

# Probabilistic Load-Margin Assessment using Vine Copula and Gaussian Process Emulation

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

**Abstract**—The increasing penetration of renewable energy along with the variations of the loads bring large uncertainties in the power system states that are threatening the security of power system planning and operation. Facing these challenges, this paper proposes a cost-effective, nonparametric method to quantify the impact of uncertain power injections on the load margins. First, we propose to generate system uncertain inputs via a novel vine copula due to its capability in simulating complex multivariate highly dependent model inputs. Furthermore, to reduce the prohibitive computational time required in the traditional Monte-Carlo method, we propose to use a nonparametric, Gaussian-process-emulator-based reduced-order model to replace the original complicated continuation power-flow model. This emulator allows us to execute the time-consuming continuation power-flow solver at the sampled values with a negligible computational cost. The simulations conducted on the IEEE 57-bus system, to which correlated renewable generation are attached, reveal the excellent performance of the proposed method.

**Index Terms**—Probabilistic load margin, Gaussian process emulator, vine copula, uncertainty, voltage stability.

## 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. To address this problem, research activities have focused on uncertainty assessment in power system planning, operation, and control. Examples include a probabilistic power flow [1], [2], an uncertainty quantification for power system dynamic simulation [3], a stochastic economic dispatch [4], [5], and a probabilistic load-margin formulation [6], [7], among others. The latter formulation is critical to ensuring the voltage stability of modern power systems with increasing penetration of renewables and, therefore, is chosen as the scope of this paper.

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).

K. Karra is with the Applied Physics Lab at Johns Hopkins University Baltimore, MD 21218 USA (e-mail: kkarranc@vt.edu).

M. Korkali is with the Computational Engineering Division, Lawrence Livermore National Laboratory, Livermore, CA 94550 USA (e-mail: korkali1@llnl.gov).

X. Chen is with the Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94550 USA (e-mail: chen73@llnl.gov)

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

This work was supported, in part, by the U.S. National Science Foundation 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-PROC-795980.

Traditionally, Monte-Carlo (MC) simulations have been utilized to address this problem [8]. However, the computing time of a single continuation power flow (CPF) case is much longer than that of a simple power-flow case. Therefore, the straightforward MC method based on the evaluations of tens of thousands samples in the CPF model will be prohibitively time-consuming. This is even true for relatively small systems. Although some analytical methods have been proposed to reduce the computational burden, there exists no very accurate closed-form solution today due to the nonlinearity of the CPF model [6].

To overcome the abovementioned shortcomings, this paper proposes, for the first time, to utilize a method based on a Gaussian process emulator (GPE) to solve the probabilistic CPF problem. Known as a Bayesian-learning-based method for a nonlinear regression problem in statistics [9], the GPE can serve as a nonparametric, reduced-order model representation for the nonlinear CPF model. This emulator allow us to evaluate the time-consuming CPF solver at the sampled values with a negligible computational cost. Furthermore, to simulate the high-dimensional dependent samples that represent the uncertainties from the loads and renewables, we propose to adopt a novel vine-copula technique [10], which is capable of modeling the high-dimensional, dependent multivariate with a variety of bivariate copulae such as Frank and Gumbel copulae to better represent tail dependence in the correlated samples. This vine copula performs better than the the Gaussian copula [7], [11] since the latter has been proven to have 0 tail dependence and, therefore, is less precise when describing the complicated dependence structures existing in renewable generation [12], [13]. Simulation results carried out on the IEEE 57-bus test system system reveal that our proposed method can accurately estimate the probability density function (pdf) of the load margin with more than two-order-of-magnitude improvement in computing speed compared to the traditional MC method.

## II. PROBLEM FORMULATION

This section formulates the probabilistic load-margin assessment problem.

