Mechanical Systems and Signal Processing

Bayesian Calibration of Multi-Level Model with Unobservable Distributed Response and Application to Miter Gates

Chen Jiang[1], Manuel A. Vega[2, 3], Mukesh K. Ramancha[2], Michael D. Todd[2], Joel P. Conte[2], Matthew Parno[4], and Zhen Hu[1*]

[1]*Department of Industrial and Manufacturing Systems Engineering, University of Michigan-Dearborn, Dearborn, MI 48128, USA*

[2]*Department of Structural Engineering, University of California San Diego, La Jolla, CA 92093, USA*

[3]*Advanced Engineering Analysis Group, Los Alamos National Laboratory, Los Alamos, NM 87545*

[4]*Department of Mathematics, Dartmouth College, Hanover, NH, 03755, USA*

ABSTRACT

Bayesian calibration plays a vital role in improving the validity of computational models' predictive power. However, the presence of unobservable distributed responses and uncertain model parameters in multi-level models poses challenges to Bayesian calibration, due to the lack of direct observations and the difficulty in identifying the hidden and distributed model discrepancy under uncertainty. This paper proposes a Bayesian calibration framework for multi-level simulation models to calibrate an unobservable distributed model using measurements of an observable model. In the proposed framework, the distributed model discrepancy of an unobservable model with distributed response is first represented as a series of orthogonal polynomials, with the polynomial coefficients modelled by surrogate models with unknown hyper-parameters. A two-phase machine learning method is then developed to construct surrogate models of the polynomial coefficients based on measurements of an observable model. The constructed model discrepancy is finally used to update the uncertain model parameters by following a modularized Bayesian calibration scheme. The developed framework is applied to the joint Bayesian calibration of the uncertain gap length and unobservable and distributed boundary condition model for a miter gate problem. Results of the miter gate application demonstrate the efficacy of the proposed framework.

Keywords: Multi-level model; Bayesian calibration; Unobservable distributed response; Distributed model discrepancy; Surrogate model; Miter gates.

[*] Corresponding author: 2340 HPEC, University of Michigan-Dearborn, Dearborn, MI 48128, USA, Tel:+1-313-583-6312, Email: zhennhu@umich.edu

1 INTRODUCTION

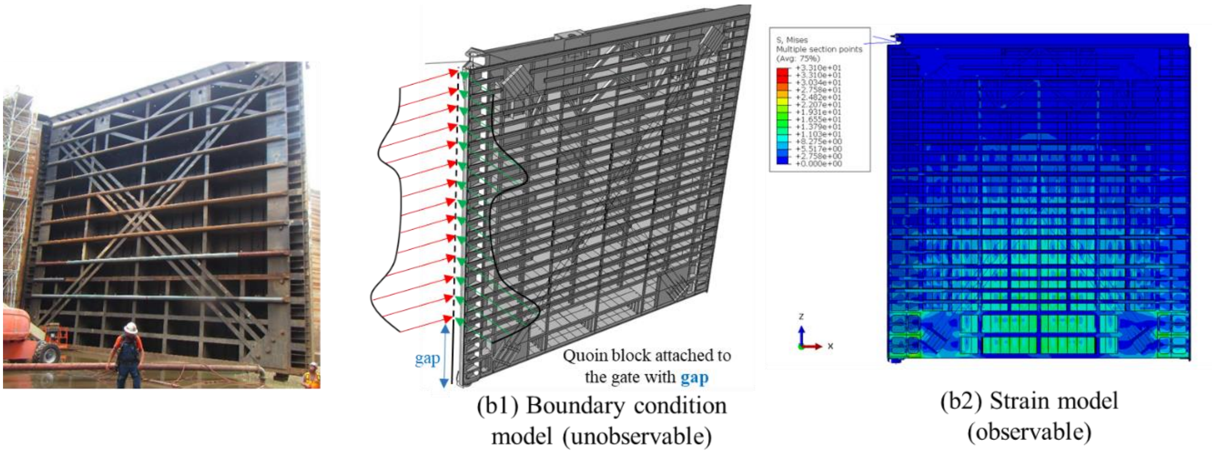
The calibration of Modeling and simulation (M&S) framework plays an important role in improving predictive power by blending observational data with parameterized models. This is important in simulation-based design, analysis, and decision making under uncertainty. Conceptually, calibration is the process of correcting the model using experiments/observations to better capture the true underlying physics and thereby increase its prediction accuracy.

Model calibration, which is also referred as model updating [1], has been extensively studied using both deterministic/non-probabilistic (e.g., optimization) and probabilistic methods (e.g., Bayesian) [2]. Non-probabilistic approaches, such as optimization-based methods, usually calibrate model parameters by minimizing the differences between computer simulation outputs and experimental observations through a certain metric. Some commonly used metrics include but are not limited to the Euclidean distance, Manhattan distance, Minkowski distance, similarity metric, etc. [3]. Uncertainties stemming from noisy observations, unconstrained model parameters, and discrepancies between models and reality (model form uncertainty) are ubiquitous in real-world model calibration problems, which makes probabilistic approaches, especially Bayesian methods, particularly attractive to the model calibration community. Bayesian approaches provide a versatile calibration framework for probabilistically characterizing and propagating these uncertainties.

Existing Bayesian calibration approaches may be roughly classified into three categories depending on what uncertainties are considered in the problem: (1) calibration of unknown model parameters only [4], (2) quantification of model discrepancy only [5], and (3) simultaneous parameter and model discrepancy characterization [6-8]. The first category has been widely studied for structural performance assessment, and response prediction of static or dynamic systems [9-11]. For instance, the uncertain model parameters are updated to minimize the discrepancy between the model-predicted responses and measured data [12, 13]. To improve the prediction accuracy of predictive models of linear dynamic systems, hierarchical Bayesian approaches have been developed for civil structural systems under changing ambient/environmental conditions [14], for crack detection [15], and for dynamic systems in a wide range of excitation levels [16]. Moreover, Ni et al. [17] proposed a likelihood-free Bayesian inference approach of civil structures to significantly reduce the required computational time for model calibration. Chiachio et al. [18] proposed a multi-level Bayesian method in the context of ultrasound-based damage identification of composite laminates. Ramancha et al. [19] investigated the non-identifiability issue in Bayesian model updating of nonlinear finite element models. The second category assigns all possible biases between simulations and observations to the model discrepancy function. The third category separately considers contributions of uncertain model parameters and model discrepancy in a holistic framework [20, 21]. It has been shown that the third category is more applicable when the calibrated model is subsequently used to make predictions under new, previously unseen, conditions [22, 23]. A widely used framework that falls into the third category is the Kennedy and O'Hagan framework (referred to as the KOH framework) [20]. The KOH framework and its variants [22-24] explicitly account for the model inadequacy by approximating the prediction model and model discrepancy with Gaussian process models [25, 26] and considering various sources of uncertainty in the Bayesian calibration process.

Although the aforementioned Bayesian calibration approaches can address the model calibration of various linear or nonlinear dynamic problems (for example, a Bayesian state-space approach is proposed to infer the latent states and parameters of the state dynamics for damage detection [27]), it is still necessary to point out that they are not applicable if the predicted quantity of interest (QoI) is a spatially distributed latent response that cannot be directly measured. If the QoI is fundamentally unobservable then an additional model relating the unobservable QoI to an observable quantity must be employed during calibration. This observable model is coupled with the original unobservable model in a cascaded manner during the simulation. In this paper, this type of unobservable-observable model architecture is referred to as a *multi-level* model.

As an example of multi-level model, consider the boundary condition finite element (FE) model of a miter gate structure (more fully explained in Sec. 4) as shown in Fig. 1. The QoI predicted by this model is the boundary force between the quoin block attached to the lock wall and that attached to the miter gate; however, this force cannot be observed directly and must be related to strain measurements using an observable model of the strain response. The boundary force considered here plays an important role in analyzing the fatigue crack initiation at the quoin block due to the rolling contact between the wall and the gate [28].



(a) Miter gate **(b) Two-level model of analysis**

Fig. 1 Miter gate model with uncertain and unobservable boundary condition

Even though models have been developed to predict the boundary force at the quoin block by using a contact analysis model [29], the predictions may not be accurate due to contact modeling assumptions, model simplification, and/or numerical discretization errors. In addition, the unknown damage state (e.g., loss of contact between the gate quoin block and the wall quoin block at the bottom of the gate, hereafter referred to as the “gap”) of the miter gate further complicates the boundary load condition analysis [30, 31]. Additional information from observational data could help characterize the boundary forces, but the boundary force cannot be measured and the boundary condition model is an unobservable model (Fig. 1 (b1)). However, the strain response of the gate can be observed and can be exploited as an observable model (Fig. 1 (b2)).

After calibration, the output of the boundary force model may then be used to simulation other QoIs, such as damage growth (at any stage) or fatigue. The model that uses the boundary force as an input and predicts these other QoIs is referred to as a prediction model. In this work, this kind of system containing an unobservable model, observable model, and prediction model is called a multi-level model (see Fig. 2). Multi-level models are also very common in material science where multi-scale models are usually connected together to predict the macroscopic material properties [32, 33]. The multi-level architecture of simulators, the presence of unknown model parameters, and distributed unobservable response introduce great challenges to the Bayesian calibration of real-world M&S architectures. Even though some efforts have been made recently to calibrate this type of multi-level models and quantify the hidden model discrepancy using the embedded model uncertainty approaches [34, 35], the current methods are inapplicable to problems with both uncertain model parameters and unobservable and distributed responses.

In response to this challenge, this paper proposes a Bayesian calibration framework for multi-level models where the unobservable model has a distributed response. In the proposed framework, model discrepancy is explicitly accounted for within the modularized Bayesian calibration scheme. A new approach is first proposed to estimate both a regression parameter and the distributed model discrepancy of the unobservable model using measurements of the observable model output, with the consideration of uncertainty in the model parameters. After that, the uncertain model parameters are updated based on the modeled discrepancy function and experimental observations. Finally, the proposed framework quantifies the uncertainty in the prediction after Bayesian calibration by accounting for various sources of uncertainty in the multi-level model. The developed new approach is applied to the practical miter gate problem described in detail below to demonstrate its efficacy by simultaneously estimating the gate damage state and correcting the distributed boundary conditions. The main contributions of this paper can therefore be summarized as: (1) systematic formulation of the Bayesian calibration problem for the multi-level model with unobservable distributed response; (2) development of a two-phase machine learning method for model discrepancy quantification of the unobservable distributed response; (3) an approach to quantify uncertainty in the prediction after Bayesian calibration; and (4) application of the developed framework to the calibration of a boundary condition model of a miter gate problem.

The remainder of this paper is organized as follows. Section 2 provides background on Bayesian calibration of M&S and introduces the multi-level model (i.e. composition of multiple sub-models in a hierarchical architecture) with unobservable distributed response. Section 3 presents details of the proposed framework. An application to Bayesian calibration of the unobservable boundary condition model of a miter gate is used to demonstrate the proposed framework in Section 4, followed by conclusions in Section 5.

2 BACKGROUND

2.1. Bayesian Calibration of Modeling and Simulation Under Uncertainty

Model uncertainty (i.e. model discrepancy, or model bias, or model form error) can be caused by modelling assumptions, missing physics, and/or numerical discretization errors. In the KOH framework [20], model uncertainty is accounted for by relating a computational simulation model $y(\mathbf{x})=g(\mathbf{x},\boldsymbol{\theta})$ to the underlying true physical process $y^{true}(\mathbf{x})$ via

$$y^{true}(\mathbf{x})=\rho g(\mathbf{x},\boldsymbol{\theta}^{*})+\delta(\mathbf{x}), \tag{1}$$

where $\mathbf{x}$ are controllable input variables (e.g., experimental conditions), $\boldsymbol{\theta}^{*}$ are the underlying true, but unknown, model parameters (i.e. best estimates of model parameters in practical applications), ρ is an unknown regression coefficient that weighs the prediction of simulation model against the model discrepancy term during Bayesian calibration, and $\delta(\mathbf{x})$ is a model discrepancy term representing the mismatch between the model and reality. Note that $\boldsymbol{\theta}$ are fixed to specific values (i.e. $\boldsymbol{\theta}^*$) in numerical experiments in the case study section, which are unknown to $g(\mathbf{x}, \boldsymbol{\theta})$ in the Bayesian calibration process due to lack of knowledge to mimic reality.

A prior distribution $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$ is usually provided for $\boldsymbol{\theta}$ based on expert elicitation, physical constraints, or other knowledge, and the expected output of $g(\mathbf{x}, \boldsymbol{\theta})$ before Bayesian calibration is given by

$$\hat{y}(\mathbf{x}) = \int_{\boldsymbol{\theta}} g(\mathbf{x}, \boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \mathbf{d}\boldsymbol{\theta}. \tag{2}$$

To improve the accuracy of the prior prediction, several model calibration strategies have been developed to infer the uncertain parameters $\boldsymbol{\theta}$ and quantify the model discrepancy $\delta(\mathbf{x})$ using experimental data $\{\mathbf{x}^e,\ \mathbf{y}^e\}$. The experimental data can be related to the underlying true physics $y^{true}(\mathbf{x})$ as follows

$$y^e(\mathbf{x}) = y^{true}(\mathbf{x}) + \varepsilon, \tag{3}$$

where $y^e(\mathbf{x})$ is the experimental observation, and ε is the observation error, which is modeled as a Gaussian random variable here, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$.

For experiment settings $\mathbf{x}^e \doteq \{\mathbf{x}_{e,1}, \ldots, \mathbf{x}_{e,N_e}\}$, where N_e is the number of experimental data points, we have observed responses $\mathbf{y}^e \doteq \{y^e(\mathbf{x}_{e,1}), \ldots, y^e(\mathbf{x}_{e,N_e})\}$. Using $\mathbf{x}^e$, $\mathbf{y}^e$, and following the modularized Bayesian method [24], a surrogate model of $\hat{\delta}(\mathbf{x}, \boldsymbol{\omega}^*)$ with hyperparameters $\boldsymbol{\omega}^*$ may be constructed [36], and the regression coefficient ρ^* can be estimated. With these point estimates, the posterior distribution $f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta} \mid \mathbf{x}^e, \mathbf{y}^e, \rho^*, \boldsymbol{\omega}^*)$ may be obtained using Bayes' rule:

$$f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta} \mid \mathbf{x}^e, \mathbf{y}^e, \rho^*, \boldsymbol{\omega}^*) = \frac{f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}^e \mid \mathbf{x}^e, \boldsymbol{\theta}, \rho^*, \boldsymbol{\omega}^*)\, f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}{\int f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}^e \mid \mathbf{x}^e, \boldsymbol{\theta}, \rho^*, \boldsymbol{\omega}^*)\, f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) d\boldsymbol{\theta}}, \tag{4}$$

in which $f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}^e \mid \mathbf{x}^e, \boldsymbol{\theta}, \rho^*, \boldsymbol{\omega}^*)$ denotes the likelihood of observing $\mathbf{y}^e$ at $\mathbf{x}^e$ given values of $\boldsymbol{\theta}$. Note that the likelihood function is derived by accounting for the correlations between different observations through the Gaussian process in the KOH framework [20]. After Bayesian calibration, the expected model prediction from Eqs. (1) and (2) becomes

$$\hat{y}(\mathbf{x}) \mid \mathbf{x}^e, \mathbf{y}^e = \int_{\boldsymbol{\theta}} \rho^* g(\mathbf{x}, \boldsymbol{\theta})\, f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta} \mid \mathbf{x}^e, \mathbf{y}^e, \rho^*, \boldsymbol{\omega}^*) \mathbf{d}\boldsymbol{\theta} + \hat{\delta}(\mathbf{x}, \boldsymbol{\omega}^*). \tag{5}$$

The above Bayesian calibration scheme has been extensively studied and applied in many fields [37-40] due to its advantages in simultaneously correcting the model discrepancy and updating the uncertain model parameters. More details of existing Bayesian calibration methods can be found in [2, 22].

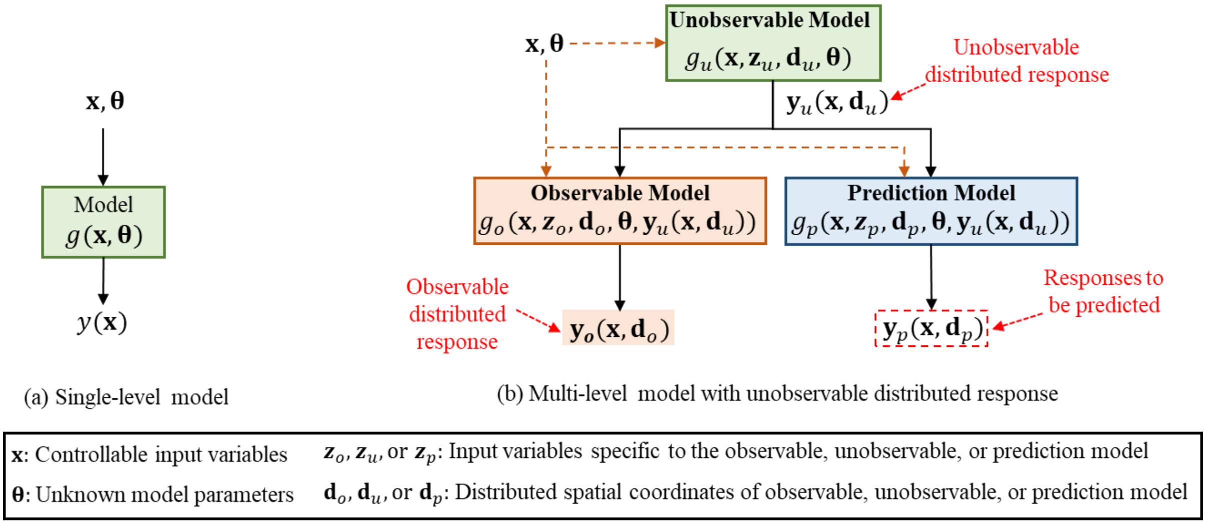
2.2. Multi-Level Model with Unobservable Distributed Response

Fig. 2 shows a comparison between a generalized single-level simulation model and a generalized multi-level model. For the sake of explanation and illustration, two-level models are used in this paper to explain the proposed framework, where the first level refers to the unobservable model and the second level refers to the observable model. The distributed responses of the unobservable model are part of the inputs of the observable model (i.e. the two models are cascaded). The proposed framework, however, extends naturally to problems with more than two levels and that setting will be further investigated in our future work.

The unobservable model with distributed responses is defined as

$$\mathbf{y}_u(\mathbf{x}, \mathbf{d}_u) = g_u(\mathbf{x}, \mathbf{z}_u, \mathbf{d}_u, \boldsymbol{\theta}) \tag{6}$$

where $\mathbf{x}$ and $\boldsymbol{\theta}$ are the controllable input variables and unknown model parameters, respectively, $\mathbf{y}_u(\mathbf{x},\mathbf{d}_u)$ is the unobservable distributed response, $\mathbf{z}_u$ are the inputs specific to the unobservable model, and $\mathbf{d}_u$ are the spatial coordinates of $\mathbf{y}_u(\mathbf{x},\mathbf{d}_u)$.



$\mathbf{x}, \boldsymbol{\theta}$
Model
$g(\mathbf{x}, \boldsymbol{\theta})$
$y(\mathbf{x})$
(a) Single-level model
$\mathbf{x}, \boldsymbol{\theta}$
Unobservable Model
$g_u(\mathbf{x}, \mathbf{z}_u, \mathbf{d}_u, \boldsymbol{\theta})$
Unobservable distributed response
$\mathbf{y}_u(\mathbf{x}, \mathbf{d}_u)$
Observable Model
$g_o(\mathbf{x}, \mathbf{z}_o, \mathbf{d}_o, \boldsymbol{\theta}, \mathbf{y}_u(\mathbf{x}, \mathbf{d}_u))$
Prediction Model
$g_p(\mathbf{x}, \mathbf{z}_p, \mathbf{d}_p, \boldsymbol{\theta}, \mathbf{y}_u(\mathbf{x}, \mathbf{d}_u))$
Observable distributed response
$\mathbf{y}_o(\mathbf{x}, \mathbf{d}_o)$
Responses to be predicted
$\mathbf{y}_p(\mathbf{x}, \mathbf{d}_p)$
(b) Multi-level model with unobservable distributed response
$\mathbf{x}$: Controllable input variables
$\boldsymbol{\theta}$: Unknown model parameters
$\mathbf{z}_o$, $\mathbf{z}_u$, or $\mathbf{z}_p$: Input variables specific to the observable, unobservable, or prediction model
$\mathbf{d}_o$, $\mathbf{d}_u$, or $\mathbf{d}_p$: Distributed spatial coordinates of observable, unobservable, or prediction model

Fig. 2 Comparison between single-level model and multi-level model with unobservable distributed response

Similarly, the observable model is defined as

$$\mathbf{y}_o(\mathbf{x},\mathbf{d}_o) = g_o(\mathbf{x}, \mathbf{z}_o, \mathbf{d}_o, \boldsymbol{\theta}, \mathbf{y}_u(\mathbf{x},\mathbf{d}_u)), \tag{7}$$

where $\mathbf{z}_o$ are the input variables specific to the observable model, $\mathbf{y}_o(\mathbf{x},\mathbf{d}_o)$ is the observable distributed response, and $\mathbf{d}_o$ are the spatial coordinates of $\mathbf{y}_o(\mathbf{x},\mathbf{d}_o)$.

For any input setting $\mathbf{x}$, we have $\mathbf{y}_u(\mathbf{x},\mathbf{d}_u) \doteq [y_u(\mathbf{x},\mathbf{d}_{u,1}), \cdots, y_u(\mathbf{x},\mathbf{d}_{u,N_u})]$, where N_u is the number of spatial coordinates in the response of $\mathbf{y}_u(\mathbf{x},\mathbf{d}_u)$, and $\mathbf{y}_o(\mathbf{x},\mathbf{d}_o) \doteq [y_o(\mathbf{x},\mathbf{d}_{o,1}), \cdots, y_o(\mathbf{x},\mathbf{d}_{o,N_o})]$, where N_o is the number of spatial coordinates in the response of $\mathbf{y}_o(\mathbf{x},\mathbf{d}_o)$.

The prediction model is defined similarly as

$$\mathbf{y}_p(\mathbf{x},\mathbf{d}_p) = g_p(\mathbf{x}, \mathbf{z}_p, \mathbf{d}_p, \boldsymbol{\theta}, \mathbf{y}_u(\mathbf{x},\mathbf{d}_u)), \tag{8}$$

where $\mathbf{z}_p$ and $\mathbf{d}_p$ denote the input variables and distributed spatial coordinates specific to the prediction model, $\mathbf{y}_p(\mathbf{x}, \mathbf{d}_p) \doteq [y_p(\mathbf{x}, \mathbf{d}_{p,1}), \cdots, y_p(\mathbf{x}, \mathbf{d}_{p,N_p})]$, and N_p is the number of spatial coordinates in the response of the prediction model. Note that $\mathbf{y}_u(\mathbf{x}, \mathbf{d}_u)$ could be an input to multiple prediction models. For the sake of illustration, only one model is used in this paper.

