Simulation Failure Robust Bayesian Optimization for Data-Driven Parameter Estimation

Chakrabarty, Ankush; Bortoff, Scott A.; Laughman, Christopher R.

TR2022-168 December 20, 2022

Abstract

Advances in modeling and computation have resulted in high-fidelity digital twins capable of simulating the dynamics of a wide range of industrial systems. These simulation models often require calibration, or the estimation of an optimal set of parameters in some goodness-of-fit sense, to reflect a system's observed behavior. While searching over the parameter space is an inevitable part of the calibration process, simulation models are rarely designed to be valid for arbitrarily large parameter spaces. Application of existing calibration methods, therefore, often results in repeated model evaluations using parameters that can cause the simulations to be impractically slow or even result in catastrophic failure. In general, the shape of subregions in the parameter space that could result in simulation failure is unknown. In this paper, we propose a novel failure- robust Bayesian optimization (FR-BO) algorithm that learns these failure regions from online simulations and informs a Bayesian optimization algorithm to avoid failure regions while optimizing model parameters. This results in acceleration of the optimizer's convergence and prevents wastage of time trying to simulate parameters with high failure probabilities. The effectiveness of the proposed failure-robust Bayesian optimization algorithm is demonstrated via a well-known benchmark example where we compare against state-of-the-art gradient matching techniques, and a practical example related to parameter es- timation for digital twins of buildings.

IEEE Transactions on Systems, Man, and Cybernetics: Systems 2022

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Simulation Failure Robust Bayesian Optimization for Data-Driven Parameter Estimation

Ankush Chakrabarty[†], *Senior Member, IEEE*, Scott A. Bortoff, and Christopher R. Laughman[*]

***Abstract*—Advances in modeling and computation have resulted in high-fidelity digital twins capable of simulating the dynamics of a wide range of industrial systems. These simulation models often require calibration, or the estimation of an optimal set of parameters in some goodness-of-fit sense, to reflect a system's observed behavior. While searching over the parameter space is an inevitable part of the calibration process, simulation models are rarely designed to be valid for arbitrarily large parameter spaces. Application of existing calibration methods, therefore, often results in repeated model evaluations using parameters that can cause the simulations to be impractically slow or even result in catastrophic failure. In general, the shape of subregions in the parameter space that could result in simulation failure is unknown. In this paper, we propose a novel failure-robust Bayesian optimization (FR-BO) algorithm that learns these failure regions from online simulations and informs a Bayesian optimization algorithm to avoid failure regions while optimizing model parameters. This results in acceleration of the optimizer's convergence and prevents wastage of time trying to simulate parameters with high failure probabilities. The effectiveness of the proposed failure-robust Bayesian optimization algorithm is demonstrated via a well-known benchmark example where we compare against state-of-the-art gradient matching techniques, and a practical example related to parameter estimation for digital twins of buildings.**

***Index Terms*—Digital twin; machine learning; Bayesian optimization; simulation; dynamical systems; system identification; numerical methods; Gaussian processes.**

I. Introduction

Current trends towards model-based system development and the application of digital twins place an increasing emphasis on the use of modeling and simulation for large-scale systems [1], [2]. One essential step in the development of these technologies involves model calibration, which ensures that the simulation models accurately represent observed system behavior [3]. Simulation models and digital twins are ubiquitous in modern engineering applications, including building energy systems [4], [5], biomedical systems [6], spacecraft [7], battery charging [8], vehicles [9], and networked systems such as power [10] or water [11] distribution networks.

It is rare for a closed-form solution to high-fidelity simulation models to exist, so iterative methods that leverage simulation data are widely used to compute parameters that result in the model exhibiting good predictive performance [12]. Usually, data-driven model calibration requires simulating a model forward for a prescribed timespan using a candidate set of parameter values and comparing this solution with the observational data [13]. The candidate set is then updated in a way that tends to produce a better solution and the process is repeated until some terminating criterion is met [14]. A common goodness-of-fit metric used to obtain parameters is the sum of squared error. Assuming a prior distribution for the data generation process, one can use maximum likelihood estimators to obtain parameter estimates and quantify the uncertainty associated with these estimates [15]–[17]. However, these methods rarely result in globally optimal parameter sets

TABLE I
List of mathematical symbols.

Symbol	Meaning
Preliminaries	
$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of natural numbers
$\log$	Natural logarithm operator
T	Total simulation time
$\mathcal{N}(\mu, \sigma)$	Gaussian density with mean μ and variance σ^2
θ	Parameters to be calibrated
$\theta^\star$	Optimal parameter vector
Θ	Search space of admissible parameters
n_θ	Number of calibrated parameters
n_y	Number of measurable outputs
$y_{0:T}$	Simulated output vector on time interval $[0, T]$
$y^\star_{0:T}$	True measured outputs on time interval $[0, T]$
$\mathcal{M}_T(\theta)$	Forward model for simulation parameterized by θ
$\mathcal{M}_T(\theta^\star)$	Optimally parameterized forward model for simulation
J	Calibration cost function
Classical Bayesian Optimization	
GP	Gaussian process regression
$\mathcal{D}_j$	Parameter-cost paired data at j-th BO iteration
$\hat{J}_j$	GP's estimated cost with dataset $\mathcal{D}_j$
$\mathcal{K}$	Kernel function used in GP regression
μ	Mean function used in GP
σ	Standard deviation function used in GP
EI	Expected improvement acquisition function
γ, Γ	PDF and CDF of $\mathcal{N}(0, 1)$ Gaussian distribution
Failure-Robust Bayesian Optimization	
$\Theta_\mathsf{F}, \hat{\Theta}_\mathsf{F}$	Failure region (estimate)
ℓ	Failure label
$\mathcal{D}_j$	Parameter-label-cost tuple data at j-th FRBO iteration
ϕ	Prior distribution of VGPC
$\mathcal{B}$	Bernoulli distribution
p	Posterior distribution of VGPC
q	Variational approximation used in VGPC
PF	Failure probability assigned by VGPC
$\tilde{K}$	Kernel matrix for VGPC
E	Expectation operator
KL	Kullback–Leibler divergence
L	Cholesky decomposition
β	regularization coefficient in VGPC loss
Var	Variance operator
FREI	Failure-robust expected improvement acquisition function

[†] Corresponding author. Email: `achakrabarty@ieee.org`. Phone: +1 (617) 758-6175.

[*] All authors are affiliated with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA.

and often get stuck in local optima [18]. Bayesian methods allow for a richer characterization of uncertainty and incorporate prior information about model parameters, enabling parameter estimation over wide ranges of parameters [5], [19], [20]. However, these methods often require a large number of model simulations, which is often impractical, as each model simulation is time-consuming and computationally expensive. An alternative model-free approach that exists in the literature is gradient matching [21]. Gradient matching interpolates the right-hand-side of the model ODEs directly from the data using Gaussian process models [22]–[25]. This interpolant, or meta-model, is fast to execute, but typically does not scale beyond tens of states, which limits their utility in many applications modeled by high-dimensional simulation models.

Due to the black-box nature of modern high-fidelity software-based simulation models, it is not uncommon for simulations to fail at some combination of parameter values. For instance, many existing simulation-oriented models exhibit multi-scale dynamics, significant nonlinearities, and numerically stiff behavior that can result in simulations that take a significant amount of time to run or fail entirely due to the complex and non-intuitive shape of the admissible parameter set [26]. These challenges are particularly common in building energy models that seek to describe the temporal behavior of occupied buildings with their associated closed-loop space conditioning systems [27], due to their widely separate timescales, hybrid continuous/discrete behavior, and nonlinear interactions between physical subsystems. Rather than expend significant effort to identify parameter sets that result in valid simulations, it is common practice to ignore the information conveyed by a failed simulation for a given parameter set and simply re-run the simulation with a different set of parameters.

Practical calibration methods are often designed to estimate near-optimal parameters without extensive simulations to avoid this expenditure of significant time and resources without a corresponding increase in simulation performance. Recently, Bayesian optimization (BO) [28] has emerged as an effective method for learning parameters based on limited data in a few-shot manner [4], [29]: that is, with markedly fewer evaluations of the cost function (equivalently, model simulations) than population-based methods. Furthermore, Bayesian optimization inherently balances exploration and exploitation and can incorporate non-convex constraints via modified acquisition functions [30], making it a powerful and easy-to-use learner for model calibration.

