Digital Twins: State-of-the-Art and Future Directions for Modeling and Simulation in Engineering Dynamics Applications

D. J. Wagg
Professor
Dynamics Research Group,
Department of Mechanical Engineering,
University of Sheffield,
Sheffield S1 3JD, UK
e-mail: david.wagg@sheffield.ac.uk

K. Worden
Professor
Dynamics Research Group,
Department of Mechanical Engineering,
University of Sheffield,
Sheffield S1 3JD, UK
e-mail: k.worden@sheffield.ac.uk

R. J. Barthorpe
Dynamics Research Group,
Department of Mechanical Engineering,
University of Sheffield,
Sheffield S1 3JD, UK
e-mail: r.j.barthorpe@sheffield.ac.uk

P. Gardner
Dynamics Research Group,
Department of Mechanical Engineering,
University of Sheffield,
Sheffield S1 3JD, UK
e-mail: p.gardner@sheffield.ac.uk

This paper presents a review of the state of the art for digital twins in the application domain of engineering dynamics. The focus on applications in dynamics is because: (i) they offer some of the most challenging aspects of creating an effective digital twin, and (ii) they are relevant to important industrial applications such as energy generation and transport systems. The history of the digital twin is discussed first, along with a review of the associated literature; the process of synthesizing a digital twin is then considered, including definition of the aims and objectives of the digital twin. An example of the asset management phase for a wind turbine is included in order to demonstrate how the synthesis process might be applied in practice. In order to illustrate modeling issues arising in the construction of a digital twin, a detailed case study is presented, based on a physical twin, which is a small-scale three-story structure. This case study shows the progression toward a digital twin highlighting key processes including system identification, data-augmented modeling, and verification and validation. Finally, a discussion of some open research problems and technological challenges is given, including workflow, joints, uncertainty management, and the quantification of trust. In a companion paper, as part of this special issue, a mathematical framework for digital twin applications is developed, and together the authors believe this represents a firm framework for developing digital twin applications in the area of engineering dynamics. [DOI: 10.1115/1.4046739]

1 Introduction

Society is experiencing an era of *digital transformation*. It is now common to hear concepts discussed in the technical literature and wider media relating to this transition. Concepts such as *Industry 4.0*, *the Internet-of-Things* [1], and *Big* Data [2], have become increasingly widely used, particularly in relation to engineering applications. Often mentioned in this context, and promoted as a potentially transformative idea for engineers working in all areas, is the idea of a *digital twin*. In this paper, the focus will be on modeling and simulation, and in this context, a digital twin can be defined as a virtual duplicate of a system built from a fusion of models and data. This is made possible by combining models and data using state-of-the-art algorithms, expert knowledge, and digital connectivity. The potential benefit of the digital twin is a significant improvement in predictive capability compared with current technologies.

Like all areas of modern endeavor, the vast majority of engineering applications are becoming increasingly reliant on computing, for example, creating numerical simulations that are used to inform decisions about the design and management of key components, structures, and systems. In the last few decades, high-performance computing (HPC) has been employed extensively to build increasingly high-fidelity models in the belief that this would remove *model form* uncertainties associated with the engineering application being considered. While this has given considerable benefit, there are still a large number of engineering problems with high levels of uncertainty even after the application of HPC [3], and this serves to dispel the idea that increasing levels of model fidelity is a panacea, although it is undoubtedly helpful in many situations. As a result, obtaining a useful virtual model is no longer a question of increasing model fidelity, but now rests in the more difficult problem of developing *trust* (or conversely dealing with the remaining uncertainties) in the model(s) through other means.

An important technical example where this situation occurs is in the problem of modeling mechanical joints. The physics associated with mechanical joints is still the subject of considerable research, and as a result, physics-based models are subject to considerable epistemic uncertainty. One reason for this situation is that many of the physical processes happen at the tribological scale (microns), whereas the modeling of the whole joint, and the rest of the structural behavior, is required at much larger (macro) scales. Common phenomena like friction and hysteresis are difficult to model for the same reason. In addition, systems operating in dynamic environments are often highly sensitive to very small disturbances to the structure (typically assumed to be aleatory uncertainties). In the case of joints, for example, small differences in tolerances, and other joint properties, such as friction, are highly sensitive to temperature variations in the operating environment, which can all lead to large deviations in the dynamic behavior of a jointed structure. From a modeling perspective, it is very difficult to bring together models of all these different physical processes, and their associated uncertainties, which happen at different length scales, into an accurate model of a complete structure, even when large amounts of computing power are available.

In parallel, an organizational example of the problems faced in creating effective simulations of modern engineering applications occurs in the way problems are analyzed, designed, and simulated as subsystems. This is a natural approach because most modern engineering systems are highly complex, and as a result, it makes

Manuscript received April 30, 2019; final manuscript received February 18, 2020; published online May 12, 2020. Assoc. Editor: Sifeng Bi.

sense to have multiple teams of experts carrying out computations of the subsystems in parallel. However, once this type of division is made, there is a natural tendency for the teams to work in *silos*. This silo effect, combined with the fact that the subsystems are often defined based on the different physics or scales involved, means that the resulting subsystem models often cannot be unified into a model of the complete application. This mixing of technical objectives with inappropriate organizational culture can lead to undesirable outcomes such as *analysis paralysis* [4], among others.

The main transformative aspect of the digital twin is to improve predictive capability by augmenting computational models using data; this again reflects the wider digital transformations happening in society. Analysis of data, particularly through internet and social media applications, has been a very important modern phenomenon. For example, techniques such as machine learning are now used in order to provide bespoke targeting of consumer behavior, such as advertising and other related activities. In engineering, advancements in sensor technology mean that many systems now have the potential to gather and process very large amounts of data. Structures are increasingly being built with sensors embedded, and this combined with advances in structural health monitoring and associated *data-based* techniques, means that the potential to exploit information obtained from data is rapidly increasing [5].

For the purposes of the applications considered here, the main idea of the digital twin is to combine these model-based and data-based approaches to create a virtual prediction tool that can evolve over time. In doing so, the digital twin concept offers the potential to assist in engineering applications for both technical and organizational problems, such as the two examples mentioned above. In the technical example, the main idea would be to reduce the epistemic uncertainties from the limitations of the physics-based modeling, using data. These data would be obtained from the real structure, which is called the *physical twin*, or laboratory tests using components from the structure—in either case, it is important that the data gathered are specific to the structure being twinned, as the digital twin is *entirely bespoke* to this structure. To address the organizational example, the digital twin concept incorporates a hierarchical format, enabling multiscale and multiphysics processes to be incorporated, but most importantly a highly connected organizational framework that should offer solutions to the problem of silos, and related cultural issues. The digital twin approach also seeks to break down unhelpful organizational barriers (i.e., improve connectivity) by providing a logical interface of outputs and inputs from different computational models (in different silos), ideally by using robust Verification and Validation (V&V) methods that build trust in the subsystems prior to the assembly of these into a full system digital twin.

In terms of maturity, the digital twin is a relatively new idea, one that has attracted significant attention in many areas of engineering and beyond; it offers a range of highly attractive potential solutions to engineers who are tasked with designing and managing ever more complex engineering systems. However, there are substantial challenges to be overcome in order for digital twin technology to reach full maturity.

The aim of this paper is twofold; first, it is to assess the current state of the art of digital twins when applied to engineering systems with time-dependent (i.e., dynamic) behavior; second, is to summarize the outstanding open research problems and technological challenges. The reason for focusing on applications in dynamics is that: (i) they offer some of the most challenging aspects of creating an effective digital twin, and (ii) they are relevant to important industrial applications such as energy generation and transport systems. In a companion paper, as part of this special issue, a mathematical framework for digital twin applications is developed, and together the authors believe this represents a firm framework for developing digital twin applications in the area of engineering dynamics.

The paper is structured as follows: In Sec. 2 the background to, and history of, the digital twin will be discussed, including examples of the current state of the art in engineering dynamics. In Sec. 3, the process of synthesizing a digital twin is discussed in detail. Then, in Sec. 5, an example of a simulation digital twin for the asset management phase of a wind turbine structure is presented. In Sec. 6, a case study of a digital twin of a small-scale three-story building is presented, in order to demonstrate how a selection of model and data-based algorithms can be unified into the digital twin. After that, open research problems and technological challenges are discussed in Sec. 7, before the conclusions are given in Sec. 8.

2 History and Background to the Digital Twin

The origins of the twinning concept have been attributed by some authors [6], to the work of NASA during the Apollo program. The term digital twin appears to have developed from work relating to product lifecycle management (see Ref. [7] and references therein), although other names were being used for similar concepts in other domains at around the same time, for example, *digital counterpart* [8], *virtual engine* [9], or *intelligent prognostics tool* [10], among others [11]. The term digital twin captured the zeitgeist and as a result is now typically taken as a generic term to encompass all these related phrases, although, as previously stated above, the meaning relies heavily on the specific context involved. The idea has received considerable attention since then in the area of product design, with particular overlap with existing digital design tools such as computer-aided design (CAD) [12,13], big data and data-driven design [14–17], knowledge graphs and relations to ontologies [18,19], middleware [20,21], and blockchain [22].

The concept has also been considered extensively in the domain of manufacturing processes, including autonomous manufacturing [6,11], real-time manufacturing [23], computer-aided design [24,25], additive manufacturing [26,27], and more general innovations in manufacturing processes including links to cyber-physical systems [28–40].

In terms of asset management, digital twins have been considered for tasks such as damage detection and structural-health/condition monitoring [10,28,41] and uncertainty quantification (UQ) [42]. In addition to the application areas already mentioned, digital twins have also been considered for application in the areas of offshore drilling [43], offshore wind turbines [44,45], space structures [46], and nuclear fusion [47].

An important consideration for the concept is, how the digital twin relates to the life-cycle of the product or process in question [48]. The majority of applications cited above are applied to manage the performance of an engineering application after its design and manufacture, but a digital twin would ideally be delivered with the product at the start of its operational life, and would also capture all aspects of the manufacturing process [3]. Therefore, whenever possible, the digital twin would need to be first implemented during the *design phase*, and persist throughout the entire operational life of the product (which is called the *asset management phase*) [49]. In both lifecycle phases, valuable information may be provided by data or models aggregated from similar structures, or even from the wider population. It is anticipated that for engineering applications, one of the most important high-level objectives that a digital twin can be used for is structural life prediction. Examples including the current state of the art in engineering dynamics are considered next.

2.1 Structural Life Prediction Using a Digital Twin.

In 2011, Tuegel et al. proposed a new way of estimating the life of an aircraft [3]. The authors imagined a future scenario where every new aircraft was delivered with a digital twin. The digital twin would represent the real aircraft (the physical twin) so closely that it could, for example, include the effects of manufacturing anomalies, and details of the material microstructure. As a result, the digital twin could be used to give ultrarealistic predictions about the life of the aircraft.

Of course, this vision of ultrahigh fidelity modeling has been a long-held ambition in many industrial design sectors. The example put forward by Tuegel et al. was distinguished, not only because it proposed the digital twin as a solution but also because it articulated some of the key challenges to achieving this vision.

There are three important problems that Tuegel et al. [3] describe that are common for a wide range of engineering applications. The first problem is that of multiscale modeling—called by some *the tyranny of scales* [50]. This term refers to the problem of modeling the behavior of physical phenomena that display radically different, dominant behaviors at different length scales. This issue is also closely linked to the problem of dealing with different types of physical modeling at different scales (or domains), and creating effective interfaces between them—often given the catch-all label of *multiphysics modeling*. The second problem Tuegel et al. identified is the *gap between hardware capability and software performance*, something recognized in the HPC research community, and a major factor in limiting the ability of engineers to harness the full benefit of increasing amounts of computing power. The third problem is that of *historical processes* during the design stage, with the result that the historical nature of the process is a restriction to progress.

In particular, digital computing has been applied to design and analysis to make computations faster, more efficient, and of higher resolution than previously possible; but often, the design process is still based on the predigital computer methods. Furthermore, rather than offering more freedom to designers of complex engineering systems, the rapid advancement of computational methods has meant that designers are increasingly locked into existing processes. This situation is often, in large part, because of the necessity to do many parts of the design in parallel. Typically, large teams of engineers will work on just one part of the overall system. This practice often creates silos that as the computational methods become increasingly sophisticated, become so deeply engrained, that any form of integration with other parts of the design process becomes extremely difficult.

Furthermore, the pursuit of a digital twin will involve improving physics-based modeling techniques. A key area of improvement will be geometry adaptation and morphing throughout the life of the structure. This may be required in order to capture behaviors due to manufacturing anomalies as stated by Tuegel et al. [3]. The ability to have CAD representations that are a one-to-one mapping of the physical twin will be necessary for certain models. In addition, with multiple models integrated to generate a digital twin, links such as joint models will play a vital role. Joints pose a major challenge because a large portion of modeling difficulties will come from subsystem interactions. Solutions to these problems may not lie completely in physics-based modeling itself. Data augmentation may provide an additional avenue for correcting physics-based models so that they more closely reflect the physical twin. This crucial "building block" interacts with all others, and will be discussed further in Sec. 5.5.

2.2 Verification and Validation Using Digital Twins. For engineering dynamics, there is a well-established set of techniques for V&V. More specifically, as most dynamics applications are assumed to have linear dynamics, *modal analysis and testing* has become the defacto method for validation against measured data, for example, see Ref. [51] and references therein. This methodology makes a direct connection between the model(s) and the measured data using the concept of *modes of vibration*. In fact, the methods have been extended so that operational modal analysis can be applied using only response data recorded from the structure under normal operation conditions [52]. More generally, the vibration modes can be interpreted in both a physics-based model context (typically a finite element model representing the geometric and material properties of the system) and as an identification technique (or data-based model).

A more general framework for verification and validation processes encompasses the concepts of white, gray and black-box models [53,54]. Starting from the assumption that a model can be built with physics-based reasoning, then the object of interest is called a "white-box model." At the other end of the spectrum, "black-box" models are derived entirely from measured data, with no assumed knowledge of the physics at all. In between these two extremes, gray-box models are a combination of both physics-based reasoning and data. This combination of models and data is exactly the format required for a digital twin. That said, it is natural to ask: "what is the difference between the digital twin and a validated model?" The answer will be context specific, but a digital twin will typically be time-evolving and make much more extensive use of data [3].

