A Comprehensive Review of Digital Twin - Part 2: Roles of Uncertainty Quantification and Optimization, a Battery Digital Twin, and Perspectives

Adam Thelen[1], Xiaoge Zhang[2], Olga Fink[3], Yan Lu[4], Sayan Ghosh[5], Byeng D. Youn[6], Michael D. Todd[7], Sankaran Mahadevan[8], Chao Hu[1*] and Zhen Hu[9*]

[1]Department of Mechanical Engineering, Iowa State University, Ames, 50011, IA, USA.
[2]Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong.
[3]Intelligent Maintenance and Operations Systems, EPFL, Lausanne, 12309, Switzerland.
[4]Information Modeling and Testing Group, National Institute of Standards and Technology, Gaithersburg, 20877, MD, USA.
[5]Probabilistic Design and Optimization group, GE Research, Niskayuna, 12309, NY, USA.
[6]Department of Mechanical Engineering, Seoul National University, Gwanak-gu, 151-742, Seoul, Republic of Korea.
[7]Department of Structural Engineering, University of California, San Diego, La Jolla, 92093, CA, USA.
[8]Department of Civil and Environmental Engineering, Vanderbilt University, Nashville, 37235, TN, USA.
[9]Department of Industrial and Manufacturing Systems Engineering, University of Michigan-Dearborn, Dearborn, 48128, MI, USA.

*Corresponding author(s). E-mail(s): chaohu@iastate.edu; zhennhu@umich.edu;
Contributing authors: acthelen@iastate.edu; xiaoge.zhang@polyu.edu.hk; olga.fink@epfl.ch; yan.lu@nist.gov; sayan.ghosh1@ge.com; bdyoun@snu.ac.kr; mdtodd@eng.ucsd.edu; sankaran.mahadevan@vanderbilt.edu;

Abstract

As an emerging technology in the era of Industry 4.0, digital twin is gaining unprecedented attention because of its promise to further optimize process design, quality control, health monitoring, decision and policy making, and more, by comprehensively modeling the physical world as a group of interconnected digital models. In a two-part series of papers, we examine the fundamental role of different modeling techniques, twinning enabling technologies, and uncertainty quantification and optimization methods commonly used in digital twins. This second paper presents a literature review of key enabling technologies of digital twins, with an emphasis on uncertainty quantification, optimization methods, open source datasets and tools, major findings, challenges, and future directions. Discussions focus on current methods of uncertainty quantification and optimization and how they are applied in different dimensions of a digital twin. Additionally, this paper presents a case study where a battery digital twin is constructed and tested to illustrate some of the modeling and twinning methods reviewed in this two-part review. Code and preprocessed data for generating all the results and figures presented in the case study are available on Github.

Keywords: Digital twin; Optimization; Machine learning; Enabling technology; Perspective; Industry 4.0, Review

1 Introduction

This paper is the second in a series of two that analyze the role of modeling and twinning enabling technologies, uncertainty quantification (UQ), and optimization in digital twins. Modeling and twinning enabling technologies are fundamental methods used to bridge the information gap between a physical system and its digital counterpart.

Part 1 of our two-part review provided an introduction to current state-of-the art methods used for digital twin modeling and proposed a five-dimensional digital twin model (Thelen et al., 2022) based on the flow of data through the model. Additionally, Part 1 reviewed modeling and twinning technologies commonly used to model a physical system as a digital counterpart (P2V) and to model the return of decisions/actions determined by the digital twin back to the physical system which will carry them out (V2P).

In this paper, we review and analyze many methods and modeling techniques currently used to quantify uncertainty and support probabilistic inference and estimation in the presence of uncertainty in a digital twin. In addition, we also examine the crucial role of optimization in bridging the gap between a physical system and its digital counterpart through informative data collection and modeling. As indicated in Fig. 1, UQ and optimization play vital roles in all three dimensions (i.e., modeling, P2V, and V2P) of digital twins discussed in Part 1 of the two-part review paper. For instance, quantifying uncertainty that arises from various sources in modeling a physical system is essential for building an accurate digital twin and making informed decisions under uncertainty. Another example lies in ensuring effective P2V connection for model updating, fault diagnostics, failure prognostics, and other reasoning tasks. It is very important to optimize how data is collected from a physical system to maximize the value of information in the collected data. Moreover, optimization is indispensable for most tasks in the V2P dimension of digital twins, such as system reconfiguration, process control, production planning, maintenance scheduling, and path planning. The three dimensions reviewed in Part 1 are the fundamental pillars of digital twins, while UQ and optimization are essential to ensure the seamless synthesis of the three dimensions to allow digital twins to effectively perform their intended functions, such as design optimization, quality control, and maintenance planning, in uncertain environments. This part of the review paper is dedicated to the roles of UQ and optimization in digital twins. We also explicitly demonstrate the benefits of predictive decision making augmented by optimization in several applications. To demonstrate many of the concepts discussed in both Part 1 and Part 2 of this review, we construct a battery digital twin and use this digital twin to optimize the retirement of a battery cell from its first life application, which vividly showcases the application of digital twin in the context of predictive maintenance scheduling (Sec. 2.2.3). Last, we close by discussing digital twin trends in industry, and present some open source software and datasets which may be of use to researchers and practitioners. Figure 2 gives an overview on the topics covered in this paper.

We begin by analyzing the integration of UQ and optimization for use in digital twins in Sec. 2. In what follows, Sec. 3 demonstrates some of the reviewed techniques with a case study of a battery digital twin. Sec. 4 reviews the applications of digital twins at industrial scale and available open-source tools and datasets related to digital twins. Next, we discuss challenges and future research directions in Sec. 5. Finally, concluding remarks are given in Sec. 6.

2 Roles of UQ and optimization in digital twins

In this section, we discuss the role of UQ in digital twins, cover UQ of ML models and UQ of dynamic system models. Following that, we review the role of optimization in digital twins, and discuss optimization methods for sensor placement, physical system modeling, and predictive decision making.

2.1 UQ for digital twins

First mentioned in the definition by Glaessgen and Stargel in their conference paper (Glaessgen and Stargel, 2012), digital twin is *"an integrated multiphysics, multiscale,* probabilistic *simulation of..."*. Probabilistic simulation plays an essential role in digital twins since variability is inherent and inevitable. A large and heterogeneous set of uncertainty sources is present in the five dimensions of the proposed digital twin model in Fig. 3

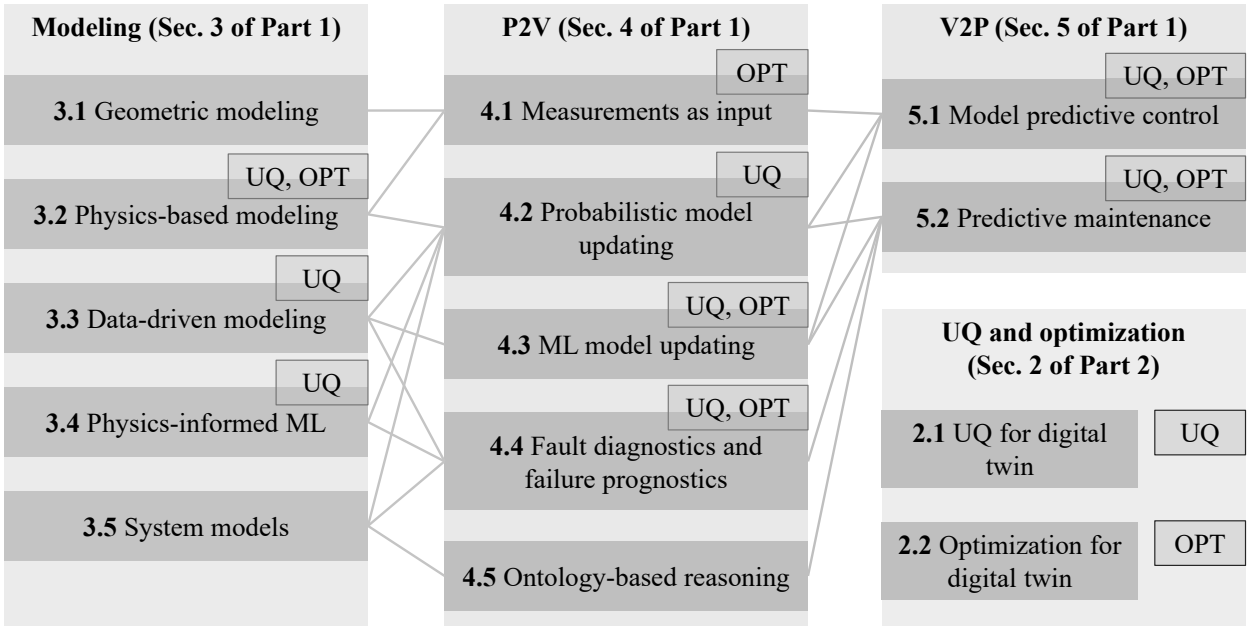


Fig. 1: Roles of UQ and Optimization in different dimensions of digital twins

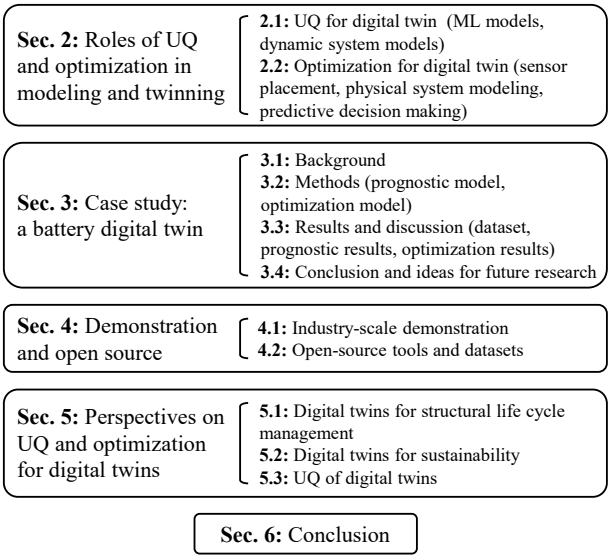


Fig. 2: Overview of topics covered in this paper

of Thelen et al. (2022). The uncertainty sources in a digital twin can be classified into two categories (Der Kiureghian and Ditlevsen, 2009):

- *Aleatory uncertainty* refers to uncertainty due to natural variability, which is inevitable and irreducible. Examples include sensor measurement errors and variability in material properties and load conditions. This intrinsic uncertainty can often be captured by fitting a probability distribution to a limited amount of data.
- *Epistemic uncertainty* refers to uncertainty caused by limited data, lack of knowledge, and/or model simplifications and assumptions. These sources of uncertainty are reducible when more information or data becomes available. For example, model uncertainty discussed in Sec. 2.1.2 is one of the most important epistemic uncertainty sources. Another example is ML models will have high predictive uncertainty if trained on small volumes of data. This model uncertainty can be reduced by either gathering more data (Sec. 4.2 on probabilistic model updating in Thelen et al. (2022) or incorporating known physics (see Sec. 3.4 on physics-informed ML in Thelen et al. (2022)).

A simple graphical comparison of aleatory uncertainty and epistemic uncertainty is given in Fig. 3, where a simple sine function is used to generate synthetic data with the addition of low (top figure) and high (bottom figure) white Gaussian noise. Hu and Mahadevan (2017) provides a detailed discussion on how to model various uncertainty sources in the context of additive manufacturing.

In this paper, we mainly focus on the quantification of epistemic uncertainty that is particularly relevant to digital twins. In addition, it is worth mentioning that there are many different ways of modeling uncertainty, such as probabilistic versus non-probabilistic and frequentist versus Bayesian statistics. For example, various non-probabilistic methods, including interval theory (Gao et al., 2010), fuzzy method (Bing et al., 2000), and evidence theory (Zhang et al., 2017; Soundappan et al., 2004), have been investigated in the past decades to model epistemic uncertainty in different engineering domains. According to the literature review in Part 1, probabilistic methods are more widely used than non-probabilistic methods in digital twins. Therefore, this paper mainly focuses on probabilistic methods for UQ in digital twins.

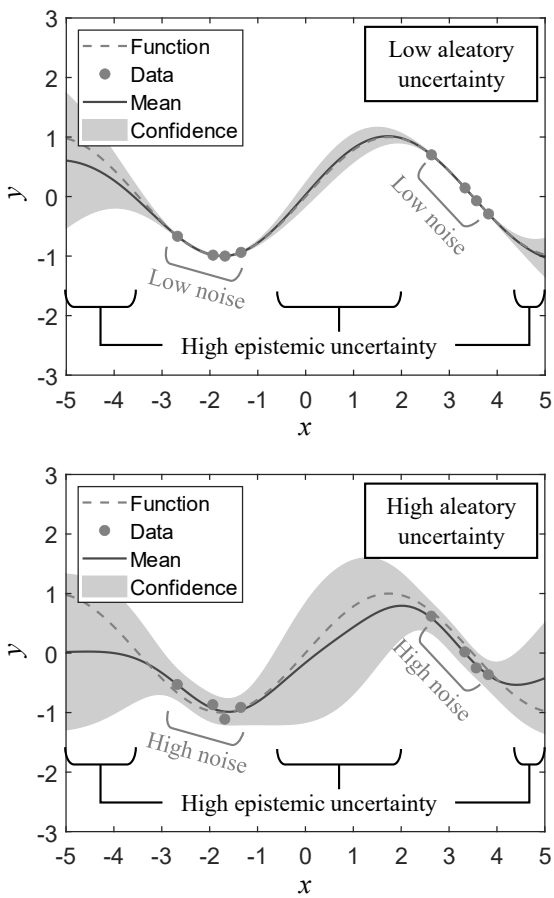


Fig. 3: A graphical comparison of aleatory uncertainty and epistemic uncertainty.

2.1.1 UQ of ML models

From the literature review, it is observed that ML models are extensively used in constructing digital models in the virtual space (see Sec. 3.3.2 of our Part 1 paper on ML models). As data-driven models, the performance of ML models is significantly affected by the quantity and quality of the data used for model training. As discussed in Sec. 4.3 of our Part 1 paper on ML model updating, ML models have difficulties generalizing to test data outside of a training distribution. When these trained ML models are deployed in digital twins, they may fail unexpectedly on out-of-distribution (OOD) samples. These unexpected failures reduce end users' trust and limit industry-scale, real-world adoptions of digital twin. This generalizability issue can be mitigated, to some degree, by incorporating physics (see Sec. 3.4 of our Part 1 paper on physics-informed ML) or by fine-tuning ML models based on newly labeled samples (see Sec. 4.3 of our Part 1 paper). However, predictions by physics-informed ML models and those with online updating still will not be perfect. It is highly desirable to quantify the predictive uncertainty of ML models, and in some safety-critical applications, such as autonomous driving and medical diagnostics, UQ of ML models becomes crucial. High quality estimation of an ML model's predictive uncertainty provides an accurate estimate of the model's confidence in a certain prediction, may allow for the detection of a data/concept shift, and most importantly, helps determine when the model is likely to fail. Over the past few years, UQ of ML models has become an established subdiscipline of ML, developed and promoted by the ML community. This subsection aims to provide an overview of this subdiscipline. More detailed and dedicated reviews can be found in two recent review papers (Abdar et al., 2021; Gawlikowski et al., 2021) and some benchmarking work has been presented in Nado et al. (2021).

UQ of ML models mainly deals with two tasks. First, it measures the predictive uncertainty for every training/test sample. For example, for the direct mapping strategy in Fig. 22 of our Part 1 paper, ML models capable of UQ can predict a probability distribution of RUL for every vector/matrix of input features rather than a point estimate. The so-called "calibration curve" or "reliability curve" can be plotted to visualize the

quality of uncertainty estimation by a probabilistic ML model (Niculescu-Mizil and Caruana, 2005; Kuleshov et al., 2018). In a calibration curve, the observed model confidence (y) is plotted against the expected model confidence (x) for equally spaced values between 0 and 1. An ML model with perfect uncertainty estimation should produce a reliability curve that follows the straight line $y = x$. For example, at a confidence level of 95%, we expect the observation (ground truth) to fall inside the 95% confidence interval produced by the perfect ML model 95% of the time. This way to evaluate the accuracy of UQ is interestingly similar to the U-pooling method used in the statistical validation of computer simulation models (Ferson et al., 2008; Liu et al., 2011) in that they both look at the area difference between an observed curve and an ideal straight line $y = x$. However, the U-pooling method is used to measure the disagreement between the probability distributions of a model prediction and an experimental observation, not the difference between the expected and actual model confidence.

Approaches for UQ of ML models mostly focus on estimating epistemic uncertainty (incomplete knowledge due to lack of data), as aleatory uncertainty (inherent noise in data) can be learned directly from data. Table 1 compares four popular approaches to quantify the uncertainty of ML models, and these four approaches are elaborated in what follows. We note that UQ of ML models is an active and quickly evolving field of research, and many new approaches (not discussed in this review) are emerging to estimate the predictive uncertainty of ML models.

- **Gaussian process regression** is probably one of the earliest probabilistic ML algorithms that can capture epistemic uncertainty (Williams and Rasmussen, 1995). The basic idea of Gaussian process regression is to assume the ys (i.e., output values of training data) at x coordinates follow a multivariate Gaussian and derive the conditional Gaussian of the y at a new x coordinate (test point), given the y values observed at some x coordinates (training points). Gaussian process regression has a rigorous mathematical formulation and deduction. It allows one to estimate the model predictive uncertainty in a closed-form expression. A limitation of Gaussian process regression is its difficulty in scaling to high-dimensional input spaces. Some dimensionality reduction or feature extraction will be needed for high-dimensional problems, but this intermediate step may degrade the prediction accuracy.
- **Bayesian neural networks** represent a principled way to measure the predictive uncertainty of a neural network. In this approach, a probabilistic neural network is built by first assuming the network weights and biases follow some prescribed probability distributions (often referred to as a prior) and then inferring the posterior based on the prior and some training data (Kendall and Gal, 2017). Bayesian estimation of neural network parameters is similar to the standard Bayesian inference approach (Category 1: Parameter calibration) described in Sec. 2.1.2 (a). As discussed in Sec. 3.3.2 of Part 1, when dealing with high input dimensions and given large volumes of training data, deep neural networks become an attractive alternative to traditional ML algorithms such as Gaussian process regression and random forest. However, training Bayesian deep neural networks involves approximate Bayesian inference such as Markov chain Monte Carlo (Andrieu et al., 2003) or variational inference on many network parameters. It requires significant changes to the standard model training procedure and is more computationally costly than training non-Bayesian deep neural networks (Papamarkou et al., 2021).
- **Ensembles of neural networks**, also called deep ensembles in the case of deep neural networks, are widely accepted as a powerful approach for UQ of ML models (Lakshminarayanan et al., 2017). An ensemble consists of independently trained neural networks with an identical architecture. For regression problems, a Gaussian layer is often added at the end of each network, allowing for predicting the mean and variance of a Gaussian output. Two central ideas of this ensemble-based approach are: (1) a measure of the difference between different predictors can be used as a proxy for epistemic uncertainty, and (2) the Gaussian layer of each network captures aleatory uncertainty. Deep ensembles are simple to train and test. Although more efficient to train and test than Bayesian neural networks, deep ensembles

still require high computational costs (multiple forward passes) and a large memory footprint (use of multiple neural networks). These issues impede their adoption in real-world digital twin applications where computational power and resources are limited.

- **More efficient approaches** than deep ensembles include Monte Carlo dropout (Gal and Ghahramani, 2016a) and approaches using deterministic neural networks (Van Amersfoort et al., 2020; Liu et al., 2020; Mukhoti et al., 2021). Monte Carlo dropout samples multiple sets of network weights to build multiple predictors from the same trained neural network and uses all predictors when predicting. It has been extensively studied on CNNs (Gal and Ghahramani, 2015) and recurrent neural networks (Gal and Ghahramani, 2016b) and is sometimes viewed as an efficient approximation (via variational inference) of a Bayesian neural network. Gal and Ghahramani (2016a) proved that Monte Carlo dropout minimizes the Kullback–Leibler divergence between the approximate posterior and true posterior of a Bayesian neural network. Monte Carlo dropout shares some similarities with deep ensembles, although deep ensembles use multiple trained networks at test time and showed better accuracy in uncertainty estimation in several studies (Lakshminarayanan et al., 2017; Nemani et al., 2021). Unlike deep ensembles and Monte Carlo dropout, which require multiple forward passes, deterministic neural network approaches require only a single forward pass to estimate uncertainty and have a shorter inference time (Van Amersfoort et al., 2020; Liu et al., 2020; Mukhoti et al., 2021). The basic idea of these deterministic approaches is to estimate the density of training points close to a test point in the embedded feature space learned by an ML model and use this density estimate as a proxy for epistemic uncertainty. The logic behind this idea is that adding new training points close to a test point in a high-level feature space is expected to reduce the epistemic uncertainty at the test point significantly. Aleatory uncertainty for in-distribution samples can be captured by including a softmax function in the final layer of a neural network classifier (Mukhoti et al., 2021) or adding a Gaussian output layer to a neural network regressor (Lakshminarayanan et al., 2017). This way, these deterministic network networks not only quantify the overall predictive uncertainty attributable to aleatory uncertainty and epistemic uncertainty, but they may also separate the contributions from the two types of uncertainty.

The second task is to identify test samples where a trained ML model has low confidence in predicting. These test samples often differ significantly from the samples the ML model is trained on and can be called OOD test samples, which we have discussed many times (thus, this task is sometimes referred to as OOD detection). These low confidence predictions are not trustworthy and should be examined by domain experts if a time delay from a prediction to a decision is acceptable. The need for extra caution is because low-confidence predictions are likely largely incorrect, and decisions made based on them without consideration of uncertainty will be flawed. For example, an ML model for fault diagnostics used in the ML pipeline for predictive maintenance shown in Fig. 26 of Part 1 may produce a false alarm at an OOD sample due to large measurement noise, warning that maintenance is needed on a pump that has only degraded slightly and has plenty of useful life left. This false-positive scenario may lead to an unnecessary maintenance action that can erode the trust of the end-users. A better alternative would be to associate this prediction with low confidence. Reliability and maintenance engineers can then investigate the model input features and determine if the model prediction makes physical sense before taking any maintenance actions.

Approaches to measuring model confidence include quantifying the degree of disagreement among an ensemble of ML models (Weigert et al., 2018) and measuring distances between a test sample and its nearest training neighbors in the embedded space learned by an ML model (Mandelbaum and Weinshall, 2017; Liu et al., 2020). The ensemble disagreement approach is inspired by and based on deep ensembles that have been discussed. It computes the average difference between the predictive distributions of the ML models in an ensemble and the predictive distribution of the ensemble. Although the

Table 1: A comparison of four different approaches for UQ of ML models

Quantity of interest	Gaussian process regression	Bayesian neural networks	Ensembles of neural networks	Monte Carlo dropout	Deterministic approaches
Accuracy in UQ	High	High-medium[a]	High	Medium	High-medium
Computational efficiency (test time)	High[b]	Low[c]	Medium-low	Medium-low	High
Ability to detect OOD samples	Strong	Weak (may estimate low uncertainty)	Weak (may estimate low uncertainty)	Weak (may estimate low uncertainty)	Strong
Scalability to high dimensions	Low	High	High	High	High

[a] Accuracy is largely affected by the quality of the assumed prior.
[b] Efficient only for problems of low dimensions (typically < 10).
[c] Could be efficient if variational Bayesian methods are employed to approximate the output posterior in in Bayesian neural networks.

Kullback–Leibler divergence was used as the distribution difference measure (Weigert et al., 2018), other measures of how one probability distribution is different from another can also be used. The distance-based approach calculates a confidence score for each predicted class at a test point based on the local density of training points with the same class in the embedded space (Mandelbaum and Weinshall, 2017). It was originally developed for classification problems but could be extended for regression problems. It is closely related to the deterministic approaches to estimating epistemic uncertainty, and the confidence score can be viewed as a side-product of epistemic uncertainty estimation.

2.1.2 UQ of dynamic system models

UQ of dynamic system models is a process of quantifying uncertainty in certain system outputs due to both aleatory and epistemic uncertainty sources (see the classification of uncertainty sources in Sec. 2.1). It could be forward uncertainty propagation or inverse UQ (Smith, 2013). The former focuses on propagating various uncertainty sources to the uncertainty of outputs. The latter emphasizes quantifying model uncertainty of computer simulation models based on observations. We focus on the latter in this section, and UQ herein means quantification of model uncertainty, an important type of uncertainty in digital twins that needs to be properly quantified and managed.

Model uncertainty arises from two main sources:

1. *Model parameter uncertainty*: This is the uncertainty in model parameters $\boldsymbol{\theta}$ due to lack of knowledge. It is worth mentioning that model parameter uncertainty could be either epistemic uncertainty due to lack of knowledge/data, or aleatory uncertainty due to natural variability (i.e., specimen to specimen variability), or both. In this section, we mainly focus on epistemic uncertainty.
2. *Model form uncertainty*: It results from imperfect modeling due to model assumption, simplification, lack of good understanding of the physics, etc. It is also referred to as model structure errors, model discrepancy, model bias, and model form error in the literature (Jiang et al., 2020; Arendt et al., 2012; Kennedy and O'Hagan, 2001).

We first introduce three categories of methods for UQ of general system models, which we call generalized methods. Following that, we discuss methods for UQ of dynamic system models, focusing on UQ of measurement equations and UQ of state transition equations.

(a) Generalized methods

Due to the difficulty in completely separating uncertainty in model parameters from uncertainty in model form, quantification of model uncertainty remains a challenging issue in the modeling and simulation of various engineered systems (Arendt et al., 2012). To improve the prediction accuracy of computer codes/simulation models, various approaches have been developed in the past decades and they can be roughly classified into three categories:

- **Category 1: Parameter calibration** This group of methods captures model uncertainty using uncertain model parameters and a noise term (Astroza et al., 2019; Song et al., 2019). A generalized model is formulated as

$$\vartheta_e = g_m(\mathbf{z},\ \boldsymbol{\theta}) + \gamma, \tag{1}$$

  where $\vartheta_e \in \mathbb{R}$ is an observation, $\vartheta = g_m(\mathbf{z},\ \boldsymbol{\theta})$ is a computer simulation model with output ϑ and inputs $\mathbf{z}$ and $\boldsymbol{\theta}$, $\mathbf{z} \in \mathbb{R}^{n_z}$ is a vector of measurable/controllable input variables which may change with observations, $\boldsymbol{\theta} \in \mathbb{R}^{n_t}$ is a vector of uncertain model parameters, the true values of $\boldsymbol{\theta}$ are usually fixed but unknown to us, and γ stands for model noise which is typically modeled as a Gaussian random variable with either unknown mean and variance (Song et al., 2019) or zero mean and unknown variance (Astroza et al., 2019). The noise term γ accommodates observation noise and part of model form uncertainty that is not captured by $\boldsymbol{\theta}$. We note that the model output, observation, and noise could be vectors in dynamic system models (see Sec. 2.1.2(b)). Here, they are constrained to be scalars to simplify the explanation (Kennedy and O'Hagan, 2001).

