemal Subulan, A. Serdar Taşan & Nurhan Dudaklı (2019) A review of fleet planning problems in single and multimodal transportation systems, Transportmetrica A: Transport Science, 15:2, 631-697

[31] Erikstad, Stein. (2017). Merging Physics, Big Data Analytics and Simulation for the Next-Generation Digital Twins.

[32] Gardner, Josh, et al. "Driving with data: Modeling and forecasting vehicle fleet maintenance in Detroit." arXiv preprint arXiv:1710.06839 (2017).

[33] Perdikaris, Paris. “Physics-informed deep learning.” University of Pennsylvania, 28 July 2020, 15th National U.S. Congress on Computational Mechanics, Austin TX. Lecture.

CHAPTER 6 SURVEY OF DECISION-MAKING TECHNIQUES

Authors: Jason Provancher [1], Matthew Collette[2], Brian Cuneo[1], Conner Goodrum [1], Jackie Jones [1], Daniel Snyder[1], Michael Sypniewski [1]
1 – Martin Defense Group
2 – University of Michigan, Department of Naval Architecture and Marine Engineering
Date: November 2020

This section outlines the major branches of decision-making approaches developed to date, provides some examples of how they might interact with digital twins developed on this project, and forms a reference for engineers working on demonstrator systems. The report addresses the following three specific objectives:

1. To provide an overview of decision-making methodologies, techniques, tools, and references to further resources.
2. To serve as an aid to engineers/scientists developing decision-making strategies through guidance on selecting appropriate methodologies and tools.
3. To demonstrate decision-making strategies through a varied set of examples.

SECTION 6.1 OVERVIEW

As was shared in Chapter 2 A Standardized Definition and Preliminary Taxonomy for Digital Twins, decision-making should not be considered a requisite component of a digital twin since the digital twin could exist and function without the Decision Maker. In regard to decision-making, our definition views digital twins as supporting, but not encompassing, the decision-making processes. However, for many roles envisioned for digital twins, the twin will be interfacing with either on-board or off-board decision makers for the platform. Therefore, to better understand the decision-making needs a digital twin should support we have provided this summary of decision-making approaches.

Decision-making is a critical aspect of how agents, whether they be human, robot, or otherwise, effect a desired change in their environments. Decision-making spans numerous fields of research, from traditional decision theory to autonomous systems trained to choose actions through reinforcement learning algorithms. In its most simple incarnation, decision-making involves an agent that chooses amongst a set of possible actions. *How* a decision is reached varies widely depending on the agent, the type of decision-making problem, and the environment in which the agent acts. An agent may simply react to external stimuli in some predefined manner. In more complex problems, an agent may plan multiple sequences of actions, predicting the outcome of those actions, and weighing uncertain risks and rewards before selecting a final course of action.

This survey focuses on introducing common methods and important considerations when developing an algorithmic decision-maker, through:

Section 6.2: Questions to ask about the decision-making problem when selecting an approach.

Section 6.3: Representations and abstractions of decision-making problems.

Section 6.4: Common approaches and methodologies to building a decision-maker and/or solving decision-making problems.

Section 6.5: Examples implementations of decision-makers on a diverse set of systems.

The survey is written with the intention that a newcomer to decision-making can understand important considerations when designing their own agent, as well as the ability to select a preliminary approach with resources to pursue a more in-depth solution. We cover a wide range of decision-making approaches and representations that can be applied to building an autonomous decision-maker, from mathematical optimization and reinforcement learning to goal-driven autonomy and case-based reasoning. Additionally, the examples serve as a starting point for the development of future decision-makers based on the approaches covered in this survey.

SECTION 6.2 CHARACTERIZE THE DECISION

This section details a list of questions that are used to define the attributes of the domain and the decision that needs to be made. The questions cover the type of environment an agent is interacting with, such as if the environment is static or dynamic, the uncertainty that is present in the environment, and whether the agent has information on the entire environment or only what it can perceive at a given time step. The questions also address the effects of a decision, determining if a decision can be revisited or if it is a one-shot decision and if the decision space is continuous or there is a set number of actions available to the agent. These questions will be used to inform which type of decision-making method should be used for a given use case. These questions also highlight the relationship between decision-making frameworks and digital twins which may provide information to the decision-making frameworks. The capabilities of the digital twin, and the types of information it can provide, will help define the applicable decision-making frameworks for each problem.

SECTION 6.2.1 STATIC OR DYNAMIC ENVIRONMENT?

First, the environment an agent is operating in needs to be classified as either static or dynamic based on the source of the variables that may change them. For a static environment, only the agent's actions can change the environment. It is important to note that dynamic movement of the agent does not qualify the environment it operates in as dynamic. For example, in the case of robotic navigation in an office with no moving objects or people, the environment – the office – would still be considered static even if the robot's movement causes changes to it. There is nothing changing inside the environment except for changes the agent makes to its own location or those of objects it interacts with. Another example would be speech recognition. The context for recognition is not changing, only the audio that is being input.

In a dynamic environment, changes happen in the environment out of the agent's control. Having other agents in an environment makes it dynamic because they will cause changes outside of the agent's control. For example, robotic navigation in an office with people. Compared to the static office with no other agents (people), there are now changes happening in the environment that are outside of the agent's control, such as people moving between offices or moving objects in the environment. Understanding if an agent will operate in a static or dynamic environment is crucial to choosing a proper decision-making algorithm.

SECTION 6.2.2 SEQUENTIAL OR ONE-SHOT (EPISODIC) DECISION-MAKING?

Beyond understanding if an environment is static or dynamic, it must be known how an agent's decisions and actions will affect future decisions. The time horizon for how far out the agent makes decisions must be understood to properly design how an agent will make decisions. To understand the time dependent impacts on decisions, the agent's experience is divided into atomic "episodes" where each episode consists of the agent perceiving and then performing a single action.

In sequential decision-making, decisions are made in a series. Current decisions affect future decisions or rely on previous ones. Most environments (and agents) are sequential. For example, path planning for robotic navigation. In order to reach its goal, the agent will plan out a path to take from A to B. This path consists of a series of smaller decisions where each decision feeds into the next.

For one-shot/episodic decision-making, the decision is one that will never be possible to revisit. The choice of action in each episode depends only on the episode itself. The agent does not look to optimize for future decisions. An example would be expert advice systems. In this case, an episode is a single question and answer given by the agent. The agent's answer does not need to plan for future question because the questions are independent episodes.

SECTION 6.2.3 FULLY VS PARTIALLY OBSERVABLE ENVIRONMENT?

The final question related to the environment the agent is acting in refers to the agent's ability to perceive the environment. In many real-world environments, it will not be possible for the agent to have perfect and complete perception of the state of the environment. The agent will make observations of the state of the environment, but these observations may be noisy and provide incomplete information. Please note: the "complete state" of an environment refers to information that is relevant to the agent (that is to say, your robotic agent may not care that the walls of the room are blue when it is path planning).

In a fully-observable environment, the agent has sensor(s) that can sense or access the complete state of an environment at each point in time. There is no need to maintain the internal state to keep a history of the world. One example would be a chess board. If the agent can access the state of all pieces at once, this is fully observable. Another example would be an image which is being processed with image recognition software. The image is fully observable even if it has not been classified or categorized.

In a partially-observable environment, the agent has sensor(s) that can sense or access part of the state of an environment at each point in time. The agent may utilize a memory system in order to compile more information about the environment. This problem is also referred to as the problem of "incomplete perception," "perceptual aliasing," or "hidden state." Often, if other agents are involved, their intentions are not observable, but their actions are. For example, a robotic agent might observe whether it is in a corridor, an open room, a T-junction, etc., and those observations might be error prone or incorrect because on the robot's sensing capabilities.

SECTION 6.2.4 DETERMINISTIC OR NON-DETERMINISTIC ENVIRONMENT?

Where different aspects of the environment affect which decision-making strategy is deployed, consideration must also be taken as to if there is uncertainty in both the environment and the outcomes of which decision is made. In a deterministic environment, any action taken by an agent has a single guaranteed effect. There is no element of uncertainty or randomness in the environment. Given a particular input state, it will always produce the same output. For example, in a Tic-tac-toe game, the outcome of the game is determined by who gets three symbols in a row. There is no randomness to the outcome of the decision. The outcomes of the other agent's moves may be unknown, however that is captured in whether the environment is static or dynamic. If the agent decides to place its mark in location X, the mark will always land at location X and not another random box.

In a non-deterministic environment, the environment contains some inherent randomness. Given an input state, it will not always produce the same output. This randomness can be stochastic/probabilistic or purely random. Uncertainty could also come from lack of a good environment model, or lack of complete sensor coverage. For example, a robot in the outdoors. The robot may attempt the same action twice, such as solar charging. In one instance it may succeed because it has access to sunlight. In another, it may fail because there was cloud cover. Unlike the Tic-tac-toe game, the outcome of the decision to charge can have multiple outcomes, the agent took the same action in both scenarios, "solar charge," but the action resulted in different outcomes: success or failure to charge.

SECTION 6.2.5 DISCRETE DECISIONS VS CONTINUOUS DECISIONS?

The final question for determining which type of decision-making strategy to implement refers to the quantity of actions that the agent can choose from in the environment. For discrete decisions, only a finite number of actions can be performed within the environment. The agent must choose from one of these specified actions. Time can also be quantified in fixed steps as opposed to a continuous flow. For example, a game of chess. This has a set number of moves the agent can make and time proceeds where each turn is a discrete step.

For continuous decisions, an infinite number of actions can be chosen from to be performed within the environment. This does not necessarily mean the agent has full control over the environment, only that it is not limited in its options for actions to take. Time moves continuously with new data constantly being processed. An example would be setting a course with a steering wheel. There are an infinite number of setting for the angle of the turn. The agent can be designed to limit the angles of the wheel that can be selected; however, the actual settings are infinite.

SECTION 6.3 STATE-SPACE REPRESENTATIONS AND SPECIFICATIONS

In order to properly frame a decision-making problem, the agent and environment must be structured to properly implement the algorithms that will be defined in section 6.4. This chapter will build the definitions to structure the state space where an agent will act. The definitions in this chapter are developed to design the state space while considering the questions described in section 6.1. All domains can be described with some combination of the following variables:

- V – The vocabulary used to describe states and actions and events.
- S – The State Space of an Agent.
- A – The Actions available at a given State.
- T – The Transition properties between States.
- C – The Constraints on the State Space.
- O – The Observations of an Agent.
- R – The Reward associated with a given State.
- G – The Goals of the Agent.
- K – The Knowledge the Agent has of the State Space

SECTION 6.3.1 VOCABULARY (V)

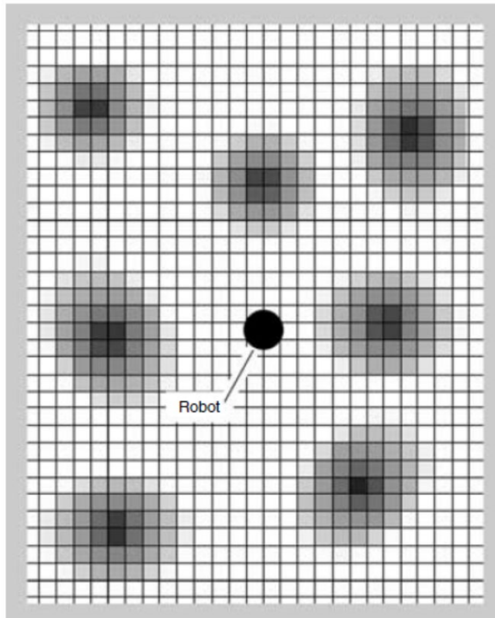
All automated decision processes require some encoding of the environment that we can interact with programmatically. The choice of vocabulary must consider two main factors: what information can you extract from the environment itself, and what kinds of decisions will you make with this abstraction. This section considers some commonly used abstractions for decision-making, as well as the types of domains where each abstraction may be used.