In this paper, we use the information that results from a failed simulation during the model calibration process to accelerate the convergence of these methods and improve the quality of parameter estimates. We thus propose a novel variant of BO called *failure-robust Bayesian optimization* (FR-BO), which comprises modules that use simulated data to ascertain regions in the parameter space where the system is likely to fail. Subsequently, we design a novel acquisition function, inspired by constrained BO acquisition functions [30], that allows searching for optimizer candidates that not only fit the data well, but also are unlikely to result in simulation failures. Specific contributions of this work include: (i) identification of the phenomenon of simulation failure during model calibration tasks, as predominant approaches in the extant literature either use heuristics to ignore simulation failure or simplify the simulation models to avoid failures in the search-space; (ii) innovation of a generalizable method that can reduce computational time and resources pursuing simulations that are likely to result in failure; (iii) proposal of a novel acquisition function in the Bayesian optimization algorithm to automatically avoid regions likely to result in simulation failure during the calibration process; and, (iv) demonstration of the potential of the proposed FRBO approach by comparing performance against other state-of-the-art algorithms.

The rest of the paper is organized as follows. In Section II, we formally present the problem of data-driven parameter estimation. The proposed FR-BO algorithm is presented in greater detail in Section III. In Section IV, we demonstrate the potential of our proposed FR-BO method on a benchmark example on which we show that FR-BO can outperform cutting-edge gradient matching methods, and a real-world building envelope model in Modelica software which has no simple closed-loop representation and whose parameters cannot be obtained from first-principles knowledge. Based on our simulation experiments, we report that: (i) our proposed FR-BO algorithm converges significantly faster than classical BO or MCMC methods on a benchmark example; (ii) the quality of model calibration using FR-BO can outperform competitive gradient-matching algorithms and their fast variants; (iii) the method can scale to simulation models with many parameters; (iv) the total amount of wasted time during simulations can be curtailed using FR-BO; and, (v) our algorithm is easy to implement using open-source machine learning toolkits such as `PyTorch`.

II. Preliminaries

A. Background

We denote by

$$y_{0:T} = \mathcal{M}_T(\theta) \tag{1}$$

a general model of a dynamical system, where the constant parameters of the model are described by $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$, where n_θ is the number of model parameters. The admissible set of parameters Θ is assumed to be known. For instance, Θ could denote a set of upper and lower bounds on parameters obtained from archived data or domain knowledge. Since the model is black-box, it is not uncommon for such a range to be purely a guess, and therefore, not tight around the true parameter set. The output vector $y_{0:T} \in \mathbb{R}^{n_y \times T}$ contains all measured outputs from the dynamical system obtained over a time period $[0, T]$; note that n_y denotes the number of measured outputs, that is, $y_t \in \mathbb{R}^{n_y}$. We do not make any assumptions on the underlying structure of the model $\mathcal{M}_T(\theta)$, where simulating $\mathcal{M}_T(\theta)$ forward with a fixed (and admissible) set of parameters θ yields a vector of outputs

$$y_{0:T} := \begin{bmatrix} y_0 & y_1 & \cdots & y_t & \cdots & y_T \end{bmatrix},$$

with each output measurement $y_t \in \mathbb{R}^{n_y}$.

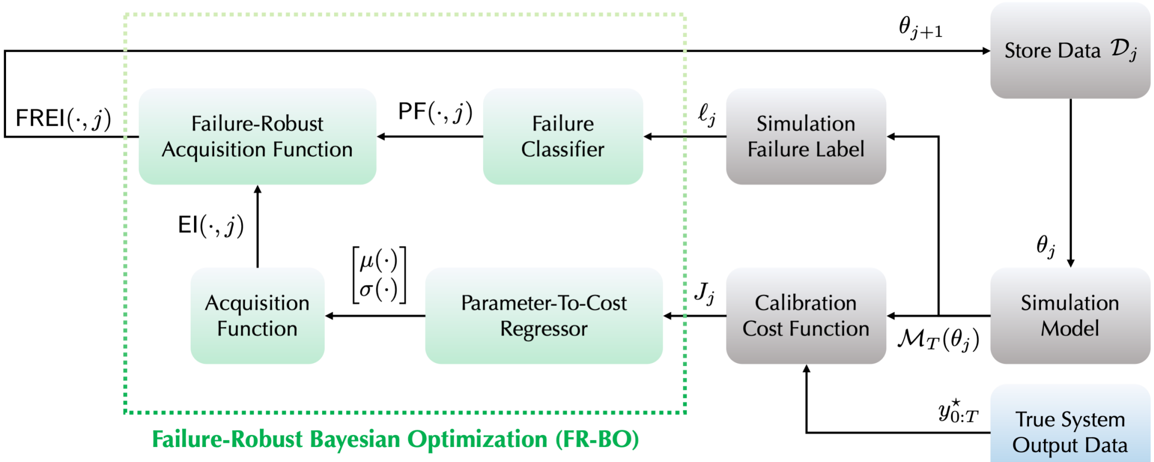

θ_{j+1}
Store Data $\mathcal{D}_j$
FREI$(\cdot, j)$
Failure-Robust Acquisition Function
PF$(\cdot, j)$
Failure Classifier
ℓ_j
Simulation Failure Label
EI$(\cdot, j)$
$\begin{bmatrix}\mu(\cdot)\\ \sigma(\cdot)\end{bmatrix}$
θ_j
Acquisition Function
Parameter-To-Cost Regressor
J_j
Calibration Cost Function
Simulation Model
$\mathcal{M}_T(\theta_j)$
$y^\star_{0:T}$
True System Output Data
Failure-Robust Bayesian Optimization (FR-BO)

Fig. 1. Overall schematic of the proposed parameter estimation scheme using FR-BO.

Example 2.1: Consider a state-space representation of a nonlinear dynamical system

$$\dot{x} = f_{\mathsf{ODE}}(x, u, \theta_1), \qquad y = h_{\mathsf{ODE}}(x, u, \theta_2),$$

with state x, initial condition x_0, control input u, and parameters θ_1 and θ_2. Given a set of parameters $\theta := \{\theta_1\} \cup \{\theta_2\}$, one can numerically integrate (i.e., simulate) the system of ordinary differential equations forward for $t \in [0, T]$, and subsequently obtain the sequence of outputs $y_{0:T}$. Thus, these system dynamics can be represented by a parameter-to-output map as in (1). ◇

Example 2.2: Following the same rationale as in Example 2.1, the class of dynamical systems represented by differential algebraic equations

$$0 = f_{\mathsf{DAE}}(\dot{x}, x, u, \theta_1), \qquad y = h_{\mathsf{DAE}}(x, u, \theta_2)$$

can also be modeled by (1). ◇

We aim to estimate parameters $\theta^\star$ that minimize (in some sense) the modeling error

$$\varepsilon \triangleq y^\star_{0:T} - \mathcal{M}_T(\theta^\star), \tag{2}$$

where $y^\star_{0:T}$ denotes the measured outputs collected from a real system, and $\mathcal{M}_T(\theta^\star)$ denotes the estimated outputs from the model $\mathcal{M}_T(\theta)$ using the estimated parameters $\theta^\star$. To this end, we propose optimizing a calibration cost function $J(y^\star_{0:T}, \mathcal{M}_T(\theta))$ to yield the optimal parameters

$$\theta^\star = \arg\min_{\theta \in \Theta} J(y^\star_{0:T}, \mathcal{M}_T(\theta)). \tag{3}$$

Recent work has shown that Bayesian optimization (BO) is effective at finding global optima of functions whose gradients are not available and are expensive to evaluate, as is the case in black-box model calibration [4].

B. Classical Bayesian optimization

Classical BO methods assume the presence of one global optimum, and smoothness of the θ to J map. Since J is typically assumed to be continuous, one can leverage the data at the j-th iteration to construct a surrogate GP model of the reward, given by

$$\hat{J}_j := \mathsf{GP}\left(\mu(\theta; \mathcal{D}_j), \sigma(\theta, \theta'; \mathcal{D}_j)\right), \tag{4}$$

where $\mu(\cdot)$ is the predictive mean function, and $\sigma(\cdot, \cdot)$ is the predictive variance function, and $\mathcal{D}_j$ containing $\{\theta_{[0:j]}, J_{[0:j]}\}$ is the dataset collected thus far. Typically, the variance is expressed through the use of kernel functions [28].

At the j-th learning iteration, for a new query sample $\theta \in \Theta$, the GP model predicts the mean and variance of the reward to be

$$\mu(\theta) = k_j(\theta)^\top K_j^{-1} J_{0:j}$$

and

$$\sigma(\theta) = \mathcal{K}(\theta, \theta) - k_j(\theta) K_j^{-1} k_j(\theta)^\top,$$

where

$$k_j(\theta) = \begin{bmatrix}\mathcal{K}(\theta_0, \theta) & \mathcal{K}(\theta_1, \theta) & \cdots & \mathcal{K}(\theta_j, \theta)\end{bmatrix},$$

and K_j is defined in (6).