  A benefit of formulating the quantification of model uncertainty problem in Eq. (1) is that it casts the problem as a standard Bayesian model updating problem, allowing Bayesian inference methods to be used directly for model updating. Attributing model form uncertainty to $\boldsymbol{\theta}$ and the noise term γ, however, makes it independent from the inputs $\mathbf{z}$. As a result, it may overestimate or underestimate model form uncertainty for some values of $\mathbf{z}$.
- **Category 2: Bias correction** A point estimate $\boldsymbol{\theta}^*$ of the uncertain model parameters $\boldsymbol{\theta}$ is first obtained using the maximum likelihood estimation method or another offline calibration method discussed in Sec. 2.2.2. Afterwards, this category of methods fixes the uncertain model parameters $\boldsymbol{\theta}$ at $\boldsymbol{\theta}^*$ and uses a model discrepancy function and a noise term to account for model uncertainty. The computer simulation model $\vartheta = g_m(\mathbf{z},\ \boldsymbol{\theta})$ after adding the model discrepancy/bias function is given by (Wang et al., 2009)

$$\vartheta_e = g_m(\mathbf{z},\ \boldsymbol{\theta}^*) + \delta(\mathbf{z}) + \gamma, \tag{2}$$

  where $\delta(\mathbf{z})$ is a model discrepancy function that corrects the original computer simulation model. In the equation above, $\delta(\mathbf{z})$ accommodates most of the model form uncertainty and γ represents the residual model form uncertainty and observation noise. Similar to methods in Category 1, γ is modeled as a Gaussian random variable with either unknown mean and variance or zero mean and unknown variance (Xiong et al., 2009). The rationale of the above formulation is that the additional model form uncertainty caused by the inaccurate estimation of $\boldsymbol{\theta}^*$ can be compensated by the model discrepancy function $\delta(\mathbf{z})$ and the noise term γ.

  A data-driven model is usually constructed for $\delta(\mathbf{z})$ using methods discussed in Sec. 3.3 of our Part 1 paper on data-driven modeling. The predictive capability of data-driven models enables this line of methods to improve the prediction accuracy of the model under previously unseen conditions, as long as it is within the prediction capability of the data-driven model. The challenge for this category of methods is how to build an accurate model of $\delta(\mathbf{z})$.
- **Category 3: The KOH framework** In order to simultaneously quantify various sources of model uncertainty, Kennedy and O'Hagan (2001) developed a model calibration framework using Bayesian method and Gaussian process regression models. It is now the most widely used and commonly referred to as the KOH framework in the literature. The KOH framework constructs a Gaussian process regression model for the computer simulation model and another Gaussian process regression model as the model bias term. The two Gaussian process regression models are related to each other and

other uncertainty sources as follows:

$$\vartheta_e = \rho \hat{g}_m(\mathbf{z},\ \boldsymbol{\theta};\ \boldsymbol{\eta}_g) + \hat{\delta}(\mathbf{z};\ \boldsymbol{\eta}_\delta) + \gamma, \quad (3)$$

where ρ is an unknown regression coefficient within the range of $[0, 1]$ and it accounts for model uncertainty using a multiplication in addition to an additive bias term $\hat{\delta}(\cdot)$ and a noise term γ, $\hat{\vartheta} = \hat{g}_m(\mathbf{z},\ \boldsymbol{\theta};\ \boldsymbol{\eta}_g)$ and $\hat{\delta}(\mathbf{z};\ \boldsymbol{\eta}_\delta)$ are, respectively, the Gaussian process regression models of the computer simulation model $\vartheta = g_m(\mathbf{z},\ \boldsymbol{\theta})$ and the model bias term, $\boldsymbol{\eta}_g$ and $\boldsymbol{\eta}_\delta$ are hyperparameters of the two Gaussian process regression models respectively. For the noise term γ, it is modeled as a Gaussian random variable with zero mean and unknown standard deviation σ_γ in the original KOH framework. In some variants of the KOH framework, however, γ is modeled as a Gaussian random variable with unknown mean and unknown standard deviation (Xiong et al., 2009). To estimate the unknowns (i.e., ρ, $\boldsymbol{\theta}$, $\boldsymbol{\eta}_g$, $\boldsymbol{\eta}_\delta$, and statistical parameters of γ), full or modular Bayesian approaches have been developed (Arendt et al., 2012; Kennedy and O'Hagan, 2001).

It has been shown in various applications that the KOH framework is more effective in general than the other two categories of approaches when quantifying model uncertainty and improving prediction accuracy of computer simulation models (Jiang et al., 2020; Arendt et al., 2012; Xiong et al., 2009). The implementation of the KOH framework, however, is much more complicated than its counterparts due to the higher number of unknowns to be estimated. Additionally, the accuracy of the KOH framework could be affected by the prior distributions of $\boldsymbol{\theta}$ as shown in Jiang et al. (2020); Arendt et al. (2012), since the KOH framework is fundamentally a Bayesian method, of which prior distribution is a vital part.

The above reviewed three categories of methods have been applied to various computer simulation models, including static, quasi-static, and dynamic models. Since digital models of a digital twin usually are dynamic, next, we summarize variants of the above three categories of methods for dynamic system models. When the digital models are formulated in a state-space form as given in Eq. (9) of our Part 1 paper for model updating in the P2V connection (see Sec. 4.2 of Part 1 on probabilistic model updating) and control in the V2P connection (see Sec. 5.1 of Part 1 on model predictive control), the above reviewed three categories of methods could be applied to either the state transition equation or the measurement equation. According to which equation in a state-space model that the quantification of model uncertainty method is applied to, we classify the existing methods into two groups, namely (1) UQ of measurement equation, and (2) UQ of state transition equation (governing equations).

(b) UQ of measurement equations

Let us now look at UQ of dynamic system models, in particular state-space models such as the ones in Eqs. (3), (8), and (9) of our Part 1 paper. A typical state transition equation, $\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1},\ \mathbf{u}_{k-1}) + \boldsymbol{\omega}_k$, is a vector of difference state transition functions plus a vector of noises. Given an initial estimate of $\mathbf{x}_0$, the outputs of the state transition equation at time step k (i.e., state variables $\mathbf{x}_k$) are essentially functions of exogenous inputs $\mathbf{u}_{0:k}$ or functions of $\mathbf{u}_{0:k}$ and uncertain model parameters $\boldsymbol{\theta}$ if $\boldsymbol{\theta}$ is also considered in state transition (Beck and Katafygiotis, 1998). Therefore, $\mathbf{x}_k$ can be represented as $\mathbf{x}_k = \mathbf{F}(\mathbf{u}_{0:k},\ \boldsymbol{\theta}) + \boldsymbol{\epsilon}_k$, where $\mathbf{F}(\cdot)$ are numerical solutions to the recursion using $\mathbf{f}(\cdot)$ and $\boldsymbol{\epsilon}_k$ are the residual model form errors caused by this representation. $\mathbf{F}(\cdot)$ do not have close form expressions for most problems and need to be solved numerically. Moreover, the measurement function $\mathbf{g}(\mathbf{x}_k)$ are quasi-static models with state variables $\mathbf{x}_k$ as the input. If we embed the state transition function $\mathbf{f}(\cdot)$ in Eq. (9) of our Part 1 paper or $\mathbf{F}(\mathbf{u}_{0:k},\ \boldsymbol{\theta})$ into the measurement function $\mathbf{g}(\cdot)$, the overall dynamic system model as a whole can be written as

$$\begin{aligned} \mathbf{y}_k &= \mathbf{g}(\mathbf{F}(\mathbf{u}_{0:k},\ \boldsymbol{\theta})) + \boldsymbol{\gamma}_k, \\ &= \mathbf{h}(\mathbf{u}_{0:k},\ \boldsymbol{\theta}) + \boldsymbol{\gamma}_k, \end{aligned} \quad (4)$$

in which $\mathbf{h}(\cdot)$ is a new function created by implicitly embedding $\mathbf{f}(\cdot)$ into $\mathbf{g}(\cdot)$, and $\boldsymbol{\gamma}_k$ is a vector of Gaussian noise variables with either an unknown mean vector and covariance matrix or zero mean and an unknown covariance matrix. The initial conditions are omitted from the above equation to simplify the notations. Note that $\boldsymbol{\gamma}_k$ is used here

to account for observation noises and all residual model uncertainty that is not accounted for by the uncertain model parameters $\boldsymbol{\theta}$.

Based on the representation given in Eq.(4) and to facilitate the quantification of model uncertainty, the state-space model given in Eq. (9) of our Part 1 paper is re-formulated similar to Eq. (8) of Part 1 as (Astroza et al., 2019; Song et al., 2019; Burns et al., 2018)

$$\begin{aligned} &\text{State transition}: \boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \mathbf{r}_k, \\ &\text{Measurement}: \ \mathbf{y}_k = \mathbf{h}(\mathbf{u}_{0:k},\ \boldsymbol{\theta}_k) + \boldsymbol{\gamma}_k, \end{aligned} \tag{5}$$

where $\mathbf{y}_k \in \mathbb{R}^{n_y \times 1}$ is a vector of observations at t_k, $\mathbf{h}(\cdot)$ is the new measurement function as mentioned above. When the uncertain model parameters $\boldsymbol{\theta}$ do not change with time, Eq. (5) will reduce to $\mathbf{y}_k = \mathbf{h}(\mathbf{u}_{0:k},\ \boldsymbol{\theta}) + \boldsymbol{\gamma}_k$.

One may notice that the measurement equation in Eq. (5) or the reduced form of Eq. (5) (i.e., $\mathbf{y}_k = \mathbf{h}(\mathbf{u}_{0:k},\ \boldsymbol{\theta}) + \boldsymbol{\gamma}_k$) is very similar to the equations given in the aforementioned three categories of UQ methods (i.e., Eqs. (1)-(3) in Sec. 2.1.2 (a)). This similarity is beneficial as it allows us to quantify the uncertainty of dynamic system models by directly using methods originally developed for static models.

- For instance, based on the formulation in Eq. (5), Astroza et al. (2019); Song et al. (2019); Behmanesh et al. (2017) suggested several approaches for the quantification of model uncertainty of structural dynamic system models. Since $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}_k$ are used to account for all possible model uncertainty in their methods, those approaches can be classified as the *category 1* method (see Sec. 2.1.2 (a)). Moreover, in dynamic system models, the uncertain model parameters $\boldsymbol{\theta}$ change very slowly or do not change with time, while statistical parameters of $\boldsymbol{\gamma}_k$ change relatively faster due to the variability of model form uncertainty over time along with the exogenous inputs. The quantification of model uncertainty using the *category 1* methods based on the formulation given in Eq. (5) is, therefore, very similar to the state and parameter estimation discussed in Sec. 4.2.4 of our Part 1 paper. The difference in the time scales of $\boldsymbol{\theta}$ and statistical parameters of $\boldsymbol{\gamma}_k$ needs to be considered in the UQ process. To this end, Astroza et al. (2019) applied dual Kalman filter to simultaneously estimate $\boldsymbol{\theta}$ and the diagonal elements of the covariance matrix of $\boldsymbol{\gamma}_k$ over time. Song et al. (2019); Behmanesh et al. (2017) employed hierarchical Bayesian updating methods to estimate $\boldsymbol{\theta}$ and parameters of $\boldsymbol{\gamma}_k$. Since the *category 1* methods convert the quantification of model uncertainty problem into a standard Bayesian updating problem, the formulation given in Eq. (5) makes it possible to perform online model-uncertainty quantification using various Bayesian inference methods reviewed in Sec. 4.2 of Part 1.
- The *category 3* methods (i.e., the KOH framework reviewed in Sec. 2.1.2 (a) and its variants) have also been applied to quantify model uncertainty of the measurement equation based on the formulation given in Eq. (5). For example, by following the KOH framework, Burns et al. (2018); Ramancha et al. (2022); Burns et al. (2014); Ward et al. (2021) added a model discrepancy term to the measurement equation in addition to $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}_k$ to quantify model uncertainty. To address the computational challenge introduced by the KOH framework, Burns et al. (2018, 2014) used a parametric function as model discrepancy term such that the problem becomes to be a standard Bayesian updating problem which is similar to the category 1 methods. Ramancha et al. (2022) assumed the dynamic system model to be linear, which, as a result, reduced the computational burden required in applying the KOH framework. Ward et al. (2021) compared a variant of the KOH framework based on particle filter against a sequential KOH approach in the context of digital twins. They concluded that the computational effort required by the sequential KOH framework to track time-varying model parameters is high, which makes it not suitable for online updating in digital twin applications. The particle filter-based variant is computationally cheaper than the sequential KOH framework for digital twins (Ward et al., 2021).

In summary, the benefit of formulating the state-space model as Eq. (5) is that it allows us to directly apply the three categories of UQ methods reviewed in Sec. 2.1.2 (a) to quantify model uncertainty of the measurement equation (Ward et al., 2021; Astroza et al., 2019; Song et al., 2019; Behmanesh et al., 2017; Burns et al., 2018, 2014;

Ramancha et al., 2022). The disadvantage is that the non-linearity of the new measurement function $\mathbf{h}(\cdot)$ could be much higher than that of $\mathbf{g}(\cdot)$ or $\mathbf{f}(\cdot)$. As a result, the model discrepancy of the measurement function ($\mathbf{h}(\cdot)$) given in Eq. (5) would be more difficult to be quantified than that of $\mathbf{g}(\cdot)$ or $\mathbf{f}(\cdot)$ in the state-space model given in Eq. (9) of our Part 1 paper. For instance, as illustrated in Fig. 4, a very simple linear bias term in the state transition equation ($x_k = f(x_{k-1}, u_k) + \omega_k$) (see Fig. 4 (b)) could be translated into a highly nonlinear model discrepancy between the observation and the predicted output of the dynamic system model (i.e., model discrepancy of $h(\cdot)$, as illustrated in Fig. 4 (d)). In that case, it is more preferable to quantify model uncertainty of the state transition equation using the state-space model given in Eq. (9) of our Part 1 paper than that of the measurement equation using the formulation given in Eq. (5).

Next, we briefly summarize methods for the quantification of model uncertainty of the state transition equation.

(c) UQ of state transition equations

We now assume that the measurement equation in a state-space model is adequately modeled, and we mainly focus on UQ of the state transition equation. This assumption holds for many problems since the measurement equation is usually simpler than the state transition equation. If this assumption does not hold (i.e., the measurement equation has a large model bias), a two-step process can be followed. Since the measurement equation is quasi-static in nature, it can be corrected first using methods for static models based on data collected in a controlled environment (Xi et al., 2019). Following this first step, model uncertainty of the state transition equation can be quantified using the methods reviewed below. It is worth noting that, since the state transition equation models the transition of state variables over time (e.g., rate of change of state variables) and governs the dynamics of a dynamic system, Subramanian and Mahadevan (2019) referred to the bias of the state transition equation as "model form error" and the resulting discrepancy between observation and prediction of the model output as "model discrepancy". They also pointed out an important distinction between UQ of measurement equation and UQ of state transition equation (governing equation) that the recovery of the missing physics in the state transition equation allows for improving the prediction accuracy of the state-space model for extrapolation while it is difficult to achieve this purpose by just performing UQ of the measurement equation.

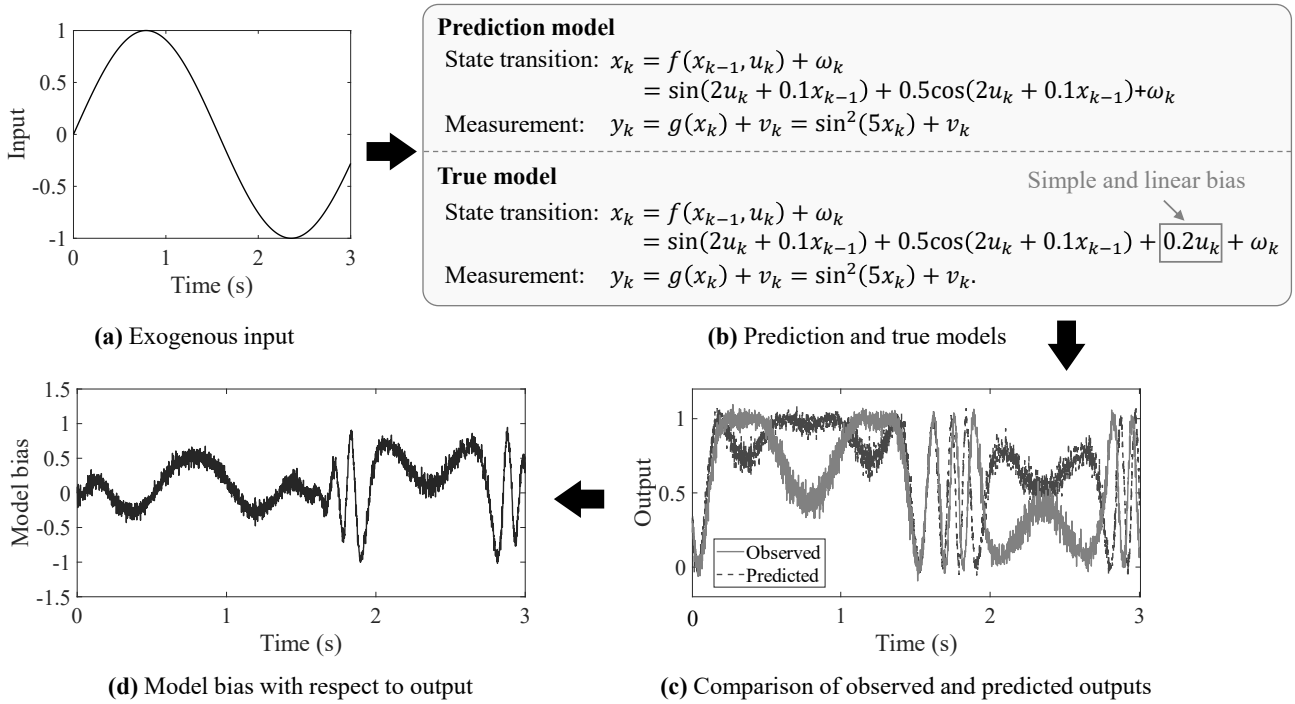
Numerous methods have been proposed in recent years to quantify model uncertainty of state transition equations (Wilkinson et al., 2011; Zhang et al., 2019; Subramanian and Mahadevan, 2019; Hu et al., 2019; Viana et al., 2021; Jiang et al., 2022c; Yucesan and Viana, 2020). These methods can be classified as the *category 2* methods reviewed in Sec. 2.1.2 (a), since a model discrepancy function and a noise term are used to account for model uncertainty of the governing equation. After model uncertainty is accounted for using a *category 2* method, the state transition equation (governing equation) given in Eq. (9) of our Part 1 paper becomes

$$\begin{aligned}\mathbf{x_k} &= \mathbf{f}(\mathbf{x}_{k-1},\ \boldsymbol{\theta}^*,\ \mathbf{u}_{k-1}) \\ &+ \hat{\boldsymbol{\delta}}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1};\ \boldsymbol{\eta}_\delta) + \boldsymbol{\gamma}_k,\end{aligned} \tag{6}$$

where $\hat{\boldsymbol{\delta}}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1};\ \boldsymbol{\eta}_\delta)$ is the model discrepancy function with unknown model parameters $\boldsymbol{\eta}_\delta$, similar to Eq. (5), $\boldsymbol{\gamma}_k$ is a vector of Gaussian noise variables with either an unknown mean vector and covariance matrix or zero means and an unknown covariance matrix.

According to how the unknown parameters $\boldsymbol{\eta}_\delta$ of the model discrepancy function are estimated, methods of this group can be further divided into two sub-groups.

- The first sub-group sets $\boldsymbol{\gamma}_k$ as $\boldsymbol{\omega}_k$ which is the process noise of the original state transition equation given in Eq. (3) or (9) of our Part 1 paper. After that, model bias $\boldsymbol{\delta}_k$ at each time instant t_k is estimated along with state variables $\mathbf{x}_k$ using one of the Bayesian filters given in Sec. 4.2.2 of Part 1 on state estimation and Bayesian filters. Based on the estimated $\boldsymbol{\delta}_k$, a predictive model $\hat{\boldsymbol{\delta}}(\mathbf{x},\ \mathbf{u};\ \boldsymbol{\eta}_\delta)$ is constructed to correct the state transition equation. To account for uncertainty in the estimated $\boldsymbol{\delta}_k$ and uncertainty introduced by setting $\boldsymbol{\gamma}_k$ as $\boldsymbol{\omega}_k$, a probabilistic predictive model such as a Gaussian process regression model is usually



(a) Exogenous input **(b)** Prediction and true models

(d) Model bias with respect to output **(c)** Comparison of observed and predicted outputs

Fig. 4: Illustration of how a simple linear model bias in the state transition equation could be translated into a highly nonlinear model discrepancy of the dynamic system output

constructed as $\hat{\boldsymbol{\delta}}(\mathbf{x},\ \mathbf{u};\ \boldsymbol{\eta}_\delta)$. Examples of such methods include Subramanian and Mahadevan (2019); Zhang et al. (2019); Hu et al. (2019).

- The second sub-group treats $\boldsymbol{\gamma}_k$ as a vector of Gaussian noise variables with either an unknown mean vector and covariance matrix or zero means and an unknown covariance matrix (i.e., the same treatment as Eq. (5)). The unknown distributional parameters of $\boldsymbol{\gamma}_k$ are then estimated along with unknown parameters $\boldsymbol{\eta}_\delta$ of $\hat{\boldsymbol{\delta}}(\mathbf{x},\ \mathbf{u};\ \boldsymbol{\eta}_\delta)$. To enable for the end-to-end training of a data-driven model of $\hat{\boldsymbol{\delta}}(\mathbf{x},\ \mathbf{u};\ \boldsymbol{\eta}_\delta)$, $\hat{\boldsymbol{\delta}}(\mathbf{x},\ \mathbf{u};\ \boldsymbol{\eta}_\delta)$ needs to be integrated with the original state-space model (i.e., more specifically the original state transition equation) in the training process. Examples of this sub-group include Hu et al. (2019); Yucesan and Viana (2020); Jiang et al. (2022c); Wilkinson et al. (2011); Viana et al. (2021). This sub-group of methods is very similar to Approach 4 (i.e., delta learning) of physics-informed ML in Sec. 3.4 of Part 1, which uses a data-driven ML model as $\hat{\delta}(\cdot)$ to recover the unmodeled physics.

Since the above two subgroups of methods can be classified as the *category 2* methods discussed in Sec. 2.1.2 (a), they inherit the advantage of the category 2 methods that the predictive capability of the model discrepancy function $\hat{\boldsymbol{\delta}}(\mathbf{x},\ \mathbf{u};\ \boldsymbol{\eta}_\delta)$ can help improve the prediction accuracy of the state transition equation under previously unseen conditions. Constructing an accurate model of $\boldsymbol{\delta}(\mathbf{x},\ \mathbf{u};\ \boldsymbol{\eta}_\delta)$, however, can be very challenging since the output of the state transition equation is time-varying and not directly measurable.

From the above review, we can conclude that most of the current UQ methods for dynamic system models implement either the *category 1* methods on measurement equations (Sec. 2.1.2 (b)) or the *category 2* methods on state transition equations (Sec. 2.1.2 (c)). Only a few methods apply the KOH framework (i.e., the *category 3* methods) to dynamic system models based on the formulation given in Eq. (5) (see Sec. 2.1.2 (b)).

As illustrated in Fig. 4, a small bias in the state transition equation could escalate as a highly nonlinear model discrepancy in the measurement equation, especially for a nonlinear dynamic system model. It implies that the reformulation of the state-space model in Eq. (5) could significantly increase the difficulty in quantifying model uncertainty. Since the state transition equation governs

the dynamics of a dynamic system, it is envisioned that quantifying model uncertainty of the state transition equation (Sec. 2.1.2 (c)) could be more effective in improving the prediction accuracy of a state-space model than quantifying model uncertainty of the measurement equation (Sec. 2.1.2 (b)). Given that the *category 1* and *category 2* methods for UQ of dynamic system models are getting mature, it is worth investing more research effort in the *category 3* methods. We expect such an increased investment will yield newer, more mature *category 3* methods that can help further improve the validity of digital models in digital twins, especially for the state transition equation.

The quantification of model uncertainty of dynamic system models could improve the accuracy and robustness of MPC by improving the prediction accuracy of state-space models (Rohrs et al., 1982, 1985; Liu and Li, 2002; Li et al., 2016), as has been discussed in Sec. 5.1 of our Part 1 paper on MPC, and enable model-based risk assessment for decision making under uncertainty. It plays a vital role in ensuring the effectiveness of digital twins in personalized control and optimization.

2.2 Optimization for digital twins (OPT)

The role of optimization in digital twins can be classified into two categories: offline optimization and online optimization (as illustrated in Fig. 3 of our Part 1 paper). Offline optimization occurs prior to the deployment of a digital twin. Online optimization takes place when a digital twin has been deployed and is in operation. In the subsequent sections, we briefly discuss various optimization techniques used for digital twins.

2.2.1 Optimization for sensor placement (offline)

Sensing is the forefront of the P2V connection and an indispensable element of a digital twin. Many different types of sensors, such as strain gauges, acoustic emission sensors, thermal cameras, optical cameras, and others, can be employed to collect data capturing different aspects of the physical system performance in-situ. Sensor data collected from a physical system serves as the inputs to the twinning enabling techniques reviewed in Sec. 4 of Part 1 paper, and are essential to establishing the P2V connection.

No matter what type of sensor is used, an essential question that needs to be answered is where the sensors should be placed in the physical system. The locations where the sensors are placed could significantly affect the quality of the collected data and ultimately the inference of other information, which would affect the effectiveness of the P2V connection and, eventually, the performance of the digital twin. Therefore, it is particularly important to optimize the sensor placement at the design stage before online deployment of a digital twin. Moreover, it is worth mentioning that the number of sensors and sensor types can also be treated as design variables in sensor network design. That would add another level of complexity to sensor network design since sensor network performance would be conditional on the number of sensors and sensor types. Several studies have been conducted in recent years to address sensor network design at this higher level of complexity. For example, some studies consider the number of sensors as another design variable in addition to sensor locations. Yang et al. (2020a) treated the number of sensors and sensor locations as design variables in a genetic algorithm-based sensor network design method. Similarly, An et al. (2022a,b) optimized the number of sensors and sensor locations simultaneously using a non-dominated sorting genetic algorithm II in sensor network design for vibration-based damage detection. While optimizing the number of sensors and sensor types is also important for sensor network design, this section intentionally concentrates on sensor placement optimization, since it is fundamental to various sensor network design problems.

As mentioned in Sec. 2.1, two types of uncertainty sources, namely aleatory and epistemic, are presented in digital twins. Epistemic uncertainty could be reduced through the data collected from sensors in the P2V connection. Aleatory uncertainty, however, is irreducible and is inherent in a digital twin. To ensure that the sensors collect the most informative data in the presence of natural variability (i.e., aleatory uncertainty), it is important to consider aleatory uncertainty in the optimization of sensor placements. To this end, a generalized model for sensor placement optimization under uncertainty can be formulated as

follows

$$\mathbf{d}^* = \arg\max_{\mathbf{d}\in\Omega}\{L(\mathbf{d},\ \boldsymbol{\alpha})\}, \tag{7}$$

where $\mathbf{d}$ is a vector or matrix consisting of sensor locations such as the coordinates where the sensors are placed, Ω is the spatial design domain, $\boldsymbol{\alpha}$ is a vector of random variables representing the aleatory uncertainty during the operation of a physical system and its digital twin, and $L(\mathbf{d},\ \boldsymbol{\alpha})$ is a cost function of $\mathbf{d}$ and $\boldsymbol{\alpha}$. As mentioned above, accounting for aleatory uncertainty $\boldsymbol{\alpha}$ in the sensor placement optimization is vital to ensuring that the designed sensor network can well perform its intended function when the digital twin is put into online operation.