Let us first formulate the power system forward model as

$$y = f(\mathbf{x}). \quad (1)$$

Here,  $y$  stands for the quantity of interest (QoI), which, in our case, is the load margin of a bus;  $\mathbf{x} = [x_1, x_2, \dots, x_p]$  is a vector of uncertain model parameters described by some distribution functions with finite variance. In our work, the

active power and reactive power of the loads are considered to follow a Gaussian distribution and the wind power generation are assumed to follow the Weibull distribution; the  $f(\cdot)$  is the nonlinear function that represents the continuation power system model, which maps the model parameters,  $\mathbf{x}$ , to the QoI,  $y$ . The detailed implementation step has been described by Ajjarapu [14].

To obtain the probabilistic description of the load margin  $y$  under these uncertain model parameters, a typical MC method draws a large number of  $N_{\text{sample}}$  samples,  $\{\mathbf{x}^{(j)}\}_{j=1}^{N_{\text{sample}}}$ , that not only reflect the pdfs of the input parameters but also the correlation between them. Then, for each  $\mathbf{x}^{(j)}$ ,  $j = 1, \dots, N_{\text{sample}}$ ,  $y^{(j)} = f(\mathbf{x}^{(j)})$  is solved to get  $N_{\text{sample}}$  load-margin solutions,  $\{y^{(j)}\}_{j=1}^{N_{\text{sample}}}$ , from which the pdf of the load margin is determined. The CPF method is typically employed in power systems despite the fact that even a single evaluation of  $f(\cdot)$  will involve multiple prediction and correction steps to obtain the load margin,  $y$ , and, hence, is admittedly a complicated, time-consuming solver—not to mention that  $N_{\text{sample}}$  is typically required to be a large number in the MC sampling to ensure good computing accuracy. Therefore, the goal of this paper is to greatly reduce the computational time of this method and to precisely model the correlation between the uncertain parameters as model inputs.

### III. UNCERTAINTY MODELING

In this section, we present the way to generate dependent high-dimensional samples as model inputs via vine copula.

1) *Copula*: Recently, copulae have been proven to be successful in many industrial and financial applications for modeling the dependency between random inputs [10], [13]. According to Sklar's theorem, any joint 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

$$F_{\mathbf{X}}(\mathbf{x}) = C(F_{X_1}(x_1), F_{X_2}(x_2), \dots, F_{X_p}(x_p)). \quad (2)$$

Here,  $F_{X_i}(x_i)$  is the  $i$ th input marginal and  $C(\cdot)$  is a copula that describes the dependence structure between the  $p$ -dimensional input variables [15]. Accordingly, its joint multivariate density function,  $f_{\mathbf{X}}$ , can be obtained via

$$f_{\mathbf{X}}(\mathbf{x}) = c(F_{X_1}(x_1), \dots, F_{X_p}(x_p)) \prod_{i=1}^p f_i(x_i). \quad (3)$$

Here,  $c$  is the  $p$ -variate copula density and  $f_i(x_i)$  is the marginal density for  $i$ th variable. Since there exist different copula families, the choice of the copula function will influence the accuracy of the dependence modeling. The Gaussian copula is advantageous in certain applications thanks to its ability to generate high-dimensional correlated samples [7]. Archimedean copulas are more useful in scenarios which require nonlinear tail dependence modeling. However, they are generally not scalable due to being limited to the bivariate case [12], [16]. To overcome these shortcomings, we resort to vine copula next.

2) *Vine Copula*: Being a powerful tool in simulating high-dimensional correlated samples with various types of tail-dependence structures involved, vine copula is known for its capability of decomposing a multivariate density function into a cascade of bivariate pair copulae [12], [17]. Starting from the factorization on the joint density function, we get

$$\begin{aligned} f_{\mathbf{X}}(\mathbf{x}) &= f_p(x_p) \cdot f_{p-1|p}(x_{p-1}|x_p) \cdot f_{p-2|p-1,p}(x_{p-2}|x_{p-1}, x_p) \\ &\quad \cdots \cdot f_{1|2,\dots,p}(x_1|x_2, \dots, x_p) \\ &= \prod_{i=1}^p f_{i|i+1,\dots,p}(x_i|x_{i+1}, \dots, x_p). \end{aligned} \quad (4)$$