The accuracy of predicted mean and variance is strongly linked to the selection of the kernel and the best (in some sense) set of hyperparameters such as length scales and variance parameters of the kernels and estimated noise. We obtain these hyperparameters by maximizing the log marginal likelihood function (MLL)

$$-\frac{1}{2}\log|K_j| - \frac{1}{2}J(\theta)^\top K_j^{-1} J(\theta) - \frac{n_\theta}{2}\log 2\pi.$$

This optimization problem constitutes training the GP with a fixed kernel. Although the problem is non-convex, it can be solved efficiently to local optima using quasi-Newton methods or adaptive gradient methods. A full derivation of the MLL function can be found in [31, Chapter 2]. Note that the MLL function is maximized at every BO iteration, in an online manner, since the variance, length-scale, and noise parameters need to be re-learned every time the dataset is updated.

In Bayesian optimization, we use the mean and variance of the surrogate model $\hat{J}_j$ in (4) to construct an acquisition function to inform the selection of a θ_j that increases the likelihood of minimizing the current best cost. To this end, we compute the incumbent $\hat{J}_j^\star := \min_{\theta\in\Theta} \mu(\theta;\mathcal{D}_j)$ and use it to define an expected improvement (EI) acquisition function that has the form

$$\mathsf{EI}(\theta, j) = \begin{cases} \sigma(\theta)\gamma(z) + (\hat{J}_j^\star - \mu(\theta))\Gamma(z), & \text{if } \sigma(\theta) > 0, \\ 0 & \text{if } \sigma(\theta) = 0. \end{cases}$$

where $z = \frac{\hat{J}_j^\star - \mu(\theta)}{\sigma(\theta)}$, and $\gamma(\cdot)$, $\Gamma(\cdot)$ are the PDF and the CDF of the zero-mean unit-variance normal distribution, respectively.

In the j-th iteration of learning, we use the data $\mathcal{D}_j$ to construct the EI acquisition function using the surrogate $\hat{J}_j$. Subsequently, we compute the optimizer candidate

$$\theta_{j+1} = \arg\max_{\theta\in\Theta} \mathsf{EI}(\theta, j), \tag{5}$$

which serves as the parameter estimate θ in (1) in the next BO iteration. In practice, other acquisition functions such as the lower confidence bound or entropy search could also be used instead of EI [28].

C. Problem Statement and Proposed Solution

While various methods have been proposed for solving (3), most (if not all) these solutions assume that $\mathcal{M}_T(\theta)$ exists for every $\theta \in \Theta$, which implies that the model $\mathcal{M}_T(\theta)$ can be simulated from the time-span of interest $[0, T]$ for any parameter in the admissible parameter space Θ. Unfortunately, this is not always the case and model simulations can fail to complete in a timespan of interest. By failure, we include scenarios such as: (i) a numerical integration scheme terminates prematurely due to parameter-dependent stiffness in the underlying dynamics; (ii) the underlying dynamics do not have a solution due to parameters not adhering to basic validity assumptions, for instance, if the underlying system has $\log(1-\theta^2)$ terms and $\{|\theta| > 1\} \subset \Theta$; (iii) a subset of the Θ renders the underlying dynamics unstable (*e.g.*, $\dot{x} = \theta x^2$ for $\theta > 0$) or a controller/estimator designed using an approximation of the model (such as a linearization) makes the closed-loop dynamics unstable; and, (iv) the simulation takes exorbitantly long for some parameters so the code is terminated based on heuristics after a prefixed termination time; to name a few. Models that exhibit simulation failures typically do so in some *failure region* $\Theta_\mathsf{F} \subset \Theta$ and the failure does not always occur instantaneously. For instance, in the case (iv) of the previous paragraph, the failure will be flagged after a designated termination time that could be large. Consequently, data-driven algorithms that have been designed agnostic to simulation failure could potentially continue to compute optimizer candidates that reside in the failure region Θ_F. In such cases, the algorithm could deteriorate in performance and lead to large amounts of computational resources and CPU time being wasted.

Under the critical assumption that $\theta^\star \notin \Theta_\mathsf{F}$, our **objective** is to design a data-driven parameter estimation framework that can learn from simulation failures and incorporate this information to increase the probability of selecting sets of parameters that lead to successful simulations, thereby enabling us to optimize (3) without wasting computational resources and time. To fulfill this objective, we propose a failure-robust Bayesian optimization (FR-BO) approach wherein we first design a probabilistic classifier that can estimate the failure region Θ_F from simulation data obtained by sampling within Θ. Since function evaluations are assumed to be expensive, we employ an entropy-based active learning method to reduce the sample complexity of this step [32]. Once an estimate $\hat{\Theta}_\mathsf{F}$ of Θ_F is obtained, the classifier provides probabilities of simulation failure over the entire parameter space of interest Θ. Failure probabilities can be embedded into a Bayesian optimization framework through a failure-classifier informed acquisition function, ensuring that optimizer candidates are biased to reside outside $\hat{\Theta}_\mathsf{F}$. We posit that if the classifier is well designed, it will suggest optimizer candidates that lie within the set difference $\Theta \setminus \Theta_\mathsf{F}$ with high probability.

We reiterate the workflow of the FR-BO algorithm more concretely in Fig. 1. We begin from the bottom right, where the inputs to the algorithm are a simulation model $\mathcal{M}_T(\theta)$ parameterized by θ_j in the j-th iteration of FR-BO, and the true system output data used for calibration. The simulation model is simulated forward for the time T of interest, and the model and true outputs contribute to a calibration cost value J_j if the simulation is completed successfully. Additionally, based on the success/failure of the simulation, a label is assigned. The label and prior θ values inform a binary classifier that estimates the failure region of the calibration problem, and embeds this information into a customized failure-robust acquisition function computed using (θ_j, J_j) pairs in order to compute the next most promising candidate θ_{j+1} to use for simulations. By iterating through these steps, the FR-BO attains an optimizer

$$\theta^\star = \arg\min_{\theta\in\Theta\setminus\Theta_\mathsf{F}} J(y^\star_{0:T}, \mathcal{M}_T(\theta))$$

while avoiding parameters likely to result in failed simulations. To summarize, the main components of the FR-BO framework include:

(i) a failure classifier for estimating likelihoods of simulation success and failure on Θ by learning the set Θ_F;
(ii) a parameter-to-cost regressor for approximating the calibration cost function J from any parameter $\theta \in \Theta$; and
(iii) a failure-robust acquisition function that incorporates probability of simulation failure into the optimization framework.

III. Failure Robust Bayesian Optimization

In this section, we describe a way to learn the failure region from data obtained during simulations via a scalable variational Gaussian process classifier (VGPC) [33]. The probabilities can be used via active learning to accelerate the Bayesian optimization step, but also to guide where best to simulate the dynamical model $\mathcal{M}_T(\theta)$ to get better estimates of the failure region boundaries. We also describe the steps involved in classical Bayesian optimization, and explain how to incorporate information from the VGPC via a constraint-weighted acquisition function to ensure that parameters are chosen avoiding failure regions.

A. Learning Failure Regions

1) Data collection: Since the admissible parameter search domain Θ is known, one can sample on this space to obtain a training set for learning the failure region subset Θ_{F}. In the sequel, we will discuss how to select $\theta \in \Theta$ that are most informative (in an information-theoretic sense), but we will assume that such a training dataset is initially available, for instance, obtained by random sampling on Θ. At the j-th iteration of training the failure classifier, the training dataset

$$\mathcal{D}_j = \theta_{[0:j]} \times \ell_{[0:j]} \times J_{[0:j]}$$

comprises a sequence of parameters $\theta_{[0:j]}$, a sequence of corresponding simulation failure labels $\ell_{[0:j]}$, and cost function values $J_{[0:j]}$. Each failure label is denoted $+1$ if there is simulation failure and -1 if not. For failed simulations, we set the corresponding cost function value to some nonsense value, e.g., `NaN`. If the simulation is successful, the cost function yields a real-valued scalar.

2) Scalable variational Gaussian process classifiers: At the j-th iteration of learning the failure region, one can utilize the θ and labels ℓ of the dataset $\mathcal{D}_j$ to set a Gaussian process prior at the observed parameter sets. This can be written as $\phi \sim \mathcal{N}(0, K_j)$, where ϕ_j is the prior function using $\mathcal{D}_j$ and

$$K_j = \begin{bmatrix} \mathcal{K}(\theta_0, \theta_0) & \cdots & \mathcal{K}(\theta_0, \theta_j) \\ \vdots & \ddots & \vdots \\ \mathcal{K}(\theta_j, \theta_0) & \cdots & \mathcal{K}(\theta_j, \theta_j) \end{bmatrix}, \tag{6}$$

with a user-specified kernel function $\mathcal{K}(\cdot,\cdot)$ such as a squared exponential kernel or a Matern kernel; see [31] for more details about kernel functions.

