International Journal for Uncertainty Quantification, 2 (4): 413–439 (2012)

ITERATIVE METHODS FOR SCALABLE UNCERTAINTY QUANTIFICATION IN COMPLEX NETWORKS

Amit Surana,[*] *Tuhin Sahai, & Andrzej Banaszuk*

United Technologies Research Center, East Hartford, Connecticut 06118, USA

Original Manuscript Submitted: 05/10/2011; Final Draft Received: 24/01/2012

In this paper we address the problem of uncertainty management for robust design, and verification of large dynamic networks whose performance is affected by an equally large number of uncertain parameters. Many such networks (e.g., power, thermal, and communication networks) are often composed of weakly interacting subnetworks. We propose intrusive and nonintrusive iterative schemes that exploit such weak interconnections to overcome the dimensionality curse associated with traditional uncertainty quantification methods (e.g., generalized polynomial chaos, probabilistic collocation) and accelerate uncertainty propagation in systems with a large number of uncertain parameters. This approach relies on integrating graph theoretic methods and waveform relaxation with generalized polynomial chaos, and probabilistic collocation, rendering these techniques scalable. We introduce an approximate Galerkin projection that based on the results of graph decomposition computes "strong" and "weak" influence of parameters on states. An appropriate order of expansion, in terms of the parameters, is then selected for the various states. We analyze convergence properties of this scheme and illustrate it in several examples.

KEY WORDS: *collocation, polynomial chaos, graph decomposition, waveform relaxation*

1. INTRODUCTION

The issue of management of uncertainty for robust system operation is of interest in a large family of complex networked systems. Examples include power, thermal, and communication networks which arise in several instances such as more electric aircrafts, integrated building systems, and sensor networks. Such systems typically involve a large number of heterogeneous, connected components, whose dynamics is affected by possibly an equally large number of uncertain parameters and disturbances.

Uncertainty quantification (UQ) methods provide means of calculating probability distribution of system outputs, given the probability distribution of input parameters. Outputs of interest could include, for example, latency in communication network, power quality and stability of power networks, and energy usage in thermal networks. The standard UQ methods such as Monte Carlo (MC) [1] exhibit poor convergence rates whereas others such as quasi Monte Carlo (QMC) [2, 3], generalized polynomial chaos (gPC) [4], and the associated probabilistic collocation method (PCM) [5] suffer from the curse of dimensionality (in parameter space), and become practically infeasible when applied to a network as a whole. Improving these techniques to alleviate the curse of dimensionality is an active area of current research (see [6] and references therein): notable methods include the sparse-grid collocation method [7, 8] and analysis of variance (ANOVA) decomposition [9] for sensitivity analysis and dimensional reduction of the uncertain parametric space. However, none of such extension exploits the underlying structure and dynamics of the networked systems. In fact, many networks of interest (e.g., power, thermal, and communication networks) are often composed of weakly interacting subsystems. As a result, it is plausible to simplify and accelerate the simulation, analysis, and uncertainty propagation in such systems by suitably decomposing them. For example, authors in [10, 11] used graph decomposition to facilitate stability and robustness analysis of large-scale interconnected dynamical sys-

[*] Correspond to Amit Surana, E-mail: surana@utrc.utc.com

tems. Mezic et al. [12] used graph decomposition in conjunction with Perron Frobenius operator theory to simplify the invariant measure computation and uncertainty quantification, for a particular class of networks. While these approaches exploit the underlying structure of the system, they do not take advantage of the weakly coupled dynamics of the subsystems.

In this paper, we propose an iterative UQ approach that exploits the weak interactions among subsystems in a networked system to overcome the dimensionality curse associated with traditional UQ methods. We refer to this approach as probabilistic waveform relaxation (PWR), and propose both intrusive and nonintrusive forms of PWR. PWR relies on integrating graph decomposition techniques and waveform relaxation scheme, with gPC and PCM. Graph decomposition to identify weakly interacting subsystems can be realized by spectral graph theoretic techniques [13, 14]. Waveform relaxation [15] (WR), a parallelizable iterative method exploits this decomposition and evolves each subsystem forward in time independently but coupled with the other subsystems through their solutions from the previous iteration. In the intrusive PWR, the subsystems obtained from decomposing the original system are used to impose a decomposition on system obtained by Galerkin projection based on the gPC expansion. Further, the weak interactions are used to discard terms which are expected to be insignificant in the gPC expansion, leading to what we call an approximate Galerkin projected (AGP) system. We then propose to apply WR relaxation on the decomposed AGP system to accelerate the UQ computation. In the nonintrusive form of PWR, rather than deriving the AGP system, one works directly with subsystems obtained from decomposing the original system. At each waveform relaxation iteration we propose to apply PCM at the subsystem level, and use gPC to propagate the uncertainty among the subsystems. Bevause UQ methods are applied to relatively simpler subsystems which typically involve a few parameters, this renders a scalable nonintrusive iterative approach to UQ. We prove convergence of the PWR approach under very general conditions. Note that spectral graph decomposition can be done completely in a distributed fashion using a recently developed wave equation-based clustering method [16]. Moreover, one can further exploit timescale separation in the system to accelerate WR using an adaptive form of WR [17]. PWR when combined with wave equation-based distributed clustering and adaptive WR can lead to a highly scalable and computationally efficient approach to UQ in complex networks. Note that to the best of the authors' knowledge, it is the first attempt to extend polynomial chaos-based methods to parallel environments such that every step in the process, from clustering to the simulations, is completely decentralized.

This paper is organized in six sections. In Section 2 we give the mathematical preliminaries for setting up the UQ problem for networked dynamical systems, and present an overview of gPC and PCM techniques. In Section 3 we discuss graph decomposition and waveform relaxation methods, which form basic ingredients of PWR. Here we also describe adaptive WR and wave equation based distributed graph decomposition techniques. We introduce the intrusive and nonintrusive PWR in Section 4 through a simple example, and then describe these methods in a more general setting. We also prove convergence of PWR, and analyze the scalability of the method. In Section 5 we illustrate the intrusive and nonintrusive PWR in several examples. Finally, in Section 6 we summarize the main results of this paper, and present some future research directions.

2. UNCERTAINTY QUANTIFICATION IN NETWORKED SYSTEMS

Consider a nonlinear system described by a system of random differential equation

$$\begin{aligned} \dot{x}_1 &= f_1(\mathbf{x}, \xi_1, t), \\ &\vdots \\ \dot{x}_n &= f_n(\mathbf{x}, \xi_n, t), \end{aligned} \tag{1}$$

where $\mathbf{f} = (f_1, f_2, ..., f_n) \in \mathbb{R}^n$ is a smooth vector field, $\mathbf{x} = (x_1, x_2, ..., x_n) \in \mathbb{R}^n$ are state variables, $\xi_i \in R^{p_i}$ is a vector of random variables affecting the ith system. Let $\xi = (\xi_1^T, ..., \xi_n^T)^T \in \mathbb{R}^p$ be the $p = \sum_{i=1}^n p_i$ dimensional random vector of uncertain parameters affecting the complete system. The solution to the initial value problem $\mathbf{x}(t_0) = \mathbf{x}_0$ will be denoted by $\mathbf{x}(t; \xi)$, where for brevity we have suppressed the dependence of solution on initial time t_0 and initial condition $\mathbf{x}_0$. We shall assume that the system (1) is Lipschitz

$$||\mathbf{f}(\mathbf{x}_1, \xi, t) - \mathbf{f}(\mathbf{x}_2, \xi, t)|| \leq L(\xi)||\mathbf{x}_1 - \mathbf{x}_2||, \tag{2}$$

where the Lipschitz constant $L(\xi)$ depends on the random parameter vector and $||\cdot||$ is a Euclidean norm. We will assume that $\sup_{\xi\in\mathbb{R}^p} L(\xi) = L < \infty$.

Let us also define a set of quantities

$$\mathbf{z} = (z_1, z_2, ..., z_d) = G(\mathbf{x}) = [g_1(\mathbf{x}), ..., g_d(\mathbf{x})], \tag{3}$$

as observables or quantities of interests. The goal is to numerically establish the effect of input uncertainty of ξ on output observables $\mathbf{z}$. Naturally, the solution for system (1) and the observables (3) are functions of the same set of random variables ξ, i.e.,

$$\mathbf{x} = \mathbf{x}(t;\xi), \qquad \mathbf{z} = \mathbf{z}(t,\xi) = G(\mathbf{x}). \tag{4}$$

In what follows we will adopt a probabilistic framework and model $\xi = (\xi_1, \xi_2, ..., \xi_p)$ as a p-variate random vector with independent components in the probability space $(\Omega, \mathcal{A}, \mathcal{P})$, whose event space is Ω and is equipped with σ-algebra $\mathcal{A}$ and probability measure $\mathcal{P}$. Throughout this paper, we will assume that the parameters $\Sigma = \{\xi_1, \xi_2, ..., \xi_p\}$ are mutually independent of each other. Let $\rho_i : \Gamma_i \rightarrow \mathbb{R}^+$ be the probability density of the random variable $\xi_i(\omega)$, with $\Gamma_i = \xi_i(\Omega) \subset \mathbb{R}$ being its image. Then, the joint probability density of any random parameter subset $\Lambda = \{\xi_{i_1}, \xi_{i_2}, ..., \xi_{i_m}\} \subset \Sigma$ is given by

$$\rho_\Lambda(\xi_{i_1}, ..., \xi_{i_m}) = \prod_{j=1}^{|\Lambda|} \rho_{i_j}(\xi_{i_j}), \quad \forall(\xi_{i_1}, ..., \xi_{i_m}) \in \Gamma_\Lambda, \tag{5}$$

with a support

$$\Gamma_\Lambda = \prod_{j=1}^{|\Lambda|} \Gamma_{i_j} \subset \mathbb{R}^{|\Lambda|}, \tag{6}$$

where $|\cdot|$ denotes the cardinality of the set. Without loss of generality we will assume that $\Gamma_i = [-1 \;\; 1], i = 1, ..., p$.

Remark 2.1. While throughout the paper we will work with ordinary differential equation (ODE) (1) with parametric uncertainty, the PWR framework developed in this paper can be naturally extended to deal with (1) system of differential algebraic equations (DAEs), and (2) time-varying uncertainty. Both these extensions are illustrated through examples in Section 5.

2.1 Uncertainty Quantification Methods

In this section, we describe two inter-related UQ approaches: generalized polynomial chaos and the probabilistic collocation method. The gPC is an intrusive approach which requires explicit access to system equations (1), while PCM is a related sampling-based nonintrusive [and hence treats the system (1) as a black box] way of implementing gPC.

2.1.1 Generalized Polynomial Chaos

In the finite dimensional random space Γ_Σ defined in (6), the gPC expansion seeks to approximate a random process via orthogonal polynomials of random variables. Let us define one-dimensional orthogonal polynomial space associated with each random variable $\xi_k, k = 1, ..., p$ as

$$W^{k,d_k} \equiv \{v : \Gamma_k \rightarrow \mathbb{R} : v \in \text{span}\{\psi_i(\xi_k)\}_{i=0}^{d_k}\}, \tag{7}$$

where $\{\psi_i(\xi_k)\}_{i=0}^{d_k}$ denotes the orthonormal polynomial basis from the so-called Wiener-Askey polynomial chaos [4]. The Askey scheme of polynomials contains various classes of orthogonal polynomials such that their associated weighting functions coincide with probability density function of the underlying random variables. The corresponding P-variate orthogonal polynomial space in Γ_Σ is defined as

$$W^{\Sigma,P} \equiv \bigotimes_{|\mathbf{d}|\in\mathbb{P}} W^{i,d_i}, \tag{8}$$

where the tensor product is over all possible combinations of the multi-index $\mathbf{d} = (d_1, d_2, ..., d_{|\Sigma|}) \in \mathbb{N}^{|\Sigma|}$ in set $\mathbb{P}$,

$$\mathbb{P} = \{\mathbf{d} \in \mathbb{N}^{|\Sigma|} : |\mathbf{d}| = \sum_{i=1}^{|\Sigma|} d_i \leq \overline{P} \quad \text{and} \quad d_i \leq P_i\} \tag{9}$$

and $P = (P_1, ..., P_{|\Sigma|})^T \in \mathbb{N}^{|\Sigma|}$ is a vector of integers which restricts the maximum order of expansion of the ith variable ξ_i to be P_i, and $\overline{P} = \max_i P_i$. Thus, $W^{\Sigma,P}$ is the space of N-variate orthonormal polynomials of total degree at most $\overline{P}$ with additional constraints on individual degrees of polynomials, and its basis functions $\Psi_i^{\Sigma,P}(\xi)$ satisfy

$$\int_{\Gamma_\Sigma} \Psi_i^{\Sigma,P}(\xi)\Psi_j^{\Sigma,P}(\xi)\rho_\Sigma(\xi)d\xi = \delta_{ij}, \tag{10}$$

for all $1 \leq i, j \leq N_\Sigma = \dim(W^{\Sigma,P})$. Note that in standard gPC all expansion orders are taken to be identical, i.e., $P_1, = P_2 ... = P_{|\Sigma|} = \overline{P}$, so that $\dim(W^{\Sigma,p}) = (\overline{P} + |\Sigma|)!/\overline{P}!|\Sigma|!$. We will however take advantage of an adaptive expansion, a notion which will be fully developed in Section 4.3.

