Uncertainty Quantification in Stochastic Economic Dispatch using Gaussian Process Emulation

Zhixiong Hu[1], Yijun Xu[2], Mert Korkali[3], Xiao Chen[4], Lamine Mili[2], and Charles H. Tong[4]

[1]Department of Statistics, University of California-Santa Cruz, Santa Cruz, CA 95064 USA, e-mail: zhu95@ucsc.edu
[2]Department of Electrical and Computer Engineering, Virginia Tech, Northern Virginia Center, Falls Church, VA 22043 USA, e-mail: {yijunxu,lmili}@vt.edu
[3]Computational Engineering Division, Lawrence Livermore National Laboratory, Livermore, CA 94550 USA, e-mail: korkali1@llnl.gov
[4]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94550 USA, e-mail: {chen73,tong10}@llnl.gov

***Abstract*—The increasing penetration of renewable energy resources in power systems, represented as random processes, converts the traditional deterministic economic dispatch problem into a stochastic one. To solve this stochastic economic dispatch, the conventional Monte Carlo method is prohibitively time consuming for medium- and large-scale power systems. To overcome this problem, we propose in this paper a novel Gaussian-process-emulator-based approach to quantify the uncertainty in the stochastic economic dispatch considering wind power penetration. Based on the dimension-reduction results obtained by the Karhunen-Loève expansion, a Gaussian-process emulator is constructed. This surrogate allows us to evaluate the economic dispatch solver at sampled values with a negligible computational cost while maintaining a desirable accuracy. Simulation results conducted on the IEEE 118-bus system reveal that the proposed method has an excellent performance as compared to the traditional Monte Carlo method.**

I. Introduction

Power systems are inherently stochastic. Sources of stochasticity include time-varying loads, renewable energy intermittencies, and random outages of generating units, lines, and transformers, to cite a few. These stochasticities translate into uncertainties in the power system models. To address this problem, research activities have focused on uncertainty quantification in power system planning, monitoring, and control [1]–[6]. Among them, the topic of stochastic economic dispatch (SED) has recently attracted considerable academic attention due to the increasing penetration of renewable energy resources.

To account for these uncertainties, some researchers propose to adopt a scenario-based optimization approach. However, this approach only considers a finite set of sampling realizations, which is obviously an oversimplification of the numerous cases that may occur in reality [7], [8]. By contrast, other researchers propose to make use of uncertainty quantification techniques via Monte Carlo sampling. However, all the traditional Monte Carlo methods are prohibitively time consuming when accurate estimation of uncertain model outputs are needed. This problem calls for the development of new computationally efficient and accurate uncertainty modeling techniques for power system applications [1], [9].

In this paper, we develop a new SED method based on a Gaussian process emulator (GPE) for power systems to which are connected wind power generation. The GPE allows us to evaluate, with a negligible computational cost, the SED solver at sampled values through a nonparametric reduced-order representation [10]. To further improve the computational efficiency in the construction of the surrogate models, a model reduction is achieved via the application of the Karhunen-Loève expansion (KLE) to real-world data, collected from real-world wind farms [11]. The simulation results conducted on a modified IEEE 118-bus system reveal that the proposed method can greatly improve the computational efficiency of the SED as compared to the traditional Monte Carlo method while maintaining a desirable estimation accuracy.

II. Problem Formulation

Traditionally, under some physical and economic constraints, the economic dispatch in power systems is known as a deterministic optimization problem. This problem aims to identifying an optimal set of power outputs of a fixed set of online thermal generating units that yields a minimum cost, denoted by $Q(\mathbf{g})$. The cost Q is generally thought to be nonrandom since the traditional thermal generating units $\mathbf{u}$ can be optimized and set equal to some deterministic optimal values.

However, in the face of the increasing penetration of renewable energy resources, the abovementioned statement cannot hold true. Due to the intrinsic randomness of the renewable generation, represented (using random fields) as functions of a vector of random variables, $\boldsymbol{\omega}$, denoted by $\mathbf{p}(\boldsymbol{\omega})$, the deterministic economic problem for finding $Q(\mathbf{g}) = \arg\min_{\mathbf{g}}\{f(\mathbf{g})\}$ is extended to an SED problem described by

$$Q(\mathbf{g}, \mathbf{p}(\boldsymbol{\omega})) = \arg\min_{\mathbf{g}}\{f(\mathbf{g}, \mathbf{p}(\boldsymbol{\omega}))\}. \tag{1}$$