Based on the property that all the conditioned pdfs in (4) can be further transformed into the product of only bivariate copulae and one-dimensional density, e.g.,  $f_{2|1}(x_2|x_1) = c_{2|1}(F_2, F_1) \cdot f_2(x_2)$ ,  $f_{3|1,2}(x_3|x_1, x_2) = c_{3,2|1}(F_{3|1}, F_{2|1}) \cdot c_{3,1}(F_3, F_1) \cdot f_3(x_3)$ , and so on for the higher-dimensional cases [12], it is easy to infer that the joint density,  $f_{\mathbf{X}}$ , can be decomposed into a form that involves only bivariate copulae and marginal densities. Let us take  $p = 4$  as an example. Specifically,  $f_{\mathbf{X}}(x_1, x_2, x_3, x_4)$  can be decomposed as

$$\begin{aligned} f_{\mathbf{X}}(x_1, x_2, x_3, x_4) &= f_1(x_1) \cdot f_2(x_2) \cdot f_3(x_3) \cdot f_4(x_4) \cdot c_{1,2}(F_1(x_1), F_2(x_2)) \\ &\quad \cdot c_{2,3}(F_2(x_2), F_3(x_3)) \cdot c_{3,4}(F_3(x_3), F_4(x_4)) \\ &\quad \cdot c_{1,3|2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)) \\ &\quad \cdot c_{2,4|3}(F_{2|3}(x_2|x_3), F_{4|3}(x_4|x_3)) \\ &\quad \cdot c_{1,4|2,3}(F_{1|2,3}(x_1|x_2, x_3), F_{4|2,3}(x_4|x_2, x_3)). \end{aligned} \quad (5)$$

The decomposition enables us to use multiple bivariate copulae to precisely describe the high-dimensional data structure.

However, it is worth pointing out that the order of pairwise conditioning on (4) and (5) is not unique. Thus, we need a systematic way to decompose it and provide a unique solution. Two popular choices are the canonical vine (C-vine) and the drawable vine (D-vine). Both of them make use of a graphical tool to facilitate their decomposition into a set of cascade copula densities forming  $p - 1$  trees. For the C-vine copula, a  $p$ -dimensional joint density is decomposed as

$$f_{\mathbf{X}}(x_1, \dots, x_p) = \prod_{i=1}^p f_i(x_i) \prod_{j=1}^{p-1} \prod_{i=1}^{p-j} c_{j,j+i|1,\dots,j-1}. \quad (6)$$

Similarly,  $f_{\mathbf{X}}$  is decomposed via the D-vine copula as

$$f_{\mathbf{X}}(x_1, \dots, x_p) = \prod_{i=1}^p f_i(x_i) \prod_{j=1}^{p-1} \prod_{i=1}^{p-j} c_{i,j+i|i+1,\dots,i+j-1}. \quad (7)$$

Here,  $c_{j,j+i|1,\dots,j-1}$  is short for  $c_{j,j+i|1,\dots,j-1}(F_j(x_j|x_1,\dots,x_{j-1}), F(x_{j+i}|x_1,\dots,x_{j-1}))$  and  $c_{i,j+i|i+1,\dots,i+j-1}$  is short for  $c_{i,j+i|i+1,\dots,i+j-1}(F(x_i|x_{i+1}, \dots, x_{i+j-1}), F(x_{j+i}|x_{i+1}, \dots, x_{i+j-1}))$ . A simple graph demo for a 4-dimensional C-vine and D-vine is displayed in Fig. 1. For more details about their descriptions and implementations, the reader is referred to [12]. It is also easy to see that (5) is obtained from the D-vine copula. Using this vine-copula technique, we are able to generate the correlated samples that reflect the precise dependent structures of the model inputs such as loads and renewables.

### IV. REDUCED-ORDER MODELING

In this section, we present a nonparametric, reduced-order modeling technique using GPE.