The major advantage of applying the gPC is that a random differential equation can be transformed into a system of deterministic equations. A typical approach is to employ a stochastic Galerkin projection, in which all the state variables are expanded in polynomial chaos basis with corresponding modal coefficients $[a_{ik}(t)]$, as

$$x_k(t,\xi) \approx x_k^{\Sigma,P}(t,\xi) = \sum_{i=1}^{N_\Sigma} a_{ik}(t)\Psi_i^{\Sigma,P}(\xi), \quad k = 1, ..., n, \tag{11}$$

where the sum has been truncated to a finite order. Substituting these expansions in Eq. (1), and using the orthogonality property of polynomial chaos (10), we obtain for $k = 1, ..., n, \quad j = 1, ..., N_\Sigma$,

$$\dot{a}_{jk} = F_{jk}(\mathbf{a}, t), \tag{12}$$

a system of deterministic ODEs describing the evolution of the modal coefficients, with initial conditions

$$a_{jk}(0) = \int_{\Gamma_\Sigma} x_k(0,\xi)\Psi_j^{\Sigma,P}(\xi)\rho_\Sigma(\xi)d\xi, \tag{13}$$

and $\mathbf{a} = (a_{11}, ..., a_{N_\Sigma 1}, ..., a_{1n}, ..., a_{N_\Sigma n})^T$,

$$F_{jk}(\mathbf{a}, t) = \int_{\Gamma_\Sigma} f_k[\mathbf{x}^{\Sigma,P}(\xi, t), \xi_k, t]\Psi_j^{\Sigma,P}(\xi)\rho_\Sigma(\xi)d\xi, \tag{14}$$

with $\mathbf{x}^{\Sigma,P}(t,\xi) = [x_1^{\Sigma,P}(t,\xi), ..., x_n^{\Sigma,P}(t,\xi)]$. This system can be solved with any numerical method dealing with initial-value problems, e.g., the Runge-Kutta method. Similarly, the observable can be expanded in the gPC basis as

$$z_k(t,\xi) \approx z_k^{\Sigma,P}(t,\xi) = \sum_{i=1}^{N_\Sigma} b_{ik}(t)\Psi_i^{\Sigma,P}(\xi), \tag{15}$$

where

$$b_{jk}(t) = \int_{\Gamma_\Sigma} z_k^{\Sigma,P}(\xi)\Psi_j^{\Sigma,P}(\xi)\rho_\Sigma(\xi)d\xi \tag{16}$$

with $k = 1, ..., d$. Hence, once the solution to the system (12) has been obtained, the coefficients b_{jk} can be approximated as

$$b_{jk} \approx \int_{\Gamma_\Sigma} g_k[\mathbf{x}^{\Sigma,P}(t,\xi)]\Psi_j^{\Sigma,P}(\xi)\rho_\Sigma(\xi)d\xi. \tag{17}$$

Such Galerkin procedures have been used extensively in the literature. In many instances Galerkin projection may not be possible due to the unavailability of direct access to the system of equations (1). In many other instances such intrusive methods are not feasible even in cases when the system equations are available, because of the cost of deriving and implementing a Galerkin system within available computational tools. To circumvent this difficulty, the probabilistic collocation method has been developed.

2.1.2 Probabilistic Collocation Method

PCM is a nonintrusive approach to solving stochastic random processes with the gPC. Instead of projecting each state variable onto the polynomial chaos basis, the collocation approach evaluates the integrals of form (16) by evaluating the integrand at the roots of the appropriate basis polynomials [5]. Given a one-dimensional (1D) probability density function $\rho_j(\xi_j)$, the PCM based on Gauss quadrature rule approximates an integral of a function g with respect to density $\rho_j(\xi_j)$ as follows:

$$\int_{-1}^{1} g(\xi_j)\rho(\xi_j)d\xi_j \approx \mathcal{U}_{l_j}[g] = \sum_{k=1}^{m_{l_j}} w_{l_j k} g(r_{l_j k}), \quad j = 1, ..., p, \tag{18}$$

where, $r_{l_j k} \in C_{l_j}$ is the set of Gauss collocation points with associated weights $w_{l_j k}$, l_j is the accuracy level of quadrature formula, and m_{l_j} is the number of quadrature points corresponding to this accuracy level. Building on the 1D quadrature formula, the full grid PCM leads to following cubature rule:

$$\int_{-1}^{1}\int_{-1}^{1} ... \int_{-1}^{1} g(\xi_1, ..., \xi_p)\rho_\Sigma(\xi)d\xi \approx \mathcal{I}(l_1, ..., l_p, p)[g] = (\mathcal{U}_{l_1} \otimes \mathcal{U}_{l_2} ... \mathcal{U}_{l_p})[g] = \sum_{j_1=1}^{m_{l_1}} ... \sum_{j_p=1}^{m_{l_p}} w_{\mathbf{lj}} g(\mathbf{r_{lj}}), \tag{19}$$

where $\mathbf{r_{lj}} = (r_{l_1 j_1}, ..., r_{l_p j_p})$, with $\mathbf{l} = (l_1, ..., l_p)$, and $\mathbf{j} = (j_1, ..., j_p)$ and $w_{\mathbf{lj}} = w_{l_1 j_1} ... w_{l_n j_n}$. To compute $\mathcal{I}(\mathbf{l}, p)$ we need to evaluate the function on the full collocation grid $C(\mathbf{l}, p)$ which is given by the tensor product of 1D grids

$$\mathcal{C}(\mathbf{l}, p) = C_{l_1} \times ... \times C_{l_p}, \tag{20}$$

with the total number of collocation points being $Q = \prod_{j=1}^{p} m_{l_j}$. In this framework, therefore, for any t the approximations to the model coefficients $a_{jk}(t)$ [see Eq. (11)] and $b_{jk}(t)$ [see Eq. (15)] can be obtained as

$$a_{jk}(t) = \int_{\Gamma_\Sigma} x_k^{\Sigma,P}(t,\xi)\Psi_j^{\Sigma,P}(\xi)\rho_\Sigma(\xi)d\xi \approx \sum_{j_1=1}^{m_{l_1}} ... \sum_{j_p=1}^{m_{l_p}} w_{\mathbf{lj}} \Psi_j^{\Sigma,P}(\mathbf{r_{lj}}) x_k(t, \mathbf{r_{lj}}), \tag{21}$$

with a similar expression for $b_{jk}(t)$. Note that to compute the summations arising in (21), the solution $\mathbf{x}(t, \mathbf{r_{lj}})$ of the system (1) is required for each collocation point $\mathbf{r_{lj}}$ in the full collocation grid $C(\mathbf{l}, p)$. Thus, the simplicity of collocation framework only requires repeated runs of deterministic solvers, without explicitly requiring the projection step in gPC.

If we choose the same order of collocation points in each dimension, i.e., $m_{l_1} = m_{l_2}, ... = m_{l_p} \equiv l$, the total number of points is $Q = l^p$, and the computational cost increases rather steeply with the number of uncertain parameters p. Hence, for large systems ($n \gg 1$) with a large number of uncertain parameters ($p \gg 1$), PCM becomes computationally intensive. As discussed in the Introduction, alleviating this curse of dimensionality is an active area of current research [6]. In this paper we propose a new uncertainty quantification approach which exploits the underlying network structure and dynamics to overcome the dimensionality curse associated with PCM. The key methodologies for accomplishing this are the graph decomposition and waveform relaxation, which are discussed in subsequent sections.

3. GRAPH DECOMPOSITION AND WAVEFORM RELAXATION

3.1 Waveform Relaxation

In this section we describe the basic mathematical concept of the waveform relaxation (WR) method for iteratively solving the system of differential equations of the form (1). For purposes of discussion here, we fix the parameter values ξ in the system (1) to fixed mean values. The general structure of a WR algorithm for analyzing system (1) over a given time interval $[0, T]$ consists of two major steps: the *assignment partitioning process* and the *relaxation process* [15, 18].

Assignment-partitioning process: Let $\mathcal{N} = \{1, ..., n\}$ be the set of state indices, and $\mathcal{C}_i, i = 1, ..., m$ be a partition of $\mathcal{N}$ such that

$$\mathcal{N} = \bigcup_{i=1}^{m} \mathcal{C}_i, \qquad \mathcal{C}_i \bigcap \mathcal{C}_j = \emptyset, \forall i \neq j. \tag{22}$$

We shall represent by $\mathcal{D} : \mathcal{N} \to \mathcal{M} \equiv \{1, 2, ..., m\}$ a map which assigns the state index to its partition index, i.e., $\mathcal{D}(i) = j$, where j is such that $i \in \mathcal{C}_j$. Without loss of generality, we can rewrite Eq. (1) after the assignment-partitioning process as

$$\begin{aligned} \dot{\mathbf{y}}_1 &= \mathbf{F}_1(\mathbf{y}_1, \mathbf{d}_1(t), \Lambda_1, t) \\ &\vdots \\ \dot{\mathbf{y}}_m &= \mathbf{F}_m(\mathbf{y}_m, \mathbf{d}_m(t), \Lambda_m, t), \end{aligned} \tag{23}$$

where, for each $i = 1, ..., m$,

$$\mathbf{F}_i \equiv (f_{j_1^i}, ..., f_{j_{M_i}^i})^T, \tag{24}$$

$$\mathbf{y}_i \equiv (x_{j_1^i}, ..., x_{j_{M_i}^i})^T, \tag{25}$$

with initial condition

$$\mathbf{y}_{0i} \equiv (x_{0j_1^i}, ..., x_{0j_{M_i}^i})^T, \tag{26}$$

and

$$\Lambda_i \equiv (\xi_{j_1^i}, ..., \xi_{j_{M_i}^i})^T, \tag{27}$$

are the subvectors assigned to the ith partitioned subsystem, such that $j_k^i \in \mathcal{C}_i, \quad k = 1, ..., M_i = |\mathcal{C}_i|$, and

$$\mathbf{d}_i(t) \equiv (\mathbf{y}_{j_i}^T, ..., \mathbf{y}_{j_{N_i}}^T)^T, \tag{28}$$

is a decoupling vector, with $j_k \in \mathcal{M}_i$ and $k = 1, ..., N_i = |\mathcal{M}_i|$. Here, $\mathcal{M}_i$ is the set of indices of the partitions (or subsystems) with which the ith partition (or subsystem) interacts, and is given by

$$\mathcal{N}_i = \mathcal{M} \setminus \Im[\mathcal{D}(\mathcal{N}_i^c)], \tag{29}$$

where $\mathcal{N}_i^c = \{j \in \mathcal{N} : \partial \mathbf{F}_i / \partial x_j = \mathbf{0}\}$ and $\Im(\cdot)$ denotes the image of the map $\mathcal{D}$.

Relaxation process: The relaxation process is an iterative procedure, with the following steps:

Step 1: (Initialization of the relaxation process) Set $I = 1$ and guess an initial waveform $\{\mathbf{y}_i^0(t) : t \in [0 \quad T]\}$ such that $\mathbf{y}_i^0(0) = \mathbf{y}_{0i}$, $i = 1, ..., m$.

Step 2: (Analyzing the decomposed system at the Ith WR iteration) For each $i = 1, ..., m$, set

$$\mathbf{d}_i^I(t) = [(\mathbf{y}_{j_i}^{I-1})^T, ..., (\mathbf{y}_{j_{N_i}}^{I-1})^T]^T, \tag{30}$$

and solve the subsystem

$$\dot{\mathbf{y}}_i^I = \mathbf{F}_i[\mathbf{y}_i^I, \mathbf{d}_i^I(t), \Lambda_i, t], \tag{31}$$

over the interval $[0, T_s]$ with initial condition $\mathbf{y}_i^I(0) = \mathbf{y}_{0i}$, to obtain $\{\mathbf{y}^I(t) : t \in [0, T_s]\}$.

Step 3: Set $I = I + 1$ and go to step 2 until satisfactory convergence is achieved.

Note that unlike traditional parallel DAE solvers, WR is agnostic to the numerical scheme [15, 19]. One can partition the original system based on timescales and then use appropriate numerical schemes (with different time steps) on various subsystems. This is particularly attractive for stiff dynamic systems [15]. Moreover, competing DAE solvers require communication at every function evaluation; this can be expensive for implicit methods since function evaluations are required for every Newton iteration [19]. WR requires communication only at the end of the simulation for the entire time window. The overall data communicated in WR also tend to be smaller since one is not forced to exchange information related to every function evaluation [20]. Since sending larger messages less frequently is more efficient from a communication point of view [21], the communication overhead of WR is significantly lower [19]. For a comparison of WR to competing approaches, see [15, 19, 20]. WR has been used in applications related to neural modeling [19], communication networks [17], and power grids [22]. Note that co-simulation [23] strategies for simulating heterogeneous systems, and utilizing different solvers for various subsystems, also frequently employ WR.

The general conditions for convergence of WR for a system of differential algebraic equations (DAEs) can be found in [17, 18]. Here, we recall a result from [17] specializing it for a system of differential equations.

Proposition 3.1. *Convergence of WR for ODEs (see [18] for proof): Given that the system (1) is Lipschitz (condition 2), then for any initial piecewise continuous waveform* $\{\mathbf{y}_i^0(t) : t \in [0, T_s]\}$ *such that* $\mathbf{y}_i^0(0) = \mathbf{y}_{0i}$ *[see definition (26)],* $i = 1, ..., m$*, the WR algorithm converges to the solution of (1) with initial condition* $\mathbf{x}_0$.

A more intuitive analysis of error at each waveform iteration is described in [17]. Let $\bar{\mathbf{y}}$ be the exact solution of the differential equation (23) and define E_I to be the error of the Ith iterate, that is

$$E_I(t) = \mathbf{y}^I(t) - \bar{\mathbf{y}}(t). \tag{32}$$

As shown in [17], the error $|E_I|$ on the interval $[0, T]$ is bounded as follows:

$$|E_I(t)| \leq \frac{C^I \eta^I T^I}{I!} |E_0(t)|, \tag{33}$$

with $C = e^{\mu T}$. Here C and η are related to the Lipschitz constants of the waveform relaxation operator [17]. Note here that the $I!$ in the denominator dominates the error. Thus, with enough iterations one can make the error fall below any desired threshold. It is also evident from Eq. (33) that the error of standard waveform relaxation crucially depends on T. The longer the time interval, the greater is the number of iterations needed to bound the error below a desired tolerance. Based on this observation, a novel “adaptive” version of waveform relaxation has been developed in [17], which we refer to as adaptive waveform relaxation (AWR). The idea here is to perform waveform relaxation in “windows” of time that are picked so as to reduce I in Eq. (33). Specifically, one can pick small time intervals for computation when the solution to (1) changes significantly [implying $E_0(t)$ is large] and pick large intervals when the solution changes little [consequently $E_0(t)$ is small]. The solution from one time interval is extrapolated to the next using a standard extrapolation formula [24] and the initial error is estimated using

$$\tilde{E}_{I+1,0}(t) = \frac{\phi^{(l)}[x^{I+1}(\xi), x^{I+1}(\xi)]}{(l+1)!}\, \omega(t), \tag{34}$$

where

$$\omega(t) = (t - t_0)(t - t_1) \dots (t - t_l). \tag{35}$$

Here $t_0, t_1, \dots, t_l$ are points through which one passes the extrapolating polynomial [17]. Note that $\phi^{(l)} = d^l \phi / dt^l$ is the lth derivative of the waveform relaxation operator ϕ with respect to t (see [17, 24]). We point out to the reader that upper bounds on the window size can be placed using the memory restrictions at the various computer processors. Also, at a particular processor, one does not need to store the waveforms of all the state variables. At a processor, one only needs to store the waveforms of state variables that influence the ones that are being computed locally.