Here, f represents the objective function. For this problem, the randomness brought by $\mathbf{p}(\boldsymbol{\omega})$ will lead to different optimized values of $\mathbf{g}$, which will inevitably change the deterministic cost, $Q(\mathbf{g})$, into a random cost, $Q(\mathbf{g}, \mathbf{p}(\boldsymbol{\omega}))$. In this paper, we consider the randomness brought by the wind farms as a spatiotemporal random field, which is denoted by $\mathbf{p}_i(\boldsymbol{\omega}, \mathbf{t})$ for the power generation of the ith wind farm. Here, the time $t \in \mathbb{T}$ and $\mathbb{T}$ is a finite integer set representing hours in a day, namely, $\mathbb{T} = \{1, 2, \ldots, 24\}$. Let us take an example. Suppose that we conduct a day-ahead SED problem over 24 hours of a power system with three farms. Then, we have an input of three random fields, $\{\mathbf{p}_1(\boldsymbol{\omega}, \mathbf{t})$, $\{\mathbf{p}_2(\boldsymbol{\omega}, \mathbf{t})$, $\{\mathbf{p}_3(\boldsymbol{\omega}, \mathbf{t})\}$, consisting of 72 random variables in total. Our uncertainty quantification goal is to quantify the statistical moments of $Q(\mathbf{g}, \mathbf{p}(\boldsymbol{\omega}))$, such as the mean and variance for a day-ahead forecast.

Remark. *Note that since we focus on the quantification of uncertainties in the SED problem instead of their modeling, the detailed description of "f" as well as all the equality and inequality constraints of the SED topic directly follow from [1].*

978-1-7281-3103-0/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: to IEEExplore provided by University Libraries | Virginia Tech. Downloaded on May 30,2020 at 06:43:21 UTC from IEEE Xplore. Restrictions apply.

III. Theoretical Background

In this section, we briefly present the theory of the GPE and of the KLE to assist us in the SED problem.

A. Gaussian Process Emulator

1) Basic Theory: The GPE is known to be a powerful Bayesian-learning-based method based on a nonlinear regression problem [10], [12]. To describe this method, let us first denote the SED model by $f(\cdot)$ and its corresponding vector-valued input of p dimensions by $\mathbf{x}$. Due to the randomness of $\mathbf{x}$, we may observe n samples as a finite collection of the model input as $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. Accordingly, its model output $f(\mathbf{x})$ also becomes random and has its corresponding n realizations, denoted by $\{f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)\}$.

If we assume that the model output is a realization of a Gaussian process, then the finite collection, $\{f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)\}$, of the random variables, $f(\mathbf{x})$, will follow a joint multivariate normal probability distribution, that is, we have

$$\begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_n) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & \cdots & k(\mathbf{x}_1,\mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n,\mathbf{x}_1) & \cdots & k(\mathbf{x}_n,\mathbf{x}_n) \end{bmatrix} \right). \tag{2}$$

Here, $\boldsymbol{m}(\cdot)$ is the mean function and $\boldsymbol{k}(\cdot,\cdot)$ is a kernel function that represents the covariance function. Let us further denote an $n \times p$ matrix, denoted by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\intercal$. Then, (2) is simplified into

$$\boldsymbol{f}(\mathbf{X}) \,|\mathbf{X} \sim \mathcal{N}\left(\boldsymbol{m}(\mathbf{X}), \boldsymbol{k}(\mathbf{X},\mathbf{X})\right), \tag{3}$$

where $\boldsymbol{f}(\mathbf{X}) = (f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n))^\intercal$ and $\boldsymbol{m}(\mathbf{X}) = (m(\mathbf{x}_1), m(\mathbf{x}_2), \ldots, m(\mathbf{x}_n))^\intercal$.

Now, if an observation noise $\boldsymbol{\varepsilon}$ is added to $\boldsymbol{f}(\mathbf{X})$, we get

$$\mathbf{Y} = \boldsymbol{f}(\mathbf{X}) + \boldsymbol{\varepsilon}. \tag{4}$$

For independent, identically and normally distributed noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n)$ (where $\mathbf{I}_n$ and σ^2 are an n-dimensional identity matrix and the variance, respectively), using normality property, we obtain

$$\mathbf{Y}|\mathbf{X} \sim \mathcal{N}\left(\boldsymbol{m}(\mathbf{X}), \boldsymbol{k}(\mathbf{X},\mathbf{X}) + \sigma^2 \mathbf{I}_n\right). \tag{5}$$

Note that $\boldsymbol{\varepsilon}$ is also called a "nugget". If $\sigma^2 = 0$, then $f(x)$ is observed without noise. However, in practical implementation, the nugget is always added for the sake of numerical stability.