SECTION 6.3.1.1 OCCUPANCY GRID

In a spatial environment, neighboring states are locations that are related by the Euclidean distance between them. This is true for tracking the location of an object or an obstacle in the real world. We can model spatial environments using occupancy grids which discretize the observed space using a grid. Within the grid, cells are identified by an index for their location along each axis. It then iterates over the grid and calculates the probability of occupation for each cell. The resulting map of the space is the representation of the state at that time.

For example, consider the problem of a robot trying to localize itself. Suppose it has an initial location that it knows as fact and has access to a map of the room split into a grid as described above. As it moves through the room, the robot receives information about its change in location from a series of sensors such as an accelerometer, wheel rotation sensor, distance sensor, and/or cameras. At any given point in time, all of these sensors provide information about how much and in which direction the robot has moved since the last time step. Each sensor comes with inherent error, however, and this error compounds over time. Many different mathematical approaches exist to solve this problem and predict the location of the robot with an assigned probability (e.g., a Kalman filter). By providing the probability at each point in the grid and displaying it, either in a visual way with graded color over the map or numerically as a list of tuples, we arrive at a better representation of the state than one predicted position would provide.

This method may also be used to determine the locations of objects external to the robot. As seen in Figure 20, the probabilities of object locations are depicted with shades of gray.



Occupancy grid

Figure 20: Occupancy Grid of objects in a room. Black circle at center represents location of robot perceiving said objects.

SECTION 6.3.1.2 FACTS

Expert systems and other knowledge-based reasoners use facts to represent the state of the system. Facts may be flat assertions about variables, relationships between them, or any other user defined classification. The list of facts asserted at a given time is the state of the system at that time. As an example, consider a farm as the environment. A flat assertion fact about this system might be

farm has-chickens X

where "X" is an integer number of chickens on the farm. To further describe the environment, we might assert a fact to establish a relationship between variables such as:

coop is-near stye

Subject Matter Experts (SMEs), in addition to flat assertion facts, can define facts to classify elements of the system. Such facts may be harder to attain via pure logic.

We can also use facts to classify elements of the system in ways defined by subject matter experts (SMEs) that might be harder to attain via pure logic. Say an additional fact is,

farm has-pigs Y

where "Y" is significantly smaller than "X." Through rules set by a SME, a deduction would be

farmer prefers chickens

Notions of preference, or other subjective measurements we wish to take, would be harder to establish logically in another state representation format, but in series of facts combined with SME input they are created easily.

SECTION 6.3.1.3 CONTINUOUS AND DISCRETE VARIABLES

One of the most common methods of state-space representation is using state space vectors (described in section 6.3). In these vectors, each of the variables chosen to describe the system – whether observed or calculated – is assigned an axis and plotted along it. These variables may be either continuous or discrete. Continuous variables may be plotted anywhere along their axis and be assigned any real value in that range. For example, the temperature of a room may be 67.6°F, 95.123°F, 25°F, or any other real, possible value. The only limit to precision is the tool used to measure the temperature, but as a continuous variable it may be assigned any real number and plotted at that point along a line.

Discrete variables may not be assigned any numeric value, but instead are limited to a discrete subset of values. Some variables may naturally be discrete. For example, the number of children in a classroom will always be an integer value, and therefore will be plotted as discrete. Continuous variables may also be turned into discrete variables by setting partitions of continuous data and assigning labels to those partitions. For the temperature example above, a room at 25°F may be labeled as “cold”, the 67.6°F room may be considered “warm,” and the 95.123°F room may be labeled as “hot.” For plotting, “cold” might be assigned the value 1, “warm” might be assigned the value 2, and “hot” might be assigned the value 3. Therefore, along the discrete temperature axis, you may be in one of three positions.

SECTION 6.3.1.4 BINARY VARIABLES

Binary variables are the most basic way of describing the state of a system. These variables provide two options for each question asked. For example, a system may be “on” or “off”, a coffee may be “hot” or “iced”, and a pool might be “filled” or “empty”. In each of these cases where either of two states is possible, one state may be represented by a 1 or TRUE, and the other by a 0 or FALSE.

SECTION 6.3.2 STATE SPACE (S)

The state space S describes the environment of the decision-making problem. A state space can consist of a broad range of variable types for representing the environment. Mostly commonly, an individual state within the state space is a continuous real number, or discrete (integer or binary) variable. For example, a state space can represent the velocity of a vehicle as a continuous number, or the velocity could be binned into a binary variable where 0 represents the vehicle being stopped, and 1 represents the vehicle moving forward.

As demonstrated by the previous example, the designer of the state space must make important decisions on the state representation. There are two primary considerations when selecting a state space: 1. Does the state space represent the decision-making problem with high enough fidelity to gain value from the solution to the problem? 2. Can a solution to the decision-making problem be reached the state space in the time required by the application? The answers to each question are highly application-dependent and often rely on expert knowledge for both the design of the state space, and the solution to the decision-making problem. If the state space is too large, consider breaking up the problem into smaller, easier to solve, sub-problems.

SECTION 6.3.2.1 STATE SPACE REPRESENTATIONS

SECTION 6.3.2.1.1 STATE SPACE VECTOR

One of the most simple and common ways to represent a state space is as a set of vectors. In this representation, a state variable is the smallest possible subset of system variables that can fully describe the state of the system. These variables are represented by increments along a Euclidean Graph with a dimension for each modeled state. The state is represented by a vector in this graph, defined by the locations of the set of state variables at the given time step. This vector can be represented by a matrix for easier manipulation and computation. Actions are represented by transfer functions, typically differential equations, or difference equations, using the state variables and some user defined inputs and outputs to describe transitions from an initial state to future states.

SECTION 6.3.2.1.2 KNOWLEDGE BASE

Knowledge bases are used to define the state for implementations of expert systems. Knowledge bases represent the state of the system through a set of facts and rules. Facts about the system may describe measured numeric values of state information such as sensor outputs. They may also describe relationships between components or discretized or otherwise abstracted information about the state of a physical or virtual system. Rules take facts or objects as inputs and through logic create either new facts or instances of objects. When new objects are created, they are used as inputs to other rules such that when an iteration of the program is complete a set of facts is the output. This set of facts can be used as the state space representation for case-based reasoning (CBR) or other expert systems.

SECTION 6.3.2.1.3 GRAPHS

Another common method of representing the state space of a system is with a graph. Graphs come in many forms, but always include a set of nodes and edges. Every modeled variable has a representative node containing its relevant information. The relationships between nodes, whether causal, correlational, or other, are represented by edges. Edges may take the form of arrows, lines, or curved lines depending on the type and implementation of the graph. A few common types of graphs are detailed below.

SECTION 6.3.2.1.3.1 PLANNING GRAPH

Planning graphs are used to describe propositional problems and contain a series of levels corresponding to time steps in the proposed plan where level 0 is the initial state. In each level, there is a set of literals and a set of actions which describe what might be possible at that time step. Persistence actions, or actions that maintain the state, must be included for completeness. Mutual exclusions also must be included to increase search efficiency across different potential paths. Mutual exclusions are represented by a curved line and can apply to both sets of literals as well as sets of actions.

SECTION 6.3.2.1.3.2 CAUSAL GRAPH

Causal graphs represent the decision-making state space using probabilistic graphical models of observed variables and their relationships with one another. In such a graph, each modeled variable is represented as a node. When a given variable affects another, all else held constant, an arrow is drawn from the affecting node to the node of the impacted variable. The set of variables that point to a given variable are called its "parents" or "direct causes." These parents are represented by the term Pa(Y) where Y represents the impacted variable. This form of model usually includes error terms or omitted factors, but these are typically excluded from the graph. If two variables have error terms that depend on one another, then the error terms are included and connected in the graph via a bidirected arc.

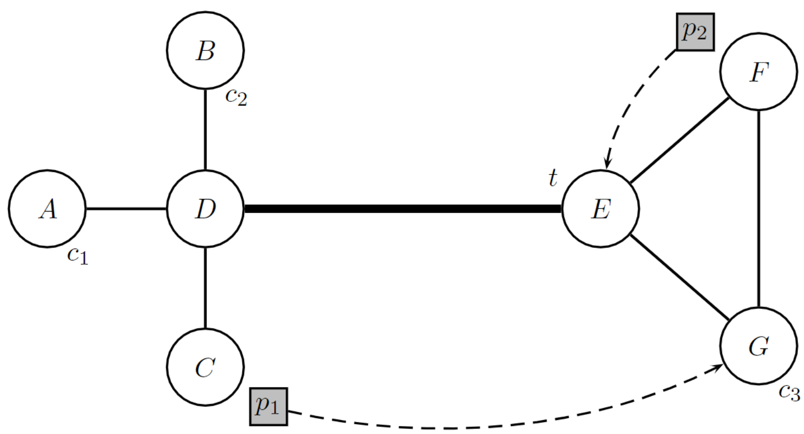


Figure 21 A Transportation planning task. Deliver parcel p1 from C to G and parcel p2 from F to E, using the cars c1, c2, c3 and truck t. The cars may only use inner-city roads (thin edges), and the truck may only use the highway (thick edge). [1]

SECTION 6.3.3 ACTION SPACE (A)

The action space A is a representation of the actions the decision-maker can take to affect the environment and state space in some way. Like the state space, the designer can represent the action space by both continuous real numbers and discrete choices. For example, an autonomous vehicle may have a discrete choice between taking route A, which is short but has difficult terrain, or route B, which is a longer, but with easier terrain. On the other hand, navigation along each route depends on continuous actions sent to the vehicle's motors.

SECTION 6.3.4 TRANSITION MODEL (T)

A transition model T describes how the state of the environment changes based on actions from the decision-maker. The transition model aids decision-making for a broad set of approaches by predicting the response of the environment to possible actions taken by the decision-maker. This allows the decision-maker to shape the state of the environment in predictable ways. The transition model for a system may be explicitly defined by an expert, learned from data collected on the environment, or produced using a combination of expert knowledge and data.

One common form of the transition model, used in Markov Decision Processes, is the discrete time case in a stochastic environment:

$$\Pr(s_t \mid s_{t-1}, a_{t-1})$$

Where the model computes the probability, Pr, of being in a state at time t, s_t, if the state at time $t-1$ is given as s_{t-1}, and the action is a_{t-1}. This form uses the Markovian assumption that the state of the next time step depends only on the previous time steps (and not time steps preceding that one).

Examples of transition models commonly used in practice are:

- Robot and vehicle motion models: describe how the position, heading, velocity, etc. of the vehicle change from time step to time step based on actions, such as steering wheel angle.
- Dynamical systems models: general description of the future state of a system based on initial conditions.
- Models from a simulator: if a simulator of the decision-making environment is available, the transition model can be learned through data from the simulator. For example, simulations from the video game Pong may be used to learn a transition model for better decision-making in the game.

SECTION 6.3.5 OBSERVATIONS (O)

In many cases, the state of the environment is not fully observable, but instead partially observable. An autonomous vehicle may have noisy sensor measurements on the state of the environment, but a complete description of the environment state. For example, the location of an adversary may be occluded by an object. Observations are commonly described by continuous or discrete set of measurements on the environment, such as by an accelerometer, GPS, imagery, or LiDAR.