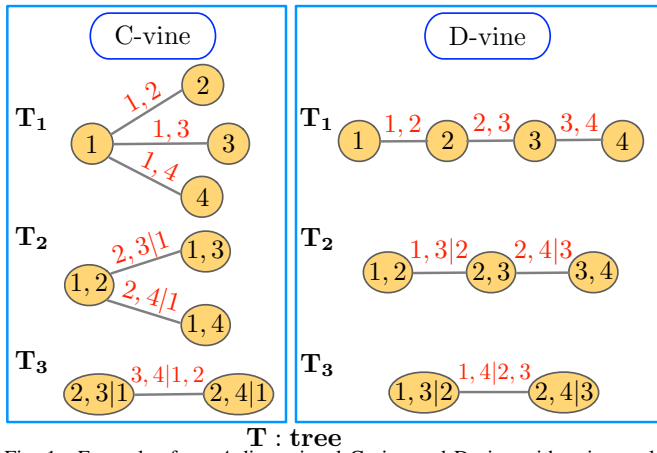


Fig. 1. Examples for a 4-dimensional C-vine and D-vine with pair copulae (red).

### A. Problem Description

Let us first formulate the probabilistic load-margin assessment problem in the GPE framework. Here, the CPF 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  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ . Accordingly, its evaluated model output  $f(\mathbf{x})$ , i.e., load margin, also becomes random, and has its corresponding  $n$  realizations denoted by  $\{f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_n)\}$ . If we assume that the model output is a realization of a Gaussian process, then the finite collection,  $\{f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_n)\}$ , of the random variables,  $f(\mathbf{x})$ , will follow a joint multivariate normal probability distribution as

$$\begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_n) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix} \right). \quad (8)$$

Here, let us denote  $m(\cdot)$  as the mean function and  $k(\cdot, \cdot)$  as a kernel function that represents the covariance function. Then, (8) can be simplified as

$$\mathbf{f}(\mathbf{X}) | \mathbf{X} \sim \mathcal{N}(\mathbf{m}(\mathbf{X}), \mathbf{k}(\mathbf{X}, \mathbf{X})), \quad (9)$$

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

Now, if an independent and identically distributed (i.i.d.) Gaussian noise  $\varepsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n)$  (where  $\mathbf{I}_n$  and  $\sigma^2$  are an  $n$ -dimensional identity matrix and the variance, respectively) is considered in the system output,  $\mathbf{f}(\mathbf{X})$ , the observations  $\mathbf{Y}$  will be expressed as

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

Note that  $\varepsilon$  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.

### 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

$$\mathbf{y}(\mathbf{x}) | \mathbf{x} \sim \mathcal{N}(\mathbf{m}(\mathbf{x}), \mathbf{k}(\mathbf{x}, \mathbf{x}) + \sigma^2 \mathbf{I}_{n_x}). \quad (11)$$

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

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

where  $\mathbf{K}_{11} = \mathbf{k}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_n$ ;  $\mathbf{K}_{12} = \mathbf{k}(\mathbf{X}, \mathbf{x})$ ;  $\mathbf{K}_{21} = \mathbf{k}(\mathbf{x}, \mathbf{X})$ ; and  $\mathbf{K}_{22} = \mathbf{k}(\mathbf{x}, \mathbf{x}) + \sigma^2 \mathbf{I}_{n_x}$ .