2) Bayesian Inference: Here, we present the way to use the abovementioned finite collection of n samples, $\{\mathbf{Y}, \mathbf{X}\}$, to infer the unknown system output, $\mathbf{y}(\mathbf{x})$, on the sample space of $\mathbf{x} \in \mathbb{R}^p$ in a Bayesian inference framework. We assume that the readers have the basic knowledge of Bayesian inference.

Here, the finite collection of samples $\{\mathbf{Y}, \mathbf{X}\}$ provides us with the observations. To infer a Bayesian posterior distribution of the unknown system output $\mathbf{y}(\mathbf{x})$, we must assume a Bayesian prior distribution of $\mathbf{y}(\mathbf{x})$, expressed as

$$\mathbf{y}(\mathbf{x})|\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{m}(\mathbf{x}), \boldsymbol{k}(\mathbf{x},\mathbf{x}) + \sigma^2 \mathbf{I}_{n_x}\right). \tag{6}$$

Then, we can formulate the joint distribution of $\mathbf{Y}$ and $\mathbf{y}(\mathbf{x})$ using (5) and (6) as

$$\begin{bmatrix} \mathbf{Y} \\ \mathbf{y}(\mathbf{x}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{m}(\mathbf{X}) \\ \boldsymbol{m}(\mathbf{x}) \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \right), \tag{7}$$

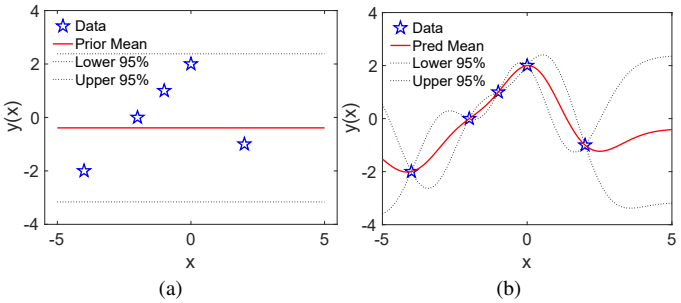


Fig. 1. (a) The prior and (b) updated posterior means as well as 95% confidence intervals (CIs) of a GP with constant mean function and squared exponential kernel. The posterior captures much more information of data than the prior does.

where $\mathbf{K}_{11} = \boldsymbol{k}(\mathbf{X},\mathbf{X}) + \sigma^2 \mathbf{I}_n$, $\mathbf{K}_{12} = \boldsymbol{k}(\mathbf{X},\mathbf{x})$, $\mathbf{K}_{21} = \boldsymbol{k}(\mathbf{x},\mathbf{X})$ and $\mathbf{K}_{22} = \boldsymbol{k}(\mathbf{x},\mathbf{x}) + \sigma^2 \mathbf{I}_{n_x}$. Now, we can infer $\mathbf{y}(\mathbf{x})$ based on previous observations $(\mathbf{Y}, \mathbf{X})$. Using the rules of the conditional Gaussian distribution (a.k.a. Gaussian conditioning or statistical linearization) [13], the Bayesian posterior distribution of the system output $\mathbf{y}(\mathbf{x})$ conditioned upon the observations $(\mathbf{Y}, \mathbf{X})$ follows a Gaussian distribution given by

$$\mathbf{y}(\mathbf{x})|\mathbf{x}, \mathbf{Y}, \mathbf{X} \sim \mathcal{N}\left(\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\Sigma}(\mathbf{x})\right), \tag{8}$$

where

$$\boldsymbol{\mu}(\mathbf{x}) = \boldsymbol{m}(\mathbf{x}) + \mathbf{K}_{21}\mathbf{K}_{11}^{-1}(\mathbf{Y} - \boldsymbol{m}(\mathbf{X})), \tag{9}$$

$$\boldsymbol{\Sigma}(\mathbf{x}) = \mathbf{K}_{22} - \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{K}_{12}. \tag{10}$$

To this point, the form of the GPE has been derived. Now, on one hand, we may directly use (9) as a surrogate model (a.k.a. the response surface or reduced-order model) to capture very closely the behavior of the complicated, original simulation model of a power system while being computationally inexpensive to evaluate. On the other hand, we may use (10) to quantify the uncertainty of the surrogate itself. In this paper, we only need to use (9) as a surrogate model. Fig. 1 shows a simple example of how five 1-dimensional data points update a Gaussian process prior to a posterior.

3) Mean and Covariance Functions: To further define the GPE, we need to select the forms of the mean function $\boldsymbol{m}(\cdot)$ and the covariance function represented via the kernel $\boldsymbol{k}(\cdot,\cdot)$.