In structural dynamics and other branches of computational mechanics, there have been many previous advancements in this area. For example, finite element updating methods adjust model parameters based on experimental observations, in order to match the model parameters to the measured experimental system [55]. This type of *model updating* will need to be a key functionality of the digital twin, with frequent updates, ultimately in near real-time, creating the time-evolving property required of the twin. This would provide a mechanism for performing structural health monitoring and would aid asset management decision making. In combination with updating, the digital twin can make use of a range of data-based algorithms, for example, to carry out condition monitoring of the structure, based on an evolving history of measured data. Already, machine learning methods are proving to be some of the most productive algorithms used for this purpose [5,56], and this will continue to develop as a key part of digital twin technology, for example, see Refs. [57] and [58].

In recent years, a number of application-specific guidelines have been proposed for implementing the model validation process (verification is discussed in Sec. 5.3). For example, one of the first such frameworks to focus on physics-based engineering models was that produced in 1999 by the AIAA for computational fluid dynamics problems [59]. These frontrunners have been followed more recently by a series of standards introduced by the American Society of Mechanical Engineers (ASME), currently comprising the ASME Guide for V&V in Computational Solid Mechanics in 2006 [60] and the Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer in 2009 [61]. These documents provide a firm basis for the application of validation methods, and many aspects of these frameworks can be transferred to dynamic problems. However, validation of nonlinear dynamical models presents additional challenges that are yet to be fully addressed, and an issue of particular interest is how to account for potential bifurcations in the response of a nonlinear system.

3 Synthesizing a Digital Twin

As discussed in Secs. 1 and 2, validated models and process control are both natural starting point for synthesizing a digital twin. Of course, the digital twin is much more than just a validated model or a control process. In this context, a digital twin needs to be a robustly validated, time-evolving virtual duplicate of the physical twin that aids decision making. Ideally, the digital twin would be synthesized during the design phase, and continue to evolve during manufacture, commissioning, operation, and finally decommissioning.

3.1 Process Control and Condition Monitoring. It is important to note that some aspects of the digital twin concept have evolved from condition monitoring of plant, or supervision of other processes (i.e., process control). At a most basic level, *supervision* is the first desirable aim; beyond this, many industrial plant and asset management systems have highly developed

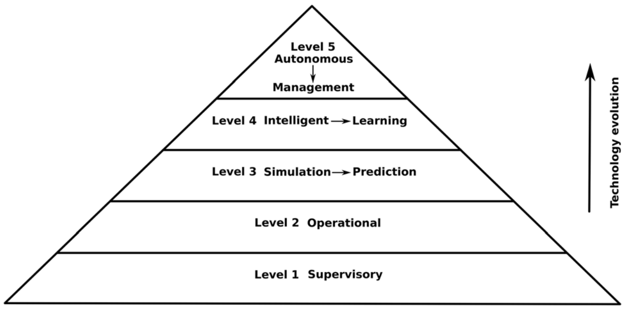

Fig. 1 A capabilities hierarchy for digital twins, where each level incorporates all the previous capabilities of the levels below

operational capabilities. This type of interactive capability represents the second category, which will be called *operational*, meaning that the operational decisions are informed and supported by relevant information. Both supervision and operational capabilities are long established, and although some authors mention these as digital twins, here they are considered to be *predigital twins*, meaning a system that has the capability to be a digital twin but currently does not contain all the essential elements (where essential elements means those elements that give the required functionalities required of a digital twin, which in this paper are taken to be; simulation; learning and management).

The next level of sophistication is that described by Tuegel et al. [3], and is categorized as a *simulation digital twin*. It is important to note that this typically incorporates both supervision and operation into its processes as well as simulation. In this sense, it builds on and enhances the predigital twin capabilities by adding the ability to simulate, based on models and data, the physical twin. This type of digital twin will also be able to allow the user to visualize a graphical interpretation of the physical twin, and carry out predictions to support design or operational decisions. As stated in the Introduction, here a key requirement of a simulation digital twin is that it should be able to provide the user with a quantitative assessment of the level of trust (via uncertainty quantification) for each simulation or prediction it produces.

Building on the concept of a simulation digital twin (or simulation twin) are two more levels of sophistication, both of which are currently aspirations for the digital twin. The first advance is to add an increased degree of "intelligence" to the digital twin, to give an *intelligent digital twin*. This object includes all the capabilities of the simulation twin, adds the ability to learn from data (via machine learning), and also adds increased levels of decision support and scenario planning.

The final level of sophistication is the digital twin that allows the physical twin to be autonomous. As before, the digital twin would include all previous capabilities, and add the ability for the twin to carry out all decision-making (within prescribed parameters) and manage the asset concerned with minimal human intervention. There is also the possibility of adding higher levels of learning and intelligence capabilities, via artificial intelligence techniques, although this is not discussed here.

The hierarchy of possible capabilities is shown in Fig. 1.

A key distinguishing feature of a digital twin (and hence the dividing line between levels 2 and 3 in Fig. 1) is that it can be used as a predictive tool. A process control interpretation naturally relates to asset management tasks, but aims for the twin can also be defined in the design phase, as will be discussed later in Sec. 4.1.

4 Context Specific Aim and Objectives of a Digital Twin

For nearly all applications, the primary aim of creating a digital twin is to enable the user to have as much information as possible about the current status and future behavior of the physical twin such that optimal decisions can be made. The precise objectives of the digital twin will depend on the context that is required, but a typical simulation twin for a dynamics application might allow the user to:

(1) quickly understand the outputs with fast (possibly real-time if required) visualization of results;
(2) incorporate and update the geometry of the digital twin through integrated CAD;
(3) navigate through the CAD model to specific components or subassemblies of interest and perform isolated tasks;
(4) identify spurious behavior, potential damage, or the need for system maintenance;
(5) view a hierarchical representation of physical behavior at different length scales;
(6) interrogate the current state of the structure, whether in real-time or historically, and perform data analysis (diagnosis);
(7) simulate future scenarios to make predictions (prognosis and decision support);
(8) design controllers, perform hardware-in-the-loop simulation, and/or set control processes for the physical twin;
(9) quantify a level of confidence (trust) that can be given to simulation outputs.

Note that the abilities to predict future outcomes, and quantify the level of confidence in these predictions, are particularly important features. The synthesis of a simulation digital twin during first the design, and then the asset management phases, is now considered.

4.1 Digital Twins During the Product Design Phase. The design phase is considered first here, as envisaged, for example, by Tuegel et al. [3], where a new product (an aircraft in the case cited) is delivered to the customer with a digital twin that can then be used for asset management. Design processes are also context dependent, and for the broad context of dynamics applications, a typical standpoint is to use the "Design V" model as shown in Fig. 2, that emphasizes the role of verification and validation during the design manufacture and commissioning.

Figure 2(*a*) shows the traditional V model, where starting at the top left with customer requirements, the design is first developed going down the left-hand part of the V to manufacture. The product is then verified and validated as the process continues up the right-hand side of the V until commissioning is complete. In this context, verifying is checking that all the tasks in the process are carried out correctly (Fig. 3: did we build the thing right?), and validation is checking to see that the final product delivers the required overall performance (Fig. 3: did we build the right thing?). Figure 2(*b*) shows how the V model can be modified to include a digital twin cycle. In this scenario, the verification and validation process is used to build a digital twin, starting with component-level testing data, and progressing to subsystem and finally full (or as full as possible), system tests. Note also that a new step can be included for a first stage validation of the digital twin, shown in Fig. 2(*b*) as the culmination of the digital twin cycle. This is a first-stage validation, because the digital twin will need to be regularly revalidated through-out its life, in order to ensure that it can continue to deliver highly trusted outputs.

As the digital twin is a combination of models and data, the first stage of the cycle shown in Fig. 2(*b*) is the development of physics-based models through the detailed design phase. These models are then augmented with data collected from the product testing and commissioning phase to build a product-specific digital twin. A specific example of this type of data augmentation process will be given in Sec. 5.5.

To be more specific, the initial design phase can be separated from the virtual modeling and commissioning phases, as shown schematically in Fig. 3, where the model now resembles a W. In

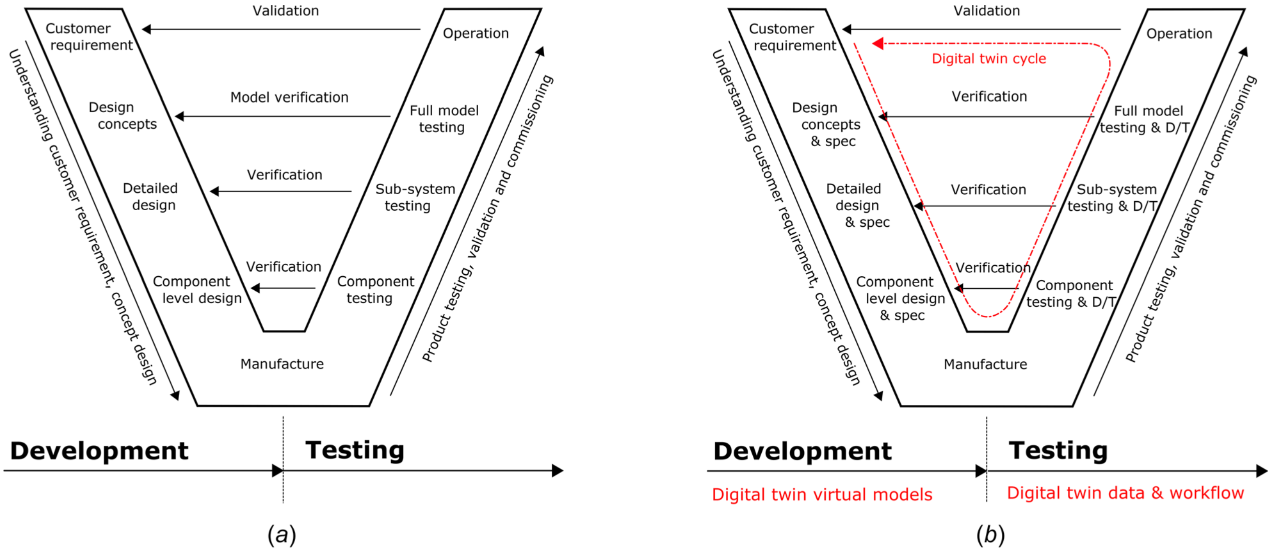

Fig. 2 Schematic representation of the V model for product design. (*a*) The traditional V model, and (*b*) the V model with a digital twin cycle added. Note that D/T is shorthand for notation for digital twin, and P/T is shorthand notation for physical twin.

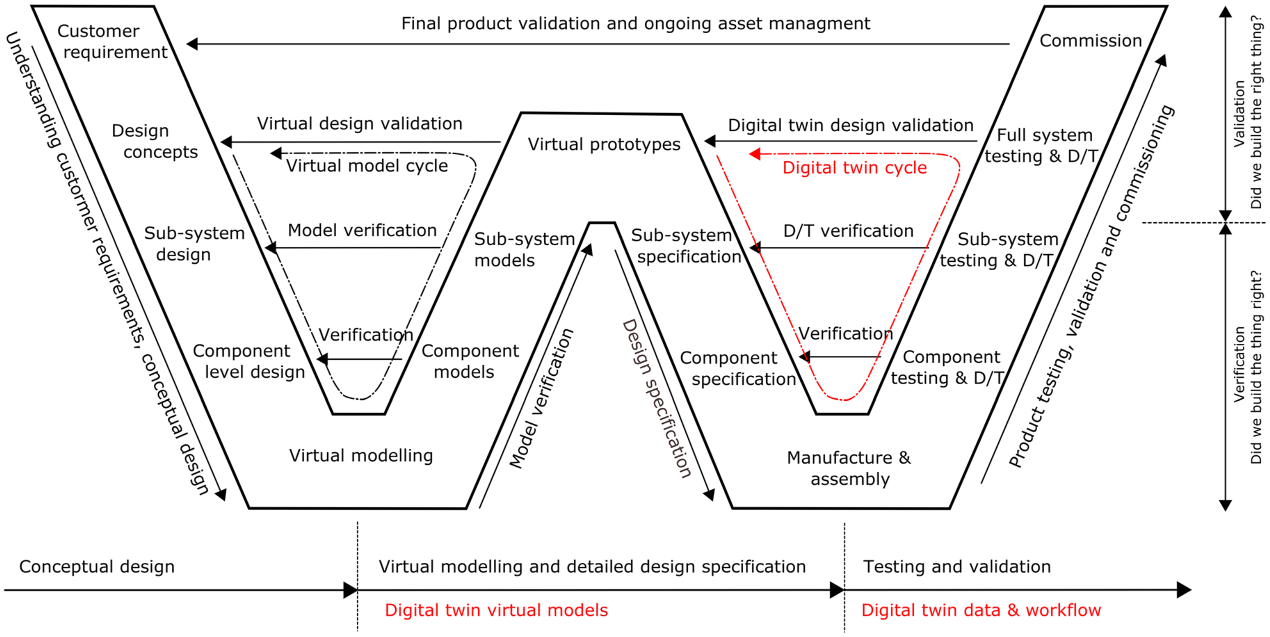

Fig. 3 Schematic representation of the W model for product design. In this case, a specific virtual prototyping stage is included. The virtual prototype is then used as the basis for a digital twin in the second cycle.

this case, a specific virtual prototyping stage is included that precedes the testing and validation phase. The virtual prototypes then form the basis for synthesizing the digital twin in the second part of the cycle. Note that the idea of a W model has been previously proposed in the context of software engineering [62–64]. The concept is somewhat different from the idea proposed here. For example in Ref. [62], the W model defines one V for the component development process, while the other V is for the system development process, and these two V's are integrated into a single overall method.

In this context, the primary aim of the digital twin is to reduce uncertainties by incorporating component/subsystem data, and where possible, shorten the testing and validation phase for the full system based on the assumed reduction in uncertainty. Another important consideration is how the digital twin will transfer into the asset management phase, and this is discussed next.

4.2 Digital Twins During the Asset Management Phase.

Assuming the digital twin has already been synthesized during the design phase, it then needs to be extended into the asset management phase. To do this, the required digital twin capability level is first selected, typically levels 3, 4, or 5, as shown in Fig. 1. Then, depending on the context and the required functionality, the *essential elements* are selected, based on the required aim and objectives of the digital twin. For the selected essential elements, a matrix of building blocks can be created, and a representative example is shown in Fig. 4, that for the purposes of giving insight, includes all the capability levels from Fig. 1.