Now, using the rules of the conditioned Gaussian distribution [18], 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

$$\mathbf{y}(\mathbf{x}) | \mathbf{x}, \mathbf{Y}, \mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\Sigma}(\mathbf{x})), \quad (13)$$

where

$$\boldsymbol{\mu}(\mathbf{x}) = \mathbf{m}(\mathbf{x}) + \mathbf{K}_{21} \mathbf{K}_{11}^{-1} (\mathbf{Y} - \mathbf{m}(\mathbf{X})), \quad (14)$$

$$\boldsymbol{\Sigma}(\mathbf{x}) = \mathbf{K}_{22} - \mathbf{K}_{21} \mathbf{K}_{11}^{-1} \mathbf{K}_{12}. \quad (15)$$

To this point, the general form of the GPE has been derived. On one hand, we can directly use (14) as a surrogate model (a.k.a. the response surface or reduced-order model) to very closely capture the behavior of the nonlinear CPF model while being computationally inexpensive to evaluate. On the other hand, we may use (15) to quantify the uncertainty of the surrogate itself. In this paper, we only need to use (14) as a surrogate model.

### C. Mean and Covariance Functions

Let us describe the mean function  $m(\cdot)$  and the covariance function represented via the kernel  $k(\cdot, \cdot)$  that characterizes the GPE. The mean function models the prior belief about the existence of a systematic trend expressed as

$$\mathbf{m}(\mathbf{x}, \boldsymbol{\beta}) = \mathbf{H}(\mathbf{x}) \boldsymbol{\beta}. \quad (16)$$

Here,  $\mathbf{H}(\mathbf{x})$  can be any set of basis functions. For example, let  $\mathbf{x}_i = [x_{i1}, \dots, x_{ip}]$  be the  $i$ th sample, where  $i = 1, 2, \dots, n$ , wherein  $x_{ik}$  represents its  $k$ th element, where  $k = 1, 2, \dots, p$ . For instance,  $\mathbf{H}(\mathbf{x}_i) = 1$  is a constant basis;  $\mathbf{H}(\mathbf{x}_i) = [1, x_{i1}, \dots, x_{ip}]$  is a linear basis;  $\mathbf{H}(\mathbf{x}_i) = [1, x_{i1}, \dots, x_{ip}, x_{i1}^2, \dots, x_{ip}^2]$  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 include the square exponential kernel ( $k_{SE}$ ), the exponential kernel ( $k_E$ ), the rational quadratic kernel ( $k_{RQ}$ ), and the Martin 3/2 kernel ( $k_{3/2}$ ); they are listed in Table I. As for the parameters of a kernel function, they are defined as follows:  $\tau$  and  $\ell_k$  are the hyperparameters defined in the positive real line;  $\sigma^2$

and  $\ell_k$  correspond to the order of the magnitude and the speed of variation in the  $k$ th input dimension, respectively. Let  $\boldsymbol{\theta} = [\tau, \ell_1, \dots, \ell_p]$  contain the hyperparameters of the covariance function, i.e.,

$$k(\mathbf{x}_i, \mathbf{x}_j | \boldsymbol{\theta}) = \text{Cov}(\mathbf{x}_i, \mathbf{x}_j | \boldsymbol{\theta}). \quad (17)$$

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

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

## V. PROBABILISTIC LOAD-MARGIN ASSESSMENT

Here, we illustrate the steps for conducting the probabilistic load-margin assessment using the GPE.

1) *Training Sample Generation*: In order to acquire the GPE-based surrogate described in (14), 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 CPF model,  $f(\cdot)$ . To avoid long training time of the GPE,  $n$  should be small. To meet this requirement, the Latin hypercube sampling is typically chosen. It generates near-random samples and, therefore, has a faster convergence rate than the MC sampling, which generates pure random samples. This is especially true in our case since  $n$  needs to be small [19] and, therefore, the MC sampling based on the central limit theorem is not suggested here. Note that the Latin hypercube sampling generates i.i.d. samples while the correlation between renewable energy generations are inevitable. Therefore, we need to use the aforementioned vine copula to further transform these i.i.d. samples into the correlated ones to improve the training performances.