Three key research questions usually need to be addressed in solving the above sensor network design optimization model:

1. How to formulate the cost function $L(\mathbf{d},\ \boldsymbol{\alpha})$?
2. How to efficiently and accurately evaluate the cost function in the presence of uncertainty?
3. How to efficiently solve the optimization model given in Eq. (7)?

In what follows, we briefly review commonly used approaches to tackle the above three research questions.

(a) Cost function

The cost function $L(\mathbf{d},\ \boldsymbol{\alpha})$ needs to be formulated in consideration of the P2V connection and the sensor type.

For instance, for a network of wireless sensors, the cost function could be formulated as the resilience or vulnerability of the sensor network and needs to consider the routing algorithm for effective communication among different wireless sensors (Anand et al., 2005). Since wireless sensors are usually self-powered, energy efficiency has also been an important consideration in the cost function for sensor network optimization (Sachan et al., 2012). A few representative review papers about various cost functions and the corresponding optimization models of wireless sensor networks are available in Kulkarni and Venayagamoorthy (2010); Adnan et al. (2013); Asorey-Cacheda et al. (2017).

For wired sensors, many performance metrics have been proposed in the past decades to optimize their placement. The commonly used cost functions can be roughly grouped into the following categories

- **Information gain**: This class of metrics/cost functions measures the amount of information contained in the data collected from a sensor network design for uncertainty reduction (Nath et al., 2017; Yang et al., 2021; Gomes et al., 2019; Hu et al., 2017; Meo and Zumpano, 2005; Kammer, 1991). Various metrics have been proposed in the information science domain to quantify information gain from data. The most widely used ones in sensor placement optimization include

  1. *Fisher's information matrix*: It quantifies the information gain based on the assumption that the posterior distribution is a multivariate Gaussian distribution (Gomes et al., 2019; Hu et al., 2017; Meo and Zumpano, 2005; Kim et al., 2018; Heydari et al., 2020). Some examples of cost functions for sensor network design based on the Fisher's information matrix include A-optimality criterion (trace), D-optimality criterion (determinant), and E-optimality criterion (largest eigenvalue) (Gomes et al., 2019; Kammer, 1991; Hu et al., 2017; Kim et al., 2018; Heydari et al., 2020). For instance, Kim et al. (2018) proposed a stochastic effective independence method for optimal sensor placement with A-optimality criteria. This method showed better performance in handling system uncertainty compared to an existing method. Another study along the same line is Heydari et al. (2020), where the authors used D-optimality criterion to optimize sensor placement for source localization based on the received signal strength difference.
  2. *Kullback–Leibler divergence*: It quantifies the amount of information gained from data using the relative entropy. When the Kullback–Leibler divergence is used in a sensor network design with the consideration of various uncertainty sources, the cost function is formulated as (Nath et al., 2017; Yang et al.,

2021)

$$L(\mathbf{d},\ \boldsymbol{\alpha}) = \int_{\Omega_{\boldsymbol{\theta}}} \int_{\Omega_{\boldsymbol{\alpha}}} D_{KL}(y(\boldsymbol{\alpha},\ \mathbf{d}),\ \boldsymbol{\alpha}) \times p_{Y|\boldsymbol{\theta}}(y(\boldsymbol{\alpha},\ \mathbf{d})|\boldsymbol{\alpha},\ \boldsymbol{\theta}) f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \mathbf{d}\boldsymbol{\alpha}\mathbf{d}\boldsymbol{\theta}, \tag{8}$$

where $D_{KL}(y(\boldsymbol{\alpha},\ \mathbf{d}),\ \boldsymbol{\alpha})$ is the Kullback–Leibler divergence for given realization of $\boldsymbol{\alpha}$ and synthetic observations $y(\boldsymbol{\alpha},\ \mathbf{d})$ which are generated using physics-based modeling (see Sec. 3.2 of Part 1), $\boldsymbol{\theta}$ is a vector of epistemic model parameters (see Sec. 4.2.4 of Part 1), such as model parameters of Paris's law for crack growth or capacity of a battery, $f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})$ and $f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ are respectively the joint probability density function of $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$.

It is worth mentioning that the Kullback–Leibler divergence is one type of $f-$divergence. Other types of $f-$divergences can also be used to quantify the information gain. A comparison of different $f-$divergences for sensor network design optimization is given in Yang et al. (2021).

- **Probability of detection**: Cost functions falling within this group aim to minimize the type I and type II errors (Downey et al., 2018; Flynn and Todd, 2010; Guratzsch and Mahadevan, 2010; Wang et al., 2015). The type I error is related to the scenario that a healthy (undamaged) state is incorrectly identified as damaged, i.e., a false alarm. The type II error is related to the probability that a damaged state is classified as healthy (undamaged), i.e., false negative, missed detection, or error of omission. For instance, Downey et al. (2018) optimized the placement of sensors used in the construction of accurate strain maps for large-scale structural components by minimizing the type I and II errors. Flynn and Todd (2010) proposed Bayes risk-based function for sensor placement optimization by associating decision costs with the type I and II errors. Similarly, the probability of detection has been employed as a metric for sensor placement optimization in Guratzsch and Mahadevan (2010); Wang et al. (2015), where aleatory uncertainty in the operation of physical systems was explicitly accounted for via probabilistic analysis.
- **Modal assurance criterion**: Modal assurance criterion is a metric that is widely used in structural dynamics domain to quantify the similarity of mode shapes (Allemang, 2003). It has also been applied to sensor placement optimization for SHM. For example, Carne and Dohrmann (1994) proposed an approach to determine the optimal number and locations of sensors, where the modal assurance criterion was used to correlate a modal test with an FEA model. Following the work of Carne and Dohrmann (1994), Yi et al. (2011) minimized the off-diagonal elements of the modal assurance criterion matrix to optimize the sensor configuration for SHM. An et al. (2022a) considered model uncertainty in the root mean square error derived from the off-diagonal elements of a modal assurance criterion matrix in sensor network design for vibration-based damage detection.
- **Value of information (VoI)**: VoI has emerged as a cost framework for sensor network design optimization in recent years and has gained much attention in a broad range of domains (Bisdikian et al., 2013; Malings and Pozzi, 2016; Basagni et al., 2014; Cantero-Chinchilla et al., 2020; Chadha et al., 2021). This metric is particularly attractive because it directly quantifies the expected VoI of the data collected from a particular location and/or time by considering various costs associated with decision alternatives. The generalized form of the expected VoI for a sensor network design is defined as (Malings and Pozzi, 2016; Chadha et al., 2021)

$$\mathrm{EVOI}(\mathbf{d}) = \Psi_{\mathrm{prior}}(\boldsymbol{\chi}_{\mathrm{prior}}) - \Psi_{\boldsymbol{\alpha},\ \boldsymbol{\theta}}(\boldsymbol{\chi}_{\mathbf{d}},\ \mathbf{d}), \tag{9}$$

where $\Psi_{\mathrm{prior}}(\boldsymbol{\chi}_{\mathrm{prior}})$ is the cost associated with the decision $\boldsymbol{\chi}_{\mathrm{prior}}$ by only considering the prior information of the epistemic uncertain parameters $\boldsymbol{\theta}$, $\Psi_{\boldsymbol{\alpha},\ \boldsymbol{\theta}}(\boldsymbol{\chi}_{\mathbf{d}},\ \mathbf{d})$ is the expected cost associated with optimal decision $\boldsymbol{\chi}_{\mathbf{d}}$ based on pre-posterior analysis of $\boldsymbol{\theta}$ for a sensor placement design $\mathbf{d}$ and with the consideration of other aleatory uncertain variables $\boldsymbol{\alpha}$ in decision making.

Note that the above reviewed metrics are not exhaustive, but representative. Interested readers can find more comprehensive reviews on various

metrics for sensor placement optimization in Ostachowicz et al. (2019); Tan and Zhang (2020); Gupta et al. (2010).

(b) UQ methods

As mentioned above, uncertainty sources need to be considered in the cost function since offline sensor placement optimization is performed before the online deployment of a digital twin. While the consideration of uncertainty sources could ensure the performance of the designed sensor network in collecting the most useful information after the deployment, it poses significant computational challenges to the evaluation of the cost functions. Efficient and accurate UQ methods have been developed to tackle the computational challenge and can be categorized into two classes.

- **Analytical or numerical approximation**: This class of methods approximates the cost function in the presence of uncertainty using analytical expressions (Long et al., 2013) or numerical approximations (Yang et al., 2021; Guratzsch and Mahadevan, 2010; Wang et al., 2015). For example, due to the lack of an analytical solution to the Kullback–Leibler divergence and the required high-dimensional integration to compute the expected value, solving Eq. (8) is generally computationally demanding. To address the computational challenge, analytical approximation of the Kullback–Leibler divergence has been pursued using Laplace approximations (Long et al., 2013). Motivated by tackling the same computational challenge in sensor placement optimization, Yang et al. (2021) approximated the high-dimensional integration in Eq. (8) using univariate dimension reduction. When the probability of detection is employed as the cost function and needs to be estimated using physics-based probabilistic analysis, Monte carlo sampling-based approximations using finite element simulations have been developed in Guratzsch and Mahadevan (2010) and Wang et al. (2015).
- **Surrogate-based approximation**: The first step in this class of methods is to construct a surrogate of
  - *either* the physics-based simulation model that is used to generate synthetic observations for sensor placement optimization (Huan and Marzouk, 2014; Eshghi et al., 2019),
  - *or* the cost function with respect to uncertain variables (Nath et al., 2017; An et al., 2022b).

  After the construction of the surrogate, Monte Carlo simulation is employed to evaluate the cost function. For instance, polynomial chaos expansion and Gaussian process regression surrogates have been built to replace the original physics-based models for the evaluation of the Kullback–Leibler divergence (Huan and Marzouk, 2014) and probability of detection (Eshghi et al., 2019), respectively. For the direct surrogate modeling of cost functions, Nath et al. (2017) constructed a Gaussian process regression model (surrogate) of the Kullback–Leibler divergence, making it possible to efficiently compute the expected Kullback–Leibler divergence in sensor placement for calibration of spatially varying model parameters. An et al. (2022b) built a Gaussian process regression model for the determinant of the Fisher information matrix and then computed the mean and standard deviation of the determinant using Monte Carlo simulation based on the surrogate model.

(c) Optimization methods

Once an appropriate cost function $L(\mathbf{d}, \boldsymbol{\alpha})$ is established and the uncertainty of the cost function is quantified, the last key research question is how to efficiently solve the optimization model formulated in Eq. (7). Current optimization methods for sensor placement optimization can in general be grouped into three categories as follows.

- **Evolutionary optimization methods**: The optimization problem could be solved by an evolutionary optimization method directly, such as a genetic algorithm (Yao et al., 1993; Liu et al., 2008; Yi et al., 2011; An et al., 2022a,b,a,b; Ehsani and Afshar, 2010; Downey et al., 2018; Flynn and Todd, 2010), simulated annealing (Tong et al., 2014; Chen et al., 1991), or particle swarm optimization (Zhang et al., 2014; Li et al., 2015), if it is computationally cheap to evaluate the cost function (e.g., probability of detection with analytical expressions, modal assurance criterion) and provided that the number of sensors is small. For instance, genetic

algorithm has been employed for sensor placement optimization by using modal assurance criterion (An et al., 2022a,b) or probability of detection (Ehsani and Afshar, 2010; Downey, Hu, and Laflamme, 2018; Flynn and Todd, 2010) as cost function since they can be evaluated very efficiently.

- **Greedy algorithm-based methods**: In contrast, as the number of sensors increases or the cost function is increasingly computationally demanding to evaluate, such as the expected Kullback–Leibler divergence given in Eq. (8), it would be computationally intractable to directly solve Eq. (7) with evolutionary optimization methods, since evolutionary optimization methods usually need to evaluate the cost function thousands of times to find a near-optimal solution. To overcome this challenge, methods have been developed using greedy algorithms in conjunction with efficient UQ methods discussed in Sec. 2.2.1 (b) (Yang et al., 2021; Nath et al., 2017; Sela and Amin, 2018; Malings et al., 2015; Błachowski et al., 2020). In greedy algorithm-based methods, the sensor placement is optimized sequentially. Basically, we select the optimal sensor placement one-by-one conditioned on previous sensor placements. By doing so, it allows us to convert a high-dimensional optimization problem into multiple low-dimensional optimization problems that can be solved sequentially. For instance, in order to place 10 sensors on a 3-dimensional structure called miter gate (see Fig. 5), a 30 dimensional optimization problem needs to be solved, if an evolutionary optimization method is employed directly. Instead of directly solving the 30 dimensional optimization problem, Yang et al. (2021) optimized the sensor placement one-by-one conditioned on previous sensor placements. In each iteration, only a three-dimensional optimization problem is solved using Bayesian optimization method (Yang et al., 2021).
- **Reinforcement learning-based methods**: Even though the greedy algorithm-based methods make sensor placement optimization under uncertainty computationally more tractable, it may lead to sub-optimal solutions due to the nature of greedy algorithms. Reinforcement learning-based methods have recently been proposed to alleviate the limitation of greedy-based methods. In reinforcement learning-based methods, sensor placement optimization is formulated as a sequential decision-making problem, such as a Markov decision process model. This problem is solved using one of many reinforcement learning algorithms, such as dynamic programming, Q-learning, policy gradient reinforcement learning, and deep Q-learning (Alsheikh et al., 2015; Wang and Wang, 2006; Wang et al., 2019; Kaveh et al., 2022; Shen and Huan, 2021). A unique properly of this problem is it considers the impact of a candidate sensor placement solution on both current decision making and the placements of other sensors and decision making at future time instances (i.e., look-ahead). For example, Wang et al. (2019) and Shen and Huan (2021) have developed reinforcement learning-based sensor placement optimization methods for spatiotemporal modeling and Bayesian model updating, respectively. Results of their papers show that reinforcement learning-based methods tend to be more effective in finding optimal solutions than greedy algorithm-based methods and generic algorithm-based methods (Wang et al., 2019; Shen and Huan, 2021). A dedicated discussion on other applications of deep reinforcement learning in digital twins can be found in Sec. 6.3 of our Part 1 paper.

Optimizing a sensor network for a physical system allows the most informative data to be collected from the physical system. The collection of informative datasets will significantly improve the performance of the P2V connection for model updating as well as the efficacy of the overall digital twin in support of real-time decision making and control. Figure 5 shows an example of sensor placement optimization as part of a miter gate digital twin project sponsored by the U.S. Army Corps of Engineers (Vega et al., 2021). A finite element structural analysis model was first developed to predict the structural response under different conditions, as shown in Fig. 5 (a). Based on the structural analysis model, sensor placement was optimized to collect data from the miter gate to estimate the level of structural damage located at the lower-left corner of the gate. The damage level

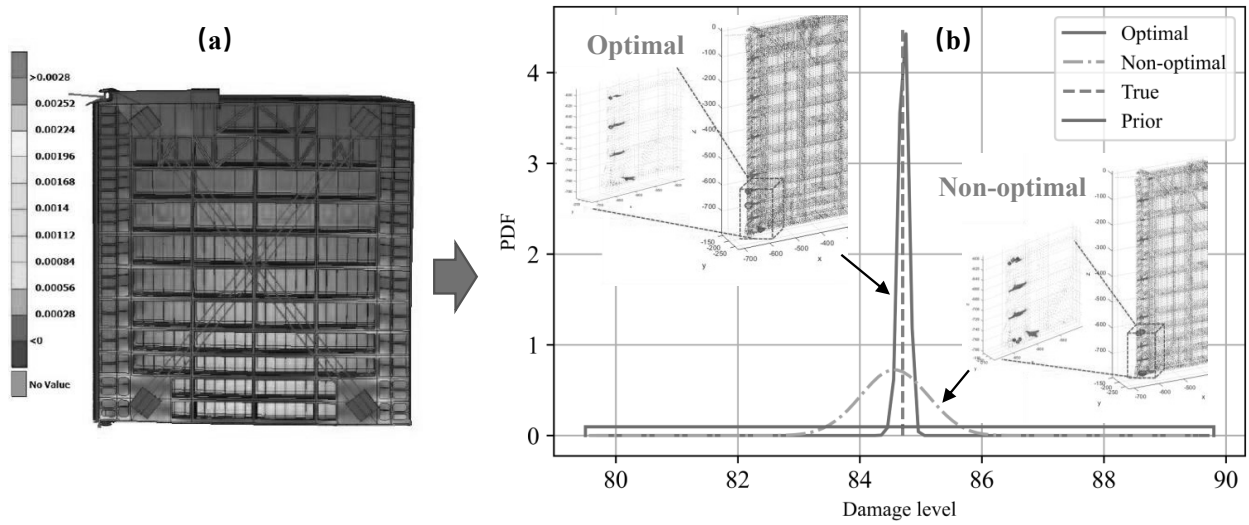


Fig. 5: Impact of sensor network design on probabilistic model updating (P2V connection). (a) Finite element simulation of a miter gate; (b) Comparison of posterior distributions of damage level obtained from the optimal sensor design and a non-optimal design. (Yang et al., 2021)

(i.e., gap length) needed to be estimated by updating the finite element model given in Fig. 5 (a) using Bayesian filters described in Sec. 4.2.2 of our Part 1 paper. Figure 5 (b) compares the estimated posterior damage level obtained from the optimal sensor design and a non-optimal sensor design at a certain time instant. This figure shows that the posterior damage estimate obtained from the optimal sensor design is much more concentrated than that from the non-optimal design. Clearly, sensor network optimization led to a significant reduction in the uncertainty of the structural damage estimate compared to a non-optimal design. This example highlights the added value of sensor network design optimization in digital twins.

2.2.2 Optimization for physical system modeling (offline)

As mentioned in Sec. 4.2.1 of our Part 1 paper, digital state space is usually designed to be simple enough to make online model updating feasible and tractable. A digital state consists of

- a set of state variables $\mathbf{x}$ which are the smallest set of variables that determine the state of the physical system, and
- a group of model parameters $\tilde{\boldsymbol{\theta}} = [\boldsymbol{\lambda},\ \boldsymbol{\theta}]$, in which $\boldsymbol{\theta}$ is a very small subset of the $\tilde{\boldsymbol{\theta}}$ that will be updated online along with state variables $\mathbf{x}$ using methods described in Sec. 4.2.4 of Part 1, and $\boldsymbol{\lambda}$ represents the remaining model parameters that will not be updated online.

In order to bridge the gap between the initial digital model and the physical counterpart, the uncertain model parameters $\tilde{\boldsymbol{\theta}}$ need to be calibrated offline using experimental data before deploying the digital twin. Numerous approaches have been proposed to perform such an offline calibration of digital models, including Bayesian calibration-based methods and optimization-based methods.

In Bayesian calibration-based methods, the optimal model parameters $\tilde{\boldsymbol{\theta}}^*$ can be estimated by solving the following optimization problem

$$\tilde{\boldsymbol{\theta}}^* = \arg\max_{\tilde{\boldsymbol{\theta}} \in \Omega} \{f(\tilde{\boldsymbol{\theta}}|\mathbf{y}_e)\}, \tag{10}$$

where $\mathbf{y}_e \in \mathbb{R}^{n_e}$ is a vector of experimental observations, $f(\tilde{\boldsymbol{\theta}}|\mathbf{y}_e)$ is the posterior distribution of $\tilde{\boldsymbol{\theta}}$ for given experimental observation $\mathbf{y}_e$. Note that the true values of $\tilde{\boldsymbol{\theta}}$ are fixed but unknown to us, since the uncertain model parameters $\tilde{\boldsymbol{\theta}}$ are considered to be epistemic uncertainty only (see Sec. 2.1.2).

$f(\tilde{\boldsymbol{\theta}}|\mathbf{y}_e)$ can be obtained using various Bayesian updating methods, such as the classical

Markov chain Monte Carlo simulation, Gibbs sampling, and slice sampling. When model uncertainty of a digital model is considered, the uncertain model parameters $\tilde{\boldsymbol{\theta}}$ can be calibrated concurrently with a model discrepancy term using the KOH framework (Kennedy and O'Hagan, 2001) (see Eq. (3) in Sec. 2.1.2). It has been shown that accounting for model discrepancy during Bayesian calibration can not only improve the prediction accuracy of the digital model, but it can also lead to an improved accuracy in estimating the posterior distribution of the unknown model parameters $\tilde{\boldsymbol{\theta}}$. As indicated in Fig. 6, the posterior distribution of a unknown model parameter θ considering model discrepancy is closer to the true value than its counterpart without accounting for model discrepancy. Additionally, it is worth mentioning that the Bayesian filters described in Sec. 4.2.2 of Part 1 can also be used to obtain $f(\tilde{\boldsymbol{\theta}}|\mathbf{y}_e)$ since a special case of online model updating is the offline calibration where the Bayesian filters are used to update only model parameters instead of both state variables and model parameters.

The estimation given in Eq. (10) is also called maximum a posteriori estimate. In practice, a point estimate is used instead of the joint posterior distribution is to keep the number of uncertain model parameters as low as possible in the digital model, and thus make online model updating of digital twins feasible and tractable. While Bayesian calibration methods under the KOH framework have been shown to be accurate and robust in estimating uncertain model parameters, the implementation of those methods is relatively complicated and the required computational effort may be quite high, especially when $\tilde{\boldsymbol{\theta}}$ is high-dimensional. Therefore, optimization-based methods are often employed as an alternative that largely alleviates the computational burden in practice.

Optimization-based calibration methods estimate $\tilde{\boldsymbol{\theta}}$ by maximizing or minimizing a calibration metric as follows

$$\tilde{\boldsymbol{\theta}}^* = \arg\min_{\tilde{\boldsymbol{\theta}}\in\Omega} \text{ or } \arg\max_{\tilde{\boldsymbol{\theta}}\in\Omega}\{C(\mathbf{y}_e,\ g(\tilde{\boldsymbol{\theta}},\ \boldsymbol{\alpha}))\}, \tag{11}$$

where $C(\mathbf{y}_e,\ g(\tilde{\boldsymbol{\theta}},\ \boldsymbol{\alpha}))$ is a calibration metric, $g(\tilde{\boldsymbol{\theta}},\ \boldsymbol{\alpha})$ is the prediction of the digital model for given $\tilde{\boldsymbol{\theta}}$ and $\boldsymbol{\alpha}$, and $\boldsymbol{\alpha}$ stands for the aleatory uncertain variables in the digital model (the same as that in Sec. 2.2.1).

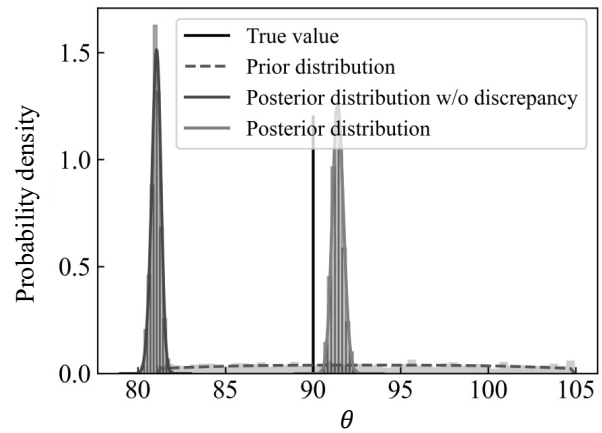


Fig. 6: Comparison of posterior distributions obtained with and without accounting for model discrepancy term (Jiang et al., 2022b).

Various calibration metrics have been proposed in the past decades. For instance, the least squares method as described in Eq. (7) in Sec. 4.2.4 of our Part 1 paper is an optimization-based method with the mean squared error as the calibration metric. The least squares method is the easiest to implement, and is probably the most widely used one in industry. But it is sensitive to outliers in experimental data. If a likelihood function is used as the calibration metric, it is called the maximum likelihood estimation method (Xiong et al., 2009). This method has shown similar performance as Bayesian methods under the KOH framework (Xiong et al., 2009). But it may not perform well if the amount of experimental data for calibration is small. Some other examples of calibration metrics include the moment matching metric which compares the difference between the statistical moments obtained by experiments and prediction (Bao and Wang, 2015), similarity metric that measures the similarity between prediction and experiments (Cha, 2007), and the marginal probability and correlation residual metric considering both marginal probability and correlation coefficient residuals (Kim et al., 2020).

As mentioned above, all methods have their own advantages and disadvantages. Among them, Bayesian methods, the maximum likelihood estimation method, and the least squares method are the three most widely used ones. The selection of

an appropriate method is mainly dependent on the amount of available data and decision maker's acceptable level of complexity. Furthermore, the following topics also play a vital role in bringing the initial digital model closer to the physical system.

1. *Model validation*: Model validation is an essential step to validate the digital model after calibration of the model offline. It is "the process of determining the degree to which a model or a simulation is an accurate representation of the real world from the perspective of the intended uses of the model or the simulation" (NASA, 2008; Mahadevan et al., 2022). In order to quantitatively quantify the agreement between the digital model prediction and experimental observations, various statistical metrics have been proposed, including Bayesian hypothesis testing (Jiang and Mahadevan, 2009), reliability-based metric (Rebba and Mahadevan, 2008; Ao et al., 2017b), area metric (Li et al., 2014), etc. An essential characteristic of various validation metrics is that they account for various uncertainty sources in the digital model used for calibration and in the experiments used to evaluate model validity. Liu et al. (2011) and Ling and Mahadevan (2013) analyzed the pros and cons of different metrics through comparative studies. For instance, Liu et al. (2011) pointed out that small perturbations in the pre-specified confidence level could significantly affect the rejection or non-rejection of a digital model using classical hypothesis testing. Ling and Mahadevan (2013) concluded that both a Bayes factor and reliability-based metric could be mathematically related to the p-value metric in classical hypothesis testing. An appropriate validation metric should be selected to validate the calibrated digital model according to the application by analyzing the pros and cons of different metrics. We direct interested readers to Liu et al. (2011) and Ling and Mahadevan (2013) for more detailed discussions on this important topic.
2. *Experimental design optimization*: Experimental design optimization is a process of optimizing experimental input settings to collect the most informative experimental data for model calibration and validation (Ao et al., 2017a; Hu et al., 2017; Huan and Marzouk, 2013, 2014). Even though formulated in a different context, experimental design optimization is fundamentally the same as sensor placement optimization discussed in Sec. 2.2.1 and can be considered as a sub-topic of sensor placement optimization. It can help reduce the required number of experiments for the calibration of a digital model offline.