Here, it can be seen that the predigital twins do not contain all the essential elements required for a digital twin; neither do they have the key distinguishing features of a digital twin, namely, the ability to predict, learn and manage. Within the matrix, individual building blocks are shown, although it should be noted that these

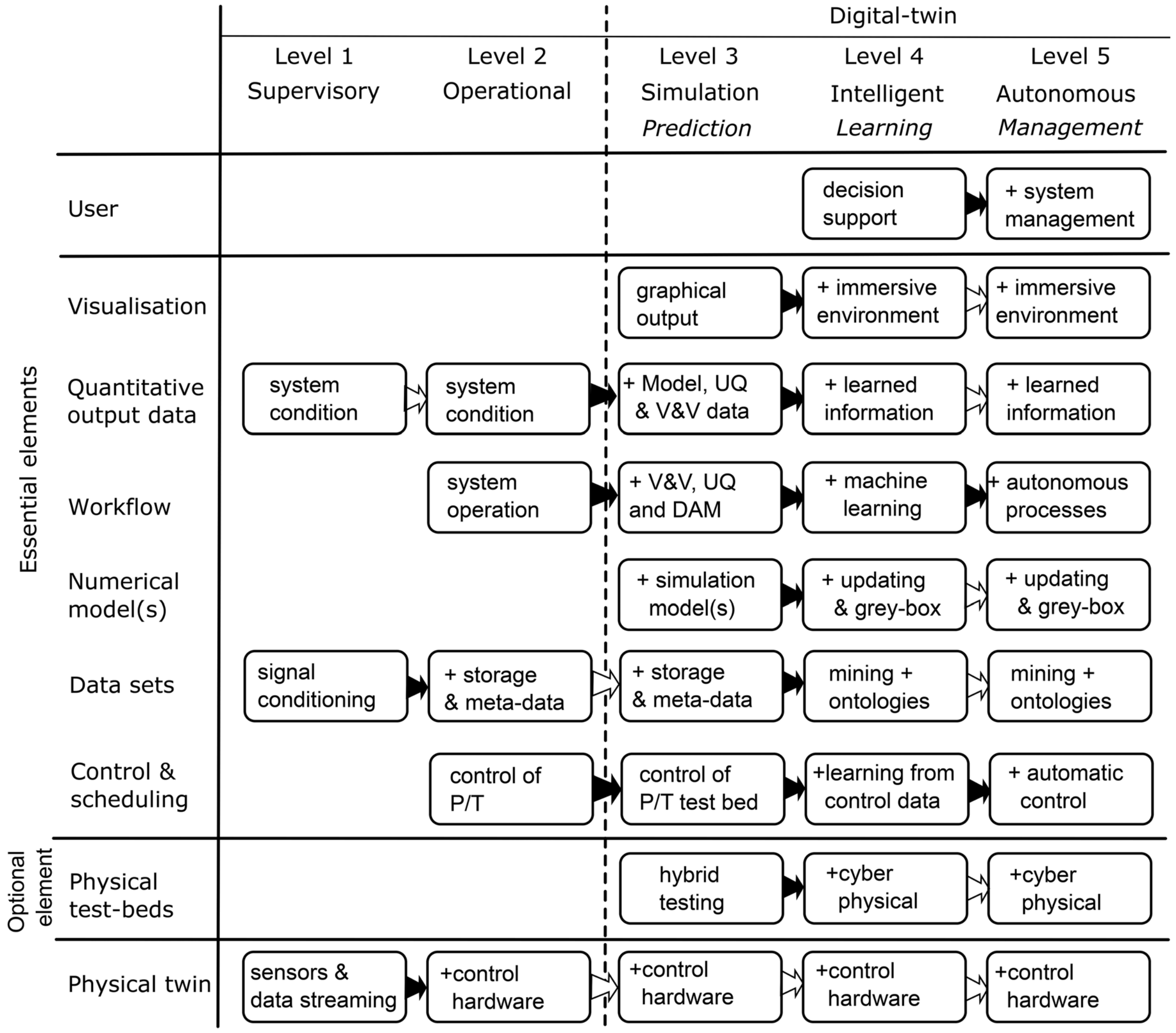

Fig. 4 Schematic representation of the building blocks required for the five levels of digital twin. Note that only a selection of the possible process building blocks is shown in the workflow. Moving from left to right, each block incorporates the functionality of the previous block. Solid black arrows indicate new functionality, and white arrows indicate no new functionality. P/T is physical twin, V&V is verification and validation.

are indicative rather than prescriptive. The exact requirements will depend on the precise context. Note also that only a selection of the possible process building blocks is shown in the workflow.

The capability levels in Fig. 4 increase from left to right. Furthermore, moving from left to right, each building block incorporates the functionality of the previous block. Solid black arrows indicate that a new functionality has been added, while white arrows indicate no new functionality. Again this arrangement should be regarded as indicative rather than prescriptive, as specific choices will be made by the digital twin designer. For example, moving from level 3 to level 4 in the numerical models element adds model updating and gray-box modeling capability in Fig. 4, indicating an increase in sophistication of the new level. It should be noted, however, that changes between levels will be dependent on the exact context of the digital twin.

An important distinction is made between level 3, and levels 4 and 5, with regard to the user. In level 3, it is assumed that the user is responsible for all the "cognitive" tasks, such as deciding which workflow processes to run, making decisions, and the overall management of the physical twin. At levels 4 and 5, some of these tasks are anticipated to be incorporated into the digital twin functionality. This matter is a key area of future development for digital twin technology. The discussion is now extended further by using the example of a simulation digital twin.

5 Example Elements of Simulation Digital Twin

As an example, a simulation twin for the asset management phase of a wind turbine is determined from the matrix in Fig. 4. A schematic representation of the simulation digital twin is shown in Fig. 5.

Here, datasets are recorded from the physical twin, and control and scheduling commands fed back as required (enabling supervision and operation). The recorded data (potentially in real-time and from similar or legacy sources) are used for tasks in combination with the numerical model(s) and physical test bed(s) (which can include further online devices, systems or databases) to give the required simulation capability. The interaction of these different elements is coordinated by a *workflow*, which also provides the user with visualization and quantitative feedback.

The concept of workflow is well established in the domain of software engineering [65–69] and business process management [70]. Several authors have considered the problem of verifying the soundness of workflows, for example, see Refs. [71–73].

In the context of this work, the role of the workflow is to deliver and coordinate all the *required processes* that the digital twin is expected to perform. There must also be a user interface enabling commands to be received by the digital twin and also to provide quantitative and visual feedback. Once the commands are received, the workflow will coordinate and sequence the required

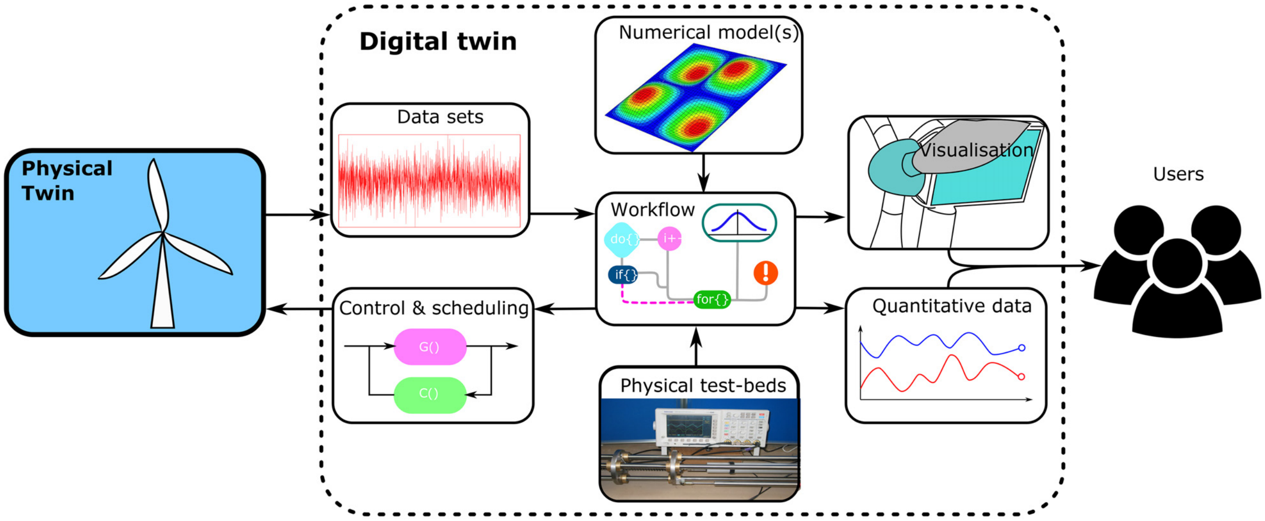

Fig. 5 Schematic representation of a simulation digital twin during an asset management phase, showing the essential elements for the simulation twin and their interrelations

processes, based on the aims and objectives of the digital twin. The required processes themselves can be built from relevant algorithms coordinated within the workflow (these algorithms can be aligned to the building blocks shown in Fig. 5).

The example considered here is of a simulation twin requiring UQ, and so it shall be assumed that the required algorithms are

(1) physics-based modeling;
(2) software integration and management;
(3) verification and validation;
(4) uncertainty quantification;
(5) data-augmented modeling;
(6) output visualization.

In addition to a workflow process related to each building block, it is possible that additional workflow processes can be created by combining and further augmenting these underlying building blocks. For the current example of a simulation twin, each of the separate building blocks listed above is now discussed briefly.

5.1 Physics-Based Modeling. Physics-based modeling is a well-established field within engineering. In essence, it is the process of using knowledge about physics, based on experimental observations, in order to construct mathematical representations of the system of interest. This takes many forms in engineering, from first-principles models, to approximation-based techniques such as finite element analysis (FEA), computational fluid dynamics, and multibody physics models. Of course, a key starting point in the development of many digital twins will be the generation of physics-based models (which will be formed from expert elicitation).

It is important to recognize that, despite the application of large amounts of computing power, the vast majority of engineering applications do not have a single ultrahigh fidelity model that captures all possible physics; this is because it is typically impossible to simultaneously replicate the behavior of all the physical processes happening, at all scales for anything except the most simple applications. As a result, engineers typically use multiple models, capturing different physical processes, at different length scales, and with a range of fidelities, for the same system. The essence of the digital twin concept is that these models can be augmented with available data, and beyond that with each other (a process that overlaps with existing techniques of model verification [74]). The primary purpose of this data and model augmentation is to increase the confidence in the prediction being made with the physics-based model.

Within the workflow, two other important processes are required of the physics-based models. The first deals with combining multiple models of the structure into the complete digital twin. Typically, companies will produce multiple models of the structure. This normally occurs due to the department divisions within an organization, due to expertise or the design process. These models will capture different physics, be modeled at different fidelities, and be at different scales, e.g., component, subassemblies. As a result, new opportunities for validating the complete digital twin occur, whereby these validated component or subsystem models are used to provide an understanding of the validity of the (full system) digital twin. By combining multiple models, the workflow provides significant gains on the current concepts of isolated validated models.

The second scenario occurs when specific models for particular tasks are needed at a particular level of efficiency. For example, a high-fidelity FEA model may be constructed, but may be too computationally expensive to run for an online fatigue estimation task for a hotspot of interest (although with the increase in computational power, this may become a less frequent problem). Instead, a bespoke, more efficient, model may be generated from the FEA model for this task; this could be a reduced-order model, as commonly utilized in dynamics applications, or could be an efficient surrogate or emulator of a complex computer model [75,76].

5.2 Software Integration and Management. The set of physics-based models utilized as part of a digital twin will need managing and integrating. The variety of solvers, software providers, and outputs will all require interactions with a main user interface (and potentially with each other) via workflows; the question is: "how might this be achieved?" One possible solution is that the digital twin workflow will coordinate, and call as required, other software packages or bespoke pieces of code to perform required subtasks—this is called *loose coupling* by Ref. [3], as opposed to using a single solver for all the physical processes, which is called *tight coupling*. In this sense, the workflow would operate (at least in part) as a "wrapper" with a user interface. Multiple existing subtasks can then be run in parallel, or cross-coupled to create new *super-tasks*, some of which may not have been previously achievable.

However, linking pieces of proprietary software together is fraught with its own set of difficulties. In addition to this, writing bespoke pieces of code for each application could be considered inefficient in the long term. Several authors have suggested using the concept of *blockchains* for digital twins, based on open source code [22,34]. It has been suggested that blockchains could be used to implement a range of different features, based on a clearly defined software architecture, for example, a "visual program" interface, that enables users to connect "programming blocks" together to obtain the required functionality. However, it seems that that the blockchain concept has evolved more toward secure transaction applications, which may not be so relevant in engineering, except where there is overlap with connected business processes. Whatever software architecture is used, the workflow will need to encode a series of logical steps in each process (for example, see Ref. [71]), in order to capture the sophisticated level of task coordination required.

Key to any implementation is effective representation of coupled physical processes, either through multiphysics modeling or coupling software/simulation codes to capture the required behavior; this will often be made more challenging due to large differences in temporal and spatial scales.

5.3 Verification and Validation. Verification is defined by Oberkampf and Roy as "the process of determining that the numerical algorithms are correctly implemented in the computer code and of identifying errors in the software" [74]. The subject is divided into subcategories of software quality assurance (SQA) and *algorithm verification*, where SQA relates to checking the interactions of code as part of a wider software, and algorithm verification is interested in the correctness of the implementation of particular mathematical formulae. These two categories must both be implemented and used for a digital twin to be realized, as in practice, fundamental verification will be expected as part of employing any commercial software. Here, the particular challenges in verifying the software integration and management strategies described in Sec. 5.2 (part of SQA) are discussed. Moreover, an outline of the verification (algorithm verification) of machine learning and black-box approaches that may be incorporated as part of data-augmented modeling is given.

A fundamental task of a digital twin is to perform predictions. To gain any confidence in these predictions, validation must be conducted. The process of validating a model requires: (i) quantitatively measuring the accuracy of the model output against experimental data, (ii) providing a measure of confidence in the predictions, both when interpolating or extrapolating, in the models intended context of use, and (iii) determining whether the accuracy of the model is appropriate for the intended use [77]. In the context of a digital twin this becomes the process of validating several models, with different outputs, where (as previously mentioned) the tyranny of scales applies. Consequently, validation must be considered at a system level in combination with the submodel level. Moreover, it is argued that a digital twin cannot be fully realized without incorporating the quantification and propagation of uncertainties. As a result, validation processes and metrics will need to accommodate these uncertainties.