Multi-level model problems are very common in practical applications [41-43]. For instance, as shown in Fig. 1, we need to use the observable strain response to calibrate the unobservable distributed boundary force response of a boundary condition model (i.e. unobservable model), to improve the prediction accuracy of the fatigue crack initiation (i.e. prediction model) on the contact surface of the quoin block [28]. As shown in Fig. 2, the accuracy of the unobservable model plays an important role in that of the prediction model, but it is affected by the uncertain model parameters $\boldsymbol{\theta}$ and potential model discrepancy. Since it is not directly measured and the response is distributed, standard Bayesian approaches for single-level models, like those reviewed in Sec. 2.1, cannot be directly applied to multi-level models. In order to increase the accuracy of the prediction model $\mathbf{y}_p(\mathbf{x}, \mathbf{d}_p)$, a new Bayesian calibration approach is required to calibrate $\mathbf{y}_u(\mathbf{x}, \mathbf{d}_u)$ and estimate the unknown model parameters $\boldsymbol{\theta}$ using measurements of the observable model $\mathbf{y}_o(\mathbf{x}, \mathbf{d}_o)$. The next section discusses the details of our approach to solving this multi-level Bayesian calibration problem.

3 BAYESIAN CALIBRATION OF MULTI-LEVEL MODEL WITH UNOBSERVABLE DISTRIBUTED RESPONSE (BA-MUDI)

3.1 Problem Formulation

If model discrepancy was ignored, the composition of the observable and unobservable models in Eq. (7) could be used directly in a single-level calibration framework. However, accounting for model discrepancy in the unobservable model makes the calibration process more complicated. This is exacerbated by the unobservable model's high dimensional distributed response. Similar to the discrepancy for a single-level model, we define the true physics of the unobservable response as follows

$$\mathbf{y}_u^{true}(\mathbf{x},\mathbf{d}_u)=\rho g_u(\mathbf{x},\mathbf{z}_u,\mathbf{d}_u,\boldsymbol{\theta}^*)+\delta_u(\mathbf{x},\mathbf{z}_u,\mathbf{d}_u), \tag{9}$$

where $\delta_u(\mathbf{x},\mathbf{z}_u,\mathbf{d}_u)$ is the discrepancy function of the unobservable model, ρ is the unknown regression coefficient, and $\boldsymbol{\theta}^*$ is the true (or best in a real-world problem) values of model parameters but unknown to $g_u(\mathbf{x},\mathbf{z}_u,\mathbf{d}_u,\boldsymbol{\theta})$.

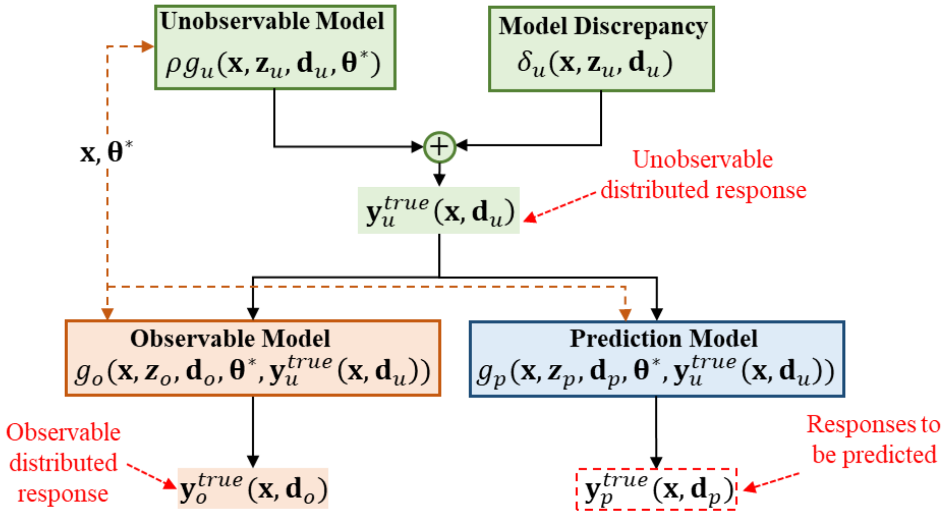
The observable model and prediction model after the correction of the unobservable model are given by

$$\mathbf{y}_o^{true}(\mathbf{x},\mathbf{d}_o)=g_o(\mathbf{x},\mathbf{z}_o,\mathbf{d}_o,\boldsymbol{\theta}^*,\mathbf{y}_u^{true}(\mathbf{x},\mathbf{d}_u)), \tag{10}$$

and

$$\mathbf{y}_p^{true}(\mathbf{x},\mathbf{d}_p)=g_p(\mathbf{x},\mathbf{z}_p,\mathbf{d}_p,\boldsymbol{\theta}^*,\mathbf{y}_u^{true}(\mathbf{x},\mathbf{d}_u)). \tag{11}$$

Fig. 3 shows the connections between different models after model correction of the unobservable model. Since the output of the observable model can be measured and through experimental data collected in a controlled experimental environment in which the inputs can be controlled (e.g. apply known boundary forces to the miter gate), its model discrepancy term can be pre-calibrated by following existing model calibration approaches for single-level models [22] as discussed in Section 2.1. Based on this observation and for explanation sake, we will not explicitly account for observable model discrepancy and will assume that either the observable model is accurate or the model discrepancy term has been incorporated into $\mathbf{y}_o(\mathbf{x},\mathbf{d}_o)$. The main uncertainty of the observable model thus stems from the uncertain inputs of the observable model, such as the random input variables, the uncertain distributed response of the unobservable model, and the unknown model parameters $\boldsymbol{\theta}$.



Unobservable Model
$\rho g_u(\mathbf{x}, \mathbf{z}_u, \mathbf{d}_u, \boldsymbol{\theta}^*)$
Model Discrepancy
$\delta_u(\mathbf{x}, \mathbf{z}_u, \mathbf{d}_u)$
$\mathbf{x}, \boldsymbol{\theta}^*$
Unobservable distributed response
$\mathbf{y}_u^{true}(\mathbf{x}, \mathbf{d}_u)$
Observable Model
$g_o(\mathbf{x}, \mathbf{z}_o, \mathbf{d}_o, \boldsymbol{\theta}^*, \mathbf{y}_u^{true}(\mathbf{x}, \mathbf{d}_u))$
Prediction Model
$g_p(\mathbf{x}, \mathbf{z}_p, \mathbf{d}_p, \boldsymbol{\theta}^*, \mathbf{y}_u^{true}(\mathbf{x}, \mathbf{d}_u))$
Observable distributed response
$\mathbf{y}_o^{true}(\mathbf{x}, \mathbf{d}_o)$
Responses to be predicted
$\mathbf{y}_p^{true}(\mathbf{x}, \mathbf{d}_p)$

Fig. 3 Multi-level model with unobservable distributed response after model correction

Based on the above definitions, we can represent observations of the observable model as

$$\mathbf{y}_o^e(\mathbf{x}, \mathbf{d}_o) = g_o(\mathbf{x}, \mathbf{z}_o, \mathbf{d}_o, \boldsymbol{\theta}^*, \mathbf{y}_u^{true}(\mathbf{x}, \mathbf{d}_u)) + \boldsymbol{\varepsilon}(\mathbf{d}_o), \tag{12}$$

where $\boldsymbol{\varepsilon}(\mathbf{d}_o) \doteq [\varepsilon(\mathbf{d}_{o,1}), \cdots, \varepsilon(\mathbf{d}_{o,N_o})]$ are the measurement errors at different spatial locations, $\varepsilon(\mathbf{d}_{o,q}) \sim N(0, \sigma_\varepsilon^2), \forall q = 1, 2, \cdots, N_o$ are modeled as statistically independent identically distributed Gaussian random variables, and $\mathbf{y}_o^e(\mathbf{x}, \mathbf{d}_o) \doteq [y_o^e(\mathbf{x}, \mathbf{d}_{o,1}), \cdots, y_o^e(\mathbf{x}, \mathbf{d}_{o,N_o})]$. Note that $\mathbf{y}_o^e(\mathbf{x}, \mathbf{d}_o)$ is different from $\mathbf{y}_o(\mathbf{x}, \mathbf{d}_o)$ due to the difference between $\mathbf{y}_u(\mathbf{x}, \mathbf{d}_u)$ and $\mathbf{y}_u^{true}(\mathbf{x}, \mathbf{d}_u)$ caused by the unobservable model discrepancy.

For fixed experimental settings $\mathbf{x}^e \doteq \{\mathbf{x}_{e,1}, \ldots, \mathbf{x}_{e,N_e}\}$, the multilevel Bayesian calibration problem considered here can be summarized as

$$\begin{aligned} &\textbf{Given}: \mathbf{x}_{e,i} \text{ and } \mathbf{y}_o^e(\mathbf{x}_{e,i}, \mathbf{d}_o),\ i = 1, \cdots, N_e, \\ &\textbf{Find}: \text{model discrepancy } \delta_u(\mathbf{x}, \mathbf{z}_u, \mathbf{d}_u) \text{ of the unobservable} \\ &\qquad \text{distributed response } \mathbf{y}_u(\mathbf{x}, \mathbf{d}_u), \text{ and unknown model parameters } \boldsymbol{\theta}. \end{aligned} \tag{13}$$

For the sake of explanation, in the subsequent sections, we use $g_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\theta})$, $\delta_u(\mathbf{x}, \mathbf{d}_u)$, $g_o(\mathbf{x}, \mathbf{d}_o, \boldsymbol{\theta}, \mathbf{y}_u(\mathbf{x}, \mathbf{d}_u))$ and $g_p(\mathbf{x}, \mathbf{d}_p, \boldsymbol{\theta}, \mathbf{y}_u(\mathbf{x}, \mathbf{d}_u))$ to represent $g_u(\mathbf{x}, \mathbf{z}_u, \mathbf{d}_u, \boldsymbol{\theta})$, $\delta_u(\mathbf{x}, \mathbf{z}_u, \mathbf{d}_u)$, $g_o(\mathbf{x}, \mathbf{z}_o, \mathbf{d}_o, \boldsymbol{\theta}, \mathbf{y}_u(\mathbf{x}, \mathbf{d}_u))$ and $g_p(\mathbf{x}, \mathbf{z}_p, \mathbf{d}_p, \boldsymbol{\theta}, \mathbf{y}_u(\mathbf{x}, \mathbf{d}_u))$ by omitting variables $\mathbf{z}$, which are specific to individual models. Next, an overview of the proposed framework is presented followed by details on this framework that answers the research question posed in Eq. (13).

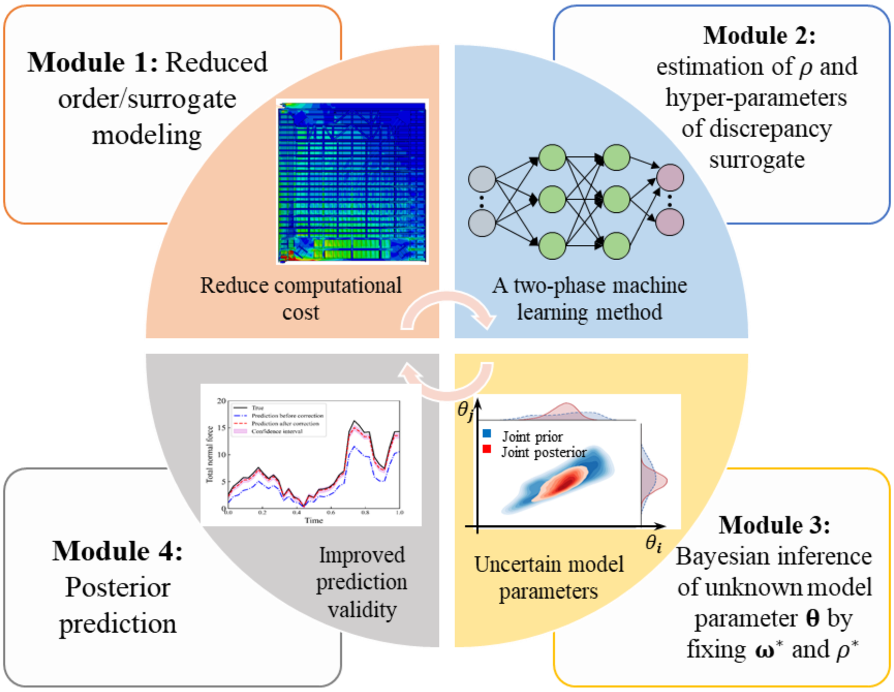
3.2 Overview of BA-MUDI

In the proposed BAyesian calibration of Multi-level model with Unobservable DIstributed response (BA-MUDI) framework, in order to improve the prediction accuracy of $g_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\theta})$ at new input conditions of $\mathbf{x}$ in a manner similar to Bayesian calibration of single-level models, we construct a surrogate model of $\delta_u(\mathbf{x}, \mathbf{d}_u)$ as $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$, where $\boldsymbol{\omega}$ are the hyper-parameters of the surrogate model. Replacing $\delta_u(\mathbf{x}, \mathbf{d}_u)$ with $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$ in Eq. (9), we have the corrected $\mathbf{y}_u(\mathbf{x}, \mathbf{d}_u)$ conditioned on ρ, $\boldsymbol{\theta}$, and $\boldsymbol{\omega}$ as

$$\mathbf{y}_u^{ct}(\mathbf{x}, \mathbf{d}_u) \mid \rho, \boldsymbol{\omega}, \boldsymbol{\theta} = \rho g_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\theta}) + \hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega}), \tag{14}$$

where $\mathbf{y}_u^{ct}(\mathbf{x}, \mathbf{d}_u)$ is the corrected $\mathbf{y}_u(\mathbf{x}, \mathbf{d}_u)$, and the corrected observable model and prediction models can be updated accordingly based on Eq. (10) and Eq. (11) by replacing $\mathbf{y}_u^{true}(\mathbf{x}, \mathbf{d}_u)$ with $\mathbf{y}_u^{ct}(\mathbf{x}, \mathbf{d}_u) \mid \rho, \boldsymbol{\omega}, \boldsymbol{\theta}$. Note that the right-hand side of Eq. (14), i.e. unobservable model corrected with the calibrated model discrepancy term, is referred to hereafter as the “corrected model”.

The task of Bayesian calibration of multi-level model is then to estimate $\boldsymbol{\omega}$ and ρ, and infer $\boldsymbol{\theta}$ using $\mathbf{x}_{e,i}$ and $\mathbf{y}_o^e(\mathbf{x}_{e,i}, \mathbf{d}_o), i = 1, \cdots, N_e$. Similar to Bayesian calibration of single-level models, there are many different ways of estimating these parameters, such as full Bayesian analysis [44], and optimization-based approaches [3]. As suggested in the KOH framework [20], the full Bayesian method is computationally intractable, so in this paper we adopt the modularized Bayesian strategy [8, 24]. Fig. 4 presents an overview of the proposed BA-MUDI framework for calibration of multi-level models with unobservable distributed response.



Module 1: Reduced order/surrogate modeling
Reduce computational cost
Module 2: estimation of ρ and hyper-parameters of discrepancy surrogate
A two-phase machine learning method
Module 3: Bayesian inference of unknown model parameter $\boldsymbol{\theta}$ by fixing $\boldsymbol{\omega}^*$ and ρ^*
Uncertain model parameters
Joint prior
Joint posterior
θ_j
θ_i
Module 4: Posterior prediction
Improved prediction validity
True
Prediction before correction
Prediction after correction
Confidence interval
Total normal force
Time

Fig. 4 Overview of proposed BA-MUDI framework

As shown in Fig. 4, the proposed BA-MUDI framework consists of four main modules which are briefly explained as below

(1) *Module one - reduced-order modeling or surrogate modeling of* $g_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\theta})$ and $g_o(\mathbf{x}, \mathbf{d}_o, \boldsymbol{\theta}, \mathbf{y}_u(\mathbf{x}, \mathbf{d}_u))$: In the case that $g_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\theta})$ and/or $g_o(\mathbf{x}, \mathbf{d}_o, \boldsymbol{\theta}, \mathbf{y}_u(\mathbf{x}, \mathbf{d}_u))$ are computationally expensive simulation models, computationally “cheap” surrogate models or reduced-order models are first constructed to replace the original models. For instance, in the miter gate application, a reduced-order model is constructed for $g_o(\mathbf{x}, \mathbf{d}_o, \boldsymbol{\theta}, \mathbf{y}_u(\mathbf{x}, \mathbf{d}_u))$ using static condensation to reduce the computational effort [29]. However, in general, alternative methods, such as surrogate models with deterministic predictions (e.g. neural networks [45], support-vector regression [46], etc) are available.

(2) *Module two - estimation of* ρ and *surrogate modeling of* $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$: This module estimates the value of ρ and constructs a surrogate model $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$ for the model

discrepancy term of the unobservable model $g_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\theta})$ with distributed response, using observations $\{\mathbf{x}^e, \mathbf{y}_o^e\}$ of the observable model $g_o(\mathbf{x}, \mathbf{z}_o, \mathbf{d}_o, \boldsymbol{\theta}^*, \mathbf{y}_u^{true}(\mathbf{x}, \mathbf{d}_u))$.

(3) *Module three - Bayesian inference of uncertain model parameter* $\boldsymbol{\theta}$: This module performs Bayesian updating of the uncertain model parameters $\boldsymbol{\theta}$ by fixing the hyper-parameters of $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$ at $\boldsymbol{\omega} = \boldsymbol{\omega}^*$ and ρ at $\rho = \rho^*$, which are obtained from Module two. The main challenge that needs to be solved in this module is how to account for the uncertainty of $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$ in the inference of $\boldsymbol{\theta}$.

(4) *Module four - Posterior prediction after Bayesian calibration*: Based on the discrepancy surrogate model and ρ^* obtained from Module two and posterior distribution of $\boldsymbol{\theta}$ obtained from Module three, this module performs posterior prediction after Bayesian calibration.

Although the above four modules are similar to that for Bayesian calibration of single-level model, the challenges that are solved in this paper and the implementation details are completely different due to the presence of distributed response and the unobservable model in Bayesian calibration. Efforts have been made in the past for Bayesian calibration of either an unobservable model or a distributed response separately; no approach has been developed for problems with unobservable and distributed responses which is a very common and challenging issue in many engineering applications. To the authors' knowledge, the proposed solutions to the challenging problem of Bayesian calibration with unobservable and distributed responses presented in this paper cannot be found in the literature. The novelty of the paper is multifold. First, it creates a meta-algorithm that builds upon and synthesizes multiple existing algorithms including Bayesian inference, machine learning, and model discrepancy quantification in an innovative architecture to solve a new challenging problem involving composition of multiple models in a multi-level structure. Second, it proposes a creative way to solve a coupled optimization problems (i.e. upper-level and lower-level optimization problems) in Bayesian calibration using a two-phase machine learning method. Third, it demonstrates the proposed method using a real-world miter gate application problem where the contact force analysis model with distributed response needs to be calibrated. In the subsequent sections, we will present the proposed framework in detail with a focus on Modules two to four since Module one is relatively straightforward.

3.3 Module Two: Estimating ρ and Surrogate Modeling of $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$

As discussed above, the model discrepancy of the unobservable model $g_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\theta})$ is a function with distributed response. It is difficult to directly construct such a surrogate model $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$ since we cannot measure the distributed response of the unobservable model in practice. A possible solution could be mapping measurements of the observable model to the response of the unobservable model for training of $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$ and estimation of ρ. However, such a mapping is also difficult because the spatial coordinates of the observable model and unobservable model are different and the observable model is generally not invertible. Thus, we do not directly have training data for training of $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$. The coupling (compensation effects) between ρ, $\boldsymbol{\theta}$, and $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$ further complicates the model discrepancy quantification of $g_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\theta})$. This module addresses these challenges through the representation of the spatially distributed response using orthogonal polynomials and the estimation of the input-dependent polynomial coefficients with a two-phase machine learning method.

3.3.1 Distributed discrepancy representation

In order to make it possible to build a surrogate model for $\delta_u(\mathbf{x}, \mathbf{d}_u)$ given in Eq. (9), we separate the space-dependent response from the input-dependent response in the model discrepancy term using orthogonal polynomials as shown below, by following the space-time domain decoupling strategy suggested by [47, 48]

$$\delta_u(\mathbf{x}, \mathbf{d}_u) = \sum_{k=1}^{N_k} \alpha_k(\mathbf{x}) \mathbf{L}_k(\mathbf{d}_u) = \mathbf{L}(\mathbf{d}_u) \boldsymbol{\alpha}(\mathbf{x}), \tag{15}$$

where N_k is the number of orthogonal polynomial terms, $\mathbf{L}_k(\mathbf{d}_u) = [L_k(\mathbf{d}_{u,1}), \ldots, L_k(\mathbf{d}_{u,N_u})]^T \in \mathbb{R}^{N_u \times 1}$ is the k-th order polynomial basis, $\mathbf{L}(\mathbf{d}_u) = [\mathbf{L}_1(\mathbf{d}_u), \ldots, \mathbf{L}_{N_k}(\mathbf{d}_u)] \in \mathbb{R}^{N_u \times N_k}$ is an $N_u \times N_k$ orthogonal polynomial matrix, and $\boldsymbol{\alpha}(\mathbf{x}) = [\alpha_1(\mathbf{x}), \ldots, \alpha_{N_k}(\mathbf{x})]^T \in \mathbb{R}^{N_k \times 1}$ are the input-dependent coefficients. Note that this is similar to the Karhunen-Loève decomposition commonly employed with random processes.

Based on the above representation, we can then construct surrogates for the coefficients $\alpha_k(\mathbf{x}), \forall k = 1, \cdots, N_k$. Letting the surrogate of $\alpha_k(\mathbf{x})$ be $\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k), \forall k = 1, \cdots, N_k$, where $\boldsymbol{\upsilon}_k$ are the hyper-parameters of the k-th surrogate model, we then have $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$ as

$$\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega}) = \sum_{k=1}^{N_k} \hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k) \mathbf{L}_k(\mathbf{d}_u), \tag{16}$$

where $\boldsymbol{\omega} = [\boldsymbol{\upsilon}_1, \cdots, \boldsymbol{\upsilon}_{N_k}]$ are the hyperparameters of the surrogate models.

In this paper, Legendre orthogonal polynomials [49] are employed for $\mathbf{L}_k(\mathbf{d}_u), \forall k = 1, \cdots, N_k$. Alternatives, such as Hermite polynomials and Chebyshev polynomials, can also be used [47].

Using the polynomial representation of $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$, the corrected unobservable model in Eq. (14) is rewritten as

$$\mathbf{y}_u^{ct}(\mathbf{x}, \mathbf{d}_u) \mid \rho, \boldsymbol{\omega}, \boldsymbol{\theta} = \rho g_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\theta}) + \sum_{k=1}^{N_k} \hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_i) \mathbf{L}_k(\mathbf{d}_u). \tag{17}$$

Substituting Eq. (17) into Eq. (12), we have the observable model after correction of the unobservable model,

$$\hat{\mathbf{y}}_o(\mathbf{x}, \mathbf{d}_o) \mid \rho, \boldsymbol{\omega}, \boldsymbol{\theta} = g_o(\mathbf{x}, \mathbf{d}_o, \boldsymbol{\theta}, \rho g_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\theta}) + \sum_{k=1}^{N_k} \hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k) \mathbf{L}_k(\mathbf{d}_u)) + \boldsymbol{\varepsilon}(\mathbf{d}_o). \tag{18}$$

The task now is to estimate the hyper-parameters $\boldsymbol{\omega}$ and the regression coefficient ρ using observations of the observable model output $\{\mathbf{x}^e, \mathbf{y}_o^e\}$, where $\mathbf{y}_o^e \triangleq \{\mathbf{y}_{o,1}^e, \cdots, \mathbf{y}_{o,N_e}^e\}$ with $\mathbf{y}_{o,i}^e \doteq \mathbf{y}_o^e(\mathbf{x}_{e,i}, \mathbf{d}_o), \forall i = 1, \cdots, N_e$. This however, is still very challenging for two main reasons: (1) $\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k)$ can be any complicated function of $\mathbf{x}$, and the values of the coefficients $\hat{\alpha}_k(\mathbf{x}_{e,i}, \boldsymbol{\upsilon}_k)$ are unknown for any observation $\{\mathbf{x}_{e,i}, \mathbf{y}_{o,i}^e\}, \forall i = 1, \cdots, N_e$, of the observable model; and (2) $\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k), \forall k = 1, \cdots, N_k$ are also affected by the unknown regression coefficient ρ and the unknown model parameters $\boldsymbol{\theta}$. This paper presents a novel approach to solve the above challenges through the creation of a two-phase machine learning method. The proposed approach as described below fills a gap in the existing literature and makes it possible to simultaneously calibrate unknown model parameters and model discrepancy terms for models with unobservable and distributed responses.