To perform classification with this prior, one needs to transform the function ϕ through a squashing function such as the cumulative density function of a zero-mean unit-variance normal distribution $\gamma(\cdot) := \mathcal{N}(\cdot|0,1)$, given by $\Gamma(z) = \int_{-\infty}^{z} \gamma(\alpha)\,\mathrm{d}\alpha$. Consequently, a Bernoulli distribution can be used to represent a likelihood function conditioned on the transformed data as follows:

$$\mathcal{B}(\ell_j|\Gamma(\phi_j)) = \Gamma(\phi_j)^{\ell_j} \cdot (1 - \Gamma(\phi_j)^{1-\ell_j}).$$

The joint distribution of ℓ and ϕ thus becomes

$$p(\ell, \phi) = \prod_{r=1}^{j} \mathcal{B}\left(\ell_j|\Gamma(\phi_j)\right) \mathcal{N}(0, K_j). \tag{7}$$

Two more distributions are required to optimize hyperparameters and perform inference: the marginal likelihood $\mathsf{PF}(\ell, j)$ and the posterior $p(\phi|\ell, j)$. Both of these distributions require the inversion of the $j \times j$ kernel matrix (6), which incurs cubic complexity and does not scale well to the large values of j that may be required for FR-BO to compute good solutions. We therefore resort to the use of approximation methods that leverage pseudo-inputs, which are more commonly known as inducing points [34].

Inducing points $\tilde{\theta} \in \Theta$ are design variables that are augmented with the latent variables ϕ_j that also respect the Gaussian prior and therefore yield a joint distribution

$$(\phi, \tilde{\phi}) \sim \mathcal{N}\left(0, \begin{bmatrix} K_j & \tilde{K}_{jm} \\ \tilde{K}_{jm}^\top & \tilde{K}_m \end{bmatrix}\right).$$

$\tilde{K}_{jm}$ denotes the covariance matrix computed by evaluating the kernel across j data points and m inducing inputs, while $\tilde{K}_m$ denotes the covariance matrix computed by evaluating the kernel on all pairs of the inducing inputs. Exploiting the properties of the Gaussian distribution, one can rewrite the joint distribution of the latent variables and the inducing variables as

$$p(\ell, \phi, \tilde{\phi}) = p(\ell|\phi) p(\phi|\tilde{\phi}) p(\tilde{\phi}).$$

To get a variational approximation of the likelihood, the following inequality is used from [33]:

$$\log p(\ell|\tilde{\phi}) \geq \mathsf{E}_{p(\phi|\tilde{\phi})}[\log p(\ell|\phi)].$$

Defining a variational distribution q, we get the well-known variational lower bound

$$\log p(\ell) \geq \mathsf{E}_{q(\phi)}[\log p(\ell|\phi)] - \mathsf{KL}[q(\tilde{\phi}) \| p(\tilde{\phi})]. \tag{8}$$

The optimal hyperparameters for the VGPC can be obtained by minimizing the loss function formed by the negative of the right hand side of this inequality using quadrature methods. If we assume $q \sim \mathcal{N}(\tilde{\phi}|\tilde{\mu}, \tilde{\Sigma})$, then

$$q(\phi) = \mathcal{N}(L\tilde{\mu}, K_j + L(\tilde{\Sigma} - \tilde{K}_m)L^\top), \tag{9}$$

with $L = \tilde{K}_{jm}\tilde{K}_m^{-1}$, which is an $m \times m$ matrix, and eventually $m \ll j$, so this matrix is cheaper to invert, which makes this method scalable.

Note that one could additionally introduce a regularization parameter $\beta \in (0, 1)$ in the training loss function to trade-off the effect of the KL-divergence term, as suggested in [35]. In such a case, the variational lower bound (8) would become

$$\log p(\ell) \geq \mathsf{E}_{q(\phi)}[\log p(\ell|\phi)] - \beta \mathsf{KL}[q(\tilde{\phi}) \| p(\tilde{\phi})].$$

3) Active learning: For inference at a set of test points $\{\theta_*\}$, we transform those into $\{\phi_*\}$, and the approximate posterior is then given by

$$p(\phi_*|\ell) = \int p(\phi_*|\tilde{\phi}) q(\tilde{\phi})\,\mathrm{d}\tilde{\phi},$$

which can be computed in a manner similar to (9). Initially, when the failure region is not estimated well, it is necessary to select informative elements of $\{\theta_*\}$ that can yield good estimates of Θ_{F} without exhaustive sampling. To this end, we propose an active learning strategy wherein the most informative $\theta_j^\star \in \theta_*$ is selected based on the maximum entropy of the posterior distribution. Since the mean and variance of the posterior $p(\phi_*|\ell)$ is computed for each parameter in θ_*, one can compute the entropy (assuming q is Gaussian) and fix

$$\theta_j^\star = \arg\max_{\theta' \in \theta_*} \frac{1}{2} \log\left(2\pi \mathsf{Var}(\theta')\right). \tag{10}$$

We then evaluate $\theta_j^\star$ by simulating $\mathcal{M}_T(\theta_j^\star)$ to ascertain ℓ_{j+1} and J_{j+1}, which yields the updated dataset $\mathcal{D}_{j+1}$ and the process iterates till a termination criterion such as a maximum number of iterations is achieved.

While any probabilistic classifier could be used in place of VGPC, there are some caveats to classifier selection that motivate our selection of VGPC as an exemplar. First, the boundary of the failure region is not always regular in geometry. Therefore, a nonlinear classifier capable of generating complex geometries with few data points is necessary. Second, the classifier should be re-trainable and small increments to the dataset should be reflected in the classifier's decision boundary. This requirement of frequent retraining is a major reason why we do not use deep neural networks. Empirically, we have found kernel-based nonparameteric classifiers such as SVMs and GPCs perform the best in these scenarios, and GPCs offer a probabilistic output without further modification (unlike SVMs where one has to perform additional operations to obtain probabilistic outputs). Finally, the entropy function used for active learning has a simple form for the VGPC that leverages the variance, which is a component of the VGPC output. We reiterate that other probabilistic nonlinear classifiers such as probabilistic versions of SVMs and deep neural networks could be used in place of VGPC.

B. Incorporating failure probabilities

Recall the notation presented in Section II-B for classical BO. Note that for construction of the GP regressor that approximates the calibration cost, as described in (4), the cost values corresponding to failed simulations are ignored, and only the calibration cost incurred during successful simulations are utilized. Once this GP regressor is constructed, we can compute an acquisition function value such as the expected improvement and combine that with the outputs of the VGPC to ensure robustness to simulation failures.

Since the VGPC generates a probabilistic output, we can directly incorporate it into the acquisition function, as proposed in [30]. This yields the failure-robust EI acquisition function

$$\mathsf{FREI}(\theta, j) = \mathsf{EI}(\theta, j) \cdot (1 - \mathsf{PF}(\theta, j)), \tag{11}$$

where $\mathsf{PF}(\theta, j)$ is the likelihood of failure calculated by training the VGPC using data up to the j-th iteration, and then evaluating the likelihood of the VGPC at θ, and EI is described in (5).

If the VGPC algorithm does not find any θ such that $\mathsf{PF}(\theta, j) > 0$, then the acquisition function (11) is zero for every $\theta \in \Theta$ and future candidates are selected randomly until at least one θ is found which allows for a successful simulation. This scenario is rarely seen in practice. A more plausible scenario is that both successful simulations and failure simulations have been observed, and the VGPC has been trained on a non-trivial classification problem. In such a case, the higher the value of $\mathsf{P}(\theta, j)$, the higher the probability that a particular candidate will be selected, so long as its expected improvement is high as well. The multiplicative nature of the components in (11) seeks to ensure that neither one component can outweigh the other, and candidates will be selected only if they are both a candidate for optimization and feasibility (i.e., is expected to yield a successful simulation). Along the same lines as (5), FR-BO selects the next optimizer candidate as follows:

$$\theta_{j+1} = \arg\max_{\theta \in \Theta} \mathsf{FREI}(\theta, j). \tag{12}$$

A key difference with FREI and the constrained expected-improvement acquisition function proposed in [30] is that the PF component is a probability induced by a probabilistic classifier trained on binary labels ℓ, whereas constrained BO estimates probabilities based on continuous slack variables obtained from the constraints. Both methods are similar in that they use Gaussian process proxies for constraint-modeling, although in our case it is a classifier rather than the regressor proposed in [30].

It is important to note that from an implementation perspective, it may be expensive to retrain a VGPC after the collection of each new data sample. Empirically, we have observed that this is not always necessary: in fact, as long as the VGPC has been trained initially with some data, it can be retrained infrequently. Of course, how frequently the retraining has to occur is problem dependent, although heuristics such as retraining the VGPC when the FR-BO is 'stuck' at a local optimum for a pre-decided number of iterations can be useful.