2) *GPE Construction*: With  $(\mathbf{Y}, \mathbf{X})$ , we can estimate the hyperparameters  $\boldsymbol{\eta}$  in the GPE. Following Gelman *et al.* [20], we choose to adopt the Gaussian maximum likelihood estimator (MLE) since it is the most efficient estimator under a Gaussian distribution, which is followed by the calculated residuals, and it is easy to compute. First, to indicate the hyperparameters, let us rewrite (10) as

$$\mathbf{Y} | \mathbf{X}, \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{m}(\mathbf{X}), \mathbf{k}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_n). \quad (18)$$

Then, using the Gaussian MLE, we obtain

$$\hat{\boldsymbol{\eta}} = \left(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}}, \hat{\sigma}^2\right) = \arg \max_{\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2} \log P(\mathbf{Y} | \mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2). \quad (19)$$

Using (16)–(18) and using  $\mathbf{H}$  instead of  $\mathbf{H}(\mathbf{x})$  for simplicity, the marginal log-likelihood can be expressed as

$$\begin{aligned} & \log P(\mathbf{Y} | \mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2) \\ &= -\frac{1}{2} (\mathbf{Y} - \mathbf{H}\boldsymbol{\beta})^\top [\mathbf{k}(\mathbf{X}, \mathbf{X} | \boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n]^{-1} (\mathbf{Y} - \mathbf{H}\boldsymbol{\beta}) \\ & \quad - \frac{n}{2} \log 2\pi - \frac{1}{2} \log |\mathbf{k}(\mathbf{X}, \mathbf{X} | \boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n|, \end{aligned} \quad (20)$$

which implies that the Gaussian MLE of  $\boldsymbol{\beta}$  conditioned on  $\boldsymbol{\theta}$  and  $\sigma^2$  is a weighted least-squares estimator given by

$$(\boldsymbol{\beta}, \sigma^2) = \left[ \mathbf{H}^\top [\mathbf{k}(\mathbf{X}, \mathbf{X} | \boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n]^{-1} \mathbf{H} \right]^{-1} \mathbf{H}^\top [\mathbf{k}(\mathbf{X}, \mathbf{X} | \boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n]^{-1} \mathbf{Y}. \quad (21)$$

Since  $\hat{\boldsymbol{\beta}}$  is a function of  $(\hat{\boldsymbol{\theta}}, \hat{\sigma}^2)$ , let us insert (21) into (20) to reduce the number of the hyperparameters. Then, (19) is further simplified as

$$(\hat{\boldsymbol{\theta}}, \hat{\sigma}^2) = \arg \max_{\boldsymbol{\theta}, \sigma^2} \log P(\mathbf{Y} | \mathbf{X}, \hat{\boldsymbol{\beta}}(\boldsymbol{\theta}, \sigma^2), \boldsymbol{\theta}, \sigma^2). \quad (22)$$

Now, we only need to estimate the hyperparameters  $(\hat{\boldsymbol{\theta}}, \hat{\sigma}^2)$  instead of  $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}}, \hat{\sigma}^2)$ . Then, we utilize a gradient-based optimizer to solve this optimization as described in [21]. Once  $\hat{\boldsymbol{\eta}}$  is obtained, the GPE model is fully constructed. More details can be found in [20].

3) *Sample Evaluation*: Now, we can execute an MC sampling procedure to generate a large amount of samples and transform them into the correlated ones through a vine copula. These large amount of samples can be evaluated through the GPE-based surrogate expressed in (14) at almost no computational cost. Finally, the pdf of the system response can be obtained.

## VI. SIMULATION RESULTS

A case study is conducted using the IEEE 57-bus system [22]. The algorithms are tested with MATPOWER package using MATLAB<sup>®</sup> R2018a version on a desktop with 3.50-GHz Intel<sup>®</sup> Xeon(R) CPU E5-1650 v2 processors and a 32 GB of main memory.