To address the above issues in the surrogate modeling of $\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k), \forall k = 1, \cdots, N_k$, in the following Section 3.3.2, we first estimate the unknown regression coefficient ρ and the polynomial coefficients at the specific experimental conditions where we have observations $\hat{\boldsymbol{\alpha}}^e = [\hat{\boldsymbol{\alpha}}_{e,1}, \hat{\boldsymbol{\alpha}}_{e,2}, \cdots, \hat{\boldsymbol{\alpha}}_{e,N_e}] \in \mathbb{R}^{N_k \times N_e}$, where $\hat{\boldsymbol{\alpha}}_{e,i} = [\hat{\alpha}_{e,i}^{(1)}, \cdots, \hat{\alpha}_{e,i}^{(N_k)}]^T$, $\forall i = 1, \cdots, N_e$ for the given observation $\{\mathbf{x}_{e,i}, \mathbf{y}_{o,i}^e\}$. After that, we construct surrogate models of $\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k), \forall k = 1, \cdots, N_k$ based on the estimated $\hat{\boldsymbol{\alpha}}^e$ and the observations of $\{\mathbf{x}^e, \mathbf{y}_o^e\}$. The surrogate models enable the polynomial coefficients to be computed for conditions $\mathbf{x}$ that are not included in the experiment data. Next, we will discuss how to efficiently estimate ρ and $\hat{\boldsymbol{\alpha}}^e$ for given observations $\{\mathbf{x}^e, \mathbf{y}_o^e\}$.

3.3.2 Estimation of ρ and polynomial coefficients $\hat{\boldsymbol{\alpha}}^e$ based on observations

A straightforward way of estimating ρ along with $\hat{\boldsymbol{\alpha}}^e$ based on $\{\mathbf{x}^e, \mathbf{y}_o^e\}$ is to use the maximum likelihood estimation (MLE) method as below

$$\hat{\boldsymbol{\alpha}}^e, \rho^* = \arg\max_{\boldsymbol{\alpha}^e, \rho}\{L(\mathbf{y}_o^e \,|\, \mathbf{x}_e, \boldsymbol{\alpha}^e, \rho)\}, \tag{19}$$

where $L(\mathbf{y}_o^e \,|\, \mathbf{x}_e, \boldsymbol{\alpha}^e, \rho)$ is the likelihood function of observing $\mathbf{y}_o^e$ for given $\boldsymbol{\alpha}^e$ and ρ, and is given by

$$L(\mathbf{y}_o^e \,|\, \mathbf{x}_e, \boldsymbol{\alpha}^e, \rho) = \prod_{i=1}^{N_e} L(\mathbf{y}_{o,i}^e \,|\, \mathbf{x}_{e,i}, \boldsymbol{\alpha}_{e,i}, \rho). \tag{20}$$

As shown in the above two equations, the simultaneous estimation of $\hat{\boldsymbol{\alpha}}^e = [\hat{\boldsymbol{\alpha}}_{e,1}, \cdots, \hat{\boldsymbol{\alpha}}_{e,N_e}]$ and ρ using the MLE method or a Bayesian method is computationally intractable since the dimension is very high (i.e. the dimension is $N_e \times (2 \times N_k) + 1 = 6001$ for the miter gate problem). To address this issue, we transform the MLE model given in Eq. (19) into a two-level optimization problem with multiple lower-dimensional optimization problems at the lower level. The top-level optimization problem takes the form

$$\begin{aligned} &\rho^* = \arg\max_{\rho}\{L(\mathbf{y}_o^e \,|\, \mathbf{x}_e, \hat{\boldsymbol{\alpha}}^e, \rho)\}, \\ &\text{s.t. } \hat{\boldsymbol{\alpha}}^e = \arg\max_{\boldsymbol{\alpha}^e}\{L(\mathbf{y}_o^e \,|\, \mathbf{x}_e, \boldsymbol{\alpha}^e, \rho)\}, \end{aligned} \tag{21}$$

where $\hat{\boldsymbol{\alpha}}^e = \arg\max_{\boldsymbol{\alpha}^e}\{L(\mathbf{y}_o^e \,|\, \mathbf{x}_e, \boldsymbol{\alpha}^e, \rho)\}$ is a lower-level optimization model for a given ρ, which can be further written as

$$\hat{\boldsymbol{\alpha}}^e = \arg\max_{\boldsymbol{\alpha}^e}\left\{\sum_{i=1}^{N_e}\log\left(L(\mathbf{y}_{o,i}^e \,|\, \mathbf{x}_{e,i}, \boldsymbol{\alpha}_{e,i}, \rho)\right)\right\}. \tag{22}$$

Since the likelihood functions $L(\mathbf{y}_{o,i}^e \,|\, \mathbf{x}_{e,i}, \boldsymbol{\alpha}_{e,i}, \rho), \forall i = 1, \cdots, N_e$ are independent from each other for a given ρ, the lower-level optimization model given in Eq. (22) can be further decomposed into N_e independent low-dimensional (each of dimension N_k) optimization models as follows

$$\hat{\boldsymbol{\alpha}}_{e,i} = \underset{\boldsymbol{\alpha}_{e,i}}{\arg\max} \left\{ \log\left(L(\mathbf{y}^{e}_{o,i} \mid \mathbf{x}_{e,i}, \boldsymbol{\alpha}_{e,i}, \rho) \right) \right\}, \forall i = 1, \cdots, N_e, \tag{23}$$

where $L(\mathbf{y}^{e}_{o,i} \mid \mathbf{x}_{e,i}, \boldsymbol{\alpha}_{e,i}, \rho)$ is given by

$$L(\mathbf{y}^{e}_{o,i} \mid \mathbf{x}_{e,i}, \boldsymbol{\alpha}_{e,i}, \rho) = \prod_{q=1}^{N_o} \phi\left(\frac{y^{e}_{o}(\mathbf{x}_{e,i}, \mathbf{d}_{o,q}) - \left[\mu_{y^{e}_{o}}(\mathbf{x}_{e,i}, \mathbf{d}_{o,q}) \mid \boldsymbol{\alpha}_{e,i}, \rho \right]}{\sigma_{\varepsilon}(\mathbf{d}_{o,q})} \right), \tag{24}$$

in which $\sigma_{\varepsilon}(\mathbf{d}_{o,q})$ $(\sigma_{\varepsilon}(\mathbf{d}_{o,q}) = \sigma_{\varepsilon}, \forall q = 1, \cdots, N_o)$ is the standard deviation of $\varepsilon(\mathbf{d}_{o,q})$ at the q-th spatial coordinate of the observable model, $\mu_{y^{e}_{o}}(\mathbf{x}_{e,i}, \mathbf{d}_{o,q}) \mid \boldsymbol{\alpha}_{e,i}, \rho$ is the mean prediction based on Eq. (18) at the q-th spatial coordinate of the observable model, and it is computed by integrating out $\boldsymbol{\theta}$ based on its prior distribution $f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ as

$$\begin{aligned} \mu_{y^{e}_{o}}(\mathbf{x}_{e,i}, \mathbf{d}_{o,q}) \mid \boldsymbol{\alpha}_{e,i}, \rho &= \int [\mu_{y^{e}_{o}}(\mathbf{x}_{e,i}, \mathbf{d}_{o,q}) \mid \boldsymbol{\alpha}_{e,i}, \boldsymbol{\theta}, \rho] f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \mathbf{d}\boldsymbol{\theta}, \\ &= \int [g_o(\mathbf{x}_{e,i}, \mathbf{d}_{o,q}, \boldsymbol{\theta}, \rho g_u(\mathbf{x}_{e,i}, \mathbf{d}_u, \boldsymbol{\theta}) + \mathbf{L}(\mathbf{d}_u)\boldsymbol{\alpha}_{e,i})] f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \mathbf{d}\boldsymbol{\theta}, \end{aligned} \tag{25}$$

in which $\mathbf{L}(\mathbf{d}_u) \in \mathbb{R}^{N_u \times N_k}$ is given in Sec. 3.3.1 and $\boldsymbol{\alpha}_{e,i} \in \mathbb{R}^{N_k \times 1}$. The Monte Carlo simulation (MCS) method [50] is employed to solve this integration in this paper. For higher-dimensional integration as shown in Eq. (25), the MCS-based integration method could lead to large errors. In that case, advanced numerical integration methods, such as sparse grid [51], dimension-reduction-based approaches [52], can be employed.

Substituting Eq. (25) into Eq. (23), the lower-level optimization models become

$$\begin{aligned} \hat{\boldsymbol{\alpha}}_{e,i} &= \underset{\boldsymbol{\alpha}_{e,i}}{\arg\min} \left\{ \sum_{q=1}^{N_o} \left(\frac{y^{e}_{o}(\mathbf{x}_{e,i}, \mathbf{d}_{o,q}) - \left[\mu_{y^{e}_{o}}(\mathbf{x}_{e,i}, \mathbf{d}_{o,q}) \mid \boldsymbol{\alpha}_{e,i}, \rho \right]}{\sigma_{\varepsilon}(\mathbf{d}_{o,q})} \right)^2 \right\}, \\ &= \underset{\boldsymbol{\alpha}_{e,i}}{\arg\min} \left\{ \left\| \mathbf{y}^{e}_{o,i} - \left[\boldsymbol{\mu}_{\mathbf{y}^{e}_{o}}(\mathbf{x}_{e,i}, \mathbf{d}_{o}) \mid \boldsymbol{\alpha}_{e,i}, \rho \right] \right\|_2^2 \right\}, \end{aligned} \tag{26}$$

where $\|\cdot\|_2$ is the Euclidean norm,

$\left[\boldsymbol{\mu}_{\mathbf{y}_o^e}(\mathbf{x}_{e,i},\mathbf{d}_o)\,|\,\boldsymbol{\alpha}_{e,i},\rho\right]=\{\mu_{y_o^e}(\mathbf{x}_{e,i},\mathbf{d}_{o,q})\,|\,\boldsymbol{\alpha}_{e,i},\rho,\forall q=1,\cdots,N_o\}$, and $\mu_{y_o^e}(\mathbf{x}_{e,i},\mathbf{d}_{o,q})\,|\,\boldsymbol{\alpha}_{e,i},\rho$ is given in Eq. (25).

The above equation indicates that a MLE value of $\hat{\boldsymbol{\alpha}}_{e,i}$ will minimize the distance between $\mathbf{y}_{o,i}^e$ and $\boldsymbol{\mu}_{\mathbf{y}_o^e}(\mathbf{x}_{e,i},\mathbf{d}_o)\,|\,\boldsymbol{\alpha}_{e,i},\rho$. Even though the MLE estimate of $\hat{\boldsymbol{\alpha}}_{e,i}$ is possible, repeatedly solving this multi-level optimization problem is still computationally expensive because the lower-level MLE problem needs to be solved multiple times to determine the optimal value ρ^* at the upper level. Since the distance between $\mathbf{y}_{o,i}^e$ and $\boldsymbol{\mu}_{\mathbf{y}_o^e}(\mathbf{x}_{e,i},\mathbf{d}_o)\,|\,\boldsymbol{\alpha}_{e,i},\rho$ will be minimized at the MLE point, it implies that we can approximate $\mathbf{y}_{o,i}^e$ using $\boldsymbol{\mu}_{\mathbf{y}_o^e}(\mathbf{x}_{e,i},\mathbf{d}_o)\,|\,\boldsymbol{\alpha}_{e,i},\rho$ at the MLE point. In other words, we can build an approximate relationship between $\mathbf{y}_{o,i}^e$ and $\hat{\boldsymbol{\alpha}}_{e,i}$ using $\boldsymbol{\mu}_{\mathbf{y}_o^e}(\mathbf{x}_{e,i},\mathbf{d}_o)\,|\,\boldsymbol{\alpha}_{e,i},\rho$. Based on this observation and in order to avoid repeatedly solving the model given by Eq. (26), we develop a machine learning-based method as shown in Fig. 5 to efficiently estimate $\hat{\boldsymbol{\alpha}}_{e,i}$ for any given observation $\{\mathbf{x}_{e,i},\mathbf{y}_{o,i}^e,\forall i=1,\cdots,N_e\}$. As shown in Fig. 5 (b), there are two main phases in the machine learning-based method, namely the offline-training phase and online estimation phase. The offline-training phase is conducted to train an approximated mapping between the mean predictions of observable responses $\boldsymbol{\mu}_{\mathbf{y}_o^e}(\mathbf{x}_{e,i},\mathbf{d}_o)\,|\,\boldsymbol{\alpha}_{e,i},\rho$ and the coefficients $\hat{\boldsymbol{\alpha}}_{e,i}$ of the distributed model discrepancy. For any given measurements, the online estimation phase is to predict the corresponding coefficients of the distributed model discrepancy during model calibration using the machine learning model trained in the off-line phase. Since the required computational time for the machine learning model prediction is usually very low, this makes it possible to efficiently estimate the unknow coefficients $\hat{\boldsymbol{\alpha}}_{e,i}$ in an online manner.

(a) Offline training of machine learning model

In the offline phase, a machine learning model $\boldsymbol{\alpha} = f_{ML}(\rho, \mathbf{x}, \boldsymbol{\mu}_o(\mathbf{x}, \boldsymbol{\alpha}, \rho))$ is trained according to the explanations as discussed in Eq. (20) through (26), where $\boldsymbol{\mu}_o(\mathbf{x}, \boldsymbol{\alpha}, \rho) = \left[\boldsymbol{\mu}_{\mathbf{y}_o^e}(\mathbf{x}, \mathbf{d}_o) \,|\, \boldsymbol{\alpha}, \rho\right]$. In order to train such a machine learning model, we first generate training samples of ρ, $\mathbf{x}$, and $\boldsymbol{\alpha}$. Defining the training samples as $\boldsymbol{\rho}_t = \{\rho_{t,c}, c = 1, \cdots, N_t\}$, $\mathbf{x}_t = \{\mathbf{x}_{t,c}, c = 1, \cdots, N_t\}$, and $\boldsymbol{\alpha}_t = \{\boldsymbol{\alpha}_{t,c}, c = 1, \cdots, N_t\}$, where N_t is the number of training points, for each sample of $\rho_{t,c}$, $\mathbf{x}_{t,c}$, and $\boldsymbol{\alpha}_{t,c}$, $\forall c = 1, \cdots, N_t$, we obtain $\boldsymbol{\mu}_o(\mathbf{x}_{t,c}, \boldsymbol{\alpha}_{t,c}, \rho_{t,c})$ using Eq. (25) as

$$\boldsymbol{\mu}_o(\mathbf{x}_{t,c}, \boldsymbol{\alpha}_{t,c}, \rho_{t,c}) = \int [g_o(\mathbf{x}_{t,c}, \mathbf{d}_o, \boldsymbol{\theta}, \rho_{t,c} g_u(\mathbf{x}_{t,c}, \mathbf{d}_u, \boldsymbol{\theta}) + \mathbf{L}(\mathbf{d}_u)\boldsymbol{\alpha}_{t,c})] f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{27}$$

where $\boldsymbol{\mu}_o(\mathbf{x}_{t,c}, \boldsymbol{\alpha}_{t,c}, \rho_{t,c}) = [(\mu_{y_o^e}(\mathbf{x}_{t,c}, \mathbf{d}_{o,1}) \,|\, \boldsymbol{\alpha}_{t,c}, \rho_{t,c}), \cdots, (\mu_{y_o^e}(\mathbf{x}_{t,c}, \mathbf{d}_{o,N_o}) \,|\, \boldsymbol{\alpha}_{t,c}, \rho_{t,c})] \in \mathbb{R}^{1 \times N_o}$. As mentioned above, the above integration can be solved using numerical integration methods. This paper uses MCS-based approach [50], but other quadrature points-based approaches, such as sparse grid [53] and dimension reduction-based method [52] can also be adopted if the dimension of $\boldsymbol{\theta}$ is high.

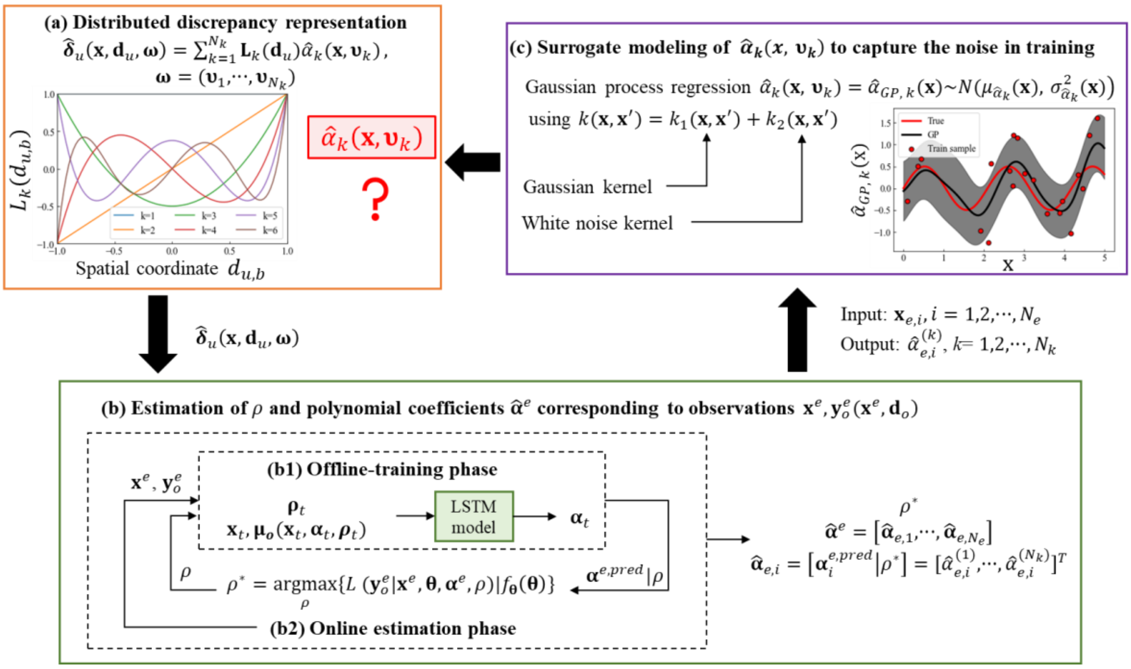

(a) Distributed discrepancy representation
$\hat{\boldsymbol{\delta}}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega}) = \sum_{k=1}^{N_k} \mathbf{L}_k(\mathbf{d}_u)\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k)$,
$\boldsymbol{\omega} = (\boldsymbol{\upsilon}_1, \cdots, \boldsymbol{\upsilon}_{N_k})$
$L_k(d_{u,b})$
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
k=1
k=2
k=3
k=4
k=5
k=6
Spatial coordinate $d_{u,b}$
$\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k)$
?
(c) Surrogate modeling of $\hat{\alpha}_k(x, \boldsymbol{\upsilon}_k)$ to capture the noise in training
Gaussian process regression $\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k) = \hat{\alpha}_{GP,k}(\mathbf{x}) \sim N(\mu_{\hat{\alpha}_k}(\mathbf{x}), \sigma^2_{\hat{\alpha}_k}(\mathbf{x}))$
using $k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$
Gaussian kernel
White noise kernel
True
GP
Train sample
$\hat{\alpha}_{GP,k}(\mathrm{X})$
1.5
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4
5
X
$\hat{\boldsymbol{\delta}}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$
Input: $\mathbf{x}_{e,i}, i = 1,2,\cdots,N_e$
Output: $\hat{\alpha}_{e,i}^{(k)}$, k= $1,2,\cdots,N_k$
(b) Estimation of ρ and polynomial coefficients $\hat{\boldsymbol{\alpha}}^e$ corresponding to observations $\mathbf{x}^e, \mathbf{y}_o^e(\mathbf{x}^e, \mathbf{d}_o)$
(b1) Offline-training phase
$\mathbf{x}^e$, $\mathbf{y}_o^e$
$\boldsymbol{\rho}_t$
$\mathbf{x}_t, \boldsymbol{\mu}_o(\mathbf{x}_t, \boldsymbol{\alpha}_t, \boldsymbol{\rho}_t)$
LSTM model
$\boldsymbol{\alpha}_t$
ρ
$\rho^* = \underset{\rho}{\mathrm{argmax}}\{L\,(\mathbf{y}_o^e|\mathbf{x}^e, \boldsymbol{\theta}, \boldsymbol{\alpha}^e, \rho)|f_{\boldsymbol{\theta}}(\boldsymbol{\theta})\}$
$\boldsymbol{\alpha}^{e,pred}|\rho$
(b2) Online estimation phase
ρ^*
$\hat{\boldsymbol{\alpha}}^e = [\hat{\boldsymbol{\alpha}}_{e,1}, \cdots, \hat{\boldsymbol{\alpha}}_{e,N_e}]$
$\hat{\boldsymbol{\alpha}}_{e,i} = [\boldsymbol{\alpha}_i^{e,pred}|\rho^*] = [\hat{\alpha}_{e,i}^{(1)}, \cdots, \hat{\alpha}_{e,i}^{(N_k)}]^T$

Fig. 5 Module 2: estimation of ρ and surrogate modeling of distributed discrepancy

$$\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$$

With the training data of $\rho_{t,c}$, $\mathbf{x}_{t,c}$, and $\boldsymbol{\alpha}_{t,c}$, and $\boldsymbol{\mu}_o(\mathbf{x}_{t,c}, \boldsymbol{\alpha}_{t,c}, \rho_{t,c})$, $\forall c = 1, \cdots, N_t$, we then build a machine learning model that maps $\mathbf{x}_{t,c}$, $\rho_{t,c}$, and $\boldsymbol{\mu}_o(\mathbf{x}_{t,c}, \boldsymbol{\alpha}_{t,c}, \rho_{t,c})$ to $\boldsymbol{\alpha}_{t,c}$. Since both the inputs and outputs are multi-dimensional, multi-input/multi-output machine learning method is needed. Considering the advantage of long short-term memory (LSTM) model in dealing with sequence to sequence data [54], in this paper, LSTM is employed to construct the model $\boldsymbol{\alpha} = f_{ML}(\rho, \mathbf{x}, \boldsymbol{\mu}_o(\mathbf{x}, \boldsymbol{\alpha}, \rho))$. The developed framework, however, is not limited to LSTM. Other types of multi-input/multi-output machine learning models can also be employed.