C. Intuition via illustrative example

Rather than solely rely upon the previous exposition of this general method, we have found that an illustrative 2-dimensional example can provide some intuition as to why we expect the FR-BO algorithm will converge to a feasible optimal solution. The underlying cost function is shown in Figure III-B using grayscale, and while there are four global optimizers for this function on the defined search space, only one of these at the location of the the five-pointed star coordinate is feasible. The true failure region is shown using red shading. The FR-BO algorithm operates in 3 main stages: (i) the early stage in which there are few samples on Θ, resulting in a poor estimate of the failure region; (ii) the active learning stage where the FR-BO selects feasible and infeasible samples that result in improvement of the feasible region estimates; and, (iii) the late stage after the failure region is estimated to a satisfactory degree of accuracy, wherein the FR-BO algorithm behaves similar to constrained BO due to our proposed FREI acquisition function. We show exemplar iterations from all three stages in Fig. III-B(a–c). In all subplots, the parameters that resulted in successful simulations are shown with green circles and the parameters that failed are shown using red crosses. A representative contour corresponding to a given probability of the probabilistic classifier is shown using a dashed yellow line, and a yellow four-pointed star denotes the optimal value found up to the current iteration using the FR-BO algorithm.

In Fig. III-B(a), we indicate that in early iterations, FR-BO typically has few data samples on Θ. The classifier consequently generates a poor estimate of the failure region. Since only a few feasible (green circle) samples are available to construct the parameter-to-cost regressor, the approximation of the calibration cost is poor, and the best parameter candidate based on this limited set of feasible data is far removed from the true optimum. The FREI acquisition function, based on this inaccurate failure classifier, will lead to the selection of parameter candidates that are unlikely to improve the estimate of the failure region in the early iterations. We thus demonstrate the effect of active learning in subplot (b) to this end. We observe that the actively learned samples are concentrated around the boundary of the failure region estimate, and thus the collection of labels (by simulation) for those parameters greatly improves our understanding of the feasible region. However, the increased amount of feasible samples do not necessarily imply that a better approximation of the underlying cost has been obtained. The best parameter candidate found by active learning may thus remain far from the true optimizer. Once the failure region is estimated to a satisfactory accuracy (which happens naturally by selecting actively learned samples), the FREI acquisition function used in FR-BO quickly searches over promising areas in the parameter space while avoiding the failure region to obtain feasible and near-optimal samples. The behavior of the FREI acquisition function is similar to the constrained EI acquisition function proposed in [30] in late-stage iterations, and is therefore expected to converge to a feasible optimizer. A key difference between the approaches is that our 'constraint' function is actually not continuous-valued like those considered in [30], because the FR is akin to a set induced by an indicator function. The probabilistic failure region classifier provides a continuous approximation of this indicator function, thereby allowing the use of constrained-BO-like acquisition functions.

IV. Results and Discussion

We provide two examples that demonstrate the effectiveness of FR-BO. The first involves parameter estimation for a well-studied stiff nonlinear system of chemical kinetics, and the second is a real-world building model calibration problem. All code was implemented in `GPyTorch` [36], `PyTorch` [37], and `Python 3.9`.

Example 1: Comparative study on Lotka-Volterra dynamics

To compare with existing parameter estimation algorithms, we use the well-studied Lotka-Volterra system

$$\dot{x}_1 = \theta_1 x_1 - \theta_2 x_1 x_2, \quad \dot{x}_2 = -\theta_3 x_2 + \theta_4 x_1 x_2,$$

most recently explored in [25] for data-driven parameter estimation. The measurements from the system are obtained in the time span $T = [0, 2]$ at 20 evenly spaced observation times. The true parameters of the system are given by $\theta^\star = [2, 1, 4, 1]$, and the system has initial state $x_0 = [5, 3]^\top$ for simulation. The measurements are corrupted by zero-mean 0.25-variance Gaussian noise, as in the more difficult test scenario studied in [25]. We assume that the range of admissible parameters $\Theta = [0.5, 3] \times [-0.5, 1.5] \times [2.5, 5.0] \times [-0.5, 1.5]$, which is a non-trivial uncertainty range on the parameters.

TABLE II
FR-BO implementation for Example 1.

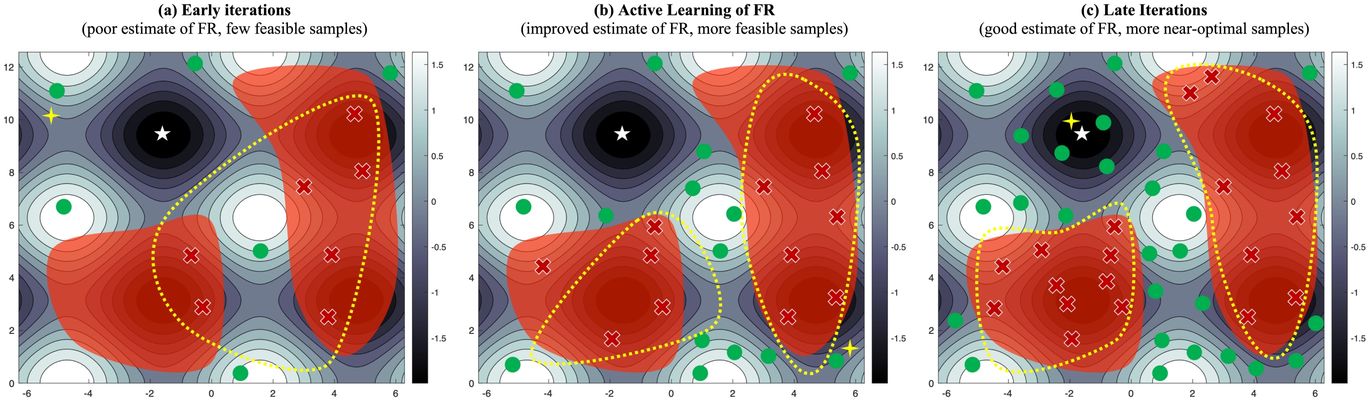
	VGPC	GP
Model	Approximate GP	Exact GP
Kernel	Squared Exponential	Matern-3/2
Inducing Points	Yes	Yes
Likelihood	Bernoulli	Gaussian
Loss Function	Variational ELBO	MLL
Optimizer	Adam	Adam
Learning Rate	0.01	0.05
Training Iters	2000	500

For each θ, we can then compute $\mathcal{M}_T(\theta)$ and a parameter estimation cost function

$$J(y^\star_{0:T}, \mathcal{M}_T(\theta)) = \log\left(\sum_{t=0}^{T}(y^\star_t - y_t)^\top W(y^\star_t - y_t)\right), \quad (13)$$

where y_t is the output vector obtained from $\mathcal{M}_T(\theta)$ at the t-th time instant. The matrix W is a scaling matrix to ensure that the three output components are of similar magnitudes. For this particular example, W is the identity matrix. Table II describes the hyperparameters used for both the VGPC and the GPR learners needed for FR-BO. For FR-BO, we run 100 random initial iterations, 300 active learning iterations to learn the classifier, and 400 iterations for FR-BO exploitation. Note that we need a total of 800 iterations to obtain a good set of parameters, whereas the algorithms proposed in [25, Supplementary §7.5] report requiring over 100,000 iterations for some algorithms and over 33,000 iterations for their proposed faster variant. At termination, the FR-BO algorithm yields the parameters $\hat{\theta}_1 = 1.865$, $\hat{\theta}_2 = 0.925$, $\hat{\theta}_3 = 4.445$, and $\hat{\theta}_4 = 1.118$.

Over 800 iterations, we note from Fig. 3 that not only does the FR-BO provide a better solution than classical BO or latin-



(a) Early iterations
(poor estimate of FR, few feasible samples)

(b) Active Learning of FR
(improved estimate of FR, more feasible samples)

(c) Late Iterations
(good estimate of FR, more near-optimal samples)

Fig. 2. Explanation of stages of FR-BO. The white ★ represents the global optimum and the yellow ✦ represents the optimal value estimated by the parameter-to-cost regressor. The contours of an exemplar cost function is shown using grayscale, and the true failure region (FR) with red shading. The 0.5-probability of failure contour estimated by an exemplar probabilistic classifier is shown using dashed yellow lines.

