Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Optimal Bayesian experiment design for nonlinear dynamic systems with chance constraints

Joel A. Paulson [1], Marc Martin-Casas [1], Ali Mesbah *

Department of Chemical and Biomolecular Engineering, University of California, Berkeley, CA 94720, USA

ARTICLE INFO

Article history:
Received 16 October 2018
Accepted 20 January 2019
Available online 7 February 2019

Keywords:
Bayesian optimal experiment design
Stochastic dynamic optimization
Utility function
Arbitrary polynomial chaos
Chance constraints

ABSTRACT

The optimal design of experiments is crucial for maximizing the information content of data across a wide-range of experimental goals. This paper presents a Bayesian approach to optimal experiment design (OED) for parameter inference in constrained, dynamic, and nonlinear systems under noisy, incomplete, and indirect measurements. Bayesian OED maximizes an expected utility objective, which accounts for prior and posterior uncertainty in the model parameters from an information-theoretic standpoint. Due to the complicated form of the expected utility, it must be estimated using sample-based methods and, in particular, a nested Monte Carlo estimator that is expensive to evaluate using the full dynamic model. We propose a novel surrogate model based on arbitrary polynomial chaos (aPC), which readily applies to any type of prior distribution. The aPC expansions are constructed locally at each design visited during the iterative optimization procedure. The main cost in aPC, which is the determination of the expansion coefficients, is minimized by estimating these coefficients from only a minimal set of dynamic model evaluations. Although sample-based estimators can also be applied to the chance constraints, this leads to a potentially large number of binary variables, such that a smooth moment-based approximation is preferred in this work. Numerical simulations indicate that the proposed surrogate can significantly lower the computational cost of the Bayesian OED, while guaranteeing the original chance constraints are satisfied without noticeably increasing the average time to find a solution. As such, this methodology appears to have the potential to pave the way for real-time or sequential dynamic experiment design in a fully Bayesian setting.

1. Introduction

The selection of optimal conditions (or designs) for conducting experiments is crucial for improving the information content of observations, especially when experiments are expensive and/or time-consuming to perform. Optimal experiment design (OED) uses a mathematical relationship between the design variables, parameters, and observables of a system to systematically select experimental conditions that maximize some design metric that is relevant to, for example, parameter inference or model discrimination [1–5].

Experiment design has been extensively studied in the classical (or frequentist) framework, in both theory and practice [2]. Classical OED design criteria for parameter inference are generally defined in terms of some scalar metric of the Fisher information matrix (FIM) such as the *alphabetic optimality* criteria *A*-, *D*-, and *E*-optimality [6]. Alternatively, OED can adopt a Bayesian perspective, where the design criteria are expressed in terms of an *expected utility* quantity that accounts for both prior and posterior uncertainty in the model parameters from a decision-theoretic point of view [7]. In the case of linear models subject to Gaussian uncertainties, the Bayesian alphabetic optimality criteria reduce to mathematical forms that are equivalent to their classical FIM counterparts [6]. For example, Bayesian *D*-optimality corresponds to a utility function equal to the Shannon information of the parameter estimates.

For nonlinear models, however, analytic expressions do not exist for Bayesian design criteria. Thus, extensions of Bayesian OED to nonlinear models are commonly based on linearization of the model and Gaussian approximations of the posterior distribution in order to derive tractable design criteria in terms of the FIM [8,9]. Many OED approximations involve prior expectations of the FIM, which can be interpreted as robust (or stochastic) versions of classical OED and have also been referred to as *pseudo-Bayesian* OED in the literature [10,11]. Note that this is in contrast to classical (or "locally optimal") OED approaches that maximize some function of

* Corresponding author.
E-mail address: mesbah@berkeley.edu (A. Mesbah).

[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.jprocont.2019.01.010
0959-1524/© 2019 Elsevier Ltd. All rights reserved.

the FIM evaluated around a current "best guess" for the unknown model parameters, which directly leads to deterministic optimization problems [12,13]. In both cases, the resulting design criteria can be interpreted as *approximations of an underlying expected utility* in the Bayesian setting. None of these approximations, however, are suitable when the prior distribution is broad (i.e., has large variance) or deviates from normality (i.e., is non-Gaussian), and are known to lead to increasingly suboptimal designs.

