Probabilistic Power-Flow Calculation Based on a Novel Gaussian Process Emulator

Yijun Xu, *Member, IEEE*, Zhixiong Hu, Lamine Mili, *Life Fellow, IEEE*, Mert Korkali, *Senior Member, IEEE*, and Xiao Chen

***Abstract*—In this letter, a novel Gaussian process emulator is proposed, for the first time, to conduct the probabilistic power-flow calculation. Based on Bayesian inference, a Gaussian process emulator is trained and served as a nonparametric, reduced-order model of the nonlinear power-flow model. This emulator allows us to evaluate the time-consuming power-flow solver at the sampled values with a negligible computational cost. The simulations reveal the excellent performance of this method.**

***Index Terms*—Probabilistic power flow, Gaussian process emulator, Latin hypercube sampling, copula.**

I. Introduction

POWER systems exhibit stochastic dynamics, in particular due to the continuous variations of the loads and the intermittency of the renewable generation, among other causes. This results in modeling uncertainties that bring formidable challenges to power system planning and operation process. To solve this problem, the probabilistic power-flow (PPF) problem attracts more academic attention recently since it can quantify the uncertainties induced in the bus voltages and the line power flows by the uncertainties of the loads, the power generation, and the network parameters [1]. Traditionally, Monte-Carlo (MC) simulations have been utilized to address this problem. However, it turns out that tens of thousands of MC simulations are required to achieve sufficiently accurate results, which is too time-consuming for practical power system applications. Some analytical methods have been proposed to reduce the computation burden, but at the expense of the estimation accuracy. This is due to some assumptions in the mathematical derivation, such as the linearization for the nonlinear power-flow model [1], [2].

To overcome the abovementioned shortcomings, this letter proposes, for the first time, to utilize a method based on a Gaussian process emulator (GPE) to solve the PPF problem. Known as a Bayesian-learning-based method for a nonlinear regression problem in the statistics community [3], the GPE can serve as a nonparametric, reduced-order model representation for the nonlinear power-flow model.

Unlike the linearity assumption used in some derivative-based methods, the GPE method can preserve the nonlinearity information better. This is very important in practice when the system is under heavy load. Furthermore, unlike some parametric models that suffer from the limitations arising from the type of probability distributions assumed in the model inputs [4], our nonparametric GPE-based method has no limitation on the type of input distributions. Furthermore, the GPE method greatly relieves the "curse of dimensionality" in the traditional response-surface method [4]. Besides, it does not have some drawbacks of other methods in handling nonlinearity such as the bad tail behavior of Cornish-Fisher and Edgeworth expansions, and the negative cumulative probability values of Gram-Charlier series [4]. All these further extend the applicability of the proposed method. Simulation results reveal the impressive performance of the proposed method.

Y. Xu and L. Mili are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Northern Virginia Center, Falls Church, VA 22043 USA (e-mail:{yijunxu,lmili}@vt.edu).

Z. Hu is with the Department of Statistics, University of California-Santa Cruz, Santa Cruz, CA 95064 USA (e-mail: zhu95@ucsc.edu).

M. Korkali and X. Chen are with Lawrence Livermore National Laboratory, Livermore, CA 94550 USA (e-mail: {korkali1,chen73}@llnl.gov).

This work was supported, in part, by the U.S. NSF under Grant 1917308 and 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. Document released as LLNL-JRNL-790158.