The mean function models the prior belief about the existence of a systematic trend expressed as

$$\boldsymbol{m}(\mathbf{x}, \boldsymbol{\beta}) = \mathbf{H}(\mathbf{x})\boldsymbol{\beta}. \tag{11}$$

Here, $\mathbf{H}(\mathbf{x})$ can be any set of basis functions. For example, let $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$ indicate the ith sample, $i = 1, 2, \ldots, n$ and x_{ik} represents its kth element, $k = 1, 2, \ldots, p$. For instance, $\mathbf{H}(\mathbf{x}_i) = 1$ is a constant basis; $\mathbf{H}(\mathbf{x}_i) = (1, x_{i1}, \ldots, x_{ip})$ is a linear basis; $\mathbf{H}(\mathbf{x}_i) = (1, x_{i1}, \ldots, x_{ip}, x_{i1}^2, \ldots, x_{ip}^2)$ is a pure quadratic basis; and $\boldsymbol{\beta}$ is a vector of hyperparameters.

Since the covariance function is represented by a kernel function, choosing the latter is a must. Table I provides several popular covariance kernels.

As for the parameters of a kernel function, they are defined as follows: τ and ℓ_k are the hyperparameters defined in the positive real line; σ^2 and ℓ_k correspond to the order

Authorized licensed use limited to: to IEEExplore provided by University Libraries | Virginia Tech. Downloaded on May 30,2020 at 06:43:21 UTC from IEEE Xplore. Restrictions apply.

TABLE I. COMMONLY USED COVARIANCE KERNELS FOR GAUSSIAN PROCESS

$k_{\text{SE}}(\mathbf{x}_i, \mathbf{x}_j)$	$\tau^2 \exp\left(-\sum_{k=1}^{p} \frac{r_k^2}{2\ell_k^2}\right)$
$k_{\text{E}}(\mathbf{x}_i, \mathbf{x}_j)$	$\tau^2 \exp\left(-\sum_{k=1}^{p} \frac{\lvert r_k \rvert}{\ell_k}\right)$
$k_{\text{RQ}}(\mathbf{x}_i, \mathbf{x}_j)$	$\tau^2 \left(1 + \sum_{k=1}^{p} \frac{r_k^2}{2\alpha\ell_k^2}\right)^{-\alpha}$
$k_{3/2}(\mathbf{x}_i, \mathbf{x}_j)$	$\tau^2 \left(1 + \sum_{k=1}^{p} \frac{\sqrt{3}r_k}{\ell_k}\right) \exp\left(-\sum_{k=1}^{p} \frac{\sqrt{3}r_k}{\ell_k}\right)$
	$(r_k = \lvert x_{ik} - x_{jk} \rvert)$

Abbrv.: square exponential (SE), exponential (E), rational quadratic (RQ), and Martin 3/2 (3/2) kernels.

of magnitude and the speed of variation in the kth input dimension, respectively. Let $\boldsymbol{\theta} = (\tau, \ell_1, \ldots, \ell_p)$ contains the hyperparameters of the covariance function, i.e.,

$$k(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\theta}) = \text{Cov}(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\theta}). \tag{12}$$

Until now, the model structure of the GPE has been fully defined. For simplicity, we write $\boldsymbol{\eta} = (\sigma^2, \boldsymbol{\beta}, \boldsymbol{\theta})$ to represent all the hyperparameters in the GPE model.

4) Hyperparameter Estimation: Optimizing a GPE model is equivalent to estimating $\boldsymbol{\eta}$ given the data $(\mathbf{Y}, \mathbf{X})$. Although different methods exist for estimating the hyperparameters $\boldsymbol{\eta}$ [12], we choose to adopt the Gaussian maximum likelihood estimator (MLE) since it meets our demand and is straightforward to compute.

First, to indicate the hyperparameters, let us rewrite (5) as

$$\mathbf{Y}|\mathbf{X}, \boldsymbol{\eta} \sim \mathcal{N}\left(\boldsymbol{m}(\mathbf{X}), \boldsymbol{k}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_n\right). \tag{13}$$

Then, using MLE, we obtain

$$\widehat{\boldsymbol{\eta}} = \left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2\right) = \underset{\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2}{\arg\max} \log P\left(\mathbf{Y}|\mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2\right). \tag{14}$$

Using (11)–(13) and simplifying $\mathbf{H}(\mathbf{x})$ into $\mathbf{H}$, the marginal log-likelihood can be expressed as

$$\begin{aligned} &\log P\left(\mathbf{Y}|\mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2\right) \\ &= -\frac{1}{2}(\mathbf{Y} - \mathbf{H}\boldsymbol{\beta})^{\intercal} \left[\boldsymbol{k}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n\right]^{-1} (\mathbf{Y} - \mathbf{H}\boldsymbol{\beta}) \\ &\quad - \frac{n}{2} \log 2\pi - \frac{1}{2} \log \left|\boldsymbol{k}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n\right|, \end{aligned} \tag{15}$$

which implies that the MLE of $\boldsymbol{\beta}$ conditioned upon $\boldsymbol{\theta}$ and σ^2 is a weighted least-squares estimate given by