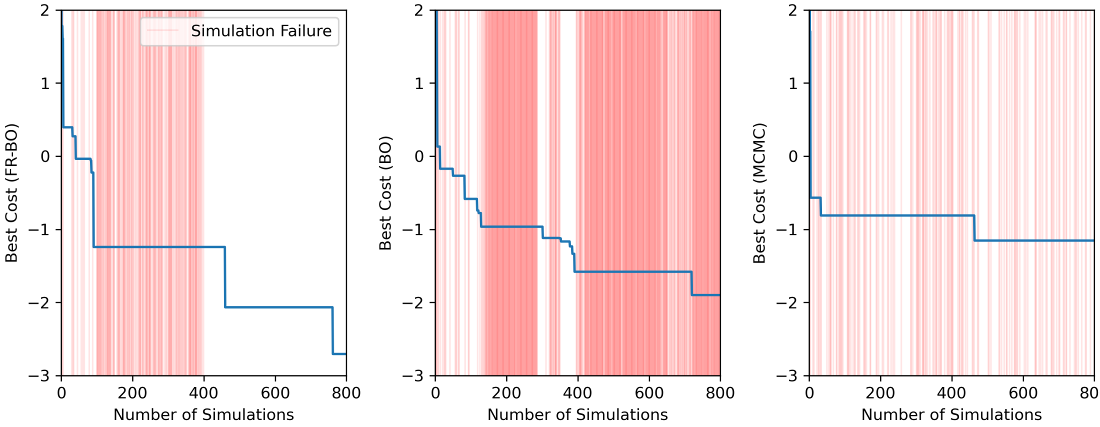


Fig. 3. Comparison of cost convergence for FR-BO, classical BO, and MCMC sampling on Example 1.

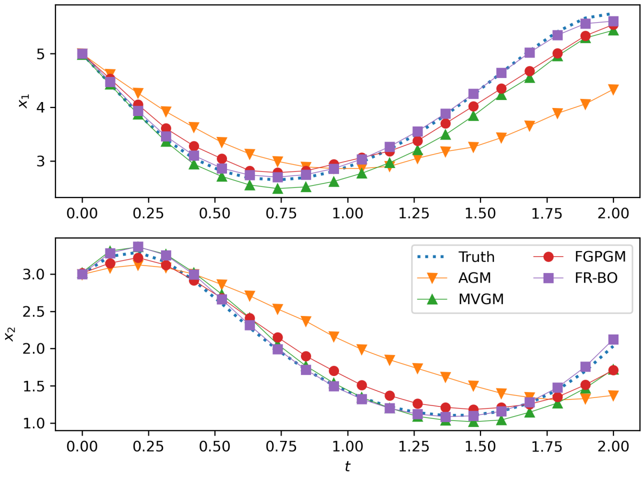


Fig. 4. Comparison of trajectories generated by simulations using parameter estimation with FR-BO against other fast parameter estimation approaches.

hypercube sampling (LHS), the number of simulation failures with FR-BO after the initial 400 iterations (random and GPC training) reduce to zero; i.e., there are no wasted simulations during BO exploitation. This is not the case for classical BO, which has a large number of failed simulations throughout the parameter estimation procedure because the acquisition function repeatedly drives the sampling to a failure region. Sampling using space-filling LHS methods ensures that the same regions is not sampled repeatedly and results in more uniform simulation failure events during the search, but does not effectively exploit learned information, resulting in a poor final cost compared to the BO methods. Therefore, the FR-BO shows clear benefits in parameter estimation quality.

Fig. 4 illustrates the effectiveness of the proposed approach compared against fast GP gradient matching (FGPGM) method proposed in [25] and the other parameter estimation algorithms considered in that paper, such as the adaptive gradient matching (AGM) method [23], and the maximum likelihood variational gradient matching (MVGM) algorithm [38]. These are all variants of gradient matching algorithms, which are designed to estimate the state update map of an unknown dynamical system, typically approximated using GP models. These GP models have hyperparameters that need to be tuned for this learning task, and each variant listed above performs this hyperparameter search differently. The AGM method uses MCMC sampling to find optimal hyperparameters, which is usually slow and requires inversion or determinant computation of large matrices. The MVGM algorithm improves upon this by adopting a mean field variational inference approach and maximizing a negative log likelihood; this improves computational efficacy and is reported to find better solutions than its predecessors. The FPGPM algorithm constructs a different probabilistic graphical model involved in gradient matching, and with the use of new auxiliary random variable, greatly improves the convergence speed, while providing theoretical justifications to their proposed graphical model. Fig. 4 shows that the FR-BO algorithm provides better estimates of the predicted states compared to its competitors with the same dataset provided to each method and fewer iterations than its competitors. The improved quality of the final parameter estimate is corroborated by the RMSE values computed over both states, which is 0.053 for FR-BO; significantly less than 0.654 for AGM, 0.210 for MVGM and 0.117 for FGPGM.

Example 2: Calibration of a real-world building digital twin

Simulation models of building energy behavior for digital twin applications are generally designed to predict multi-scale nonlinear dynamics that result from the temporal interactions between the many different building subsystems. These subsystems can include the building envelope, space-conditioning systems such as heat pumps or ventilation systems, lighting systems, and building occupants, among others. These different subsystems often act over a wide range of interacting timescales; whereas the building thermal dynamics and ambient weather may vary over minutes to months, the behavior of vapor-compression systems and airflow will vary over much smaller timescales of milliseconds to hours. These coupled timescales present significant challenges in constructing simulations of the overall system due to the accompanying numerical stiffness. Furthermore, large portions

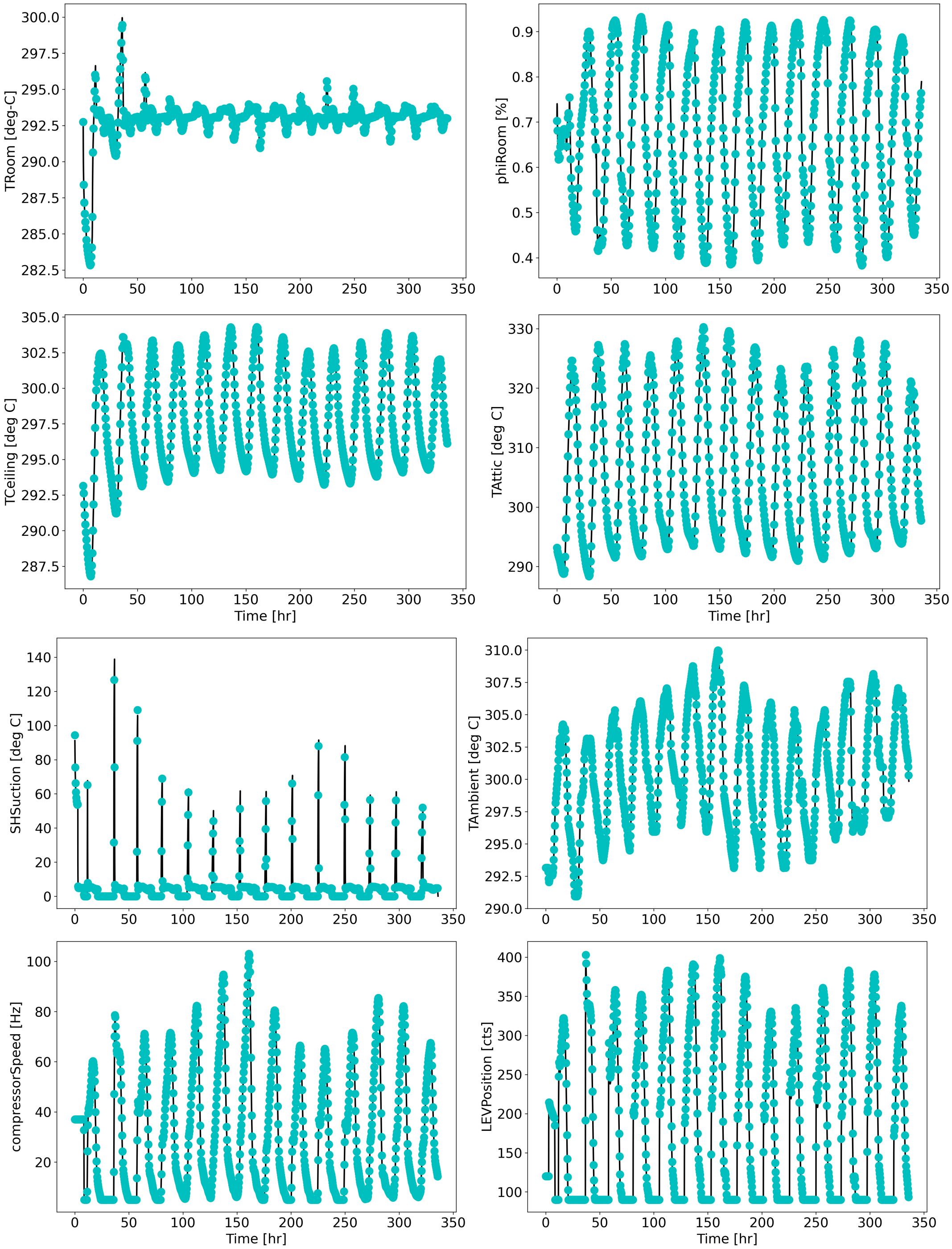


Fig. 5. Integrated building system: Estimated and true outputs of the HVAC equipment and building envelope. Green circles are true noisy data points and continuous lines are model estimates using the FR-BO optimized model parameters.

of these models are often embedded in software binaries or computed using input-output maps, so that no gradients with respect to parameters can be computed. The simplest representation for the coupled building dynamics in many circumstances is thus the proposed model (1).