In order to perform validation, datasets must be obtained. Obtaining datasets is a particular challenge for many full-system structures. It may be possible to obtain data for one time instance, but impossible to acquire data for all possible outcomes a user may wish to model or for multiple repeats. The validation process therefore needs to be conducted for parts of the digital twin where data are obtainable.

5.4 Uncertainty Quantification. The aspiration of a digital twin is: a close one-to-one mapping between a physical and virtual system, which is only achievable through acknowledging uncertainties involved in both physical observations and computer models. A classification of these uncertainties, outlined by Kennedy and O'Hagan [75], follows:

- *Parameter uncertainties*—computer models inevitably contain parameters, which may be measurable (in which case there is parametric variability) but in most cases are not fully known or accessible.
- *Model discrepancy*—following the famous quote by Box [78] that "All models are wrong but some are useful," it is understood that even when the parameters are deterministic and "truly" known (in an engineering context this will occur when the parameters have physical meaning), there will still be mismatches between the model output and the "true" physical process (without observational uncertainty).
- *Residual variability*—given the same set of inputs the process may produce different outputs, due to a chaotic (due to not knowing the inputs to the required accuracy) or stochastic nature. This is often a problem with the inputs not being sufficiently detailed.
- *Parametric variability*—the situation in which the model is utilized may vary because inputs cannot be fully controlled or specified. A model may however require a specification of a single deterministic value, which should be varied based on knowledge of the process.
- *Observational uncertainty*—measuring any real-world structure will result in a level of measurement error or noise.
- *Code uncertainty*—most computer models are sufficiently complex that the output from a model is unknown until it is evaluated. An approach commonly utilized within surrogate modeling is therefore to treat it as uncertain at locations where the computer model has not yet been evaluated.

The task of UQ in a general context is to provide a measure of these sources of uncertainty, often jointly, in order to reflect the overall level of uncertainty inherent in both the model predictions and the gathered data. It is common practice to subsequently propagate the identified uncertainties through the model in order to evaluate variability in the predicted quantities of interest. Comparison of these predictions with experimental data over some appropriately specified validation domain lies at the heart of model validation, discussed in Sec. 5.3. The core processes involved in uncertainty quantification are model selection and parameter estimation (in different contexts, referred to as system identification or model updating). The processes of quantifying uncertainty in parameters may be achieved via a variety of approaches. Linear and nonlinear regression are widely used in a frequentist context, but make an assumption that parameters are fixed but unknown and offer a limited characterization of parameter distributions. Bayesian methods [79] have proven hugely popular in recent years with application of Markov chain Monte Carlo (MCMC) methods (e.g., the Metropolis Hastings algorithm [80]) being key to their practical application. Such techniques offer the possibility of building a detailed description of the distributions of uncertain model parameters at the cost of being computationally demanding; computational cost concerns for challenging distributions are addressed to some extent through developments of the basic MCMC algorithm (e.g., transitional MCMC [81]). With regard to model selection, and the errors that will inevitably occur as a result of the computational model not being able to perfectly reflect the underlying physics of the modeled process, there are two principle schools of thought. The effect of model form error/discrepancy is typically handled through a choice to either subsume this error within the parameter estimates, potentially biasing them (something of particular concern in cases where the parameters have physical meaning); or to explicitly model the discrepancy as considered in Sec. 5.5. The tradeoff between these approaches is considered in more detail in Ref. [82].

In a digital twin context, the uncertainty quantification process may involve application of techniques from the general toolbox of methods for UQ to multiple contributing models. The process is complicated by the fact that system-level predictions may be the result of result of multiple, interacting submodels. If there is coupling between these models (for example, the bidirectional coupling typically required in multiphysics or multiscale model; or in multilevel models where parameters at one level form states at another level [83]), the complexity of the UQ task grows substantially. Further, decision-making on the basis of multiple, uncertain model outputs is a substantially more complex task than for a single model. Ensemble forecasting, where weightings are applied in a principled fashion to the predictions of multiple generating models, offers a potential direction of travel in this area [84].

Finally, a key distinguishing feature of a digital twin is their evolution over time. The implication here is that any uncertainty quantification technique may need to operate in, or close to, real-time—a major constraint on many current technologies. Achieving real-time (or near real-time) UQ for a digital twin may require the development of highly computationally efficient estimation techniques [85]; the adoption of fast-running statistical surrogates that approximate the response of the underlying computational models within the digital twin [86,87]; or periodic updating when differences between the physical and digital twin outputs are deemed to have occurred.

5.5 Data-Augmented Modeling. It is never possible to fully capture all possible physics affecting a structure within a computer model, regardless of the level of fidelity. Consequently, a digital twin cannot be formulated solely from the outputs of physics-based computer models if the aim is to achieve ultrarealistic predictions. As outlined in the uncertainty quantification section, this problem is captured by the model discrepancy term. Using the knowledge that computer modeling alone will provide inadequate solutions to generating a digital twin, models must be augmented using information available from data in order to improve predictive capabilities.

One approach to data-augmented modeling assumes that a computer model can be embodied as [75]

$$\mathbf{z}(\mathbf{x}) = \mathbf{y}(\mathbf{x}) + e = \eta(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}) + e \tag{1}$$

where $\mathbf{z}(\mathbf{x})$ are the observations of the system outputs $\mathbf{y}(\mathbf{x})$, which are subject to uncertainties represented by the error term e. The bias (or model discrepancy)-corrected computer model outputs $\mathbf{y}(\mathbf{x})$ are functions of the inputs $\mathbf{x}$. Equation (1) states that $\mathbf{y}(\mathbf{x})$ is equal to the sum of the computer model $\eta(\mathbf{x}, \boldsymbol{\theta})$ and the model discrepancy $\delta(\mathbf{x})$, where $\boldsymbol{\theta}$ are parameters of the computer model.

Equation (1) provides a framework for utilizing additive machine learning methods in order to infer the model discrepancy and noise process. Without acknowledgment that model discrepancy exists and parameters inferred during uncertainty quantification will be biased and/or overconfident, which will lead to inaccurate predictions [88]. More generally, gray-box modeling—the combination of a white box (a physics-based model) and a black box (from machine learning or a statistical process)—encompasses the range of approaches whereby machine learning methods are inserted into physical model structures such that unknown physics can be accounted for and inferred from data.

5.6 Output Visualisation. Digital twins will organize a vast amount of information, most of which will be processed through well-established visualization techniques. In addition, new methods of data visualization will become possible. Notably, augmented/virtual reality or augmented/virtual inspection, as proposed by Moreu et al. [89], is expected to become more prevalent. By having a one-to-one mapping in the virtual domain, inspection tools can be combined in real-time with the outputs of the digital twin, to guide and inform inspectors.

6 Case Study: Toward a Digital Twin of a Small Scale Three-Story Building

In order to illustrate the philosophy of moving from predigital-twins to a digital-twin, specifically one incorporating elements of levels 3 and 4 of a digital twin, a three-story structure is introduced as a case study. In this scenario, the experimental test structure is taken to be the physical twin with the asset management objective being to construct a digital twin that can predict and monitor the accelerations at each of the three stories.

In between the top two floors of the physical twin is a “bumper” mechanism—two aluminum blocks, where one is attached to the top floor and the other to the middle floor. When specific excitation and initial conditions are met, the two blocks come into contact, introducing a harsh nonlinearity. This nonlinearity provides a demonstration of when traditional approaches to generating a validated model may fail. As a result, technologies are presented that move toward the aspiration of a digital twin.

In this case study, a scenario is imagined in which the structure is designed to operate under a random excitation applied at the first floor at a consistent forcing level. In the design and construction phase of the physical twin, it is assumed that the “bumper” mechanism will not come into contact, and therefore the system is treated as linear in the initial modeling stage. This was decided as under initial testing there was no activation of the “bumper” mechanism. This reflects common decisions made within industry, where often due to modeling difficulties, computational capacity, and prior assumptions, a simplified (often linear) computer model is generated, as long as it provides adequate predictive performance. Once in the operational phase and under the same band-limited white noise forcing level, the “bumper” mechanism of the physical twin is shown to occasionally introduce the harsh nonlinearity. The case study therefore reflects common real-world scenarios whereby unforeseen behavior occurs from the physical twin, and the digital twin is expected to replicate or at least inform the operators of these events. This case study, therefore, presents some of the challenges and technologies required in creating a digital twin.

6.1 Experimental Setup and Data Gathering. The physical twin is illustrated in Fig. 6 and has three stories. Each floor is constructed from an aluminum block with a mass of 5.2 kg and dimensions 350 × 255 × 5 mm ($L \times w \times h$). The floors are joined by vertical columns, with each column having a mass of 55 g and dimensions 555 × 25 × 1.5 mm. The blocks used to connect the columns to the floors have a mass of 18 g and dimensions 25 × 25 × 13 mm. For each of these connections, four Viraj A2-70 grade bolts (Viraj, Andheri (East), Mumbai, India) were used with a mass of 10 g each.

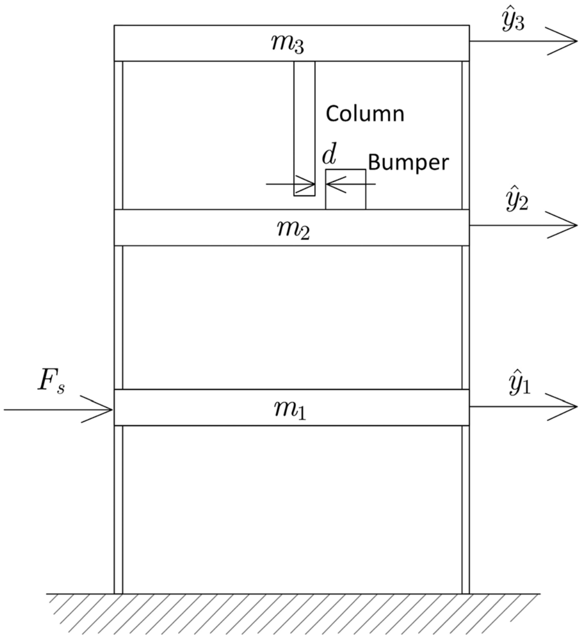

Fig. 6 The three story structure physical twin—a schematic diagram detailing the shaker attachment and accelerometer positioning

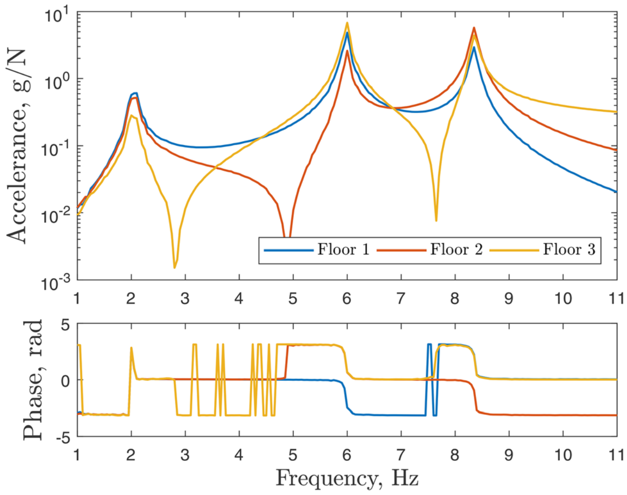

Fig. 7 Frequency response functions between the first floor and the accelerations from each of the three floors

The system is excited by a shaker attached at the first floor and a transducer is used to measure the force applied by the shaker. The experimental data were acquired using an LMS CADA system connected to a SCADAS-3 interface. Data were recorded at a sampling frequency of 51.2 Hz using piezoelectric accelerometers fixed to each story as shown in Fig. 6. The structure was consistently excited with a 25.6 Hz band-limited white noise source at the same excitation level.

Three datasets were collected. Each of the three datasets was 20 s observations of the structure under the random excitation source. In the first two datasets, used as a training and testing set in the following analyses, the "bumper" mechanism did not come into contact. The third dataset is a scenario in which there was contact in the bumper mechanism.

6.2 Initial Modeling: System Identification. Although the physical twin is ultimately a nonlinear system, the initial data from the physical twin in operation did not include any contact in the bumper mechanism. For this reason, the initial modeling stage assumed a linear model with which to perform system identification. Frequency response functions for the system, shown in Fig. 7, indicate that a three-degree-of-freedom model of the physical twin should be sufficient to capture the main dynamics of the structure. The proposed model is given by

$$\begin{aligned}\ddot{y}_1 &= (F_s - k_1 y_1 - k_2(y_1 - y_2) - c_1\dot{y}_1 - c_2(\dot{y}_1 - \dot{y}_2))/m_1\\ \ddot{y}_2 &= (k_2(y_1 - y_2) - k_3(y_2 - y_3) + c_2(\dot{y}_1 - \dot{y}_2) - c_3(\dot{y}_2 - \dot{y}_3))/m_2\\ \ddot{y}_3 &= (k_3(y_2 - y_3) + c_3(\dot{y}_2 - \dot{y}_3))/m_3 \end{aligned} \quad (2)$$

where $\{m_i\}_{i=1:3}$ are the masses, $\{c_i\}_{i=1:3}$ are the damping coefficients, and $\{k_i\}_{i=1:3}$ are the stiffness coefficients for each of the three floors (indexed by i). Additionally the force, displacement, velocity and acceleration terms are denoted as, F_s, $\{y_i\}_{i=1:3}$, $\{\dot{y}_i\}_{i=1:3}$, and $\{\ddot{y}_i\}_{i=1:3}$, respectively. The physics-based model selected here is analytical; however, the principles and techniques discussed are applicable to more complex model forms, such as finite element or multiphysics models.

Parameters for this model were identified using the Self-Adaptive Differential Evolution (SADE) algorithm. For full details of this algorithm, the reader is referred to the original paper [90]; for details of how it is implemented as an identification method, see Refs. [91] and [92]. Briefly, as in all evolutionary optimization procedures, a population of possible solutions (here, the vector of parameter estimates) is iterated in such a way that succeeding generations of the population contain better solutions to the problem in accordance with the Darwinian principle of survival of the fittest. The problem is framed here as a minimization problem with the cost function defined as a normalized mean-square error (NMSE) between the measured data and that predicted using a given parameter estimate

$$J_i(\boldsymbol{\theta}) = \frac{100}{N\sigma_{\ddot{y}_i}^{2}} \sum_{i=1}^{N} \left(\ddot{y}_i - \hat{\ddot{y}}_i(\boldsymbol{\theta})\right)^2 \quad (3)$$

where $\sigma_{\ddot{y}_i}^2$ is the variance of the measured sequence of relative accelerations and the caret denotes a predicted quantity; N is the number of "training" points used for identification, and θ is the parameter. The total cost function J was then taken as the average of the J_i. Previous experience has shown that a cost value of less than 5.0 represents a good set of model predictions (or parameter estimates). In order to generate the predictions $\hat{\ddot{y}}_i$, the coupled Eq. (2) were integrated forward in time in MATLAB using a fixed-step fourth-order Runge–Kutta scheme for initial value problems. The excitations for the predictions were established by using the measured forces. The SADE identification scheme is computationally expensive, with the main overhead associated with integrating trial equations forward in time. For this reason, the training set (or identification set) used here was composed of only $N = 400$ points. To avoid problems associated with transients, the cost function was only evaluated from the final 200 points of each predicted record. The first of the four datasets where the physical twin exhibited linear behavior is used as the training dataset.