After the offline calibration of the digital model, $\tilde{\boldsymbol{\theta}}^*$ obtained from Eq. (10) or (11) will be used as the initial values of $\tilde{\boldsymbol{\theta}}$. A small subset of $\tilde{\boldsymbol{\theta}}$ denoted as $\boldsymbol{\theta}$ will be updated along with state variables $\mathbf{x}$ using the methods discussed in Sec. 4.2.4 of our Part 1 paper to account for the fact that some parameters (e.g., battery capacity) change very slowly over the life-cycle of a physical system.

2.2.3 Optimization for predictive decision making (online)

(a) Real-time requirements of digital twins

When discussing real-time requirements of digital twins, it is important to understand that the definition of "real-time" varies depending on the application. In general, the definition of real-time is *the minimum computational speed required to achieve seamless and uninterrupted optimization, prediction, and control of the system of interest.* Ultimately, the timescale of the system of interest is what defines the requirements for real-time computing. Take for example a digital twin built to model the degradation of a lithium-ion battery cell. A Li-ion cell is designed to last many thousands of cycles, which in standard applications, is on the time scale of years. In this case, real-time optimization and control related to a Li-ion cell's degradation needs to be computed on the time scale of days or weeks in order to enable timely control of its usage. On the other hand, high-rate systems like ultrasonic vehicles, hypersonic weapons, blast mitigation systems, and vehicle crashes operate on much shorter time scales, often $100ms$ or shorter (Dodson et al., 2022). When modeling these systems with a digital twin, it is much more difficult to ensure that sensing, prediction, and control can take place on the desired time scales. Research in this area is actively investigating modeling techniques which can meet

the demanding updating and predicting requirements. Sometimes, it is often intractable to set the requirement that the state estimation model operate on timescales shorter than the timescale of the system of interest. This is especially the case for very high-rate (< 100 μs) and ultra high-rate systems (< 1 μs). When this is the case, researchers will define an acceptable time delay, that if the state estimation model can achieve, would be useful to a larger predictive control framework. Examples of high-rate system modeling research include work by Yan et al. (2021). In their paper, they investigated using a simplified physics-based model to track, update, and predict the state of highly dynamic systems. Their experiments on two different test setups showed the model was able to update and predict with an average computation time of 93 μs. Other work by Barzegar et al. (2022) investigated using a deep-learning-based model architecture for high-rate system state prediction. Their proposed recurrent neural network, used for state estimation in high-rate structural health monitoring (HRSHM) applications, achieved accurate predictions with an average computational time of 25 μs.

(b) Real-time optimization of additive manufacturing processes

Offline optimization as described in above sections is commonly applied to high-level functions in smart manufacturing that do not require real-time optimization. Examples of these high-level functions include product design, production planning, and maintenance scheduling (although online optimization, not in real-time, is required to schedule maintenance in some cases). These functions, as defined in ISA-95, have typical cycles from hours to months (ISA, 2010). For those activities, open-loop optimizations are applied and no feedback-based adjustments involved in executing the optimal decisions. However, process controls, especially for complex processes with high uncertainties and significant disturbances, require continuous adaption of control strategy and real-time optimization.

Additive manufacturing is one of such complex layer-by-layer fabrication processes. For example, the metal powder bed fusion (LPBF) process involves spreading a thin layer of metal powder followed by exposure to high-intensity laser energy directed in scanned trajectories defined by digital models. The build process involves multiple physical phenomena: heat absorption, melt pool formation, solidification, and even re-melting and re-solidification (Frazier, 2014). A great number of factors affect the quality of additively manufactured parts, including processing parameters such as laser power and scan velocity, environmental parameters such as chamber temperature and humidity, as well as the non-deterministic material powder characteristics. The complex and stochastic nature of the additive manufacturing process requires real-time optimization for stable process and controllable part quality.

Figure 7 shows a layerwise real-time optimization strategy for laser powder bed fusion process control. Process control commands and melt pool monitoring data collected from previous builds are used as training data for a melt pool size prediction model, which can be represented as:

$$y_i = f(t_i, P_i, v_i, \theta_i^{\Delta t}, \theta_i^{\Delta d}, J, \lambda, A_{avg}, A_{max}, A_{avr}) \tag{12}$$

where y_i represents the melt pool size at step i, t_i represents the scan time, P_i is the current laser power, v_i is the current scan speed, $\theta_i^{\Delta t}$ is the temporal-accumulated prior scan effects, $\theta_i^{\Delta d}$ is the spatial-accumulated prior scan effect, J is the total energy input on the previous layer, λ represents the laser idle time from the end of the previous layer to start of current layer, and A_{avg}, A_{max}, and A_{var} represent the statistical features of the melt pool size within the previous layer neighborhood of the current scanning position (Yang et al., 2020b). A neural network was trained to predict melt pool size accurately based on process parameters and earlier melt pool measurements at the same layer and from the previous layer.

The machine learning model trained from prior builds can be used for real-time layerwise scan parameter optimization (Yeung et al., 2020). In Fig. 7, the objective of the optimization is to regulate the melt pool size. To achieve this goal, the potential control variables for optimization can be laser power, scan speed, and laser scan path. However, modifying scan speed or laser scan path requires significant computing efforts. Therefore only laser power is selected as the single control

variable for the optimization, which focuses on managing the melt pool size into a desired range.

(c) Real-time mission planning

The continual data flow from a physical system allows its digital twin to closely map, monitor, and control real-world entities. Combined with P2V twinning enabling technologies, such as the Bayesian model updating methods described in Sec. 4.2 of Part 1, digital twins provide an adaptive mechanism to support data-driven decision automation in accordance with asset-specific conditions.

One representative application that demonstrates a digital twin's capability in adaptive decision support is mission planning. For example, Kapteyn et al. (2021) considered the structural health of an UAV in a mission planning task, where the right wing of the UAV was assumed to have two defective regions on the upper surface. Based on a unifying probabilistic graphical model built upon Bayesian inference, Kapteyn et al. (2021) showcased the application of a structural digital twin in the operations management of a UAV (see Fig. 13 of our Part 1 paper for a graphical illustration). The structural digital twin, after it was calibrated to a specific as-manufactured asset, assimilated incoming sensor data on the UAV to perform in-flight SHM and carry out health-aware mission planning, where optimal maneuvers (i.e., turn and bank angle) were derived using the structural digital twin of the UAV in consideration of its evolving structural health. Sisson et al. (2022) pursued a similar idea to minimize the stress experienced by critical mechanical components of rotorcraft via optimizing the horizontal and vertical flight velocities for mission planning. Towards this goal, a digital twin approach was developed to diagnose component health and estimate the damage growth under different velocity conditions.

Another example is the real-time mission planning of off-road autonomous ground vehicles as illustrated in Fig. 8. These vehicles operate in highly stochastic, harsh, and uncontrolled off-road environments. They have been used to replace humans in the battlefield for military applications (Jiang et al., 2021), in the agricultural field to reduce labor requirements (Mousazadeh, 2013), and in space for exploration of the Moon and Mars (Johnson et al., 2015). A major challenge in realizing off-road autonomy is the high uncertainty in vehicle mobility (e.g., energy consumption and maximum attainable speed) caused by complex terrain conditions, such as mud, sand, and grass (Jiang et al., 2022a). Digital twin technology provides a promising solution to this challenging issue and plays a vital role in enabling real-time mission planning of off-road autonomous ground vehicles.

As illustrated in Fig. 8, a high-fidelity digital model of an off-road autonomous ground vehicle has been developed as part of the modeling and simulation effort of the U.S. Army Ground Vehicle Systems Center (ARC, 2022). The digital model predicts vehicle mobility under different terrain conditions and allows for the generation of a probabilistic mobility map (Jiang et al., 2021). The mobility map provides critical information about where the vehicle can go (i.e., GO) and where it cannot (i.e., NOGO). It is essential for the mission planning of the autonomous vehicle since it enables the vehicle to predict where the obstacles are. As the physical system of a vehicle travels along a path, as indicated in the orange color in Fig. 8, vehicle mobility data and terrain information are collected through torque and speed sensors, LiDAR, and cameras. The collected mobility data can be used to update the digital mobility model using a Bayesian model updating method discussed in Sec. 4.2 of Part 1. Subsequently, the probabilistic mobility map can be updated. The real-time mission planning can then be performed based on the updated mobility map to ensure the success of a mission under uncertainty (as illustrated in Fig. 8).

Real-time path planning of off-road autonomous ground vehicles is a Nondeterministic Polynomial (NP)-hard optimization problem. Approaches have been developed for mission planning using sampling-based methods and search-based methods. Sampling-based methods randomly draw samples within a map and construct a space-filled tree to identify the optimal path. Some examples of sampling-based methods include Rapid-exploring Random Tree (RRT) (Kuffner and LaValle, 2000), RRT* (Gammell et al., 2014), R2-RRT* (Jiang et al., 2022a), etc. Search-based methods identify the shortest path based on a pre-defined network or graph using searching algorithms, such as Dijkstra's algorithm

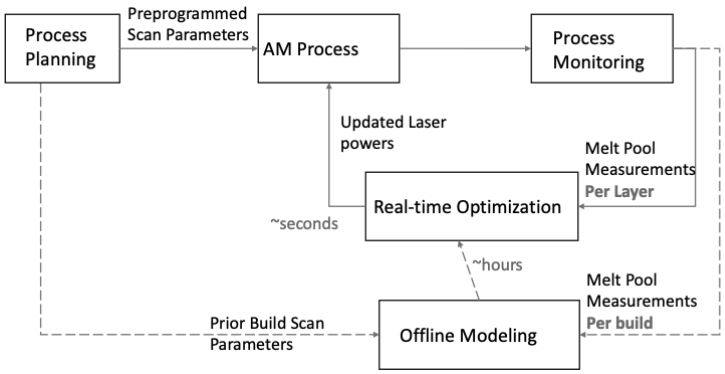


Fig. 7: Real-time optimization for powder bed fusion process control

(Barbehenn, 1998), and its optimized version, A* algorithm (Duchoň et al., 2014).

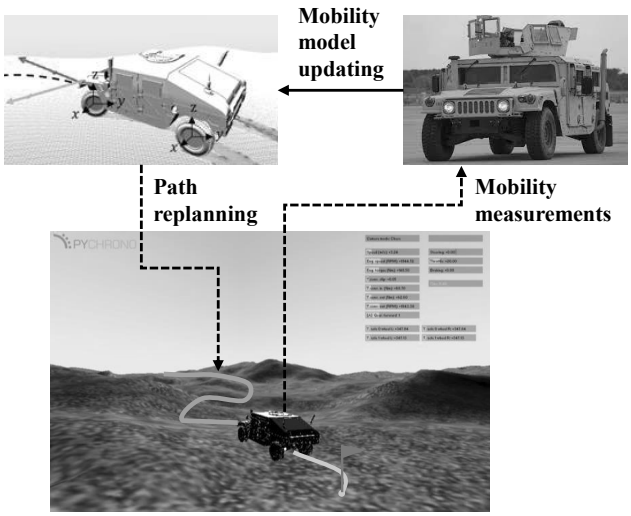


Fig. 8: Real-time mission planning of off-road autonomous ground vehicle (AGV) (Liu et al., 2021c,b).

(d) Predictive maintenance scheduling

Another area that may significantly benefit from digital twin technology is maintenance scheduling. In a traditional setting like a manufacturing plant, maintenance is typically performed in a reactive approach (after the machine has failed) or in a proactive approach (fixed interval repairs regardless of machine status). However, the reactive approach can be dangerous, as it allows machines to fully fail, which can harm nearby personnel. Similarly, the proactive approach of constantly servicing the machine on a fixed interval is wasteful because the machine may be healthy and does not require repair for some time. A real solution to this problem is individualized digital twin models which can predict a machine's future degradation trajectory and online optimize the maintenance schedule.

To effectively online optimize the maintenance of a unit, two key models are needed: (1) a prognostic model which can estimate the remaining useful lifetime of the unit, and (2) an optimization model which considers the RUL estimate from the prognostic model as well as the maintenance engineer's preferences before determining the optimal time to conduct maintenance. If the prognostic model is accurate, maintenance can be conducted "just-in-time," effectively maximizing the function of the unit while minimizing downtime and repair costs typically incurred using reactive or preventative maintenance approaches (Lee et al., 2013a,b).

The first key component in a digital twin for predictive maintenance scheduling is the prognostic model. Much research has already been done regarding prognostic methods for estimating RUL, and the major methods are discussed in detail in Sec. 5.2 of Part 1. The second key component is the optimization scheduling algorithm. Research in this area is focused on developing optimization methods to determine the best time to perform maintenance. In some cases, the main driving factor for optimizing the maintenance schedule is the long-term cost per unit of time (Grall et al., 2002; Bousdekis et al., 2019). To solve these maintenance optimization problems, a cost function is formulated which captures the costs associated with maintenance, and is minimized using

an applicable solver (Camci, 2009; Suresh and Kumarappan, 2013). Additionally, the cost functions can take into consideration the predicted remaining useful life of the unit at the time of maintenance. This is important because any remaining lifetime the unit had will discarded as soon as maintenance happens (Lei and Sandborn, 2018).

In other scenarios, the cost of maintenance is not the main consideration, and other objectives need to be prioritized. Take for example a major highway. The highway serves many people, and needs to be maintained on a regular basis. The optimal time to close a highway for construction would be when the weather is best and when traffic is the least (Sabatino et al., 2015; Allah Bukhsh et al., 2019). These objectives are harder to quantify and optimize together, so researchers have proposed creating utility functions, which map different objectives to the same range, so they can be more easily combined and optimized (Huber, 1974; Winterfeldt and Fischer, 1975). This method, referred to as multi-attribute utility theory, is a popular method for optimizing non-traditional objectives (Froger et al., 2016; Camci, 2015).

Digital twin is a promising technology that has the potential to online optimize the maintenance times of units operating in the field. The automatic data flow between the prognostic and optimization models in the digital twin enables optimization solutions which are specific to a single unit in the field, and take into consideration its current and future health status. These ideas are further investigated in the case study, where we show how one might build a battery digital twin to optimize the retirement of a battery cell from its first life use. Ultimately, the strongest aspect of a digital twin is its ability to optimize over dimensions of interest. In the future, we see predictive maintenance scheduling on a unit-by-unit basis as a core application of digital twins.

2.2.4 Summary of commonly used optimization methods in digital twins

Table 2 summarizes applications of different optimization methods in different dimensions of digital twins as discussed above. The advantages and disadvantages of different methods are also briefly summarized in this table. As illustrated in Table 2, evolutionary optimization methods, such as genetic algorithms, simulated annealing, etc., in general, can be applied to the dimensions of modeling (i.e., optimization of physical system modeling) and V2P connection (e.g., process control, mission planning, and maintenance scheduling) of a digital twin. The required number of function evaluations by the evolutionary optimization methods is usually very high. Therefore, in digital twin applications, evolutionary optimization methods are often used (1) when the objective function is very cheap to evaluate or (2) in conjunction with machine learning-based surrogate models to alleviate the required computational effort. In contrast, greedy algorithms are computationally much cheaper while it suffers from the downside of converging to local optima (or locally optimal solutions).

In recent years, reinforcement learning-based optimization methods have emerged as promising global optimization tools in many digital twin applications. This recent development was largely due to the significant increases in computational power and the many technological advances in deep learning (see Sec. 6.3 of our Part 1 paper). Nevertheless, reinforcement learning-based methods require high volumes of data for training and are complex to implement in practice. They are usually employed when an optimization problem is too complicated to be solved by conventional optimization methods.

Bayesian optimization based on Gaussian process regression is another widely used optimization method in digital twins (Snoek et al., 2012), even though it is not elaborated in detail in the discussions above. But Bayesian optimization may suffer from the curse of dimensionality because of the scalability limitation of Gaussian process regression. To enable real-time mission planning in V2P connection, a group of mission/path planning algorithms (e.g., A*, RRT*) have been developed in the past decades. Even though they may lead to sub-optimal solutions, they were shown to be computationally efficient and highly effective in finding near-optimal paths in real-time for practical applications.

Table 2: Summary of optimization methods in digital twins

Method		**Evolutionary optimization methods**	**Greedy algorithms**	**Reinforcement learning-based methods**	**Bayesian optimization**	**A*, RRT, RRT*, Dijkstra algorithm**
Online/offline		Online/Offline	Offline	Online/Offline	Online/Offline	Online
Application	Sensor placement optimization	✓	✓	✓	✓	✗
	Physical system modeling	✓	✗	✗	✓	✗
	Process control	✓	✓	✓	✓	✗
	Mission planning	✓	✗	✓	✗	✓
	Maintenance	✓	✓	✓	✗	✗
Pros		High-dimensional optimization; Global optimization;	Relatively low number of function evaluations	Global optimization with long-term rewards	Global optimization	Real-time global optimization
Cons		Large number of function evaluations	Local optimal	Data-intensive and high complexity	Curse of dimensionality	Suboptimal solution

3 Case study: a battery digital twin

Next, we use a battery digital twin case study to illustrate the implementation of a digital twin. Code and preprocessed data for generating all the results and figures presented in the case study are available on Github.

3.1 Background

Lithium-ion (Li-ion) batteries power phones, laptops, and more recently electric vehicles. They can supply a great deal of power for many hundreds of cycles, but eventually, they degrade and lose capacity due to irreversible internal electrochemical changes (LU LG et al., 2013; Liu et al., 2021a). It is therefore of great importance to accurately model and forecast future capacity degradation to estimate the lifetime cells operating in the field (Hu et al., 2020; Zhang and Lee, 2011). Accurate RUL predictions of cells operating in the field can empower manufacturers and operators to make informed decisions regarding the best time for cell maintenance or replacement/retirement. Recently, there has been much discussion around second life applications for retired cells as the global supply of used batteries is rapidly increasing (Engel et al., 2019; Casals et al., 2019). However, not all used batteries are the same because their usage during their first life application shapes their present health status and remaining capacity (Birkl et al., 2017). For example, an automaker cannot simply retire cells from vehicles using a fleet-wide kilometer threshold because each vehicle would have been subjected to a unique driving pattern that may have been more or less degrading to the cells' capacity. To make the most out of repurposed Li-ion cells, the time at which the cells are retired from their first life application needs to be optimized and determined online, in real-time, and on a per-cell or per-pack basis. One of the most promising solutions to the challenge of unit-specific real-time modeling is digital twin.

Unfortunately, many of the digital twin models reported in the literature are not truly a digital twin model, as they only operate in three or four of the five total dimensions outlined in Fig. 3 in Part 1 of this review (Thelen et al., 2022). Many

of the research papers we reviewed which claim to have created a digital twin have actually proposed a prognostic model for predicting the RUL of an engineered system. However, a prognostic model on its own is not fully a digital twin. In a prognostic model, sensor data from the physical system (PS) is used to update the digital state (P2V) of the digital system (DS). Then, the digital system is used to make a prediction about the future state of the physical system (V2P), i.e. the virtual prognostic model is used to predict the RUL of the physical system. The data flow then ends after RUL prediction of the physical system, thus not achieving all five of the data flow dimensions which characterize digital twin. This type of work neglects the most important dimension: optimization (OPT). To complete the data flow through all five dimensions, the RUL prediction from the prognostic model needs to be used as input into an optimization and control algorithm to manage some other system attributes.

In this case study, we develop and discuss a proof-of-concept battery digital twin which can be used to determine the optimal time to retire a Li-ion cell from its first life application in advance. The proposed five-dimensional digital twin model consists of two key pieces of software: 1) a particle filter battery prognostic model which can estimate the future capacity fade trajectory and predict cell RUL (PS, P2V, DS, V2P), and 2) a multi-attribute utility-based optimization model which takes into account user preference and the projected future capacity trajectory when determining the optimal time to retire a cell from its first life (OPT). The first piece of software, the particle filter battery prognostic model, is used to capture four of the five dimensions required for a digital twin: PS, P2V, DS, V2P. The last dimension, optimization (OPT) is captured by the second piece of software, the multi-attribute utility-based optimization model. The optimization algorithm interfaces with the physical system by triggering actions from operators, completing the data flow loop through all five dimensions of digital twin. The proposed digital twin model can operate online and is recursively updated each time a new battery capacity measurement becomes available. As a first attempt at a maintenance oriented digital twin for Li-ion batteries, it is our hope that the ideas discussed, and the code provided with this paper will help spark interest and innovation in this new area of research.

3.2 Methods

An overview of the proposed battery digital twin framework for optimizing the retirement of Li-ion cells from their first life application is shown in Fig. 9. The digital twin model, covers all five of the key data transfer and modeling dimensions outlined in Fig. 3 of Part 1 of this review (Thelen et al., 2022). In the offline phase, previously collected run-to-failure battery data is used to optimize the initial parameters of the particle filter. Then during the online phase, the particle filter is used to predict the future capacity fade trajectory of a cell. When the online cell's capacity measurements reach 95% their initial value, the first life retirement optimization code is triggered, and the predicted capacity trajectory from the particle filter is used to determine the optimal cycle to remove the cell from its first life application. This concept of first optimizing the particle filter offline and then using it on cells online is shown visually in Fig. 10. In the sections that follow, we provide more details about the individual algorithms used in the proposed digital twin framework.

3.2.1 Particle filter prognostic model

Li-ion batteries are a complex electrochemical system which degrade and lose capacity based on how they are used. Use conditions like the charging/discharging rate, depth of discharge, time, and ambient temperature all significantly affect Li-ion cell degradation and capacity loss (Birkl et al., 2017). As a result of the many aging factors, the shape of a cell's remaining capacity vs time plot can take on many forms (i.e. linear, exponential, power law, double exponential, all with respect to time). This can make modeling and forecasting of battery RUL difficult, especially when more than one capacity fade trend is present.

To accurately forecast remaining battery capacity and RUL, researchers have traditionally used empirical mathematical models to describe a cell's capacity loss as a function of time or cycle number. This approach, referred to as a model-based prognostics method (see Sec. 4.4 in Part 1), uses knowledge obtained from a small number of previous run-to-failure tests to inform

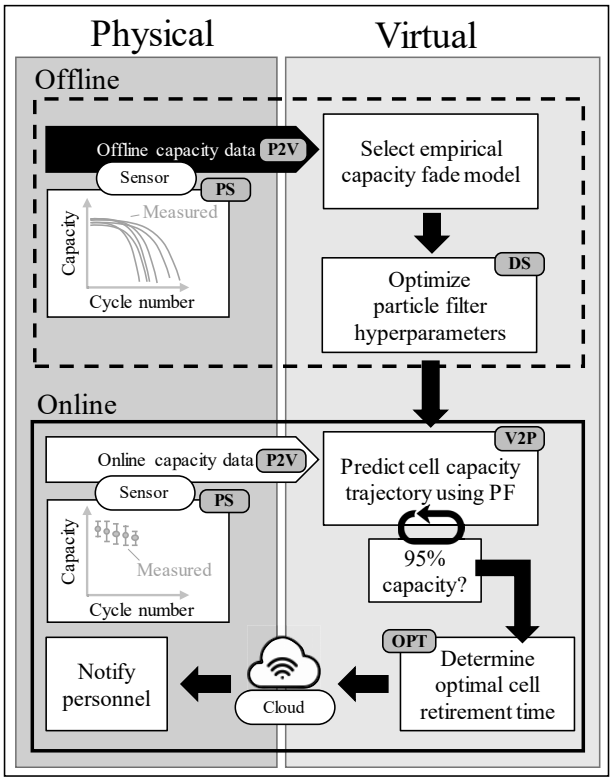


Fig. 9: Overview of the proposed battery digital twin framework.

researchers and practitioners about which mathematical model will most accurately capture the observed degradation trend. In the battery prognostics community, much work has been done to develop and test different empirical capacity degradation models for prognostics. Common mathematical functions used to model capacity loss as a function of time include linear (Honkura et al., 2011), exponential (Scott et al., 2005; Miao et al., 2013), and power law functions (Attia et al., 2020; Lui et al., 2021), or combinations of them (Hu et al., 2014; Wang et al., 2013; Hu et al., 2018; He et al., 2011; Diao et al., 2019).

The most widely used implementation of an empirical capacity fade model is to use a recursive filtering technique to estimate the values of the model parameters and extrapolate the prediction to the end-of-life threshold at each time step, (see Secs. 4.2.2 and 4.2.4 in Part 1). The extrapolated prediction and its associated uncertainty can be used to generate a probabilistic RUL distribution (see Sec. 2). Since many of the empirical capacity fade models are non-linear, Bayesian filtering methods like the extended Kalman filter and unscented Kalman filter are common choices (Plett, 2006, 2004). However, the assumption of Gaussian noise in the Kalman filter is rather restrictive and sometimes not desirable. A popular alternative is to use a particle filter for battery RUL prediction because it is generally a more flexible approach that can easily switch between multiple models and estimate a non-parametric distribution, which is generally more exact (Walker et al., 2015; He et al., 2011; Miao et al., 2013; Saha et al., 2008).

In this work, we have chosen to use a power law model with two parameters, as it can accurately model the diverse capacity fade trends observed in the open-source dataset used in this study. The formulation for the power law model is as follows

$$Q_k = 1 - ak^b, \tag{13}$$

where Q_k is the normalized capacity of the cell at cycle k, and a and b are the model parameters which affect the shape of the estimated capacity fade trend (Attia et al., 2020; Lui et al., 2021; Diao et al., 2019). To estimate the parameters a and b, we chose to use a particle filter because it generally converges quicker and also provides non-parametric RUL distribution predictions. In a particle filter algorithm, the power law model is the *measurement equation*, and the two *state transition* equations for a and b are as follows

$$a_{k+1} = a_k + w_{a,k+1}, \tag{14}$$

$$b_{k+1} = b_k + w_{b,k+1}, \tag{15}$$

where k denotes the cycle number and w is the process noise terms, specific to a and b. State estimation of model parameters and filtering are key components in a digital twin and have been discussed in detail in Secs. 4.2.2 and 4.2.4 in Part 1. Likewise, a detailed discussion regarding the mathematics for recursively updating a particle filter can be found in Appendix A of Part 1. Detailed pseudo code which accurately describes the steps one would take to implement a particle filter is presented in Fig. 28 of Part 1 and the working code used to implement the particle filter in this case study can be found on Github.