LSTM is an artificial recurrent neural network architecture as shown in Fig. 6. It has attracted tremendous interests since it enables the neural network to sequentially learn and determine whether to forget the previous hidden states and update the hidden states or not. In the operation procedure within a LSTM module, the first step is to determine what previous information to forget by a sigmoid layer called the “forget gate” layer [55, 56]. Next, the LSTM module determines what new information $\tilde{C}_t$ to store through a sigmoid layer called “input gate” layer and a tanh layer. The third step updates the new cell state C_t by combining the information inherited from the previous cell state C_{t-1} with the new candidate information $\tilde{C}_t$. Finally, we can obtain the output of this LSTM module. The step-by-step operation is given as below

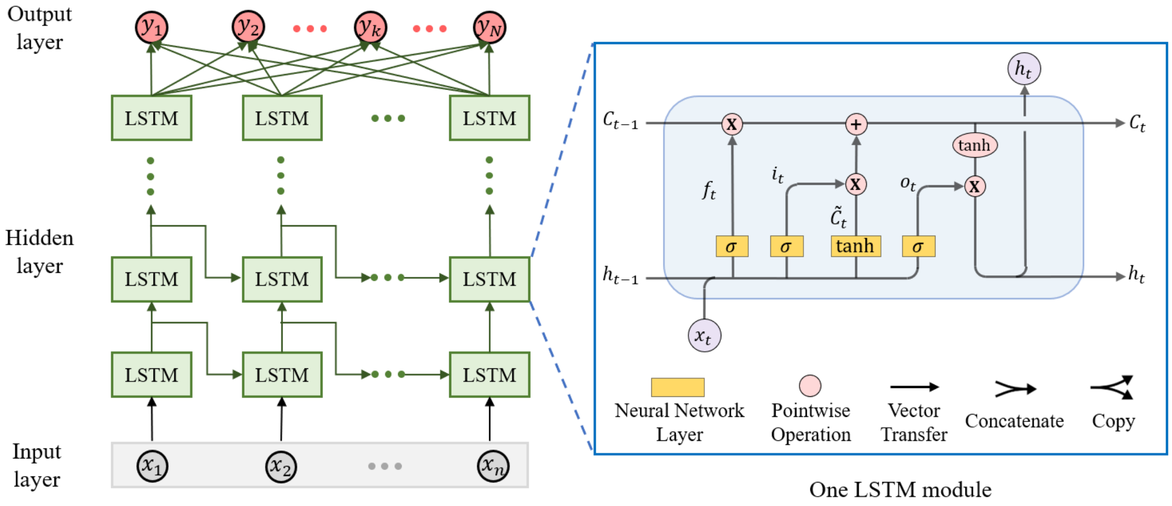


Fig. 6 LSTM neural network architecture [55]

$$
\begin{aligned}
&\text{Step 1: } f_t = \sigma(W_f[h_{t-1}, x_t] + b_f);\\
&\text{Step 2: } i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)\\
&\qquad\quad\ \ \tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C);\\
&\text{Step 3: } C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t;\\
&\text{Step 4: } o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)\\
&\qquad\quad\ \ h_t = o_t \times \tanh(C_t),
\end{aligned}
\tag{28}
$$

where W_f, W_i, W_C and W_o denote the weight matrices of each unit, and b_f, b_i, b_C and b_o are the bias terms associated with each unit. An important step in the training of an LSTM model the determination of the number of hidden layers, neurons, etc. These parameters are determined in this paper through cross-validation by splitting the data into training and testing. Note that even though LSTM is more widely used for time series prediction, it can also be employed to perform multiple-inputs to multiple-outputs mapping which is similar to the sequence-to-sequence mapping.

(b) Online prediction of $\hat{\boldsymbol{\alpha}}^e$ and ρ using $\boldsymbol{\alpha} = f_{ML}(\rho, \mathbf{x}, \boldsymbol{\mu}_o(\mathbf{x}, \boldsymbol{\alpha}, \rho))$

In the online prediction phase, for given ρ and $\mathbf{x}_{e,i}, \mathbf{y}^e_{o,i}$, $\boldsymbol{\alpha} = f_{ML}(\rho, \mathbf{x}, \boldsymbol{\mu}_o(\mathbf{x}, \boldsymbol{\alpha}, \rho))$ is used to predict $\boldsymbol{\alpha}_i^{pred} \,|\, \rho$ based on the fact that $\boldsymbol{\mu}_{\mathbf{y}_o^e}(\mathbf{x}_{e,i}, \mathbf{d}_o) \,|\, \hat{\boldsymbol{\alpha}}_{e,i}, \rho$ can be used to approximate $\mathbf{y}^e_{o,i}$ at the MLE point as explained in Sec. 3.3.2. Using

$\boldsymbol{\alpha} = f_{ML}(\rho, \mathbf{x}, \boldsymbol{\mu}_o(\mathbf{x}, \boldsymbol{\alpha}, \rho))$ and plugging $\mathbf{y}_{o,i}^e$ into the model as $\boldsymbol{\mu}_o(\mathbf{x}, \boldsymbol{\alpha}, \rho)$, we have

$$[\boldsymbol{\alpha}_i^{pred} \mid \rho] \approx f_{ML}(\rho, \mathbf{x}_{e,i}, \mathbf{y}_{o,i}^e), \forall i = 1, \cdots, N_e. \tag{29}$$

Using the machine learning model, the lower-level optimization models given in Eq. (24) can be solved efficiently for any given value of ρ . Using the predicted $\boldsymbol{\alpha}_i^{pred} \mid \rho, \forall i = 1, \cdots, N_e$, the unknown regression coefficient ρ is then estimated as

$$\rho^* = \arg\min_{\rho} \left\{ \sum_{i=1}^{N_e} \left\| \mathbf{y}_{o,i}^e - \mathbf{y}_{o,i}^{pred}(\rho) \right\|_2 \right\}, \tag{30}$$

where $\mathbf{y}_{o,i}^{pred}(\rho)$ is obtained by plugging $\boldsymbol{\alpha}_i^{pred} \mid \rho$ predicted in Eq. (29) into Eq. (27) as follows

$$\mathbf{y}_{o,i}^{pred}(\rho) = \int [g_o(\mathbf{x}_{e,i}, \mathbf{d}_o, \boldsymbol{\theta}, \rho g_u(\mathbf{x}_{e,i}, \mathbf{d}_u, \boldsymbol{\theta}) + \mathbf{L}(\mathbf{d}_u)[\boldsymbol{\alpha}_i^{pred} \mid \rho])] f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \mathbf{d}\boldsymbol{\theta}. \tag{31}$$

The above optimization corresponds to the upper-level optimization defined in Eq. (21). After ρ^* is estimated using Eq. (30), we have $\hat{\boldsymbol{\alpha}}^e = [\hat{\boldsymbol{\alpha}}_{e,1}, \cdots, \hat{\boldsymbol{\alpha}}_{e,N_e}]$ as

$$\hat{\boldsymbol{\alpha}}_{e,i} = [\boldsymbol{\alpha}_i^{pred} \mid \rho^*], \forall i = 1, \cdots, N_e. \tag{32}$$

With $\mathbf{x}_{e,i}$ and the estimate $\hat{\boldsymbol{\alpha}}_{e,i}, i = 1, \cdots, N_e$, we then construct surrogate models for $\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k), k = 1, \cdots, N_k$.

3.3.3 Surrogate modeling of $\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k), k = 1, \cdots, N_k$

In this section, we construct surrogate models $\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k), k = 1, \cdots, N_k$ (i.e. estimate $\boldsymbol{\upsilon}_k$) as indicated in Eq. (16) for the input-dependent coefficients of the model discrepancy polynomial representation of the unobservable model using $\mathbf{x}_{e,i}$ and $\hat{\boldsymbol{\alpha}}_{e,i} = [\hat{\alpha}_{e,i}^{(1)}, \cdots, \hat{\alpha}_{e,i}^{(N_k)}]^T, i = 1, \cdots, N_e$. For the surrogate modeling of $\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k)$, the input training data are $\mathbf{x}_{e,i}, i = 1, \cdots, N_e$ and the output training data are $\hat{\alpha}_{e,i}^{(k)}, i = 1, \cdots, N_e$.

In order to account for the noises in $\hat{\alpha}_{e,i}^{(k)}, i = 1, \cdots, N_e, \forall k = 1, \cdots, N_k$ introduced by the predictions of the LSTM model in Eq. (29), and to quantify the prediction uncertainty at new input settings, Gaussian process (GP) surrogate modeling (i.e. given in the APPENDIX) is employed in this paper to build the surrogate models $\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k), k = 1, \cdots, N_k$. To capture the potential noise in $\hat{\alpha}_{e,i}^{(k)}, i = 1, \cdots, N_e$, we build the GP models with noise-level estimation by adding a white noise to the covariance function (also called kernel function) as follows

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}'), \tag{33}$$

where $\mathbf{x}$ and $\mathbf{x}'$ represent two different input settings, $k_1(\mathbf{x}, \mathbf{x}')$ is the squared exponential kernel function (also known as Gaussian covariance function), which is given by

$$k_1(\mathbf{x}, \mathbf{x}') = \sigma_1^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2}{2l^2}\right), \tag{34}$$

in which σ_1^2 is the overall variance (σ_1 is also known as amplitude), and l denotes length scale. Note that the squared exponential kernel may not be the best choice for some problems. Other type of kernel functions, such as the Matern kernel or user-defined kernels, can also be employed. The kernel function of the GP model can be selected through cross-validation during the training of the GP models.

The white noise kernel $k_2(\mathbf{x}, \mathbf{x}')$ is defined as

$$k_2(\mathbf{x}, \mathbf{x}') = \sigma_2^2 \delta_{\mathbf{x}\mathbf{x}'}, \tag{35}$$

where σ_2^2 is the variance of noise, $\delta_{\mathbf{x}\mathbf{x}'}$ takes one when $\mathbf{x} = \mathbf{x}'$, otherwise, $\delta_{\mathbf{x}\mathbf{x}'}$ is equal to zero. For the given noise model, the variance of GP prediction at the training points is non-zero. More details of GP model can be found in the APPENDIX section.

From the GP surrogate models, for any given specific input $\mathbf{x}$ we have

$$\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k^*) \sim N(\mu_{\alpha_k}(\mathbf{x}), \sigma_{\alpha_k}^2(\mathbf{x})), \tag{36}$$

where $\boldsymbol{\upsilon}_k^*$ are the hyperparameters of the k-th GP model, $\mu_{\alpha_k}(\mathbf{x})$ and $\sigma_{\alpha_k}^2(\mathbf{x})$ are respectively the mean prediction and variance of the prediction at input setting $\mathbf{x}$.

Consequently, $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega}^*)$ is modeled as

$$\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega}^*) = \sum_{k=1}^{N_k} \mathbf{L}_k(\mathbf{d}_u)\hat{\alpha}_k(\mathbf{x}, \boldsymbol{\upsilon}_k^*). \tag{37}$$

Next, we will discuss how to infer the uncertain model parameters $\boldsymbol{\theta}$ by fixing ρ to ρ^* and the discrepancy model $\delta_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$ to the surrogate model $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega}^*)$, where $\boldsymbol{\omega}^* = \{\boldsymbol{\upsilon}_k^*, k = 1, \cdots, N_k\}$.

3.4 Module Three: Bayesian Inference of Uncertain Model Parameter $\boldsymbol{\theta}$

This module updates the uncertain model parameters $\boldsymbol{\theta}$ by fixing ρ^* and $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega}^*)$ (as indicated in Fig. 4) using Bayes' theorem as

$$f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta} \,|\, \mathbf{y}_o^e, \mathbf{x}^e, \boldsymbol{\omega}^*, \rho^*) \propto f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_o^e \,|\, \mathbf{x}^e, \boldsymbol{\theta}, \boldsymbol{\omega}^*, \rho^*)\, f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}), \tag{38}$$

where $f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_o^e \,|\, \mathbf{x}^e, \boldsymbol{\theta}, \boldsymbol{\omega}^*, \rho^*)$ is the likelihood function of observing $\mathbf{y}_o^e$ at input setting $\mathbf{x}^e$ for given $\boldsymbol{\theta}$ after fixing ρ^* and $\boldsymbol{\omega}^*$ (i.e. hyper-parameters of $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega}^*)$).

In order to compute the likelihood function $f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_o^e \,|\, \mathbf{x}^e, \boldsymbol{\theta}, \boldsymbol{\omega}^*, \rho^*)$, we have the predicted response of the observable model after introducing the GP-based bias surrogate model of the unobservable model into Eq. (18) as

$$\hat{\mathbf{y}}_o(\mathbf{x}, \mathbf{d}_o) \,|\, \boldsymbol{\theta}, \boldsymbol{\omega}^*, \rho^* = g_o(\mathbf{x}, \mathbf{d}_o, \boldsymbol{\theta}, \rho^* g_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\theta}) + \sum_{k=1}^{N_k} \hat{\alpha}_{GP,k}(\mathbf{x}, \boldsymbol{\upsilon}_k^*)\mathbf{L}_k(\mathbf{d}_u)) + \boldsymbol{\varepsilon}(\mathbf{d}_o). \tag{39}$$

Based on the above equation, we then have $f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_o^e \,|\, \mathbf{x}^e, \boldsymbol{\theta}, \boldsymbol{\omega}^*, \rho^*)$ given by

$$f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_o^e \,|\, \mathbf{x}^e, \boldsymbol{\theta}, \boldsymbol{\omega}^*, \rho^*) = (2\pi)^{-\frac{N_S}{2}} \left|\mathbf{V}_o(\boldsymbol{\theta})\right|^{-\frac{1}{2}} \times \exp\left\{-\frac{1}{2}\left[\mathbf{y}_o^e - \mathbf{E}_o(\boldsymbol{\theta})\right]^T \mathbf{V}_o(\boldsymbol{\theta})^{-1}\left[\mathbf{y}_o^e - \mathbf{E}_o(\boldsymbol{\theta})\right]\right\}, \tag{40}$$

where $N_S = N_o N_e$ is the total number of observations, $\mathbf{E}_o(\boldsymbol{\theta})$ is the mean prediction for given

$\mathbf{x}^e$ and is given by

$$\mathbf{E}_o(\boldsymbol{\theta}) = g_o(\mathbf{x}^e, \mathbf{d}_o, \boldsymbol{\theta}, \rho^* g_u(\mathbf{x}^e, \mathbf{d}_u, \boldsymbol{\theta}) + \sum_{k=1}^{N_k} \mu_{\alpha_k}(\mathbf{x}^e)\mathbf{L}_k(\mathbf{d}_u)), \tag{41}$$

and $\mathbf{V}_o(\boldsymbol{\theta})$ is the covariance matrix of the prediction, which is given by

$$\mathbf{V}_o(\boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{\Sigma}_{11}(\boldsymbol{\theta}) & \cdots & \boldsymbol{\Sigma}_{1N_e}(\boldsymbol{\theta}) \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{N_e 1}(\boldsymbol{\theta}) & \cdots & \boldsymbol{\Sigma}_{N_e N_e}(\boldsymbol{\theta}) \end{bmatrix} \in \mathbb{R}^{N_S \times N_S}, \tag{42}$$

where $\boldsymbol{\Sigma}_{ij}(\boldsymbol{\theta}) \in \mathbb{R}^{N_o \times N_o}, \forall i = 1, \cdots, N_e;\ j = 1, \cdots, N_e$ is a covariance matrix given by

$$\boldsymbol{\Sigma}_{ij}(\boldsymbol{\theta}) = \begin{bmatrix} c_{ij}^{(1,1)}(\boldsymbol{\theta}) & \cdots & c_{ij}^{(1,N_o)}(\boldsymbol{\theta}) \\ \vdots & \ddots & \vdots \\ c_{ij}^{(N_o,1)}(\boldsymbol{\theta}) & \cdots & c_{ij}^{(N_o,N_o)}(\boldsymbol{\theta}) \end{bmatrix}, \tag{43}$$

in which $c_{ij}^{(p,q)}(\boldsymbol{\theta}) = Cov([\hat{y}_o(\mathbf{x}_e^{(i)}, \mathbf{d}_{o,p}) \mid \boldsymbol{\theta}, \boldsymbol{\omega}^*, \rho^*], [\hat{y}_o(\mathbf{x}_e^{(j)}, \mathbf{d}_{o,q}) \mid \boldsymbol{\theta}, \boldsymbol{\omega}^*, \rho^*])$, $\forall p, q = 1, \cdots, N_o$. $c_{ij}^{(p,q)}(\boldsymbol{\theta})$ represents the covariance between the predictions at different input settings $\mathbf{x}_e^{(i)}$ and $\mathbf{x}_e^{(j)}$, and different spatial locations $\mathbf{d}_{o,p}$ and $\mathbf{d}_{o,q}$ of the response of the observable model.

The covariance $c_{ij}^{(p,q)}(\boldsymbol{\theta})$ can be computed analytically if the observable model given in Eq. (39) is a linear model (e.g. the strain model in the miter gate application problem as shown in Section 4.2). If the model given in Eq. (39) is nonlinear, the observations from the observable model may not follow the multivariate Gaussian form as given in Eq. (40). Even if the multi-variate Gaussian joint PDF might be a reasonable approximation, $c_{ij}^{(p,q)}(\boldsymbol{\theta})$ does not have an analytical form. A possible way of computing the likelihood function $f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_o^e \mid \mathbf{x}^e, \boldsymbol{\theta}, \boldsymbol{\omega}^*, \rho^*)$ in that case is to use sampling-based method in conjunction with the copula-based method [57] by propagating the uncertainty from $\hat{\alpha}_{GP,k}(\mathbf{x}, \boldsymbol{\upsilon}_k^*), \forall k = 1, \cdots, N_k$ to $\hat{\mathbf{y}}_o(\mathbf{x}, \mathbf{d}_o) \mid \boldsymbol{\theta}, \boldsymbol{\omega}^*, \rho^*$. Other alternative approaches could be approximating the likelihood function using sampling-based method such as pseudo-marginal Markov Chain Monte Carlo

(MCMC) or approximating the posterior distributions using likelihood-free inference methods (e.g. approximate Bayesian computation [58]). For problems with nonlinear observable models, the calibration will be even more complicated, and it is a problem that is worth investigating in our future work.

After the posterior distribution of $\boldsymbol{\theta}$ is obtained, $f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta} \,|\, \mathbf{y}_o^e, \mathbf{x}^e, \boldsymbol{\omega}^*, \rho^*)$, ρ^* , and $\hat{\delta}_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega})$ are used to correct the simulation of the unobservable model to improve its prediction validity. The mean posterior prediction of the unobservable model is given by

$$\boldsymbol{\mu}_u^{ct}(\mathbf{x}, \mathbf{d}_u) \,|\, \mathbf{x}^e, \mathbf{y}_o^e = \int \rho^* g_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\theta}) \, f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta} \,|\, \mathbf{y}_o^e, \mathbf{x}^e, \boldsymbol{\omega}^*, \rho^*) \mathbf{d}\boldsymbol{\theta} + \sum_{k=1}^{N_k} \mu_k(\mathbf{x}) \mathbf{L}_k(\mathbf{d}_u). \tag{44}$$

The covariance of the posterior prediction of the unobservable model can be computed similarly to Eqs. (41) through (43). In Module 4, the prediction uncertainty of the unobservable model can then be propagated to the prediction model or observable model to perform prediction under new conditions as shown in Fig. 7.

3.5 Implementation Procedure

Fig. 7 summarizes the implementation procedure of the proposed BA-MUDI framework for Bayesian calibration of multi-level model with unobservable distributed response. Following that, in Sec. 4, we use a miter gate application to demonstrate the proposed framework in detail.

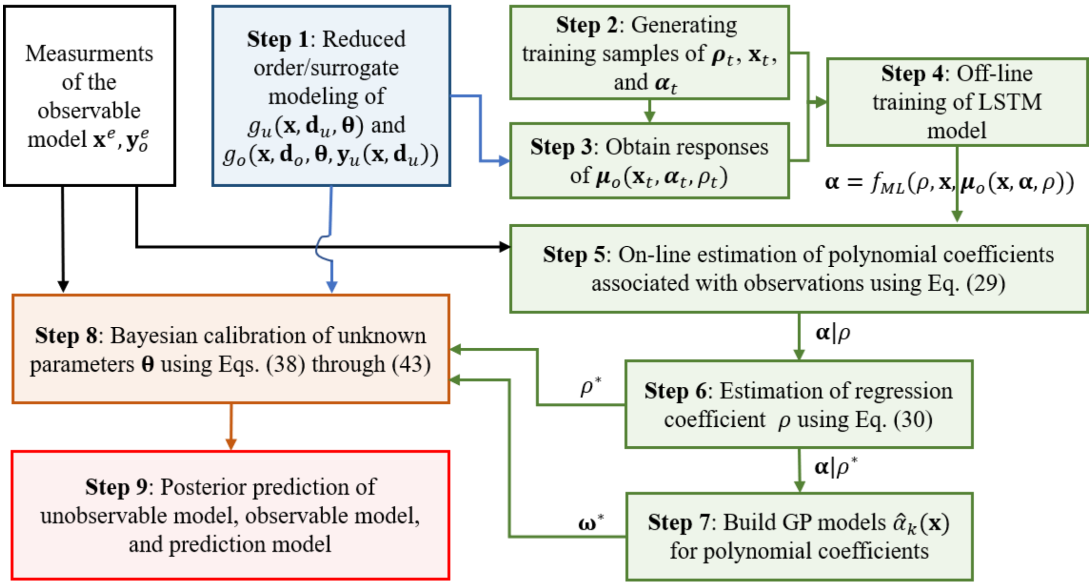
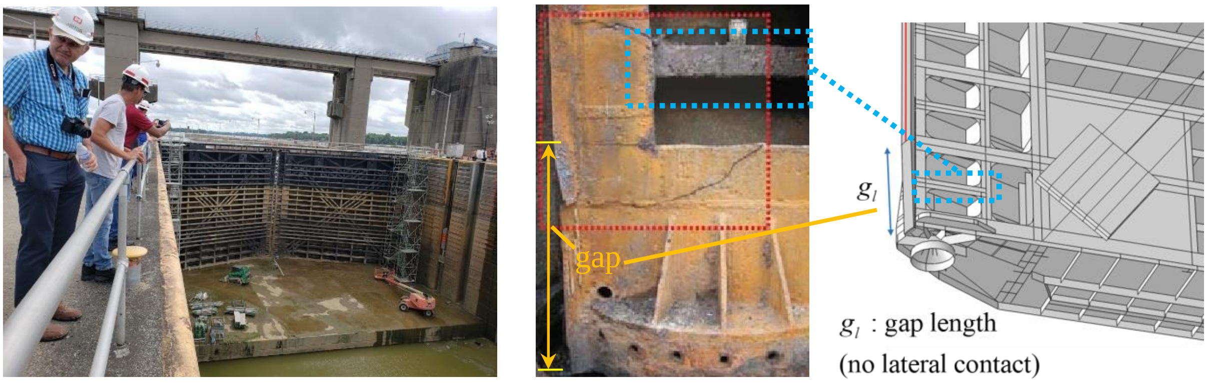

Measurments of the observable model $\mathbf{x}^e, \mathbf{y}_o^e$
Step 1: Reduced order/surrogate modeling of $g_u(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\theta})$ and $g_o(\mathbf{x}, \mathbf{d}_o, \boldsymbol{\theta}, \mathbf{y}_u(\mathbf{x}, \mathbf{d}_u))$
Step 2: Generating training samples of $\boldsymbol{\rho}_t$, $\mathbf{x}_t$, and $\boldsymbol{\alpha}_t$
Step 3: Obtain responses of $\boldsymbol{\mu}_o(\mathbf{x}_t, \boldsymbol{\alpha}_t, \rho_t)$
Step 4: Off-line training of LSTM model
$\boldsymbol{\alpha} = f_{ML}(\rho, \mathbf{x}, \boldsymbol{\mu}_o(\mathbf{x}, \boldsymbol{\alpha}, \rho))$
Step 5: On-line estimation of polynomial coefficients associated with observations using Eq. (29)
$\boldsymbol{\alpha}|\rho$
Step 8: Bayesian calibration of unknown parameters $\boldsymbol{\theta}$ using Eqs. (38) through (43)
ρ^*
Step 6: Estimation of regression coefficient ρ using Eq. (30)
$\boldsymbol{\alpha}|\rho^*$
Step 9: Posterior prediction of unobservable model, observable model, and prediction model
$\boldsymbol{\omega}^*$
Step 7: Build GP models $\hat{\alpha}_k(\mathbf{x})$ for polynomial coefficients

Fig. 7 Implementation procedure (workflow of proposed meta-algorithm) of BA-MUDI

4 APPLICATION OF BA-MUDI FRAMEWORK TO MITER GATE

In the United States, the US Army Corps of Engineers (USACE) maintains and operates 236 miter gates at 191 sites [59], the network of which plays an important role in the inland waterway transportation system. More than half of these structural assets have exceeded their economic design life of 50 years [60]. The repair and maintenance plans are essential for optimal-scheduled closure of these miter gates, since unexpected closures are extremely expensive because they prevent the shippers from fulfilling their scheduled transport missions. One of the most common damage of miter gates stems from the occurrence of a gap (loss of contact between the gate quoin block and the wall quoin block the bottom of the gate), as shown in Fig. 8(b) and 8(c) [29]. Such a gap leads to stress redistribution in the gate structure and results in high-stress zones which may exceed the limit states, leading to failure. Therefore, accurate gap identification and boundary force analysis along the quoin blocks are crucial for repair and maintenance planning. The boundary force prediction model is also important to analyze the fatigue crack initiation at the quoin blocks due to the rolling contact between the wall and the gate. In this section, the proposed BA-MUDI framework is applied to an in-service USACE miter gate [61].