The SADE algorithm was initialized with a population of randomly selected parameter vectors or individuals. The parameters were generated using uniform distributions on specified initial ranges. The initial ranges (estimated based on engineering judgment) were $[4.5, 7]$ for the masses, $[0, 6]$ for the damping and $[0, 2 \times 10^5]$ for the stiffness. A population of 200 individuals was chosen for the SADE runs with a maximum number of generations of 100. In order to sample different random initial conditions for the DE algorithm, ten independent runs were made. Each of the ten runs of the DE algorithm converged to a good solution to the problem in the sense that cost function values of around 2% or below were obtained in all cases; the summary results are given in Table 1. The best solution gave a cost function value of 1.620. A

Table 1 Parameter estimates from ten independent SADE runs

Parameter	Best	Maximum	Minimum	Mean	Standard deviation	Coefficient of variation
m_1	4.864	4.887	4.501	4.763	0.140	0.029
m_2	5.353	6.884	5.353	5.976	0.443	0.074
m_3	5.380	6.304	5.380	5.840	0.262	0.045
c_1	4.541	6.000	3.242	4.913	1.010	0.206
c_2	0.000	0.083	0.000	0.011	0.027	2.418
c_3	0.000	1.172	0.000	0.349	0.461	1.323
k_1 $(\times 10^3)$	4.100	4.425	2.887	4.012	0.505	0.126
k_2 $(\times 10^3)$	4.146	4.768	4.058	4.303	0.212	0.049
k_3 $(\times 10^3)$	4.906	6.134	4.906	5.467	0.347	0.064
J	1.620	2.457	1.620	2.042	0.307	0.150

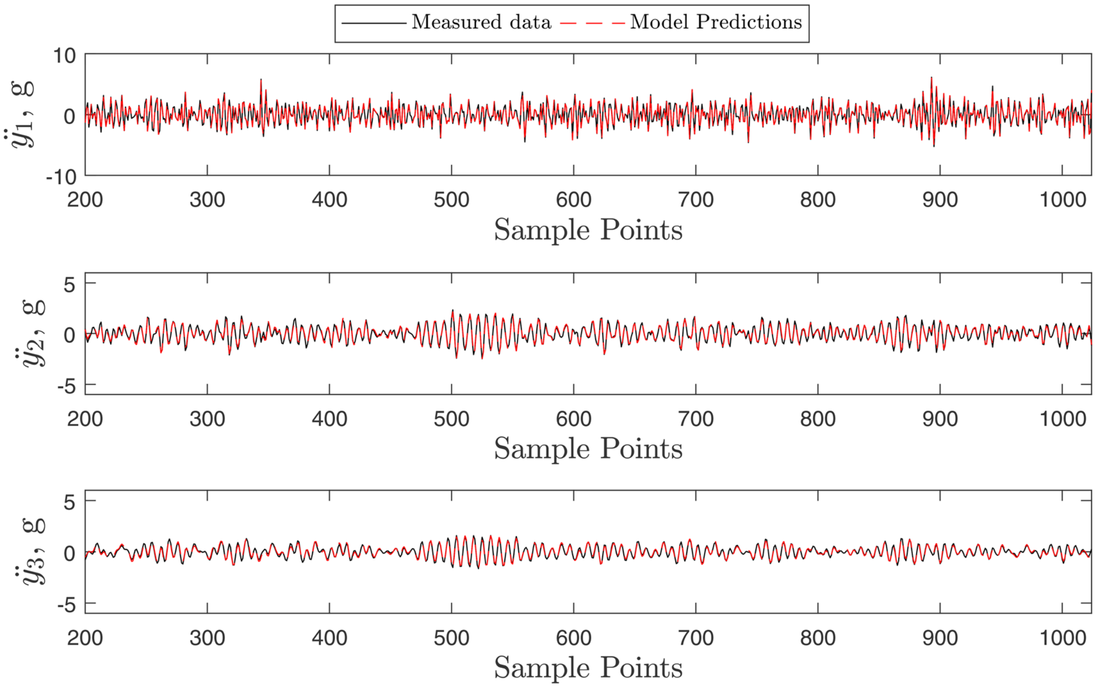

Fig. 8 SADE model predictions on testing data (no bumper contact)

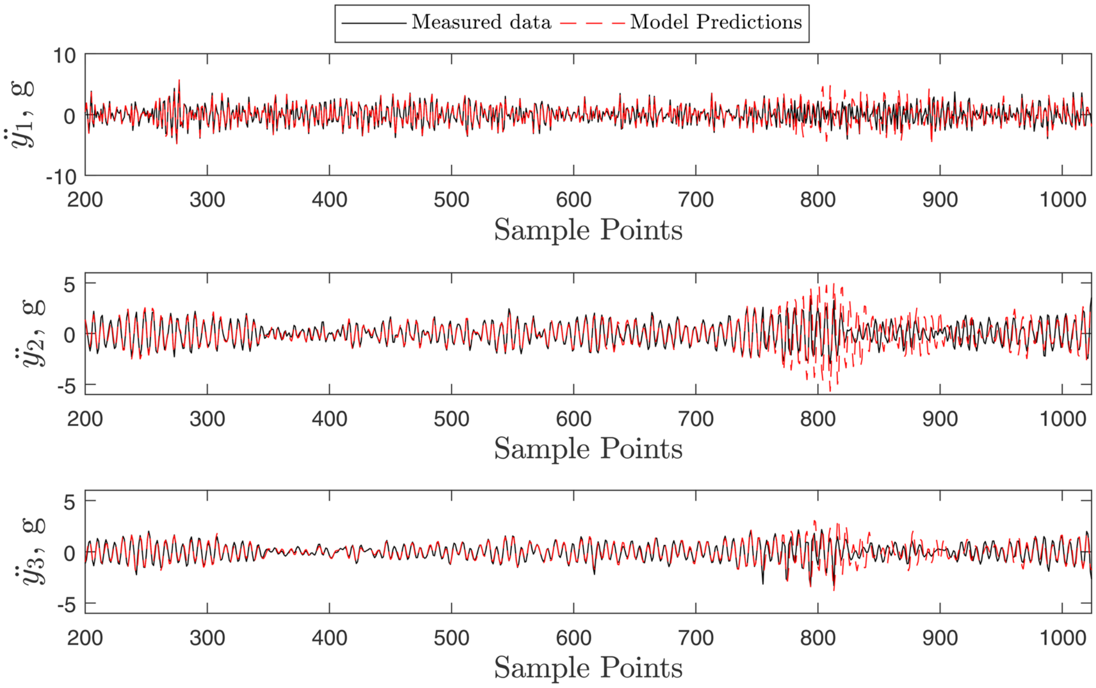

Fig. 9 SADE model predictions on validation data (bumper contact)

visual comparison of the "true" experimental responses for an unseen test dataset (the second dataset where the bumper mechanism did not make contact) and the predicted response given the best parameter set is given in Fig. 8.

These results show that the objective of the digital twin has been met. Based on the validation metric (NMSE), the digital twin is shown to have good performance on the test set. Traditionally, this digital twin would then be expected to operate for the duration of the structure's life. The process shown here is compared to industry norms, in which a model may be deterministically calibrated and validated and expected to predict the structure performance. However, the calibrated model is then applied to the third dataset, in which the bumper mechanism comes into contact, introducing a harsh nonlinearity. Predictions of the digital twin in this region fail as presented in Fig. 9. The NMSEs for these predictions are 20.644, 55.724, and 34.421 for the acceleration at each floor. This is compared to, 0.317, 1.640, and 1.928 on the training dataset and 0.417, 2.877, and 3.778 on the test dataset. This shows that the model has failed in its objective of predicting the accelerations at each floor in the new context.

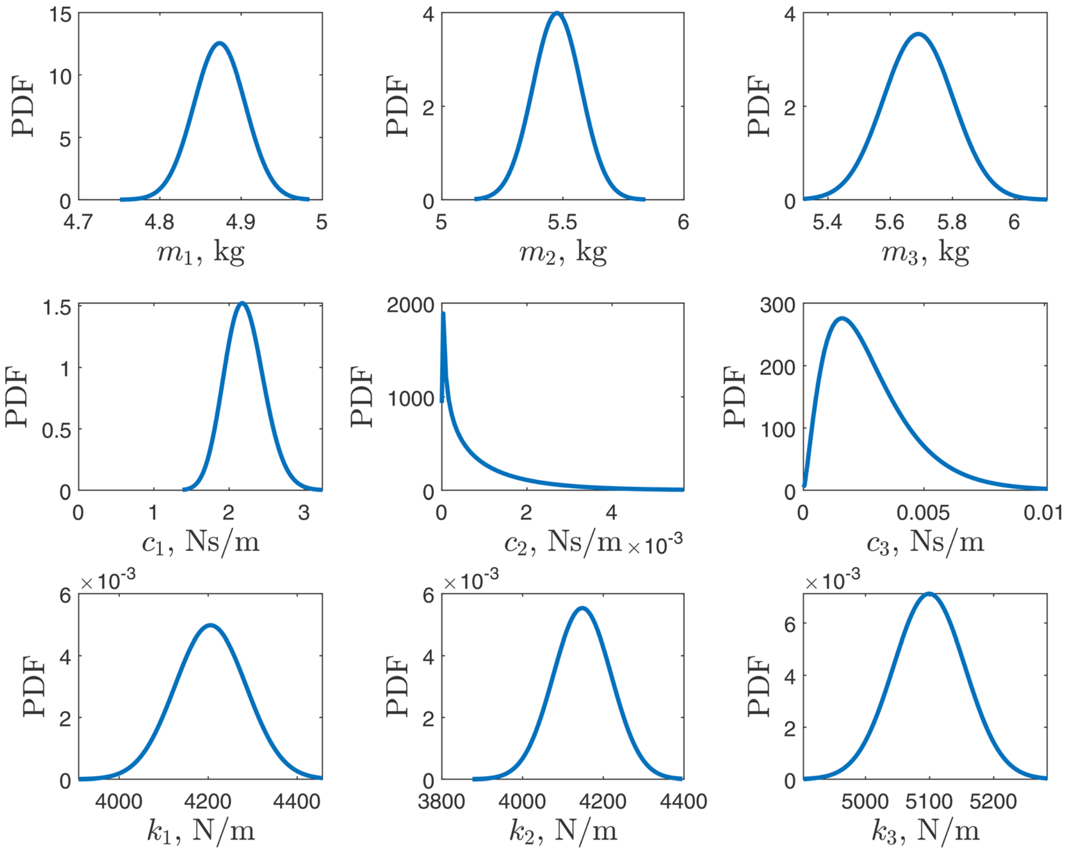

Fig. 10 MCMC parameter distributions

6.3 Uncertainty Quantification. Moving toward level three of a digital twin means incorporating knowledge about the uncertainty in the system. Given the same physical model and training, testing and validation datasets, Bayesian calibration (or system identification) was performed. By incorporating uncertainty estimation within the workflow allows the extraction of more information about the performance of the digital twin.

In this case study, MCMC—using the Metropolis Hastings algorithm—was used to perform Bayesian inference for the same linear analytical model (Eq. (2)). A joint Gaussian likelihood (the product of the Gaussian likelihood for each floor) was used, where the noise variance was fixed ($\sigma^2 = 3 \times 10^{-3}$) reflecting engineering judgment of the sensors. Gaussian priors were also formulated for the mass, stiffness, and damping coefficients where the mean for each prior was the best fit from the SADE analysis (shown in Table 1), with variances of $\sigma^2 = 10$ for the mass and damping coefficients and 1×10^8 for the stiffness coefficients. Four MCMC chains were run in parallel with random start locations and the $\hat{R}$ statistics measured to check convergence. As Bayesian parameter estimation is not the topic of this paper, the reader is referred to Refs. [80] and [82] for more details on MCMC for uncertainty quantification.

Four independent MCMC chains were run all initialized at different random initial conditions. 10,000 samples were obtained with a burn-in of period of 2500 samples. The $\hat{R}$ statistics were checked for all the parameters. It was found that although the chains had satisfactorily converged, the likelihood was relatively insensitive to the damping coefficients. Every twentieth sample was taken from the first chain; this is performed in order to protect against any residual autocorrelation in the chains. The estimate parameter distributions from the MCMC analysis are shown in Fig. 10. It can be seen that the values estimated by SADE (Table 1) are all within the estimated distributions apart from the first damping term c_1. This again shows the difficulty of estimating damping, due to the relative insensitivities in the acceleration in a lightly damped metallic structure—confirmed by the high coefficient of variation in the SADE estimates. This shows the information gained about the structure from uncertainty quantification.

The output predictions for these samples are shown on the testing (no bumper contact) and validation (bumper contact) sets in Figs. 11 and 12. These figures illustrate good predictive performance for the test dataset; however, as expected, they fail to predict the validation set. The histogram of the NMSEs for the outputs from the parameter samples is shown in Fig. 13, stating that the model performs well on the test data and fails on the validation data. The figure also shows that the SADE NMSE results are at the lower end of the histograms.

6.4 Data Augmented Modeling. An additional step in moving toward a digital twin is to augment the model with data. Here, a Gaussian process (GP) model is used to infer model discrepancies for the predicted output from the linear model. This is the equivalent of performing Eq. (1) in two stages, i.e., a parameter inference step to determine the parameters $\boldsymbol{\theta}$ of the computer model $\eta(\mathbf{x}, \boldsymbol{\theta})$ and then a discrepancy step to infer $\delta(\mathbf{x}) + e$.