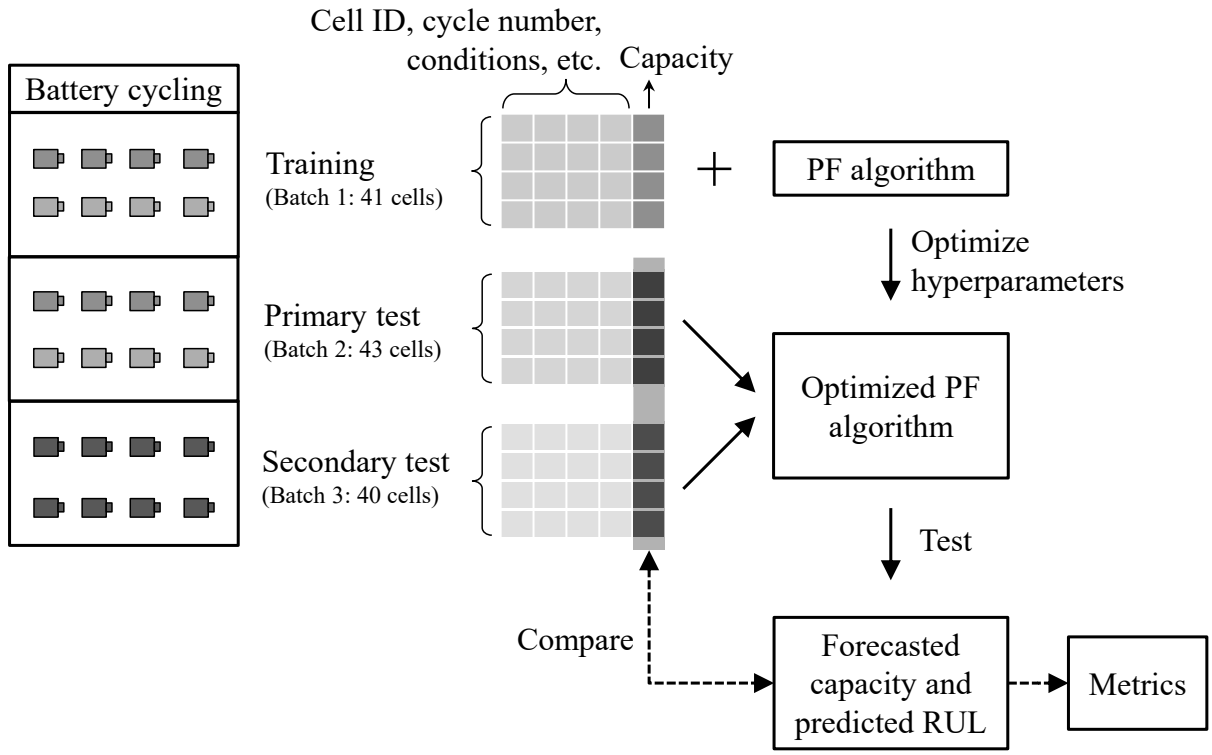


Fig. 10: Overview of the method used to offline optimize the particle filter ("PF" in the figure) and test it online.

3.2.2 Multi-attribute utility-based optimization model

Multi-attribute utility theory, sometimes referred to as the acronym MAUT, is a method which aims to consider multiple optimization objectives of different scales or magnitudes by defining unique utility functions which map the quantities of interest to a common range for easier evaluation and optimization (Huber, 1974; Winterfeldt and Fischer, 1975). Using this method, a unique utility function is defined for each value of interest (attributes) that is to be included in the optimization objective function. The utility function serves to scale the different attributes to a common range, typically [0,1]. If n attributes are mutually preferential and independent, the general multi-attribute utility function is defined as follows (Engel and Wellman, 2010)

$$\Lambda(\upsilon_1,\ \upsilon_2,\ \cdots,\ \upsilon_n) = \frac{1}{n}\sum_{i=1}^{n}\varphi_i(\upsilon_i), \qquad (16)$$

where υ_i denotes the value of the ith attribute, $\varphi_i(\cdot)$ is the utility function over the values of the ith attribute, and $\Lambda(\cdot)$ is the overall utility function. Then, the utility function is maximized by using any number of common optimization methods.

Multi-attribute utility theory has been widely applied in various maintenance scheduling applications because it is easy to incorporate non-traditional optimization objectives. For example, when servicing a large piece of equipment at a factory, it would be wise to incorporate knowledge of the company's sales when determining the optimal time to perform maintenance so that it has minimal impact on business. To achieve this, one can define a utility function which considers the company's sales, encoding their preference for maintenance times which will have a minimal impact on sales into the optimization problem. Similar work in the civil engineering field uses multi-attribute utility theory to schedule maintenance for bridges and other large structures (Sabatino et al., 2015; Allah Bukhsh et al., 2019). When the number of utility functions is large, the

design space increases drastically, and it becomes more important to select a proper solver, such as a genetic algorithm (Bukhsh et al., 2020; Garmabaki et al., 2016). In the energy field, multi-attribute utility theory has been applied to optimize the location of hybrid energy storage systems in the grid (Feng et al., 2018).

In this work, we take on the perspective of an EV manufacturer who has an agreement with a utility company to provide used EV batteries to them for grid-scale energy storage. We aim to optimize the time at which a Li-ion cell is retired from its first life application by considering two important attributes related to its first life use in an EV. The first attribute, the total discharge ampere-hours (Ah) before removal, encodes the preference that the cell should be used as much as possible in its first life. As the total Ah increases in value, so does the utility function. This utility function also encodes the preference to allow the customer to use the vehicle as much as possible and delay retirement as long as possible. The second attribute, the mean time between charges ($MTBC$), encodes the customers preference for a longer driving range on a single charge. This attribute correlates strongly with the health and remaining capacity of the cell, and conflicts with first attribute, total Ah. The mean time between charges will decrease as the cells ages, proportional to the capacity fade. By assuming that the two attributes are mutually preferential and independent, the two utility functions are combined using Eq. (17) as follows

$$\Lambda(Ah,\ MTBC) = \frac{1}{2}\varphi_1(Ah(x_c)) + \frac{1}{2}\varphi_2(MTBC(x_c)) \tag{17}$$

where x_c is the decision variable (i.e., cycle number), $\varphi_1(Ah)$ and $\varphi_2(MTBC)$ are respectively the utility function of the total discharge ampere-hours (i.e., Ah) and the mean time between charges (i.e., MTBC) which are explained in details in Sec. 3.3.3, and $\Lambda(Ah,\ MTBC)$ is the overall utility function to be maximized.

3.3 Results and discussion

3.3.1 Open-source battery dataset

The open-source battery dataset used in this case study was released by the group in Severson et al. (2019). The dataset consists of 124 lithium iron phosphate/graphite cells cycled with various two-step fast charging protocols. The discharge capacity fade curves for all the cells in the dataset are plotted in Fig. 11. The variance in the capacity fade trajectories and lifetimes of the cells is due to the different fast charging protocols used. All cells are discharged at a constant current of 4C, where C denotes the amperage required to charge a battery from 0-100% SOC in 1 hour. The large variance in the lifetime of the cells will provide a useful real-world assessment of the flexibility of the particle filter and optimization algorithms.

The dataset is divided into three groups, a *training* dataset, and a *primary* and *secondary* test dataset. In this case study, we will use the training dataset to roughly determine suitable prognostic model hyperparmeters before testing the digital twin model online using the two test datasets.

In both industry and academia, Li-ion cells are generally removed from their first life application when their remaining capacity approaches 80% its initial value. Then, in a second life application, the cell would be further used until its remaining capacity is much lower, perhaps less than 60%. We have therefore linearly extrapolated the last 30 capacity measurements of each cell well into the future, and set the end of second life limit at 50% the initial capacity so that we have a more realistic problem to solve.

3.3.2 Prognostic results

Particle filter tuning

An important step in building a prognostic model for online prediction is carefully selecting the initial hyperparameters. The particle filter model used in this study has five hyperparmeters that need to be initialized and tuned to maximize performance. The first parameter, the measurement noise, was set to 0.01 to represent approximately 1% error in the capacity measurements. The second and third parameters, the process noise terms for a and b from the empirical capacity fade model in Eqs. (14) and (15), were both set to 0.05 because it produced acceptable results and reasonably quick convergence. The last two parameters, the initial values of a and b, were set to the median values ($10^{-15.77}$, 5.45) as determined by fitting the empirical model to each of the cells in the training dataset.

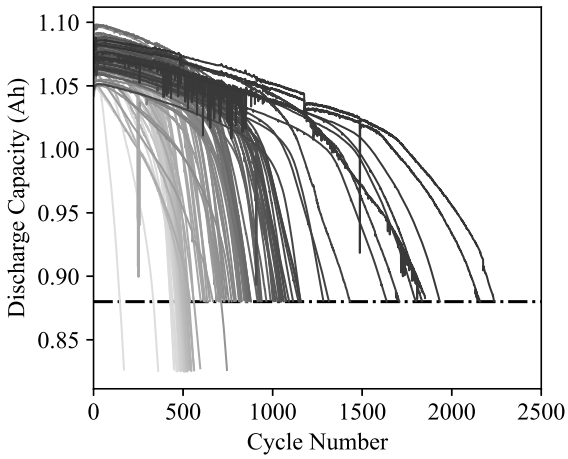


Fig. 11: MIT open-source battery dataset comprised of 124 LFP cells (Severson et al., 2019).

In a real world scenario where the particle filter is to be deployed online, it would be best to perform a more rigorous search for the optimal hyperparmaeters. However, since our goal is mainly to shed light on the many digital twin enabling technologies and provide insights on how they can be used in a digital twin framework like the one proposed in this case study, we did not take the time to further optimize the hyperparmeters of the particle filter.

RUL prediction results

First, we take a look at an example capacity projection from the particle filter. In Fig. 12, we plot the measured capacity and the median projected capacity from the particle filter. Using the many particles, we can project many capacity curves into the future and determine an empirical end-of-life distribution, also shown in the figure.

Figure 14 shows some example RUL predictions from the particle filter. Right away, it can be seen that the particle filter mostly underestimates the RUL of the cells. The underestimation is further exacerbated for cells which have extremely long lifetimes (> 2000 cycles). This prediction pattern largely arises because of two factors. First, since the capacity fade trajectories of the cells were extrapolated using a linear model fit to the last 30 measurements, the late life capacity loss follows a linear trend (see Sec. 3.3.1). Therefore, we cannot expect the power law capacity fade model (Eq. (13)) to accurately predict the long-term trend of the cells which have a linear degradation trajectory. This is visualized in Fig. 13 where we plot the measured capacity of a cell and the projections from the particle filter at different cycles.

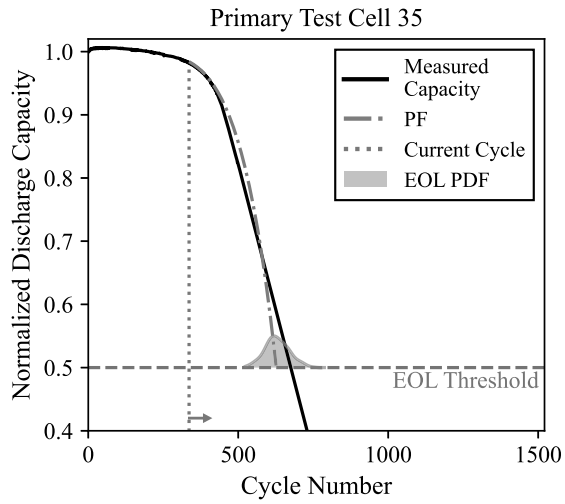


Fig. 12: Example particle filter capacity projection and empirical end-of-life distribution. EOL in the figure stands for "end of life".

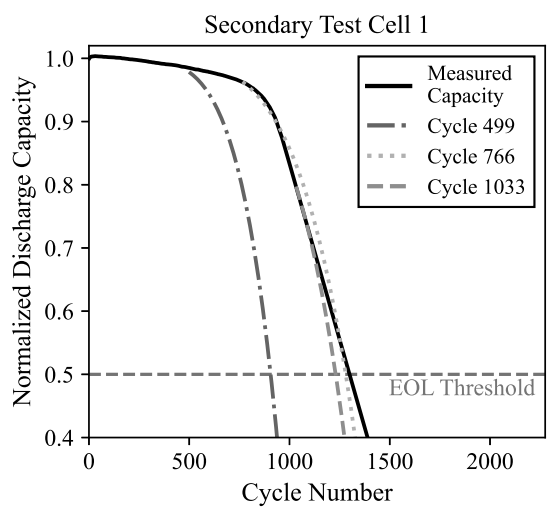


Fig. 13: Example particle filter capacity projections showing the underestimation of cell end of life. EOL in the figure stands for end of life.

Second, the initial values for the state parameters a and b were set to the median of the dataset, which encourages the initial lifetime prediction to be most accurate for cells of median lifetime.

The median lifetime of training dataset is 750 cycles while the median lifetime of the two test datasets is 1125 cycles. The larger lifetime of the cells in the test datasets is an example of distribution shift (see Sec. 4.3 in Part 1), causing the prognostic model to generally underestimate test cell RUL. Additionally, the measurement and process noise parameters are likely not properly tuned. The current settings for the noise parameters does not allow the particle filter enough variability with each update of the states, causing it to slowly converge to the true RUL. There may be some additional performance to gain by rigorously optimizing the hyperparameters of the particle filter.

3.3.3 Optimization results

Defining the utility functions

In order to optimize the time at which a cell is removed from its first life, we have to define the two utility functions which encode our preferences for how the cell is used in its first life. Drawing from previous literature, we aim to define two exponential utility functions. Used commonly in the finance industry, exponential utility functions are used to map real-world monetary gain to perceived value (utility). The general exponential utility function is defined as follows

$$\varphi(\upsilon) = 1 - e^{-\upsilon/R}, \tag{18}$$

where $\varphi(\cdot)$ is the perceived utility, υ is the monetary gain, and R is the risk tolerance. Additionally, the minimum monetary gain may be not always be zero, and the curve should be shifted to account for this. The exponential utility function can then be rewritten as follows

$$\varphi(\upsilon) = \varsigma - \tau e^{-\upsilon/R}, \tag{19}$$

where ς and τ are scaling parameters which can be used to set the upper and lower limits, and are defined as

$$\varsigma = \frac{e^{-L_u/R}}{e^{-L_u/R} - e^{-H_u/R}}, \tag{20}$$

$$\tau = \frac{1}{e^{-L_u/R} - e^{-H_u/R}}, \tag{21}$$

where R is the risk tolerance that defines the shape of the exponential curve, and L_u and H_u are the lower and upper limits to the utility function.

We adopt a similar approach for crafting the utility functions used in the battery digital twin. The first utility function maps the total Ah throughput from the cell to the range $[0, 1]$. This utility function is increasing, and takes the following form

$$\varphi_1(Ah(x_c)) = 1.0311 - 4.6212e^{-Ah(x_c)/200}, \tag{22}$$

where the parameters ς and τ, defined by the lower and upper bounds L_u and H_u and the shape parameter R are determined by examining the total Ah throughput for cells in the training dataset. From the electric vehicle manufacturer point of view, vehicles are typically rated to last a certain number of miles, years, or charge/discharge cycles. The bounds for this utility function are used to encode this requirement. For this case study, L_u, H_u, and R were set to 300 Ah, 1000 Ah, and 200, respectively. These bounds are rounded values which are close to the 5th and 95th percentiles of the training dataset. The utility function for the total Ah throughput is shown in Fig. 15.

The second utility function maps the mean time between charges ($MTBC$) to the range $[0, 1]$. This utility function is also increasing, but is conflicting with the first utility function because it is affected by cell aging and capacity loss. The function takes the following form

$$\begin{aligned} \varphi_2(MTBC(x_c)) = 1.0746 \\ -1292405e^{-MTBC(x_c)/0.015} \end{aligned} \tag{23}$$

where once again, the parameters ς and τ, defined by the lower and upper bounds L_u and H_u and the shape parameter R are determined by examining typical values for the cells in the training dataset. As the cell ages, the mean time between charges decreases, meaning that a customer driving the vehicle will have to recharge more frequently. This is not ideal, as customers prefer that the vehicle be as efficient as possible, having the largest possible range on a single charge. The parameters for this utility function are set in such a way to consider the diminishing utility a customer gets from a vehicle that needs to recharge too frequently. For this case study, L_u, H_u, and R were set to 0.21 h,

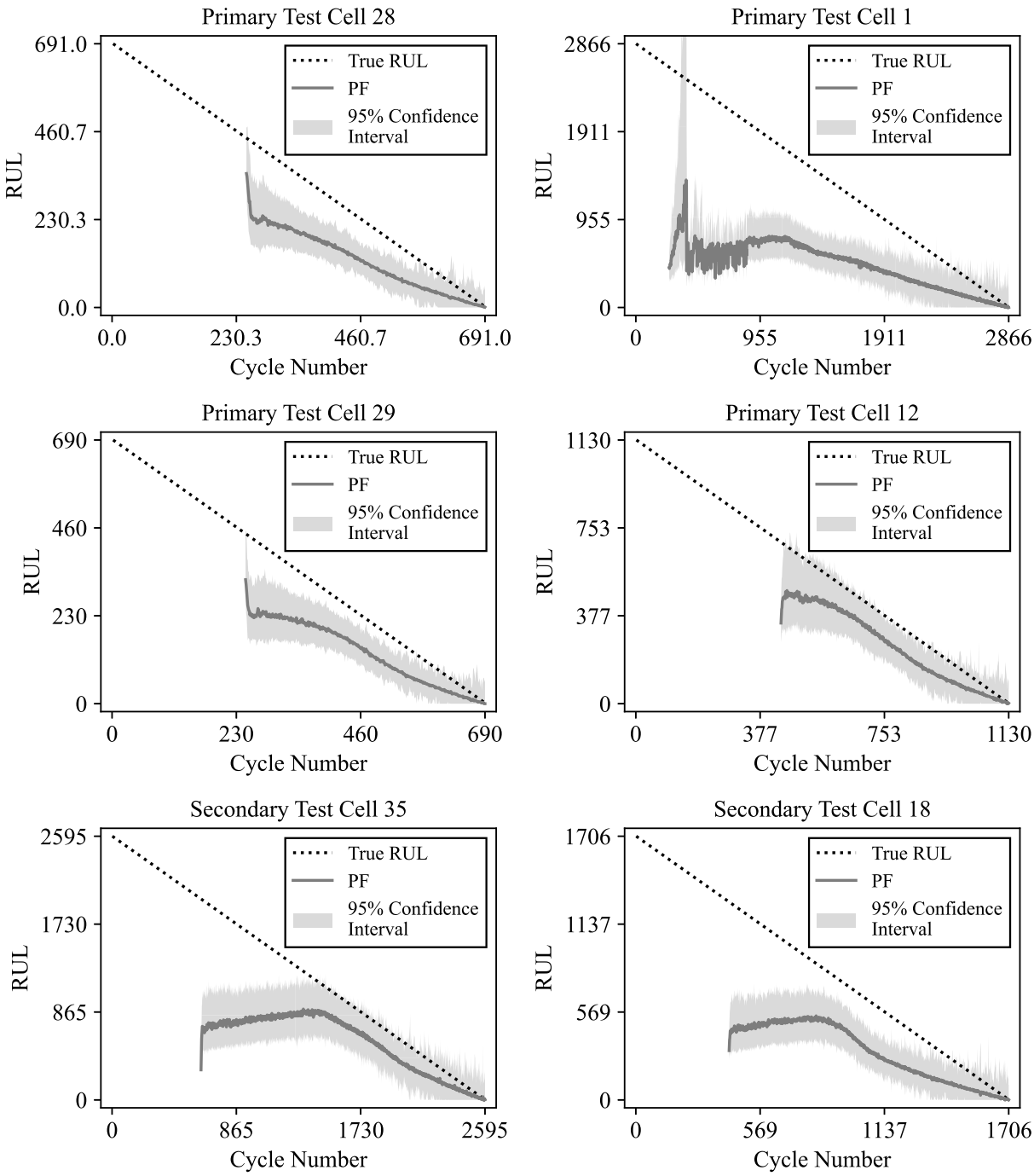


Fig. 14: Particle filter RUL predictions for six randomly selected cells from the two test datasets.

0.25 h, and 0.015, respectively. The utility function for the mean time between charges is shown in Fig. 16.

To determine the optimal replacement time, the two utility functions are combined into a single meta-utility function using an equal weight average (see Eq. (17)). Then, the combined utility function can be maximized by minimizing its negative using any off-the-shelf solver. However, the simple utility functions used in this case study are easy to evaluate, and the range of possible cycles at which the cell can be retired from its first life is finite, so we can simply evaluate the combined utility function at every possible cycle, and plot

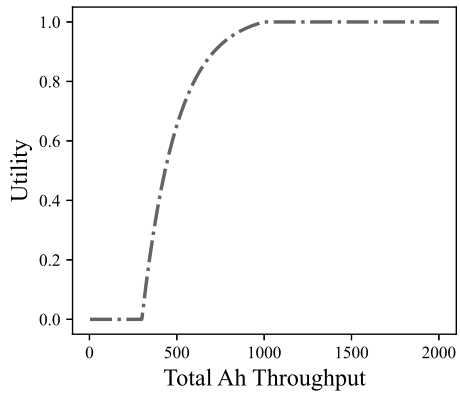


Fig. 15: Exponential utility function for total Ah throughput.

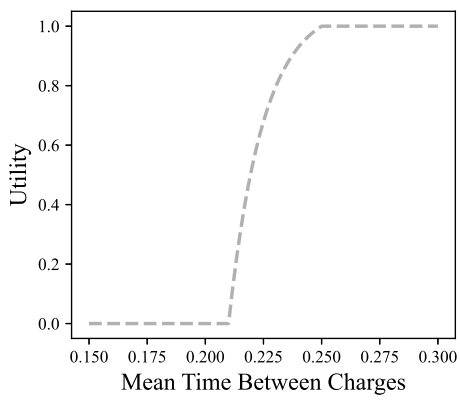


Fig. 16: Exponential utility function for the mean time between charges.

the utility curve to get a better idea of what is going on.

Analyzing the optimization results

Optimization results for three of the cells shown earlier (primary test cell 28 and secondary test cells 18 and 35) are in Figs. 17, 18 and 19. These cells were chosen because they have different lifetimes (average, long, and longest, respectively), and provide a good contrast in how the optimization method might perform in the field.

Looking at all three figures together, the first thing we observe is that the cells with longer lifetimes, cells 18 (Fig. 18) and 35 (Fig. 19) from the secondary test datset, have optimal retirements which are much closer to the current cycle. This is because the utility function for the total Ah throughput was defined around the mean values for *Ah* throughput of the training dataset, which were previously mentioned to have lower lifetimes than the two test datasets (median lifetimes for the training and test datasets are 750 and 1125 cycles respectively). By defining the utility function centered around the training cells with slightly lower lifetimes, it causes any cells with longer lifetimes, like cells 18 and 35, to have maximum *Ah* throughput utility for all possible retirement cycles. Since the *Ah* throughput utility is nearly constant over possible retirement cycles, this forces the optimal retirement decision to be made entirely based on the mean time between charges, which only decreases over the lifetime of the cell. Ultimately, this causes the optimal retirement to be as early as possible, to maximize the mean time between charges utility function.

For cells with lifetimes closer to the median of the training dataset, like cell 28 from the primary test dataset (Fig. 17), the utility functions work as intended, and the optimium retirement cycle strikes a perfect balance of the two utility functions.

In general, we found that the optimal time to retire the cell is most closely related to the observed rate of capacity loss (essentially the slope of the capacity vs cycle number curve). Typically, the optimal replacement time is shortly after the cell's knee point, which is the bend in the capacity fade curve where more rapid capacity loss begins to occur. Intuitively, this makes sense, as the total *Ah* utility can remain high, because the cell is still able to output a great deal of power even after its knee point, but the mean time between charges begins to rapidly increase, because the capacity is fading more quickly. As the capacity fades, the mean time between charges decreases, and the customer has to charge their vehicle more frequently, which is not desirable. In turn, the utility function for the mean time between charges decreases in value to reflect this.

Lastly, in testing, we observed that the optimal retirement time was not drastically affected by the particle filter's projected shape of the capacity fade trajectory. As long as the shape was fairly similar to that of the measured curve, the results were consistent. What we did notice, however, was that for cells with longer lifetimes, the particle filter would significantly underestimate the capacity trajectory, and the optimal replacement time would move closer to the current cycle to account for the perceived earlier EOL.

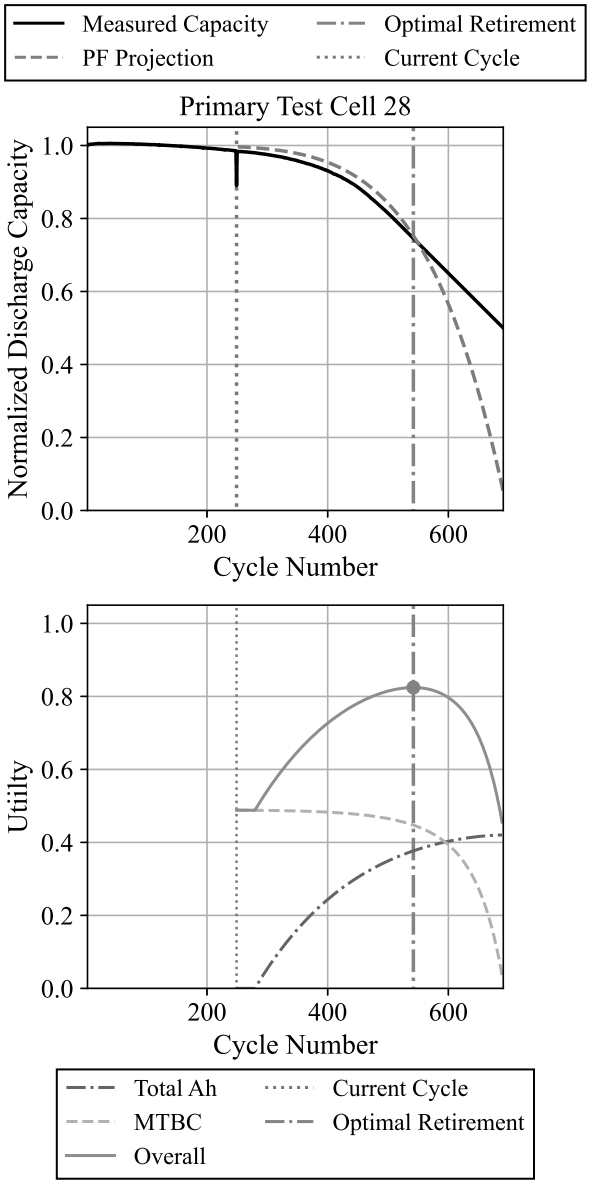


Fig. 17: Optimal cycle to retire primary test cell 28 from its first life application in an electric vehicle using the projected capacity from the particle filter.

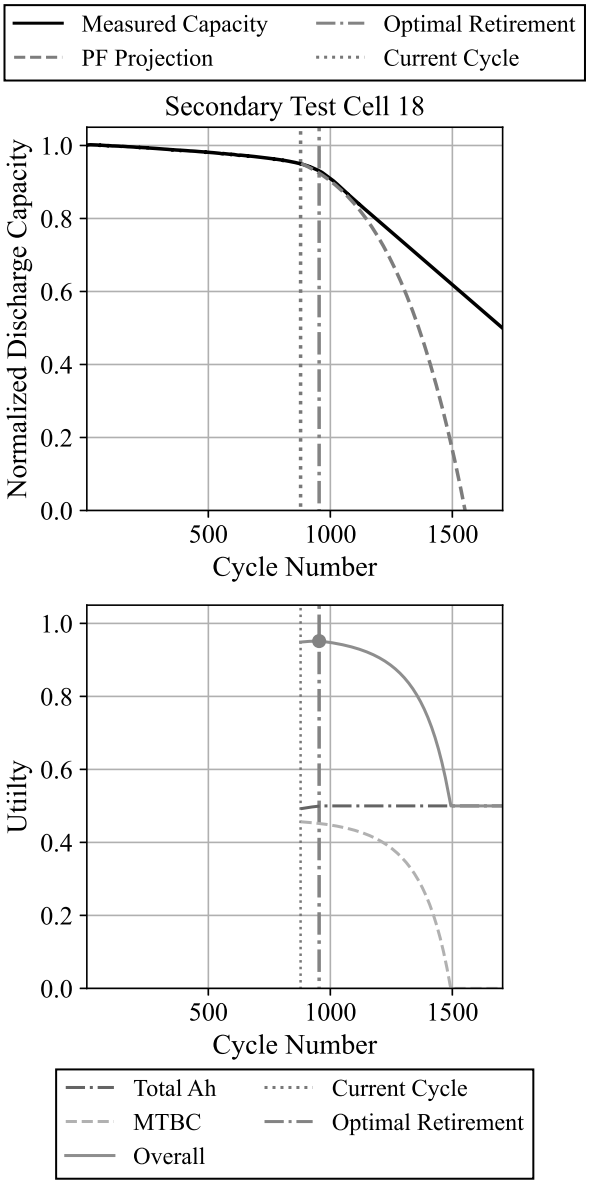


Fig. 18: Optimal cycle to retire secondary test cell 18 from its first life application in an electric vehicle using the projected capacity from the particle filter.