Many decision-making approaches use an observation model to make sense of the observations collected on an environment. Partially Observable Markov Decision Processes (POMDPs) and Kalman filters require observation models which form a mapping from the state of the environment to the expected observation, given that state. Most commonly, this is a static (not time-dependent) mapping expressed by:

$$\Pr(o_t \mid s_t)$$

Which expresses the probability of an observation o_t given the state s_t at time t. Partial observability of the environment makes the decision-making problem much more challenging but is necessary to consider to many real-world applications.

SECTION 6.3.6 CONSTRAINTS (C)

Constraints are bounds placed on the state space (and sometimes the action space) that the decision-maker must avoid violating. For example, the designer of an autonomous drone may include a constraint in the planner that requires the drone to return to base for recharging before the drone's battery depletes. Different approaches to decision-making incorporate constraints in a variety of different methods. Forward simulation of the environment, perhaps through a transition model, can aid in predicting whether a constraint will be violated.

SECTION 6.3.7 VALUE JUDGMENTS (R)

We use R to describe value judgments that shape the decision-making problem. Different fields of research use different terms to express these value judgments, including: cost, loss, reward, risk, utility, regret, etc. Many approaches center around making decisions that maximize or minimize these value judgments. The decision-making problem may seek to maximize reward or minimize cost. While the exact definition and use of each term varies from approach to approach, most implementations of these value judgments require a level of *design*. In some cases, the design is simple. A simple design may be minimizing the cost, in dollars, of manufacturing a product. More complicated value judgments may have multiple objectives, such a cost that penalizes deviation from a mission plan at the same time as fuel. The field of multi-objective optimization explores these tradeoffs in objectives.

SECTION 6.4 DECISION-MAKING METHODS

The algorithms and classes of algorithms used to make decisions in each representation are described in this section. There are often multiple methods that apply to the same set of decisions; however, it is up to the designer to identify which tool is best for their application. In this section each of the algorithm classes listed below will first be defined, then guidance for when the algorithm classes are best applied will be given. The algorithm classes will also be formulated to align with the definitions described in section 6.3 State-space representations and specifications, and when available, off-the-shelf implementations of the algorithms will be provided. The algorithm classes described in this section are:

- Optimization-based approaches
- Search-based planning
- Reinforcement learning
- Expert systems
- Belief-space planning
- Case-based reasoning
- Goal-driven autonomy
- Game theory

SECTION 6.4.1 OPTIMIZATION-BASED APPROACHES

Description: Optimization-based approaches to decision-making revolve around the definition of an objective function and subsequent selection of an action that minimizes (or maximizes) that objective function. The objective function defines what is important to the decision-maker, such as maximizing the return on an investment, or minimizing the fuel consumption of an autonomous vehicle. Often, optimization problems augment the objective function constraints that limit the feasible solution space. For example, the autonomous vehicle must minimize fuel consumption, subject to the constraint that it reaches a specific location.

While mathematical optimization is powerful due to its generality, the challenge in real-world applications is to define the optimization problem such that a numerical solver finds a good, if not optimal, solution in a reasonable amount of time. This is particularly difficult for high-dimensional problems and/or problems with poorly structured objective functions (e.g., non-convex functions). However, optimization problems that are structured appropriately for off-the-shelf solvers can achieve fast, reliable solutions.

A general decision-making optimization problem may be structured as:

$$\begin{aligned}&minimize\ J(a)\\&subject\ to\ g(a) \leq 0\end{aligned}$$

In other words, find the action a that minimizes the objective function J, subject to some general constraint g(a)<=0. Solvers exist to handle general optimization problems, such as MATLAB's *fmincon* or SciPy's *optimize*. However, we recommend attempting to formulate decision-making problems, especially problems with many variables, with structures amenable to fast solvers. For problems with continuous decision variables, examples of this include linear programming (MATLAB or SciPy's *linprog*), quadratic programming (MATLAB's *quadprog* or Python CVXOPT), or convex programming (CVXOPT).

Many problems contain discrete/integer/binary decision variables. Further specialized solvers exist for these types of problems (see MATLAB's Mixed Integer Linear Programming or GUROBI Optimization). Evolutionary algorithms, such as genetic algorithms, may also work very high dimensional cases, but can be unreliable. In general, the designer must structure the optimization problem carefully and choose an appropriate solver.

The general formulation may be extended to cover sequential decision-making to find sequences of actions. For example:

$$\begin{aligned}&minimize\ J(a_0, \dots, a_{T-1}, s_0, \dots, s_T)\\&subject\ to\ g(a_0, \dots, a_{T-1}, s_0, \dots, s_T) \leq 0\\&and \qquad\quad s_t = f(s_{t-1}, a_{t-1})\end{aligned}$$

where the transition model, f, between states becomes part of the optimization problem. Optimization as an overall approach has been widely developed in sub-fields, including examples in Model Predictive Control (MPC), trajectory optimization as well as a wide range of robot planning problems.

Optimization problems can similarly be formulated to handle uncertainty. Uncertainty in the environment implies that a distribution of possible outcomes exists when making a decision. The designer of the optimization problem may, for example, may wish to minimize the objective function J of a decision *on average*. Some most common forms of this problem include:

- Maximizing Expected Utility: from decision theory, select the decision that maximizes the "expected utility" of a decision with a probabilistic outcome, see: https://www.cs.cornell.edu/courses/cs5846/2010fa/cs576wk1.pdf
- Empirical Risk Minimization: minimize the average loss of a model, such as a neural network, in predicting an outcome based on data. Commonly used in supervised machine learning problems and often solved with stochastic gradient decent. For general review of related statistical learning problems, see https://web.stanford.edu/~hastie/Papers/ESLII.pdf

- Chance Constrained/Stochastic Optimization: optimize the expected value of the objective function, subject to a "chance constraint" on the probability of an event occurring, see https://stanford.edu/class/ee364a/lectures/chance_constr.pdf

Another approach to stochastic optimization is to apply structural reliability theory, where the state variables are assumed to have known stochastic properties, but their scalar values are unknown. Kim and Frangopol examined such a framework for making decisions when to monitor ship structures for fatigue (e.g., the actions relate to requesting more sensor data to characterize the ship's condition) [2]. They were able to generate Pareto fronts of optimal service lives, delay in detecting damage and reliability in service, the chose rewards for this problem. However, as a multi-objective formulation, selecting a single the final decision relies on another decision-making step.

When to Use: A broad array of decision-making problems can take advantage of optimization-based approaches. However, it is best applied in small, well-scoped problems with clear objectives and constraints so that reliable numerical solvers can be used.

Problem formulation $(\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{T}, \boldsymbol{C}, \boldsymbol{R})$

- S – The state of the system, or if the decision-making problem is sequential, the state of the system as it varies through time
- A – The actions (or decisions) need to be encoded as the variables in the system of equations, so that the optimizer can find the appropriate values for them
- T – The transition function for system dynamics, how the actions affect the system, are inherently built into the system of equations. For example, a moving system will calculate a change in position using the last position, the velocity, and the time step
- C – The set of constraints
- R – The reward value that you are trying to minimize or maximize, calculated from the value of the states and the costs of the actions

Pros: Extremely broad applicability.

Cons: The optimization problem must be carefully defined, as solutions can easily become computationally infeasible. See Kim and Frangopol for an example of a discussion of computational cost, especially when the reward is multi-objective [2].

Tools: MATLAB (fmincon, quadprog, linprog, Mixed integer linear programming); Python (linprog, CVXOPT, SciPy optimize)

SECTION 6.4.2 SEARCH-BASED PLANNING

Often a decision cannot be well-formulated into an optimization problem: the value function may be non-convex, too many degrees of freedom, or some other complicating factor that will cause an optimizer to take too long to find a solution or fail to converge outright. In some cases, an easy-to-find *acceptable* solution can become preferable to a hard-to-find *optimal* one. Most search-based methods are guaranteed to find solutions if they exist, with some computational cost, if the transitions and the utility of the individual decisions are defined.

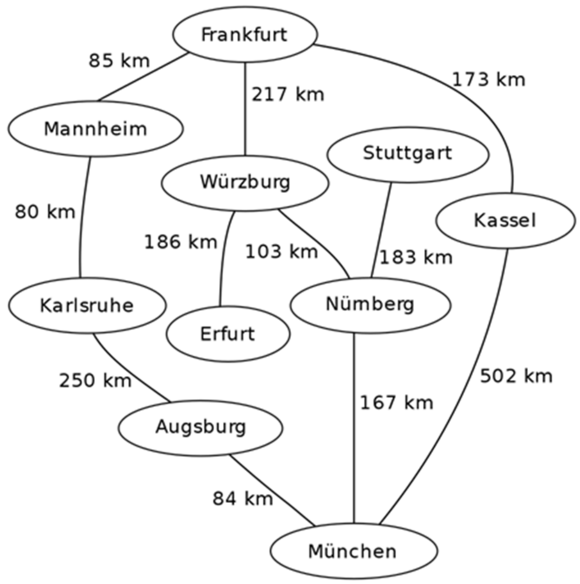


Figure 22 A graph of connections between cities in Germany https://en.wikipedia.org/wiki/Breadth-first_search

Search-based planners seek to find a sequence of actions that will take the system from a known starting state to a known goal state. Typically, a search algorithm will begin at the starting state, and recursively evaluate the next states that are accessible from the current state by applying each action until the goal state is found in the evaluation. Two common examples of this method are breadth-first and depth-first search.

Consider the problem of finding a path from the city of Frankfurt to Stuttgart. Figure 22 shows the distances and travel links that connect several German cities. A breadth-first approach will first identify all the cities that can directly be reached from Frankfurt: Mannheim, Wurzburg, and Kassel, then all the cities that can be reached from those cities and so on. Figure 23 (a) shows the state of the search tree when the goal location, Stuttgart, is found and the path is returned. A depth-first search approach also begins with the known starting state, but rather than fully expand each node in the order in which they were found, a depth-first search expands nodes as they are found. In the previous example, this search method would first find Mannheim, then find Karlsruhe, and so on. Figure 23 (b) shows the state of the search tree for a depth-first search when the goal location is found. Note that the paths found by the two methods differ in length and cost. If given a different goal, say Augsburg, depth-first search would have found a path with fewer node expansions than breadth-first. One major drawback to depth-first searches is that they are not guaranteed to find a solution and may get stuck searching indefinitely if the planning horizon is infinite.

Most research in search-based planning focuses on identifying heuristics to guide a search towards the goal faster, without compromising a completeness guarantee. Heuristics provide a guess at the true cost of an action, and if the guess strictly underestimates the true cost, then the sequences of actions can be guaranteed to be optimal within the chosen representation used for the search. The most used heuristic-based search algorithm is A*, which performs a breadth-first style search but orders the nodes to expand by the cost and a heuristic [3].

When to use: Searched based methods require that the problem can be modeled like a graph, where the states, transitions, costs, and goals are known (or knowable). Given an arbitrary state the system needs to know what states each of its action will take it to, and at what cost. Search is also best used when there is a known heuristic to guide the search, for example in a path planning algorithm the Euclidean distance between a candidate node and the goal node will always underestimate or exactly guess the cost to the goal.