We evaluate the performance of these calibration methods on a dynamical model, built in the `Modelica` language, of a building envelope coupled to a heat pump. The heat pump model consists of a vapor compression system that uses dynamic models of the heat exchangers and algebraic (static) models of the compressor and expansion valve. The structure of this cycle models is described in more detail in [39]. The building envelope is based on a one-story residence with nominal 2009 IECC-based construction, and is based on the model used in [27]. The convective, radiative, and latent heat flows resulting from the influence of occupants and weather variations on the building envelope are described by the Modelica Buildings library [40]. This model was exported from Modelica using the Functional Mockup Interface[1], and the resulting functional mockup unit (FMU) was imported into Python using the FMPy package[2] to enable integration of the model with `PyTorch` modules. The inputs and outputs of this model were chosen to be similar to those which may be observed in a realistic experimental setting. The inputs of the HVAC system include the room temperature set-point, the evaporator superheat set-point, and the fan speeds. The inputs for the building envelope model include the convective, radiative, and latent heat loads as well as the weather variables provided in the `TMY3` standard weather file for the Hartsfield-Jackson airport in Atlanta, GA, USA.

We collect ground-truth data for calibration by simulating the Modelica model from July 1-14 with 17 parameters of the model set to their true values. These parameters were chosen to be representative of the HVAC system and the building and are provided in Table III. The search space of the parameters are $\pm 15\%$ of the true parameters with $\pm 5\%$ random translation of the upper and lower bounds for each parameter. The eight measured output sequences $y^\star_{0:T}$ of the model are collected at 5 minute intervals, and the FR-BO components have the same hyperparameters as in Table II, with two differences: the training iterations are kept at 2000 for the GP, and the kernels are Matern-5/2 for the VGPC.

TABLE III
DESCRIPTION OF PARAMETERS FOR INTEGRATED BUILDING. (HTC = HEAT TRANSFER COEFFICIENT, HEX = HEAT EXCHANGER)

PARAMETER	TRUE	FR-BO
Building Parameters		
Airflow infiltration rate	3.368×10^{-2}	3.319×10^{-2}
Thickness of the floor	1.016×10^{-1}	9.593×10^{-2}
Infrared emissivity of roof (outer)	9.000×10^{-1}	8.646×10^{-1}
Solar emissivity of roof (outer)	9.000×10^{-1}	8.435×10^{-1}
Infrared emissivity of roof (inner)	7.000×10^{-1}	6.456×10^{-1}
Solar emissivity of roof (inner)	7.000×10^{-1}	7.592×10^{-1}
Interior room air HTC	3.000	3.093
Exterior air HTC	1.000×10^{1}	0.990×10^{1}
HVAC Parameters		
Outdoor HEX HTC adjustment factor	1.000	1.032
Indoor HEX HTC adjustment factor	1.000	0.904
Indoor HEX Lewis number	8.540×10^{-1}	8.650×10^{-1}
Outdoor HEX vapor HTC	5.000×10^{2}	5.152×10^{2}
Outdoor HEX 2-phase HTC	3.000×10^{3}	3.215×10^{3}
Outdoor HEX liquid HTC	7.000×10^{2}	7.122×10^{2}
Indoor HEX vapor HTC	5.000×10^{2}	5.240×10^{2}
Indoor HEX 2-phase HTC	2.000×10^{3}	1.995×10^{3}
Indoor HEX liquid	7.000×10^{2}	7.258×10^{2}

A comparison of the measured and estimated simulation outputs of the building and HVAC is shown in Fig. 5. Note that the simulation outputs are generated by using the optimal parameters found by our FR-BO algorithm. The green circles are true data points and the continuous lines are the estimated outputs from model simulation. It is clear from the figure that the output error is small, and this is further evident from Table III, where we see that most parameters have been estimated with high accuracy.

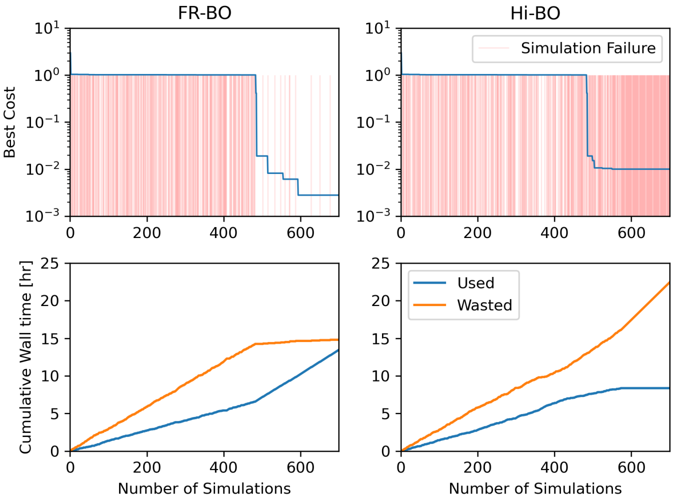


Fig. 6. Integrated building system: Regret and number of simulation failures.

It is also interesting to note the effectiveness of FR-BO in reducing the amount of time wasted on failed simulations during optimization. We compare FR-BO with Hi-BO, and allow the VGPC to first collect data via active learning for 450 iterations before it begins the optimization process. As seen from the top row of Fig. 6, the incorporation of the failure-robust EI acquisition function drastically reduces the number of simulation failures. This improves the convergence of the Bayesian optimization step, so that the FR-BO converges to its best solution within approximately 700 iterations, which results in a cost that is less than half of Hi-BO's final cost. From the lower row of subplots, we see that the total time that is used for Bayesian optimization is higher for FR-BO (15 hr vs. 9 hr) and the total time wasted simulating failures is significantly less (15 hr vs. 23 hr), which equates to saving 8 hours of wall time.

V. CONCLUSIONS

Simulation software is usually designed and validated over a limited region of the available parameter space. Thus, regions of validity for model parameters, or regions over which the simulation will fail, are rarely known during downstream design. Calibrating the model for different datasets requires exploring the parameter space and could involve trying parameters that will result in simulation failures. In this paper, we provided a methodology for model calibration using failure-robust Bayesian optimization that involves learning the failure region and embedding that information into the exploitation step via a failure-robust acquisition function, and demonstrated the efficacy of this method on both a simple example problem and a large-scale simulation model of the coupled behavior of a building and its accompanying HVAC system. This method thus has good potential for application to parameter estimation and calibration problems that contain numerically stiff, nonlinear dynamical sets of differential equations.

Specific advantages of the proposed FR-BO algorithm compared against the state-of-the-art calibration methodologies include: (1) FR-BO takes advantage of the sample efficiency inherent to Bayesian optimization methods and therefore requires fewer simulations than, for example, MCMC calibration methods, to converge to a good set of parameters; (2) FR-BO

[1] Modelica, Functional Mockup Interface for Model Exchange and Co-Simulation, Version 2.0.1.

[2] `https://github.com/CATIA-Systems/FMPy`

explicitly provides an estimate of regions in the parameter space where the model is likely to waste time computing forward simulations or fail altogether: the reduction of this time wastage and computational resource expenditure can improve workflows by enable faster calibration; and, (3) the method is model-agnostic and therefore can be applied to any simulation model that has the requisite input-output structure and known parameter search space bounds. Of course, the method requires some more iterations than classical BO due to additional iterations required for the failure classifier learning problem. Even though we have proposed an active learning framework to reduce the number of iterations required to that end, in future work, we plan to investigate a data-efficient architecture and loss function that trades-off failure region classification and model optimization.

References

[1] J. Lu, D. Chen, G. Wang, D. Kiritsis, and M. Törngren, "Model-based systems engineering tool-chain for automated parameter value selection," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–15, 2021.

[2] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21 980–22 012, 2020.

[3] J. Na, Y. Xing, and R. Costa-Castelló, "Adaptive estimation of time-varying parameters with application to roto-magnet plant," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 2, pp. 731–741, 2021.

[4] A. Chakrabarty, E. Maddalena, H. Qiao, and C. Laughman, "Scalable Bayesian optimization for model calibration: Case study on coupled building and HVAC dynamics," *Energy and Buildings*, vol. 253, p. 111460, 2021.