II. Gaussian Process Emulator

A. Problem Description

Let us first formulate the PPF problem in the GPE framework. Here, the ac power-flow model is denoted by $f(\cdot)$. Its corresponding vector-valued random input of p dimensions is denoted as $\mathbf{x}$, which accounts for the uncertainties from the variations of the loads and the renewable energy generation. 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 $f(\mathbf{x})$, such as voltage magnitude and line power flow, 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 as

$$\begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_n) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix} \right). \tag{1}$$

Here, let us denote $\boldsymbol{m}(\cdot)$ as the mean function and $\boldsymbol{k}(\cdot, \cdot)$ as a kernel function that represents the covariance function. Then, (1) can be simplified as

$$\boldsymbol{f}(\mathbf{X}) \,|\mathbf{X} \sim \mathcal{N}\left(\boldsymbol{m}(\mathbf{X}), \boldsymbol{k}(\mathbf{X}, \mathbf{X})\right), \tag{2}$$

where $\mathbf{X}$ is an $n \times p$ matrix, denoted by $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\mathsf{T}$; $\boldsymbol{f}(\mathbf{X})$ stands for $[f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)]^\mathsf{T}$; and $\boldsymbol{m}(\mathbf{X})$ represents $[m(\mathbf{x}_1), m(\mathbf{x}_2), \ldots, m(\mathbf{x}_n)]^\mathsf{T}$.

Now, if an independent, identically (and normally) distributed (i.i.d.) 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) is considered in the system output, $\boldsymbol{f}(\mathbf{X})$, using normality property, we have the observations $\mathbf{Y}$ represented as

$$\mathbf{Y}|\mathbf{X} \sim \mathcal{N}\left(\boldsymbol{m}(\mathbf{X}), \boldsymbol{k}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_n\right). \tag{3}$$

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 [5]. Here, we set its initial value to 0.001.

B. 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.

It is well-known that a Bayesian posterior distribution of the unknown system output can be inferred from a Bayesian prior distribution of $\mathbf{y}(\mathbf{x})$ and the likelihoods obtained from the observations. Let us first assume a Bayesian prior distribution of $\mathbf{y}(\mathbf{x})|\mathbf{x}$, expressed as $\mathbf{y}(\mathbf{x})|\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{m}\left(\mathbf{x}\right), \boldsymbol{k}\left(\mathbf{x},\mathbf{x}\right) + \sigma^2\mathbf{I}_{n_x}\right)$. Combined with the observations provided by the finite collection of the samples $\{\mathbf{Y}, \mathbf{X}\}$, we can formulate the joint distribution of $\mathbf{Y}$ and $\mathbf{y}(\mathbf{x})|\mathbf{x}$ as

$$\begin{bmatrix} \mathbf{Y} \\ \mathbf{y}(\mathbf{x})|\mathbf{x} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{m}\left(\mathbf{X}\right) \\ \boldsymbol{m}\left(\mathbf{x}\right) \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix}\right), \tag{4}$$

where $\mathbf{K}_{11} = \boldsymbol{k}\left(\mathbf{X},\mathbf{X}\right) + \sigma^2\mathbf{I}_n$; $\mathbf{K}_{12} = \boldsymbol{k}\left(\mathbf{X},\mathbf{x}\right)$; $\mathbf{K}_{21} = \boldsymbol{k}\left(\mathbf{x},\mathbf{X}\right)$; and $\mathbf{K}_{22} = \boldsymbol{k}\left(\mathbf{x},\mathbf{x}\right) + \sigma^2\mathbf{I}_{n_x}$. Note that since we employ a Gaussian process to emulate $\mathbf{y}(\mathbf{x})$, $\mathbf{y}(\mathbf{x})$ should have a finite mean and a finite variance. Now, using the rules of the conditional Gaussian distribution (a.k.a. Gaussian conditioning [6]), we can infer the Bayesian posterior distribution of the system output $\mathbf{y}(\mathbf{x})$ conditioned upon the observations $(\mathbf{Y}, \mathbf{X})$. It follows a Gaussian distribution given by $\mathbf{y}(\mathbf{x})|\mathbf{x}, \mathbf{Y}, \mathbf{X} \sim \mathcal{N}\left(\boldsymbol{\mu}\left(\mathbf{x}\right), \boldsymbol{\Sigma}\left(\mathbf{x}\right)\right)$. Here, we have

$$\boldsymbol{\mu}\left(\mathbf{x}\right) = \boldsymbol{m}(\mathbf{x}) + \mathbf{K}_{21}\mathbf{K}_{11}^{-1}(\mathbf{Y} - \boldsymbol{m}(\mathbf{X})), \tag{5}$$

$$\boldsymbol{\Sigma}\left(\mathbf{x}\right) = \mathbf{K}_{22} - \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{K}_{12}. \tag{6}$$