(a) Miter gate dewatered for repair (b) Gap on structure (c) Modeling of gap

Fig. 8 Miter gate and gap at the interface of the wall quoin block and gate quoin block

4.1 Physics-based Multi-Level Simulation Model of Miter Gate

In a previous USACE study, a high-fidelity ABAQUS finite element simulation model as shown in Fig. 9 was developed for the structural analysis of a miter gate in the Greenup lock system (Kentucky, USA) [14]. As shown in Fig. 8(c), the bearing gap (loss of contact) was modelled by not constraining the gate laterally over the length of the bearing gap. To reduce the computational cost of such a large model, the majority of elements in the gate were 3D linear shells elements as shown in Fig. 9. Furthermore, a reduced-order model of the strain analysis model was developed using the static condensation method [61]. In addition, the strain analysis model of the gate was validated in a controlled experimental environment by USACE [61].

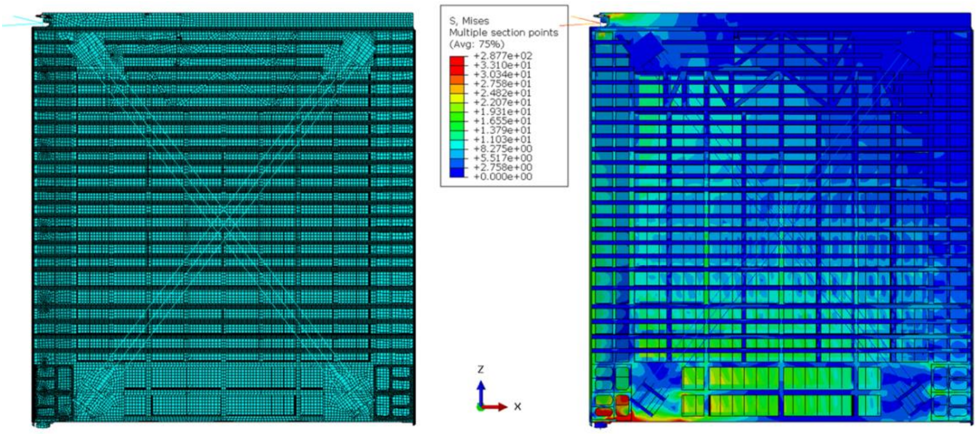


Fig. 9 Finite element model (Left: mesh; Right: stress analysis results)

In order to model the boundary condition of the miter gate, a linear contact analysis model was developed by assuming that the connected bodies are linear and a contact element connects the two linear bodies by satisfying the complementarity condition. The linear complementarity problem, which is equivalent to a quadratic programming optimization problem, is solved using the Lagrange multiplier method [61]. In the linear contact model, the normal force is proportional to the penetration distance (see e.g., Section 2.3 of Ref. [62]). To approximate tangential contact forces arising from static friction, we employ a tangential force that is proportional to the product of the normal force and tangent displacement. The proportionality constants in both the normal and tangential contact models are dependent on the material properties and contact geometry. Here we treat them as tuning parameters in the linear contact model and set their values to 1×10^6, which results in forces that are similar to those seen in more sophisticated contact simulations performed with Abaqus. Note that a nonlinear contact analysis using nonlinear element is more realistic than the linear model for practical applications. The nonlinear model, however, is much more difficult to solve (i.e., requires incremental iterative solution algorithms). Fig. 10 shows the boundary forces in the normal and tangential directions, respectively, obtained from the analysis for a given gap length.

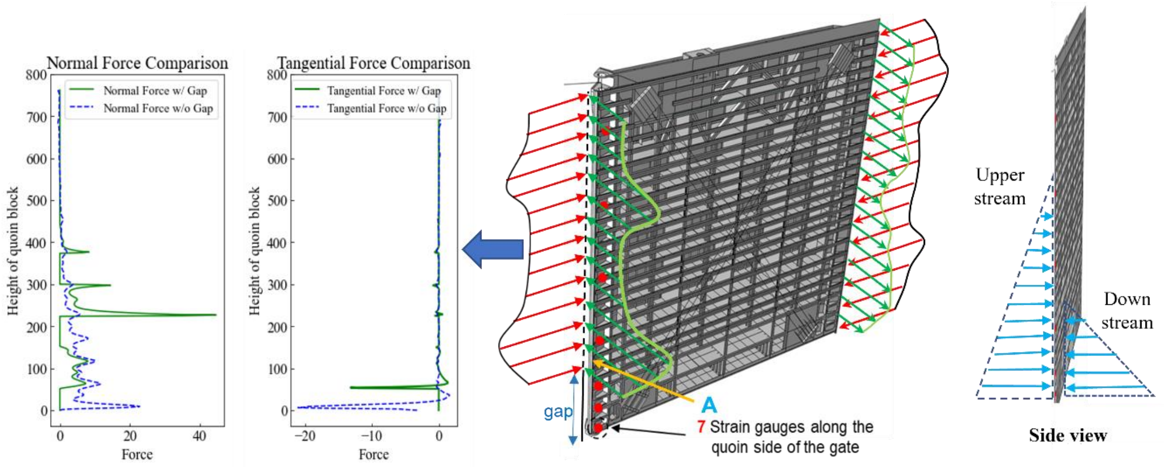
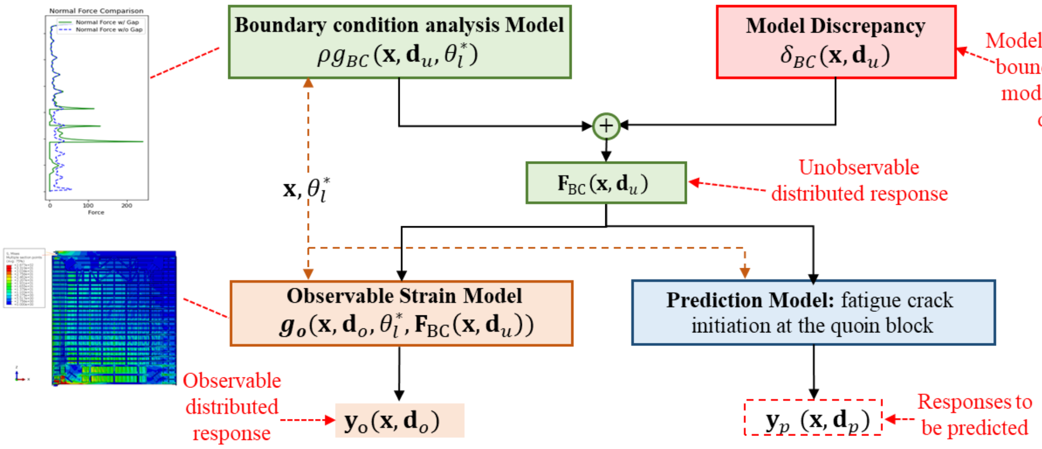


Fig. 10 Contact force analysis of the miter gate (“with” and “without gap”)

The distributed boundary force analysis results will be used as inputs for strain analysis.

Even though these two analysis models share the same finite element source data, the analyses are performed in a multi-level manner. For the miter gate problem, the observable model is the strain analysis model as shown in Fig. 9, and the unobservable model is the contact force analysis model as illustrated in Fig. 10, which should be an extremely complicated nonlinear model if we want to accurately model the contact forces. As mentioned above, the contact force analysis may be used as inputs for the fatigue initialization analysis at the interface between the miter gate and quoin block. The fatigue analysis model or any other model that takes the forces as inputs for the analysis is therefore the prediction model. In this case study, the strain analysis model under new prediction conditions is employed as the prediction model for demonstration purpose. Fig. 11 summarizes the multi-level analysis architecture of the miter gate problem.

As shown in Fig. 11, the controllable inputs $\mathbf{x}$ are the upstream and downstream water levels that govern the net hydrostatic load on the gates, and the unknown model parameter θ_l is the length of the gap which is illustrated in Fig. 8(c). Since the contact analysis model and the strain analysis model share the same FE data, there are no additional inputs $\mathbf{z}_u$ or $\mathbf{z}_o$ that are specific to the unobservable or observable model. The observable response is the distributed strain response $\mathbf{y}_o(\mathbf{x}, \mathbf{d}_o)$, where $\mathbf{d}_o$ are the spatial locations of the strain gauges (i.e. 7 gauges as shown in Fig. 10), and the unobservable distributed response consists of the boundary contact forces $\mathbf{F}_{\text{BC}}(\mathbf{x}, \mathbf{d}_u)$, where $\mathbf{d}_u$ are the spatial coordinates of the quoin block. Due to the simplifying assumptions in the contact force analysis model, such as simplification of load conditions, assumption of linearity of the connection and connected bodies, etc., $\mathbf{F}_{\text{BC}}(\mathbf{x}, \mathbf{d}_u)$ may not accurately represent the true physics. The discrepancy of $\mathbf{F}_{\text{BC}}(\mathbf{x}, \mathbf{d}_u)$ and the unknown gap length θ_l for a specific gate in a practical situation need to be estimated simultaneously using the strain measurements $\mathbf{y}_o(\mathbf{x}, \mathbf{d}_o)$. The proposed BA-MUDI framework will be used in the next section to achieve this.



Normal Force Comparison
Normal Force w/ Gap
Normal Force w/o Gap
Force
0
100
200
Boundary condition analysis Model
$\rho g_{BC}(\mathbf{x}, \mathbf{d}_u, \theta_l^*)$
Model Discrepancy
$\delta_{BC}(\mathbf{x}, \mathbf{d}_u)$
Model discrepancy of boundary condition model needs to be quantified
$\mathbf{x}, \theta_l^*$
$\mathbf{F}_{\mathrm{BC}}(\mathbf{x}, \mathbf{d}_u)$
Unobservable distributed response
Observable Strain Model
$g_o(\mathbf{x}, \mathbf{d}_o, \theta_l^*, \mathbf{F}_{\mathrm{BC}}(\mathbf{x}, \mathbf{d}_u))$
Prediction Model: fatigue crack initiation at the quoin block
Observable distributed response
$\mathbf{y}_o(\mathbf{x}, \mathbf{d}_o)$
$\mathbf{y}_p(\mathbf{x}, \mathbf{d}_p)$
Responses to be predicted

Fig. 11 Multi-level analysis architecture of miter gate problem

4.2 BA-MUDI Framework for Bayesian Calibration of the Unobservable Distributed Boundary Forces

As discussed in Sec. 3 and shown in Fig. 11, the underlying true unobservable boundary contact force can be modeled as

$$\mathbf{F}_{BC}^{true}(\mathbf{x}, \mathbf{d}_u) = \begin{cases} \rho g_{BC}(\mathbf{x}, \mathbf{d}_u, \theta_l^*) + \boldsymbol{\delta}_{BC}(\mathbf{x}, \mathbf{d}_u), \text{ if } \mathbf{d}_u \geq \theta_l^* \\ 0, \text{otherwise} \end{cases}, \tag{45}$$

where $g_{BC}(\mathbf{x}, \mathbf{d}_u, \theta_l^*)$ denotes the contact force analysis model that predicts the unobservable distributed contact force response along the height of the quoin block [61], $\boldsymbol{\delta}_{BC}(\mathbf{x}, \mathbf{d}_u)$ denotes the model discrepancy of the contact force model caused by the model assumptions and simplifications or discretization errors, $\mathbf{d}_u$ are the spatial coordinates of the quoin block, θ_l^* is the true length of the bearing gap which is unknown in real-world conditions since the gap is underwater, and ρ is an unknown regression coefficient.

Since there are both normal and tangential contact forces at each coordinate of the quoin block as shown in Fig. 10, the distributed boundary force is represented as

$\mathbf{F}_{BC}(\mathbf{x},\mathbf{d}_u)=\begin{bmatrix}\mathbf{F}_{BC}(\mathbf{x},\mathbf{d}_{u,1})\\ \vdots \\ \mathbf{F}_{BC}(\mathbf{x},\mathbf{d}_{u,N_u})\end{bmatrix}\in\mathbb{R}^{2N_u\times 1}$, where $\mathbf{F}_{BC}(\mathbf{x},\mathbf{d}_{u,b})\doteq\begin{bmatrix}F_N(\mathbf{x},\mathbf{d}_{u,b})\\ F_T(\mathbf{x},\mathbf{d}_{u,b})\end{bmatrix}$, $\forall b=1,\ldots,N_u$, in which $\mathbf{d}_{u,b}$ is the b-th spatial coordinate at the quoin block and $N_u=509$ is the number of spatial coordinates in the current application adopted for the description of location on the quoin block. Accordingly, we represent the model discrepancy functions of the normal force and tangential force as $\boldsymbol{\delta}_{BC}(\mathbf{x},\mathbf{d}_u)=\begin{bmatrix}\boldsymbol{\delta}_{BC}(\mathbf{x},\mathbf{d}_{u,1})\\ \vdots \\ \boldsymbol{\delta}_{BC}(\mathbf{x},\mathbf{d}_{u,N_u})\end{bmatrix}\in\mathbb{R}^{2N_u\times 1}$, where $\boldsymbol{\delta}_{BC}(\mathbf{x},\mathbf{d}_{u,b})\doteq\begin{bmatrix}\delta_N(\mathbf{x},\mathbf{d}_{u,b})\\ \delta_T(\mathbf{x},\mathbf{d}_{u,b})\end{bmatrix}$, $\forall b=1,\ldots,N_u$. We therefore have the true unobservable boundary contact forces as

$$\mathbf{F}_{BC}^{true}(\mathbf{x},\mathbf{d}_u)=\begin{bmatrix}\mathbf{F}_{BC}^{true}(\mathbf{x},\mathbf{d}_{u,1})\\ \vdots \\ \mathbf{F}_{BC}^{true}(\mathbf{x},\mathbf{d}_{u,N_u})\end{bmatrix}$$
$$=\begin{cases}\rho g_{BC}(\mathbf{x},\mathbf{d}_u,\theta_l^*)+\begin{bmatrix}\boldsymbol{\delta}_{BC}(\mathbf{x},\mathbf{d}_{u,1})\\ \vdots \\ \boldsymbol{\delta}_{BC}(\mathbf{x},\mathbf{d}_{u,N_u})\end{bmatrix}\in\mathbb{R}^{2N_u\times 1}, \text{if } \mathbf{d}_{u,b}, \forall b=1,\cdots,N_u\geq\theta_l^* \\ 0, \text{otherwise}\end{cases}. \tag{46}$$

For the observable strain response model as shown in Fig. 11, after static condensation [29], the strain response with corrected boundary force inputs is modeled using a linear model as

$$\mathbf{y}_o^{true}(\mathbf{x},\mathbf{d}_o)=g_o(\mathbf{x},\mathbf{d}_o,\theta_l^*,\mathbf{F}_{BC}^{true}(\mathbf{x},\mathbf{d}_u))$$
$$=\begin{cases}\mathbf{A}(\rho g_{BC}(\mathbf{x},\mathbf{d}_u,\theta_l^*)+\boldsymbol{\delta}_{BC}(\mathbf{x},\mathbf{d}_u)+\mathbf{F}_{water}(\mathbf{x},\mathbf{d}_u)), \text{if } \mathbf{d}_u\geq\theta_l^* \\ 0, \text{otherwise}\end{cases}, \tag{47}$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_{N_o} \end{bmatrix} \in \mathbb{R}^{N_o \times 2N_u}$ is a matrix obtained from static condensation (i.e. Eq (3.14) in Ref. [61], which is a function of a stiffness matrix and a differentiation matrix), $\mathbf{A}_p = \left[\mathbf{A}_{p,1}, \cdots, \mathbf{A}_{p,N_u}\right], \forall p = 1, \cdots, N_o$, $\mathbf{A}_{p,b} = [A_{N,p,b}, A_{T,p,b}], \forall b = 1, \cdots, N_u$, $A_{N,p,b}$ and $A_{T,p,b}$ are elements associated with the normal and tangential forces, respectively, $\mathbf{F}_{water}(\mathbf{x}, \mathbf{d}_u) \in \mathbb{R}^{2N_u \times 1}$ are the hydrostatic loads due to the different water levels on both sides of the lock chamber obtained from the finite element analysis, $\mathbf{y}_o^{true}(\mathbf{x}, \mathbf{d}_o) = [y_o^{true}(\mathbf{x}, \mathbf{d}_{o,1}), \ldots, y_o^{true}(\mathbf{x}, \mathbf{d}_{o,N_o})]^T \in \mathbb{R}^{N_o \times 1}$ are the true strain responses at N_o strain gauges, $\mathbf{d}_{o,i}, i = 1, \cdots, N_o$ are the spatial coordinates of the strain gauges, and $N_o = 7$ strain gauges are used in this study, which are shown as red dots in Fig. 10.

Based on Eq. (12), the measured strain $\mathbf{y}_o^e$ for a given $\mathbf{x}$ is predicted by

$$\mathbf{y}_o^e(\mathbf{x}, \mathbf{d}_o) = \mathbf{y}_o^{true}(\mathbf{x}, \mathbf{d}_o) + \boldsymbol{\varepsilon}(\mathbf{d}_o), \tag{48}$$

where $\boldsymbol{\varepsilon}(\mathbf{d}_o) = [\varepsilon(\mathbf{d}_{o,1}), \ldots, \varepsilon(\mathbf{d}_{o,N_o})]^T$ are the measurement errors of $\mathbf{y}_o^e$.

To calibrate the unobservable boundary condition model and estimate θ_l at the same time, following the BA-MUDI framework proposed in Sec. 3, we construct surrogate models for $\delta_N(\mathbf{x}, \mathbf{d}_u)$ and $\delta_T(\mathbf{x}, \mathbf{d}_u)$ using the approach presented in Sec. 3.3. After that, we obtain the regression coefficient ρ^* and surrogate models of $\boldsymbol{\delta}_{BC}(\mathbf{x}, \mathbf{d}_u)$ as

$$\hat{\boldsymbol{\delta}}_{BC}(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega}^*) = \begin{bmatrix} \sum_{k_1=1}^{N_{k1}} \hat{\alpha}_{N,k_1}(\mathbf{x}, \boldsymbol{\upsilon}_{N,k_1}) \mathbf{L}_{k_1}(\mathbf{d}_u) \\ \sum_{k_2=1}^{N_{k2}} \hat{\alpha}_{T,k_2}(\mathbf{x}, \boldsymbol{\upsilon}_{T,k_2}) \mathbf{L}_{k_2}(\mathbf{d}_u) \end{bmatrix}, \tag{49}$$

in which $\mathbf{L}_{k_1}(\mathbf{d}_u) = [L_{k_1}(\mathbf{d}_{u,1}), \ldots, L_{k_1}(\mathbf{d}_{u,N_u})]^T \in \mathbb{R}^{N_u \times 1}$ is the k_1-th polynomial basis,

$\mathbf{L}_{k_2}(\mathbf{d}_u) \in \mathbb{R}^{N_u \times 1}$ is the k_2-th polynomial basis, $\boldsymbol{\omega}^* = \{\boldsymbol{\upsilon}_{N,k_1}, k_1 = 1, \cdots, N_{k1}; \boldsymbol{\upsilon}_{T,k_2}, k_2 = 1, \cdots, N_{k2}\}$, where N_{k1} and N_{k2} are number of expansion terms used for the normal contact force and tangential contact force, respectively. In this case study, $N_{k1} = N_{k2} = 6$ which minimizes the errors between the prediction and observations quantified by Eq. (30).

After the surrogate modeling of $\hat{\boldsymbol{\delta}}_{BC}(\mathbf{x}, \mathbf{d}_u, \boldsymbol{\omega}^*)$ and the estimation of ρ^*, we perform Bayesian estimation of the gap length θ_l following the method presented in Sec. 3.4. For the likelihood function given in Eq. (40), the mean prediction of strain at the input setting $\mathbf{x}^e$ is given by

$$\mathbf{E}_o(\theta_l) = \mathbf{A}(\rho^* g_{BC}(\mathbf{x}^e, \mathbf{d}_u, \theta_l) + \boldsymbol{\mu}_{BC}(\mathbf{x}^e, \mathbf{d}_u, \boldsymbol{\omega}^*) + \mathbf{F}_{water}(\mathbf{x}^e, \mathbf{d}_u), \tag{50}$$

where $\boldsymbol{\mu}_{BC}(\mathbf{x}^e, \mathbf{d}_u, \boldsymbol{\omega}^*) = \begin{bmatrix} \sum_{k_1=1}^{N_{k1}} \mu_{N,k_1}(\mathbf{x}^e, \boldsymbol{\upsilon}_{N,k_1}) \mathbf{L}_{k_1}(\mathbf{d}_u) \\ \sum_{k_2=1}^{N_{k2}} \mu_{T,k_2}(\mathbf{x}^e, \boldsymbol{\upsilon}_{T,k_2}) \mathbf{L}_{k_2}(\mathbf{d}_u) \end{bmatrix} \in \mathbb{R}^{2N_u \times 1}$.