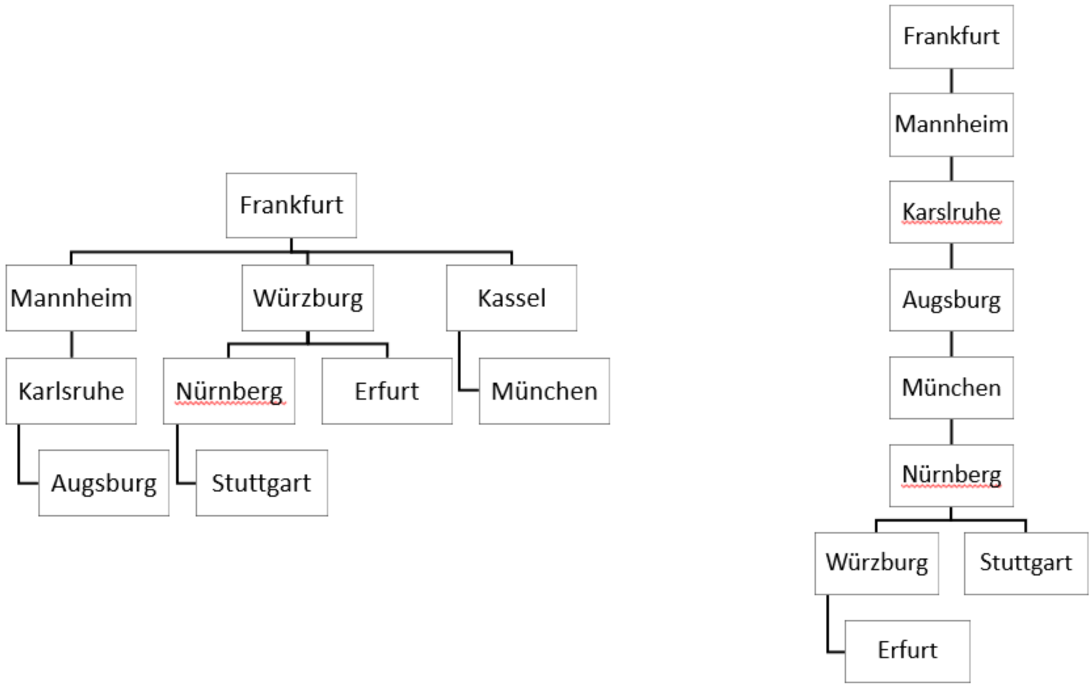


Figure 23 (A) Left: Search tree of a breadth-first search when seeking a path from Frankfurt to Stuttgart. (B) Right: Search tree of a depth-first search seeking a path from Frankfurt to Stuttgart.

S – The state (or states) of the system. If the transition model describes how the state is changed by an action, then only the current state is needed as all future states can be calculated from that one. If the transition model only describes the probability of moving to another pre-defined state, then the entire state-space must be constructed first. For example, the map of cities in Germany is needed before it can be searched.

- A – The set of actions that the system can take.
- T – The transition model, either edges on a graph or descriptions of how the state will change with a given action and set of conditions
- R – The value judgement, this is not necessary for finding a goal, however if there is a cost to minimize that cost must be encoded.

Pros: Search is a well-developed method with significant progress in the development of general heuristics [4, 5, 6, 1]. Able to determine sequences of actions to achieve goals.

Cons: Requires that the effects of all actions are predictable, and finite. Typically, does not scale well in very complex domains without well-defined heuristics, and heuristics may need to be domain specific.

SECTION 6.4.3 REINFORCEMENT LEARNING

Reinforcement learning (RL) is a policy estimation method useful for when the immediate reward or cost of an action is not well known. Any reinforcement learning problem begins with a formulation called a Markov Decision Process (MDP), which is a problem formulation that models the states, actions, transitions, and value judgements of state/action pairs. MDPs carry the assumption that the domain is Markovian, meaning that the transition to any given state is only dependent on the previous state and the action taken from that state.

While an RL agent seeks to model the entire set of reachable states it does not need them a priori, rather the agent will explore the space and after each learning episode will update its model with new states and updated transitions probabilities and rewards. After enough time exploring the space, the agent will have learned a policy from anywhere in the reachable state space towards the state of highest utility. This type of reinforcement learning is exemplified in Q-Learning [7].

If the state space, transition function, and value function is known ahead of time then value iteration can be used to develop a policy like that of what Q-learning would find. This policy would again try to maximize the expected utility of the actions or decisions taken [8]. A similar method to value iteration is policy iteration, which instead instantiates an arbitrary policy and updates the policy until the value of that policy converges [9].

When to Use: This technique is useful for situations where the transition models or the reward function is not well defined, but the system has a simulation or some other tool for exploring the state space repeatedly to develop a policy.

Problem formulation $(\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{T}, \boldsymbol{R}, \boldsymbol{\gamma})$

- S – The complete set of states that are reachable from the starting state, either precomputed or found through exploration
- A – The set of actions for the system to take
- T – The transition function between states
- R – The value judgement of each state, for training a reinforcement learning agent the reward function can simply be based on achieving a goal and the value will be propagated to the intermediate states.
- γ – A discount factor, this is a value between 0 and 1 used to weigh the importance of immediate reward versus future reward. A high discount factor will take immediate actions that may be less desirable with the expectation that a more desirable action will be available in a future state as a result of this action.

Pros: Once a policy is developed, there is no further computational costs for running that policy. This is particularly useful when rapid decisions are needed, as there may not be time or computational resources for running a planner or an optimizer.

Cons: Once the policy is developed, it is only valid for parts of the state space that the agent has already seen. Changes to the policy will require additional data and training time.

Use Cases: When the state-space is well constrained and well understood, then a reinforcement learning approach is appropriate.

Tools: Value iteration, Q-Learning, Policy iteration

SECTION 6.4.4 EXPERT SYSTEMS

Expert systems are one of the earliest attempts at capturing knowledge from a subject-matter expert and encode it into a reasoning system capable of emulating the decision-making process of that expert. This knowledge is captured as a set of rules for inferring information from the information that is already available. An expert system uses a reasoning engine for processing these rules on the current knowledge base.

There are two main types of reasoning engines used with expert systems: forward-chaining and backward-chaining. Forward-chaining reasoning engines look at the current set of knowledge about the system and infers everything it can about the system given its set of rules. A backward-chaining system looks at *possible* facts and applies rules backwards to find if that fact could be true. Forward-chaining is useful for when you want to know everything you can about the system, and backward-chaining is useful when you need to query the system for specific facts.

The rules for inferring new facts about the state of the system can also be treated as decision or actions. If the preconditions for a rule are found to be true and the rule results in a desired outcome, then the decision maker will decide to take the action (apply the rule). An example of this process is shown in Figure 24.

One downside to the use of an expert system is the intensive process of gathering information from subject-matter experts and encoding that information into rules. Recent work with expert systems seeks to learn new rules by themselves by using large amounts of data and machine learning techniques, in a paradigm called intelligent systems.

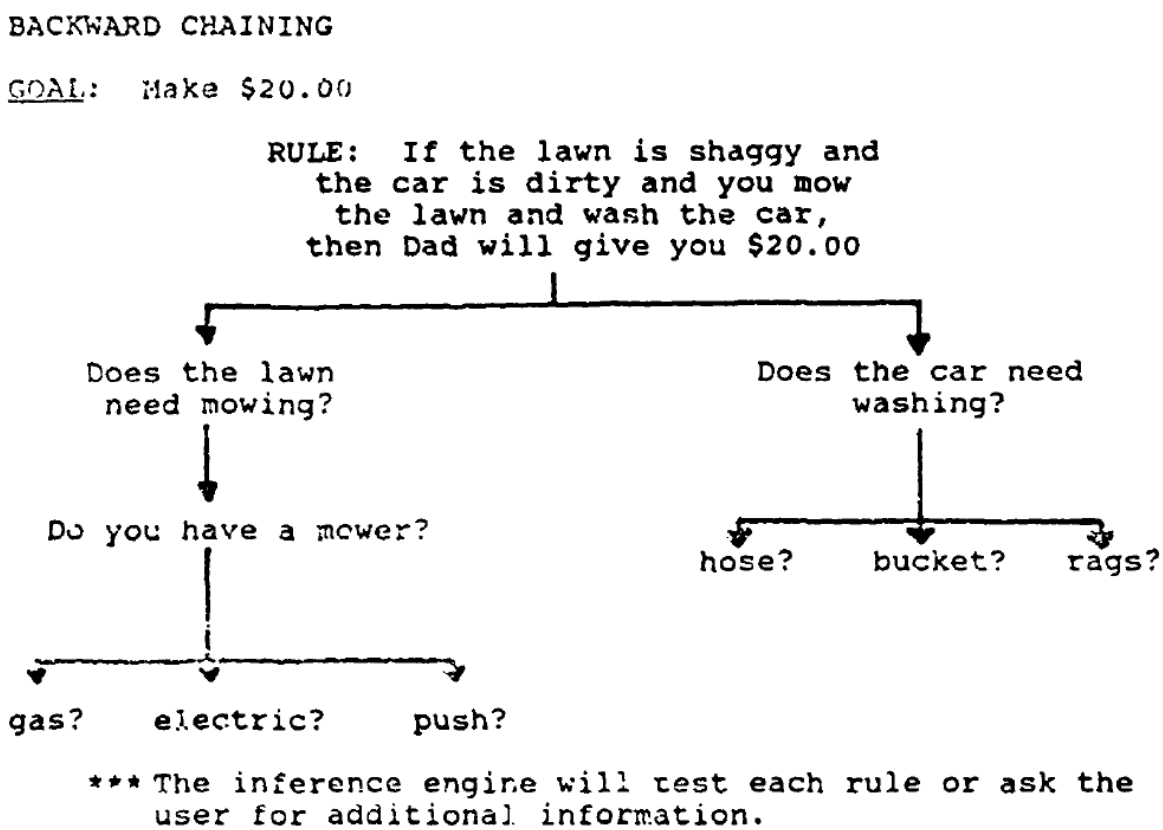


Figure 24 Backward chaining expert system [10]

When to use: Expert systems are useful for domains in which an expert's knowledge is often critical for making decisions. Previous domains include medical diagnosis, identifying organic compounds. In a maintenance problem, an expert system can be used to identify faults and corrective procedures.

Problem formulation (S, K)

- S – What is the current state of the system
- K – The set of rules that make up the expert system

Pros: An expert system encodes information that is well established by experts, rather than needing to learn that information by itself over time.

Cons: Encoding expert knowledge is an expensive process, requiring many hours of the expert's time and that of the developer to encode the knowledge into rules. Additionally, as the domain changes and new relationships form, the expert system needs to be updated continuously to keep up.

Tools: CLIPS reasoning system, PROLOG

SECTION 6.4.5 BELIEF SPACE PLANNING

Rarely are autonomous systems able to observe every aspect of their environment. In most cases their sensor measurements are a proxy for the information they are trying to acquire, like measuring the number of turns of a wheel and the time elapsed to estimate speed. In more complex environments suitable proxies may not exist, and so a belief state on the value of interest can be used to estimate what that value might be based on other observations and the actions taken by the agent. This type of problem is often formulated as a Partially Observable Markov Decision Process (POMDP).

In their work, Platt *et al.* provide a method for continuous state space planning using linear-quadratic regulator and seek to minimize the covariance in the belief that the system will end up in a desired state [11]. This approach is like a model-predictive controller, but rather than minimize the error between the controlled trajectory and the reference trajectory they minimize the uncertainty that the system will be in some desired region of the state space. They found their approach to be locally optimal in regions where the state dynamics are deterministic, and the observation dynamics are linear.