The algorithm to compute the length of the time windows is as follows:

Adaptive waveform relaxation: To compute the time interval for ΔT_{I+1}, execute the following steps:

1. Set $\Delta T_{I+1} = 2\Delta T_I$ and $\delta = (1/20)\Delta T_I$.

2. Evaluate $\tilde{E}_{I+1,0}(T_I + \Delta T_{I+1})$ using Eq. (34) to estimate $||E_{I+1,0}||$ and compute $||\hat{E}_{I+1,r}||$ with the aid of the following equation:
$$||\hat{E}_{I+1,r}|| = \frac{\left(e^{\mu\Delta T_{I+1}}\eta\Delta T_{I+1}\right)^r}{r!}||E_{I+1,0}||. \tag{36}$$

3. If $||\hat{E}_{I+1,r}|| > \varepsilon$ and $\Delta T_{I+1} > (1/2)\Delta T_I$, set $\Delta T_{I+1} = \Delta T_{I+1} - \delta$ and repeat step 2.

We define the minimal window length to be $\Delta T_{I+1} = (1/50)T$. The above procedure gives a sequence of time intervals $[0, T_1]$, $[T_1, T_2]$, $\ldots$, $[T_{\nu-1}, T_\nu]$, where $T_\nu = T$, on which WR is performed (as described in relaxation process) with an initial "guess" waveform provided by an extrapolation of the solution on the previous interval [17]. AWR is found to accelerate simulations by orders of magnitude over traditional WR [17]. In this work, we propose to use AWR for simulating the set of differential equations that arise from intrusive polynomial chaos. As mentioned before, the curse of dimensionality gives rise to a combinatorial number of equations [4] making AWR particularly attractive.

While the convergence of WR or AWR is guaranteed irrespective of how the system is decomposed in the assignment-partitioning step, the rate of convergence depends on the decomposition [17]. For a given nonlinear system, determining a decomposition that leads to an optimal rate of AWR convergence is an NP-complete problem [17]. Ideally, to minimize the number of iterations required for convergence, one would like to place strongly interacting equations/variables on a single processor, with weak interactions between the variables or equations on different processors. In Eq. (33), η is a measure of the "strength" of connection between clusters. The smaller η is, the weaker are the interactions and consequently, fewer are the required iterations for convergence. In fact, once the system has been decomposed, by bounding η one can estimate the computational effort required by AWR, thus aiding in the decision of using this approach over sampling based methods.

We show in [17] that spectral clustering [13] along with horizontal vertical decomposition [12] is a good heuristic for decomposing systems for fast convergence in WR and AWR. For a comparison of spectral clustering with other approaches see [17]. For this task, we now discuss a novel decentralized spectral clustering approach [16] that when coupled with AWR [17] provides a powerful tool for simulating large dynamic systems, making every step of the UQ approach scalable.

3.2 Graph Decomposition

The problem of partitioning the system of equations (1) into subsystems based on how they interact or are coupled to each other can be formulated as a graph decomposition problem. Given the set of states $x_1, ..., x_n$ and some notion of dependence $\overline{w}_{ij} \geq 0, i = 1, ..., n, j = 1, ..., n$ between pairs of states, an undirected graph $G = (V, E)$ can be constructed. The vertex set $V = \{1, \ldots, n\}$ in this graph represent the states x_i, and the edge set is $E \subseteq V \times V$, where a weight $\bar{w}_{ij} \geq 0$ is associated with each edge $(i, j) \in E$, and $\overline{W} = [\bar{w}_{ij}]$ is the $n \times n$ weighted adjacency matrix of $\mathcal{G}$. In order to quantify coupling strength $\bar{w}_{ij}$ between nodes or states, we propose to use
$$\bar{w}_{ij} = \frac{1}{2}[|\overline{J}_{ij}| + |\overline{J}_{ji}|], \tag{37}$$
where $\overline{J} = \{(1/T_s)\int_{t_0}^{t_0+T_s} J_{ij}[\mathbf{x}(t;\xi_m), \xi_m, t]dt\}$, is the time average of the Jacobian,
$$J(\mathbf{x}, \xi, t) = \left[\frac{\partial f_i[\mathbf{x}(t;\xi), \xi, t]}{\partial x_j}\right], \tag{38}$$
computed along the solution $\mathbf{x}(t;\xi)$ of the system (1) for nominal values of parameters ξ_m. Use of the system Jacobian for horizontal vertical graph decomposition can also be found in [12].

We will now discuss spectral clustering (see [13], a popular graph decomposition/clustering approach that allows one to partition a undirected graph given its adjacency matrix $\overline{W}$). In this method, first a (normalized) graph Laplacian is constructed as follows [14, 25, 26]:

$$L_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\bar{w}_{ij}/\sum_{\ell=1}^{N} \bar{w}_{i\ell} & \text{if } (i,j) \in E \\ 0 & \text{otherwise}\,, \end{cases} \tag{39}$$

or equivalently as $L = I - D^{-1}\overline{W}$, where D is the diagonal matrix with the row sums of $\overline{W}$. The clustering assignment/decomposition is then obtained by computing the eigenvectors/eigenvalues of L. In particular, one uses the signs of the components of the second and higher eigenvectors to partition the nodes in the graph into clusters [13]. Traditionally, one can use standard matrix algorithms for eigenvector/eigenvalue computation [27]. However, as the size of the dynamic system or network (and thus corresponding adjacency matrix) increases, the execution of these standard algorithms becomes infeasible on monolithic computing devices. To address this issue, distributed eigenvalue/eigenvector computation methods have been developed; see for example [28].

In [16], a wave equation-based distributed algorithm to partition large graphs has been developed which computes the partitions without constructing the entire adjacency matrix $\overline{W}$ of the graph [13]. In this method one “hears” clusters in the graph by computing the frequencies [using fast Fourier transform (FFT) locally at each node] at which the graph “resonates.” In particular, one can show that these “resonant frequencies” are related to the eigenvalues of the graph Laplacian L (39) and the coefficients of FFT expansion are the components of the eigenvectors. In fact, the algorithm is provably equivalent to the standard spectral clustering [13]; for details see [16].

The steps of the wave equation-based clustering algorithm are as follows. One starts by writing the local update equation at node i based on the discretized wave equation,

$$u_i(t) = 2u_i(t-1) - u_i(t-2) - c^2 \sum_{j \in \mathcal{N}_i} L_{ij}, u_j(t-1) \tag{40}$$

where $u_i(t)$ is the value of u at node i at time t and L_{ij} are the local entries of the graph Laplacian. At each node i, $u_i(0) = u_i(-1)$ is set to a random number on the interval $[0, 1]$. One then updates the value of u_i using Eq. (40) until $t = T_{\max}$ (for a discussion on how to pick $T_{\max}$ see [16]). Note that $u_i(t)$ is a scalar quantity and one only needs nearest neighbor information in Eq. (40) to compute it. One then performs a local FFT on $[u_i(1), \ldots, u_i(T_{\max})]$ and then assigns the coefficients of the peaks of the FFT to v_{ij}. Here the frequency of the jth peak is related to λ_j, the jth eigenvalue of the graph Laplacian L, and v_{ij} is the ith component of the jth eigenvector.

In [16], it has been shown that for wave speed $c < \sqrt{2}$ in Eq. (40), the above wave equation-based algorithm is stable and converges. Moreover, the algorithm converges in $O(\sqrt{\tau})$ steps, where τ is the mixing time of the Markov chain [16] associated with the graph G. The competing state-of-the-art algorithm [28] converges in $O[\tau(\log(n)^2)]$. For large graphs or datasets, $O(\sqrt{\tau})$ convergence is shown to provide orders of magnitude improvement over algorithms that converge in $O[\tau(\log(n)^2)]$. For detailed analysis and derivations related to the algorithm, we refer the reader to [16].

4. ITERATIVE UNCERTAINTY QUANTIFICATION APPROACH

In this section we discuss how gPC and PCM can be integrated with the WR scheme, extending it to a probabilistic setting. As mentioned earlier, we refer to this iterative UQ approach as PWR. Figure 1 shows the schematic of PWR framework. In the intrusive PWR, the subsystems obtained from decomposing the original system are used to impose a decomposition on the system obtained by Galerkin projection based on the gPC expansion. Further, the weak interactions are used to discard terms which are expected to be insignificant in the gPC expansion, leading to what we call an approximate Galerkin projected (AGP) system. We then propose to apply standard or adaptive WR on the decomposed AGP system to accelerate the UQ computation. In the nonintrusive form of PWR, rather than deriving the AGP system, one works directly with subsystems obtained from decomposing the original system. At each waveform

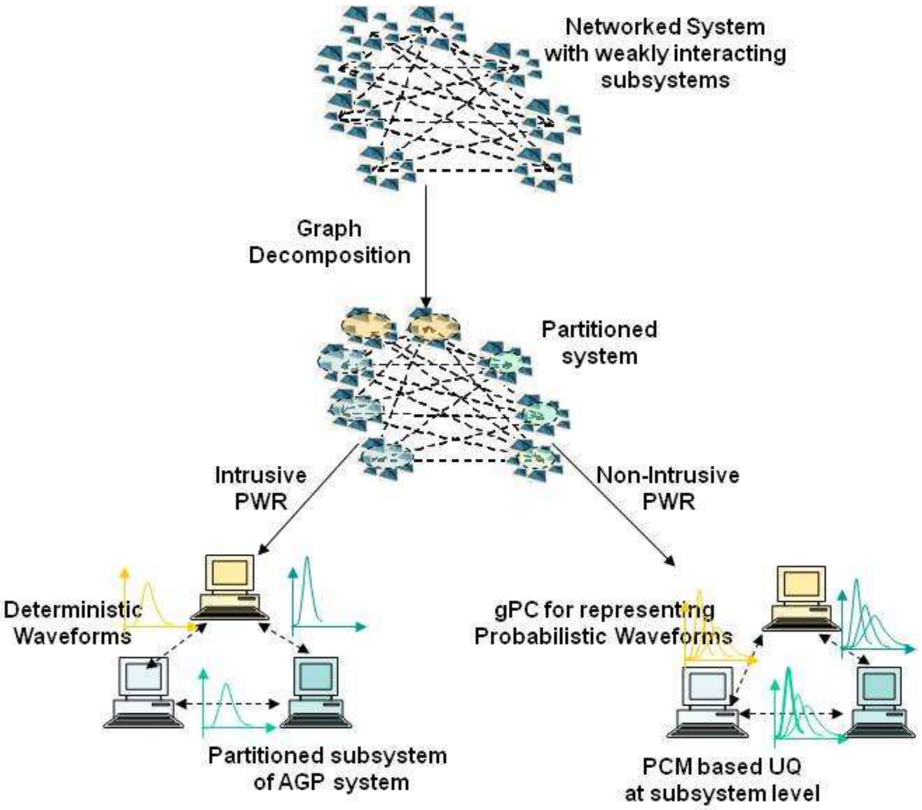

Networked System with weakly interacting subsystems
Graph Decomposition
Partitioned system
Intrusive PWR
Non-Intrusive PWR
Deterministic Waveforms
gPC for representing Probabilistic Waveforms
Partitioned subsystem of AGP system
PCM based UQ at subsystem level

FIG. 1: Schematic of intrusive (left) and nonintrusive (right) PWR.

relaxation iteration we propose to apply PCM at subsystem level and use gPC to propagate the uncertainty among the subsystems. Note that unlike intrusive PWR (where deterministic decoupling vectors or deterministic waveforms are exchanged), in nonintrusive PWR stochastic decoupling vector or probabilistic waveforms represented in gPC basis are exchanged between subsystems at each iteration.

We first describe the key technical ideas behind intrusive and nonintrusive PWR though an illustration on a simple example in Section 4.1. These notions are fully generalized later in Sections 4.2–4.4. We also prove the convergence of the PWR approach (in Section 4.5), and in Section 4.6 discuss the computational gain it offers over standard application of gPC and PCM.

4.1 Main Ideas of PWR

We illustrate the proposed PWR framework through an example of parametric uncertainty in a simple system (1). Consider the following coupled oscillator system:

$$\begin{aligned} \dot{x_1} &= f_1(x_1, x_2, \omega_1, t) = \omega_1 + K_{12}\sin(x_1 - x_2), \\ \dot{x_2} &= f_2(x_1, x_2, x_3, \omega_2, t) = \omega_2 + K_{21}\sin(x_1 - x_2) + K_{23}\sin(x_3 - x_2), \\ \dot{x_3} &= f_3(x_2, x_3, \omega_3, t) = \omega_1 + K_{32}\sin(x_2 - x_3). \end{aligned} \tag{41}$$

Here, ω_i is the uncertain angular frequency of the ith ($i = 1, 2, 3$) oscillator. We assume that parameters ω_i are mutually independent, each having probability density $\rho_i = \rho_i(\omega_i)$. The coupling matrix K

$$K = \begin{pmatrix} 0 & K_{12} & 0 \\ K_{21} & 0 & K_{23} \\ 0 & 0 & K_{32} \end{pmatrix} \tag{42}$$

is assumed deterministic with $K_{ij} = \mathcal{O}(\epsilon)$, $\epsilon \ll 1$, so that the three oscillators in (1) weakly interact with each other, i.e., the subsystem 2 weakly affects subsystem 1 and 3, and vice versa.