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}, \sigma^2) = \left[\mathbf{H}^{\intercal}\left[\boldsymbol{k}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n\right]^{-1} \mathbf{H}\right]^{-1} \mathbf{H}^{\intercal}\left[\boldsymbol{k}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n\right]^{-1} \mathbf{Y}. \tag{16}$$

Plugging (16) into (15), we get the $\boldsymbol{\beta}$-profile likelihood $\log P\left(\mathbf{Y}|\mathbf{X}, \hat{\boldsymbol{\beta}}\left(\boldsymbol{\theta}, \sigma^2\right), \boldsymbol{\theta}, \sigma^2\right)$. Then, (14) is rewritten as

$$\left(\widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2\right) = \underset{\boldsymbol{\theta}, \sigma^2}{\arg\max} \log P\left(\mathbf{Y}|\mathbf{X}, \hat{\boldsymbol{\beta}}\left(\boldsymbol{\theta}, \sigma^2\right), \boldsymbol{\theta}, \sigma^2\right), \tag{17}$$

where $\widehat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}\left(\widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2\right)$. The next goal is to find $\widehat{\boldsymbol{\eta}}$ from (15)–(17). Since $\widehat{\boldsymbol{\beta}}$ can be straightforwardly obtained from $\left(\widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2\right)$, one only needs to find $\left(\widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2\right)$ by maximizing the $\boldsymbol{\beta}$-profile likelihood over $\left(\boldsymbol{\theta}, \sigma^2\right)$. Here, we utilize a gradient-based optimizer to achieve this optimization. To overcome the presence of local optima in the objective function, we initialize σ^2 somewhere close to 0 because the global optimum is in this vicinity. Once $\widehat{\boldsymbol{\eta}}$ is obtained, the GPE model is fully constructed.

5) Sampling Strategy: In order to obtain the observation sets contained in $(\mathbf{Y}, \mathbf{X})$, we need some samples that satisfy the system function $\boldsymbol{f}(\mathbf{X})$, i.e., the SED model. Here, a popular choice to generate these samples is through the Latin hypercube sampling [14]. Unlike the Monte Carlo sampling, which generates a set of independent and identically distributed samples from the target probability distributions, the Latin hypercube sampling generates near-random samples that follow a standard uniform distribution $\mathcal{U}(0, 1)$ based on an equal-interval segmentation. For a nonuniform distribution, the inverse transformation of the cumulative distribution function is applied to map the uniformly distributed samples into the targeted distribution [15].

B. Karhunen-Loève Expansions

As it is mentioned in Section II, the dimension for the random fields representing wind-farm generation may be so high that the GPE cannot be constructed efficiently. Therefore, facing the challenge raised by a high-dimensional raw data, an efficient dimension reduction becomes a prerequisite.

1) Spectral Decomposition and Truncation: Here, we use the KLE to project the high-dimensional samples into low-dimension latent variables. Let us consider a bounded domain $D \subseteq \mathbb{R}$ and a sample space Ω, and let $X(t)$ be a zero-mean stochastic process with $t \in D$ where $X : D \times \Omega \to \mathbb{R}$. Each $X(t)$ is a random variable indexed by t. Let us assume that $X(t)$ for any time has a finite variance and let us define the covariance function C as $C(t, s) = \text{Cov}(X(t), X(s)), \forall t, s \in D$. Since C is positive definite, its spectral decomposition is obtained as

$$C(t, s) = \sum_{l=1}^{\infty} \lambda_i u_i(t) u_i(s). \tag{18}$$

Here, λ_i denotes the ith eigenvalue and u_i denotes the ith orthonormal eigenfunction of C. Then, we put $X(t)$ into a KLE framework as follows:

$$X(t) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} u_i(t) \xi_i, \tag{19}$$

where $\{\xi_i, i = 1, 2, ...\}$ are mutually uncorrelated univariate random variables with zero mean and unit variance. Here, we use an empirical covariance matrix, C, calculated from the data. By applying the inner product of $u_i(t)$ to both sides of (19) and making $u_i(t)$ orthonormal, we obtain

$$\sqrt{\lambda_i} \xi_i = \langle X(t), u_i(t) \rangle \quad \forall i. \tag{20}$$

Till now, the KLE maps $X(t)$ to latent variables ξ_i by projecting $X(t)$ onto $u_i(t)$. The inverse mapping can be achieved via (19) as well. Note that both transformations are linear. In practice, $X(t)$ is replaced by a finite summation with p elements, yielding

$$X(t) \approx \hat{X}(t) = \sum_{i=1}^{p} \sqrt{\lambda_i} u_i(t) \xi_i. \tag{21}$$

Authorized licensed use limited to: to IEEExplore provided by University Libraries | Virginia Tech. Downloaded on May 30,2020 at 06:43:21 UTC from IEEE Xplore. Restrictions apply.