This paper investigates a Bayesian approach to OED for constrained nonlinear systems with continuous (or high-dimensional) design spaces, with the goal of designing experiments that are optimal for parameter inference. Bayesian methods provide the most general framework for experiment design and inference in nonlinear systems with noisy, incomplete, and indirect data [14]. As discussed in [15], the expected utility framework can accommodate a wide variety of information-theoretic criteria. Most often, the expected utility function is explicitly defined in terms of the posterior parameter distribution. One of the most common choices for the expected utility is the mutual information between parameters and observations (equivalent to the expected information gain from the prior to the posterior), which can be expressed in terms of the Kullback–Leibler (KL) divergence from the posterior to the prior [16]. As such, the main downside of Bayesian OED is its high computational cost relative to classical approaches, which is a direct consequence of numerical evaluation of the expected utility. In general, the expected utility must be approximated using Monte Carlo (MC) integration over the joint observation and parameter space, which can be a high-dimensional space. Thus, early work on Bayesian OED focused on evaluating the expected utility over each element of a small, finite number of designs (on the order of ten) to avoid the challenge of *optimizing* the expected utility over continuous design spaces [17].

Due to the sample-based evaluation of the expected utility, Bayesian OED is naturally formulated as a stochastic optimization problem. In [18], a Markov chain Monte Carlo (MCMC) sampler of the joint design, parameter, and data space is developed such that the marginal distribution of all sampled designs is proportional to the expected utility. Here, the design that leads to the joint mode of the marginal distribution is optimal. Since finding the joint mode is increasingly difficult as the number of design variables increases, a simulated-annealing optimization method was used to achieve faster convergence [19]. However, this approach does not appear to be easily applicable for design dimensions larger than four [20]. Alternative optimization methods, including those based on the Nelder-Mead [21] and simultaneous perturbation stochastic approximation [22], have also been used for Bayesian OED in [15]. The main drawback of these methods is they require many iterations to converge, even for small problems, suggesting they could become excessively expensive for larger design spaces commonly encountered in dynamic OED problems. This is primarily due to the fact that stochastic optimization methods ignore gradient information. Alternatively, so-called "gradient-based" optimization techniques use gradient evaluations to improve the rate of convergence to a local optimum, thus requiring fewer iterations and potentially much less computational cost. When applied to problems with stochastic objectives, gradient-based optimization methods can be broadly categorized as stochastic approximation (SA) [23] or sample average approximation (SAA) [24]. Hybrids of these two approaches are also possible. The main practical difference between SA and SAA is that the i.i.d. samples are updated at each iteration in the former while they are treated as fixed in the latter. In either case, when the model used for OED is computationally intensive, evaluating the expected utility and/or its gradients can be computationally prohibitive. To address this challenge, [25] proposed the use of a *surrogate model* for fast estimation of the expected utility, where polynomial approximations (in particular, polynomial chaos expansions (PCEs) [26]) of the model outputs are constructed to capture their dependence on the uncertain parameters and design variables. The main downside of this approach is that the size of the surrogate grows exponentially with respect to the number of design variables, making it impractical for dynamic systems.

This paper extends the Bayesian OED approach of [25] in several directions. First, we introduce a PCE-based surrogate model that is particularly advantageous for dynamic systems. The proposed approach is based on developing local PCEs for the outputs around each design visited during optimization such that the exponential growth in the size of the PCEs with respect to the number of design variables is avoided. Second, we leverage the theory of orthogonal polynomials to construct the PCEs with respect to *arbitrary* probability measures of uncertain parameters (e.g., priors can be correlated or multi-modal). Thus, the proposed approach is not restricted to particular prior types and also ensures the PCEs are most accurate in high probability regions of the parameter space. Third, the proposed Bayesian OED approach can handle nonlinear probabilistic path and terminal constraints, which can be enforced to ensure safety and/or quality of the experiment. We show how probabilistic constraints can be readily incorporated into the OED problem using the PCE-based surrogate model. A key feature of the proposed Bayesian OED approach is that it can be implemented using state-of-the-art dynamic optimization methods (e.g., multiple shooting or collocation on finite elements [27]), so that the underlying structure of the optimization problem can be exploited for computational efficiency as in classical OED. The Bayesian OED approach is demonstrated on a benchmark dynamic predator-prey problem. To the best of our knowledge, this is the first study on Bayesian OED for nonlinear dynamic systems in the presence of a fairly general class of constraints.

2. Formulation of optimal Bayesian experimental design

We are interested in choosing the best experiments from a continuous design space, for the purpose of estimating model parameters from noisy and indirect measurements. In other words, we are interested in experiments that are "optimal" for parameter inference performed in the Bayesian setting, without the need for limiting assumptions such as linear models or strong observability.