Additional work uses a hierarchical approach to planning and separate the tasks of planning and estimating the current state of the world. Kaelbling demonstrates this approach with hierarchical planning in the now (HPN) which takes a determinized approximation of the problem, and then develops plans through that determinization [12]. In HPN they include the probability of each transition in the cost of that action, in order to develop plans that can trade off between optimizing a cost and maximizing the likelihood of success. Once they have a plan and execute part of that plan, they can use observations from the world to update the belief they made the intended transition or not.

Like belief-space planning is risk-constrained planning, which instead of developing plans that minimize the chance the system fails to reach a goal minimizes the chance the system takes an undesirable trajectory or enters a catastrophic state [13, 14].

When to use: Belief space planning is best for problems with a high degree of uncertainty in the state and the results of an agent's actions. This type of planning works for both discrete and continuous problems if the dynamics in the system can be reasonably linearized.

Problem formulation ($\boldsymbol{B}, \boldsymbol{A}, \boldsymbol{T}, \boldsymbol{G}$)

The problem formulation in belief space planning is like an MDP, with a few differences.

- B – The *belief* state-space. Instead, the representation of the current state, this models the probability of the system being in any one state
- A – The set of actions that the system can take
- T – The transition function that defines how the agent moves from state to state **as well as the uncertainty that is introduced during the transition**
- G – A set of goal states that the system is trying to achieve.

Different implementations of belief space planners will extend on this formulation, although all need at least these 4 components.

Pros: Belief space planning provides the system a method to develop plans with some guarantees on the chance that the system will successfully reach the goal. In cases where it is more important to achieve a goal and avoid a catastrophic failure than optimize a value function, this is very desirable.

Cons: Belief space planning is rather complex and requires maintaining an estimate of the current state of the system, which involves reasoning over many possible states all at once. The requirement that the underlying system dynamics be fairly linearizable (at least with existing techniques) limits the systems to which belief space planning is applicable.

Tools: BHPN

SECTION 6.4.6 CASE BASED REASONING

Much like how legal cases are argued and decided, case-based reasoning (CBR) makes decisions by drawing from previously encountered examples. This technique is particularly useful when the state space cannot be reasonably enumerated or sufficiently interrogated to derive a policy or plan. The method does not require an explicit domain representation, rather it only needs to identify the significant features of a case. Perhaps most importantly, CBR can learn over the lifetime of the system, adding each unique case to its memory and remembering any adapted solution. CBR consists of a 4-step process [15]:

1. Retrieve the set of similar cases to the current case
2. Reuse information about the prior cases and their solutions
3. Revise prior solutions to fit the current case
4. Retain the new case and the revised solution

The full cycle is shown in Figure 25.

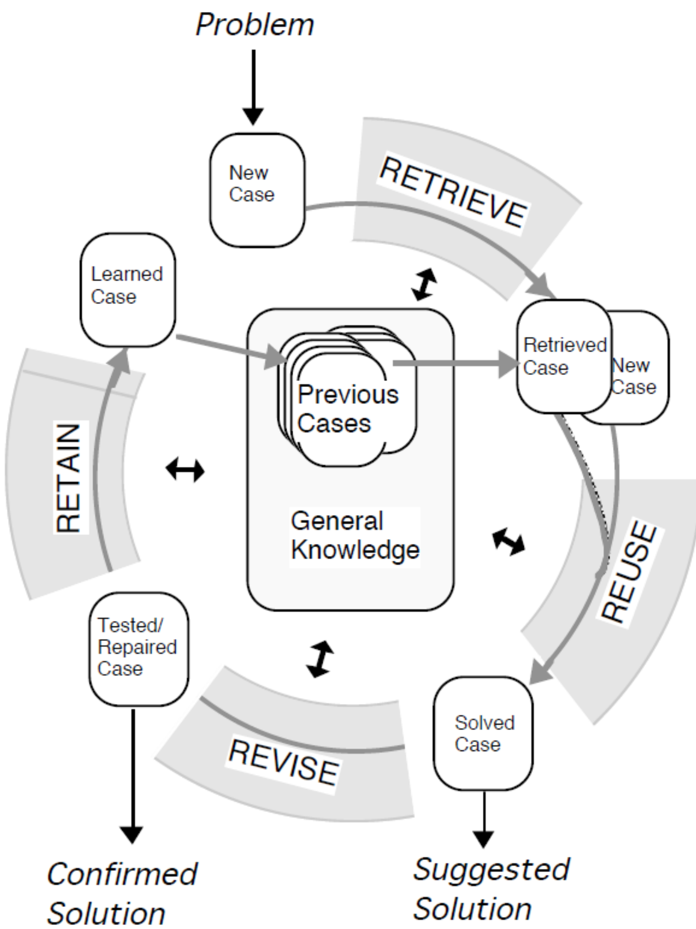


Figure 25: The CBR process

- A vocabulary: the knowledge necessary for choosing the features used to describe the cases.
- Similarity measures: The ability to compare two cases and measure how similar they are
- Adaptation knowledge: the knowledge necessary to adapt existing solutions to new cases, as well as the knowledge to evaluate the proposed solutions

When to use: Case base reasoning is helpful for when the exact outcomes of the actions are not known, and the domain is too complex to reason over like in reinforcement learning or planning. Case base reasoning is also good for leveraging the knowledge of a subject matter expert who can populate the initial case base.

Problem formulation $(\boldsymbol{V}, \boldsymbol{K})$

The formulation of the case-based reasoning system is largely going to depend on how the designer implements it. As described before, the system will at a minimum need

- V – the vocabulary to describe cases
- K – Knowledge about how to detect similar cases and adapt solutions

Pros: Does not require offline training and is robust to new parts of the state space. Borrows knowledge from previous examples to speed up the learning process

Cons: Requires an expressive ontology of the system, which takes significant development time with a subject matter expert.

SECTION 6.4.7 GOAL-DRIVEN AUTONOMY

Unexpected scenarios become inevitable when dealing with long-term deployments of any system. A component might break, a plan might fail, or some previously held assumption may be found to be false. When a system finds that it's previous goals (if it had any) are no longer achievable, it needs some method to generate new goals that still support mission objectives or partially achieve the now defunct goals.

A goal-driven autonomy (GDA) process consists of 4 steps:

1. Monitor – Check for discrepancies between the expected state and the observed state. Did something change (or not change) that wasn't expected? Is there a new goal from a user that takes priority?
2. Diagnosis – what caused the discrepancy? Is there some logic to explain what has happened?
3. Generate goals – Propose some concrete goals that will let the system recover from the change
4. Select a goal – Evaluate the proposed goals and select the best one

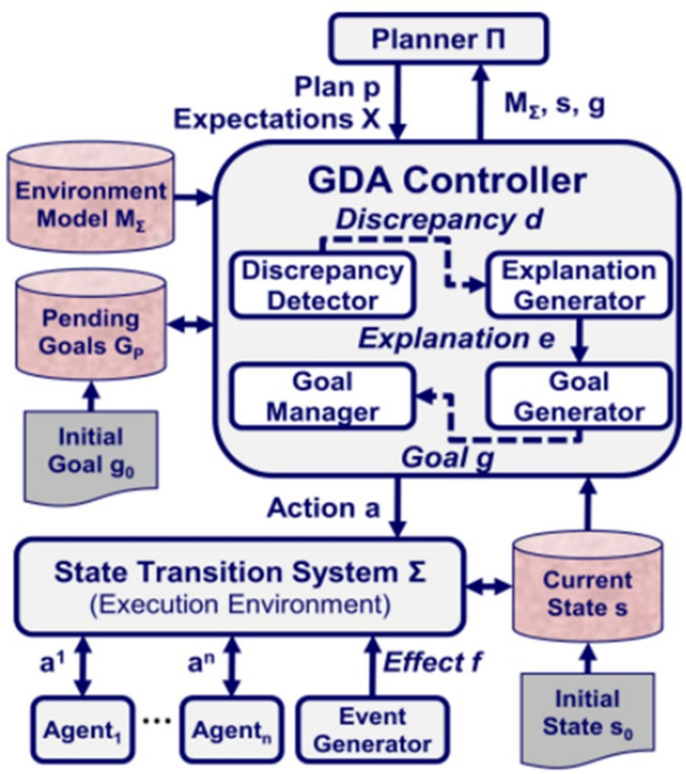


Figure 26 The ARTUE Goal-Driven Autonomy system

The first full implementation of GDA was in the Autonomous Response to Unexpected Events (ARTUE) system [16]. That system is shown in Figure 26:

- Environment is a tuple (S, A, E, T)
- The planner takes in the environment, the current goal (g), and the current state and returns a series of actions (A) that satisfy the goal, as well as a sequence of expectations (X)
- The GDA controller executes the GDA process, which first monitors for differences between the observed state and the expectations, finds explanations for the differences with its knowledge (K), produces a set of goals (G) and selects one of those goals to pursue. It then sends that goal to the planner to get a series of actions to execute.

Not all GDA processes will require a planner, however, and so at minimum the GDA process only needs a state description, a set of expectations, a goal description, and knowledge to diagnose discrepancies.

When to use: A goal-driven autonomy process enables a system to be self-directed when either its goals are unachievable or perhaps not well defined. Long-duration missions and missions where graceful degradation is preferred over outright failure are prime candidates for a GDA process

Problem formulation $(\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{X}, \boldsymbol{K}, \boldsymbol{G})$

- S – The set of all states possible in the environment
- A – The set of actions or decisions for the agent to make
- X – The set of current expectations on the system
- K – Underlying knowledge of the environment that enables reasoning about why discrepancies occur
- G – The set of goals for the agent to consider

Pros: A goal driven autonomy process lets the system "gracefully degrade" so that when an initial set of goals becomes impossible or less desirable to achieve the system can adjust to new goals.

Cons: Developing expectations often requires that the plan or policy is developed in a way that each state along the plan can be precomputed, so the complexity of environment that a GDA system can operate in is limited by the complexities of the plans that can be developed. Further, the knowledge to reason about discrepancies between observed states and expected states is difficult to encode.

Tools: ARTUE

SECTION 6.4.8 GAME THEORY

Game theory is the study of mathematical models of strategic interaction among rational decision-making agents. Its concepts can be applied in many fields including computer science, systems science, and social science. The modern definition of game theory for computer science includes the study of interactions between self-interested agents [17]. Here we will examine how game theory could be applied to decision-making in an autonomous framework. Unlike the other techniques reviewed so far, there have been far fewer demonstrated applications of game theory for this problem. Thus, this section will provide a high-level overview of the approach, but without the details and assessments that were provided for other topics. Game theory remains an excited avenue for both further research and wider test applications. Within the decision-making domain, game theory can be applied to multi-agent systems in the principles of task sharing/allocation [18], coalition formation [19], negotiation/bargaining, etc. Agents want to make choices that optimize the outcome, but in order to do so it will need to use strategic reasoning to determine what other agents in the scenario are likely to do (where each agent is optimizing for its own outcomes).

"The classic game theoretic question asked of any particular multi-agent encounter is: What is the best — most rational — thing an agent can do?"

When to use: Useful when the agent is uncertain what the result of taking an action will be. Specifically, it is useful in multi-agent scenarios where the environment is unpredictable. Such multi-agent scenarios may include interactions like negotiation and coordination. Specific game types and representations need to be developed based on the characteristics of each application. General guidance for this, in the context of system decision-making, is still a topic of research.