2) Dimension Reduction: Here, we present the dimension reduction from the variance point of view. Consider the total variance of $X(t)$ over D. Using the orthonormality property of $u_i(t)$, we have

$$\int_D \mathrm{Var}[X(t)]\,\mathrm{d}t = \sum_{i=1}^{\infty} \lambda_i. \tag{22}$$

A finite series is used instead such that most of the variance is retained after truncation, yielding

$$\int_D \mathrm{Var}[X(t)]\,\mathrm{d}t \approx \int_D \mathrm{Var}[\hat{X}(t)]\,\mathrm{d}t = \sum_{i=1}^{p} \lambda_i. \tag{23}$$

Specifically, we choose the first p largest eigenvalues as $(\lambda_1, \ldots, \lambda_p)$, with the corresponding eigenfunctions as $(u_1, \ldots, u_p)$ such that the obtained $\boldsymbol{\xi}$ contain over 95% of the total variance calculated by $(\sum_{i=1}^{p} \lambda_i) / (\sum_{i=1}^{\infty} \lambda_i)$. Since the calculated p is smaller than the original dimension of the raw data sequence, the KLE has mapped the high-dimensional correlated X to the low-dimensional uncorrelated $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_p)$. It is worth pointing out that the truncated p dimensions will serve as the input for the SED problem.

IV. Proposed Method

Using the theory explained above, we propose a GPE-based method to solve the SED problem. We first obtain $\boldsymbol{\xi}$ of the p-truncated KLE for the SED input. Then, to avoid the normality assumption, we estimate the closed-form solution of the joint probability density function (pdf) of $\boldsymbol{\xi}$ via a kernel density estimation [16]. Later, using that density estimation, the input samples are regenerated. Finally, with the training samples selected from the Latin hypercube sampling, the GPE surrogate is constructed to propagate the uncertainty from the input samples to the output. The details are described in Algorithm 1. Note that since the input of the SED is spatiotemporally correlated, our approach is naturally compatible with the modeling of a spatiotemporal structure. The local $\boldsymbol{\xi}$ is calculated by applying the KLE individually to each location. After estimating the joint pdfs of $\boldsymbol{\xi}$ at all locations, samples can be drawn at every location for each time step.

Algorithm 1 The Proposed GPE-based SED Method

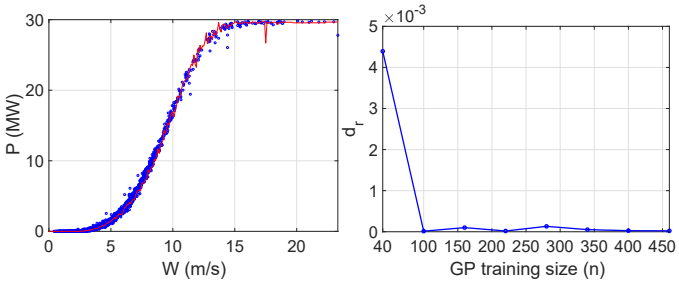
1: Read the historical raw data of the wind farms;
2: Apply the KLE on the preprocessed data to obtain p-dimensional truncated data;
3: Estimate the closed-form solution for the joint pdf of $\boldsymbol{\xi}$;
4: Regenerate a large number of samples as the model input through the pdf obtained from the kernel density estimation;
5: Generate p-dimensional n training samples $\mathbf{X}$ via the Latin hypercube sampling method;
6: Obtain realizations $\mathbf{Y}$ by evaluating $\mathbf{X}$ through the SED model;
7: Estimate the hyperparameters $\boldsymbol{\eta}$ given the training set $(\mathbf{Y}, \mathbf{X})$ for the GPE-based surrogate model;
8: Propagate a large number of samples of the model input through the GPE-based surrogate models to obtain the SED system output realizations;
9: Calculate the statistical moments of the SED output $Q(g, \mathbf{p}(\boldsymbol{\xi}))$.

V. Case Studies

We test our method on the IEEE 118-bus system [17] with the MATPOWER package using the MATLAB® R2019a version and the NREL's Western Wind Data Set [18]. In the experiment, we pick one farm in Livermore, CA (#LV) and two farms in Seattle, WA (#SE1, #SE2). For each farm, three turbines are extracted from the dataset: #9247, #9248, #9249 for #LV; #28914, #28928, #28959 for #SE1; and #29138, #29153, #29154 for #SE2. These three wind farms are added at Buses 16, 58, and 78, respectively.