To this point, the general form of the GPE has been derived. On the one hand, we can directly use (5) as a surrogate model (a.k.a. the response surface or reduced-order model) to very closely capture the behavior of the nonlinear power-flow model while being computationally inexpensive to evaluate. On the other hand, we may use (6) to quantify the uncertainty of the surrogate itself. Note that (6) is widely used in the literature pertaining to data-driven model validation to eliminate outliers [7]; however, in this letter, we do not use (6) since we do not solve a data-driven PPF problem with observation outliers. Thus, we only need to use (5) as a surrogate model.

C. Mean and Covariance Functions

Let us describe the mean function $\boldsymbol{m}(\cdot)$ and the covariance function represented via the kernel $\boldsymbol{k}(\cdot,\cdot)$ here.

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

$$\boldsymbol{m}(\mathbf{x}, \boldsymbol{\beta}) = \mathbf{H}(\mathbf{x})\boldsymbol{\beta}. \tag{7}$$

Here, $\mathbf{H}(\mathbf{x})$ can be any set of basis functions. For example, let $\mathbf{x}_i = [x_{i1}, \ldots, x_{ip}]$ be the ith sample, where $i = 1, 2, \ldots, n$, wherein x_{ik} represents its kth element, where $k = 1, 2, \ldots, p$. 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. Popular choices are listed in Table I. 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 of the magnitude and the speed of variation in the kth input dimension, respectively.[1] Let $\boldsymbol{\theta} = [\tau, \ell_1, \ldots, \ell_p]$ contain the hyperparameters of the covariance function, i.e.,

$$k\left(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\theta}\right) = \mathrm{Cov}(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\theta}). \tag{8}$$

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.

TABLE I
COMMONLY USED COVARIANCE KERNELS FOR GAUSSIAN PROCESS

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

III. PROBABILISTIC POWER FLOW USING GPE

Here, we illustrate the steps for the proposed method.

1) Generation of the Training Sample: In order to acquire the GPE-based surrogate described in (5), we need to obtain the observation sets contained in $(\mathbf{Y}, \mathbf{X})$. To obtain the system realization $\mathbf{Y}$, we must generate n samples, $\mathbf{X}$, that will be evaluated through the power-flow model, $f(\cdot)$. To avoid long training time of the GPE, n should be small. To meet this requirement, Latin hypercube sampling is typically chosen. It generates almost random samples based on equal-interval segmentation and, therefore, has a faster convergence rate than the MC sampling, which generates purely random samples [9]. This is especially true in our case since n needs to be small [10] and, therefore, the MC sampling based on the central limit theorem is not suggested here.

2) Correlation between the Samples: In power systems, since the uncertainties of the input $\mathbf{x}$ may come from renewable generation, the correlation within different inputs cannot be ignored. To handle these correlations, the use of copula is suggested. According to Sklar's theorem, any multivariate cumulative distribution function $F_{\mathbf{X}}$ of a p-dimensional random vector can be expressed in terms of its marginal distributions and a copula to represent their dependence. Formally, we have

$$F_{\mathbf{X}}(\mathbf{x}) = C(F_{X_1}(x_1), F_{X_2}(x_2), \ldots, F_{X_p}(x_p)). \tag{9}$$

Here, $F_{X_i}(x_i)$ is the ith input marginal and $C(\cdot)$ is a copula that describes the dependence structure between the p-dimensional input variables. In this letter, the Gaussian copula is used to transform the i.i.d. samples into correlated ones as suggested in [11]. Using these correlated input samples, we can obtain the realizations of the power-flow response $\mathbf{Y}$ at the value of $\mathbf{X}$, accordingly.