3.4 Case study conclusion and ideas for future research

Altogether, this case study demonstrated how one could create a battery digital twin for optimizing the retirement time of a Li-ion cell from its first life use. The success of the proposed digital twin framework relies on the integration of multiple individual pieces of software, which together, form an intelligent model capable of informing engineers and practitioners of optimal retirement times on a cell-by-cell basis.

The first key piece of software, the particle filter prognostic model, was found to accurately predict the RUL of cells with varying lifetimes. However, the particle filter used in this study was limited in that it only considered a single capacity fade model, the power law model. In future research, it would be worthwhile to investigate a multi-model particle filter, which actively switches

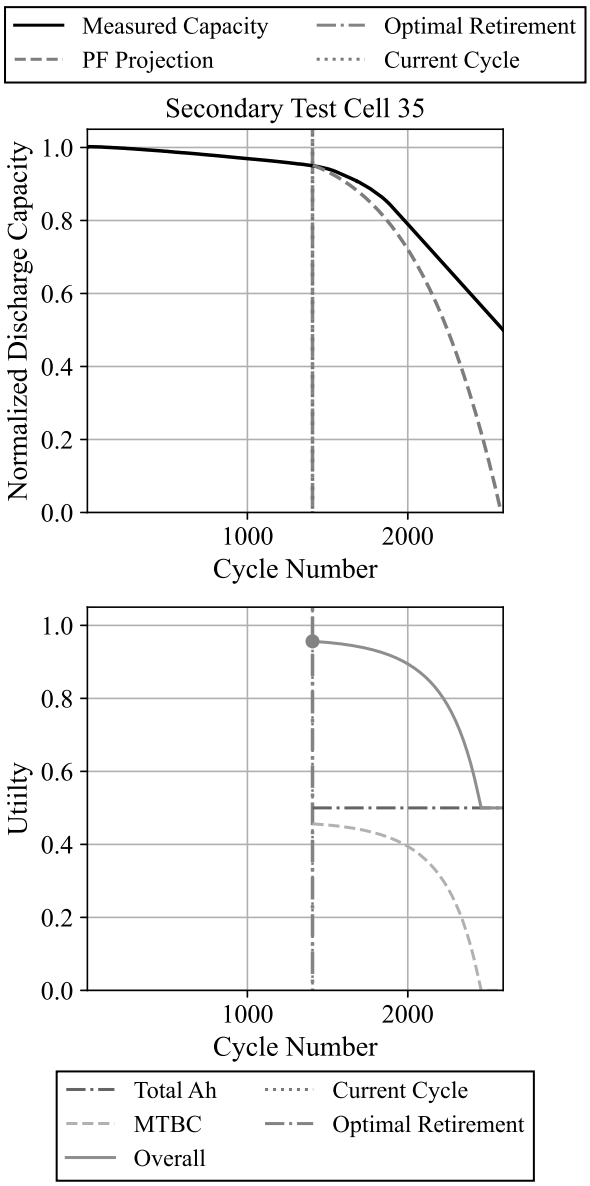


Fig. 19: Optimal cycle to retire secondary test cell 35 from its first life application in an electric vehicle using the projected capacity from the particle filter.

between different capacity fade models (Li et al., 2021b; Ye et al., 2018; Boers and Driessen, 2003).

The second key piece of software, the multi-attribute utility optimization model, was found to provide advanced notice of the optimal time to retire a cell from its first life application using the projected future capacity from the particle filter model. However, the optimization model struggled to provide meaningful optimal replacement predictions for cells which have much longer lifetimes. In future work, it would prove useful to investigate better methods of defining the many utility functions so that they can be applied to a wider range of cells. Additionally, this case study did not consider any preferences (attributes) relevant to the second life application of the Li-ion cell. The preferences of the second-life user were not considered for two main reasons. First, this was done to simplify the case study, as the goal was to demonstrate and explain in great detail the different software components that ultimately comprise the proposed digital twin model. Second, not considering the preferences of the second-life application simplified the optimization model, making it much easier to visualize and explain how it works. In reality, it is imperative to consider the preferences of all the involved parties when determining the optimal time to remove a Li-ion cell from its first life application. Both the electric vehicle manufacturer and second life grid-scale energy storage management company would have input on the attributes used in the optimization model, so that both parties' preferences are considered when determining the optimal time to switch from the first to second life application. However, we leave this to future work.

Lastly, the main strength of the particle filter is that it can output a non-parametric probabilistic prediction of a cell's capacity trajectory and end of life. However, this case study used a deterministic method to determine the optimal time to retire a cell from its first life application. In practice, it is desirable to present practitioners with confidence levels and intervals, instead of point estimates. Therefore, in future work, it would prove insightful to investigate probabilistic methods of evaluating multi-attribute utility functions, so that an optimal replacement interval can be defined, instead of a single-point prediction. Since the utility function approach to optimization is relatively lightweight, it may prove tractable to use sampling approaches to yield probabilistic results. Similar ideas are further discussed in Sec. 5.2 of Part 1 and Sec. 2.2.3, a probabilistic decision-making framework that considers model uncertainty (see Sec. 2.1) empowers maintenance engineers and practitioners with more information to make the bests decisions possible.

4 Demonstration and open source

4.1 Industry-scale demonstration of digital twin

Digital twins have become one of the key ingredients in improving the business outcomes across a wide range of industries. The global digital twin market is projected to grow from USD 3.1 billion in 2020 to USD 48.2 billion by 2026 (Markets and Markets, 2020). In many industries, digital twins have emerged as an integral part of the digitization efforts that started almost two decades ago. While the first wave of digitization focused on reducing physical databases and interactions, the second wave happening now is when the digital twin concept really kicked-off. Specifically, the second wave of digitization focuses on creating a virtual model encompassing a physical asset or process throughout its life cycle so as to analyze, predict and optimize the asset and improve business outcomes. Most use cases of digital twins in industry fall under the categories of predictive maintenance, asset life cycle management, process planning & optimization, product design, virtual prototyping and more (Sjarov et al., 2020). The benefit of incorporating digital twins is multi-fold when solving various engineering and business problems, such as reducing costs and risk, improving efficiency, security, reliability, resilience, and supporting decision-making (VanDerHorn and Mahadevan, 2021). Digital twins have been successfully applied to a broad range of industries, such as manufacturing, software, aerospace, agriculture, automobiles, healthcare, consumer goods, etc, and the usage of digital twins is still growing at a fast speed (Durão et al., 2018; Augustine, 2020).

In the aerospace industry, digital twin effort and conceptualization was pioneered and directed by NASA and the U.S. Air Force in early 2000s (see, for example, Glaessgen and Stargel (2012)). Since then, digital twin has been expanded to many areas, such as airframe, avionics, crack detection, feet level health management, etc. Digital twin has also been one of critical technologies that is listed as a part of future strategy for many leading companies, such as Boeing and Airbus (Aydemir et al., 2020). In GE electric, more than 500,000 digital twins have been deployed in various capacities (i.e., parts twins, products twins, process twins and system twins) across its production sites . In addition, GE aviation has reduced the amount of time for anomaly detections based on sensor data by 15-30 days. The reduction of anomaly detection time helps to achieve $44 million savings in the life cycle management for turbine blade engines, and $10 million savings in dynamic optimization to deliver optimal flight patterns and maintenance.

In the commercial sector, Boeing has reported 40% improvement in first-time quality of the parts and systems (Bellamy III, 2018). They have been creating and maintaining operational digital twins of components to track each individual component's unique characteristics and detect degradation rates. Similarly, in the defense sector, Boeing is using digital twins to predict and find possible fatigue maintenance hot spots in the F15 Eagle to reduce operation and maintenance costs. More recently, Boeing is planning to build an airplane in the metaverse using a digital twin (Careless, 2021).

In the automobile industry, digital twins have been playing an important role across all the stages of vehicle life cycle development, such as concept development, detailed design, and design verification (Sharma and George, 2018). In addition, digital twins have been used to improve the manufacturing and production in the automobile manufacturing plants. For example, BMW has recently used a virtual factory built upon digital twins and have produced 30% more efficient planning processes (Caulfield, 2022). In the healthcare sector, one of the main application areas lies in the development of technologies for monitoring, digital surgery, remote surgical assistance, drug development, medical treatment etc. using high-fidelity digital twin models of human bodies (Skardal et al., 2016; Bruynseels et al., 2018; Liu et al., 2019). A digital twin can also be used in building a virtual model of a physical city, also known as "smart city", to map the urban information system to into space and time. It has been used to solve a broad range of problems, such as water treatment, maintenance of facilities, and so on (Chen et al., 2018; Lehner and Dorffner, 2020; Lu et al., 2020).

Additionally, digital companies like Siemens, Ansys, IBM, and others have been playing a pivotal role in providing the infrastructure and platform to support the adoptions of digital twin by both small and large-scale industries. For example, Siemens has been one of the leading companies developing digital twin technologies for products, production and performance of various mechanics, electronics, software, or systems to minimize cost and optimize the requirements (Siemens Newsletter, 2020). Some of Siemen's product like Simatic Real-time Locating Systems (RTLS), Teamcenter X software, and UltraSoC (Markets and Markets, 2020) are representative examples of Siemen's digital twin product. Ansys has demonstrated applications of digital twins for Kärcher (Advantage Magazine, 2021), ABB (Boscaglia et al., 2019), ENGIE (Digital Engineering, 2021) and others. IBM has applied digital twins in different sectors like life cycle management, healthcare, predictive maintenance, manufacturing and more (IBM Newsletter, 2020; IBM Blog, 2019, 2020; IBM White Paper, 2020).

4.2 Open-source tools and datasets related to digital twins

We have also identified publicly available, free-to-use tools and datasets related to digital twins through our literature review. Tables 3 and 4 show, respectively, the publicly available tools and datasets. The roles of these tools and datasets in digital twins range from modeling, process mining, control, data streaming, and digital twin projects. These open-source tools and datasets could be helpful for production-level implementations of digital twins in both academia and industry. In what follows, we describe a few representative examples of open-source tools and datasets. Interested readers are again invited to consult Tables 3 and 4 for an overview of the tools and datasets we have identified.

Open-source software

- **Chrono:** Chrono is an open-source multiphysics modeling and simulation infrastructure capable of multibody dynamics simulation, finite element analysis, and fluid-solid interaction. Its core is a C++ object-oriented library called the Chrono::Engine, which can be embedded in third-party applications, as part of a large digital twin project. As noted on the PROJECTCHRONO web page, its multibody dynamics capability allows for vehicle dynamics simulations where wheeled vehicles may operate on soft, deformable soils. This effort started in 1998, and the current version, Chrono 7.0.1, was released in November 2021. It is rare to see such a long-lasting open-source effort, which has substantially pushed the boundary of open-source middleware and fostered an ecosystem of multiphysics software tools.
- **Digital twin repositories:** These open-source repositories are being managed by a non-profit organization called Digital Twin Consortium as their "open collaboration initiative". The repositories are hosted on GitHub (https://github.com/digitaltwinconsortium) and currently include several industry-relevant projects focused on manufacturing digitatlization (e.g., ManufacturingDTDLOntologies and EcolCafe-Industrie-4.0). The open-source nature of these code implementations, digital models, and training and guidance documents will drive development, interoperability, and adoption of digital twins in industry.
- **Commercial software tools:** Commercially available software tools enabling the realization of digital twins include (1) CAD software for geometric modeling, such as CATIA, Autodesk AutoCAD, Siemens NX, PTC Creo, SolidWorks, to name a few, (2) 3D manufacturing simulation software for production simulation and layout design, such as DELMIA and Tecnomatix, and (3) modeling and data analytics software for operations optimization and predictive maintenance, such as MATLAB (the Simulink modeling environment and Predictive Maintenance Toolbox), and (4) cloud-based IIoT platforms for predictive maintenance, including Siemens Mindsphere, Azure IoT Edge, Google Cloud, and Azure IoT Edge. Although not serving as digital twins on their own, these software tools have created success stories in realizing the digital twin concept in industry settings by working collaboratively with other enabling software tools. What is now needed are frameworks and software platforms that cohesively integrate modeling and simulation efforts of different forms and at various levels of detail to

Table 3: Publicly available tools related to digital twins (DS: digital system; P2V: physical to virtual; V2P: virtual to physical)

Name (link)	**Organization**	**Application(s)**	**Role(s) in digital twins**	**Brief description (paper ref)**
TE Code (Ricker)	UW	Manufacturing process control	DS, P2V, V2P	A simulation program for controlling chemical processes, known as the Tennessee Eastman Plant-wide Industrial Process Control Problem (Ricker and Lee (1995))
Chrono (Chrono; Tasora et al. (2015))	UW-Madison, UNIPR	Multiphysics simulation	DS	A modelling and simulation infrastructure for multiphysics simulations, including multibody simulation, finite element analysis, and fluid-solid interaction
ProM (Eindhoven; Van Dongen et al. (2005); Verbeek et al. (2010))	TU/e	Process Mining	DS, P2V, V2P	Business process mining software
PyMC3 (Team (a); Salvatier et al. (2016))	Collaborative	Multiple	P2V	A Python package for Bayesian modeling and probabilistic machine learning
QUESO (Team (b); Prudencio and Schulz (2012))	UT-Austin	Mechanical	P2V	A collection of algorithms and C++ classes that can be used for offline, probabilistic model updating
mFUSE (Team (2022);)	LANL/UCSD Engineering Institute	Mechanical, Civil, Structural	P2V	A set of MATLAB functions organized into modules according to the three primary stages of SHM
Digital Twin Open-Source Repository (Consortium)	Digital Twin Consortium	Manufacturing, mechanical, healthcare, etc.	DS, P2V, V2P	Open-source code implementations, collaborative documents for guidance and training, open-source models, or other assets related to digital twins
Dito (Foundation)	Eclipse Foundation	IoT middleware	P2V, V2P	IoT middleware, focusing on data modelling and IoT device connectivity
Unreal Engine (Epic Games)	Epic Games, Inc.	Transportation, mixed reality, etc.	DS, P2V	Real-time 3D modeling software

create complete digital twins in various engineering applications (Malik (2021); Malik and Brem (2021)). In this direction, some efforts have been undertaken in the commercial design software domain. For example, ESTECO modeFRONTIER is a commercially available design software tool that integrates and automates multiple CAD/CAE tools to create digital twins for engineering design. Another example is ....

Open-source data

- **Bosch Production Line Performance Dataset:** this dataset was introduced in 2016 as part of a data challenge whose goal was to reduce quality control escapes. Participants in the data challenge were to develop a model that could predict which parts will fail quality control by using extensive measurements taken at different locations along the assembly line. Large datasets like the one provided by Bosch represent the first steps toward realizing digital twin models in the production setting.
- **NASA Prognostics Data Repository:** NASA's Ames Research Center hosts 18 well-documented datasets, collectively called the Prognostics Data Repository, on their Prognostics Center of Excellence website. These datasets are mostly run-to-failure time series data collected from an engineered system or component starting at a healthy state and ending at a failed state. They were contributed by universities (e.g., ETH Zurich, the FEMTO-ST Institute, UC Berkeley, the University of Cincinnati, and Arizona State University), companies (e.g., PARC and Sentient Corporation), and government agencies (e.g., NASA Ames). This data repository intends to increase data availability for prognostic algorithm development and validation. Among the 18 datasets is (1) the first publicly available battery aging dataset Saha and Goebel (2007) that has led to the production of the first few battery prognostics papers Saha et al. (2008); Goebel et al. (2008); Saha et al. (2009) and (2) the first two publicly available engine degradation datasets Saxena and Goebel (2008a,b) that has led to the creation of numerous data-driven prognostics algorithms in the literature Wang et al. (2008); Heimes (2008); Hu et al. (2012); Moghaddass and Zuo (2014); Zhang et al. (2016); Zhao et al. (2017). This data repository is actively being expanded, and its impact on the field of predictive maintenance, which has been profound, will be increasingly significant.
- **Data Challenges in PHM and SHM:** In order to foster and encourage innovations in the PHM and SHM communities, data challenges/competitions have been organized in different PHM and SHM conferences. The released open-source datasets after the competitions have been extensively used in the literature as benchmark to test the performance of different diagnostics and prognostics algorithms (i.e., P2V connection). For instance, the datasets from the PHM Data Challenge of the PHM08 conference have been posted at NASA prognostics data repository, and have become one of the first publicly available datasets for failure prognostics. Since the data challenge in the PHM08 conference, many similar data challenges for fault diagnostics and failure prognostics have been organized by different PHM conferences including the annual PHM conference (PHM, 2022), the European PHM conference (PHME, 2022), and the Asia-Pacific PHM Conference (PHMAP, 2021). In the SHM community, the first data compitition was organized in 2020 by the Asia-Pacific Network of Centers for Research in Smart Structures Technology, Harbin Institute of Technology, the University of Illinois at Urbana-Champaign (ICSHM, 2020). In this competition, algorithms are compared in terms of accuracy for detecting fatigue crack based on images, anomaly of bridge using acceleration data, and damage of cables based on monitored cable tension data. Even though the addressed fundamental fault diagnostics and failure prognostics problems are similar, the data competition in the SHM community focuses more on fault diagnostics, anomaly detection, and prognostics of large civil infrastructures. The popularity of these data challenges shows the importance of common datasets for the development of new algorithms in establishing the P2V connection in digital twins.
- **Battery Archive:** This web-based platform, launched in 2020, is the first public platform that allows the battery community to visualize, analyze, and compare battery aging data across different organizations, studies, chemistries, and

aging conditions. The site development is led by Sandia National Laboratories (SNL), a U.S. Department of Energy (DOE) sponsored Federally Funded Research and Development Center. The Battery Archive platform is built based on an open-source platform called the Battery Life Cycle Framework De Angelis et al. (2021) and uses an open-source extract-transform-load engine called Redash. It has two unique properties: (1) all the battery datasets shared on the platform share a standard format, making it easy to compare performance across studies, battery chemistries, and aging conditions, and (2) the platform has an open-source backend, opening the possibilities of interoperating with existing software tools for more advanced analytics. Compared to the NASA Prognostics Data Repository, Battery Archive goes one step further by standardizing the data format and setting up open-source tools for data visualization and basic analysis. This further step is expected to have a long-last impact on future research studies surrounding battery degradation modeling. Further, we think Battery Archive represents an exemplary data-sharing effort that could be scaled up to benefit other digital twin applications.

- **Sandia Thermal Challenge Problem:** This thermal challenge problem was launched by SNL in 2006 and has been used extensively as a benchmarking dataset to evaluate the performance of methods and approaches for experimental validation of computer simulation models and quantification of model uncertainty (Hills et al., 2008). It is the first publicly available dataset to evaluate the performance of different UQ methods for model calibration and validation. The mathematical model of one-dimensional, linear heat conduction in a solid slab and the associated experimental data are provided by Dowding et al. (2008). The problem is concered about (1) evaluating whether the prediction accuracy of a calibrated/validated model meets a regulatory requirement and (2) quantifying the uncertainty in the accuracy assessment. In the past decades, solutions to this challenge problem (model validation approaches and results) have been developed using both Bayesian (e.g., the KOH framework discussed in Sec. 2.1.2) and non-Bayesian methods. The problem is formulated in a way that it is independent from specific applications and representative of the complexities of realistic physical systems.

More collaborative, open-source efforts are needed.

The industry-scale adoption of digital twins will be accelerated by increasing efforts in sharing tools (e.g., applications, libraries, and plug-ins), data, and best practices (e.g., tutorials, implementation guides, and training materials) between industry, academia, and government. Open-source efforts by academic researchers are often directed to create rigorous, application-agnostic solutions for specific technical components of digital twins (e.g., probabilistic model calibration, MPC). In contrast, open-source efforts by industrial practitioners tend to be focused on integrating these technical pieces into easy-to-use, well-documented software platforms thoughtfully customized to individual applications. These two groups of open-source efforts, representing two distinct ranges of technology readiness levels (TRL), are complementary. The cross-sharing of knowledge among these communities of interest will grow an open-source knowledge pool and promote the development of high quality commercial-grade software for digital twins. In this review, we want to call for the academic, industrial, and government communities to work together and share their software tools and data. Such collaborative, open-source efforts will go a long way in shaping the future of the digital twin field.

5 Perspectives on UQ and optimization for digital twins

5.1 Digital twins for structural life cycle management

The concept of digital twin for structural life cycle management is primarily defined by its usage objectives, and the properties required to perform those objectives. These digital representations or manifestations may be used for several such goals including (but not limited to) visualization/curation, structural state awareness quantification, and prediction of future structural performance (e.g., when critical limit states are expected to

Table 4: Publicly available datasets related to digital twin

Name (link)	**Organization**	**Application(s)**	**Role(s) in digital twins**	**Brief description (paper ref)**
NASA Prognostics Data Repository (PCoE)	NASA	Predictive maintenance	P2V, V2P, OPT	A collection of 18 prognostic datasets for prognostic algorithm development
Battery Archive (SNL)	DOE SNL	Battery degradation modeling	P2V, OPT	A web-based platform for visualizing, analyzing, and comparing battery aging data across different studies (De Angelis et al. (2021))
Thermal Challenge Problem (Hills et al. (2008))	DOE SNL	Model validation and UQ	P2V, OPT	A mathematical model and the solution of a one-dimensional, linear heat conduction in a solid slab. Experimental data related to the mathematical model are also provided (Dowding et al., 2008).
Kadi4Mat	KIT	Materials modeling	P2V, OPT	A web-based application facilitating easy and intuitive access, exchange, and integration of mostly battery materials data. https://demo-kadi4mat.iam-cms.kit.edu/
Bosch Production Line Performance Dataset	Kaggle	Production monitoring for quality control	P2V, V2P, OPT	https://www.kaggle.com/competitions/bosch-production-line-performance/data
3D Data Hack Dublin (Graham)	Dublin City Council etc.	3D modeling for urban planning, etc.	DS, P3V, V2P	3D datasets for creating a digital twin of the Docklands area in Dublin, Ireland White et al. (2021)

occur). Consequently, digital twins take on a variety of forms ranging from point clouds graphical representations to machine learners to physics-based simulators such as finite element models. Taken together, this collection of capabilities empowers the possibility of structural life cycle emulation. Initial designs (drawings, CAD, etc.) can be configured into baseline digital realizations associated with nominal expected performance properties (e.g., material properties, connectivity); these realizations may be updated by surveys (e.g., a UAV multi-mode lidar, thermal, visual data taken in the as-built state), and/or deployed in-situ data streams that come from a variety of IoT sources. Diagnostic analytic capability can be added by a variety of supervised or unsupervised learning techniques and physical knowledge of degradation or system state evolution. Stochastic load/demand and environmental models can be added to evolve the structure through a variety of operational scenarios, with uncertainty quantified by any of a variety of techniques, to predict future performance variables all the way to the limit states, thus completing the life cycle. This

"cradle-to-grave" capability can be further integrated with utility/cost models so that the consequences of all future states (or decisions made because of these states) may be inferred, leading to the ability to optimize the structure's life cycle management, including preventative maintenance, repair, etc. The digital twin would thus become the primary tool of the structural owner to manage the structure, leading to "smart structure"-informed decision-making that could potentially mitigate total structural asset management risk. This vision becomes more and more realizable as large-scale computing, data and predictive analytics, and data management continually empower its execution.

5.2 Digital twins for sustainability

As discussed in Sec. 1 of our Part 1 paper (Thelen et al., 2022), a physical system's life cycle can be divided into four distinct phases: development, manufacturing, service, and disposal. Most of the digital twin papers that we reviewed cover one or multiple of the first three life cycle phases. We did not find many studies that look at the disposal phase, where a physical system or product which has reached its end of life in the first life cycle could be reused, remanufactured, or recycled. Let us now look at two representative examples where digital twins have the potential to make an enormous and long-lasting impact at the disposal phase: (1) *remanufacturing* in the equipment manufacturing sector and (2) *battery repurposing* in the energy sector.

Remanufacturing is a process where used products/subassemblies are returned to an "as-new" condition (Ijomah et al., 2004) or even a "better-than-new" condition (Zhao et al., 2010) with significantly lower energy and materials consumption than original manufacturing (Li et al., 2021a). Remanufacturing has been gaining popularity in recent years, as it plays an important role in driving the global transition to a circular economy. Examples of physical systems that can be remanufactured when approaching the end of design life include a piece of industrial equipment on a production floor (Zacharaki et al., 2021) and a high-value component of an agricultural machine in the field (Li et al., 2021a). More research is needed to exploit the digital twin concept to promote wide-scale adoption of remanufacturing in the manufacturing industry. An interesting research topic along this line is the development of ML methods for non-destructive damage assessment and RUL prediction of cores (used products/-subassemblies returned for remanufacturing) on a remanufacturing floor. Although fault diagnostics and failure prognostics have found successes in industrial, structural, and energy applications, as discussed in Sec. 4.4 of our Part 1 paper, such success stories are rare in remanufacturing mainly due to difficulties in accessing historical data from returned products/subassemblies and in collecting faulty/run-to-failure data representative of returned products/subassemblies. Physics-informed ML approaches discussed in Sec. 3.4 of Part 1 can certainly be used to tackle these data challenges. A recent attempt was made to quantify and forecast accumulated fatigue damage in recycled materials (Hsu, 2021). This attempt can be classified as the ML-assisted approach (Approach 6) shown in Fig. 10 of Part 1. It is anticipated that remanufacturing will continue to grow in the manufacturing industry. We call for industry-relevant studies that investigate how to harness the power of digital twins and their building blocks, such as ML, DL, and IIoT, to build rapid diagnostic, prognostic, and decision support tools. Making these tools available to remanufacturers will lead to opportunities to reduce the cost of remanufacturing and increase the core reuse rate, eventually making remanufacturing more cost-competitive and sustainable.

The other example is repurposing used electric vehicle batteries for stationary storage applications (Zhu et al., 2021). To date, battery repurposing has been mostly for small-scale commercial applications, such as portable battery packs for camping and backpacking, waste disposal bins that use batteries to store solar energy, and off-grid lighting, as reported in Zhu et al. (2021), and small-scale recreational applications including recreational vehicles and golf cars. Building on the digital twin concept, two notable efforts are being pursued to support battery repurposing for larger-scale commercial applications.