SECTION 6.4.9 SUMMARY

The decision approaches discussed in this section are summarized in Table 5. The types of state-space representation used by each method, following the definitions from section 6.2, along with the strengths and weakness of each method are summarized. The table shows clearly that the current variety of decision-making frameworks is an important strength when considering coupling with a digital twin. Frameworks can address a variety of problem descriptions, from purely numeric optimization and search-based approaches to more language-driven expert systems approaches and case-based reasoning approaches. If a digital twin is to be part of a decision framework, it is important during the design of the twin to also specify and explore the decision approach that will be coupled to the twin. Such co-design can ensure the information available plays to the strengths of the decision approach, and that the decision problem remains tractable, avoiding the commonly-encountered problem where monitoring data is gathered and prognosis calculations take place, but the resulting information is not used in decision-making.

Table 5 Decision-making Approaches

Decision-making Approach	Optimization-based	Search-based	Reinforcement Learning	Expert Systems	Belief Space Planning	Case Based Reasoning	Goal-Driven Autonomy
State (S)							
Action (A)							
Transition (T)							
Constraints (C)							
Reward (R)							
Discount Factor (γ)							
Knowledge (K)							
Belief (B)							
Goals (G)							
Vocabulary (V)							
Expectations (X)							
Pros	Broad Applicability	General Heuristic, Able to determine sequences of actions to achieve goals	Computationally efficient once policy is developed.	Captures pre-determined information, rather than needed to learn itself	Useful for avoidance of poor outcomes	Does not require offline training, robust to novelty.	Dynamically adjusts to new goals
Cons	Optimization problem must be carefully defined	Requires effects of actions to be predictable and finite. Not easily scalable	Policy only valid for previously observed state spaces.	Encoding expert knowledge is expensive, and may need to be updated over time.	Complex, computationally expensive, requires system dynamics to be somewhat linearizable.	Requires expressive ontology of the system (see "Expert Systems")	Limited applicability to complex plans. Difficult to encode reasoning between observed and expected states

SECTION 6.5 EXAMPLES OF DECISION-MAKING IN NAVAL APPLICATIONS

Decision-making systems for autonomous vessels are not extensively published to date. Most of the marine industry's work in decision support has applied primarily to multi-attribute decision-making during the design phase of the vessel, with a variety of optimization-related methods explored as well as smaller forays into the methods covered in this report. Operational decision-making studies are slightly more common, with a recent review paper covering a range of techniques applied to oil-spill response[20], as well as a similar review paper covering weather routing frameworks [21]. Navigation challenges for autonomous vessels have also received extensive discussion in the literature. Component or single sub-system decision-making systems have been more widely studied, such as the structural system review by Kim and Frangopol (2018). However, the state-of-the-art does not extend to a platform-level or fleet-level decision support system as of the present time.

To illustrate the potential for these decision-making techniques to grow into the area of autonomous platform support, two hypothetical example scenarios will be described. First an example of how decision-making can affect a fleet of ships at a command level. The decision-making algorithms will be used to assign tasks to multiple ships to maximize mission effectiveness. The second example will demonstrate the impact of decision-making on a platform level problem. This example will illustrate the importance of decision-making on a ship where multiple systems must interact to achieve a specified goal. These examples will demonstrate how to characterize the decision, define a problem as a state space, and select a decision-making approach to address the challenges in multiple use cases.

SECTION 6.5.1 FLEET LEVEL EXAMPLE

SECTION 6.5.1.1 DESCRIPTION OF THE SYSTEM OR PROBLEM

Describe the system or the problem for which decision-making is required. Specifically identify the decision(s) that needs to be made.

For this example, there is a fleet of ships. A digital twin representation of a fleet considers multiple platforms. Each platform within a fleet twin can operate independently but is linked with other platforms spatially and temporally. Fleets are characterized by the coordinated efforts of platforms to attain higher-level objectives. Relationships between platforms may be organized and explicit, such as entities within a supply chain, or random and disjointed, such as cars on a city street [22]. Platforms within a fleet are often diverse and are developed for specific functions and are developed (perhaps not explicitly) for the purpose of exhibiting emergence. The fleet digital twin consists of the positions of the agents (ships) and high-level logistical information (e.g., range they can travel, speed they can travel, available resources, etc.). The decision to be made concerns resource allocation (i.e., ships/agents) to tasks in order to maximize the number of goals/tasks that are completed. Tasks may have different priorities, timings, or requirements to be completed (i.e., only certain types of ships can complete some tasks).

This is very similar to the Orienteering Problem, which can also have variations to include time-windows and service times.

SECTION 6.5.1.2 DEFINE THE DOMAIN USING QUESTIONS FROM SECTION 6.2

Provide answers to each of the questions from section 6.2 for this domain.

The environment is **dynamic**. There are changes in the outside world that are outside of agent's control. This may be caused by other vessels or changes in goals due to outside influence. This problem involves **sequential** decision-making. The task one ship/agent is given will affect how tasks are distrusted among the rest. Decisions may be revisited as the environment or goals change. This is a **non-deterministic** environment. A ship may be allocated to do a specific task but does not determine that it will be able to complete its mission. This is a **continuous** domain. The agent can choose from infinite options for solutions including not allocating a resource/ship to a goal at all or allocating multiple resources/ships to one goal if needed. A ship/resource may also be allocated for a specific amount of time before a different ship becomes available which may be better equipped to handle that task. The environment is **partially-observable**. The agent cannot always know the state of all other ships (outside of the controllable fleet) in the world. Other factors such as weather patterns are also likely to be partially observable.

SECTION 6.5.1.3 CHOOSE THE REPRESENTATION(S) APPROPRIATE FOR THE DOMAIN

Identify which representation, or representations, from section 6.3 that is appropriate for describing the domain and the reasoning about for making decisions.

This problem could be represented by an occupancy grid with some additional variables to hold high-level logistical information. This representation allows for the representation of the spatial environment.

SECTION 6.5.1.4 SELECT AND GO THROUGH A DECISION-MAKING PROCESS FOR THE DOMAIN

Identify which decision-making process is most appropriate given the representation selected in the last step, and step through some example decisions that would be made using that method. If one of the complex decision-making processes is used, go through that process: what decisions are made along the way and with what tools.

This problem could be solved using optimization-based approaches. This would involve defining an objective function whose goal is to maximize the number of tasks or maximize the overall reward (if each task has an assigned reward value, priority may equal reward in this case).

An example decision would be assigning a resource to Task A. Task A can only be completed by one of the vessels in the largest class in the fleet, must be completed 6 days from now, takes 2 days to complete, must be complete at a specific location, and has a high priority. In this case, the reward function would need to consider what resources are currently available in the this class of vessel, what resources will be available for the time window that is required, and how long it takes various possible ships to navigate to the location. It may also decide that a ship that is already allocated to another lower priority task (B) should complete Task A instead because it is more important (has a higher priority).

101

SECTION 6.5.2 PLATFORM LEVEL EXAMPLE

SECTION 6.5.2.1 DESCRIPTION OF THE SYSTEM OR PROBLEM

Describe the system or the problem for which decision-making is required. Specifically identify the decision(s) that needs to be made.

This example contains a single vessel. This vessel will include a set of diverse systems (i.e., power/energy, communications, armament, propulsion, etc.). The digital twin represents how each system interacts with and is allocated power. The problem is to determine how power should be allocated to achieve prioritized goals. Note that the goals and decisions made at a platform level are fundamentally different from those that exist at component or system levels, due to the presence of coordinating diverse system goals. These platform level goals are broader and more abstract than for a specific component or system. Due to their level of abstraction, the number and diversity of possible goals are much larger, and do not directly translate to a finite set of system or component goal states. As such decision-making strategies will vary across the different levels of the digital twin. A simple optimization solution will likely work well for a component or structural element, but for a more complex system with multiple goals would likely perform poorly.

Consider as well that the agent has a network of sensors to provide feedback from the systems to determine how power in consumed. As it would be impractical (impossible) to employ an array of sensors that could provide feedback on all elements of the platform, the agent must also include consideration of uncertainty in decision-making.

SECTION 6.5.2.2 DEFINE THE DOMAIN USING QUESTIONS FROM SECTION 6.2

Provide answers to each of the questions from section 6.2 for this domain.

This environment is **dynamic.** The crew of the vessel can cause changes to the system. The agent does not have control of all aspects. This will involve **sequential** decision-making. How power is allocated in one moment affects how the same decision may be handled in the next.

This is a **continuous** domain. The agent will have a constant influx of sensor feedback from the systems. The agent can choose from an infinite number of ways to allocate energy based on this feedback. This environment is **partially-observable**. Sensors provide feedback, but sensors are often noisy, late, and wrong so cannot truly observe the whole environment. Sensors within the systems also may not be able to tell you every detail of the system (i.e., is the light on in the cargo bay).

SECTION 6.5.2.3 CHOOSE THE REPRESENTATION(S) APPROPRIATE FOR THE DOMAIN

Identify which representation, or representations, from section 6.3 that is appropriate for describing the domain and the reasoning about for making decisions.

This problem could be represented with continuous variables and state-space vectors. The representation for this system will be determined by the sensor feedback in the form of continuous values representing current, voltage, power, etc.

In a more abstract way, this problem could also be represented as a graph. This graph would be less specific in terms of values. Some of the nodes in the graph would be things like "provide propulsion system with more power" or "power down weapons system."

SECTION 6.5.2.4 SELECT AND GO THROUGH A DECISION-MAKING PROCESS FOR THE DOMAIN

Identify which decision-making process is most appropriate given the representation selected in the last step, and step through some example decisions that would be made using that method. If one of the complex decision-making processes is used, go through that process: what decisions are made along the way and with what tools.

This problem could be solved with an optimization-based approach. This would involve defining objective-based functions to optimize the flow of power based on which resources required it.

SECTION 6.5.3 SUMMARY

While direct application of decision-support approaches to platform or fleet-based problems have not been widely published, the ingredients for such applications are at hand. Building off work done for operational considerations, two hypothetical decision problems were proposed, appropriate representations developed, and decision strategies enumerated. Given the mathematical nature of most digital twins, it is likely that the optimization and search-based decision frameworks will fit future decision problems well. However, an assessment of each problem's characteristics, against Table 5 at the end of section 6.4, is worthwhile as many other techniques may have strengths for specific applications.

SECTION 6.6 CONCLUSIONS

While a digital twin system does not need to include a decision-making component, decision-making as a discipline is highly tied to the types of information and models available in a digital twin. Thus, for a digital twin interfacing with a decision-making system it is important to understand the characteristics of both the physical process the twin applies to, as well as the twin itself, in order to pick the best decision-making framework. In this chapter, the characteristics of the decision-making framework were first reviewed in section 6.2 with a series of questions that can help clarify the type of domain the decision will apply to. Then in section 6.4, an overview of state-space representation approaches was made, covering the major approaches used by the decision-making methods. In section 6.5, eight common decision-making frameworks were briefly discussed, presenting their strengths, relationship to state-space descriptions, and the challenges of applying each to automated decision-making. From the combined information in section 6.5, it is possible to match potential decision frameworks with the problem description at hand for any particular digital twin solution.

Section 6.5 presents an overview of decision-making in the marine community. Automated decision-frameworks are in their infancy for autonomous platforms. To date, operational decisions have been the focus of this work, with navigation, oil-spill response and weather routing all studied in some depth. However, the application of decision-making to the general problem of maintaining a long-duration autonomous vessel is still largely unexplored. Two hypothetical examples, one covering a fleet problem and one covering a platform problem were presented to show how the information in this report can help structure the selection of a decision-making process. Using the information in this report, and this approach, should help engineers select appropriate decision frameworks when working with data-model fusion problems fed by digital twins.