3) GPE Construction: With $(\mathbf{Y}, \mathbf{X})$, we can estimate the hyperparameters $\boldsymbol{\eta}$ in the GPE. Although different methods exist for estimating the hyperparameters $\boldsymbol{\eta}$ [12], we choose to adopt the Gaussian maximum likelihood estimator (MLE) since it is optimal under the Gaussian assumption and is straightforward to compute. First, to indicate the hyperparameters, let us rewrite (3) as

$$\mathbf{Y}|\mathbf{X}, \boldsymbol{\eta} \sim \mathcal{N}\left(\boldsymbol{m}\left(\mathbf{X}\right), \boldsymbol{k}\left(\mathbf{X},\mathbf{X}\right) + \sigma^2\mathbf{I}_n\right). \tag{10}$$

[1] For more information about the kernel functions, please refer to [8, ch. 4].

Then, using the MLE, we obtain

$$\widehat{\boldsymbol{\eta}} = \left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2\right) = \underset{\boldsymbol{\beta},\boldsymbol{\theta},\sigma^2}{\arg\max} \log P\left(\mathbf{Y}|\mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2\right). \tag{11}$$

Using (7), (8), (10) and simplifying $\mathbf{H}(\mathbf{x})$ into $\mathbf{H}$, the marginal log-likelihood can be expressed as

$$\log P\left(\mathbf{Y}|\mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2\right) = -\frac{1}{2}(\mathbf{Y}-\mathbf{H}\boldsymbol{\beta})^{\intercal}\left[\boldsymbol{k}(\mathbf{X},\mathbf{X}|\boldsymbol{\theta}) + \sigma^2\mathbf{I}_n\right]^{-1}(\mathbf{Y}-\mathbf{H}\boldsymbol{\beta}) - \frac{n}{2}\log 2\pi - \frac{1}{2}\log\left|\boldsymbol{k}(\mathbf{X},\mathbf{X}|\boldsymbol{\theta}) + \sigma^2\mathbf{I}_n\right|, \tag{12}$$

which implies that the MLE of $\boldsymbol{\beta}$ conditioned upon $\boldsymbol{\theta}$ and σ^2 results in a weighted least-squares estimate expressed as

$$\hat{\boldsymbol{\beta}}\left(\boldsymbol{\theta}, \sigma^2\right) = \left[\mathbf{H}^{\intercal}\left[\boldsymbol{k}(\mathbf{X},\mathbf{X}|\boldsymbol{\theta}) + \sigma^2\mathbf{I}_n\right]^{-1}\mathbf{H}\right]^{-1}\mathbf{H}^{\intercal}\left[\boldsymbol{k}(\mathbf{X},\mathbf{X}|\boldsymbol{\theta}) + \sigma^2\mathbf{I}_n\right]^{-1}\mathbf{Y}. \tag{13}$$

Plugging (13) into (12), we obtain 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, (11) is rewritten as

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

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 (12)–(14). Since $\widehat{\boldsymbol{\beta}}$ can be straightforwardly obtained from $\left(\widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2\right)$, one only needs to find the latter by maximizing the $\boldsymbol{\beta}$-profile likelihood over $\left(\boldsymbol{\theta}, \sigma^2\right)$. Here, we utilize a gradient-based optimizer. Once $\widehat{\boldsymbol{\eta}}$ is obtained, the GPE model is fully constructed. More details can be found in [12].

Remark. *To guarantee the convergence of this algorithm, "n" should be no less than the number of unknown hyperparameters. In practice, to obtain an accurate estimation result, it is suggested to choose an "n" with a redundancy of $2 \sim 3$ with respect to the number of unknown hyperparameters.*

4) Sample Evaluation: Now, we can execute an MC sampling procedure to generate a large number of samples and transform them into correlated ones through a Gaussian copula. These samples can be evaluated through the GPE-based surrogate expressed in (5) at almost no computational cost. Finally, the probability distribution function of the system response can be obtained.