4.1.1 Approximate Galerkin Projection for the Simple Example

In standard gPC, states x_i are expanded in a polynomial chaos basis as

$$x_i^{\Sigma,P}(t,\omega) = \sum_{j=1}^{N_\Sigma} a_{ji}(t)\Psi_j^{\Sigma,P}(\omega), \quad i = 1,2,3, \tag{43}$$

where $\Psi_j^{\Sigma,P} \in W^{\Sigma,P}$, the P variate polynomial chaos space formed over $\Sigma = \{\omega_1, \omega_2, \omega_3\}$, and $P = (P_1, P_2, P_3)$ determines the expansion order (see Section 2.1.1 for details). Note that in this expansion (43), the system states are expanded in terms of all the random variables ω affecting the entire system. From the structure of system (1) it is clear that the first subsystem is directly affected by the parameter ω_1 and indirectly by parameter ω_2 through the the state x_2. We neglect the second-order effect of ω_3 on x_1. A similar statement holds true for subsystem 3, while subsystem 2 will be weakly influenced by ω_1 and ω_3 through states x_1 and x_3, respectively. This structure can be used to simplify the Galerkin projection as follows. For x_1 we consider the gPC expansion over $\Sigma_1 = \Lambda_1 \bigcup \Lambda_1^c$,

$$x_1^{\Sigma_1,P_1}(t,\omega_1,\omega_2) = \sum_{j=1}^{N_{\Sigma_1}} \overline{a}_{j1}(t)\Psi_j^{\Sigma_1,P_1}(\omega_1,\omega_2), \tag{44}$$

where

$$\Lambda_1 = \{\omega_1\}, \qquad \Lambda_1^c = \{\omega_2\}, \tag{45}$$

and $P_1 = (P_{11}, P_{12})$. Note that since ω_2 weakly affects x_1, the order of expansion P_{12} can be chosen to be smaller compared to P_{11}. Similarly, one can consider simplification for $x_3^{\Sigma_3,P_3}(t,\omega_1,\omega_3)$. For x_2 following similar steps, we define

$$\Lambda_2 = \{\omega_2\}, \qquad \Lambda_2^c = \{\omega_1, \omega_3\}, \tag{46}$$

and $P_2 = (P_{21}, P_{22}, P_{23})$. By similar argument, one will choose P_{21}, P_{23} much smaller than P_{22}. We also introduce the following two projections associated with the state x_2:

$$\mathcal{P}^{2,i}(x_2^{\Sigma_2,P_2}) = \sum_{j=1}^{N_{\Sigma_i}} \left\langle x_2^{\Sigma_2,P_2}, \Psi_j^{\Sigma_i,P_2} \right\rangle \Psi_j^{\Sigma_i,P_i}, \tag{47}$$

where $i = 1, 3$ and $\langle\cdot,\cdot\rangle$ is the appropriate inner product on $W^{\Sigma,P}$ (see Section 4.3 for details). With these expansions, and using standard Galerkin projection we obtain the following system of deterministic equations:

$$\dot{\overline{\mathbf{a}}} = \overline{\mathbf{F}}(\overline{\mathbf{a}}, t), \tag{48}$$

with appropriate initial conditions, where

$$\begin{aligned}
\overline{F}_{j1}(\overline{\mathbf{a}}) &= \int_{\Gamma_\Sigma} f_1[x_1^{\Sigma_1,P_1}, \mathcal{P}^{2,1}(x_2^{\Sigma_2,P_1}), \omega_1, t]\Psi_j^{\Sigma_1,P_1}(\omega)\rho(\omega)d\omega, \\
\overline{F}_{j2}(\overline{\mathbf{a}}) &= \int_{\Gamma_\Sigma} f_2[x_1^{\Sigma_1,P_1}, x_2^{\Sigma_2,P_2}, x_3^{\Sigma_3,P_3}, \omega_2, t]\Psi_j^{\Sigma_2,P_2}(\omega)\rho(\omega)d\omega, \\
\overline{F}_{j3}(\overline{\mathbf{a}}) &= \int_{\Gamma_\Sigma} f_3[\mathcal{P}^{2,3}(x_2^{\Sigma_2,P_2}), x_1^{\Sigma_1,P_1}, \omega_3, t]\Psi_j^{\Sigma_3,P_3}(\omega)\rho(\omega)d\omega,
\end{aligned}$$

and $\overline{\mathbf{a}} = (\overline{\mathbf{a}}_1, \overline{\mathbf{a}}_2, \overline{\mathbf{a}}_3)^T$, with $\mathbf{a}_i = (\overline{a}_{i1}, ..., \overline{a}_{iN_{\Sigma_i}})$ and $\overline{\mathbf{F}} = (\overline{\mathbf{F}}_1, \overline{\mathbf{F}}_2, \overline{\mathbf{F}}_3)^T$ with $\overline{\mathbf{F}}_i = (\overline{F}_{i1}, ..., \overline{F}_{1N_{\Lambda_i}})$. We will refer to (48) as an AGP system. The notion of AGP in a more general setting is described in Section 4.3.

4.1.2 Intrusive PWR Illustrated on the Simple Example

In intrusive PWR, after performing the AGP explicitly, the system (48) is decomposed as

$$\dot{\overline{a}}_{ij} = \overline{F}_{ji}(\overline{\mathbf{a}}_i, \overline{\mathbf{d}}_i, t), \tag{49}$$

where $\overline{\mathbf{d}}_1 = \mathcal{P}^{2,1}(\overline{\mathbf{a}}_2)$, $\overline{\mathbf{d}}_2 = (\overline{\mathbf{a}}_1, \overline{\mathbf{a}}_3)$ and $\overline{\mathbf{d}}_3 = \mathcal{P}^{2,1}(\overline{\mathbf{a}}_2)$ are the decoupling vectors [here we overloaded notation for $\mathcal{P}^{2,i}(\overline{\mathbf{a}}_2)$ to imply the coefficients in expansion (47)]. Note that the decomposition of system (48) is based on the decomposition of the original system (41). Adaptive or standard WR, described in Section 3.1, can then be applied to solve the decomposed system (49) iteratively. Since, the the system (49) is deterministic, deterministic waveforms or deterministic decoupling vectors $\overline{\mathbf{d}}_i, i = 1, 2, 3$ are exchanged in each WR iteration (see Fig. 1 for an illustration).

4.1.3 Nonintrusive PWR Illustrated on the Simple Example

In the nonintrusive form of PWR, rather than deriving the AGP, one works directly with subsystems obtained from decomposing the original system. The main idea here is to apply PCM at subsystem level at each PWR iteration, use gPC to represent the probabilistic waveforms, and iterate among subsystems using these waveforms. Recall that in standard PCM approach (Section 2.1.2), the coefficients $\overline{a}^i_m(t)$ are obtained by calculating the integral

$$\overline{a}_{mi}(t) = \int x_i^{\Lambda_i, P_i}(t, \omega)\Psi_m^{\Lambda_i, P_i}(\omega)\rho_{\Lambda_i}(\omega)d\omega. \tag{50}$$

The integral above is typically calculated by using a quadrature formula and repeatedly solving the ith subsystem over an appropriate collocation grid $\mathcal{C}^i(\Sigma_i) = \mathcal{C}^i(\Lambda_i) \times \mathcal{C}^i(\Lambda_i^c)$, where, $\mathcal{C}^i(\Lambda_i)$ is the collocation grid corresponding to parameters Λ_i (and let l_s be the number of grid points for each random parameter in Λ_i), $\mathcal{C}^i(\Lambda_i^c)$ is the collocation grid corresponding to parameters Λ_i^c (and let l_c be the number of grid points for each random parameter in Λ_i^c). Since the behavior of the ith subsystem is weakly affected by the parameters Λ_i^c, we can take a sparser grid in Λ_i^c dimension, i.e., $l_c < l_s$, as we took lower-order expansion for these random variables in Section 4.1.1. Below we outline key steps in nonintrusive PWR:

Step 1: (Initialization of the relaxation process with no coupling effect): Set $I = 1$; guess an initial waveform $x_i^0(t)$ consistent with initial condition. Set $\mathbf{d}_1^1 = x_2^0, \quad \mathbf{d}_2^1 = (x_1^0, x_3^0), \quad \mathbf{d}_3^1 = x_2^0$, and solve

$$\dot{x}_i^1 = f_i[x_i^1, \mathbf{d}_i^1(t), \omega_i, t], \tag{51}$$

with an initial condition $x_i^1(0) = x_i^0(0)$ on a collocation grid $\mathcal{C}^i(\Lambda_i)$. Determine the gPC expansion $x_i^{\Lambda_i, P_i, 1}(t, \cdot)$ by computing the expansion coefficients from the quadrature formula (50).

Step 2: (Initialization of the relaxation process, incorporating first level of coupling effect): Set $I = 2$ and let $\mathbf{d}_1^2 = x_2^{\Lambda_2, P_2, 1}, \quad \mathbf{d}_2^2 = (x_1^{\Lambda_1, P_1, 1}, x_3^{\Lambda_3, P_3, 1}), \quad \mathbf{d}_3^2 = x_2^{\Lambda_2, P_2, 1}$ be the *stochastic decoupling vectors*. Solve

$$\dot{x}_i^2 = f_i[x_i^2, \mathbf{d}_i^2(t, \cdot), \omega_i, t], \tag{52}$$

over a collocation grid $\mathcal{C}^i(\Sigma_i)$ to obtain $x_i^{\Sigma_i, P_i, 2}(t, \cdot)$. From now on we shall denote the solution of the ith subsystem at Ith iteration by $x_i^{\Sigma_i, P_i, I}$.

Step 3: (Analyzing the decomposed system at the Ith iteration): Set the decoupling vectors, $\mathbf{d}_1^I = \mathcal{P}^{2,1}(x_2^{\Sigma_2, P_2, I-1})$, $\mathbf{d}_2^I = (x_1^{\Sigma_1, I-1}, x_3^{\Sigma_3, P_3, I-1})$, $\mathbf{d}_3^I = \mathcal{P}^{2,3}(x_2^{\Sigma_2, P_2, I-1})$, and solve

$$\dot{x}_i^I = f_i[x_i^I, \mathbf{d}_i^I(t, \cdot), \omega_i, t], \tag{53}$$

over a collocation grid $\mathcal{C}^i(\Sigma_i)$ and obtain the expansion $x_i^{\Sigma_i, P_i, I}(t, \cdot)$.

Step 4: (Iteration): Set $I = I + 1$ and go to step 5 until satisfactory convergence has been achieved.

Note that in the above nonintrusive PWR, the decoupling vectors are stochastic and so at each iteration *probabilistic waveforms* are exchanged between subsystems (see Fig. 1 for an illustration). We next generalize the intrusive and nonintrusive PWR introduced above in the forthcoming sections.

4.2 Decomposition of Galerkin Projected System

We begin by revisiting the complete Galerkin system (12). To apply WR, recall that the first step is the assignment-partitioning (see Section 3.1). There are two possible approaches for partitioning the complete Galerkin system. One can first split the original dynamic system (1), and then use this decomposition to partition the complete Galerkin projection (12) by assigning the model coefficients in (11) for each state to the cluster to which it is assigned while decomposing system (1). As previously explained in Section 3.2, the partitioning is performed by representing the dynamic system (1) as a graph with the symmetrized time-averaged Jacobian (37) as the weighted adjacency matrix. One can then apply the wave equation-based decentralized clustering algorithm outlined in Section 3.2.

Alternatively, one can perform this decomposition directly on the complete Galerkin projection (12). Let the symmetrized time-averaged Jacobian for the resulting system (12) be

$$\tilde{w}_{ij} = \frac{1}{2}[|\tilde{J}_{ij}| + |\tilde{J}_{ji}|], \tag{54}$$

where $\tilde{J} = \{(1/T_s)\int_{t_0}^{t_0+T_s} J'_{ij}[a(t),t]dt\}$ is the time average of the Jacobian,

$$J'(\mathbf{a},t) = \left[\frac{\partial F_{ik}[\mathbf{a}(t)]}{\partial a_{ik}}\right], \tag{55}$$

computed along the solution $\mathbf{a}(t)$ of the system (12). This gives

$$J'(\mathbf{a},t) = \int_{\Gamma_\Sigma} \frac{\partial f_k[\mathbf{x}^{\Sigma,P}(\xi,t),\xi_k,t]}{\partial a_{jk}} \Psi_i^{\Sigma,P}(\xi)\mathbf{w}_\Sigma(\xi)d\xi. \tag{56}$$

Taylor expanding $f_k[\mathbf{x}(\xi,t),\xi_k,t]$ locally gives

$$J'(\mathbf{a},t) = J(\mathbf{a},t) + O(\mathbf{a}^2). \tag{57}$$

Thus, one expects to get similar results by performing clustering on the original system [in (1)] to that obtained based on complete Galerkin system (12). Because the dimensionality of system (1) is much lower than that of system (12), the first decomposition is less computationally challenging than the latter. In this work, we use the original system to determine the decomposition and use that to impose the partition of the Galerkin projection. Given the decomposition of system (12), one can use the WR or its adaptive form to simulate the system in a parallel fashion. However, before doing this one can further exploit the weak interaction between subsystems to reduce the dimensionality of the complete Galerkin system, as described in Section 4.1.1. We next describe this approximate Galerkin projection in a more general setting.