SECTION 6.7 REFERENCES

[1] M. Helmert, "The fast downward planning system," *Journal of Artificial Intelligence Research,* vol. 26, pp. 191-246, 2006.

[2] S. Kim and D. Frangopol, "Multi-objective probabilistic optimum monitoring planning considering fatigue damage detection, maintenance, reliability, service life and cost," *Structural and Multidisciplinary Optimization,* no. 57, pp. 39-54, 2018.

[3] P. E. Hart, N. J. Nilson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions of Systems Science and Cybernetics,* vol. 4, no. 2, pp. 100-107, 1968.

[4] P. Bercher, S. Keen and S. Biundo, "Hybrid planning heuristics based on task decomposition graphs," in *Seventh Annual Symposium on Combinatorial Search*, 2014.

[5] B. Bonet and H. Geffner, "Planning as heuristic search," *Artificial Intelligence,* vol. 129, no. 1-2, pp. 5-33, 2001.

[6] D. Ferguson, M. Likhachev and A. Stentz, "A guide to heuristic-based path planning. In Proceedings of the international workshop on planning under uncertainty for autonomous systems, international conference on automated planning and scheduling," in *International Conference on Planning and Scheduling*, Monterey, 2005.

[7] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning,* vol. 8, no. 3-4, pp. 279-292, 1992.

[8] R. Bellman, "A Markovian Decision Process," *Indiana University Mathematics Journal,* vol. 6, no. 4, pp. 679-684, 1957.

[9] R. A. Howard, "Dynamic programmic and markov processes," 1960.

[10] D. C. England, "An Expert System for the Management of Hazardous Materials at a Naval Supply Center," Naval Postgraduate School, Monterey CA, 1990.

[11] R. Platt, R. L. Tedrake, L. P. Kaelbling and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," *Proceedings of the Robotics: Science and Systems Conference (RSS),* 2010.

[12] L. P. Kaelbling and T. Lozano-Pérez, "Integrated task and motion planning in belief space.," *The International Journal of Robotics Research,* vol. 32, no. 9-10, pp. 1194-1227, 2013.

[13] J. J. Chung, A. J. Smith, R. Skeele and G. A. Hollinger, "Risk-aware graph search with dynamic," *The International Journal of Robotics Research,* vol. 38, no. 2-3, pp. 182-195, 2019.

[14] S. Feyzabadi and S. Carpin, "Risk-aware path planning using hierarchical constrained markov decision processes," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, Taipei City, 2014.

[15] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI communications,* vol. 7, no. 1, pp. 39-59, 1994.

[16] M. Molineaux, M. Klenk and D. W. Aha, "Goal-Driven Autonomy in a Navy Strategy Simulation," KNEXUS Research Corp, Springfield VA, 2010.

[17] M. W. Simon Parsons, "Game Theory and Decision Theory in Multi-Agent Systems," *Autonomous Agents and Multi-Agent Systems,* pp. 243-254, 2002.

[18] J. S. R. Gilad Zlotkin, "Negotiation and Task Sharing Among Autonomous Agents in Cooperative Domains," *IJCAI,* pp. 912-917, 1989.

[19] S. P. Ketchpel, "Coalition formation among autonomous agents," in *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 1995, p. vol 957.

[20] Z. Yang, Z. Chen, K. Lee, E. Owens, M. Boufadel, C. An and E. Taylor, "Decision support tools for oil spill response (OSR-DSTs): Approaches, challenges, and future research perspectives," *Marine Pollution Bulletin,* no. 112313, 2021.

[21] T. Zis, H. Psaraftis and L. Ding, "Ship weather routing: A taxonomy and survey.," *Ocean Engineering,* no. 107697, 2020.

[22] A. Candeo, "“Industrial IoT lifecycle via digital twins”," in *2016 Int. Conf. Hardware/Software Codesign Syst. Synth. CODES+ISSS*, 2016.

[23] "Agent Environment in AI," 2018. [Online]. Available: https://www.javatpoint.com/agent-environment-in-ai.

[24] W. Bourne, R. Gallimard and J. Tunnicliffe, "Environments," 2006. [Online]. Available: www.doc.ic.ac.uk/project/examples/2005/163/g0516302/environments/environments.html.

[25] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey and K. Konolige, "The office marathon: Robust Navigation in an indoor office environment," *IEEE international conference on robotics and automation,* pp. 300-307, 2010.

[26] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer,* pp. 22, 46-57, 1989.

CHAPTER 7 APPENDIX

SECTION 7.1 ANNOTATED REFERENCES ON FUSION METHODS

In support of Chapter 1 Survey of Fusion Approaches and Opportunities.

Weymouth, Gabriel D., and Dick K. P. Yue. 2013. "Physics-Based Learning Models for Ship Hydrodynamics." Journal of Ship Research 57 (01): 1–12.

Describes physics-based learning models using a variety of kernel-based regression approaches supplemented by low-fidelity methods to approximate higher-fidelity simulation. Fairly far from most applied fusion systems as really only exploring fusion aspects, with little application.

Nielsen, Ulrik D., Zoran Lajic, and Jørgen J. Jensen. 2012. "Towards Fault-Tolerant Decision Support Systems for Ship Operator Guidance." Reliability Engineering & System Safety 104 (August): 1–14. https://doi.org/10.1016/j.ress.2012.04.009.

Examines differences between predicted ship motions and actually measured ship motions to identify possible faults in sensor data or the model. The approaches rely on a bipartite graph to coordinate different sensor readings to detect possible sensor faults. The formulation is developed for frequency domain motions with RAO seakeeping solutions, demonstrated on a large containership application.

Zhu, Jiandao, and Matthew Collette. 2017. "A Bayesian Approach for Shipboard Lifetime Wave Load Spectrum Updating." Structure and Infrastructure Engineering 13 (2): 298–312. https://doi.org/10.1080/15732479.2016.1165709.

Proposes a learned hierarchal Bayesian network (HBN) model to correct a design-stage loading prediction to agree with at-sea measurements. The scale, bandwidth, and skewness of the loading process are modeled explicitly, and corrections based on sea state, speed, and heading are developed. The model is able to forecast new operational conditions by interpolation in the HBN approach. The approach is tested with simulated data comparing frequency-domain and time-domain motions predictions.

Coraddu, Andrea, Luca Oneto, Francesco Baldi, Francesca Cipollini, Mehmet Atlar, and Stefano Savio. 2019. "Data-Driven Ship Digital Twin for Estimating the Speed Loss Caused by the Marine Fouling." Ocean Engineering 186 (August): 106063. https://doi.org/10.1016/j.oceaneng.2019.05.045.

Proposes using a variant of the neural network and deep extreme learning machine (DELM) to predict a vessel's speed in terms of 31 input variables. The difference between predicted and measured speed is used to evaluate the vessel's current speed loss owing to fouling. The approach is trained on real-world data from two commercial ships and is shown to outperform the simpler methods in the ISO standard for detecting fouling.

Behjat, Amir, Chen Zeng, Rahul Rai, Ion Matei, David Doermann, and Souma Chowdhury. 2020. "A Physics-Aware Learning Architecture with Input Transfer Networks for Predictive Modeling." Applied Soft Computing 96 (November): 106665. https://doi.org/10.1016/j.asoc.2020.106665.

Develops a physics-aware neural network approach for the control of simple UAVs. The physics model is downstream of several initial processing layers in the network and is able to blend both simplified physics predictions and machine learning to adapt to both changing UAV configuration and well as changing control inputs.

Jiang, Li, Sarah Signal, Bowen Jeffries, Brian Earley, Kurt Junghans, David Hess, and William Faller. 2020. "A Hydrodynamic Digital Twin Concept for Underwater Vehicles." In SNH 2020. Osaka, Japan.

Develop a neural network correction layer for a submarine maneuvering algorithm. A simplified hydrodynamic model is used to predict the future position of a submarine-based on control surfaces and power settings. A second machine learning layer then corrects this model output. The machine learning layer can learn when deployed to account for changes to the external configuration of the submarine. Notable as a prototype system and for the large number (>50) variables tracked by the machine learning approach.

Schirmann, Matt, Matt Collette, and James Gose. 2020. "Improved Vessel Motion Predictions Using Full-Scale Measurements and Data-Driven Models." In SNH 2020. Osaka, Japan.

Using a large (>10,000) point training set, a Neural Network correction model for simplified ship motion predictions is developed for real-world research vessel data. Even at the large data sizes used, including the physics-based prediction improved the overall accuracy of the system when the model followed the overall trends of the data (pitch and heave predictions). Where the model struggled (roll), pure data approaches were shown to be as good as physics-informed models.

SECTION 7.1.1 MECHANICAL AND BATTERY SYSTEMS

Liu, J., W. Wang, F. Ma, Y. B. Yang, and C. S. Yang. 2012. "A Data-Model-Fusion Prognostic Framework for Dynamic System State Forecasting." Engineering Applications of Artificial Intelligence, Special Section: Dependable System Modelling and Analysis, 25 (4): 814–23. https://doi.org/10.1016/j.engappai.2012.02.015.

Proposed a novel data-driven and model-driven combined model for battery health. A trained Neural Network is used for diagnosis, while a concurrent particle filtering method is used for longer-term prognosis through refining the parameters of a battery

107

aging model. The method is applied to a set of battery aging data gathered in an academic laboratory setting with good results.

Brahma, Indranil. 2019. "Extending the Range of Data-Based Empirical Models Used for Diesel Engine Calibration by Using Physics to Transform Feature Space." SAE International Journal of Engines 12 (2): 03-12-02–0014. https://doi.org/10.4271/03-12-02-0014.

A neural network approach to predict engine performance and emission data was extended with a simple model. Rather than couple the models in series or parallel, the simple model was used to produce a reduced feature set, then the neural network was trained on the reduced feature set, with improved performance. For the engine example shown here, even miscalibration of the simple model did not greatly reduce the accuracy of the result, indicating that the form of the model was most important.

SECTION 7.1.2 STRUCTURES

An, Dawn, Nam H. Kim, and Joo-Ho Choi. 2015. "Practical Options for Selecting Data-Driven or Physics-Based Prognostics Algorithms with Reviews." Reliability Engineering & System Safety 133 (January): 223–36. https://doi.org/10.1016/j.ress.2014.09.014.

Very good review paper of data-based and physics-based modeling. Using a crack growth example, neural networks and Gaussian process regression are compared to physics-based models with parameter updates using both particle filtering and Bayesian approaches. Has extensive references.

Hu, Zhen, Sankaran Mahadevan, and Dan Ao. 2018. "Uncertainty Aggregation and Reduction in Structure–Material Performance Prediction." Computational Mechanics 61 (1–2): 237–57. https://doi.org/10.1007/s00466-017-1448-6.

Concentrated only on the reduction of uncertainty from a small number of observations in systems with both a structural model and a material damage model. Uses Bayesian networks as a reduction strategy.

Magoga, Teresa, Seref Aksu, Stuart Cannon, Roberto Ojeda, and Giles Thomas. 2019. "Through-Life Hybrid Fatigue Assessment of Naval Ships." Ships and Offshore Structures 0 (0): 1–11. https://doi.org/10.1080/17445302.2018.1550900.

A complete example of fatigue life prediction for a high-speed aluminum vessel, including using FEA with monitoring histories, some updates from field inspection results. Fusion approaches are not discussed in detail.