- **Rapid diagnostics:** Similar to remanufacturing, battery repurposing typically requires rapid testing/grading methods to estimate the state of health of used electric vehicle batteries and predict their RUL. Estimating the state of

health at the end of the first life and predicting the RUL over the second life is essential to determining the economic and technical viability of repurposing any used battery pack/module for a specific second-life application. Now we can look back at the proposed five-dimensional digital twin model in Fig. 3 of the Part 1 paper and draw connections. The sensor data (P2V or physical-to-virtual) would be voltage and current measurements from a used battery pack/module (physical system or PS) during a rapid physical test. The digital state would be the instantaneous state of health or RUL. This digital state can be the output of an ML model (digital system or DS) built on a training dataset (P2V, physical-to-virtual) to estimate either the probabilities of predefined health grades (classification) or the probability distribution of the state of health or RUL (regression). The resulting estimate of the digital state (state of health or RUL) will enable rapid, near real-time assessment of the economic value (e.g., expected revenue from a simulated second-life scenario vs. associated costs) and technical feasibility (e.g., expected remaining lifetime vs. desirable lifetime, and safety risks) of repurposing the battery pack/module. The economic and technical assessment will provide the battery repurposing company with actionable information on optimally selecting (1) used battery modules with similar state of health levels to create a repurposed pack and (2) the second-life application that a repurposed pack has the best fit for (V2P or virtual-to-physical). Rapid diagnostics for battery repurposing is an exciting application of digital twins. It is probably the right time to invest serious and dedicated research efforts in building easy-to-use and accurate tools for rapid battery diagnostics. Such efforts could start with collecting and sharing performance and degradation data from second-life applications, which we had difficulties finding when populating the open-source datasets in Sec. 4.2.

- **Battery Passport:** An emerging and attractive alternative to rapid physical testing in a battery repurposing plant is the concept of Battery Passport, initiated by a public-private collaboration platform called the Global Battery Alliance (Alliance, 2020) in November 2020 and discussed extensively in the battery modeling literature (Bai et al., 2020; Zhu et al., 2021; Ayerbe et al., 2021). The Battery Passport can be viewed as a digital twin of a physical battery that comprises all relevant information about the battery from cradle (resource extraction and original manufacturing) to gate (repurposing or recycling). It gives each battery a unique identity consisting of information stored in the cloud or on a digital chip, for example, as part of a battery management system that manages the power and health of the battery. An achievable outcome of the Battery Passport is three-fold: (1) collecting state of health estimates and use conditions of a battery over its first life, (2) forecasting the battery's state of health by comparing the collected first-life information with the expected aging patterns for this battery chemistry, and (3) predicting the cycle life over the second life based on the state of health forecasting results. An obvious benefit would be the possibility of eliminating physical testing in a battery repurposing facility, substantially lowering the overall cost of repurposing. We believe that the value of digital twins to battery repurposing is far greater than what has been realized. As the Battery Passport continues to be adopted globally and across the entire battery value chain, we expect to see increasing adoption of digital twins to drive industry-scale battery repurposing that is both profitable and environmentally responsible.

5.3 UQ of digital twins

As discussed in earlier sections, the methodologies within UQ can be grouped within several broad categories: input uncertainty quantification, model calibration (parameter and discrepancy estimation), model verification (numerical error quantification), model validation, and uncertainty aggregation towards prediction uncertainty quantification. Several methods are available for each category, and an end-to-end UQ approach can be implemented for static models. On the other hand, a digital twin, which is an evolving model being updated when new data becomes available, presents new challenges, especially in model validation and decision making. The model updated at one time instant can only be validated with future data; thus validation is a lagging

indicator of digital twin model quality. However, decisions have to be made with the currently updated model, whose validation is pending based on future data.

Validation of a digital twin is not the same as validation of a general purpose prediction model. In the case of the latter, validation experiments can be performed to test the model's performance. This is not possible in the case of a digital twin, since the digital twin is tailored to a specific specimen or realization of a physical system, not to a population of specimens or realizations. Therefore validation data also has to be obtained from future use of a particular specimen; a specimen used in laboratory validation experiment is not the particular specimen for which the digital twin has been constructed. This is why validation is a lagging indicator of a digital twin's quality.

Since online decision-making for a future time instant is based on a current model which can only be validated in the future, there is additional uncertainty due to lack of validation at the current time instant. How to quantify this uncertainty is a new challenge. One option is to quantify the validation performance and the corresponding model uncertainty in previous time instants and extrapolate to future time instants; however, this raises another question regarding how to quantify the confidence in the extrapolation. In general, the uncertainty of the digital twin is expected to decrease with time, but this is only true when the conditions of the system operation remain similar. When aging (degrading) systems, such as aircraft and civil infrastructure, are used beyond their certified design life or operated outside the envelope of design conditions, newer methods are needed to quantify the uncertainty in the system behavior and its prediction by the digital twin.

6 Conclusion

In Part 1, we reviewed key digital twin modeling methods which connect the virtual and physical worlds. However, in practice, measurements of the physical world (P2V) and virtual model outputs (V2P) are never certain. In this paper, we reviewed different methods to quantify the uncertainty of machine learning models and the many digital twin modeling methods discussed in Part 1. Additionally, we discussed various optimization techniques used to improve accuracy, reduce uncertainty, and improve the usefulness of a digital twin model.

In addition to a comprehensive review of digital twin technology, we implemented and presented a battery digital twin model used for determining the optimal time to retire a battery cell from its first life application. The proposed battery digital twin serves as an example of how degradation modeling and optimization can be combined to produce a digital twin model which provides practitioners with actionable information regarding a cell's status in its overall life cycle.

The idea of a true digital twin which can accurately mirror all aspects of a physical asset is rapidly becoming a possibility thanks to the many state-of-the-art modeling techniques discussed in this two part review. In the future, digital twin modeling stands to revolutionize the way manufacturers and end users deliver, maintain, and retire high-value assets. Industry 4.0 is upon us, and big data for digital twin modeling is leading the way.

Acknowledgements

Adam Thelen and Chao Hu would like to thank the financial support from the U.S. National Science Foundation under Grant No. ECCS-2015710. Xiaoge Zhang is supported by a grant from the Research Committee of The Hong Kong Polytechnic University under project code 1-BE6V. Sankaran Mahadevan acknowledges the support of the National Institute of Science and Technology. Michael D. Todd and Zhen Hu received financial support from the U.S. Army Corps of Engineers through the U.S. Army Engineer Research and Development Center Research Cooperative Agreement W912HZ-17-2-0024.

Competing Interests

The authors have no relevant financial or non-financial conflicts of interest to disclose.

Replication of results

The Python code and preprocessed dataset used for the battery case study are available for download on Github.

Authors' contributions

All authors read and approved the final manuscript. Hu, C. and Hu, Z. devised the original concept of the review paper. Hu, Z, Thelen, A., and Zhang, X. were responsible for the literature review. Thelen, A. was responsible for geometric modeling. Hu, C., Thelen, A., and Hu, Z. were responsible for physics-based modeling. Hu, Z. was responsible for data-driven modeling. Hu, C. and Hu, Z. were responsible for physics-informed ML. Zhang, X. was responsible for system modeling. Hu, C., and Hu, Z., were responsible for probabilistic model updating. Zhang, X. was responsible for ML model updating. Hu, C. and Hu., Z were responsible for fault diagnostics, failure prognostics, and predictive maintenance. Lu, Y. was responsible for MPC. Fink, O. was responsible for federated learning and domain adaptation. Zhang, X., Hu, Z., and Fink, O. were responsible for deep reinforcement learning. Hu, C. was responsible for UQ of ML models. Hu, Z. was responsible for UQ of dynamic system models, optimization for sensor placement, and optimization for physical system modeling. Lu, Y. was responsible for the optimization of additive manufacturing processes. Zhang, X. and Hu, Z. were responsible for real-time mission planning. Thelen, A. and Hu, C. were responsible for the case study and predictive maintenance scheduling. Hu, C. was responsible for open-source software and data. Ghosh, S. was responsible for the industry demonstration. Hu, C., Todd, M., and Mahadevan, S. were responsible for perspectives. All authors participated in manuscript writing, review, editing, and comment.

References

M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

M. A. Adnan, M. A. Razzaque, I. Ahmed, and I. F. Isnin. Bio-mimic optimization strategies in wireless sensor networks: A survey. *Sensors*, 14 (1):299–345, 2013.

A. Advantage Magazine. Kärcher cleans up with ansys twin builder, Feb 2021. URL https://www.ansys.com/advantage-magazine/volume-xv-issue-2-2021/karcher-cleans-up-with-ansys-twin-builder.

Z. Allah Bukhsh, I. Stipanovic, G. Klanker, A. O'Connor, and A. G. Doree. Network level bridges maintenance planning using multi-attribute utility theory. *Structure and Infrastructure Engineering*, 15(7):872–885, 2019.

R. J. Allemang. The modal assurance criterion–twenty years of use and abuse. *Sound and Vibration*, 37(8):14–23, 2003.

G. B. Alliance. The global battery alliance battery passport: giving an identity to the ev's most important component. *Glob. Batter. Alliance*, 2020.

M. A. Alsheikh, D. T. Hoang, D. Niyato, H.-P. Tan, and S. Lin. Markov decision processes with applications in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, 17(3):1239–1267, 2015.

H. An, B. D. Youn, and H. S. Kim. A methodology for sensor number and placement optimization for vibration-based damage detection of composite structures under model uncertainty. *Composite Structures*, 279:114863, 2022a.

H. An, B. D. Youn, and H. S. Kim. Optimal sensor placement considering both sensor faults under uncertainty and sensor clustering for vibration-based damage detection. *Structural and Multidisciplinary Optimization*, 65(3): 1–32, 2022b.

M. Anand, Z. Ives, and I. Lee. Quantifying eavesdropping vulnerability in sensor networks. In *Proceedings of the 2nd International Workshop on Data Management for Sensor Networks*, pages 3–9, 2005.

C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1):5–43, 2003.

D. Ao, Z. Hu, and S. Mahadevan. Design of validation experiments for life prediction models. *Reliability Engineering & System Safety*, 165: 22–33, 2017a.

D. Ao, Z. Hu, and S. Mahadevan. Dynamics model validation using time-domain metrics. *Journal of Verification, Validation and Uncertainty Quantification*, 2(1), 2017b.

ARC. Automotive research center at the university of michigan. https://arc.engin.umich.edu/, 2022. Last Accessed: 2022-05-06.

P. D. Arendt, D. W. Apley, and W. Chen. Quantification of model uncertainty: Calibration,

model discrepancy, and identifiability. 2012.

R. Asorey-Cacheda, A.-J. Garcia-Sanchez, F. García-Sánchez, and J. García-Haro. A survey on non-linear optimization problems in wireless sensor networks. *Journal of Network and Computer Applications*, 82:1–20, 2017.

R. Astroza, A. Alessandri, and J. P. Conte. A dual adaptive filtering approach for nonlinear finite element model updating accounting for modeling uncertainty. *Mechanical Systems and Signal Processing*, 115:782–800, 2019.

P. M. Attia, W. C. Chueh, and S. J. Harris. Revisiting the t0. 5 dependence of sei growth. *Journal of the Electrochemical Society*, 167(9):090535, 2020.

P. Augustine. The industry use cases for the digital twin idea. In *Advances in Computers*, volume 117, pages 79–105. Elsevier, 2020.

H. Aydemir, U. Zengin, and U. Durak. The digital twin paradigm for aircraft review and outlook. In *AIAA Scitech 2020 Forum*, page 0553, 2020.

E. Ayerbe, M. Berecibar, S. Clark, A. A. Franco, and J. Ruhland. Digitalization of battery manufacturing: Current status, challenges, and opportunities. *Advanced Energy Materials*, page 2102696, 2021.

Y. Bai, N. Muralidharan, Y.-K. Sun, S. Passerini, M. S. Whittingham, and I. Belharouak. Energy and environmental aspects in recycling lithium-ion batteries: Concept of battery identity global passport. *Materials Today*, 41:304–315, 2020.

N. Bao and C. Wang. A monte carlo simulation based inverse propagation method for stochastic model updating. *Mechanical Systems and Signal Processing*, 60:928–944, 2015.

M. Barbehenn. A note on the complexity of dijkstra's algorithm for graphs with weighted vertices. *IEEE Transactions on Computers*, 47 (2):263, 1998.

V. Barzegar, S. Laflamme, C. Hu, and J. Dodson. Ensemble of recurrent neural networks with long short-term memory cells for high-rate structural health monitoring. *Mechanical Systems and Signal Processing*, 164:108201, 2022.

S. Basagni, L. Bölöni, P. Gjanci, C. Petrioli, C. A. Phillips, and D. Turgut. Maximizing the value of sensed information in underwater wireless sensor networks via an autonomous underwater vehicle. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 988–996. IEEE, 2014.

J. L. Beck and L. S. Katafygiotis. Updating models and their uncertainties. i: Bayesian statistical framework. *Journal of Engineering Mechanics*, 124(4):455–461, 1998.

I. Behmanesh, B. Moaveni, and C. Papadimitriou. Probabilistic damage identification of a designed 9-story building using modal data in the presence of modeling errors. *Engineering Structures*, 131:542–552, 2017.

W. Bellamy III. Boeing ceo talks 'digital twin' era of aviation. https://www.aviationtoday.com/2018/09/14/boeing-ceo-talks-digital-twin-era-aviation/, Sep 2018.

L. Bing, Z. Meilin, and X. Kai. A practical engineering method for fuzzy reliability analysis of mechanical structures. *Reliability Engineering & System Safety*, 67(3):311–315, 2000.

C. R. Birkl, M. R. Roberts, E. McTurk, P. G. Bruce, and D. A. Howey. Degradation diagnostics for lithium ion cells. *Journal of Power Sources*, 341:373–386, 2017.

C. Bisdikian, L. M. Kaplan, and M. B. Srivastava. On the quality and value of information in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 9(4):1–26, 2013.

B. Błachowski, A. Świercz, M. Ostrowski, P. Tauzowski, P. Olaszek, and Ł. Jankowski. Convex relaxation for efficient sensor layout optimization in large-scale structures subjected to moving loads. *Computer-Aided Civil and Infrastructure Engineering*, 35(10):1085–1100, 2020.

Y. Boers and J. N. Driessen. Interacting multiple model particle filter. *IEE Proceedings-Radar, Sonar and Navigation*, 150(5):344–349, 2003.

L. Boscaglia, F. Bonsanto, A. Boglietti, S. Nategh, and C. Scema. Conjugate heat transfer and cfd modeling of self-ventilated traction motors. In *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 3103–3109. IEEE, 2019.

A. Bousdekis, K. Lepenioti, D. Apostolou, and G. Mentzas. Decision making in predictive maintenance: literature review and research agenda for industry 4.0. *IFAC-PapersOnLine*, 52(13):607–612, 2019.

K. Bruynseels, F. Santoni de Sio, and J. Van den Hoven. Digital twins in health care: ethical implications of an emerging engineering paradigm. *Frontiers in Genetics*, page 31, 2018.

Z. A. Bukhsh, I. Stipanovic, and A. G. Doree. Multi-year maintenance planning framework using multi-attribute utility theory and genetic algorithms. *European Transport Research Review*, 12(1):1–13, 2020.

J. A. Burns, E. M. Cliff, and K. Farlow. Parameter estimation and model discrepancy in control systems with delays. *IFAC Proceedings Volumes*, 47(3):11679–11684, 2014.

J. A. Burns, E. M. Cliff, and T. L. Herdman. Identification of dynamical systems with structured uncertainty. *Inverse Problems in Science and Engineering*, 26(2):280–321, 2018.

F. Camci. System maintenance scheduling with prognostics information using genetic algorithm. *IEEE Transactions on reliability*, 58(3): 539–552, 2009.

F. Camci. Maintenance scheduling of geographically distributed assets with prognostics information. *European Journal of Operational Research*, 245(2):506–516, 2015.

S. Cantero-Chinchilla, J. Chiachío, M. Chiachío, D. Chronopoulos, and A. Jones. Optimal sensor configuration for ultrasonic guided-wave inspection based on value of information. *Mechanical Systems and Signal Processing*, 135:106377, 2020.

J. Careless. Digital twinning: The latest on virtual models. https://www.aerospacetechreview.com/digital-twinning-the-latest-on-virtual-models/, Aug 2021.

T. G. Carne and C. R. Dohrmann. A modal test design strategy for model correlation. Technical report, Sandia National Labs., Albuquerque, NM (United States), 1994.

L. C. Casals, B. A. García, and C. Canal. Second life batteries lifespan: Rest of useful life and environmental analysis. *Journal of environmental management*, 232:354–363, 2019.

B. Caulfield. Nvidia, bmw blend reality, virtual worlds to demonstrate factory of the future. https://blogs.nvidia.com/blog/2021/04/13/nvidia-bmw-factory-future/, Apr 2022.

S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.

M. Chadha, Z. Hu, and M. D. Todd. An alternative quantification of the value of information in structural health monitoring. *Structural Health Monitoring*, page 14759217211028439, 2021.

G.-S. Chen, R. J. Bruno, and M. Salama. Optimal placement of active/passive members in truss structures using simulated annealing. *AIAA Journal*, 29(8):1327–1334, 1991.

X. Chen, E. Kang, S. Shiraishi, V. M. Preciado, and Z. Jiang. Digital behavioral twins for safe connected cars. In *Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, pages 144–153, 2018.

P. Chrono. Chrono. https://projectchrono.org/. Last Accessed: 2022-04-17.

D. T. Consortium. Digital twin open-source repository. https://www.digitaltwinconsortium.org/initiatives/open-source.htm/. Last accessed on 2022-04-17.

V. De Angelis, Y. Preger, and B. R. Chalamala. Battery lifecycle framework: a flexible repository and visualization tool for battery data from materials development to field implementation. 2021.

A. Der Kiureghian and O. Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31 (2):105–112, 2009.

W. Diao, S. Saxena, and M. Pecht. Accelerated cycle life testing and capacity degradation modeling of licoo2-graphite cells. *Journal of Power Sources*, 435:226830, 2019.

M. Digital Engineering. Engie lab crigen and ansys accelerate zero carbon energy, Mar 2021. URL https://www.digitalengineering247.com/article/engie-lab-crigen-and-ansys-accelerate-zero-carbon-energy,

J. Dodson, A. Downey, S. Laflamme, M. D. Todd, A. G. Moura, Y. Wang, Z. Mao, P. Avitabile, and E. Blasch. High-rate structural health monitoring and prognostics: An overview. *Data Science in Engineering, Volume 9*, pages 213–217, 2022.

K. J. Dowding, M. Pilch, and R. G. Hills. Formulation of the thermal problem. *Computer Methods in Applied Mechanics and Engineering*, 197(29-32):2385–2389, 2008.

A. Downey, C. Hu, and S. Laflamme. Optimal sensor placement within a hybrid dense sensor network using an adaptive genetic algorithm with learning gene pool. *Structural Health Monitoring*, 17(3):450–460, 2018.

F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica. Path planning with modified a star algorithm for a mobile

robot. *Procedia Engineering*, 96:59–69, 2014.

L. F. Durão, S. Haag, R. Anderl, K. Schützer, and E. Zancul. Digital twin requirements in the context of industry 4.0. In *IFIP international conference on product lifecycle management*, pages 204–214. Springer, 2018.

N. Ehsani and A. Afshar. Optimization of contaminant sensor placement in water distribution networks: multi-objective approach. In *Water Distribution Systems Analysis 2010*, pages 338–346. 2010.

Eindhoven. Prom tools, eindhoven university of technology. http://www.promtools.org/doku.php/. Last accessed on 2022-04-17.

H. Engel, P. Hertzke, and G. Siccardo. Second-life ev batteries: The newest value pool in energy storage. *McKinsey & Company*, 2019.

Y. Engel and M. P. Wellman. Multiattribute auctions based on generalized additive independence. *Journal of Artificial Intelligence Research*, 37:479–525, 2010.

I. Epic Games. Unreal engine 5. http://www.unrealengine.com/en-US/unreal-engine-5. Last Accessed: 2022-05-27.

A. T. Eshghi, S. Lee, H. Jung, and P. Wang. Design of structural monitoring sensor network using surrogate modeling of stochastic sensor signal. *Mechanical Systems and Signal Processing*, 133:106280, 2019.

X. Feng, J. Gu, and X. Guan. Optimal allocation of hybrid energy storage for microgrids based on multi-attribute utility theory. *Journal of Modern Power Systems and Clean Energy*, 6(1): 107–117, 2018. doi: 10.1007/s40565-017-0310-3.

S. Ferson, W. L. Oberkampf, and L. Ginzburg. Model validation and predictive capability for the thermal challenge problem. *Computer Methods in Applied Mechanics and Engineering*, 197(29-32):2408–2430, 2008.

E. B. Flynn and M. D. Todd. A bayesian approach to optimal sensor placement for structural health monitoring with application to active sensing. *Mechanical Systems and Signal Processing*, 24(4):891–903, 2010.

E. Foundation. Eclipse ditto: open-source framework for digital twins in the iot. https://www.eclipse.org/ditto/. Last accessed on 2022-04-23.

W. Frazier. Metal additive manufacturing: A review. *Journal of Materials Engineering and Performance*, 23:1917–1928, 2014.

A. Froger, M. Gendreau, J. E. Mendoza, E. Pinson, and L.-M. Rousseau. Maintenance scheduling in the electricity industry: A literature review. *European Journal of Operational Research*, 251(3):695–706, 2016.

Y. Gal and Z. Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016a.

Y. Gal and Z. Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. *Advances in neural information processing systems*, 29, 2016b.

J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot. Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2997–3004. IEEE, 2014.

W. Gao, C. Song, and F. Tin-Loi. Probabilistic interval analysis for structures with uncertainty. *Structural Safety*, 32(3):191–199, 2010.

A. Garmabaki, A. Ahmadi, and M. Ahmadi. Maintenance optimization using multi-attribute utility theory. In *Current trends in reliability, availability, maintainability and safety*, pages 13–25. Springer, 2016.

J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*, 2021.

Github. Link to github repository where the preprocessed data and python scripts used to generate all the results and figures in the case study section reside. https://github.com/acthelen/battery_digital_twin.

E. Glaessgen and D. Stargel. The digital twin paradigm for future nasa and us air force vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, page 1818, 2012.

K. Goebel, B. Saha, A. Saxena, J. R. Celaya, and J. P. Christophersen. Prognostics in battery health management. *IEEE instrumentation & measurement magazine*, 11(4):33–40, 2008.

G. F. Gomes, F. A. de Almeida, P. da Silva Lopes Alexandrino, S. S. da Cunha, B. S. de Sousa, and A. C. Ancelotti. A multiobjective sensor placement optimization for shm systems considering fisher information matrix and mode shape interpolation. *Engineering with Computers*, 35(2):519–535, 2019.

N. Graham. 3d data model resources for dublin docklands sdz. https://data.smartdublin.ie/dataset/3d-data-hack-dublin-resources/. Last accessed on 2022-04-23.

A. Grall, L. Dieulle, C. Bérenguer, and M. Roussignol. Continuous-time predictive-maintenance scheduling for a deteriorating system. *IEEE Transactions on Reliability*, 51(2):141–150, 2002.

V. Gupta, M. Sharma, and N. Thakur. Optimization criteria for optimal placement of piezoelectric sensors and actuators on a smart structure: a technical review. *Journal of Intelligent Material Systems and Structures*, 21(12):1227–1243, 2010.

R. F. Guratzsch and S. Mahadevan. Structural health monitoring sensor placement optimization under uncertainty. *AIAA journal*, 48(7): 1281–1289, 2010.

W. He, N. Williard, M. Osterman, and M. Pecht. Prognostics of lithium-ion batteries based on dempster–shafer theory and the bayesian monte carlo method. *Journal of Power Sources*, 196 (23):10314–10321, 2011.

F. O. Heimes. Recurrent neural networks for remaining useful life estimation. In *2008 international conference on prognostics and health management*, pages 1–6. IEEE, 2008.

A. Heydari, M. Aghabozorgi, and M. Biguesh. Optimal sensor placement for source localization based on rssd. *Wireless Networks*, 26(7): 5151–5162, 2020.

R. Hills, K. Dowding, and L. Swiler. Thermal challenge problem: summary. *Computer Methods in Applied Mechanics and Engineering*, 197 (29-32):2490–2495, 2008.

K. Honkura, K. Takahashi, and T. Horiba. Capacity-fading prediction of lithium-ion batteries based on discharge curves analysis. *Journal of power sources*, 196(23):10141–10147, 2011.

M.-H. Hsu. *Machine learning-based non-destructive evaluation of fatigue damage in metals*. PhD thesis, 2021.

C. Hu, B. D. Youn, P. Wang, and J. T. Yoon. Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. *Reliability Engineering & System Safety*, 103: 120–135, 2012.

C. Hu, G. Jain, P. Tamirisa, and T. Gorka. Method for estimating capacity and predicting remaining useful life of lithium-ion battery. In *2014 International Conference on Prognostics and Health Management*, pages 1–8. IEEE, 2014.

C. Hu, H. Ye, G. Jain, and C. Schmidt. Remaining useful life assessment of lithium-ion batteries in implantable medical devices. *Journal of Power Sources*, 375:118–130, 2018.

X. Hu, L. Xu, X. Lin, and M. Pecht. Battery lifetime prognostics. *Joule*, 4(2):310–346, 2020.

Z. Hu and S. Mahadevan. Uncertainty quantification and management in additive manufacturing: current status, needs, and opportunities. *The International Journal of Advanced Manufacturing Technology*, 93(5):2855–2874, 2017.

Z. Hu, D. Ao, and S. Mahadevan. Calibration experimental design considering field response and model uncertainty. *Computer Methods in Applied Mechanics and Engineering*, 318: 92–119, 2017.

Z. Hu, C. Hu, Z. P. Mourelatos, and S. Mahadevan. Model discrepancy quantification in simulation-based design of dynamical systems. *Journal of Mechanical Design*, 141(1), 2019.

X. Huan and Y. Marzouk. Gradient-based stochastic optimization methods in bayesian experimental design. *International Journal for Uncertainty Quantification*, 4(6), 2014.

X. Huan and Y. M. Marzouk. Simulation-based optimal bayesian experimental design for nonlinear systems. *Journal of Computational Physics*, 232(1):288–317, 2013.

G. P. Huber. Multi-attribute utility models: A review of field and field-like studies. *Management science*, 20(10):1393–1402, 1974.

I. IBM Blog. How to create a twin to improve your own performance, Sep 2019. URL https://www.ibm.com/blogs/internet-of-things/iot-digital-twin-rotterdam/.

I. IBM Blog. Profile: Ucsf health and maximo lead the way on smart medical buildings, Jul 2020. URL https://www.ibm.com/blogs/internet-of-things/iot-ucsf-health-and-maximo-smart-medical-buildings/.

I. IBM Newsletter. Siemens and ibm deliver service lifecycle management solution, Jul 2020. URL https://newsroom.ibm.com/2020-06-17-Siemens-and-IBM-Deliver-Service-Lifecycle-Management-Solution.