4.3 Approximate Galerkin Projection and Intrusive Probabilistic Waveform Relaxation

Recall that in the gPC expansion (11), all the system states are expanded in terms of random variables affecting the entire system. However, the ith subsystem in the decomposition [see Eq. (23)] is directly affected by the parameters Λ_i [see definition (27)] and indirectly by other parameters through the decoupling vector [see definition (28)]. We shall denote by

$$\Lambda_i^c = \bigcup_{j\in\mathcal{N}_i} \Lambda_j, \tag{58}$$

the set of parameters which indirectly affect the ith subsystem through the immediate neighbor interactions, and denote by

$$\Sigma_i = \Lambda_i \cup \Lambda_i^c, \tag{59}$$

the set of parameters that directly and indirectly (through nearest neighbor interaction) affect the ith subsystem. Under the hypothesis that the ith subsystem is dynamically weakly coupled with its nearest neighbors, uncertainty in parameters Λ_c^i will weakly influence the states in the ith subsystem through the decoupling vector, while the uncertainty in parameters $\Sigma \setminus \Sigma_i$ can be neglected. To capture this effect, consider a P-variate space (analogous to the P-variate space introduced in Section 2.1.1)

$$W^{\Lambda,P} \equiv \bigotimes_{|\mathbf{d}| \in \mathbb{P}} W^{k_i, d_{k_i}}, \tag{60}$$

formed over any random parameter subset $\Lambda = \{\xi_{i_1}, \xi_{i_2}, ..., \xi_{i_n}\} \subset \Sigma$. We denote the basis elements of $W^{\Lambda,P}$ by $\Psi_i^{\Lambda,P}, i = 1, ..., N_\Lambda$, where $N_\Lambda = \text{dim}(W^{\Lambda,P})$. Note that for any $\Lambda_1 \subset \Lambda_2 \subset \Sigma$,

$$W^{\emptyset} \subset W^{\Lambda_1,P_1} \subset W^{\Lambda_2,P_2} \subset W^{\Sigma,P}, \tag{61}$$

where $W^{\emptyset} = \{0\}$ is the P-variate space corresponding to the empty set. Also, recall that P_1, P_2 are vectors which control the expansion order in gPC expansion. The inner product on $W^{\Sigma,P}$ induces an inner product on $W^{\Lambda,P}$ as follows

$$\langle X_1(\xi), X_2(\xi) \rangle_\Lambda = \int_{\Gamma_\Lambda} X_1(\xi) X_2(\xi) \rho_\Lambda(\xi) d\xi, \tag{62}$$

for any $X_1(\xi), X_2(\xi) \in W^{\Lambda,P}$. Using this inner product, we introduce a projection operator

$$Pr_{\Lambda_1}^{\Lambda_2} : W^{\Lambda_2,P_2} \to W^{\Lambda_1,P_1}, \tag{63}$$

such that for any $X(\xi) \in W^{\Lambda_2,P_2}$

$$Pr_{\Lambda_1,P_1}^{\Lambda_2,P_2}(X)(\xi) = \sum_{i=1}^{N_{\Lambda_1}} \langle X, \Psi_i^{\Lambda_1,P_1 S} \rangle_\Lambda \Psi_i^{\Lambda_1,P_1}(\xi). \tag{64}$$

With respect to the given decomposition $\mathcal{D}$ imposed on the system (see Section 3.1), we define a projection operator $\mathcal{P}^{i,j}$ indexed by subsystem i and state x_j

$$\mathcal{P}^{i,j} \equiv \begin{cases} Pr_{\Sigma_i,P_i}^{\Sigma,P}, \quad \text{if} \quad \mathcal{D}(j) = i, \\ Pr_{\Lambda_{D(j)} \bigcup (\Lambda_{D(j)}^c \bigcap \Lambda_i), P_i}^{\Sigma,P} \quad \text{if} \quad \mathcal{D}(j) \neq i, \mathcal{N}_i \bigcap \mathcal{N}_{\mathcal{D}(j)} \neq \emptyset, \\ Pr_{\emptyset}^{\Sigma,P} \quad \text{if} \quad \mathcal{D}(j) \neq i, \mathcal{N}_i \bigcap \mathcal{N}_{\mathcal{D}(j)} = \emptyset. \end{cases}$$

Remark 4.1. For any subsystem i, since the parameters Λ_i^c weakly affect it, we can adaptively select component of vector $P_i = (P_{i1}, ..., P_{i,|\Sigma_i|})$ so that a lower-order expansion is used in components corresponding to random variables in Λ_i^c.

For any subsystem i and a vector $\mathbf{x}^{\Sigma,P}(\xi, t) = [x_{k_1}^{\Sigma,P}(\xi, t), ..., x_{k_n}^{\Sigma,P}(\xi, t)]$ [where $x_i^{\Sigma,P}(\xi, t)$ is standard gPC expansion (11)], we associate a vector valued projection operator as follows:

$$\mathcal{P}^i(\mathbf{x}^{\Sigma,P}) = [\mathcal{P}^{i,k_1}(x_{k_1}^{\Sigma,P}), ..., \mathcal{P}^{i,k_n}(x_{k_n}^{\Sigma,P})]. \tag{65}$$

In terms of these operators, for any state x_k an approximate Galerkin projected equation is defined as

$$\frac{d}{dt}\mathcal{P}^{i,k}[x_k^{\Sigma,P}(\xi, t)] = f_k\{\mathcal{P}^i[\mathbf{x}^{\Sigma,P}(\xi, t)], \xi_k, t\}, \tag{66}$$

where $i = \mathcal{D}(k)$ is the index of the subsystem to which the state k belongs. More precisely, the above system can be expressed as

$$\dot{a}_{jk}^i = \overline{F}_{jk}^i(\overline{\mathbf{a}}, t), \tag{67}$$

where $\overline{\mathbf{a}} = (\overline{a}_{11}^{\mathcal{D}(1)}, ..., \overline{a}_{N_{\Sigma_{\mathcal{D}(1)}},1}^{\mathcal{D}(1)}, ..., \overline{a}_{1n}^{\mathcal{D}(n)}, ..., \overline{a}_{N_{\Sigma_{\mathcal{D}(n)}},n}^{\mathcal{D}(n)})^T$

$$\overline{F}_{jk}^{i} = \int_{\Gamma_{\Sigma_i}} f_k\{\mathcal{P}^i[\mathbf{x}^{\Sigma,P}(\xi, t)], \xi_k, t\}\Psi_j^{\Sigma_i,P_i}(\xi)\rho_{\Sigma_i}(\xi)d\xi, \tag{68}$$

and $j = 1, ..., N_{\Sigma_i}$, $k = 1, ..., n$ with $i = \mathcal{D}(k)$. Let $\overline{\mathbf{F}} = (\overline{F}_{11}^{\mathcal{D}(1)}, ..., \overline{F}_{N_{\Sigma_{\mathcal{D}(1)}},1}^{\mathcal{D}(1)}, ..., \overline{F}_{1n}^{\mathcal{D}(n)}, ..., \overline{F}_{N_{\Sigma_{\mathcal{D}(n)}},n}^{\mathcal{D}(n)})^T$; then, the system (67) can be compactly written as

$$\dot{\overline{\mathbf{a}}} = \overline{\mathbf{F}}(\overline{\mathbf{a}}, t), \tag{69}$$

with appropriate initial condition [see expression 13] and will be referred to as the approximate Galerkin projection (AGP). Using this generalization of AGP system, it is straightforward to generalize the intrusive PWR introduced in Section 4.1.2.

4.3.1 Intrusive PWR Algorithm

In the intrusive PWR, one applies the WR to the AGP system. Based on the discussion in Section 4.2, the decomposition $\mathcal{D}$ of the original system (1) is used to impose a decomposition on the system (69), leading to

$$\dot{\overline{\mathbf{a}}}_i = \overline{\mathbf{F}}_i(\overline{\mathbf{a}}_i, \overline{\mathbf{d}}_i, t), \tag{70}$$

for $i = 1, ..., m$, where

$$\overline{\mathbf{a}}_i = (\overline{a}_{1k_1}^{i}, ..., \overline{a}_{N_{\Sigma_i},k_1}^{i}, ..., \overline{a}_{1k_{|\mathcal{C}_i|}}^{i}, ..., \overline{a}_{N_{\Sigma_i},k_{|\mathcal{C}_i|}}^{i})^T, \tag{71}$$

$$\overline{\mathbf{F}}_i = (\overline{F}_{1k_1}^{i}, ..., \overline{F}_{N_{\Sigma_i},k_1}^{i}, ..., \overline{F}_{1k_{|\mathcal{C}_i|}}^{i}, ..., \overline{F}_{N_{\Sigma_i},k_{|\mathcal{C}_i|}}^{i})^T, \tag{72}$$

$k_i \in \mathcal{C}_i$, and $\overline{\mathbf{d}}_i = (\overline{\mathbf{a}}_{j_i}^T, ..., \overline{\mathbf{a}}_{j_{N_i}}^T)^T$ is the decoupling vector (recall notation from Section 3.1). One then follows the procedure for waveform relaxation or its adaptive version, as described in Section 3.1. Adaptive WR can lead to a significant increase in convergence of WR as demonstrated in [17], and would be illustrated later in the Section 5.

As discussed in Section 2.1.1, the projection step (66) can be very tedious and in some cases not possible. Hence, applying waveform relaxation directly to the system (70) may not be practical in many instances. In the next section, we describe an alternative nonintrusive approach using probabilistic collocation, which does not require the projection step (66) explicitly.

4.4 Nonintrusive Probabilistic Waveform Relaxation

In terms of the projection operator (65), we can rewrite each subsystem in (23) as

$$\dot{\mathbf{y}}_i = \mathbf{F}_i\{\mathbf{y}_i, \mathcal{P}^i[\mathbf{d}_i(t, \cdot)], \Lambda_i, t\}, \tag{73}$$

where $\mathbf{d}_i(t, \cdot)$ is the *stochastic decoupling vector* or *probabilistic waveform,*

$$\mathbf{d}_i(t, \cdot) = [(\mathbf{y}_{j_i}^{\Sigma_{j_1},P_{j_1}})^T, ..., (\mathbf{y}_{j_{N_i}}^{\Sigma_{j_{N_i}},P_{j_{N_i}}})^T]^T. \tag{74}$$

where we have explicitly indicated the dependence on the parameters [see definition (28)]. Here, for any $i = 1, ..., m$, $\mathbf{y}^{\Sigma_i,P_i} = (x_{j_1^i}^{\Sigma_i,P_i}, ..., x_{j_{M_i}^i}^{\Sigma_i,P_i})^T$, with

$$x_{j_k^i}^{\Sigma_i,P_i}(\xi, t) = \sum_{m=1}^{N_{\Sigma_i}} \overline{a}_{mj_k^i}(t)\Psi_m^{\Sigma_i,P_i}(\xi) = \mathcal{P}^{i,j_k^i}(x_{j_k^i}^{\Sigma,P}). \tag{75}$$

By definition the coefficients $\overline{a}_{mj_k^i}(t)$ in the above expansion satisfy the system (67). These coefficients can be obtained by using the quadrature formula (21), by repeatedly solving the system (73) over an appropriate collocation grid $C(\mathbf{l}, n_i)$,

$$\mathcal{C}(\mathbf{l}, n_i + n_i^c) = \mathcal{C}(\mathbf{o}, n_i) \times \mathcal{C}(\mathbf{m}, n_i^c), \tag{76}$$

where $\mathbf{l} = (\mathbf{o}, \mathbf{m})$, $\mathcal{C}(\mathbf{o}, n_i) = C^1_{o_1} \times ... \times C^1_{o_{n_i}}$ is the collocation grid corresponding to parameters Λ_i, with $n_i = |\Lambda_i|$, $\mathbf{o} = (o_1, ..., o_{n_i})$, and $\mathcal{C}(\mathbf{m}, n_i^c) = C^1_{m_1} \times ... \times C^1_{m_{n_i^c}}$ is the collocation grid corresponding to parameters Λ_i^c, with $n_i^c = |\Lambda_i^c|$ and $\mathbf{m} = (m_1, ..., m_{n_i^c})$. For simplicity we take $o_1 = ... = o_{n_i} = l_s$ and $m_1 = ... = m_{n_i^c} = l_c$ for $i = 1, ..., m$. Since, the behavior of the ith subsystem is weakly affected by the parameters Λ_i^c through the decoupling vector, then consistent with remark (4.1) we can take

$$l_c < l_s, \tag{77}$$

leading to an adaptive collocation grid for each subsystem. With this, we are ready to generalize the nonintrusive PWR approach introduced in Section 4.1.3.

4.4.1 Non-Intrusive PWR Algorithm

Step 1: Apply graph decomposition (see Section 3 for details) to identify weakly interacting subsystems in system (1).

Step 2: (Assignment-partitioning process): Partition (1) into m subsystems (obtained in step 1) leading to system of equations given by (23). Obtain, Λ_i, Λ_i^c and Σ_i for each subsystem, $i = 1, ..., m$. Choose the parameters l_{si}, l_{ci}, P_i.

Step 3: (Initialization of the relaxation process with no coupling effect): Set $I = 1$ and guess an initial waveform $\{\mathbf{y}_i^0(t) : t \in [0, T_s]\}$ for each $i = 1, ..., m$ consistent with initial condition (see step 1 in relaxation process described in Section 3.1). Set

$$\mathbf{d}_i^1(t) = [\mathbf{y}_{j_1}(t), ..., \mathbf{y}_{j_{N_i}}(t)], \tag{78}$$

$i = 1, ..., m$, and solve for $\{\mathbf{y}_i^{\Lambda_i, P_i, 1}(t), t \in [0, T_s]\}$ using

$$\dot{\mathbf{y}}_i^1 = \mathbf{F}^i(\mathbf{y}_i^1, \mathbf{d}_i^1(t), \Lambda_i, t), \tag{79}$$

with an initial condition $\mathbf{y}_i^1(0) = \mathbf{y}_i^0(0)$ on a collocation grid $C(\mathbf{o}, n_i)$. Determine the gPC expansion $\mathbf{y}_i^{\Lambda_i, P_i, 1}(t, \cdot)$ over P-variate polynomial space W^{Λ_i, P_i} by computing the expansion coefficients from the quadrature formula (21).

Step 4: (Initialization of the relaxation process, incorporating first level of coupling effect): Set $I = 2$ and for each $i = 1, ..., m$, set

$$\mathbf{d}_i^2(t, \cdot) = [\mathbf{y}_{j_1}^{\Lambda_{j_1}, P_{j_1}, 1}(t, \cdot), ..., \mathbf{y}_{j_{N_i}}^{\Lambda_{j_{N_i}}, P_{j_{N_i}}, 1}(t, \cdot)], \tag{80}$$

and solve for $\{\mathbf{y}_i^2(t), t \in [0, T_s]\}$ from

$$\dot{\mathbf{y}}_i^2 = \mathbf{F}^i(\mathbf{y}_i^2, \mathbf{d}_i^2(t, \cdot), \Lambda_i, t), \tag{81}$$

with an initial condition $\mathbf{y}_i^2(0) = \mathbf{y}_i^0(0)$, over a collocation grid $C(\mathbf{l}, n_i + n_i^c)$. Obtain the expansion $\mathbf{y}_i^{\Sigma_i, P_i, 2}(t, \cdot)$ using (75). From now on we shall denote the solution vector of the ith subsystem at Ith iteration by $\mathbf{y}_i^{\Sigma_i, P_i, I}$.

Step 5: (Analyzing the decomposed system at the Ith iteration): For each $i = 1, ..., m$, set

$$\mathbf{d}_i^I = (\mathbf{y}_{j_1}^{\Sigma_{j_1} P_{j_1}, (I-1)}, ..., \mathbf{y}_{j_{N_i}}^{\Sigma_{j_{N_i}}, P_{j_{N_i}}, (I-1)}), \tag{82}$$

and solve for $\{\mathbf{y}^I(t) : t \in [0, T_s]\}$ from

$$\dot{\mathbf{y}}_i^I = \mathbf{F}^i\{\mathbf{y}_i^I, \mathcal{P}^i[\mathbf{d}_i^I(t, \cdot)], \Lambda_i, t\}, \tag{83}$$

with initial condition $\mathbf{y}_i^I(0) = \mathbf{y}_i^0(0)$ over a collocation grid $C(\mathbf{l}, n_i + n_i^c)$. Obtain the expansion $\mathbf{y}_i^{\Sigma_i, P_i, I}(t, \cdot)$ using the expansions (75).