The covariance matrix is calculated using Eq. (42) and Eq. (43) with $\mathbf{V}_o(\theta_l) \in \mathbb{R}^{N_S \times N_S}$ ($N_s = N_e N_o$). Since the $g_o(\mathbf{x}, \mathbf{d}_o, \theta_l^*, \mathbf{F}_{BC}^{true}(\mathbf{x}, \mathbf{d}_u))$ given in Eq. (47) is a linear model of the unobservable model after static condensation, we can compute $c_{ij}^{(p,q)}(\theta_l) \doteq Cov([\hat{y}_o(\mathbf{x}_e^{(i)}, \mathbf{d}_{o,p}) \,|\, \theta_l, \boldsymbol{\omega}^*, \rho^*], [\hat{y}_o(\mathbf{x}_e^{(j)}, \mathbf{d}_{o,q}) \,|\, \theta_l, \boldsymbol{\omega}^*, \rho^*])$, $\forall i, j = 1, \cdots, N_e$; $p, q = 1, \cdots, N_o$ in Eq. (43) analytically as

$$\begin{aligned} c_{ij}^{(p,q)}(\theta_l) = &\sum_{k_1=1}^{N_{k1}} \mathbf{A}_{N,p} \mathbf{L}_{k_1}(\mathbf{d}_u) \mathbf{A}_{N,q} \mathbf{L}_{k_1}(\mathbf{d}_u) \mathrm{Cov}(\hat{\alpha}_{N,k_1}(\mathbf{x}_e^{(i)}), \hat{\alpha}_{N,k_1}(\mathbf{x}_e^{(j)})) \\ &+ \sum_{k_2=1}^{N_{k2}} \mathbf{A}_{T,p} \mathbf{L}_{k_2}(\mathbf{d}_u) \mathbf{A}_{T,q} \mathbf{L}_{k_2}(\mathbf{d}_u) \mathrm{Cov}(\hat{\alpha}_{T,k_2}(\mathbf{x}_e^{(i)}), \hat{\alpha}_{T,k_2}(\mathbf{x}_e^{(j)})) + \chi \sigma_\varepsilon^2, \end{aligned} \tag{51}$$

where $\mathbf{A}_{N,p} \in \mathbb{R}^{1 \times N_u}$ and $\mathbf{A}_{T,p} \in \mathbb{R}^{1 \times N_u}$ are respectively the elements of $\mathbf{A}_p \in \mathbb{R}^{1 \times 2N_u}$ as explained below Eq. (47), $\mathbf{L}_{k_1}(\mathbf{d}_u)$ and $\mathbf{L}_{k2}(\mathbf{d}_u) \in \mathbb{R}^{N_u \times 1}$ is the k_1-th or k_2-th order polynomial basis used to approximate the normal contact force or tangential contact force respectively, $\text{Cov}(\hat{\alpha}_{N,k_1}(\mathbf{x}_e^{(i)}),\hat{\alpha}_{N,k_1}(\mathbf{x}_e^{(j)}))$ is the covariance between GP surrogate models $\hat{\alpha}_{N,k_1}(\mathbf{x}_e^{(i)},\boldsymbol{\upsilon}_{N,k_1})$ and $\hat{\alpha}_{N,k_1}(\mathbf{x}_e^{(j)},\boldsymbol{\upsilon}_{N,k})$ (i.e., GP surrogate models of the coefficients of the polynomial bias), $\text{Cov}(\hat{\alpha}_{T,k_2}(\mathbf{x}_e^{(i)}),\hat{\alpha}_{T,k_2}(\mathbf{x}_e^{(j)}))$ represents the covariance between GP surrogate models $\hat{\alpha}_{T,k_2}(\mathbf{x}_e^{(i)},\boldsymbol{\upsilon}_{T,k_2})$ and $\hat{\alpha}_{T,k_2}(\mathbf{x}_e^{(i)},\boldsymbol{\upsilon}_{T,k_2})$, and $\chi=1$ when $i=j$ and $p=q$, otherwise, $\chi=0$. If $i=j$, then $\text{Cov}(\hat{\alpha}_{N,k_1}(\mathbf{x}_e^{(i)}),\hat{\alpha}_{N,k_1}(\mathbf{x}_e^{(j)}))=\sigma_{\alpha_{k_1}}^2(\mathbf{x}_e^{(i)})$, else $\text{Cov}(\hat{\alpha}_{N,k_1}(\mathbf{x}_e^{(i)}),\hat{\alpha}_{N,k_1}(\mathbf{x}_e^{(j)}))$ needs to be computed according to the kernel function used in the GP surrogate modeling as discussed in the APPENDIX. $\text{Cov}(\hat{\alpha}_{T,k_2}(\mathbf{x}_e^{(i)}),\hat{\alpha}_{T,k_2}(\mathbf{x}_e^{(j)}))$ is computed similarly as $\text{Cov}(\hat{\alpha}_{N,k_1}(\mathbf{x}_e^{(i)}),\hat{\alpha}_{N,k_1}(\mathbf{x}_e^{(j)}))$. It is observed from Eq. (51) that the covariance mainly stems from the GP prediction uncertainty and measurement error, since $g_{BC}^m(\mathbf{x},\mathbf{d}_u,\theta_l)$ is a deterministic model. Using the above mean and covariance functions in conjunction with Eq. (40), the posterior distribution of the gap length θ_l, $f_{\theta|\mathbf{y}}(\theta_l\,|\,\mathbf{y}_o^e,\mathbf{x}^e,\boldsymbol{\omega}^*,\rho^*)$, is obtained.

The boundary force prediction is then updated as

$$\begin{aligned}\hat{\mathbf{F}}_{BC}(\mathbf{x},\mathbf{d}_u)\,|\,\mathbf{y}_o^e,\mathbf{x}^e,\boldsymbol{\omega}^*,\rho^* = &\int_{\theta_l}\rho^* g_{BC}(\mathbf{x},\mathbf{d}_u,\theta_l)\, f_{\theta|\mathbf{y}}(\theta_l\,|\,\mathbf{y}_o^e,\mathbf{x}^e,\boldsymbol{\omega}^*,\rho^*)d\theta_l \\ &+\hat{\boldsymbol{\delta}}_{BC}(\mathbf{x},\mathbf{d}_u,\boldsymbol{\omega}^*)\in\mathbb{R}^{2N_u\times 1},\end{aligned} \tag{52}$$

where $\hat{\boldsymbol{\delta}}_{BC}(\mathbf{x},\mathbf{d}_u,\boldsymbol{\omega}^*)$ is the surrogate models given in Eq. (49) and obtained from Module Two of the BA-MUDI framework.

The posterior prediction of the strain response can be updated similarly. The updated unobservable boundary force prediction model can then be used for the other analysis (e.g. fatigue initiation analysis) to improve the prediction accuracy. Next, we present results obtained from the BA-MUDI framework for the miter gate.

4.3 Results

For the purpose of demonstrating and verifying the effectiveness of the proposed BA-MUDI framework, we assume the underlying "true" discrepancy functions of the unobservable distributed normal and tangential forces. These "true" discrepancy functions are then used in conjunction with the original linear contact analysis model to generate synthetic numerical experimental data. After that, the proposed BA-MUDI algorithm is blind to the underlying "true" discrepancy functions. It tries to recover the "true" discrepancy functions using orthogonal polynomials as indicated in Eq. (15), and to estimate the unknown gap length based on the synthetically generated numerical experiments. The "true" discrepancy functions that are used to generate numerical experiments are assumed to be

$$\boldsymbol{\delta}_N(\mathbf{x}, \mathbf{d}_u) = \begin{cases} 3(h_{up} - h_{down})[(762 - \mathbf{d}_u)/1200]^3, \text{ if } \mathbf{d}_u \geq \theta_l^* \\ 0, \text{ otherwise} \end{cases}, \tag{53}$$

and

$$\boldsymbol{\delta}_T(\mathbf{x}, \mathbf{d}_u) = \begin{cases} [(h_{up} - h_{down})/100]^2 \sin(\mathbf{d}_u / 240), \text{ if } \mathbf{d}_u \geq \theta_l^* \\ 0, \text{ otherwise} \end{cases}, \tag{54}$$

where $\mathbf{x} = [h_{up}, h_{down}]$ are respectively the upstream and downstream water levels, which vary within the range of [24, 744] inches. The height of the quoin block is 762 inches, which means each component of the vector $\mathbf{d}_u$ falls within the range of $[0, 762]$ inches. The Legendre orthogonal polynomials are used to approximate the above assumed "true" model discrepancy functions. Note that the proposed framework is not limited to the above discrepancy form; it is applicable to any generalized discrepancy functions as long as the orthogonal polynomial basis can be used to capture the spatially-distributed response.

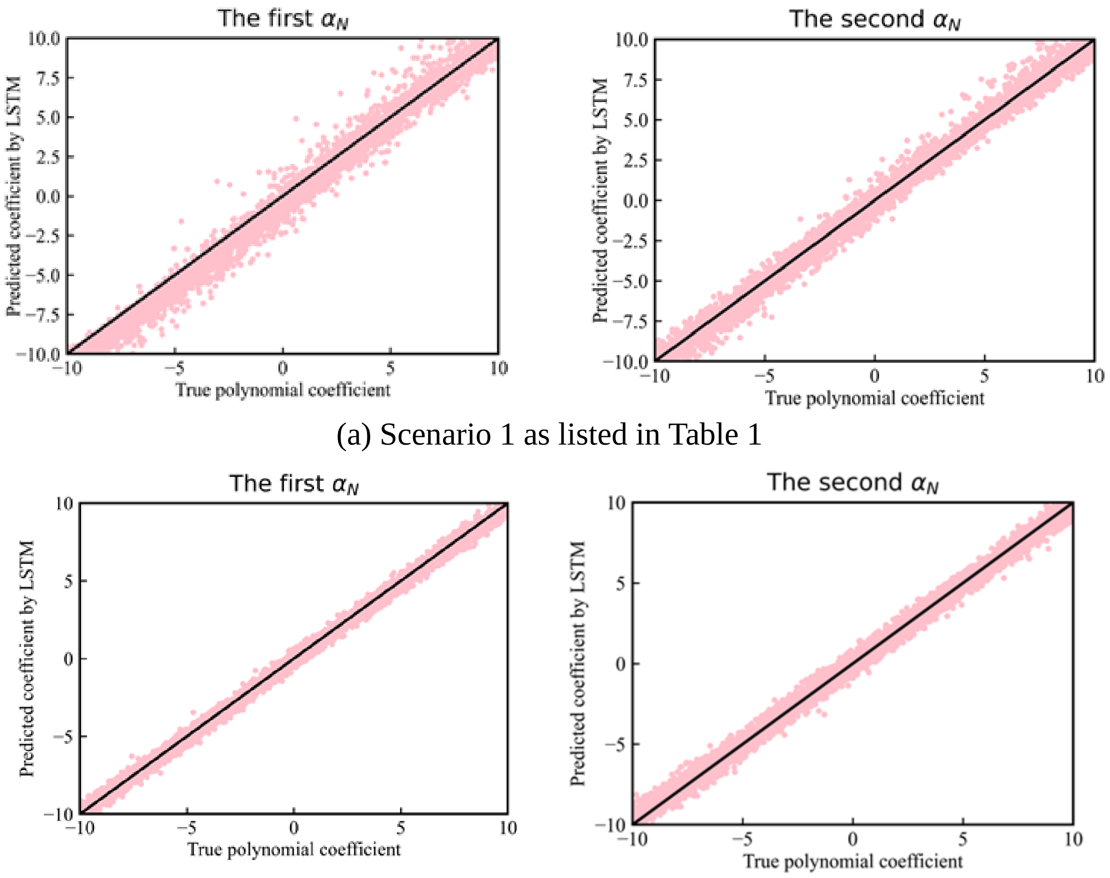
The miter gate under two different scenarios (i.e. health status) as shown in Table 1 are employed to validate the proposed framework. In *Scenario 1*, the true gap length θ_l^* is assumed to be 90 inches. A non-informative uniform distribution is assumed to be the prior distribution, and we have $\theta_l \sim \mathrm{Unif}(81,105)$ inches. In *Scenario 2*, the true gap length θ_l^* is assumed to be 150 inches, and we have $\theta_l \sim \mathrm{Unif}(135,165)$ inches as the prior distribution.

Table 1 Two scenarios of miter gate with different gap lengths

Scenario	True gap length θ_l^* (inch)	Prior distribution of θ_l
Scenario 1	90	$\mathrm{Unif}(81,105)$
Scenario 2	150	$\mathrm{Unif}(135,165)$

Based on the contact analysis model developed based on linear assumption as well as the two assumed true discrepancy functions of the unobservable model, we have the underlying "true" contact analysis model as illustrated in Fig. 11. We then use the "true" model to generate 500 sets of synthetic strain measurements (i.e. $\mathbf{x}^e$ and $\mathbf{y}_o^e$, where $N_e = 500$) by plugging Eqs. (53), (54), and θ_l^* into Eq. (48), and ρ=0.8 (i.e. "true" value of ρ) is used for both scenarios. The standard deviation of strain measurement error is assumed to be $\sigma_\varepsilon = 1$. For each set of synthetic data, there are $N_o = 7$ different strain responses collected through the seven sensors as illustrated in Fig. 10. After that, the BA-MUDI framework is assumed to be blind to ρ, θ_l^*, $\boldsymbol{\delta}_N(\mathbf{x}, \mathbf{d}_u)$, and $\boldsymbol{\delta}_T(\mathbf{x}, \mathbf{d}_u)$ for both scenarios. The BA-MUDI framework will be employed to discover the model discrepancy functions of the unobservable contact force analysis model and estimate the unknown gap length θ_l based on measurements $\mathbf{x}^e$ and $\mathbf{y}_o^e$ of the observable model.

Following the BA-MUDI framework as summarized in Fig. 7, an LSTM model is first constructed based on 50, 000 training points of ρ, $\mathbf{x}$, and $\boldsymbol{\alpha}$, which includes both $\hat{\boldsymbol{\alpha}}_N$ and $\hat{\boldsymbol{\alpha}}_T$ as given in Eq. (49). For the trained LSTM model, there are ten inputs which include ρ, $\mathbf{x}$, and the strain responses of the seven strain gauges after marginalizing the unknown gap length θ_l according to the assumed prior distribution, and 12 outputs since $N_{k1} = N_{k2} = 6$ order Legendre polynomials are used in Eq. (49). For this miter gate example, the off-line training of the LSTM model takes about 172 mins based on a personal computer with Intel i7-6700 CPU @ 3.40 GHz and 16.0 GB RAM memory. The whole process of BA-MUDI totally takes 247 mins, which are mainly spent by Module 2 (190 mins) and Module 3 (56 mins). For illustration purpose, Fig. 12(a) and Fig. 12(b) show the prediction accuracy comparison of the LSTM model for 5000 validation points for two out of the 12 output variables for *Scenario 1* and *Scenario 2*, respectively. The prediction accuracy of the other output variables is at a similar level.

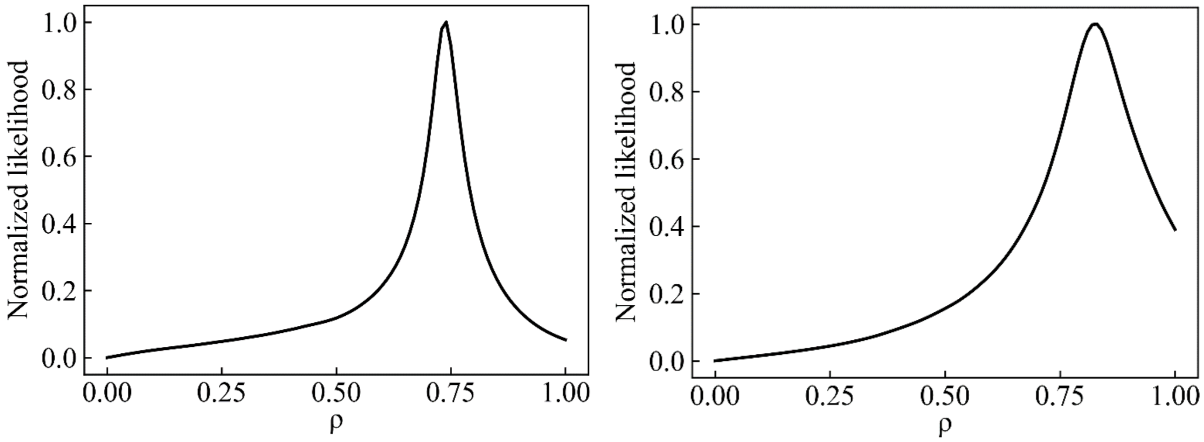


(a) Scenario 1 as listed in Table 1

(b) Scenario 2 as listed in Table 1

Fig. 12 Prediction accuracy verification of the LSTM model

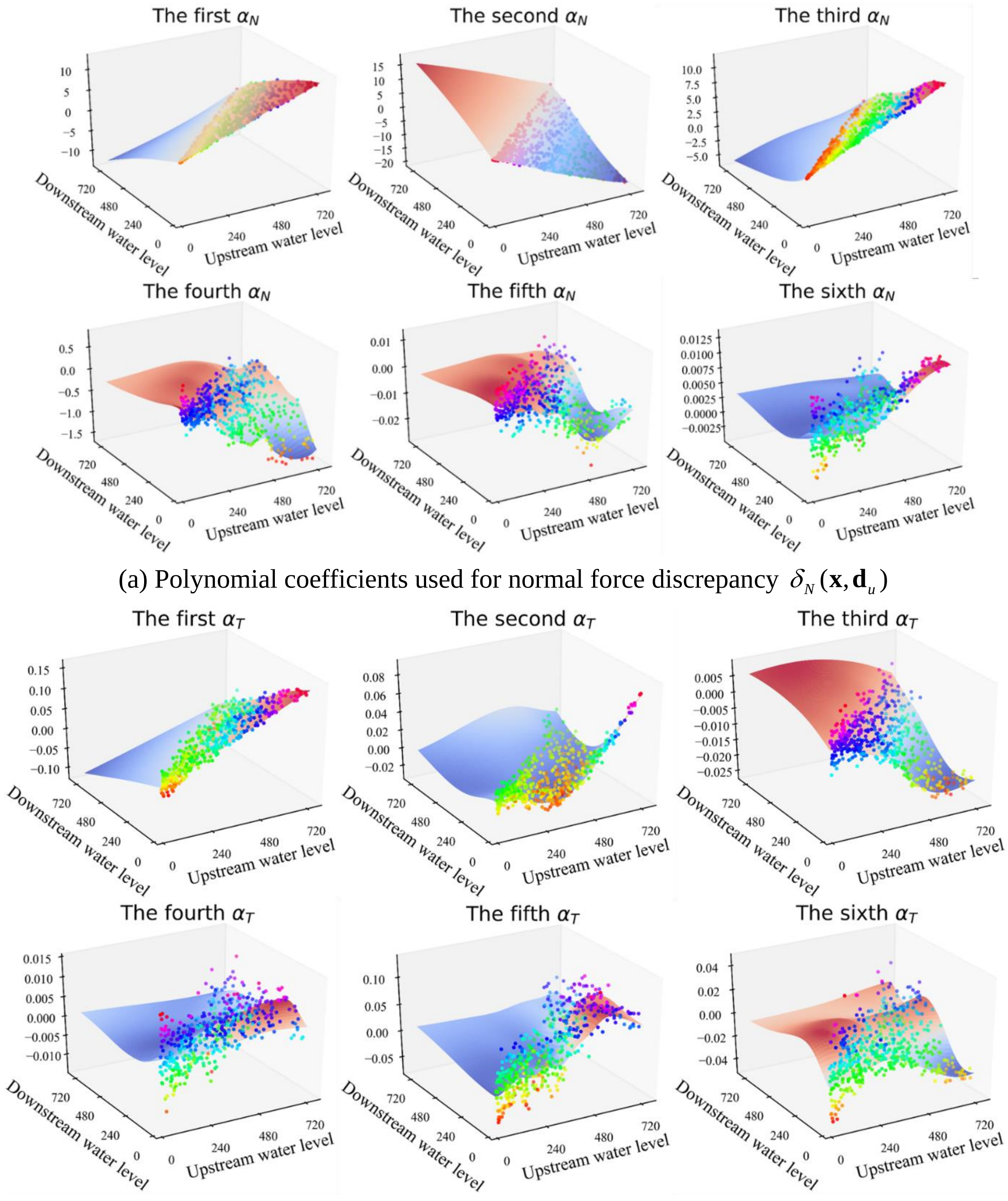
After that, the LSTM model is employed to estimate ρ^*, and $\hat{\boldsymbol{\alpha}}_{e,i}, i = 1, \cdots, N_e$. Fig. 13 shows the normalized likelihood of regression coefficient ρ over [0, 1] (i.e. Eq. (30)) for both scenarios. For *Scenario 1,* the estimated regression coefficient ρ^* is 0.74. For *Scenario 2,* ρ^* is estimated as 0.83. The estimated regression coefficients of both scenarios are approaching the assumed “true” value (i.e. ρ=0.8) that is used to generate the synthetic measurement data. This demonstrates the effectiveness of the proposed method in estimating ρ^*. Note that ρ^* is not estimated precisely, due to the uncertainty in the unknown model parameter (i.e. gap length) and the model discrepancy. The error compensation between the three factors: ρ, θ_l^*, $\boldsymbol{\delta}_N(\mathbf{x},\mathbf{d}_u)$, and $\boldsymbol{\delta}_T(\mathbf{x},\mathbf{d}_u)$ will lead to an improve prediction accuracy after Bayesian calibration through the BA-MUDI framework.



(a) Scenario 1 (b) Scenario 2

Fig. 13 Likelihood of different values of ρ while estimating ρ^*

With the estimated ρ^*, $\hat{\boldsymbol{\alpha}}_{e,i}, i=1,\cdots,N_e$ are predicted using LSTM as explained in the Module Two of the BA-MUDI framework. Twelve GP surrogate models are then constructed for $\hat{\alpha}_{N,k}(\mathbf{x},\boldsymbol{\upsilon}_{N,k}), k=1,\cdots,N_{k1}$, and for $\hat{\alpha}_{T,q}(\mathbf{x},\boldsymbol{\upsilon}_{T,q}), q=1,\cdots,N_{k2}$. Fig. 14 shows the constructed twelve GP surrogate models and the associated training points predicted in Module two for *Scenario 1*. As shown in this figure, the training data is noisy due to the prediction bias of the LSTM model. The modified kernel function of the GP model as discussed in Sec. 3.3.3 allows us to capture this part of uncertainty in the GP surrogate modeling. The GP models for *Scenario 2* are constructed using the associated training points in the same manner.