I. IBM White Paper. Digital twin technologies for high-performance manufacturing, Jan 2020. URL https://www.ibm.com/downloads/cas/KX8A3MWX.

ICSHM. International project competition for structural health monitoring. http://www.schm.org.cn/#/IPC-SHM,2020, 2020. Last accessed on 2022-05-24.

W. L. Ijomah, S. Childe, and C. McMahon. Remanufacturing: a key strategy for sustainable development. 2004.

ISA. Ansi/isa-95.00.01-2010 (iec 62264-1 mod) enterprise-control system integration - part 1: Models and terminology. https://www.isa.org/products/ansi-isa-95-00-01-2010-iec-62264-1-mod-enterprise, 2010. Last Accessed: 2022-05-29.

C. Jiang, Z. Hu, Y. Liu, Z. P. Mourelatos, D. Gorsich, and P. Jayakumar. A sequential calibration and validation framework for model uncertainty quantification and reduction. *Computer Methods in Applied Mechanics and Engineering*, 368: 113172, 2020.

C. Jiang, Z. Hu, Z. P. Mourelatos, D. Gorsich, P. Jayakumar, Y. Fu, and M. Majcher. R2-RRT*: reliability-based robust mission planning of off-road autonomous ground vehicle under uncertain terrain environment. *IEEE Transactions on Automation Science and Engineering*, 2021.

C. Jiang, Y. Liu, Z. P. Mourelatos, D. Gorsich, Y. Fu, and Z. Hu. Efficient reliability-based mission planning of off-road autonomous ground vehicles using an outcrossing approach. *Journal of Mechanical Design*, 144(4), 2022a.

C. Jiang, M. A. Vega, M. K. Ramancha, M. D. Todd, J. P. Conte, M. Parno, and Z. Hu. Bayesian calibration of multi-level model with unobservable distributed response and application to miter gates. *Mechanical Systems and Signal Processing*, 170:108852, 2022b.

C. Jiang, M. A. Vega, M. D. Todd, and Z. Hu. Model correction and updating of a stochastic degradation model for failure prognostics of miter gates. *Reliability Engineering & System Safety*, 218:108203, 2022c.

X. Jiang and S. Mahadevan. Bayesian inference method for model validation and confidence extrapolation. *Journal of Applied Statistics*, 36 (6):659–677, 2009.

J. B. Johnson, A. V. Kulchitsky, P. Duvoy, K. Iagnemma, C. Senatore, R. E. Arvidson, and J. Moore. Discrete element method simulations of mars exploration rover wheel performance. *Journal of Terramechanics*, 62:31–40, 2015.

D. C. Kammer. Sensor placement for on-orbit modal identification and correlation of large space structures. *Journal of Guidance, Control, and Dynamics*, 14(2):251–259, 1991.

M. G. Kapteyn, J. V. Pretorius, and K. E. Willcox. A probabilistic graphical model foundation for enabling predictive digital twins at scale. *Nature Computational Science*, 1(5):337–347, 2021.

A. Kaveh, A. Dadras Eslamlou, P. Rahmani, and P. Amirsoleimani. Optimal sensor placement in large-scale dome trusses via q-learning-based water strider algorithm. *Structural Control and Health Monitoring*, page e2949, 2022.

A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

T. Kim, B. D. Youn, and H. Oh. Development of a stochastic effective independence (sefi) method for optimal sensor placement under uncertainty. *Mechanical Systems and Signal Processing*, 111: 615–627, 2018.

W. Kim, H. Yoon, G. Lee, T. Kim, and B. D. Youn. A new calibration metric that considers statistical correlation: Marginal probability and correlation residuals. *Reliability Engineering & System Safety*, 195:106677, 2020.

J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium*

Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), volume 2, pages 995–1001. IEEE, 2000.

V. Kuleshov, N. Fenner, and S. Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International conference on machine learning*, pages 2796–2804. PMLR, 2018.

R. V. Kulkarni and G. K. Venayagamoorthy. Particle swarm optimization in wireless-sensor networks: A brief survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(2):262–267, 2010.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

J. Lee, E. Lapira, B. Bagheri, and H.-a. Kao. Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing letters*, 1(1):38–41, 2013a.

J. Lee, E. Lapira, S. Yang, and A. Kao. Predictive manufacturing system-trends of next-generation production systems. *Ifac proceedings volumes*, 46(7):150–156, 2013b.

H. Lehner and L. Dorffner. Digital geotwin vienna: towards a digital twin city as geodata hub, 2020.

X. Lei and P. A. Sandborn. Maintenance scheduling based on remaining useful life predictions for wind farms managed using power purchase agreements. *Renewable Energy*, 116:188–198, 2018.

J. Li, X. Zhang, J. Xing, P. Wang, Q. Yang, and C. He. Optimal sensor placement for long-span cable-stayed bridge using a novel particle swarm optimization algorithm. *Journal of Civil Structural Health Monitoring*, 5(5):677–685, 2015.

M. Li, V. P. Nemani, J. Liu, M. A. Lee, N. Ahmed, G. E. Kremer, and C. Hu. Reliability-informed life cycle warranty cost and life cycle analysis of newly manufactured and remanufactured units. *Journal of Mechanical Design*, 143(11), 2021a.

S. Li, H. Fang, and B. Shi. Remaining useful life estimation of lithium-ion battery based on interacting multiple model particle filter and support vector regression. *Reliability Engineering & System Safety*, 210:107542, 2021b.

W. Li, W. Chen, Z. Jiang, Z. Lu, and Y. Liu. New validation metrics for models with multiple correlated responses. *Reliability Engineering & System Safety*, 127:1–11, 2014.

Y. Li, S. Sui, and S. Tong. Adaptive fuzzy control design for stochastic nonlinear switched systems with arbitrary switchings and unmodeled dynamics. *IEEE transactions on cybernetics*, 47 (2):403–414, 2016.

Y. Ling and S. Mahadevan. Quantitative model validation techniques: New insights. *Reliability Engineering & System Safety*, 111:217–231, 2013.

J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33: 7498–7512, 2020.

W. Liu, W.-c. Gao, Y. Sun, and M.-j. Xu. Optimal sensor placement for spatial lattice structure based on genetic algorithms. *Journal of Sound and Vibration*, 317(1-2):175–189, 2008.

X. Liu, M. Gao, J. Zhao, X. Sun, Z. Li, Q. Li, L. Wang, J. Wang, and W. Zhuang. Effects of charging protocols on the cycling performance for high-energy lithium-ion batteries using a graphite-siox composite anode and li-rich layered oxide cathode. *Journal of Power Sources*, 495:229793, 2021a.

Y. Liu and X.-Y. Li. Decentralized robust adaptive control of nonlinear systems with unmodeled dynamics. *IEEE Transactions on Automatic control*, 47(5):848–856, 2002.

Y. Liu, W. Chen, P. Arendt, and H.-Z. Huang. Toward a better understanding of model validation metrics. *Journal of Mechanical Design*, 133(7), 2011.

Y. Liu, L. Zhang, Y. Yang, L. Zhou, L. Ren, F. Wang, R. Liu, Z. Pang, and M. J. Deen. A novel cloud-based framework for the elderly healthcare services using digital twin. *IEEE access*, 7:49088–49101, 2019.

Y. Liu, C. Jiang, Z. P. Mourelatos, D. Gorsich, P. Jayakumar, Y. Fu, M. Majcher, and Z. Hu. Simulation-based mission mobility reliability analysis of off-road ground vehicles. *Journal of Mechanical Design*, 143(3), 2021b.

Y. Liu, C. Jiang, X. Zhang, Z. P. Mourelatos, D. Barthlow, D. Gorsich, A. Singh, and Z. Hu. Reliability-based multi-vehicle path planning under uncertainty using a bio-inspired approach. *Journal of Mechanical Design*, pages 1–44, 2021c.

Q. Long, M. Scavino, R. Tempone, and S. Wang. Fast estimation of expected information gains for bayesian experimental designs based on laplace approximations. *Computer Methods in Applied Mechanics and Engineering*, 259:24–39, 2013.

Q. Lu, A. K. Parlikad, P. Woodall, G. Don Ranasinghe, X. Xie, Z. Liang, E. Konstantinou, J. Heaton, and J. Schooling. Developing a digital twin at building and city levels: Case study of west cambridge campus. *Journal of Management in Engineering*, 36(3):05020004, 2020.

H. LU LG, J. Li, et al. A review on the key issues for lithium-ion battery management in electric vehicles. *Journal of power sources*, 226:272–288, 2013.

Y. H. Lui, M. Li, A. Downey, S. Shen, V. P. Nemani, H. Ye, C. VanElzen, G. Jain, S. Hu, S. Laflamme, et al. Physics-based prognostics of implantable-grade lithium-ion battery for remaining useful life prediction. *Journal of Power Sources*, 485:229327, 2021.

S. Mahadevan, P. Nath, and Z. Hu. Uncertainty quantification for additive manufacturing process improvement: Recent advances. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 8(1):010801, 2022.

A. A. Malik. Framework to model virtual factories: A digital twin view. *arXiv preprint arXiv:2104.03034*, 2021.

A. A. Malik and A. Brem. Digital twins for collaborative robots: A case study in human-robot interaction. *Robotics and Computer-Integrated Manufacturing*, 68:102092, 2021.

C. Malings and M. Pozzi. Value of information for spatially distributed systems: Application to sensor placement. *Reliability Engineering & System Safety*, 154:219–233, 2016.

C. Malings, M. Pozzi, and I. Velibeyoglu. Sensor placement optimization for structural health monitoring. In *Proceedings of the 10th International Workshop on Structural Health Monitoring*, 2015.

A. Mandelbaum and D. Weinshall. Distance-based confidence score for neural network classifiers. *arXiv preprint arXiv:1709.09844*, 2017.

Markets and M. Markets. Digital twin market, Sep 2020. URL https://www.marketsandmarkets.com/Market-Reports/digital-twin-market-225269522.html.

M. Meo and G. Zumpano. On the optimal sensor placement techniques for a bridge structure. *Engineering structures*, 27(10):1488–1497, 2005.

Q. Miao, L. Xie, H. Cui, W. Liang, and M. Pecht. Remaining useful life prediction of lithium-ion battery with unscented particle filter technique. *Microelectronics Reliability*, 53(6):805–810, 2013.

R. Moghaddass and M. J. Zuo. An integrated framework for online diagnostic and prognostic health monitoring using a multistate deterioration process. *Reliability Engineering & System Safety*, 124:92–104, 2014.

H. Mousazadeh. A technical review on navigation systems of agricultural autonomous off-road vehicles. *Journal of Terramechanics*, 50 (3):211–232, 2013.

J. Mukhoti, A. Kirsch, J. van Amersfoort, P. H. Torr, and Y. Gal. Deterministic neural networks with appropriate inductive biases capture epistemic and aleatoric uncertainty. *arXiv e-prints*, pages arXiv–2102, 2021.

Z. Nado, N. Band, M. Collier, J. Djolonga, M. W. Dusenberry, S. Farquhar, Q. Feng, A. Filos, M. Havasi, R. Jenatton, et al. Uncertainty baselines: Benchmarks for uncertainty & robustness in deep learning. *arXiv preprint arXiv:2106.04015*, 2021.

NASA. Standard for models and simulation-nasa technical standard. *National Aeronautics and Space Administration*, Washington(DC): Standard No.NASA–STD–7009, 2008.

P. Nath, Z. Hu, and S. Mahadevan. Sensor placement for calibration of spatially varying model parameters. *Journal of Computational Physics*, 343:150–169, 2017.

V. P. Nemani, H. Lu, A. Thelen, C. Hu, and A. T. Zimmerman. Ensembles of probabilistic lstm predictors and correctors for bearing prognostics using industrial standards. *Neurocomputing*, 2021.

A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 625–632, 2005.

W. Ostachowicz, R. Soman, and P. Malinowski. Optimization of sensor placement for structural health monitoring: A review. *Structural Health*

Monitoring, 18(3):963–988, 2019.

T. Papamarkou, J. Hinkle, M. T. Young, and D. Womble. Challenges in markov chain monte carlo for bayesian neural networks. *Statistical Science*, 2021.

N. A. PCoE. Prognostics center of excellence - data repository. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/. Last accessed on 2022-04-23.

PHM. Phm society data challenge. https://data.phmsociety.org/, 2022. Last accessed on 2022-05-24.

PHMAP. Data challenge at phmap 2021. http://phmap.org/data-challenge/, 2021. Last accessed on 2022-05-24.

PHME. The annual phme data challenge. https://phm-europe.org/data-challenge, 2022. Last accessed on 2022-05-24.

G. L. Plett. Extended kalman filtering for battery management systems of lipb-based hev battery packs: Part 3. state and parameter estimation. *Journal of Power sources*, 134(2):277–292, 2004.

G. L. Plett. Sigma-point kalman filtering for battery management systems of lipb-based hev battery packs: Part 2: Simultaneous state and parameter estimation. *Journal of power sources*, 161(2):1369–1384, 2006.

E. E. Prudencio and K. W. Schulz. The parallel C++ statistical library 'QUESO': Quantification of Uncertainty for Estimation, Simulation and Optimization. In *Euro-Par 2011: Parallel Processing Workshops*, pages 398–407. Springer, 2012. URL http://dx.doi.org/10.1007/978-3-642-29737-3_44.

M. K. Ramancha, J. P. Conte, and M. D. Parno. Accounting for model form uncertainty in bayesian calibration of linear dynamic systems. *Mechanical Systems and Signal Processing*, 171: 108871, 2022.

R. Rebba and S. Mahadevan. Computational methods for model reliability assessment. *Reliability Engineering & System Safety*, 93(8): 1197–1207, 2008.

N. L. Ricker. Te code. http://depts.washington.edu/control/LARRY/TE/download.html. Last Accessed: 2022-04-17.

N. L. Ricker and J. Lee. Nonlinear model predictive control of the tennessee eastman challenge process. *Computers & Chemical Engineering*, 19(9):961–981, 1995.

C. Rohrs, L. Valavani, M. Athans, and G. Stein. Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics. *IEEE Transactions on Automatic Control*, 30(9):881–889, 1985.

C. E. Rohrs, L. Valavani, M. Athans, and G. Stein. Robustness of adaptive control algorithms in the presence of unmodeled dynamics. In *1982 21st IEEE Conference on Decision and Control*, pages 3–11. IEEE, 1982.

S. Sabatino, D. M. Frangopol, and Y. Dong. Sustainability-informed maintenance optimization of highway bridges considering multi-attribute utility and risk attitude. *Engineering Structures*, 102:310–321, 2015.

V. K. Sachan, S. A. Imam, and M. Beg. Energy-efficient communication methods in wireless sensor networks: A critical review. *International Journal of Computer Applications*, 39 (17):35–48, 2012.

B. Saha and K. Goebel. Battery data set. *NASA AMES prognostics data repository*, 2007.

B. Saha, K. Goebel, S. Poll, and J. Christophersen. Prognostics methods for battery health monitoring using a bayesian framework. *IEEE Transactions on instrumentation and measurement*, 58(2):291–296, 2008.

B. Saha, K. Goebel, and J. Christophersen. Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of the Institute of Measurement and Control*, 31 (3-4):293–308, 2009.

J. Salvatier, T. V. Wiecki, and C. Fonnesbeck. Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55, 2016.

A. Saxena and K. Goebel. Phm08 challenge data set. *NASA Ames Prognostics Data Repository*, 2008a.

A. Saxena and K. Goebel. Turbofan engine degradation simulation data set. *NASA Ames Prognostics Data Repository*, pages 1551–3203, 2008b.

E. Scott, J. Brown, C. Schmidt, and W. Howard. A practical longevity model for lithium-ion batteries: De-coupling the time and cycle-dependence of capacity fade. In *208th ECS Meeting*, 2005.

L. Sela and S. Amin. Robust sensor placement for pipeline monitoring: Mixed integer and greedy

optimization. *Advanced engineering informatics*, 36:55–63, 2018.

K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggedakis, et al. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4(5):383–391, 2019.

M. Sharma and J. George. Digital twin in the automotive industry: Driving physical-digital convergence. *Tata Consultancy Services White Paper*, 2018.

W. Shen and X. Huan. Bayesian sequential optimal experimental design for nonlinear models using policy gradient reinforcement learning. *arXiv preprint arXiv:2110.15335*, 2021.

S. Siemens Newsletter. Digitalization in industry: Twins with potential, Jun 2020. URL https://new.siemens.com/global/en/company/stories/industry/the-digital-twin.html.

W. Sisson, P. Karve, and S. Mahadevan. Digital twin approach for component health-informed rotorcraft flight parameter optimization. *AIAA Journal*, 60(3):1923–1936, 2022.

M. Sjarov, T. Lechler, J. Fuchs, M. Brossog, A. Selmaier, F. Faltus, T. Donhauser, and J. Franke. The digital twin concept in industry–a review and systematization. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 1789–1796. IEEE, 2020.

A. Skardal, T. Shupe, and A. Atala. Organoid-on-a-chip and body-on-a-chip systems for drug screening and disease modeling. *Drug discovery today*, 21(9):1399–1411, 2016.

R. C. Smith. *Uncertainty quantification: theory, implementation, and applications*, volume 12. Siam, 2013.

D. SNL. Battery archive. https://www.batteryarchive.org/. Last accessed on 2022-04-23.

J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

M. Song, B. Moaveni, C. Papadimitriou, and A. Stavridis. Accounting for amplitude of excitation in model updating through a hierarchical bayesian approach: Application to a two-story reinforced concrete building. *Mechanical Systems and Signal Processing*, 123:68–83, 2019.

P. Soundappan, E. Nikolaidis, R. T. Haftka, R. Grandhi, and R. Canfield. Comparison of evidence theory and bayesian theory for uncertainty modeling. *Reliability engineering & System safety*, 85(1-3):295–311, 2004.

A. Subramanian and S. Mahadevan. Error estimation in coupled multi-physics models. *Journal of Computational Physics*, 395:19–37, 2019.

K. Suresh and N. Kumarappan. Hybrid improved binary particle swarm optimization approach for generation maintenance scheduling problem. *Swarm and Evolutionary Computation*, 9:69–89, 2013.

Y. Tan and L. Zhang. Computational methodologies for optimal sensor placement in structural health monitoring: A review. *Structural Health Monitoring*, 19(4):1287–1308, 2020.

A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut. Chrono: An open source multi-physics dynamics engine. In *International Conference on High Performance Computing in Science and Engineering*, pages 19–49. Springer, 2015.

T. P. D. Team. Pymc3. https://docs.pymc.io/en/v3/, a. Last accessed on 2022-04-17.

T. Q. D. Team. Queso. https://github.com/libqueso/queso/, b. Last accessed on 2022-04-17.

T. S. D. Team. Shm data sets and software. https://www.lanl.gov/projects/national-security-education-center/engineering/software/shm-data-sets-and-software.php, 2022. Last accessed on 2022-05-24.

A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B. D. Youn, M. D. Todd, S. Mahadevan, C. Hu, and Z. Hu. A comprehensive review of digital twin - part 1: Modeling and twinning enabling technologies. *Structural and Multidisciplinary Optimization*, 2022.

K. Tong, N. Bakhary, A. Kueh, and A. Yassin. Optimal sensor placement for mode shapes using improved simulated annealing. *Smart Struct. Syst*, 13(3):389–406, 2014.

J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020.

B. F. Van Dongen, A. K. A. de Medeiros, H. Verbeek, A. Weijters, and W. M. van Der Aalst. The prom framework: A new era in process mining tool support. In *International conference on application and theory of petri nets*, pages 444–454. Springer, 2005.

E. VanDerHorn and S. Mahadevan. Digital twin: Generalization, characterization and implementation. *Decision Support Systems*, 145:113524, 2021.

M. A. Vega, Z. Hu, T. B. Fillmore, M. D. Smith, and M. D. Todd. A novel framework for integration of abstracted inspection data and structural health monitoring for damage prognosis of miter gates. *Reliability Engineering & System Safety*, 211:107561, 2021.

H. Verbeek, J. Buijs, B. Van Dongen, and W. M. van der Aalst. Prom 6: The process mining toolkit. *Proc. of BPM Demonstration Track*, 615:34–39, 2010.

F. A. Viana, R. G. Nascimento, A. Dourado, and Y. A. Yucesan. Estimating model inadequacy in ordinary differential equations with physics-informed neural networks. *Computers & Structures*, 245:106458, 2021.

E. Walker, S. Rayman, and R. E. White. Comparison of a particle filter and other state estimation methods for prognostics of lithium-ion batteries. *Journal of Power Sources*, 287:1–12, 2015.

D. Wang, Q. Miao, and M. Pecht. Prognostics of lithium-ion batteries based on relevance vectors and a conditional three-parameter capacity degradation model. *Journal of Power Sources*, 239:253–264, 2013.

P. Wang and T. Wang. Adaptive routing for sensor networks using reinforcement learning. In *The Sixth IEEE International Conference on Computer and Information Technology (CIT'06)*, pages 219–219. IEEE, 2006.

P. Wang, B. D. Youn, C. Hu, J. M. Ha, and B. Jeon. A probabilistic detectability-based sensor network design method for system health monitoring and prognostics. *Journal of Intelligent Material Systems and Structures*, 26(9): 1079–1090, 2015.

S. Wang, W. Chen, and K.-L. Tsui. Bayesian validation of computer models. *Technometrics*, 51 (4):439–451, 2009.

T. Wang, J. Yu, D. Siegel, and J. Lee. A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *2008 international conference on prognostics and health management*, pages 1–6. IEEE, 2008.

Z. Wang, H.-X. Li, and C. Chen. Reinforcement learning-based optimal sensor placement for spatiotemporal modeling. *IEEE Transactions on Cybernetics*, 50(6):2861–2871, 2019.

R. Ward, R. Choudhary, A. Gregory, M. Jans-Singh, and M. Girolami. Continuous calibration of a digital twin: Comparison of particle filter and bayesian calibration approaches. *Data-Centric Engineering*, 2, 2021.

M. Weigert, U. Schmidt, T. Boothe, A. Müller, A. Dibrov, A. Jain, B. Wilhelm, D. Schmidt, C. Broaddus, S. Culley, et al. Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nature methods*, 15(12): 1090–1097, 2018.

G. White, A. Zink, L. Codecá, and S. Clarke. A digital twin smart city for citizen feedback. *Cities*, 110:103064, 2021.

R. D. Wilkinson, M. Vrettas, D. Cornford, and J. E. Oakley. Quantifying simulator discrepancy in discrete-time dynamical simulators. *Journal of agricultural, biological, and environmental statistics*, 16(4):554–570, 2011.

C. Williams and C. Rasmussen. Gaussian processes for regression. *Advances in neural information processing systems*, 8, 1995.

D. V. Winterfeldt and G. W. Fischer. Multi-attribute utility theory: models and assessment procedures. *Utility, probability, and human decision making*, pages 47–85, 1975.

Z. Xi, M. Dahmardeh, B. Xia, Y. Fu, and C. Mi. Learning of battery model bias for effective state of charge estimation of lithium-ion batteries. *IEEE Transactions on Vehicular Technology*, 68 (9):8613–8628, 2019.

Y. Xiong, W. Chen, K.-L. Tsui, and D. W. Apley. A better understanding of model updating strategies in validating engineering models. *Computer Methods in Applied Mechanics and Engineering*, 198(15-16):1327–1337, 2009.

J. Yan, S. Laflamme, J. Hong, and J. Dodson. Online parameter estimation under non-persistent excitations for high-rate dynamic systems. *Mechanical Systems and Signal Processing*, 161:107960, 2021.

C. Yang, K. Liang, and X. Zhang. Strategy for sensor number determination and placement optimization with incomplete information

based on interval possibility model and clustering avoidance distribution index. *Computer Methods in Applied Mechanics and Engineering*, 366:113042, 2020a.

Y. Yang, M. Chadha, Z. Hu, M. A. Vega, M. D. Parno, and M. D. Todd. A probabilistic optimal sensor design approach for structural health monitoring using risk-weighted f-divergence. *Mechanical Systems and Signal Processing*, 161: 107920, 2021.

Z. Yang, Y. Lu, H. Yeung, and S. Kirishnamurty. 3d build melt pool predictive modeling for powder bed fusion additive manufacturing. 22662: V009T09A046, 2020b.

L. Yao, W. A. Sethares, and D. C. Kammer. Sensor placement for on-orbit modal identification via a genetic algorithm. *AIAA journal*, 31(10): 1922–1928, 1993.

M. Ye, H. Guo, R. Xiong, and Q. Yu. A double-scale and adaptive particle filter-based online parameter and state of charge estimation method for lithium-ion batteries. *Energy*, 144: 789–799, 2018.

H. Yeung, Z. Yang, and Y. Lu. A meltpool prediction based scan strategy for powder bed fusion additive manufacturing. *Journal of Additive Manufacturing*, 35:101383, 2020.

T.-H. Yi, H.-N. Li, and M. Gu. Optimal sensor placement for structural health monitoring based on multiple optimization strategies. *The Structural Design of Tall and Special Buildings*, 20(7):881–900, 2011.

Y. A. Yucesan and F. A. Viana. A physics-informed neural network for wind turbine main bearing fatigue. *International Journal of Prognostics and Health Management*, 11(1), 2020.

A. Zacharaki, T. Vafeiadis, N. Kolokas, A. Vaxevani, Y. Xu, M. Peschl, D. Ioannidis, and D. Tzovaras. Reclaim: Toward a new era of refurbishment and remanufacturing of industrial equipment. *Frontiers in Artificial Intelligence*, page 101, 2021.

C. Zhang, P. Lim, A. K. Qin, and K. C. Tan. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2306–2318, 2016.

J. Zhang and J. Lee. A review on prognostics and health monitoring of li-ion battery. *Journal of power sources*, 196(15):6007–6014, 2011.

Q. Zhang, L. Shi, M. Holzman, M. Ye, Y. Wang, F. Carmona, and Y. Zha. A dynamic data-driven method for dealing with model structural error in soil moisture data assimilation. *Advances in Water Resources*, 132:103407, 2019.

X. Zhang, J. Li, J. Xing, P. Wang, Q. Yang, R. Wang, and C. He. Optimal sensor placement for latticed shell structure based on an improved particle swarm optimization algorithm. *Mathematical problems in Engineering*, 2014, 2014.

X. Zhang, S. Mahadevan, and X. Deng. Reliability analysis with linguistic data: An evidential network approach. *Reliability Engineering & System Safety*, 162:111–121, 2017.

Y. Zhao, V. Pandey, H. Kim, and D. Thurston. Varying lifecycle lengths within a product take-back portfolio. 2010.

Z. Zhao, B. Liang, X. Wang, and W. Lu. Remaining useful life prediction of aircraft engine based on degradation pattern learning. *Reliability Engineering & System Safety*, 164:74–83, 2017.

J. Zhu, I. Mathews, D. Ren, W. Li, D. Cogswell, B. Xing, T. Sedlatschek, S. N. R. Kantareddy, M. Yi, T. Gao, et al. End-of-life or second-life options for retired electric vehicle batteries. *Cell Reports Physical Science*, 2(8):100537, 2021.