- Step 6: (Iteration): Set $I = I + 1$ and go to step 5 until satisfactory convergence has been achieved.

4.5 Convergence of PWR

Below, we give the convergence proof for the iterative PWR approach. The proof is based on showing that the AGP system is Lipschitz if the original systems is Lipschitz (see condition 2), and then invoking the standard WR convergence result (Section 3.1).

Proposition 4.1. *Convergence of PWR: The intrusive and nonintrusive PWR algorithms described in Sections 4.3.1 and 4.4.1, respectively, converge.*

Proof: We prove the result for intrusive PWR. By construction, because nonintrusive PWR algorithm solves the AGP system (69) in a different way, the convergence result holds for nonintrusive PWR as well. Consider the AGP system (69) and let

$$\mathbf{a}_1 = (\overline{a}_{11}^{1\mathcal{D}(1)}, ..., \overline{a}_{N_{\Sigma_{\mathcal{D}(1)}},1}^{1\mathcal{D}(1)}, ..., \overline{a}_{1n}^{1\mathcal{D}(n)}, ..., \overline{a}_{N_{\Sigma_{\mathcal{D}(n)}},n}^{1\mathcal{D}(n)})^T, \tag{84}$$

and

$$\mathbf{a}_2 = (\overline{a}_{11}^{2\mathcal{D}(1)}, ..., \overline{a}_{N_{\Sigma_{\mathcal{D}(1)}},1}^{2\mathcal{D}(1)}, ..., \overline{a}_{1n}^{2\mathcal{D}(n)}, ..., \overline{a}_{N_{\Sigma_{\mathcal{D}(n)}},n}^{2\mathcal{D}(n)})^T. \tag{85}$$

Let for a given $k = 1, ..., n$, $i = \mathcal{D}(k)$, then

$$\mathcal{P}^{i,k}[x_k^{l,\Sigma,P}(t,\xi)] = \sum_{j=1}^{N_{\Sigma_i}} \overline{a}_{jk}^{li}(t)\Psi_j^{\Sigma_i,P}(\xi), \tag{86}$$

and $\mathcal{P}^i(\mathbf{x}^{l,\Sigma,P}) = [\mathcal{P}^{i,1}(x_1^{l,\Sigma,P}), ..., \mathcal{P}^{i,n}(x_n^{l,\Sigma,P})]$, for $l = 1, 2$ and for simplifying notation we have dropped subscripts on P vectors. Then for each $k = 1, ..., n$, $i = \mathcal{D}(k)$, $j = 1, ..., N_{\Sigma_i}$,

$$\begin{aligned}
||\overline{F}_{jk}^i(\mathbf{a}_2) - \overline{F}_{jk}^i(\mathbf{a}_1)|| &= \left| \int_{\Gamma_\Sigma} [f_k(\mathcal{P}^i(\mathbf{x}^{\Sigma,P,2}), \xi, t) - f_k(\mathcal{P}^i(\mathbf{x}^{\Sigma,P,1}), \xi, t)] \times \Psi_j^{\Sigma_i,P}(\xi) w_\Sigma(\xi) d\xi \right| \\
&\leq \int_{\Gamma_\Sigma} L(\xi) \left[\sum_{m=1}^{n} \sum_{p=1}^{N_{\Sigma_{\mathcal{D}(m)}}} |(a_{pm}^{l,\mathcal{D}(m)} - a_{pm}^{2,\mathcal{D}(m)})\Psi_p^{\Sigma_{\mathcal{D}(m)},P}| \right] \times |\Psi_j^{\Sigma_i,P}(\xi)| w_\Sigma(\xi) d\xi \\
&\leq \sum_{m=1}^{n} \sum_{p=1}^{N_{\Sigma_{\mathcal{D}(m)}}} L_{pj}^{i\mathcal{D}(m)} |(a_{pm}^{1,\mathcal{D}(m)} - a_{pm}^{2,\mathcal{D}(m)})|
\end{aligned} \tag{87}$$

where,

$$L_{pj}^{iq} \quad = \quad \int_{\Gamma_\Sigma} L(\xi)|\Psi_p^{\Sigma_q,P}(\xi)||\Psi_j^{\Sigma_i,P}(\xi)| w_\Sigma(\xi) d\xi. \tag{88}$$

For a given $i = 1, ..., n$ and $j = 1..., N_{\Sigma_{\mathcal{D}(i)}}$, let $L_{ij}^g = \left[L_{j1}^{\mathcal{D}(i)\mathcal{D}(1)} ... L_{jN_{\Sigma_{\mathcal{D}(1)}}}^{\mathcal{D}(i)\mathcal{D}(1)} ... L_{j1}^{\mathcal{D}(i)\mathcal{D}(n)} ... L_{jN_{\Sigma_{\mathcal{D}(n)}}}^{\mathcal{D}(i)\mathcal{D}(n)} \right]$ and define

$$L_i^g = \begin{bmatrix} L_{i1}^g \\ \vdots \\ L_{iN_{\Sigma_{\mathcal{D}(i)}}}^g \end{bmatrix}, \quad L^g = \begin{bmatrix} L_1^g \\ \vdots \\ L_n^g \end{bmatrix}, \tag{89}$$

then,

$$||\overline{\mathbf{F}}(\mathbf{a}_2) - \overline{\mathbf{F}}(\mathbf{a}_1)|| \quad \leq \quad \overline{L}||\mathbf{a}_2 - \mathbf{a}_1||, \tag{90}$$

where $\overline{L} = ||L^g||$ is a matrix norm of L^g. Hence, the system (69) is Lipschitz. Thus, given the original system (1) is Lipschitz [condition (2)], the approximate system (69) is also Lipschitz as shown above (90). Hence, by proposition 3.1 (adaptation of Theorem 5.3 in [18]), we conclude that PWR converges.

The question of how the decomposition of the system and the choice of the PWR algorithm parameters P, l_s, l_c influence: (1) the rate of convergence of PWR, and (2) the approximation error [due to the truncation introduced in the AGP system (69), the use of adaptive collocation grid, i.e., condition (77) and computation of the modal coefficients by the quadrature formula], needs to be further investigated.

4.6 Scalability of PWR with Respect to Standard PCM

The scalability of nonintrusive PWR relative to PCM is shown in Fig. 2, where the ratio $\mathcal{R}_F/\mathcal{R}_I$ indicates the computation gain over standard full grid PCM approach applied to the system (1) as a whole. Here $\mathcal{R}_F = l^p$ is the number of deterministic runs of the complete system (1), which comprises m subsystems each with $p_i, i = 1, ..., m$ uncertain parameters, such that $p = \sum_{i=1}^m p_i$ and l denotes the level of full grid. Similarly, $\mathcal{R}_I = 1 + \sum_{i=1}^m l_s^{p_i} + I_{\max}(\sum_{i=1}^m l_s^{p_i} \bigotimes_{j\neq i} l_c^{p_j})$ is the total computational effort with the PWR algorithm (including additional effort incurred due to several iterations), where $I_{\max}$ is the number of PWR iterations. Clearly, the advantage of PWR becomes evident as the number of subsystems m and parameters in the network increases. Note that the linear scaling depicted in Fig. 2 is only for the ratio $\mathcal{R}_F/\mathcal{R}_I$. Moreover, PWR is inherently parallelizable.

5. EXAMPLE PROBLEMS

In this section we illustrate intrusive and nonintrusive PWR through several examples of linear and nonlinear networked systems with increasing number of uncertain parameters. While most examples are of ODEs, we also give an example application of PWR to an algebraic system. This illustrates how in principle one can extend the application of PWR to DAEs, just like the WR approach extends to DAEs [15]. Through some examples we study how the strength of interaction between subsystems affects the convergence rate and the approximation error of PWR. In one of the examples related to the building model, we also show how time-varying uncertainty can be incorporated into standard UQ framework by using the Karhunen-Loeve expansion. In all the examples, we compare the solution accuracy of PWR with other UQ approaches (e.g., Monte Carlo and quasi-Monte Carlo methods, QMC), and wherever appropriate mention computation gain offered by PWR over the standard application of gPC and PCM.

5.1 Stability Problem

We first consider a simple system, with two states $(x_1, x_2) \in \mathbb{R}^2$,

$$\dot{x}_1 = ax_1^2 + cx_2^2 - v_1, \tag{91}$$

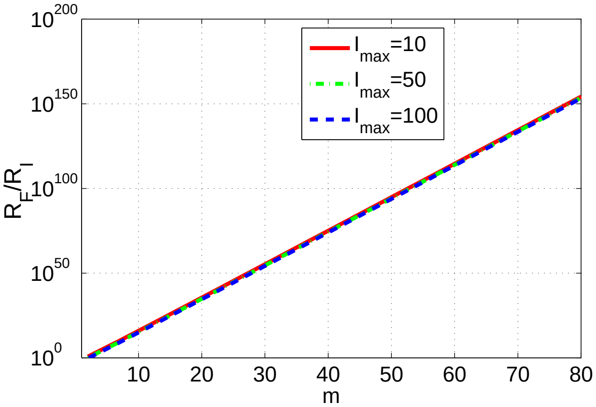


FIG. 2: Scalability of PWR algorithm with respect to standard PCM, when implemented with full grid collocation as the subsystem level UQ method, with $p_i = 5, \forall i = 1, ..., m$, $l = l_s = 5, l_c = 3$ and $I_{\max} = 10, 50, 100$: these numeric values are chosen for illustration purposes. The computational gain of PWR becomes insensitive to $I_{\max}$ (number of iterations), as the number of subsystems m increases.

$$\dot{x}_2 = cx_1^2 + bx_2^2 - v_2, \tag{92}$$

where c, v_1, v_2 are fixed parameters, and a, b are uncertain with Gaussian distribution $\mathcal{G}$ and tolerance 20% (i.e. $\sigma = 0.2\mu$, where μ is the mean and σ is the standard deviation of $\mathcal{G}$). The parameter c determines the coupling strength between two subsystems described by the two equations. The output of interest is the stability of the system, which is determined by λ_m, the maximum eigenvalue of the Jacobian

$$J(x_{10}, x_{20}; a, b, c) = \begin{pmatrix} 2ax_{10} & 2cx_{20} \\ 2cx_{10} & 2bx_{20} \end{pmatrix},$$

where x_{10}, x_{20} is the equilibrium point satisfying

$$\begin{aligned} ax_{10}^2 + cx_{20}^2 - v_1 = 0, \\ cx_{10}^2 + bx_{20}^2 - v_2 = 0. \end{aligned} \tag{93}$$

Figure 3 shows the UQ results obtained by applying PWR [with $l_s = 5$, $l_c = 3$, and $P_1 = (5, 3), P_2 = (3, 5)$] to iteratively solve the algebraic system (93). We make comparison with the *true* [to imply a more accurate result obtained by solving the complete system (93)] distribution of λ_m obtained by using a full collocation grid on the parameter space (a, b) with $l_a = 5, l_b = 5, P = (5, 5)$. PWR converges to the true mean and variance as shown in the left and right panels of Fig. 3 for two different values of c. As the coupling strength c increases (see the right panel in Fig. 3), the number of iterations required for the convergence increases, as expected.

5.2 Building Example

For energy consumption computation, a building can be represented in terms of a reduced order thermal network model of the form [29]

$$\frac{d\mathbf{T}}{dt} = A(\mathbf{u}(t); \xi)T + B \begin{pmatrix} \mathbf{Q}_e(t) \\ Q_i(t) \end{pmatrix}, \tag{94}$$

where $\mathbf{T} \in R^n$ is a vector comprising internal zone air temperatures and internal and external wall temperatures; $A[\mathbf{u}(t); \xi]$ is the time-dependent matrix with ξ being parameters, $\mathbf{u}(t)$ is control input vector (comprising zone supply flow rate and supply temperature), vectors $\mathbf{Q}_e = [T_{amb}(t), Q_s(t)]^T$ represent the external (outside air temperature and solar radiation), and Q_i is the internal (occupant) load disturbances. We consider the problem of computing uncertainty

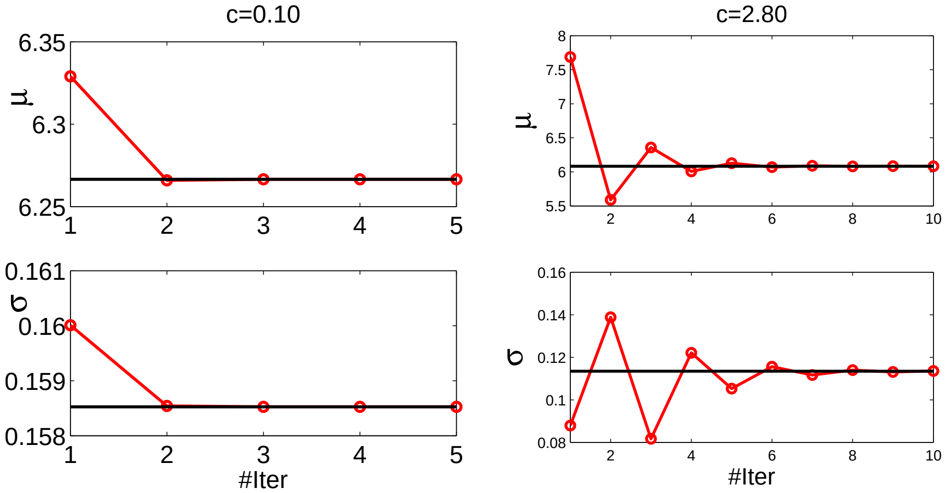


FIG. 3: Left panel: Convergence of mean (μ) and variance (σ) of λ_m for $c = 0.1$. Right panel: Convergence of mean and variance of λ_m for $c = 2.8$. Black line indicates the true values.

in building energy consumption due to uncertainty in building thermal-related properties and uncertain disturbance loads. These uncertainties can be categorized into: (i) static parametric uncertainty which include parameters such as wall thermal conductivity and thermal capacitance, heat transfer coefficient, window thermal resistance etc.; and (ii) time-varying uncertainties which include the external and internal load disturbances.

Recall that the traditional UQ approaches and PWR, which builds on them, can only deal with parametric uncertainty. To account for time-varying uncertain processes, we employ the Karhunen-Loeve (KL) expansion [30]. The KL expansion allows representation of second-order stochastic processes as a sum of random variables. In this manner, both parametric and time-varying uncertainties can be treated in terms of random variables. We next demonstrate both intrusive and nonintrusive PWR methods.