IV. Simulation Results

A case study is first conducted on the IEEE 57-bus system [13] with MATPOWER package using MATLAB® R2018a version. The uncertain inputs include the variations in the loads as well as in the wind and the solar generation. Here, it is assumed that the loads follow a Gaussian distribution with mean values equal to the original bus loads and standard deviations equal to 5% of their means. A wind generation following a Weibull distribution with the shape parameter of 2.06 and the scale parameter of 7.41 is connected to Bus 17. A 50-MW solar photovoltaic (PV) generation following a Beta distribution with shape parameters of 2.06 and 2.5 is appended to Bus 16 [4]. Considering their geographical proximity to the two renewable generation resources, the correlation matrix of these three inputs is provided in Table II.

TABLE II
CORRELATION MATRIX OF THE INPUTS

	Load	**Wind**	**Solar**
Load	1	0.49	0.301
Wind	0.49	1	0.551
Solar	0.301	0.551	1

1) Validation of the GPE Method: Let us choose the voltage magnitude at Bus 13 as the target quantity of interest. The simulation results obtained with the MC and the GPE methods are provided in Fig. 2. The simulation results obtained with the MC method with $20,000$ samples are used as a benchmark to validate the GPE-based method. To demonstrate that the result of the MC method can serve as a credible benchmark, we plot the convergence rate of the MC method in Fig. 1. It shows that the mean and variance of the MC method converge asymptotically given $20,000$ realizations. This proves the credibility of the results obtained with the MC method.

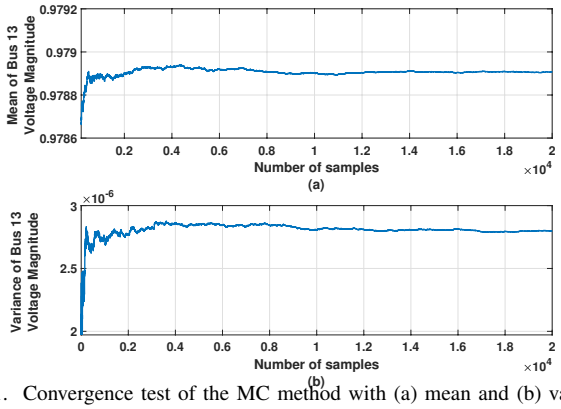

Fig. 1. Convergence test of the MC method with (a) mean and (b) variance of voltage magnitude at Bus 13.

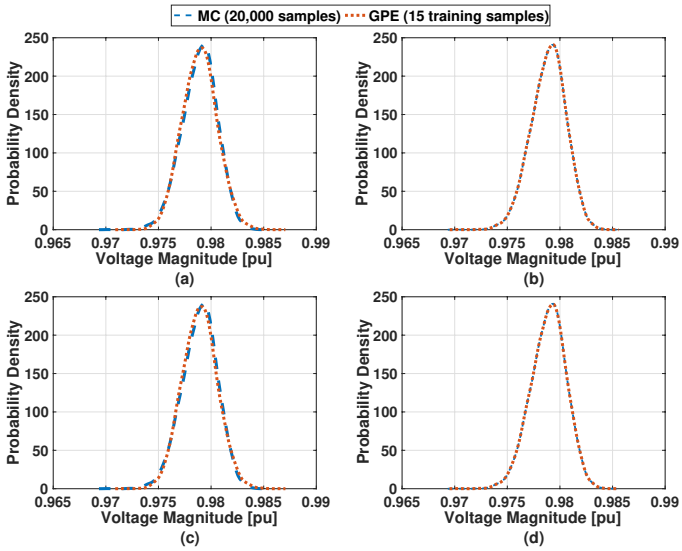

Fig. 2. Probability density plots for voltage magnitude at Bus 13 using GPE (a) with a linear basis and $k_{3/2}$; (b) with a pure quadratic basis and $k_{3/2}$; (c) with linear basis and k_{SE}; and (d) with a pure quadratic basis and k_{SE}.