The discrepancies are believed to contain dynamic information and for this reason, the inputs to the GP model are lagged outputs of the linear model and the input forcing (where the forcing is expected to be known at time t_n as it is measured), i.e., $\{\ldots, \ddot{y}_i(t_n - 3),\ \ddot{y}_i(t_n - 2),\ \ddot{y}_i(t_n - 1), \ldots, \ddot{F}(t_n - 3),\ \ddot{F}(t_n - 2), \ddot{F}(t_n - 1), \ddot{F}(t_n)\}$; where the outputs from the linear model are the averaged output prediction from the MCMC samples. This type of model is equivalent to $\delta(\mathbf{x}, \ddot{\mathbf{y}}) + e$. To determine the number of lags used within the data-augmented model, the autocorrelation of the residual between the linear model predictions and training observations were calculated. This informed that there was correlation up to around ten lags, leading to ten lags being used as inputs.

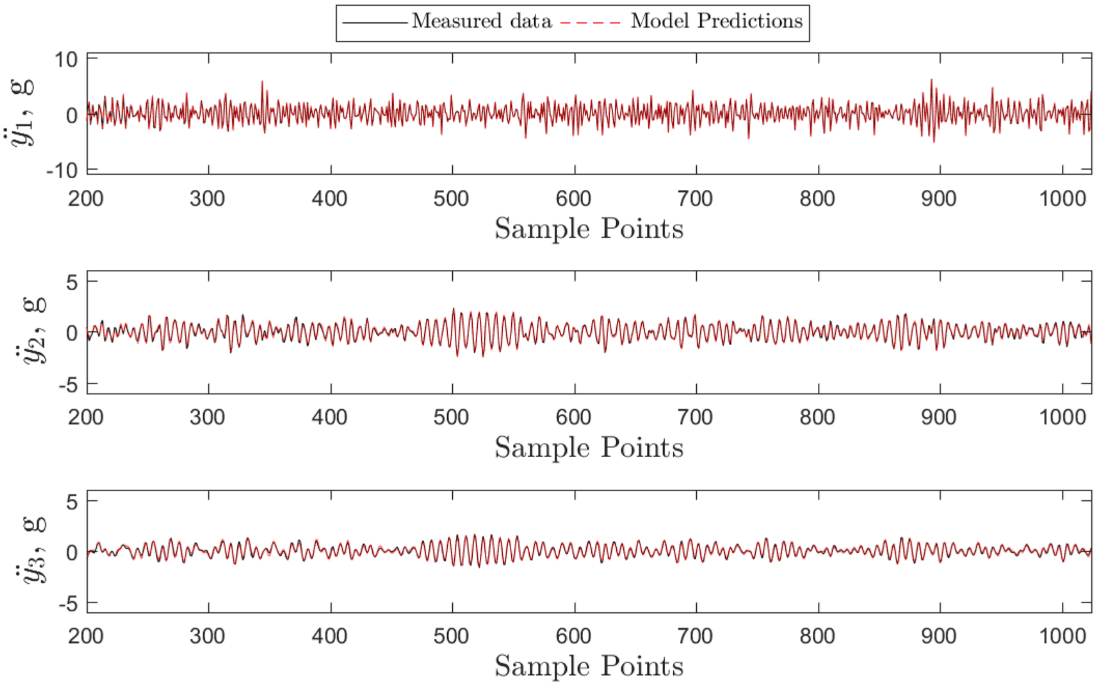

Fig. 11 MCMC model predictions on testing data (no bumper contact)

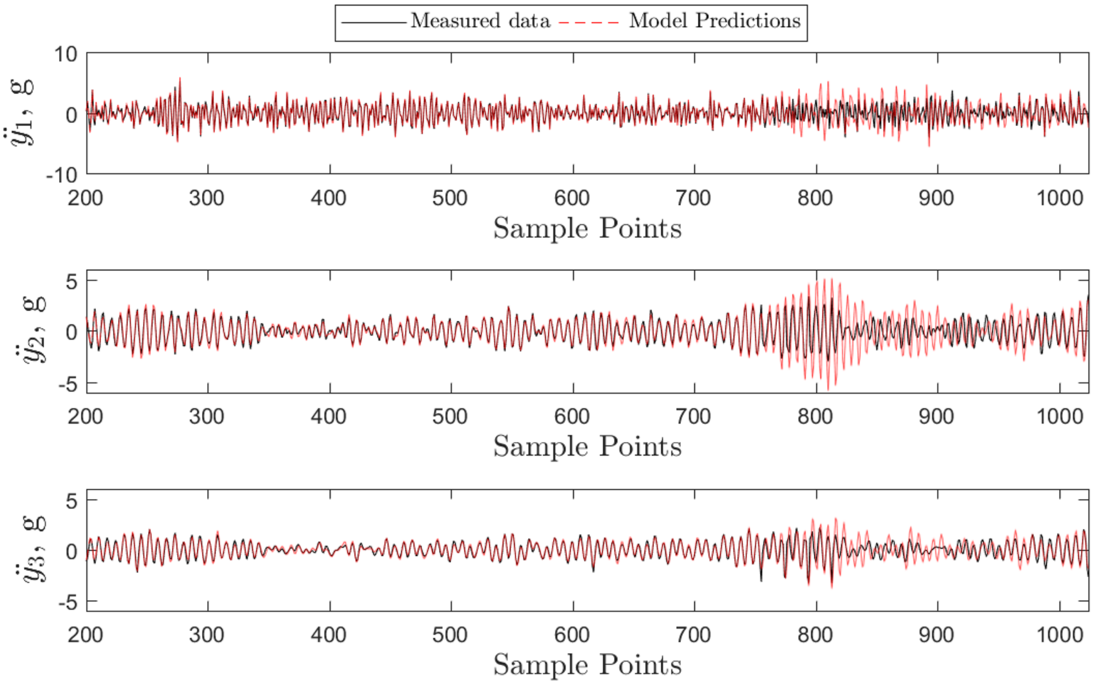

Fig. 12 MCMC model predictions on validation data (bumper contact)

Three GP models were generated (due to the single output nature of the GP). Each GP has a zero mean and Matérn 3/2 covariance function prior. The covariance function is formulated using the automatic relevance detection form, where a length scale is placed for each input, allowing lag selection to be performed within the covariance function. For more on Gaussian process regression the reader is referred to Ref. [93].

Each GP model was trained on sample points 200 to 400 of the training dataset such that the transients were removed and that the training dataset did not being prohibitively large. Once trained, the data-augmented model was used to predict on the test (where the bumper mechanism is not in contact) and validation data (where bumper mechanism is in contact) sets. The prediction for the test and validation cases is displayed in Figs. 14 and 15, respectively.

The NMSEs for the training, testing, and validation sets were: {0.901,0.386,0.163},{3.672,2.426,1.107}, {30.084,39.837, 21.054}. This demonstrates that the data augmented model improved predictions for floors two and three (over both the MCMC and SADE prediction). However, the NMSEs for floor one

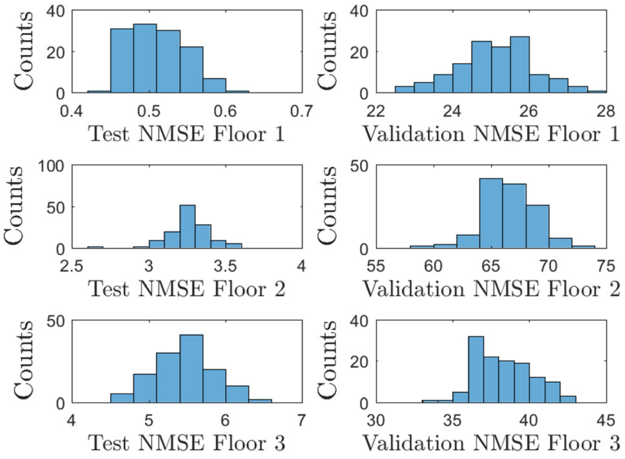

Fig. 13 NMSE for the MCMC acceleration predictions for each floor on the test (no bumper contact) and validation (bumper contact) datasets

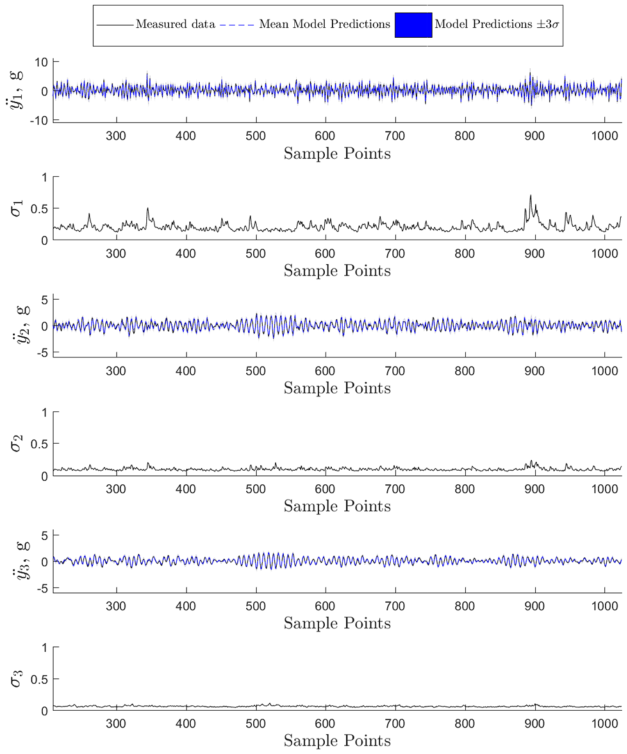

Fig. 14 Data augmented model predictions on testing data (no bumper contact) and predictive standard deviations

are larger than those from the previous analyses. This is likely due to the lack of dynamic information contained within the floor one location due to the positioning of the force.

Nonetheless, the data-augmented model provides additional benefits. The variance in the model reflects whether the GP has seen the combination of lagged inputs before. It would be expected that when the harsh nonlinearity was present, the variance (or standard deviation) of the model would increase, indicating that the model is predicting in an unseen region. This is confirmed by Figs. 14 and 15. In the test scenario, the standard deviations are small and relatively stationary for each floor. Yet, in the validation dataset, the standard deviation for the first and second floor predictions increases at the point in the data where contact occurred. This is a useful property for a digital twin as it informs about the presence of epistemic uncertainty.

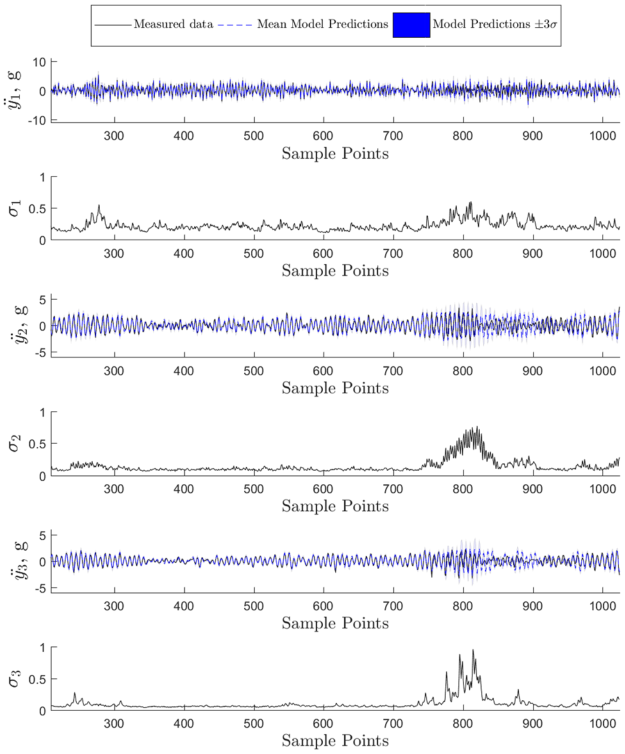

Fig. 15 Data augmented model predictions on validation data (bumper contact) and predictive standard deviations

The data augmented model can be used in an online manner to indicate when model improvements should be made. In regions of high variance, the workflow could choose to reperform the calibration step, or is could decide to improve the model form. For this example, this could lead to a bilinear stiffness model being introduced in order to capture the contact behavior. This would be a more optimal "white-box" model and would help improve predictions in the validation dataset. Unfortunately, the introduction of a nonlinear model would introduce new challenges in validation. For example, neither NMSE nor model properties would be good validation metrics as both would fail to inform whether the bifurcation point had been correctly inferred. More sophisticated would be required; otherwise, the model may perform extremely badly around the bifurcation point.

In addition, if the nonlinearity in the dataset were a breathing crack the data-augmented model would have a method in triggering a warning that the structure was damaged. By performing outlier analysis on the predictive standard deviation, for example, using a Mahalanobis distance, structural health monitoring decision can be made from the digital twin.

In conclusion, these case studies demonstrate that by moving up the levels of a digital twin more information and improved decision-making can be made. This will allow, not only better more realistic predictions, but improved decision capabilities as well.

7 Open Research Problems and Technological Challenges

7.1 Workflow, Coordination, and Time Evolution. At its core, a digital twin needs to be able to coordinate multiple tasks simultaneously. It must respond to requests from the user, in addition to continuously coordinating background tasks such as gathering and processing data from the physical twin, and updating models and databases. For many applications, all other tasks except for this central coordination and management of workflows will be existing technology.

From a research perspective, there has been much recent work on workflows and related areas such as business process models [65–67,70]. In the context of a digital twin, these are potentially most useful during the asset management phase. The types of open questions still to be answered include how workflows can be most efficiently implemented, ensuring that they are sound and robust [71,73,94]. For this purpose, formalizations of network theory appear to be the most relevant tool [95]. Furthermore, there is the question of how the workflow might *navigate* through the different elements of the digital twin, and for this purpose using the idea of a *knowledge graph* (possibly built from an initial ontology) appears to be one practical solution [96–98]. There is also the interesting question of how workflows can be adapted during the time evolution of the digital twin [72]. A key element of the digital twin functionality is decision support, and as well as other factors such as V&V; this also relates to the workflow processes [69,99,100].

Supporting all workflow processes in the digital twin will be a series of databases; these could be standalone, or online. How these databases interface with the digital twin (beyond just providing raw data) is an area of research interest. In particular, the use of knowledge graphs and ontologies [68], and ontology-driven databases appears promising. There is also the interesting question of how much the digital twin makes use of processes such as data mining [101] via tools such as the semantic web [102]; this also relates to the digital twin as part of the Internet of Things [34].