5.2.1 Two-Zone Example

We first consider a simplified two-zone building model as shown in Fig. 4. Here the state $\mathbf{T}$ is a 10-dimensional vector comprising internal wall temperatures and the internal zone air temperatures, where we have assumed that the outer wall surfaces are held at ambient temperature. We also assume that the ambient temperature and solar load are deterministic fixed quantities and there is no internal occupant load. Thus, in computing the uncertainty in the zone temperatures, we only consider parametric uncertainty. Specifically, we assume that the heat transfer coefficient and the thermal conductivity of the walls in each zone have standard deviations of 10% around their nominal values of 3.16 W/m^2/K and 4.65 W/m/K, respectively. Thus, locally each zone is affected by two uncertain parameters, with heat transfer coefficient being a common (i.e., same) parameter and thermal conductivity being the other. Using complete Galerkin projection with $P_i = (2,2,2), i = 1,2$, gives rise to a 60-state ODE model. To apply WR/AWR to this system, we first identify the weakly interacting states. By construction the two zones weakly affect each other, which is identified by the spectral clustering [13] (or wave equation-based clustering [16]) applied to the system (94). This decomposition is imposed on the complete Galerkin system, as explained in Section 4.3.1. As expected, we found that if one applies spectral clustering to the complete Galerkin system instead, one recovers the same decomposition.

Treating 1000 Monte Carlo (MC) samples as the truth, we compare the results of a simulated full Galerkin projected system using both standard waveform relaxation [18] as well as adaptive waveform relaxation [17] in Fig. 5(a). AWR provides a speed-up by a factor of ≈ 12. In Fig. 5(a), one can see that the complete Galerkin projection predicts the same temperature variation over 8 h as MC-based methods.

As explained before, one can further exploit the weak interaction between the two zones to reduce the overall number of equations in Galerkin projection. To construct the AGP system, we reduce the order of expansion for the random parameters indirectly affecting each zone so that $P_1 = (2,2,1)$ and $P_2 = (1,2,2)$. With this, the number of equations in Galerkin projection reduces from 60 to 50. The resulting solution is shown in Fig. 5(a). We see that the

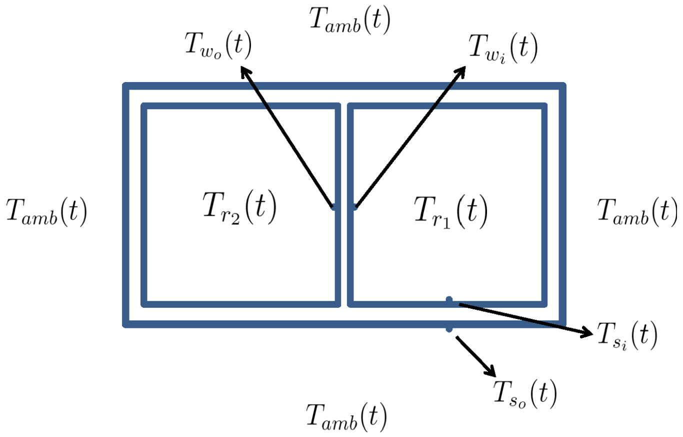


FIG. 4: Diagram of the two-zone thermal model of a building. $T_{amb}(t) = 293$ K in this example.

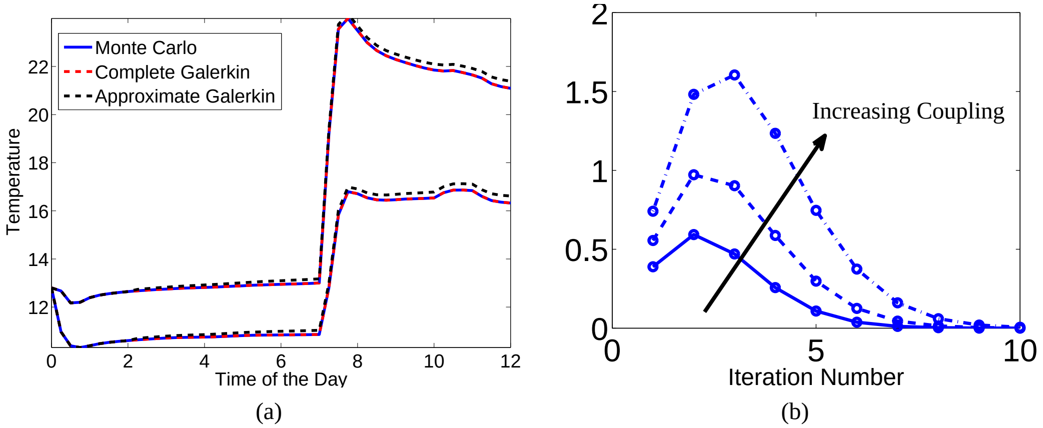


FIG. 5: (a) Comparison of Monte Carlo, complete Galerkin projection, and approximate Galerkin projection. (b) Normalized error in waveform relaxation as a function of iteration count with increasing coupling. Complete Galerkin and approximate Galerkin are shown. Approximate Galerkin system is found to have greater error as a function of iteration number.

error starts to grow as time increases. However, over 8 h the max error in the room temperatures is 5×10^{-2} K. Thus, despite reducing the computational effort, one can still get a fairly accurate answer.

Figure 5(b) shows the effect of coupling (which is the reciprocal of the coefficient of thermal conductivity of the internal wall) on errors introduced in complete and approximate Galerkin projections. As expected, the approximate Galerkin projection has a higher error [given by $E_T(t)$] than complete Galerkin projection [given by $E_C(t)$]. Moreover this error is more pronounced at low iteration numbers. From the figure, it also clear that as the coupling increases, the number of iterations required for obtaining same solution accuracy increases. For further discussion on the relationship between the coupling and number of iterations, see [17].

5.2.2 Multizone Example

In this section, we consider a larger 6-zone building thermal network model with 68 states. This model admits a decomposition into 23 subsystems, as revealed by the spectral graph approach [see Fig. 6(b)]. This decomposition is

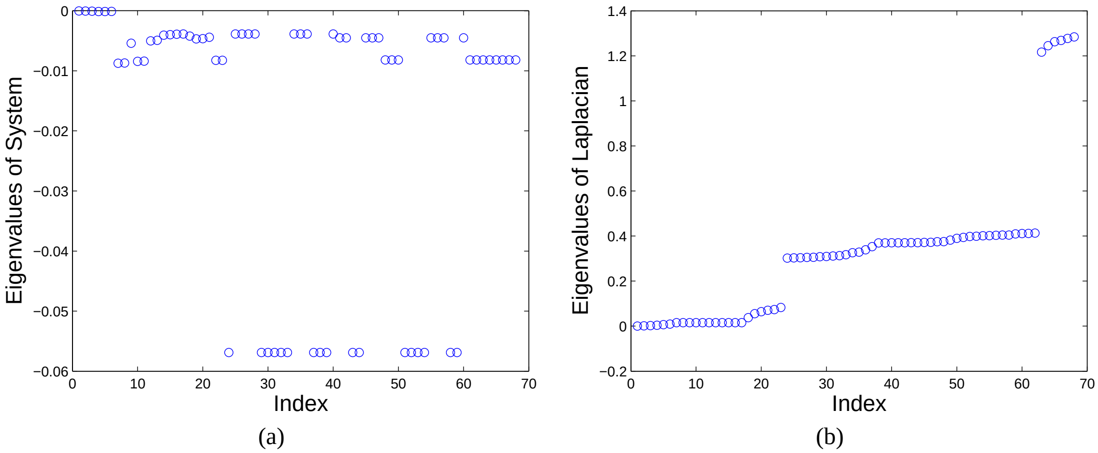


FIG. 6: (a) Bands of eigenvalues of time-averaged $A(t;\xi)$ for nominal parameter values. (b) First spectral gap in graph Laplacian revealing 23 subsystems in the network model.

consistent with three different timescales (associated with external and internal wall temperature, and internal zone temperatures) present in the system, as shown by the three bands in Fig. 6(a).

Next we demonstrate the nonintrusive PWR approach to compute uncertainty in energy consumption due to both parametric uncertainty and time varying uncertain loads. As described earlier, we use the KL expansion to transform time-varying uncertainty into parametric form.

KL Expansion [30]: Let $\{X_t = X(\xi, t), t \in [a, b]\}$ be a quadratic mean-square second-order stochastic process with covariance functions $R(t, s)$. If $\{\phi_n(t)\}$ are the eigenfunctions of the integral operator with kernel $R(\cdot, \cdot)$ and $\{\lambda_n\}$ the corresponding eigenvalues, i.e.,

$$\int_a^b R(t, s)\phi_n(s)ds = \lambda_n \phi_n(t), \qquad t \in [a, b], \tag{95}$$

then

$$X(t, \theta) = \overline{X}(t) + \lim_{N\to\infty} \sum_{n=1}^{N} \sqrt{\lambda_n} a_n(\xi)\phi_n(t), \qquad \text{uniformly for } t \in [a, b], \tag{96}$$

where $\overline{X}(t)$ is the mean of the process and the limit is taken in the quadratic mean sense. The random coefficients $\{a_n\}$ satisfy

$$a_n(\theta) = \frac{1}{\sqrt{\lambda_n}} \int_a^b [X(\xi, t) - \overline{X}(t)]\phi_n(t)dt, \tag{97}$$

and are uncorrelated $E[a_m a_n] = \delta_{mn}$. The basis functions also satisfy the orthogonality property

$$\int_a^b \phi_m(t)\phi_n(t)dt = \delta_{mn}, \tag{98}$$

and the kernel admits an expansion of the form

$$R(s, t) = \lim_{N\to\infty} \sum_{n=1}^{N} \lambda_n \phi_n(t)\phi_n(s). \tag{99}$$

Generally, an analytical solution to the eigenvalue problem (95), also known as the Fredholm equation of the second kind, is not available. Several numerical techniques have been proposed; we used the expansion method described in [31].

To apply the KL expansion to the building problem, we assume that the stochastic disturbances $[T_{amb}(t), Q_s(t), Q_{int}(t)]$ are Gaussian processes. This guarantees that the random variables a_n in the KL expansion are independent Gaussian random variables with a zero mean [31]. Let the joint distribution of a nonstationary Gaussian process be

$$f(X_t, X_s) = \frac{1}{2\pi\sigma(s)\sigma(t)\sqrt{1 - \rho(t, s)}} e^{-\{1/2[1-\rho^2(t,s)]\}\left\{[x_t^2/\sigma^2(t)]+[x_s^2/\sigma^2(s)]-[2\rho(t,s)x_s x_t/\sigma(t)\sigma(s)]\right\}} \tag{100}$$

where $\rho(t, s)$ is the correlation coefficient and is related to the covariance kernel as

$$R(t, s) = \rho(t, s)\sigma(t)\sigma(s). \tag{101}$$

We assumed the processes $T_{amb}(t), Q_s(t)$ to have a stationary exponential correlation function

$$R(t, s) = \sigma^2 e^{-(|t-s|/T_c)}, \tag{102}$$

with a constant variance σ^2 and a constant correlation timescale T_c. For the internal occupancy load $Q_{int}(t)$ we constructed $R(t, s)$ as follows. For a typical office building, we know that the occupancy load is negligible with low variance during the early and later parts of the day. During peak hours in the middle of the day the occupant load can show significantly high variability. To capture this effect we divided the normalized time domain $[0,1] = [0,t_1] \cup (t_1,t_2) \cup (t_2,1)$ and obtained the desired variation by choosing [in expression (101)]

$$\sigma(t) = \sigma\left\{\tanh[a(t-t_1)] - \tanh[a(x-t_2)]\right\}/2, \quad \rho(t,s) = e^{-[|t-s|/T_c(s,t)]}, \tag{103}$$

with the correlation timescale

$$T_c(s,t) = \{1-\tanh[a(t-t_1)]\}\{1-\tanh[a(s-t_1)]\}/4 + \{1+\tanh[a(t-t_2)]\}\{1+\tanh[a(s-t_2)]\}/4, \tag{104}$$

and parameter a controls the slope of the $\tanh$ function. Figure 7 shows the covariance kernel for external $[T_{amb}(t), Q_s(t)]$ and internal $Q_{int}(t)$ loads. For the choice of parameters indicated in Fig. 7, we found, using the expansion method [31] with Legendre polynomials as the basis functions, that KL expansion up to order 3 and up to order 6 can capture more than 90% of total variance, for internal and external loads, respectively. Finally, note that these external variables act like global variables which affect all zones in the building model. Such global variables can be easily handled in the PWR framework by treating them as local variables [and hence including them in the local parameter vector defined in (27)] for each subsystem (here the different building zone) they affect.

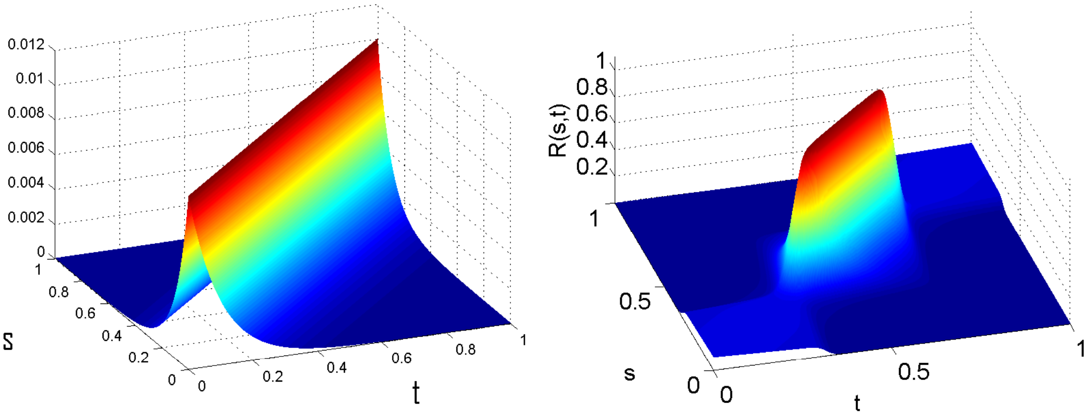
In UQ computation, we considered the effect of 14 random variables comprising external wall thermal resistance in the six zones, and first dominant random variable obtained in the KL representation of internal load (for each zone) and first two dominant random variable obtained in KL expansion for solar load. Figure 8 show the nonintrusive PWR results on the decomposed network model. As is evident, the iterations converge rapidly in two steps with a distribution close to that obtained from QMC (using a 25,000-sample Sobol sequence) applied to the 68 thermal network model (94) as a whole.