It can be seen in Fig. 2 that with only 15 training samples, the GPE with a pure quadratic basis functions can provide highly accurate simulation results under different kernel functions, such as k_{SE} and $k_{3/2}$. As shown in Table III, it only takes 0.406 s to train the GPE. Then, 20,000 samples can be evaluated with the GPE-based surrogate in just 3.42 s. Therefore, the proposed GPE-based method significantly reduces the computing time over the traditional MC method without losing accuracy.

2) Quantitative Accuracy Test: To quantitatively verify the accuracy of the proposed method, we choose to use the well-known Kullback-Leibler divergence (KLD) as the index [14], which measures the difference between the GPE-

TABLE III
CPU TIMES OF THE GPE AND MC METHODS FOR THE IEEE 57-BUS SYSTEM

Method	**MC**	**GPE (Train)**	**GPE (Samples)**	**GPE (Total)**
CPU Time	627.85 s	0.406 s	3.42 s	3.83 s

approximated pdf, π_{GPE}, and the MC-approximated pdf, π_{MC}, via $D(\pi_{\text{GPE}}||\pi_{\text{MC}}) = \int \pi_{\text{GPE}} \log \frac{\pi_{\text{GPE}}}{\pi_{\text{MC}}}$. It is obvious that $D(\pi_{\text{GPE}}||\pi_{\text{MC}})$ equals 0 when $\pi_{\text{GPE}} = \pi_{\text{MC}}$. Then, we use the KLD to quantitatively test the accuracy of the proposed method with different trend functions, $\boldsymbol{m}(\cdot)$, different kernels, $\boldsymbol{k}(\cdot,\cdot)$, and different training sample sizes, n, as shown in Table IV. It can be seen that the GPE with pure quadratic basis functions provides highly accurate results under all the types of kernels being considered since $D(\pi_{\text{GPE}}||\pi_{\text{MC}})$ tends to 0. This holds true even if n is further reduced to 10.

TABLE IV
VALIDATION OF THE PROPOSED METHODS USING KLD

	Group A: Linear ($n = 15$)		**Group B: Quadratic** ($n = 15$)	
Group	k_{SE}	k_{E}	k_{RQ}	$k_{3/2}$
A	0.3487	0.3478	0.3647	0.3624
B	$3.0253 \cdot 10^{-4}$	$4.1391 \cdot 10^{-4}$	$5.0211 \cdot 10^{-4}$	$5.3781 \cdot 10^{-4}$
	Group C: Linear ($n = 10$)		**Group D: Quadratic** ($n = 10$)	
C	0.3467	0.3641	0.3348	0.3547
D	0.0016	0.0020	0.0012	0.0014

3) Global Test: Since it is hard to visualize the pdfs for each bus voltage magnitude and phase angle, we plot the mean and the standard deviation (indicated by error bars) for voltage magnitude and phase angle at all buses obtained via the GPE and MC methods in Fig. 4. It can be seen that the GPE method yields accurate estimation results for voltage magnitude and phase angle at all buses. The same strategy to validate the global accuracy of the proposed method through the mean and standard deviation has also been adopted in [4].

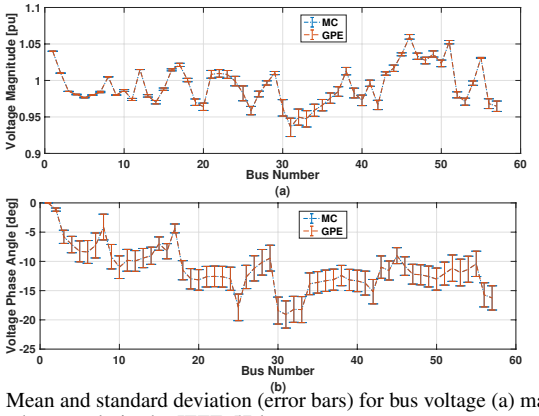

Fig. 3. Mean and standard deviation (error bars) for bus voltage (a) magnitude and (b) phase angle in the IEEE 57-bus system.