7.2 Joints and Joining. In Sec. 1, mechanical joints were highlighted as a technical example of a model ingredient for a digital twin. Alongside that example, the issue of silos existing in organizations based on natural subdivisions in a particular application was also discussed. Within the context of a digital twin there are some parallels between these two examples. The commonality comes from the fact that subdivisions of many engineering systems are very natural—after all, complex systems are typically made from multiple components and smaller subassemblages, which can naturally be modeled as simpler systems than the full system. However, the subassemblages can often be considerably complex in their own right, and so once subdivided, it is not surprising that more focus goes into modeling the subassemblage rather than how it interacts with or is joined to the rest of the system. Often, the associated models are incompatible in terms of jointing, and a bespoke interface model is needed to try and connect the software models.

For the mechanical joints problem, there is already considerable research work that has been carried out—for example, see Refs. [103–105] and references therein. Dealing with the multiscale and multiphysics nature of this problem is at the heart of the newly developed research. The possibility of making predictions based on only a partly assembled structure is an interesting area of future research, and relates to the verification and validation models discussed in Sec. 2.2.

In terms of the working in silos and interfacing software based models, both problems can be thought of as problems relating to *connectivity*. Many practitioners and researchers have already recognized these issues, and attempted to address them using more integrated procedures, as described, for example, in Ref. [48] as part of the product lifecycle management ethos. Ensuring that this factor is taken into account when developing a digital twin is largely a question of implementing current best practice [106,107], but there are always potential improvements that can be made, and this will form an ongoing research topic.

7.3 Uncertainty Management and the Quantification of Trust. It has been highlighted throughout this paper that an important issue is how to deal with uncertainty within the digital twin. In Sec. 6, a detailed example was presented that included a data-augmented modeling approach to managing the uncertainties. This is just one approach among many available, as briefly discussed in Sec. 6.3. However, it should be acknowledged that the example presented is relatively simple compared to most real engineering structures. For more complex structures, an ongoing area of research will be determining how exactly uncertainties are propagated through a digital twin in order to assess the level of confidence that can be given to the subsequent predictions.

In addition, enabling trust in digital twin predictions is essential to support engineering decision makers, for example, see Ref. [108]. To achieve this objective, the trust that can be ascribed to predictions from the digital twin must be quantified, and for this it is essential to integrate techniques from uncertainty quantification and propagation [109–112]. This quantification has to be an integral part of the digital twin (an early example is given by Ref. [42]). An area of future work to facilitate this will be to develop a risk based framework for the digital twin. Better assessment of potential risks, will help quantify trust, and support decisions.

8 Conclusions

In this paper, the application of the digital twin concept to engineering dynamics problems has been considered in detail, with a particular emphasis on modeling and simulation. A description of the current state of the art in this research area, including a detailed literature review was presented. This included the background and history of the digital twin, with particular emphasis on the topics of structural life prediction and verification and validation. Following this, a method for synthesizing a digital twin was presented, considering both design and asset management phases of the physical twin. Five levels of sophistication for a digital twin were defined, along with essential elements and required processes using the example of a simulation digital twin for a wind turbine. Methods for incorporating a digital twin into a product design phase were discussed in the context of verification and validation procedures that can be carried out in parallel with design and manufacture. To illustrate the detail of how several required processes could be implemented, an example case study of a three-story small-scale building was presented. This included a detailed description of data-augmented modeling to manage uncertainty present in the structure. Finally, three of the open research problems and technological challenges were outlined.

There are several key aspects that characterize the digital twins considered in this paper:

(1) A structured coordination of all the required processes via a bespoke workflow which provides both the interface with the user, and also the simultaneous integration of all other required processes (either bespoke or via software).
(2) Quantification, management, and ultimately reduction of model form (and other) uncertainties by use of measured data from the physical twin.
(3) Time evolution of the digital twin in order to reflect the aging of the physical twin, including the use of measured data to update and evolve the physics-based models in the digital twin.
(4) Robust methods for dealing with joints between parts of the physical twin.

In addition to this, methods from natural computing (such as machine learning) are already being used for the data-based techniques in this area, and the development of learning capabilities more generally is another area for future development. This paper has focused on the largely philosophical aspects of the topic. It is clear from the current interest in this topic that digital twin is set to have a disruptive influence on engineering applications. In a companion paper, as part of this special issue, a mathematical framework for digital twin applications is developed, and together the authors believe this represents a firm framework for developing digital twin applications in the area of engineering dynamics.

Acknowledgment

The experimental work was supported by M. Hall and J. Booth in the Jonas Laboratory at the University of Sheffield. In addition, the authors would like to acknowledge the contribution of Robin Langley, Ivan Au, Yichen Zhu, Lizzy Cross, and Nikos Dervilis for useful discussions and comments they gave before and after a previous version of this paper.

Funding Data

- EPSRC (Grant No. EP/R006768/1; Funder ID: 10.13039/501100000266).

References

[1] Gilchrist, A., 2016, *Industry 4.0: The Industrial Internet of Things*, Apress, New York.
[2] Chen, M., Mao, S., and Liu, Y., 2014, "Big Data: A Survey," Mobile Networks Appl., **19**(2), pp. 171–209.
[3] Tuegel, E. J., Ingraffea, A. R., Eason, T. G., and Spottswood, S. M., 2011, "Reengineering Aircraft Structural Life Prediction Using a Digital Twin," Int. J. Aerosp. Eng., **2011**, pp. 1–14.
[4] Langley, A., 1995, "Between 'Paralysis by Analysis' and 'Extinction by Instinct'," Sloan Manage. Rev., **36**, p. 63.
[5] Worden, K., Staszewski, W. J., and Hensman, J. J., 2011, "Natural Computing for Mechanical Systems Research: A Tutorial Overview," Mech. Syst. Signal Process., **25**(1), pp. 4–111.
[6] Rosen, R., von Wichert, G., Lo, G., and Bettenhausen, K. D., 2015, "About the Importance of Autonomy and Digital Twins for the Future of Manufacturing," IFAC-PapersOnLine, **48**(3), pp. 567–572.
[7] Grieves, M., and Vickers, J., 2017, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," *Transdisciplinary Perspectives on Complex Systems*, Springer, Berlin, pp. 85–113.
[8] Nicolai, T., Resatsch, F., and Michelis, D., 2005, "The Web of Augmented Physical Objects," International Conference on Mobile Business (ICMB 2005), Sydney, Australia, July 11–13, pp. 340–346.
[9] Morel, T., Keribar, R., and Leonard, A., 2003, "'Virtual Engine/Powertrain/Vehicle' Simulation Tool Solves Complex Interacting System Issues," SAE Paper No. 2003-01-0372.
[10] Lee, J., Ni, J., Djurdjanovic, D., Qiu, H., and Liao, H., 2006, "Intelligent Prognostics Tools and E-Maintenance," Comput. Ind., **57**(6), pp. 476–489.
[11] Ríos, J., Hernández, J. C., Oliva, M., and Mas, F., 2015, "Product Avatar as Digital Counterpart of a Physical Individual Product: Literature Review and Implications in an Aircraft," Proceedings of the 22nd ISPE Inc. International Conference on Concurrent Engineering, Delft, The Netherlands, July 20–23, pp. 657–666.
[12] Schleich, B., Anwer, N., Mathieu, L., and Wartzack, S., 2017, "Shaping the Digital Twin for Design and Production Engineering," CIRP Ann., **66**(1), pp. 141–144.
[13] Boschert, S., and Rosen, R., 2016, "Digital Twin—The Simulation Aspect," *Mechatronic Futures*, Springer, New York, pp. 59–74.
[14] Lee, J., Lapira, E., Bagheri, B., and Kao, H.-A., 2013, "Recent Advances and Trends in Predictive Manufacturing Systems in Big Data Environment," Manuf. Lett., **1**(1), pp. 38–41.
[15] Schroeder, G. N., Steinmetz, C., Pereira, C. E., and Espindola, D. B., 2016, "Digital Twin Data Modeling With automation ML and a Communication Methodology for Data Exchange," IFAC-PapersOnLine, **49**(30), pp. 12–17.
[16] Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., and Sui, F., 2018, "Digital Twin-Driven Product Design, Manufacturing and Service With Big Data," Int. J. Adv. Manuf. Technol., **94**(9–12), pp. 3563–3576.
[17] Tao, F., Sui, F., Liu, A., Qi, Q., Zhang, M., Song, B., Guo, Z., Lu, S. C.-Y., and Nee, A., 2019, "Digital Twin-Driven Product Design Framework," Int. J. Prod. Res., **57**(12), pp. 3935–3953.
[18] Banerjee, A., Dalal, R., and Joshi, K. P., 2017, "Generating Digital Twin Models Using Knowledge Graphs for Industrial Production Lines," Workshop on Industrial Knowledge Graphs, Co-located With the Ninth International ACM Web Science Conference, Troy, NY, June 25, pp. 1–6.
[19] Krueger, C., and Clements, P., 2017, "Enterprise Feature Ontology for Feature-Based Product Line Engineering and Operations," Proceedings of the 21st International Systems and Software Product Line Conference, Sevilla, Spain, Sept. 25–29, pp. 227–236.
[20] Ciavotta, M., Alge, M., Menato, S., Rovere, D., and Pedrazzoli, P., 2017, "A Microservice-Based Middleware for the Digital Factory," Procedia Manuf., **11**, pp. 931–938.
[21] Yun, S., Park, J.-H., and Kim, W.-T., 2017, "Data-Centric Middleware Based Digital Twin Platform for Dependable Cyber-Physical Systems," Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, July 4–7, pp. 922–926.
[22] Heber, D., and Groll, M., 2017, "Towards a Digital Twin: How the Blockchain Can Foster E/E-Traceability in Consideration of Model-Based Systems Engineering," Proceedings of the 21st International Conference on Engineering Design (ICED 17), Vancouver, Canada, Aug. 21–25, pp. 21–25.
[23] Uhlemann, T. H.-J., Schock, C., Lehmann, C., Freiberger, S., and Steinhilper, R., 2017, "The Digital Twin: Demonstrating the Potential of Real Time Data Acquisition in Production Systems," Procedia Manuf., **9**, pp. 113–120.
[24] Miller, A. M., Alvarez, R., and Hartman, N., 2018, "Towards an Extended Model-Based Definition for the Digital Twin," Comput.-Aided Des. Appl., **15**(6), pp. 880–891.
[25] Sierla, S., Kyrki, V., Aarnio, P., and Vyatkin, V., 2018, "Automatic Assembly Planning Based on Digital Product Descriptions," Comput. Ind., **97**, pp. 34–46.
[26] Knapp, G., Mukherjee, T., Zuback, J., Wei, H., Palmer, T., De, A., and DebRoy, T., 2017, "Building Blocks for a Digital Twin of Additive Manufacturing," Acta Mater., **135**, pp. 390–399.
[27] DebRoy, T., Zhang, W., Turner, J., and Babu, S., 2017, "Building Digital Twins of 3D Printing Machines," Scr. Mater., **135**, pp. 119–124.
[28] Cerrone, A., Hochhalter, J., Heber, G., and Ingraffea, A., 2014, "On the Effects of Modeling As-Manufactured Geometry: Toward Digital Twin," Int. J. Aerosp. Eng., **2014**, pp. 1–10.
[29] Lee, J., and Lapira, E., 2013, "Predictive Factories: The Next Transformation," Manuf. Leadership J., **20**(1), pp. 13–24.
[30] Stark, R., Kind, S., and Neumeyer, S., 2017, "Innovations in Digital Modelling for Next Generation Manufacturing System Design," CIRP Ann., **66**(1), pp. 169–172.
[31] Brenner, B., and Hummel, V., 2017, "Digital Twin as Enabler for an Innovative Digital Shopfloor Management System in the ESB Logistics Learning Factory at Reutlingen-University," Procedia Manuf., **9**, pp. 198–205.
[32] Söderberg, R., Wärmefjord, K., Carlson, J. S., and Lindkvist, L., 2017, "Toward a Digital Twin for Real-Time Geometry Assurance in Individualized Production," CIRP Ann., **66**(1), pp. 137–140.
[33] Uhlemann, T. H.-J., Lehmann, C., and Steinhilper, R., 2017, "The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0," Procedia CIRP, **61**, pp. 335–340.
[34] Datta, S. P. A., 2017, "Emergence of Digital Twins—is This the March of Reason?," J. Innovation Manage., **5**(3), pp. 14–33.
[35] Bottani, E., Cammardella, A., Murino, T., and Vespoli, S., 2017, "From the Cyber-Physical System to the Digital Twin: The Process Development for Behaviour Modelling of a Cyber Guided Vehicle in M2M Logic," XXII Summer School 'Francesco Turco'–Industrial Systems Engineering, Palermo, Italy, Sept., pp. 1–7.
[36] Alam, K. M., and El Saddik, A., 2017, "C2 PS: A Digital Twin Architecture Reference Model for the Cloud-Based Cyber-Physical Systems," IEEE Access, **5**, pp. 2050–2062.
[37] Qi, Q., and Tao, F., 2018, "Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison," IEEE Access, **6**, pp. 3585–3593.
[38] Boschert, S., Heinrich, C., and Rosen, R., 2018, "Next Generation Digital Twin," Proceedings of TMCE, Las Palmas de Gran Canaria, Spain, May 7–11, pp. 209–218.
[39] Nikolakis, N., Alexopoulos, K., Xanthakis, E., and Chryssolouris, G., 2019, "The Digital Twin Implementation for Linking the Virtual Representation of Human-Based Production Tasks to Their Physical Counterpart in the Factory-Floor," Int. J. Comput. Integr. Manuf., **32**(1), pp. 1–12.
[40] Tuegel, E., 2012, "The Airframe Digital Twin: Some Challenges to Realization," AIAA Paper No. 2012-1812.
[41] Seshadri, B. R., and Krishnamurthy, T., 2017, "Structural Health Management of Damaged Aircraft Structures Using Digital Twin Concept," AIAA Paper No. 2017-1675.
[42] Li, C., Mahadevan, S., Ling, Y., Choze, S., and Wang, L., 2017, "Dynamic Bayesian Network for Aircraft Wing Health Monitoring Digital Twin," AIAA J., **55**(3), pp. 930–941.
[43] Sharma, P., Hamedifar, H., Brown, A., and Green, R., 2017, "The Dawn of the New Age of the Industrial Internet and How It Can Radically Transform the Offshore Oil and Gas Industry," Offshore Technology Conference, Houston, TX, May 1–4, Paper No. OTC-27638-MS.
[44] Sivalingam, K., Sepulveda, M., Spring, M., and Davies, P., 2018, "A Review and Methodology Development for Remaining Useful Life Prediction of Offshore Fixed and Floating Wind Turbine Power Converter With Digital Twin Technology Perspective," Second International Conference on Green Energy and Applications (ICGEA), Singapore, Mar. 24–26, pp. 197–204.
[45] Tygesen, U. T., Jepsen, M. S., Vestermark, J., Dollerup, N., and Pedersen, A., 2018, "The True Digital Twin Concept for Fatigue Re-Assessment of Marine Structures," ASME Paper No. OMAE2018-77915.
[46] Glaessgen, E. H., and Stargel, D., 2012, "The Digital Twin Paradigm for Future NASA and US Air Force Vehicles," 53rd Structures, Structural Dynamics, and Materials Conference: Special Session on the Digital Twin, Honolulu, HI, Apr. 23–26, p. 1812.
[47] Iglesias, D., Bunting, P., Esquembri, S., Hollocombe, J., Silburn, S., Vitton-Mea, L., Balboa, I., Huber, A., Matthews, G. F., Riccardo, V., Rimini, F., and Valcarcel, D., 2017, "Digital Twin Applications for the Jet Divertor," Fusion Eng. Des., **125**, pp. 71–76.
[48] Grieves, M., 2005, *Product Lifecycle Management: Driving the Next Generation of Lean Thinking*, McGraw-Hill Professional, New York.
[49] Hankel, M., and Rexroth, B., 2015, "The Reference Architectural Model Industrie 4.0 (RAMI 4.0)," Zvei, Frankfurt, Germany.
[50] Batterman, R., 2013, "The Tyranny of Scales," *The Oxford Handbook of Philosophy of Physics*, R. Batterman, eds., Oxford University Press, New York, pp. 255–286.
[51] Ewins, D. J., 2000, *Modal Testing*, Research Studies Press, Philadelphia, PA.
[52] Au, S.-K., 2017, *Operational Modal Analysis: Modeling, Bayesian Inference, Uncertainty Laws*, Springer, New York, p. 1812.