5.3 Coupled Oscillators

Finally, we consider a coupled phase only oscillator system which is governed by nonlinear equations

$$\dot{x}_i = \omega_i + \sum_{j=1}^{N} K_{ij}\sin(x_j - x_i), \qquad i = 1, ..., n, \tag{105}$$

where $n = 80$ is the number of oscillators, $\omega_i, i = 1, ..., n$ is the angular frequency of oscillators, and $K = [K_{ij}]$ is the coupling matrix. The frequencies ω_i of every alternative oscillator, i.e., $i = 1, 3, ..., 79$, is assumed to be uncertain with



(a) (b)

FIG. 7: (a) Covariance kernel (102 for external load with $T_c = 0.1$ and $\sigma = 0.1$. (b) Covariance kernel (103) for internal load with $t_1 = t_2 = 0.3$, $a = 20$, $\sigma = 0.1$.

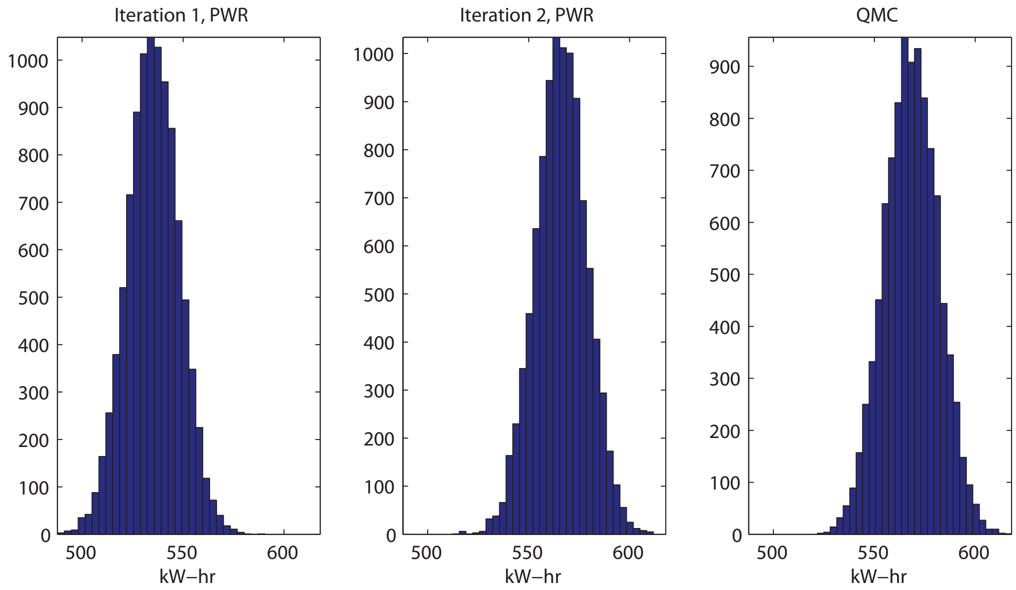


FIG. 8: Histogram of building energy computation for two iterations in PWR. Also shown is the corresponding histogram obtained by QMC for comparison.

a Gaussian distribution with 20% tolerance (i.e., with a total $p = 40$ uncertain parameters); all the other parameters are assumed to take a fixed mean value. We are interested in the distribution of the synchronization parameters $R(t)$ and phase $\phi(t)$ defined by $R(t)e^{\Phi(t)} = 1/N\sum_{j=1}^{N} e^{ix_j(t)}$. Figure 9 shows the topology of the network of oscillators (left panel), along with the eigenvalue spectrum of the graph Laplacian (right panel). The spectral gap at 40 implies 40 weakly interacting subsystems in the network.

Figure 10 shows UQ results obtained by application of nonintrusive PWR to the decomposed system with $l_s = 5$, $l_c = 2$. We make a comparison with QMC, in which the complete system (105) is solved at 25,000 Sobol points [32]. Remarkably, the PWR converges in 4–5 iterations, giving very similar results to that of QMC. It would be infeasible to use full grid collocation for the networks as a whole, since even with lowest level of collocation grid, i.e., $l = 2$ for each parameter, the number of samples required become $\mathbf{R}_F = 2^{40} = 1.0995e + 012!$.

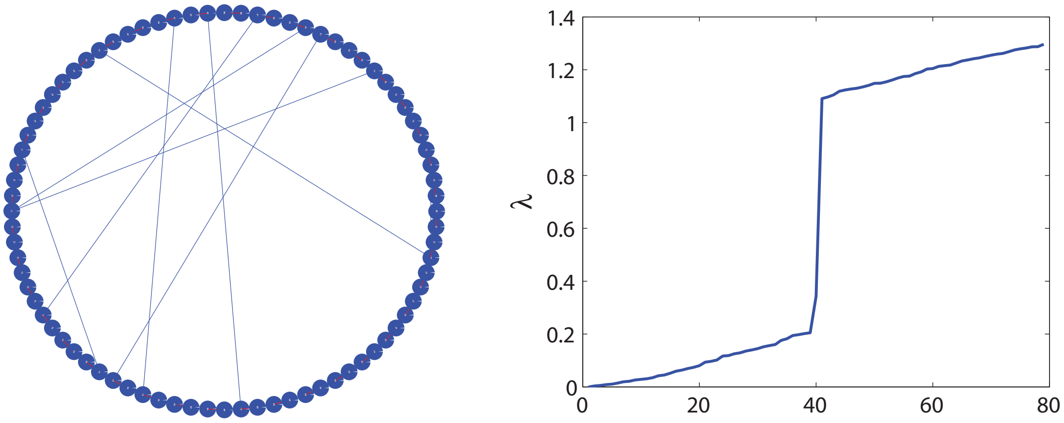


FIG. 9: Left panel shows a network of $N = 80$ phase only oscillators. Right panel shows spectral gap in eigenvalues of normalized graph Laplacian, that reveals that there are 40 weakly interacting subsystems.

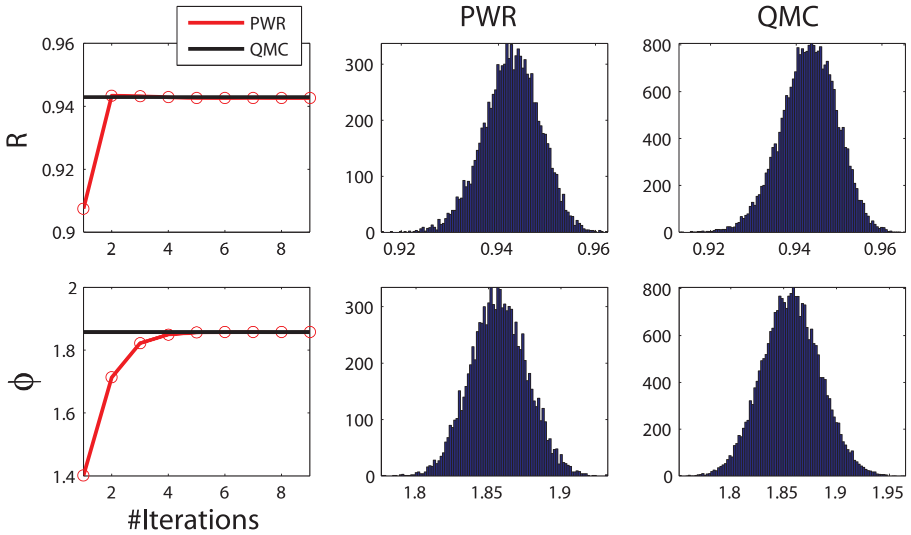


FIG. 10: Convergence of mean of the magnitude $R(t)$ and phase $\phi(t)$, and the respective histograms at $t = 0.5$.

6. CONCLUSION AND FUTURE WORK

In this paper we have proposed an uncertainty quantification approach which exploits the underlying dynamics and structure of the system. Specifically, we considered a class of networked system whose subsystems are dynamically weakly coupled to each other. We showed how these weak interactions can be exploited to overcome the dimensionality curse associated with traditional UQ methods. By integrating graph decomposition and waveform relaxation with generalized polynomial chaos and probabilistic collocation framework, we proposed an iterative UQ approach which we call *probabilistic waveform relaxation*. We have developed both intrusive and nonintrusive forms of PWR. We proved that this iterative scheme converges under weak assumptions and illustrated it in several examples with promising results. In this work, we have also introduced an approximate Galerkin projection that, based on the results of graph decomposition, computes “strong” and “weak” influence of parameters on states. An appropriate order of expansion, in terms of the parameters, is then selected for the various states.

This approach is shown to accelerate UQ considerably. Several questions need to be further investigated; these include the effect of the choice of parameters on the rate of convergence and approximation error of the PWR algorithm. In order to exploit multiple timescales that may be present in a system, multigrid extensions [33] of PWR may also be desirable. In future work, we also intend to test our approach on much larger dynamic systems that arise in building applications

ACKNOWLEDGMENTS

This work was in part supported by DARPA DSO (Dr. Cindy Daniell PM) under AFOSR contract no. FA9550-07-C-0024 (Dr. Fariba Fahroo PM). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the AFOSR or DARPA.

REFERENCES

1. Fishman, G., *Monte Carlo: Concepts, Algorithms, and Applications*, Springer-Verlag, New York, 1996.
2. Niederreter, H., *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, PA, 1992.
3. Sloan, I. H. and Joe, S., *Lattice Methods for Multiple Integration*, Oxford University Press, Oxford, 1994.

4. Xiu, D. and Karniadakis, G., The Wiener-Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.*, 24:619–644, 2002.
5. Xiu, D., Efficient collocational approach for parametric uncertainty analysis, *Commun. Comput. Phys.*, 2(2):293–309, 2007.
6. Hegland, M., Heinrich, S., and Sloan, I. H., Guest Editors' Preface, *J. Complex.*, 26(5):407, 2010.
7. Gerstner, T. and Griebel, M., Numerical integration using sparse grids, *Numer. Algorithms*, 18(3-4):209–232, 1998.
8. Foo, J., Wan, X., and Karniadakis, G., The multi-element probablistic collocation method (ME-PCM): Error analysis and applications, *J. Comput. Phys.*, 227(22):9572–9595, 2008.
9. Sobol, I. M., Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates, *Math. Comput. Simul.*, 55:271–280, 2001.
10. Callier, M., Chan, W. S., and Desoer, C. A., Input-output stability theory of interconnected systems using decomposition techniques, *IEEE Trans. Circuits Syst.*, 23(12):714–729, 1976.
11. Georgiou, T. T. and Smith, M. C., Linear systems and robustness -a graph point of view, *Lect. Notes Control Inf. Sci.*, 183:114–121, 1992.
12. Mezic, I., Coupled nonlinear dynamical systems: Asymptotic behavior and uncertainty propagation, *Proc. of 43rd IEEE Conf. on Decision and Control*, Atlantis, Paradise Island, Bahamas, Dec. 14–17, 2004.
13. Luxburg, U. V., A tutorial on spectral clustering, *Stat. Comput.*, 17:395–416, 2007.
14. Chung, F., *Spectral Graph Theory*, Conference Board of the Mathematical Sciences, Washington, 1997.
15. White, J. and Vincentelli, A. S., *Relaxation Techniques for the Simulation of VLSI Circuits*, Kluwer International Series in Engineering and Computer Science, Boston, 1987.
16. Sahai, T., Speranzon, A., and Banaszuk, A., Hearing the clusters in a graph: A distributed algorithm, *Automatica*, 48(1):15–24, 2012.
17. Klus, S., Sahai, T., Liu, C., and Dellnitz, M., An efficient algorithm for the parallel solution of high-dimensional differential equations, *J. Comput. Appl. Math.*, 235(9):3053–3062, 2011.
18. Lelarasmee, E., The waveform relaxation method for time domain analysis of large scale integrated circuits: Theory and applications, Tech. Rep. Memorandum No. UCB/ERL M82/40, UC Berkeley, 1982.
19. Thomas, E., A parallel algorithm for simulation of large neural networks, *J. Neurosci. Methods*, 98:123–134, 2000.
20. Thomas, E., *Parallel Algorithms for Biologically Realistic, Large Scale Neural Modeling*, Master's Thesis, Royal Melbourne Institute of Technology, 1998.
21. Rauber, T. and Rünger, G., Parallel implementations of iterated Runge-Kutta methods, *Int. J. Supercomput. Appl.*, 10(1):62–90, 1996.
22. Roy, S. and Dounavis, A., Longitudinal partitioning based waveform relaxation algorithm for transient analysis of long delay transmission lines, *2011 IEEE MTT-S International Microwave Symposium Digest (MTT)*, 2011.
23. McKay, W., Monti, A., Santi, E., and Dougal, R., A co-simulation approach for ACSL-based models, *Proc. Huntsville Simulation Conf.*, Huntsville, AL, Oct. 3-4, 2001.
24. Stoer, M. and Wagner, F., A simple min-cut algorithm, *J. ACM*, 44(4):585–591, 1997.
25. Fiedler, M., Algebraic connectivity of graphs, *Czech. Math. J.*, 23:289–305, 1973.
26. Fiedler, M., A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory, *Czech. Math. J.*, 25(4):619–633, 1975.
27. Golub, G. and Loan, C. V., *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1996.
28. Kempe, D. and McSherry, F., A decentralized algorithm for spectral analysis, *J. Comput. Syst. Sci.*, 74(1):70–83, 2008.
29. O'Neill, Z., Narayanan, S., and Brahme, R., Model-based thermal load estimation in buildings, *Fourth National Conf. of IBPSA-USA*, New York, USA, Aug. 11–13, 2010.
30. Stark, H. and Wood, J. W., *Probability, Random Processes and Estimation Theory for Engineers*, Prentice-Hall, Englewood Cliffs, NJ, 2002.
31. Huang, S. P., Quek, S. T., and Phoon, K. K., Convergence study of the truncated Karhunen-Loeve expansion for simulation of stochastic processes, *Int. J. Numer. Meth. Eng.*, 52:1029–1043, 2001.

32. Joe, S. and Kuo, F. Y., Remark on algorithm 659: Implementing sobol's quasirandom sequence generator, *ACM Trans. Math. Softw.*, 29:49–57, 2003.

33. Trottenberg, U., Oosterlee, C. W., and Schller, A., *Multigrid*, Academic Press, London, 2001.