Karimian, Seyed Fouad, Ramin Moradi, Sergio Cofre-Martel, Katrina M. Groth, and Mohammad Modarres. 2020. "Neural Network and Particle Filtering: A Hybrid Framework for Crack Propagation Prediction." ArXiv:2004.13556 [Eess, Stat], April. http://arxiv.org/abs/2004.13556.

Using a standard set of experimental data for shear in an aluminum lap joint, including ultrasonic data, a two-step fusion approach was proposed. First, a neural network uses the ultrasonic data to estimate the crack length. Then a Bayesian particle filtering approach is used to update crack growth parameters.

Karve, Pranav M., Yulin Guo, Berkcan Kapusuzoglu, Sankaran Mahadevan, and Mulugeta A. Haile. 2020. "Digital Twin Approach for Damage-Tolerant Mission Planning under Uncertainty." Engineering Fracture Mechanics 225 (February): 106766. https://doi.org/10.1016/j.engfracmech.2019.106766.

Provides a complete example of a mission replanning owing to fatigue crack growth. Both pitch-catch ultrasonic and high-resolution imaging are used to gauge the length of a crack in an aluminum panel. A Bayesian network fusion approach is then used to provide a future state prognosis. This is coupled with mission planning to ensure the crack stays below a critical size. Method proven with an experiment. Notable for the complete scope of the application.

Zhang, Hepeng, and Yong Deng. 2020. "Weighted Belief Function of Sensor Data Fusion in Engine Fault Diagnosis." Soft Computing 24 (3): 2329–39. https://doi.org/10.1007/s00500-019-04063-7.

Uses Dempster-Shafer theory with a new belief assignment method to examine engine sensor data and change belief state in a list of specific, discrete failure modes. Builds off of previous work in belief theory with a new weighting scheme. Very different fusion approaches from other systems.

Kapteyn, Michael G., and Karen E. Willcox. 2020. "From Physics-Based Models to Predictive Digital Twins via Interpretable Machine Learning." ArXiv:2004.11356 [Cs], April. http://arxiv.org/abs/2004.11356.

Studied damage identification in UAV composite wing structure. A classification tree approach was used. The trees were trained by an FEA simulation of the wing with different levels of damage. Then, readings from 20 strain gauges were used to classify the damage state of the wing. An advantage of the method is that the machine learning approach is intelligible to the user, owing to the structure of the decision trees.

Kapteyn, Michael G., Jacob V. R. Pretorius, and Karen E. Willcox. 2020. "A Probabilistic Graphical Model Foundation for Enabling Predictive Digital Twins at Scale." ArXiv:2012.05841 [Cs, Math], December. http://arxiv.org/abs/2012.05841.

Presents an overall Bayesian Network approach for calibrating a digital twin for an AUV through a series of experiments before flight. The goal is to remove the initial uncertainty in the twin, so it can be trusted before being used in service (this part of the approach is not described in detail). Only select details of the case study are presented. No details of the Bayesian network approach are covered.

Bazilevs, Y., X. Deng, A. Korobenko, F. Lanza di Scalea, M. D. Todd, and S. G. Taylor. 2015. "Isogeometric Fatigue Damage Prediction in Large-Scale Composite Structures Driven by Dynamic Sensor Data." Journal of Applied Mechanics 82 (9): 091008. https://doi.org/10.1115/1.4030795.

Presents a composite damage evolution model for large wind turbine blade structures. A new FEA formulation and damage model are developed. Then, the FEA model is used to simulate a fatigue test on an actual turbine blade. At four times during the experiment, the FEA model is calibrated to match the experimental process by modifying the tip displacement, so an accelerometer on the blade matches the accelerations predicted by the FEA model. This fusion is done by simply iterating through displacements until a match is achieved.

Straub, Daniel. 2009. "Stochastic Modeling of Deterioration Processes through Dynamic Bayesian Networks." Journal of Engineering Mechanics 135 (10): 1089. https://doi.org/10.1061/(ASCE)EM.1943-7889.0000024.

Proposed dynamic Bayesian networks as a fusion approach for general structural degradation problems. The approach is then developed in detail for a series of structural fatigue problems. Reliability calculation through the network shown to be roughly equal to that of conventional methods, but the network is much more powerful for fusing in-service measurements. Discretization is also discussed in detail; however, only synthetic data is used in the evaluation and demonstration.

Gockel, Brian, Andrew Tudor, Mark Brandyberry, Ravi Penmetsa, and Eric Tuegel. 2012. "Challenges with Structural Life Forecasting Using Realistic Mission Profiles." In 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. Structures, Structural Dynamics, and Materials and Co-Located Conferences. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2012-1813.

Presents a forward-only digital twin model for an aircraft. Based on recorded flight parameters, a CFD simulation is run to determine plane loading and possible damage. Only a simple, short snippet of an actual flight is used. The method has significant computational challenges, which are discussed in the paper. No reflective fusion is used.

Luque, Jesus, and Daniel Straub. 2016. "Reliability Analysis and Updating of Deteriorating Systems with Dynamic Bayesian Networks." Structural Safety 62 (September): 34–46. https://doi.org/10.1016/j.strusafe.2016.03.004.

Expands on previous work by Straub on using Bayesian Networks to model structural deterioration. Structural systems are considered, including a Daniels system of evenly loaded tension rods, and a complex offshore frame structure. A hierarchal Bayesian network approach is used, where common parameters between multiple structural members are correlated. Inference approaches for this network structure are developed, and updating with simulated data is demonstrated for both structures.

Groden, Mark, and Matt Collette. 2017. "Fusing Fleet In-Service Measurements Using Bayesian Networks." Marine Structures 54 (Supplement C): 38–49. https://doi.org/10.1016/j.marstruc.2017.03.001.

Presents a Bayesian network approach for fatigue inference on stiffened panels. Both the number of cracks and the permanent set of the panel are used to update loading and fatigue parameters for the panel. The ability of the network to fuse in-service measurements to make prognosis is studied. Generally, the network can improve future prognosis, but it requires a large number of observations to be successful. Only simulated data is used to evaluate the network.

Pegg, N.G., and S. Gibson. 1997. "Application of Advanced Analysis Methods to the Life Cycle Management of Ship Structures." Dartmouth, Nova Scotia, Canada: Defence Research Establishment Atlantic.

Presents one of the earliest twin-like structural health monitoring systems for naval vessels. A central database of loading and FEA models is updated by in-service inspections results, including corrosion and fatigue cracking. With the updated geometry, operability, safety, and repair decisions are made. The models are forward-only in that the ship's current condition and load history are used to forecast future performance without correcting the underlying simulation models.

Zhu, Jiandao, and Matthew Collette. 2015. "Updating Structural Engineering Models with In-Service Data: Approaches and Implications for the Naval Community." Naval Engineers Journal 127 (1): 63–74.

Expands upon previous work on load updating to couple the load updating model to a fatigue crack growth model. This model uses a dynamic Bayesian network to fuse crack size measurements with fatigue crack growth parameters, adjusting the growth parameters to provide prognosis of future crack size. The resulting crack size is used in a time-varying reliability plot to look at the safety of the vessel in future operations. The loads from the load updating method are used as inputs and not further adjusted during the dynamic Bayesian network part of the process.

Tygesen, Ulf T., Michael S. Jepsen, Jonas Vestermark, Niels Dollerup, and Anne Pedersen. 2018. "The True Digital Twin Concept for Fatigue Re-Assessment of Marine Structures." In. American Society of Mechanical Engineers Digital Collection. https://doi.org/10.1115/OMAE2018-77915.

Tygesen, U. T., K. Worden, T. Rogers, G. Manson, and E. J. Cross. 2019. "State-of-the-Art and Future Directions for Predictive Modelling of Offshore Structure Dynamics Using Machine Learning." In Dynamics of Civil Structures, Volume 2, edited by Shamim Pakzad, 223–33. Conference Proceedings of the Society for Experimental Mechanics Series. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-74421-6_30.

These two papers together present a 5-stage digital twin approach for offshore structures. The approach is sequential, covering first determining the errors between a digital FEA model of the structure and the response of the structure as measured in-service. A Bayesian updating approach is used on the FEA model, changing mass, stiffness, and other element properties so that the predicted and measured responses are brought into agreement. A similar validation-updating approach is then used for wave loading, although here, a simpler updating method is used based on the coefficients in Morrison's equation for drag on circular shapes. Finally, the ability of the tuned model to answer life-extension questions is demonstrated.

Mondoro, Alysson, Mohamed Soliman, and Dan M. Frangopol. 2016. "Prediction of Structural Response of Naval Vessels Based on Available Structural Health Monitoring Data." Ocean Engineering 125 (October): 295–307. https://doi.org/10.1016/j.oceaneng.2016.08.012.

Presents a fitting approach to predict the stress spectrum in conditions not yet encountered based upon measurements in a smaller number of conditions. A regression approach is taken, based on the forms of the wave energy spectra, accounting for both low and high-frequency energy components. Based on at-sea data for a large catamaran, the approach is shown to be quick and practical for developing custom fatigue stress spectra.

Stull, Christopher J., Christopher J. Earls, and Phaedon-Stelios Koutsourelakis. 2011. "Model-Based Structural Health Monitoring of Naval Ship Hulls." Computer Methods in Applied Mechanics and Engineering 200 (9–12): 1137–49. https://doi.org/10.1016/j.cma.2010.11.018.

Using a rapid FEA code and measured displacements, this fusion method attempts to infer the location of corrosion damage or internal structural damage. A range of inverse search approaches are discussed, and the method is successful at approximating most damage cases when applied to a simplified hull.

Kapteyn, M. G., D. J. Knezevic, D. B. P. Huynh, M. Tran, and K. E. Willcox. 2020. "Data-Driven Physics-Based Digital Twins via a Library of Component-Based Reduced-Order Models." International Journal for Numerical Methods in Engineering n/a (n/a). https://doi.org/10.1002/nme.6423.

Presents a multi-phase approach to data model fusion. A larger aircraft FEA model is broken into segments, and a reduced FEA model is made of each segment. Each segment is simulated with different levels of structural damage, which are assumed to be linked temporarily. A hidden Markov Model is used to estimate the transition between damage states, and 24 strain measurements from in-flight data are used to update the state probability as the aircraft flies.

Vega, Manuel A., Zhen Hu, and Michael D. Todd. 2020. "Optimal Maintenance Decisions for Deteriorating Quoin Blocks in Miter Gates Subject to Uncertainty in the Condition Rating Protocol." Reliability Engineering & System Safety 204 (December): 107147. https://doi.org/10.1016/j.ress.2020.107147.

Mitre lock gate health was assessed by fusing subjective expert inspection with strain gauge data from the gates. The gates move through six stages of deterioration, as ranked by inspectors. However, the transition between states and assignments is not always exact. Using Bayesian updating from the measured strain state in the gate and an FEA model of the structure, more accuracy in state estimation is achieved. This is then fed into an optimal maintenance algorithm which weighs the costs of different actions to decide on an optimal path.

Dourado, Arinan, and Felipe A. C. Viana. 2020. "Physics-Informed Neural Networks for Missing Physics Estimation in Cumulative Damage Models: A Case Study in Corrosion Fatigue." Journal of Computing and Information Science in Engineering 20 (061007). https://doi.org/10.1115/1.4047173.

Constructs a deep Neural Network for predicting aircraft corrosion and fatigue crack growth. A simplified physical model is inserted as a special type of node in the network, and layers containing this type of node alternate with layers that are purely data-driven. The resulting network is then trained on partial fleet data and used to predict crack sizes in the future.

113