The uncertain inputs include the variations in the loads as well as in the wind 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. Four wind farms are connected to Buses 16, 17, 47, and 48 whose generation profiles follow the Weibull distribution with the shape and scale parameter set as  $\{2.06, 7.41\}$ ,  $\{2.1, 7.2\}$ ,  $\{2.06, 7.41\}$ , and  $\{2.3, 7.2\}$ , respectively. To reflect the correlation between these five uncertain inputs, we use the D-vine copula to generate these samples since Becker claims that the latter provides the highest accuracy when simulating wind power generations [13]. Furthermore, for our problem with five-dimensional inputs, we will have  $p(p-1)/2 = 10$  pair-copulae. Although their structure can be obtained from the D-vine, we still need to estimate the copula family for every pair copula (e.g., Frank, Gumbel), and the parameters in every copula. The copula family and copula parameters can be estimated from real data sets using a maximum likelihood estimator as described in [12]. Here, for simplicity, we assume we know the copula family and the copula parameter for each pair-copula. As suggested by Becker [13], the predominant copula family is the Frank copula for wind-data dependence simulation. Therefore, we assume that we have 7 Frank copulae, 2 Gaussian copulae, and 1 Gumbel copula. The simulated samples for the four wind farms are displayed in Fig. 2. It shows that both the upper-tail and lower-tail dependence can be simulated using the D-vine copula.



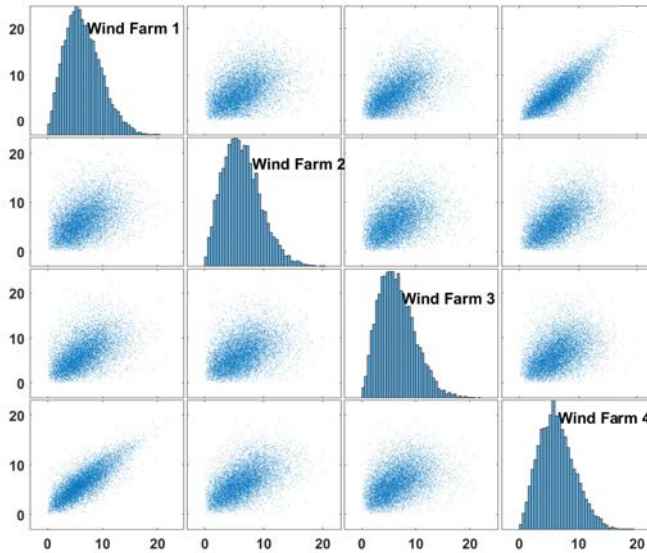


Fig. 2. 1-d histograms and scatter plots for wind farm samples (in MVA).

Let us choose the load margin at Bus 25 as the target quantity of interest. The simulation results obtained with the MC and the GPE methods are provided in Fig. 3. The simulation results obtained with the MC method with 10,000 samples are used as a benchmark to validate the GPE-based method. It can be seen that with only 15 training samples, the GPE method with a pure quadratic basis functions can provide highly accurate simulation results under different kernel functions, such as  $k_{SE}$  and  $k_{3/2}$ . The proposed GPE-based method significantly reduces the computation time over the traditional MC method without any loss of accuracy.

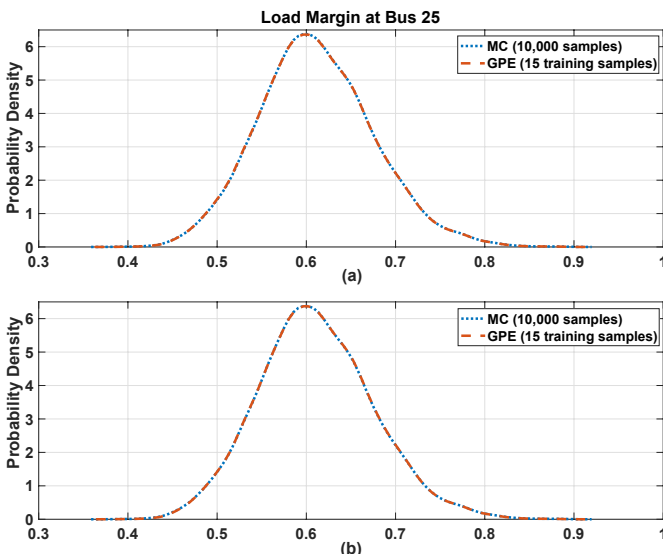


Fig. 3. Probability density plots for load margin at Bus 25 using GPE (a) with a pure quadratic basis and  $k_{3/2}$  kernel, and (b) with a pure quadratic basis and  $k_{SE}$  kernel.