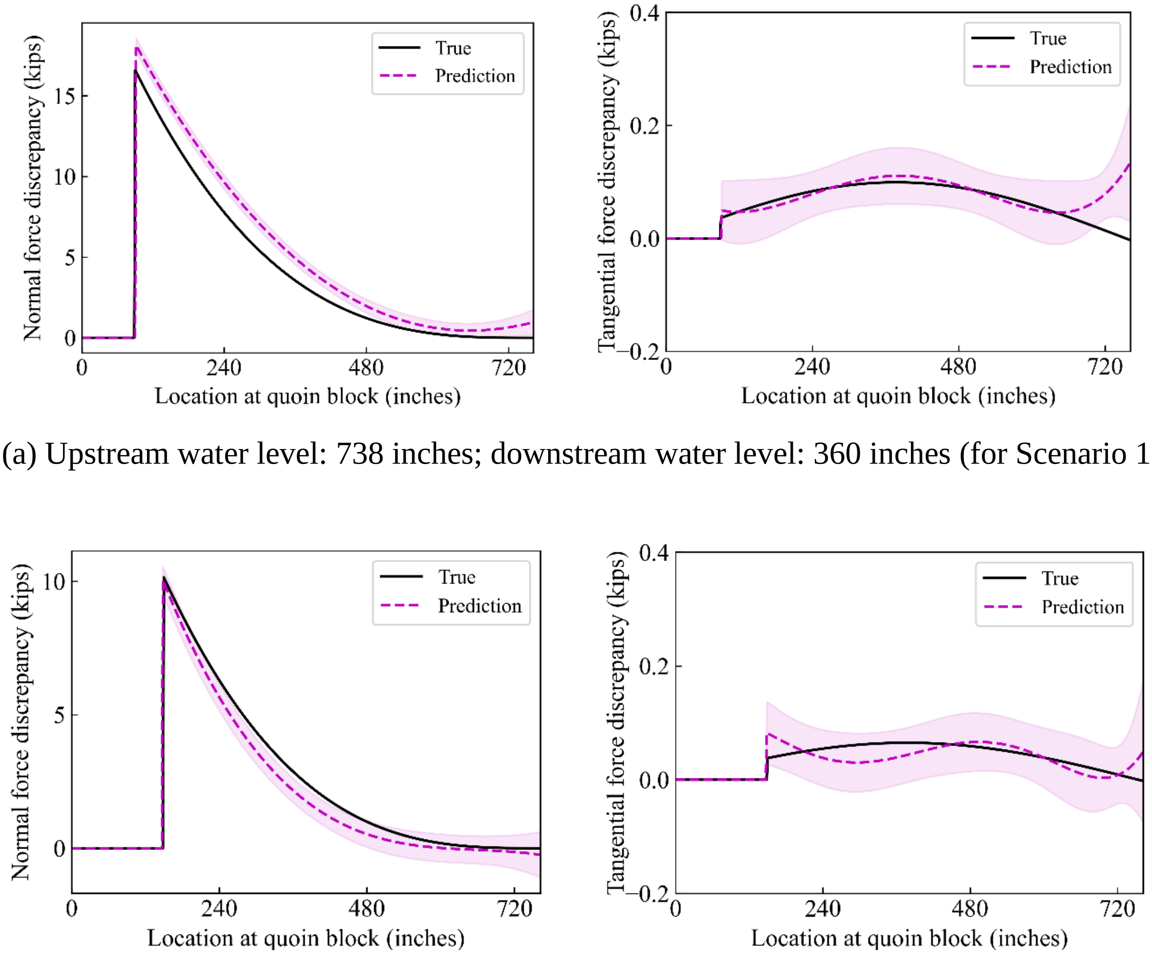


(a) Polynomial coefficients used for normal force discrepancy $\delta_N(\mathbf{x}, \mathbf{d}_u)$

(b) Polynomial coefficients used for tangential force discrepancy $\delta_T(\mathbf{x}, \mathbf{d}_u)$

Fig. 14 Gaussian process (GP) models for Legendre polynomial coefficients in Scenario 1

After the GP surrogate modeling of the coefficients with the polynomial basis, we are able to reconstruct the model discrepancy of the unobservable distributed boundary condition forces. Fig. 15 shows the comparison of the predicted model discrepancy functions at two different water levels for the two scenarios. The results show that the proposed framework is able to reasonably accurately discover the discrepancy function of the normal force with relatively small prediction uncertainty. Although the mean prediction of the distributed tangential force slightly deviates from the true force, the reconstructed discrepancy function is still accurate enough because the true tangential force lies within the 95% credible intervals. This is attributed to the fact that the tangential forces are much lower than the distributed normal forces (i.e. signal-to-noise issue, the tangential force is at the noise level).

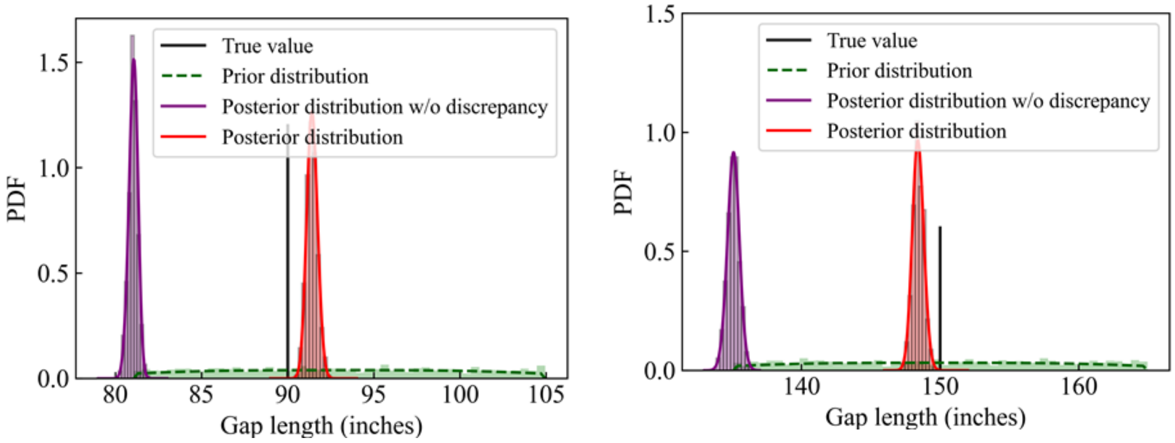


(a) Upstream water level: 738 inches; downstream water level: 360 inches (for Scenario 1)

(b) Upstream water level: 426 inches; downstream water level: 120 inches (for Scenario 2)

Fig. 15 Results of model discrepancy reconstruction of unobservable boundary conditions

In Module Three (i.e. Sec. 3.4) of the BA-MUDI framework, the posterior distribution of the gap length is obtained using the estimated regression coefficient and the reconstructed discrepancy functions. Fig. 16 presents a comparison of the prior and posterior distributions of θ_l for both scenarios. It shows that the uncertainty of the gap length is well reduced and the posterior distribution concentrates on a narrow domain closed to the true value for both scenarios. In addition, we also performed Bayesian inference of the gap length without considering the model discrepancy modeling as detailed in Module 2 of BA-MUDI framework. As shown in Fig. 16, the posterior distribution of the gap length without considering model discrepancy deviates much more from the true value than its counterpart with the consideration of model discrepancy. Model discrepancy of the unobservable model results in bias in the estimated model parameters. This indicates that the proposed framework is able to mitigate the impact of model discrepancy on model parameter estimation and the benefit of including the model discrepancy in Bayesian calibration. The results in Fig. 16 also demonstrate the effectiveness of the proposed BA-MUDI framework in simultaneously estimating the uncertain model parameter and the hidden model discrepancy of the unobservable distributed response.

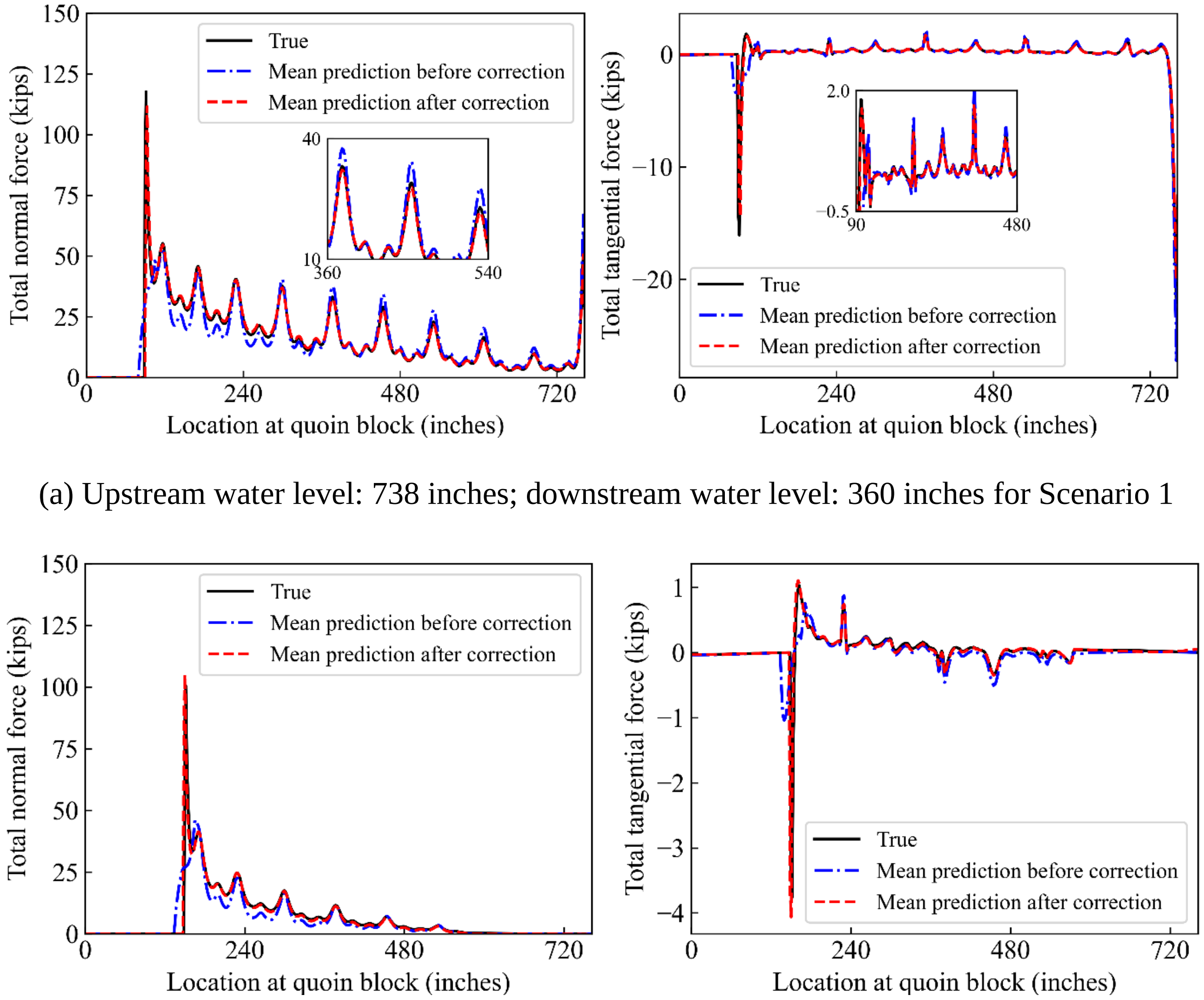


(a) Scenario 1

(b) Scenario 2

Fig. 16 Posterior distribution of miter gate gap length

Module Four of BA-MUDI corrects the predictions of the unobservable boundary condition model and the observable strain model. Fig. 17 plots the mean of the posterior prediction of the normal and tangential forces based on the estimated regression coefficient (see Fig. 14), reconstructed model discrepancy (see Fig. 15), and the updated posterior distribution of the gap length (see Fig. 16). The results of the two scenarios at different water levels, show that the BA-MUDI framework significantly improves the prediction accuracy of the boundary condition model after Bayesian calibration.

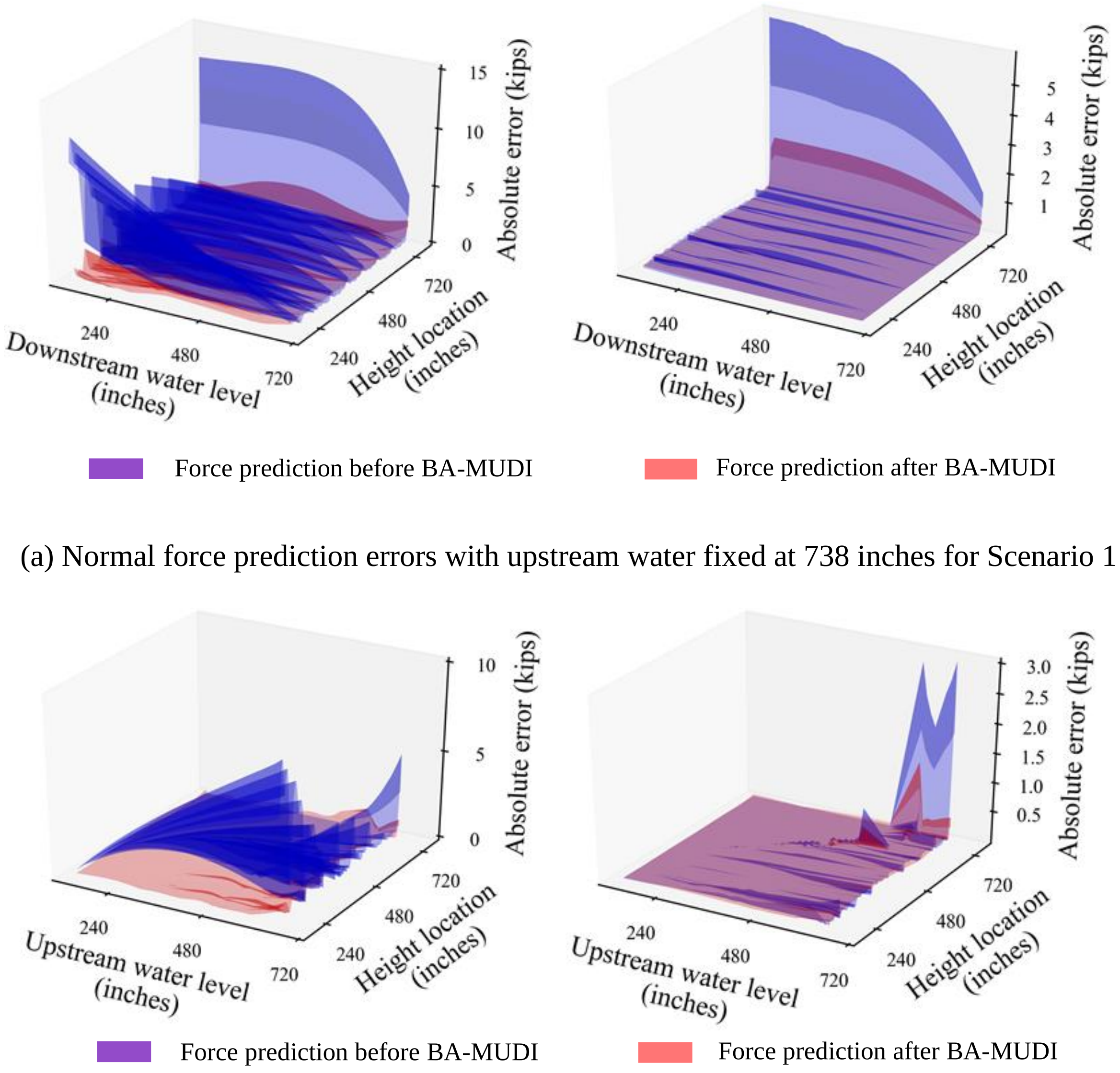


(a) Upstream water level: 738 inches; downstream water level: 360 inches for Scenario 1

(b) Upstream water level: 426 inches; downstream water level: 120 inches for Scenario 2

Fig. 17 Total force prediction results at a certain water level for the two scenarios

Moreover, Fig. 18 illustrates the prediction errors over the input domain (water level) and spatial domain (location at the quoin block) by separately fixing the upstream and downstream water levels. Fig. 18(a) shows the error comparison by fixing the upstream level at 738 inches which corresponds to Fig. 17(a) of *Scenario 1*. Similarly, Fig. 18(b) provides the prediction error comparison corresponding to Fig. 17(b) of *Scenario 2* by fixing the downstream water level at 120 inches. The comparison of the prediction errors with and without BA-MUDI indicates the superiority of BA-MUDI on improving the predictive accuracy of the boundary condition model.

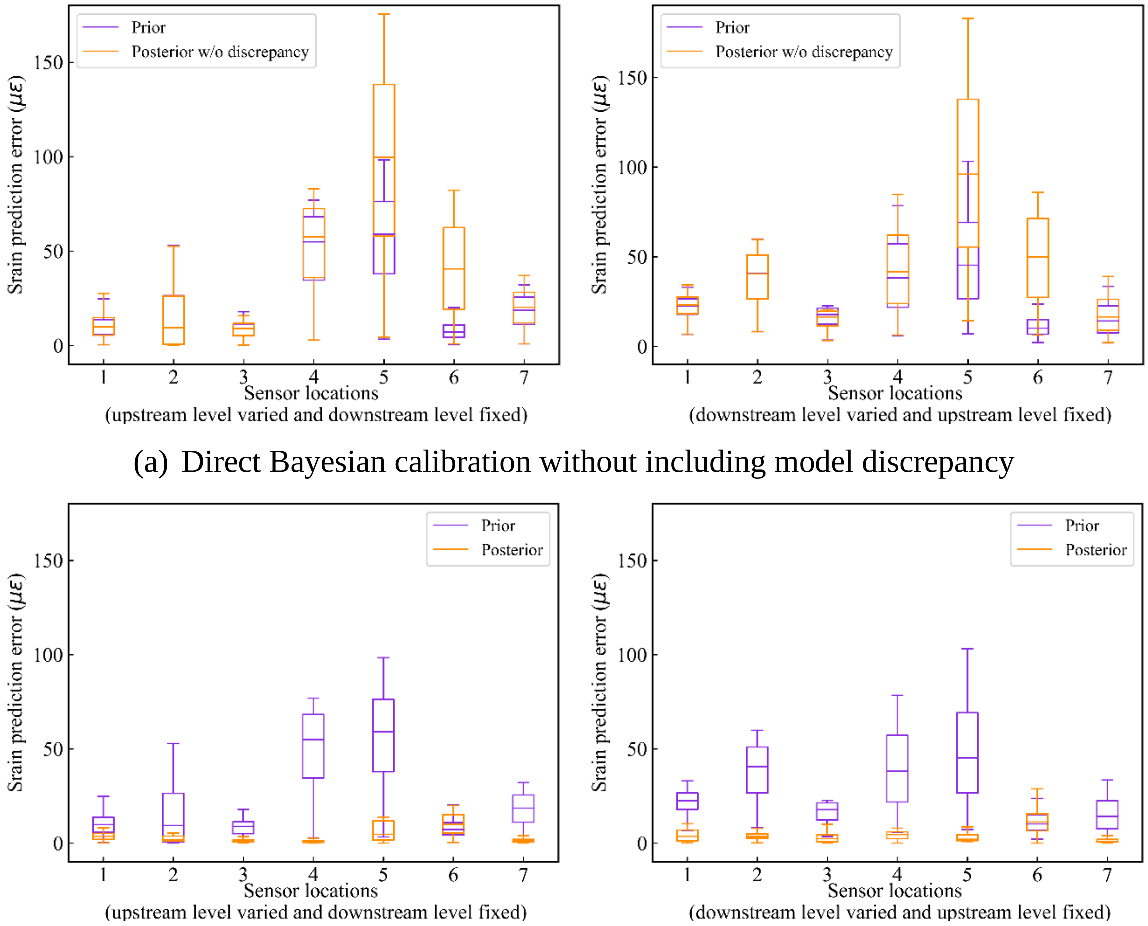


(a) Normal force prediction errors with upstream water fixed at 738 inches for Scenario 1

(b) Normal force prediction errors with downstream water fixed at 120 inches for Scenario 2

Fig. 18 Normal force prediction errors before and after BA-MUDI

Fig. 19 and Fig. 20 depict the comparison of strain response prediction errors at the seven sensor locations for 50 different random input realizations and posterior samples of the gap length for the two scenarios with and without considering model discrepancy of the unobservable boundary condition models. The results show that including the model discrepancy term for the unobservable boundarsy condition model with distributed response in Bayesian calibration can drastically increase the prediction accuracy of the observable model at new and previously unseen conditions. It demonstrates the significance of including model discrepancy in our BA-MUDI framework.

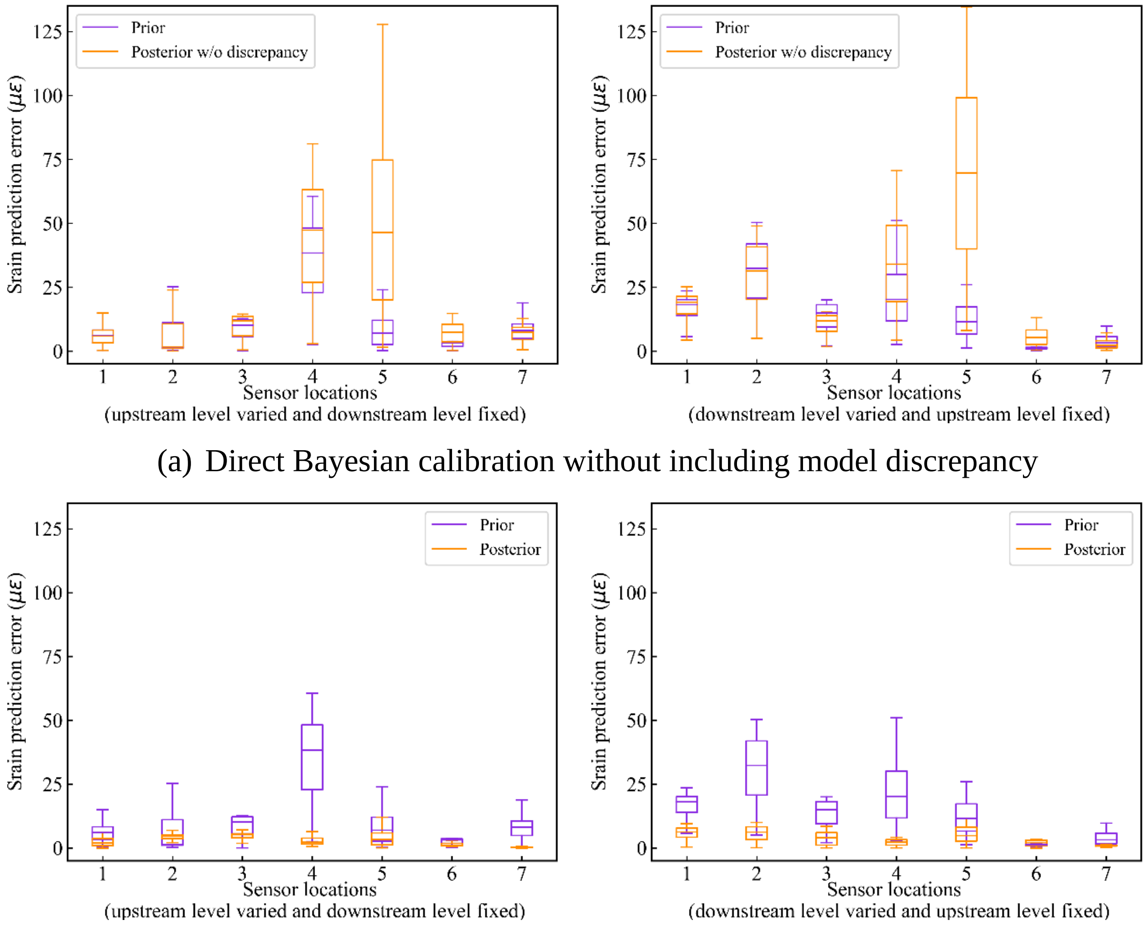


(a) Direct Bayesian calibration without including model discrepancy

(b) With model discrepancy quantification of unobservable model using BA-MUDI

Fig. 19 Strain prediction errors with and without considering model discrepancy reconstruction for Scenario 1 in Table 1

In addition, we investigated the prediction for the up- and down-stream water levels varying over a certain time period as shown in Fig. 21. Fig. 22 presents the corresponding normal and tangential boundary force at the location A as indicated in Fig. 10 of the quoin block over that time period. It shows that the prediction accuracy of the posterior mean is improved significantly for both scenarios after implementing the BA-MUDI framework, especially for the normal force prediction. Note that the tangential force is small and its impact on the observable model (i.e. strain analysis model) is very low. As a result, for certain locations, it will be very difficult to update the tangential force model based on strain measurements, even though the prediction accuracy of the mean tangential forces has been significantly improved as shown in the right-hand-side of Figs. 22(a) and (b) for location “A” as indicated in Fig. 10. In addition, the BA-MUDI framework is able to quantify the prediction uncertainty caused by the unobservable model discrepancy and the sposterior uncertainty of the gap lengths.