4) Comparison Study: Here, we compare the performance of the proposed GPE method with the recently advocated generalized-polynomial-chaos (gPC) method in the PEGASE 1,354-bus system with more than 20 renewable generation resources. All the experiment settings are exactly the same with the previous work (cf. [15]). Figure 4 shows that both the GPE method and the gPC method are very accurate. However, unlike the gPC method that suffers from the "curse of dimensionality" for a high-dimensional case, the GPE method with 100 training samples and quadratic basis significantly outperforms the gPC method (in terms of computing time) as revealed in Table V. This makes the GPE method more applicable for large-scale power system applications.

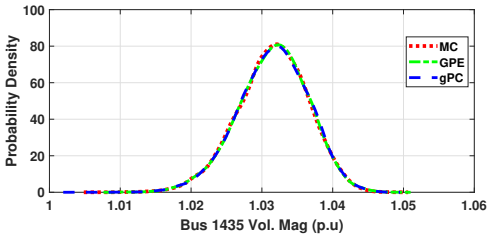

Fig. 4. Comparison study for the proposed method in the 1,354-bus system.

TABLE V
CPU TIMES OF THE GPE, GPC, AND MC METHODS FOR THE PEGASE 1,354-BUS SYSTEM

Method	**MC**	**gPC**	**GPE**
CPU Time	35 min	32 s	11 s
KLD	—	0.0227	0.0137

V. CONCLUSIONS

In this letter, we propose a novel GPE-based method for the PPF analysis. The GPE enables us to evaluate the power-flow solver at the sampled values with a negligible computational cost while maintaining high accuracy in the PPF application.

REFERENCES

[1] A. M. Leite da Silva and V. Arienti, "Probabilistic load flow by a multilinear simulation algorithm," *IEE Proc. C (Gener. Transm. Distrib.)*, vol. 137, no. 4, pp. 276–282, Jul. 1990.
[2] T. Williams and C. Crawford, "Probabilistic load flow modeling comparing maximum entropy and Gram-Charlier probability density function reconstructions," *IEEE Trans. Power Syst.*, vol. 28, no. 1, pp. 272–280, Feb. 2013.
[3] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *J. R. Stat. Soc. Ser. B (Stat. Method.)*, vol. 63, no. 3, pp. 425–464, 2001.
[4] Z. Ren, W. Li, R. Billinton, and W. Yan, "Probabilistic power flow analysis based on the stochastic response surface method," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2307–2315, May 2016.
[5] A. Pepelyshev, "The role of the nugget term in the Gaussian process method," in *mODa 9–Advances in Model-Oriented Design and Analysis*. Springer, 2010, pp. 149–156.
[6] M. L. Eaton, *Multivariate Statistics: A Vector Space Approach*. New York, NY, USA: John Wiley & Sons, 1983.
[7] L. S. Bastos and A. O'Hagan, "Diagnostics for Gaussian process emulators," *Technometrics*, vol. 51, no. 4, pp. 425–438, Nov. 2009.
[8] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[9] M. Hajian, W. D. Rosehart, and H. Zareipour, "Probabilistic power flow by Monte Carlo simulation with Latin supercube sampling," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1550–1559, May 2013.
[10] 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.
[11] G. Papaefthymiou and D. Kurowicka, "Using copulas for modeling stochastic dependence in power system uncertainty analysis," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 40–49, Feb. 2009.
[12] A. Gelman *et al.*, *Bayesian Data Analysis*, 3rd ed. Boca Raton, FL, USA: Chapman & Hall, 2014.
[13] University of Washington, Power Systems Test Case Archive. (accessed: 10/2019). [Online]. Available: https://labs.ece.uw.edu/pstca/
[14] Y. Marzouk and D. Xiu, "A stochastic collocation approach to Bayesian inference in inverse problems," *Commun. Comput. Phys.*, vol. 6, no. 4, pp. 826–847, Oct. 2009.
[15] 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, Sep. 2019.