Assuming that the turbines in the same farm have the same wind speed and wind power all the time, we calculate for each time stamp the averaged wind speed and wind power over three selected wind turbines and treat it as the prevailing wind speed and wind power of that farm. For each farm, we take hourly averages on the common wind speed and wind power to obtain the daily wind speed W and wind power P for each day in January between 2004 and 2006, which leads to a total of 93 data points. The relationship between P and W is modeled by a decision tree regression model. The KLE is used to represent the randomness of W. To preserve 95% of the total variance of W, the first 8, 4, and 5 KLE modes are kept for #LV, #SE1, and #SE2, respectively.

To test the spatial dependency of W among the three farms, we calculate distance correlation factors [19] between $\boldsymbol{\xi}$ on different farms. The results are shown in Appendix, where a way to deal with the dependency between #SE1 and #SE2 is also provided.



(a) (b)

Fig. 2. (a) Wind power P versus wind speed W at Farm #LV in January (Blue points denote real data points; whereas, red curve represents predictions by decision tree regression.). (b) The relative difference between $\mathrm{E}[Q_{\mathrm{GP}}]$ and $\mathrm{E}[Q_{\mathrm{MC}}]$ varying by n. In order to ensure that the MC converges, 8,000 realizations are used for Q_{MC}.

The GPE surrogate with a pure quadratic mean function and a squared exponential kernel is constructed for the SED test system. Let Q_{GP} denote the estimation of the minimum production cost Q from the GPE surrogate. Q_{MC} is calculated by direct Monte Carlo simulations performed on the test system. One of the most important results obtained from the SED is the expected minimum cost $\mathrm{E}[Q]$. We define the relative difference between $\mathrm{E}[Q_{\mathrm{GP}}]$ and $\mathrm{E}[Q_{\mathrm{MC}}]$ as $d_r = \frac{|\mathrm{E}[Q_{\mathrm{GP}}] - \mathrm{E}[Q_{\mathrm{MC}}]|}{\mathrm{E}[Q_{\mathrm{MC}}]} \cdot \mathrm{E}[Q_{\mathrm{MC}}]$ is a fixed baseline while $\mathrm{E}[Q_{\mathrm{GP}}]$ varies with the GPE training size n. The smaller the d_r is, the better the GPE surrogate fits its target.

Figure 2(b) depicts the trace plot of d_r over n. Notice that the approach achieves less than $10^{-3} d_r$ when $n = 100$.

Authorized licensed use limited to: to IEEExplore provided by University Libraries | Virginia Tech. Downloaded on May 30,2020 at 06:43:21 UTC from IEEE Xplore. Restrictions apply.

TABLE II. PREDICTIVE INFERENCES (MEAN, 95% CI AND STANDARD DEVIATION) OF MINIMUM PRODUCTION COST (GP TRAINING SIZE $n = 100$.)

	Mean ($\times 10^6$)	95% CI ($\times 10^6$)	Std. Dev. ($\times 10^4$)
Q_{MC}	2.955	(2.874, 3.018)	3.908
Q_{GP}	2.956	(2.869, 3.020)	4.078

Numerical inferences are listed in Table II. The GPE surrogate, trained with 100 Latin hypercube simulations, successfully calculates the SED values for the tested system under 8,000 scenarios.

TABLE III. RESULTS FROM REPLICATING EMPIRICAL MINIMUM PRODUCTION COST (GP TRAINING SIZE $n = 100$.)

	Mean ($\times 10^6$)	95% CI ($\times 10^6$)	Std. Dev. ($\times 10^4$)
Q_{data}	2.943	(2.849, 3.017)	4.996
Q^r_{GP}	2.949	(2.858, 3.025)	5.108

Ideally, a well-performing surrogate is also expected to reasonably replicate the empirical cost Q_{data} for 93 data points. Correspondingly, we have the replications Q_{GP} from the GPE-based surrogate estimation. The results in Table III indicate that the trained GPE surrogate is able to reproduce the empirical Q of the test system.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a GPE-based framework in quantifying uncertainty for the SED problem. The proposed framework utilizes the KLE to conduct an effective dimension reduction, which further accelerates the nonparametric GPE in the propagation of uncertainties. The simulation results on the modified IEEE 118-bus system show that the proposed method is significantly more computationally efficient than the traditional Monte Carlo method while achieving the desired simulation accuracy.