[5] A. Chong, K. P. Lam, M. Pozzi, and J. Yang, "Bayesian calibration of building energy models with large datasets," *Energy and Buildings*, vol. 154, pp. 343–355, 2017.

[6] A. Chakrabarty, G. T. Buzzard, and A. E. Rundell, "Model-based design of experiments for cellular processes," *WIREs Systems Biology and Medicine*, vol. 5, no. 2, pp. 181–203, 2013. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wsbm.1204

[7] W. Yang, Y. Zheng, and S. Li, "Application status and prospect of digital twin for on-orbit spacecraft," *IEEE Access*, vol. 9, pp. 106 489–106 500, 2021.

[8] Y. Peng, X. Zhang, Y. Song, and D. Liu, "A low cost flexible digital twin platform for spacecraft lithium-ion battery pack degradation assessment," in *2019 IEEE International Instrumentation and measurement technology conference (I2MTC)*. IEEE, 2019, pp. 1–6.

[9] A. Pal, L. Zhu, Y. Wang, and G. G. Zhu, "Constrained surrogate-based engine calibration using lower confidence bound," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 6, pp. 3116–3127, 2021.

[10] Y. Liang, K.-S. Tam, and R. Broadwater, "Load calibration and model validation methodologies for power distribution systems," *IEEE Transactions on Power Systems*, vol. 25, no. 3, pp. 1393–1401, 2010.

[11] M. van Tiel, K. Stahl, D. Freudiger, and J. Seibert, "Glacio-hydrological model calibration and evaluation," *Wiley Interdisciplinary Reviews: Water*, vol. 7, no. 6, p. e1483, 2020.

[12] J. Ma, B. Huang, and F. Ding, "Iterative identification of Hammerstein parameter varying systems with parameter uncertainties based on the variational Bayesian approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 3, pp. 1035–1045, 2020.

[13] Y. Zhang, M. Wang, X. Fang, and U. Ozguner, "Unifying analytical methods with numerical methods for traffic system modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 6, pp. 2068–2082, 2020.

[14] J. O. Ramsay, G. Hooker, D. Campbell, and J. Cao, "Parameter estimation for differential equations: a generalized smoothing approach," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 5, pp. 741–796, 2007.

[15] R. B. Gopaluni, "Nonlinear system identification under missing observations: The case of unknown model structure," *Journal of Process Control*, vol. 20, no. 3, pp. 314–324, 2010.

[16] W. J. Stortelder, "Parameter estimation in dynamic systems," *Mathematics and Computers in Simulation*, vol. 42, no. 2-3, pp. 135–142, 1996.

[17] L. Biegler, J. Damiano, and G. Blau, "Nonlinear parameter estimation: a case study comparison," *AIChE Journal*, vol. 32, no. 1, pp. 29–45, 1986.

[18] W. R. Esposito and C. A. Floudas, "Global optimization for the parameter estimation of differential-algebraic systems," *Industrial & Engineering Chemistry Research*, vol. 39, no. 5, pp. 1291–1310, 2000.

[19] S. Rouchier, M. Rabouille, and P. Oberlé, "Calibration of simplified building energy models for parameter estimation and forecasting: Stochastic versus deterministic modelling," *Building and Environment*, vol. 134, pp. 181–190, 2018.

[20] M. A. Beaumont, "Approximate Bayesian computation," *Annual Review of Statistics and Its Application*, vol. 6, pp. 379–403, 2019.

[21] M. Niu, S. Rogers, M. Filippone, and D. Husmeier, "Fast parameter inference in nonlinear dynamical systems using iterative gradient matching," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1699–1707.

[22] J. González, I. Vujačić, and E. Wit, "Reproducing kernel hilbert space based estimation of systems of ordinary differential equations," *Pattern Recognition Letters*, vol. 45, pp. 26–32, 2014.

[23] F. Dondelinger, D. Husmeier, S. Rogers, and M. Filippone, "ODE parameter inference using adaptive gradient matching with Gaussian processes," in *Artificial intelligence and statistics*. PMLR, 2013, pp. 216–228.

[24] D. Barber and Y. Wang, "Gaussian processes for bayesian estimation in ordinary differential equations," in *International conference on machine learning*. PMLR, 2014, pp. 1485–1493.

[25] P. Wenk, A. Gotovos, S. Bauer, N. S. Gorbach, A. Krause, and J. M. Buhmann, "Fast Gaussian process based gradient matching for parameter identification in systems of nonlinear ODEs," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1351–1360.

[26] E. Hairer and G. Wanner, "Stiff differential equations solved by Radau methods," *Journal of Computational and Applied Mathematics*, vol. 111, no. 1-2, pp. 93–111, 1999.

[27] C. Laughman *et al.*, "Modeling and control of radiant, convective, and ventilation systems for multizone residences," in *Proc. of Building Simulation 2019*, 2019, pp. 1956–1963.

[28] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. NeurIPS*, 2012, pp. 2951–2959.

[29] A. Chakrabarty and M. Benosman, "Safe learning-based observers for unknown nonlinear systems using Bayesian optimization," *Automatica*, vol. 133, p. 109860, 2021.

[30] J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian optimization with inequality constraints." in *Proc. ICML*, vol. 2014, 2014, pp. 937–945.

[31] C. K. Williams and C. E. Rasmussen, *Gaussian Processes For Machine Learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[32] A. Chakrabarty, C. Danielson, S. Di Cairano, and A. Raghunathan, "Active learning for estimating reachable sets for systems with unknown dynamics," *IEEE Transactions on Cybernetics*, 2020.

[33] J. Hensman, A. G. Matthews, and Z. Ghahramani, "Scalable variational Gaussian process classification," *J. Mach. Learn. Res.*, vol. 38, pp. 351–360, 2015.

[34] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," *Proc. NeurIPS*, vol. 18, pp. 1259–1266, 2006.

[35] I. Higgins *et al.*, "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2017.

[36] J. R. Gardner *et al.*, "GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration," in *Proc. NeurIPS*, 2018, pp. 7587—-7597.

[37] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Proc. NeurIPS*, 2019, pp. 8024–8035.

[38] N. S. Gorbach, S. Bauer, and J. M. Buhmann, "Scalable variational inference for dynamical systems," in *Advances in Neural Information Processing Systems*, 2017.

[39] H. Qiao, C. R. Laughman, D. Burns, and S. Bortoff, "Dynamic characteristics of an R410A multi-split variable refrigerant flow air-conditioning system," in *Proc. 12th IEA Heat Pump Conference*, 2017.

[40] M. Wetter *et al.*, "Modelica buildings library," *Journal of Building Performance Simulation*, vol. 7, no. 4, pp. 253–270, 2014.

Ankush Chakrabarty received the Ph.D. degree as a Ross Fellow from Purdue University, West Lafayette, IN, in Electrical and Computer Engineering (2016). From 2016–2018, he was a Postdoctoral Fellow at Harvard University, Cambridge, MA, where he worked on the conceptualization and development of an embedded artificial pancreas system. Since 2018, he is at Mitsubishi Electric Research Laboratories (MERL), where he is currently a Principal Research Scientist. His research lies in the intersection of machine learning and control engineering, focusing on Bayesian optimization, meta-learning, and state/parameter estimation using simulation environments and digital twins of building energy systems. He serves as the Outreach Coordinator at the IEEE CSS Electronic Information Committee, and as associate editor for CSS and SMC conferences. He is a Senior Member of the IEEE and has an Erdős number of 4.

Scott A. Bortoff received the B.S. and M.S. degrees from Syracuse University in 1985 and 1986, respectively, and the Ph.D. degree from the University of Illinois at Urbana-Champaign in 1992, all in Electrical Engineering. He held positions of Assistant and Associate Professor of Electrical and Computer Engineering at the University of Toronto, (1992-2000), Group and Project Leaderships in the area of control systems at United Technologies Research Center (2000-2009), and Group Manager of Mechatronics at Mitsubishi Electric Research Laboratories, Cambridge, MA, USA (2009-2016), where he is currently Chief Scientist and Distinguished Research Scientist. His research interests include system-level dynamic modeling and control of thermofluid and mechatronic systems. He is currently Associate Editor of the IEEE Control System Magazine and serves the Modelica community in various capacities.

Christopher R. Laughman received the S.B. (1999) and M.Eng. (2001) degrees in Electrical Engineering and Computer Science, as well as the Ph.D. (2008) degree in Architecture, from the Massachusetts Institute of Technology. He has been with Mitsubishi Electric Research Laboratories since 2008, where he currently holds the position of Senior Principal Research Scientist and is the Senior Team Leader of the Multiphysical Systems team. His research interests include the modeling, simulation, control, and optimization of large-scale multiphysical systems, with an emphasis on multiphase thermofluid applications. He is also a member of ASHRAE and is a member of the board of the North American Modelica Users' Group.