(a) Direct Bayesian calibration without including model discrepancy

(b) With model discrepancy quantification of unobservable model using BA-MUDI

Fig. 20 Strain prediction errors with and without considering model discrepancy reconstruction for Scenario 2 in Table 1

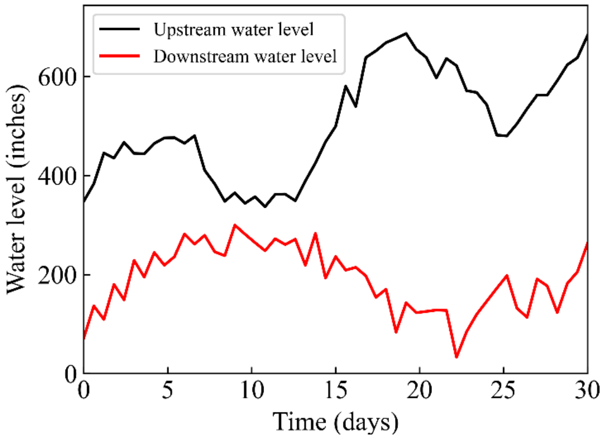
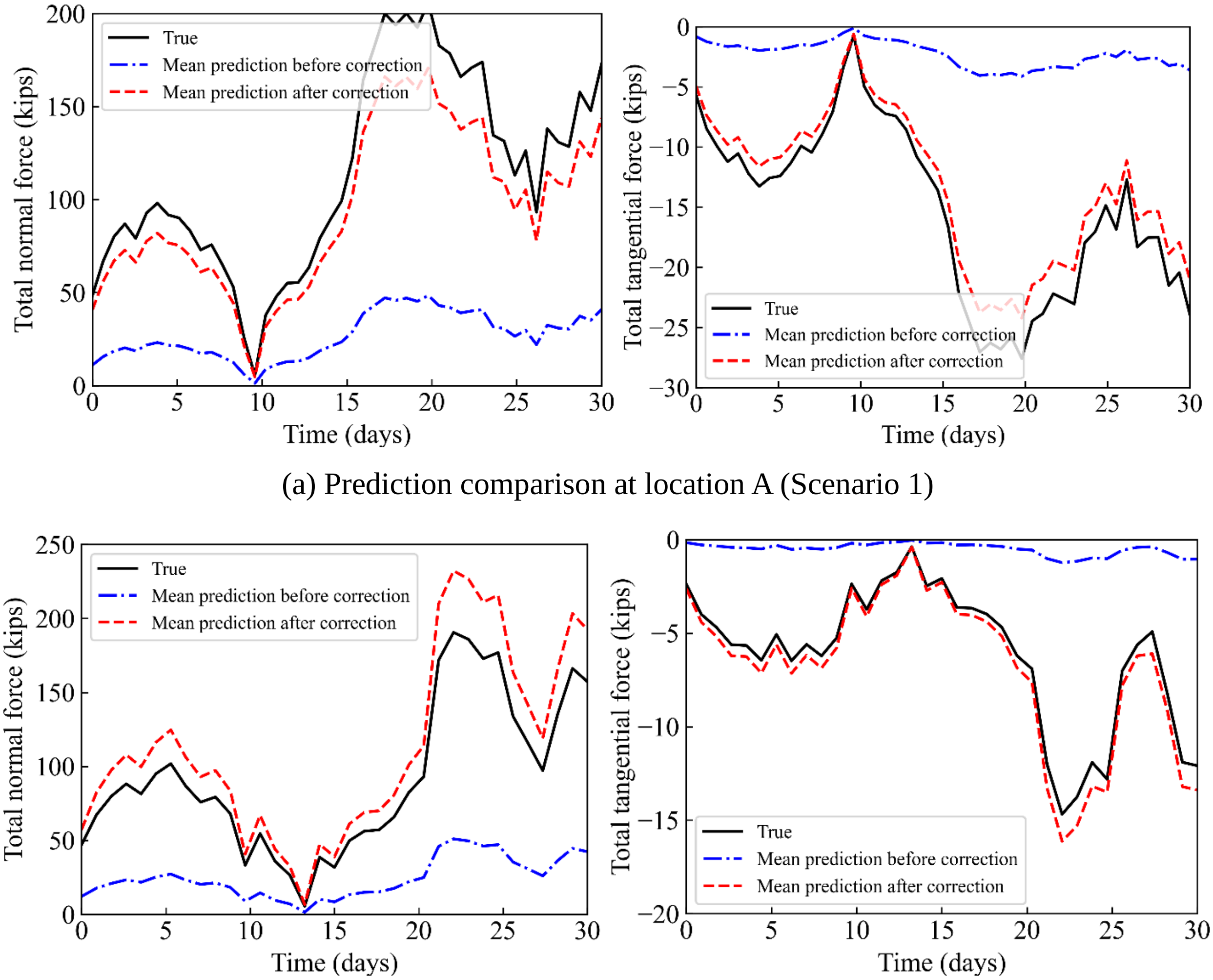


Fig. 21 Upstream and downstream water levels over time period



(a) Prediction comparison at location A (Scenario 1)

(b) Prediction comparison at location A (Scenario 2)

Fig. 22 Total force predictions at location A (see Fig. 10) of quoin block over time period

The above results demonstrate the efficacy of the proposed BA-MUDI framework for Bayesian calibration of multi-level models with unobservable distributed response.

5 CONCLUSIONS

In this work, we first formulate a multi-level model architecture with an unobservable distributed response, where the model predictive power cannot be improved using the conventional Bayesian calibration methods for single-level models. A modularized Bayesian calibration method is extended to multi-level models, named BA-MUDI, in order to tackle the challenge (not resolved in the current literature to the authors knowledge) of calibrating the unobservable model with distributed response using the measurement data of the observable model. BA-MUDI reduces the uncertainty (i.e. improves the predictive power and thus the validity) of an unobservable model from the following two aspects: (1) the unobservable model is corrected by machine learning-based predictive methods for model discrepancy, and (2) the unknown model parameters are updated through a Bayesian scheme.

The BA-MUDI framework was applied to the correction of a boundary condition model for a miter gate that is important in analyzing the fatigue crack initiation at the quoin block due to the rolling contact between the quoin block and the gate. Synthetic measurement data were generated numerically based on assumed true model discrepancy and model parameters. Then, the true model discrepancy/model parameters were kept unknown to the BA-MUDI method/framework. BA-MUDI inversely inferred the length of the gap between the quoin block and the miter gate and estimated the assumed model discrepancy terms. The successful identification of the gap length provides a good understanding of the deterioration of miter gates, which can provide useful information for health diagnostics, maintenance planning, and overall life cycle management. BA-MUDI also corrects the boundary condition model by constructing surrogate models for the hidden model discrepancy. Both the predictive accuracy for the boundary force along the quoin block and strains over the miter gate was improved significantly using BA-MUDI.

In summary, the BA-MUDI framework provides a feasible solution for correcting multi-level modeling and simulation with unobservable and distributed responses using experiments/observations. It can be easily extended to the multi-level model calibration with multiple observable models, and the unobservable response is not limited to distributed responses. It is worth noting that BA-MUDI does not consider the prediction uncertainty of the surrogate model in Module one if computationally "cheap" surrogate models are utilized to replace the original computationally expensive physics/mechanics-based simulation models. The surrogate uncertainty can be incorporated into BA-MUDI by employing

Gaussian process regression modeling similar to the KOH framework [20]. Even though the model discrepancy functions for the unobservable model are well approximated and the posterior distributions of the unknown parameters are well inferred in the miter gate application, the potential non-identifiability issue of KOH [24] may be inherited by BA-MUDI since BA-MUDI follows the modularized framework of KOH. This important issue will be investigated in the future and potentially resolved using our recently proposed sequential calibration and validation framework [22]. Moreover, there are many complex real-world engineering problems that share similarity with the studied miter gate problem, and can be formulated as a multi-level model with unobservable sub-models. The solution provided in this paper and demonstrated with the miter gate problem is directly applicable to this type of problems.

ACKNOWLEDGEMENT

Funding for this work was provided by the US Army Corps of Engineers through the U.S. Army Engineer Research and Development Center Research Cooperative Agreement W912HZ-17-2-0024. The support is gratefully acknowledged.

APPENDIX: Gaussian process (GP) model

Gaussian process (GP) regression is a widely used machine learning method for surrogate modeling of computationally expensive simulation models [63]. GP regression assumes that the deterministic response surface is a realization of the spatial-dependent random process $y(\mathbf{x})$ with prior mean $\mathbf{h}^T(\mathbf{x})\boldsymbol{\beta}$, variance σ^2 and correlation function $R(\bullet,\bullet)$. The random process can be formulated by separating the mean and covariance as

$$y(\mathbf{x}) = \mathbf{h}^T(\mathbf{x})\boldsymbol{\beta} + Z(\mathbf{x}), \quad \text{(A1)}$$

where $\mathbf{h}^T(\mathbf{x})$ is a regression vector with some specific regression functions (e.g. constant, linear, etc), $\boldsymbol{\beta}$ is a vector of regression coefficients to be estimated while modelling, $Z(\mathbf{x})$ is a stochastic process with zero mean and covariance $\sigma^2 R(\bullet,\bullet)$. The most commonly used correlation function between the responses at point $\mathbf{x}$ and $\mathbf{x}'$ is given by

$$R(\mathbf{x},\mathbf{x}') = \exp\left\{-\sum_{i=1}^{d}\gamma_i (x_i - x_i')^2\right\}, \tag{A2}$$

where d is the number of dimensions in the input $\mathbf{x}$, $\boldsymbol{\gamma} = \left(\gamma_1, \gamma_2, ..., \gamma_d\right)^T$ is a vector of the correlation length parameters. Based on the above definition, a GP model can be completely characterized by hyper-parameters $\boldsymbol{\upsilon} \triangleq \left(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\gamma}\right)$ and the predefined regression functions.

Given a set of training data $\left\{(\mathbf{x}_j, \mathbf{y}_j), j = 1, 2, \ldots, N\right\}$, the hyper-parameters $\boldsymbol{\upsilon} \triangleq \left(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\gamma}\right)$ can be estimated using the full Bayesian approach [64] or the maximum likelihood method [65] . After the estimation $\hat{\boldsymbol{\upsilon}} \triangleq \left(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2, \hat{\boldsymbol{\gamma}}\right)$ of $\boldsymbol{\upsilon}$, the posterior mean and covariance of the GP model are given by [65]

$$E\left[y(\mathbf{x}) \mid \mathbf{y}\right] = \mathbf{h}^T(\mathbf{x})\hat{\boldsymbol{\beta}} + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\hat{\boldsymbol{\beta}}), \tag{A3}$$

and

$$\begin{aligned}&Cov\left[y(\mathbf{x}), y(\mathbf{x}') \mid \mathbf{y}\right] = \\ &\hat{\sigma}^2 \Big\{R(\mathbf{x},\mathbf{x}') - \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}') + \\ &\left[\mathbf{h}^T(\mathbf{x}) - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{r}(\mathbf{x})\right]\left[\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right]^{-1}\left[\mathbf{h}^T(\mathbf{x}') - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}')\right]\Big\},\end{aligned} \tag{A4}$$

where $\mathbf{H} = \left[\mathbf{h}^T(\mathbf{x}_1), \mathbf{h}^T(\mathbf{x}_2), \ldots \mathbf{h}^T(\mathbf{x}_N)\right]^T$, $\mathbf{r}(\mathbf{x})$ is a $N \times 1$ vector with i-th element given by $R(\mathbf{x}, \mathbf{x}_i),\ i = 1, 2, \ldots, N$ and $\hat{\boldsymbol{\beta}} = \left[\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right]^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y}$.

REFERENCES

[1] A. Lye, A. Cicirello, E. Patelli, Sampling methods for solving Bayesian model updating

problems: A tutorial, Mechanical Systems and Signal Processing, 159 (2021) 107760.
[2] G. Lee, W. Kim, H. Oh, B.D. Youn, N.H. Kim, Review of statistical model calibration and validation—from the perspective of uncertainty structures, Structural and Multidisciplinary Optimization, 60 (2019) 1619-1644.
[3] Y. Xiong, W. Chen, K.-L. Tsui, D.W. Apley, A better understanding of model updating strategies in validating engineering models, Computer methods in applied mechanics and engineering, 198 (2009) 1327-1337.
[4] B.D. Youn, B.C. Jung, Z. Xi, S.B. Kim, W. Lee, A hierarchical framework for statistical model calibration in engineering product development, Computer Methods in Applied Mechanics and Engineering, 200 (2011) 1421-1431.
[5] S. Wang, W. Chen, K.-L. Tsui, Bayesian Validation of Computer Models, Technometrics, 51 (2009) 439-451.
[6] Z. Hu, D. Ao, S. Mahadevan, Calibration experimental design considering field response and model uncertainty, Computer Methods in Applied Mechanics and Engineering, 318 (2017) 92-119.
[7] Y. Ling, J. Mullins, S. Mahadevan, Selection of model discrepancy priors in Bayesian calibration, Journal of Computational Physics, 276 (2014) 665-680.
[8] P.D. Arendt, D.W. Apley, W. Chen, A preposterior analysis to predict identifiability in the experimental calibration of computer models, IIE Transactions, 48 (2016) 75-88.
[9] R. Rocchetta, M. Broggi, Q. Huchet, E. Patelli, On-line Bayesian model updating for structural health monitoring, Mechanical Systems and Signal Processing, 103 (2018) 174-195.
[10] F. Schneider, I. Papaioannou, D. Straub, C. Winter, G. Müller, Bayesian parameter updating in linear structural dynamics with frequency transformed data using rational surrogate models, Mechanical Systems and Signal Processing, 166 (2022) 108407.
[11] A. Kamariotis, E. Chatzi, D. Straub, Value of information from vibration-based structural health monitoring extracted via Bayesian model updating, Mechanical Systems and Signal Processing, 166 (2022) 108465.
[12] H.-P. Wan, W.-X. Ren, Stochastic model updating utilizing Bayesian approach and Gaussian process model, Mechanical Systems and Signal Processing, 70-71 (2016) 245-268.
[13] M. Kitahara, S. Bi, M. Broggi, M. Beer, Nonparametric Bayesian stochastic model updating with hybrid uncertainties, Mechanical Systems and Signal Processing, 163 (2022) 108195.
[14] I. Behmanesh, B. Moaveni, G. Lombaert, C. Papadimitriou, Hierarchical Bayesian model updating for structural identification, Mechanical Systems and Signal Processing, 64-65 (2015) 360-376.
[15] K.E. Tatsis, K. Agathos, E.N. Chatzi, V.K. Dertimanis, A hierarchical output-only Bayesian approach for online vibration-based crack detection using parametric reduced-order models, Mechanical Systems and Signal Processing, 167 (2022) 108558.
[16] M. Song, B. Moaveni, C. Papadimitriou, 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 (2019) 68-83.
[17] P. Ni, Q. Han, X. Du, X. Cheng, Bayesian model updating of civil structures with likelihood-free inference approach and response reconstruction technique, Mechanical Systems and Signal Processing, 164 (2022) 108204.
[18] J. Chiachío, N. Bochud, M. Chiachío, S. Cantero, G. Rus, A multilevel Bayesian method for ultrasound-based damage identification in composite laminates, Mechanical Systems and Signal Processing, 88 (2017) 462-477.
[19] M.K. Ramancha, R. Astroza, R. Madarshahian, J.P. Conte, Bayesian updating and identifiability assessment of nonlinear finite element models, Mechanical Systems and Signal

Processing, 167 (2022) 108517.
[20] M.C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63 (2001) 425-464.
[21] M.C. Kennedy, A. O’Hagan, Supplementary details on Bayesian calibration of computer models, Internal Report. URL http://www. shef. ac. uk/~ st1ao/ps/calsup. ps, 2001.
[22] C. Jiang, Z. Hu, Y. Liu, Z.P. Mourelatos, D. Gorsich, P. Jayakumar, A sequential calibration and validation framework for model uncertainty quantification and reduction, Computer Methods in Applied Mechanics and Engineering, 368 (2020) 113172.
[23] C. Jiang, M.A. Vega, M.D. Todd, Z. Hu, Model correction and updating of a stochastic degradation model for failure prognostics of miter gates, Reliability Engineering & System Safety, 218 (2022) 108203.
[24] P.D. Arendt, D.W. Apley, W. Chen, Quantification of model uncertainty: Calibration, model discrepancy, and identifiability, Journal of Mechanical Design, 134 (2012) 100908.
[25] N.-C. Xiao, K. Yuan, H. Zhan, System reliability analysis based on dependent Kriging predictions and parallel learning strategy, Reliability Engineering & System Safety, 218 (2022) 108083.
[26] Z. Meng, Z. Zhang, G. Li, D. Zhang, An active weight learning method for efficient reliability assessment with small failure probability, Structural and Multidisciplinary Optimization, 61 (2020) 1157-1170.
[27] Z. Dzunic, J.G. Chen, H. Mobahi, O. Büyüköztürk, J.W. Fisher, A Bayesian state-space approach for damage detection and classification, Mechanical Systems and Signal Processing, 96 (2017) 239-259.
[28] J. Ringsberg, M. Loo-Morrey, B. Josefson, A. Kapoor, J.H. Beynon, Prediction of fatigue crack initiation for rolling contact fatigue, International Journal of fatigue, 22 (2000) 205-215.
[29] B.A. Eick, Z.R. Treece, B.F. Spencer, M.D. Smith, S.C. Sweeney, Q.G. Alexander, S.D. Foltz, Automated damage detection in miter gates of navigation locks, Structural Control and Health Monitoring, 25 (2018) e2053.
[30] M.A. Vega, M.D. Todd, A variational Bayesian neural network for structural health monitoring and cost-informed decision-making in miter gates, Structural Health Monitoring, (2020) 147592172090454.
[31] M.A. Vega, Z. Hu, M.D. Todd, Optimal maintenance decisions for deteriorating quoin blocks in miter gates subject to uncertainty in the condition rating protocol, Reliability Engineering & System Safety, 204 (2020) 107147.
[32] P. Raghavan, S. Ghosh, Concurrent multi-scale analysis of elastic composites by a multi-level computational model, Computer Methods in Applied Mechanics and Engineering, 193 (2004) 497-538.
[33] V.P. Nguyen, M. Stroeven, L.J. Sluys, Multiscale failure modeling of concrete: Micromechanical modeling, discontinuous homogenization and parallel computations, Computer Methods in Applied Mechanics and Engineering, 201-204 (2012) 139-156.
[34] K. Sargsyan, X. Huan, H.N. Najm, Embedded model error representation for Bayesian model calibration, International Journal for Uncertainty Quantification, 9 (2019) 365-394.
[35] A. Frankel, E. Wagman, R. Keedy, B. Houchens, S.N. Scott, Embedded-error Bayesian calibration of thermal decomposition of organic materials, Journal of Verification, Validation and Uncertainty Quantification, 6 (2021) 041002.
[36] A. Forrester, A. Sobester, A. Keane, Engineering design via surrogate modelling: a practical guide, John Wiley & Sons, 2008.
[37] Y. Heo, R. Choudhary, G.A. Augenbroe, Calibration of building energy models for retrofit analysis under uncertainty, Energy and Buildings, 47 (2012) 550-560.
[38] D. Higdon, C. Nakhleh, J. Gattiker, B. Williams, A Bayesian calibration approach to the thermal problem, Computer Methods in Applied Mechanics and Engineering, 197 (2008)

2431-2441.

[39] A. Chong, K. Menberg, Guidelines for the Bayesian calibration of building energy models, Energy and Buildings, 174 (2018) 527-547.

[40] L. Zhang, Z. Jiang, J. Choi, C.Y. Lim, T. Maiti, S. Baek, Patient-Specific Prediction of Abdominal Aortic Aneurysm Expansion Using Bayesian Calibration, IEEE J Biomed Health Inform, 23 (2019) 2537-2550.

[41] A. Urbina, S. Mahadevan, T.L. Paez, A Bayes network approach to uncertainty quantification in hierarchically developed computational models, International Journal for Uncertainty Quantification, 2 (2012) 173-193.

[42] J.M. McFarland, Uncertainty Analysis for Computer Simulations through Validation and Calibration, Vanderbilt University, 2008.

[43] B.C. Jung, H. Yoon, H. Oh, G. Lee, M. Yoo, B.D. Youn, Y.C. Huh, Hierarchical model calibration for designing piezoelectric energy harvester in the presence of variability in material properties and geometry, Structural and Multidisciplinary Optimization, 53 (2016) 161-173.

[44] F. Liu, M. Bayarri, J. Berger, R. Paulo, J. Sacks, A Bayesian analysis of the thermal challenge problem, Computer Methods in Applied Mechanics and Engineering, 197 (2008) 2457-2466.

[45] D.F. Specht, A general regression neural network, IEEE transactions on neural networks, 2 (1991) 568-576.

[46] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and computing, 14 (2004) 199-222.

[47] K. Li, J. Liu, X. Han, X. Sun, C. Jiang, A novel approach for distributed dynamic load reconstruction by space–time domain decoupling, Journal of Sound and Vibration, 348 (2015) 137-148.

[48] L. Wang, Y. Liu, K. Gu, T. Wu, A radial basis function artificial neural network (RBF ANN) based method for uncertain distributed force reconstruction considering signal noises and material dispersion, Computer Methods in Applied Mechanics and Engineering, 364 (2020) 112954.

[49] Y. Liu, Adomian decomposition method with orthogonal polynomials: Legendre polynomials, Mathematical and Computer Modelling, 49 (2009) 1268-1273.

[50] W.H. Press, G.R. Farrar, Recursive stratified sampling for multidimensional Monte Carlo integration, Computers in Physics, 4 (1990) 190-195.

[51] J. Shen, H.J.S.J.o.S.C. Yu, Efficient spectral sparse grid methods and applications to high-dimensional elliptic problems, 32 (2010) 3228-3250.

[52] S. Rahman, H.J.P.E.M. Xu, A univariate dimension-reduction method for multi-dimensional integration in stochastic mechanics, 19 (2004) 393-408.

[53] T. Gerstner, M. Griebel, Numerical integration using sparse grids, Numerical algorithms, 18 (1998) 209.

[54] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation, 9 (1997) 1735-1780.

[55] C. Olah, Understanding LSTM Networks, https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[56] X. Zhang, S. Mahadevan, Bayesian neural networks for flight trajectory prediction and safety assessment, Decision Support Systems, 131 (2020) 113246.

[57] Z. Hu, S.J.R.E. Mahadevan, S. Safety, Probability models for data-driven global sensitivity analysis, 187 (2019) 40-57.

[58] K. Csilléry, M.G. Blum, O.E. Gaggiotti, O.J.T.i.e. François, evolution, Approximate Bayesian computation (ABC) in practice, Trends in ecology & evolution, 25 (2010) 410-418.

[59] U.S. Army Corps of Engineers Headquarters. Navigation 2018,

https://www.usace.army.mil/Missions/Civil-Works/Navigation/.

[60] S.D. Foltz, Investigation of mechanical breakdowns leading to lock closures, ERDC-CERL CHAMPAIGN United States, 2017.

[61] M. Parno, D. O'Connor, M. Smith, High dimensional inference for the structural health monitoring of lock gates, arXiv preprint arXiv:1812.05529, (2018).

[62] V.L. Popov, Contact mechanics and friction, Berlin: Springer Berlin Heidelberg, 2010.

[63] C.E. Rasmussen, C.K. Williams, Gaussian processes for machine learning, MIT press Cambridge, MA, 2006.

[64] J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, Design and Analysis of Computer Experiments, Statistical Science, 4 (1989) 409-423.

[65] D.R. Jones, A Taxonomy of Global Optimization Methods Based on Response Surfaces, Journal of Global Optimization, 21 (2001) 345-383.