APPENDIX
MODELING SPATIAL CORRELATIONS OF WIND SPEEDS BETWEEN WIND FARMS IN SEATTLE

TABLE IV. DISTANCE CORRELATIONS FOR THE FIRST THREE OF $\boldsymbol{\xi}$ BETWEEN WIND FARMS

	#LV – #SE1	#LV – #SE2	#SE1 – #SE2
ξ_1	0.141	0.123	0.592
ξ_2	0.221	0.166	0.289
ξ_3	0.131	0.231	0.319

Table IV shows that ξ_1 between #SE1 and #SE2 have a relatively high distance correlation. One may still treat them independently since the value is not too close to 1. In addition, we provide a way to handle the dependency as described below.

Considering that the Pearson correlation of ξ_1 between #SE1 and #SE2 is calculated to be 0.618, it is proper to assume that they are linearly dependent, which can be modeled using a linear regression. If we use ξ_{11} and ξ_{12} to distinguish ξ_i in #SE1 and #SE2, the joint samples of ξ_{11}, ξ_{12} can be obtained in two steps:

1) Sample ξ_{11} from its density function;
2) Sample $\xi_{12} \sim \mathcal{N}(\hat{\beta}_0 + \xi_{11}\hat{\beta}_1, \hat{\sigma}^2)$, where $\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2$ are the estimated intercept, slope, and mean squared error from the results of the linear regression between ξ_{12} and ξ_{11}, respectively.

ACKNOWLEDGMENTS

This work was supported, in part, by the United States Department of Energy Office of Electricity Advanced Grid Modeling Program and performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, and by the U.S. National Science Foundation under EPAS Grant 1917308. Document released as LLNL-CONF-788518.

REFERENCES

[1] C. Safta, R. L.-Y. Chen, H. N. Najm, A. Pinar, and J. P. Watson, "Efficient uncertainty quantification in stochastic economic dispatch," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 2535–2546, Jul. 2017.

[2] Y. Xu, L. Mili, A. Sandu, M. R. von Spakovsky, and J. Zhao, "Propagating uncertainty in power system dynamic simulations using polynomial chaos," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 338–348, Jan. 2019.

[3] Y. Xu, L. Mili, and J. Zhao, "Probabilistic power flow calculation and variance analysis based on hierarchical adaptive polynomial chaos-ANOVA method," *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 3316–3325, Sept. 2019.

[4] Y. Xu *et al.*, "Response-surface-based Bayesian inference for power system dynamic parameter estimation," *IEEE Trans. Smart Grid*, 2019. [Online]. Available: https://doi.org/10.1109/TSG.2019.2892464

[5] X. Xu *et al.*, "Maximum loadability of islanded microgrids with renewable energy generation," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 4696–4705, Sept. 2019.

[6] H. Sheng and X. Wang, "Applying polynomial chaos expansion to assess probabilistic available delivery capability for distribution networks with renewables," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6726–6735, Nov. 2018.

[7] P. A. Ruiz, C. R. Philbrick, E. Zak, K. W. Cheung, and P. W. Sauer, "Uncertainty management in the unit commitment problem," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 642–651, May 2009.

[8] S. Takriti, J. R. Birge, and E. Long, "A stochastic model for the unit commitment problem," *IEEE Trans. Power Syst.*, vol. 11, no. 3, pp. 1497–1508, Aug. 1996.

[9] J. Li, N. Ou, G. Lin, and W. Wei, "Compressive sensing based stochastic economic dispatch with high penetration renewables," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1438–1449, Mar. 2019.

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

[11] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*. Mineola, NY, USA: Dover Publications, Inc., 2003.

[12] A. Gelman *et al.*, *Bayesian Data Analysis*, 3rd ed. Boca Raton, FL, USA: Chapman & Hall, 2014.

[13] M. L. Eaton, *Multivariate Statistics: A Vector Space Approach*. New York, NY, USA: John Wiley & Sons, 1983.

[14] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*, 2nd ed. New York, NY, USA: Springer, 2018.

[15] L. Devroye, "Sample-based non-uniform random variate generation," in *Proc. 18th ACM Conf. Winter Simul.*, 1986, pp. 260–265.

[16] B. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, 1998.

[17] University of Washington, Power Systems Test Case Archive. (accessed on 8/30/2019). [Online]. Available: http://www.ee.washington.edu/research/pstca/

[18] NREL Western Wind Data Set. (accessed on 8/30/2019). [Online]. Available: https://www.nrel.gov/grid/western-wind-data.html

[19] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *Ann. Statist.*, vol. 35, no. 6, pp. 2769–2794, 2007.

Authorized licensed use limited to: to IEEExplore provided by University Libraries | Virginia Tech. Downloaded on May 30,2020 at 06:43:21 UTC from IEEE Xplore. Restrictions apply.