## VII. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel GPE-based method for the PPF analysis. The GPE serves as a reduced-order model for the nonlinear CPF that enables an evaluation of the time-consuming CPF solver at the sampled values with a negligible computational cost. The simulation results reveal that the proposed method exhibits an impressive performance

as compared to the traditional MC method. As a future work, we will use real data to estimate the copula family and copula parameters of the vine copula and will attempt to improve the performance of the proposed method when it is applied to very-large-scale power systems.

## REFERENCES

- [1] M. Fan, V. Vittal, G. T. Heydt, and R. Ayyanar, "Probabilistic power flow analysis with generation dispatch including photovoltaic resources," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1797–1805, May 2013.
- [2] Y. Xu, M. Korkali, L. Mili, X. Chen, and L. Min, "Risk assessment of rare events in probabilistic power flow via hybrid multi-surrogate method," *IEEE Trans. Smart Grid*, 2019. [Online]. Available: <https://doi.org/10.1109/TSG.2019.2940928>
- [3] 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.
- [4] 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.
- [5] Z. Hu *et al.*, "Uncertainty quantification in stochastic economic dispatch using Gaussian process emulation," *arXiv:1909.09266*, Sep. 2019.
- [6] E. Haesen, C. Bastiaensen, J. Driesen, and R. Belmans, "A probabilistic formulation of load margins in power systems with stochastic generation," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 951–958, May 2009.
- [7] X. Xu, Z. Yan, M. Shahidehpour, H. Wang, and S. Chen, "Power system voltage stability evaluation considering renewable energy with correlated variabilities," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3236–3245, May 2018.
- [8] A. M. Leite da Silva, I. Coutinho, A. Z. De Souza, R. Prada, and A. Rei, "Voltage collapse risk assessment," *Electr. Power Syst. Res.*, vol. 54, no. 3, pp. 221–227, Jun. 2000.
- [9] 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.
- [10] I. Konstantelos *et al.*, "Using vine copulas to generate representative system states for machine learning," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 225–235, Jan. 2019.
- [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] M. Jan-Frederik and M. Scherer, *Simulating Copulas: Stochastic Models, Sampling Algorithms, and Applications*, 2nd ed. Singapore: World Scientific, 2017.
- [13] R. Becker, "Generation of time-coupled wind power infeed scenarios using pair-copula construction," *IEEE Trans. Sustainable Energy*, vol. 9, no. 3, pp. 1298–1306, Jul. 2018.
- [14] V. Ajarapu, *Computational Techniques for Voltage Stability Assessment and Control*. Springer Science & Business Media, 2007.
- [15] R. B. Nelsen, *An Introduction to Copulas*. Springer Science & Business Media, 2007.
- [16] Z. Wang, W. Wang, C. Liu, Z. Wang, and Y. Hou, "Probabilistic forecast for multiple wind farms based on regular vine copulas," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 578–589, Jan. 2018.
- [17] H. Joe, "Families of  $m$ -variate distributions with given margins and  $m(m-1)/2$  bivariate dependence parameters," *Distributions with Fixed Marginals and Related Topics*, vol. 28, pp. 120–141, 1996.
- [18] M. L. Eaton, *Multivariate Statistics: A Vector Space Approach*. New York, NY, USA: John Wiley & Sons, 1983.
- [19] 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.
- [20] A. Gelman *et al.*, *Bayesian Data Analysis*, 3rd ed. Boca Raton, FL, USA: Chapman & Hall, 2014.
- [21] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
- [22] University of Washington, Power Systems Test Case Archive. (accessed: 10/2019). [Online]. Available: <https://labs.ece.uw.edu/pstca/>