[53] Worden, K., Wong, C. X., Parlitz, U., Hornstein, A., Engster, D., Tjahjowidodo, T., Al-Bender, F., Rizos, D. D., and Fassois, S. D., 2007, "Identification of Pre-Sliding and Sliding Friction Dynamics: Grey Box and Black Box Models," Mech. Syst. Signal Process., **21**(1), pp. 514–534.

[54] Worden, K., Barthorpe, R. J., Cross, E. J., Dervilis, N., Holmes, G. R., Manson, G., and Rogers, T. J., 2018, "On Evolutionary System Identification With Applications to Nonlinear Benchmarks," Mech. Syst. Signal Process., **112**, pp. 194–232.

[55] Friswell, M., and Mottershead, J. E., 1995, *Finite Element Model Updating in Structural Dynamics*, Vol. 38, Springer Science & Business Media, New York.

[56] Worden, K., and Green, P. L., 2014, "A Machine Learning Approach to Nonlinear Modal Analysis," *Dynamics of Civil Structures*, Vol. 4, Springer, New York, pp. 521–528.

[57] Verner, I., Cuperman, D., Fang, A., Reitman, M., Romm, T., and Balikin, G., 2018, "Robot Online Learning Through Digital Twin Experiments: A Weightlifting Project," *Online Engineering & Internet of Things*, Springer, New York, pp. 307–314.

[58] Tygesen, U., Worden, K., Rogers, T., Manson, G., and Cross, E., 2019, "State-of-the-Art and Future Directions for Predictive Modelling of Offshore Structure Dynamics Using Machine Learning," *Dynamics of Civil Structures*, Vol. 2, Springer, New York, pp. 223–233.

[59] AIAA, 1998, "AIAA Guide for the Verification and Validation of Computational Fluid Dynamics Simulations (G-077-1998)," American Institute of Aeronautics and Astronautics, Reston, VA, Report No. G-077-1998.

[60] Schwer, L., 2006, "ASME V&V 10 Guide for Verification and Validation in Computational Solid Mechanics," American Society of Mechanical Engineers, New York, Report No. PTC 60 / V&V 10.

[61] Schwer, L., 2009, "ASME V&V 20 Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer," American Society of Mechanical Engineers, New York, Report No. V&V 20–2009.

[62] Lau, K.-K., Taweel, F. M., and Tran, C. M., 2011, "The w Model for Component-Based Software Development," 37th EUROMICRO Conference on Software Engineering and Advanced Applications, Oulu, Finland, Aug. 30–Sept. 2, pp. 47–50.

[63] Khan, A. I., and Khan, U., 2012, "An Improved Model for Component Based Software Development," Software Eng., **2**(4), pp. 138–146.

[64] Tiwari, S., Bajpai, S., and Arshad, M., 2014, "Modified w Model for Handheld Application Development," Int. J. Eng. Tech. Res., **2**(12), pp. 2321–0869.

[65] Faustmann, G., 1996, "Workflow Management and Causality Trees," Proceedings of the Second International Conference on the Design of Cooperative Systems, Juan-les-Pins, France, June 12–14.

[66] van Der Aalst, W. M., Ter Hofstede, A. H., Kiepuszewski, B., and Barros, A. P., 2003, "Workflow Patterns," Distrib. Parallel Databases, **14**(1), pp. 5–51.

[67] Van Der Aalst, W., and Van Hee, K. M., 2004, *Workflow Management: Models, Methods, and Systems*, MIT Press, Cambridge, MA.

[68] Bowers, S., and Ludäscher, B., 2004, "An Ontology-Driven Framework for Data Transformation in Scientific Workflows," International Workshop on Data Integration in the Life Sciences, Leipzig, Germany, Mar., Springer, Berlin, pp. 1–16.

[69] Rozinat, A., Wynn, M. T., van der Aalst, W. M., ter Hofstede, A. H., and Fidge, C. J., 2009, "Workflow Simulation for Operational Decision Support," Data Knowl. Eng., **68**(9), pp. 834–850.

[70] Van Der Aalst, W. M., 2013, "Business Process Management: A Comprehensive Survey," ISRN Software Eng., **2013**, pp. 1–37.

[71] Bi, H. H., and Zhao, J. L., 2004, "Applying Propositional Logic to Workflow Verification," Inf. Technol. Manage., **5**(3/4), pp. 293–318.

[72] Rinderle, S., Reichert, M., and Dadam, P., 2004, "Correctness Criteria for Dynamic Changes in Workflow Systems—A Survey," Data Knowl. Eng., **50**(1), pp. 9–34.

[73] Van Der Aalst, W. M., van Hee, K. M., ter Hofstede, A. H., Sidorova, N., Verbeek, H., Voorhoeve, M., and Wynn, M. T., 2011, "Soundness of Workflow Nets: Classification, Decidability, and Analysis," Formal Aspects Comput., **23**(3), pp. 333–363.

[74] Oberkampf, W. L., and Roy, C. J., 2010, *Verification and Validation in Scientific Computing*, Cambridge University Press, Cambridge, UK.

[75] Kennedy, M. C., and O'Hagan, A., 2001, "Bayesian Calibration of Computer Models," J. R. Stat. Soc.: Ser. B, **63**(3), pp. 425–464.

[76] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., 1989, "Design and Analysis of Computer Experiments," Stat. Sci., **4**(4), pp. 409–423.

[77] Oberkampf, W. L., and Trucano, T. G., 2008, "Verification and Validation Benchmarks," Nucl. Eng. Des., **238**(3), pp. 716–743.

[78] Box, G. E. P., and Draper, N. R., 1987, *Empirical Model-Building and Response Surfaces*, Wiley, Hoboken, NJ.

[79] Beck, J. L., and Katafygiotis, L. S., 1998, "Updating Models and Their Uncertainties—I: Bayesian Statistical Framework," J. Eng. Mech., **124**(4), pp. 455–461.

[80] Hastings, W. K., 1970, "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," Biometrika, **57**(1), pp. 97–109.

[81] Ching, J., and Chen, Y. C., 2007, "Transitional Markov Chain Monte Carlo Method for Bayesian Model Updating, Model Class Selection, and Model Averaging," J. Eng. Mech., **133**(7), pp. 816–832.

[82] Smith, R. C., 2013, *Uncertainty Quantification: Theory, Implementation, and Applications*, Society for Industrial and Applied Mathematics, Philadelphia, PA.

[83] Sankararaman, S., and Mahadevan, S., 2015, "Integration of Model Verification, Validation, and Calibration for Uncertainty Quantification in Engineering Systems," Reliab. Eng. Syst. Saf., **138**, pp. 194–209.

[84] Parmigiani, G., and Inoue, L., 2009, *Decision Theory: Principles and Approaches*, Wiley, Hoboken, NJ.

[85] Chatzi, E. N., Smyth, A. W., and Masri, S. F., 2010, "Experimental Application of on-Line Parametric Identification for Nonlinear Hysteretic Systems With Model Uncertainty," J. Struct. Saf., **32**(5), pp. 326–337.

[86] Govers, Y., Haddad Khodaparast, H., Link, M., and Mottershead, J. E., 2015, "A Comparison of Two Stochastic Model Updating Methods Using the DLR AIRMOD Test Structure," Mech. Syst. Signal Process., **52–53**, pp. 105–114.

[87] Rocchetta, R., Broggi, M., Huchet, Q., and Patelli, E., 2018, "On-Line Bayesian Model Updating for Structural Health Monitoring," Mech. Syst. Signal Process., **103**, pp. 174–195.

[88] Brynjarsdóttir, J., and O'hagan, A., 2014, "Learning About Physical Parameters: The Importance of Model Discrepancy," Inverse Probl., **30**(11), p. 114007.

[89] Moreu, F., Bleck, B., Vemuganti, S., Rogers, D., and Mascarenas, D., 2017, "Augmented Reality Tools for Enhanced Structural Inspection," Structural Health Monitoring, Stanford, CA, Sept. 1–4.

[90] Storn, R., and Price, K., 1997, "Differential Evolution–A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces," J. Global Optim., **11**(4), pp. 341–359.

[91] Worden, K., and Manson, G., 2012, "On the Identification of Hysteretic Systems—Part I: Fitness Landscapes and Evolutionary Identification," Mech. Syst. Signal Process., **29**, pp. 201–212.

[92] Worden, K., and Becker, W., 2012, "On the Identification of Hysteretic Systems—Part II: Bayesian Sensitivity Analysis and Parameter Confidence," Mech. Syst. Signal Process., **29**, pp. 213–227.

[93] Rasmussen, C. E., and Williams, C. K. I., 2006, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA.

[94] Van der Aalst, W. M., 1998, "The Application of Petri Nets to Workflow Management," J. Circuits, Syst., Comput., **8**(1), pp. 21–66.

[95] Benveniste, A., Fabre, E., and Haar, S., 2003, "Markov Nets: Probabilistic Models for Distributed and Concurrent Systems," IEEE Trans. Autom. Control, **48**(11), pp. 1936–1950.

[96] Uschold, M., and Gruninger, M., 1996, "Ontologies: Principles, Methods and Applications," Knowl. Eng. Rev., **11**(2), pp. 93–136.

[97] Shames, P., and Skipper, J., 2006, "Toward a Framework for Modeling Space Systems Architectures," SpaceOps Conference, Rome, Italy, June 19–23, p. 5581.

[98] Paulheim, H., 2016, "Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods," Semantic Web, **8**(3), pp. 489–508.

[99] Yam, R., Tse, P., Li, L., and Tu, P., 2001, "Intelligent Predictive Decision Support System for Condition-Based Maintenance," Int. J. Adv. Manuf. Technol., **17**(5), pp. 383–391.

[100] Maggi, F. M., and Westergaard, M., 2017, "Designing Software for Operational Decision Support Through Coloured Petri Nets," Enterp. Inf. Syst., **11**(5), pp. 576–596.

[101] Ristoski, P., and Paulheim, H., 2016, "Semantic Web in Data Mining and Knowledge Discovery: A Comprehensive Survey," Web Semantics, **36**, pp. 1–22.

[102] Berners-Lee, T., Hendler, J., and Lassila, O., 2001, "The Semantic Web," Sci. Am., **284**(5), pp. 34–37.

[103] Witteveen, W., and Irschik, H., 2009, "Efficient Mode Based Computational Approach for Jointed Structures: Joint Interface Modes," AIAA J., **47**(1), pp. 252–263.

[104] Bograd, S., Reuss, P., Schmidt, A., Gaul, L., and Mayer, M., 2011, "Modeling the Dynamics of Mechanical Joints," Mech. Syst. Signal Process., **25**(8), pp. 2801–2826.

[105] Brake, M. R., 2017, *The Mechanics of Jointed Structures: Recent Research and Open Challenges for Developing Predictive Models for Structural Dynamics*, Springer, New York.

[106] Reynes, R., 1999, "Training to Manage Across Silos," Res.-Technol. Manage., **42**(5), pp. 20–24.

[107] Rilling, J., Witte, R., Schuegerl, P., and Charland, P., 2008, "Beyond Information Silos—An Omnipresent Approach to Software Evolution," Int. J. Semantic Comput., **2**(4), pp. 431–468.

[108] Atkinson, S. R., Caldwell, N. H., Maier, A. M., and Clarkson, P. J., 2011, "Collaborative Trust Networks in Engineering Design Adaptation," Proceedings of the 18th International Conference on Engineering Design (ICED 11), Vol. 7, Impacting Society Through Engineering Design, Lyngby/Copenhagen, Denmark, Aug. 15–19, pp. 152–161.

[109] Beck, A. T., and de Santana Gomes, W. J., 2012, "A Comparison of Deterministic, Reliability-Based and Risk-Based Structural Optimization Under Uncertainty," Probab. Eng. Mech., **28**, pp. 18–29.

[110] Jansen, M., Lombaert, G., and Schevenels, M., 2015, "Robust Topology Optimization of Structures With Imperfect Geometry Based on Geometric Nonlinear Analysis," Comput. Methods Appl. Mech. Eng., **285**, pp. 452–467.

[111] Aughenbaugh, J. M., and Paredis, C. J., 2006, "The Value of Using Imprecise Probabilities in Engineering Design," ASME J. Mech. Des., **128**(4), pp. 969–979.

[112] Zhang, S., Zhu, P., Chen, W., and Arendt, P., 2013, "Concurrent Treatment of Parametric Uncertainty and Metamodeling Uncertainty in Robust Design," Struct. Multidiscip. Optim., **47**(1), pp. 63–76.