Let $(\Omega, \mathfrak{F}, P)$ be a probability space, where Ω is the sample space (or abstract set of outcomes), $\mathfrak{F}$ is a σ-algebra of the subsets of Ω, and $P : \mathfrak{F} \to [0, 1]$ is a probability measure. Let the vector of real-valued random variables $\theta : \Omega \to \Theta \subseteq \mathbb{R}^{n_\theta}$ denote the uncertain model parameters of interest, i.e., these are parameters that we aim to estimate from experimental data. A probability measure μ_θ is induced by the random variables θ, such that $\mu_\theta(A) = P(\theta^{-1}(A))$ for all $A \in \mathbb{R}^{n_\theta}$ (often referred to as the induced or pushforward measure). We can then define $p_\theta(\theta) = d\mu_\theta / d\theta$ as the probability density of θ with respect to the Lebesgue measure. This density is guaranteed to exist as long as the random variables are continuous, which we will assume throughout this work. For simplicity of notation, we shall use $p(\cdot)$ to represent all density functions, and which specific distribution it corresponds to is reflected by its arguments, e.g., $p(\theta)$ denotes $p_\theta(\theta)$. When needed for clarity, we will explicitly include a subscript that denotes the associated random variable.

In a similar fashion, we treat the observations from the experiment $y \in \mathcal{Y}$ (also referred to as "noisy measurements" or "data") as a real-valued random vector with an appropriate probability density, and $d \in \mathcal{D}$ as the design (also referred to as "control" or "input") variables. Since we are particularly interested in the dynamic evolution of the experiment, we focus on systems modeled by a collection of nonlinear ordinary differential equations (ODEs) for ease of presentation

$$\dot{x}(t) = f(t, x(t), d(t), \theta), \quad \forall t \in [0, t_f] \tag{1}$$

where $x : [0, t_f] \to \mathbb{R}^{n_x}$ are the state variables with time derivatives $\dot{x}$ and initial conditions $x(0) = x_0$ and $d : [0, t_f] \to \mathbb{R}^{n_{\text{input}}}$ are the design variables. As such, the state evolution $x(t;d, \theta)$ is implicitly a function of the input trajectory and the model parameters. We assume that the dynamic evolution of (1) is constrained so that it must satisfy hard input constraints $d(t) \in \mathbb{D} \subset \mathbb{R}^{n_d}$ and probabilistic (or chance) state constraints of the form

$$P(x(t; d, \theta) \in \mathbb{X}) \geq 1 - \beta, \quad \forall t \in [0, t_f], \tag{2}$$

where $\beta \in [0, 1]$ is the allowed probability of constraint violation. The constraints (2) can be interpreted as a generalization of nominal ($\beta = 0.5$) or worst-case ($\beta = 0$) enforcement of state constraints, and can be used to ensure safety or quality throughout the experiment. Terminal state constraints can be handled in a similar fashion to the path constraints (2) so we neglect them here to limit the notational complexity. We also assume that measurements can be taken throughout the experiment at discrete times $t_1, \ldots, t_T$ and can be modeled as

$$y_i = g(t_i, x(t_i; d, \theta)) + \epsilon_i, \quad i = 1, \ldots, T, \tag{3}$$

where $y_i \in \mathbb{R}^{n_{y_i}}$ and ϵ_i denote the measurement and the noise in the measurement at time t_i, respectively. The set of observations is then given by $y = (y_1, \ldots, y_T)$ while the corresponding noise vector is given by $\epsilon = (\epsilon_1, \ldots, \epsilon_T)$. Note that we have not made any assumptions on the noise model so that it can have any distribution.

It is important to note that the observation space $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$, $n_y = n_{y_1} + \cdots + n_{y_T}$ is represented by a finite number of dimensions. The design space $\mathcal{D}$, on the other hand, is represented by the set of all continuous-time *trajectories* that satisfy $d(t) \in \mathbb{D}$, which is an infinite-dimensional space. Because of this fact, the design space must be discretized in order to approximate $\mathcal{D} \subseteq \mathbb{R}^{n_d}$ numerically on a computer with finite n_d. Some common approximations include piecewise constant and piecewise linear though, in theory, any finite-dimensional parametrization can be utilized. Letting N_T denote the number of parameters used to approximate the continuous-time trajectories, the total number of design variables becomes $n_d = n_{\text{input}} N_T$, which can be quite large in practice.

If an experiment is performed under a given design d and a realization of the data y is then measured, the change in the state of knowledge/information about the parameters is given by Bayes' rule:

$$p(\theta | y, d) = \frac{p(y | \theta, d) p(\theta | d)}{p(y | d)}, \tag{4}$$

where $p(\theta|d)$ is the prior density, $p(y|\theta, d)$ is the likelihood function, $p(\theta|y, d)$ is the posterior density of interest, and $p(y|d) = \int_{\Theta} p(y|\theta, d) p(\theta|d) d\theta$ is the evidence, which represents a normalizing constant that is a function of the design and data. In most practical applications, the prior knowledge on θ does not vary with the choice of design, leading to the simplification $p(\theta|d) = p(\theta)$, i.e., knowing the design of the current experiment without knowing its observations does not affect our belief about the parameters. The likelihood function is assumed to be given, and describes the discrepancy between the observations and a forward model prediction in a probabilistic way. Note that the likelihood function has a one-to-one relationship with the noise model. The forward model $G : \Theta \times \mathcal{D} \to \mathcal{Y}$, generally denoted as $G(\theta, d)$, in this case is implicitly defined by (1) and (3). Using this notation, we have $y = G(\theta, d) + \epsilon$ with a corresponding likelihood function $p(y|\theta, d) = p_\epsilon(y - G(\theta, d))$. While the majority of classical OED approaches are developed around the assumption that $\epsilon \sim \mathcal{N}(0, \Sigma_\epsilon)$, i.e., a Gaussian likelihood, fully Bayesian approaches are not restricted by this choice and can handle any choice of p_ϵ including noise distributions that depend on the design and parameter values.

We take a decision-theoretic approach to define the *expected utility* (or expected reward) to quantify the value of experiments. As suggested originally in [16], the expected utility can take on the following general form:

$$\begin{aligned} U(d) &= \int_{\mathcal{Y}} \int_{\Theta} u(d, y, \theta) p(\theta, y | d) d\theta dy \\ &= \int_{\mathcal{Y}} \int_{\Theta} u(d, y, \theta) p(\theta | y, d) p(y | d) d\theta dy, \end{aligned} \tag{5}$$

where $u(d, y, \theta)$ denotes the utility function. The utility function should be chosen to reflect the usefulness of an experiment at conditions d, given a particular value of the parameters θ and outcome y. Since the precise values of θ and y are unknown when the experiment is performed, the objective is defined as the expectation of $u(d, y, \theta)$ over the joint distribution of θ and y. It is important that the utility function incorporates the experimental aims and is specific to the application of interest [15]. For example, designs that result in efficient estimation of the model parameters may not be useful for the prediction of future outcomes. A key advantage of Bayesian OED, however, is that a wide variety of experimental goals can be accommodated through the proper choice of utility function including parameter estimation, model discrimination, and the prediction of future observations [17].

Although utility functions are quite flexible and can be tailored to specific goals, in order to derive useful and illustrative results, we focus on utility functions that lead to valid measures of *information gain* on the estimated parameters from the experimental data. In particular, we use the relative entropy, also known as the Kullback–Leibler (KL) divergence, from the posterior to the prior [16]:

$$\begin{aligned} u(d, y, \theta) &= D_{KL}(p_{\theta|y,d}(\,\cdot\,) \| p_\theta(\,\cdot\,)) \\ &= \int_{\Theta} \ln \left[\frac{p(\theta | y, d)}{p(\theta)} \right] p(\theta | y, d) d\theta = u(d, y). \end{aligned} \tag{6}$$

The intuition behind this expression is that a large KL divergence from the posterior to the prior implies that the data y decreases the entropy in θ by a large amount and, hence, those data are more informative for parameter estimation. Note that this choice of utility function integrates over the parameter space and is therefore not a function of the parameters θ. As a result, substituting (6) into (5) produces the following expression [15]:

$$U(d) = \int_{\mathcal{Y}} \int_{\Theta} \ln \left[\frac{p(\theta | y, d)}{p(\theta)} \right] p(\theta | y, d) d\theta p(y | d) dy. \tag{7}$$

Thus, $U(d) = \mathbb{E}_{y|d} \left\{ D_{\text{KL}}(p_{\theta|y,d}(\,\cdot\,) || p_\theta(\,\cdot\,)) \right\}$ represents the *expected information gain* on the parameters θ where the expectation is taken over the prior predictive distribution $p(y|d)$. Note that $U(d)$ is also equivalent to the *mutual information* between the parameters θ and data y conditioned on d. As discussed in [25], the KL divergence has several desirable properties that warrants its use as a general-purpose utility function for parameter inference, which we briefly summarize here: (i) it satisfies minimal requirements to be a valid measure of information on a set of experiments, such as "always at least informative" ordering; (ii) it gives an indication of information gain in the sense of Shannon information; (iii) it applies to general nonlinear models $G(\theta, d)$ and is consistent with linear optimal design theory based on the FIM; and (iv) it has been shown to perform well for a wide range of tasks, as it provides general guidance for learning in an uncertain environment.

Finally, the optimal design is defined as the design that maximizes the expected utility $U(d)$ subject to constraints:

$$d^\star = \mathrm{argmax}_{d\in D}\; U(d), \quad \text{s.t.} \quad P(x(t;d,\theta)\in\mathbb{X})\geq 1-\beta. \tag{8}$$

There are a number of challenges that must be overcome when solving (8). The biggest difficulty is related to the probabilistic operators that define the expected utility (7) and the state chance constraints (2), which cannot be evaluated in closed-form even when the model is a simple polynomial function of θ. These challenges are addressed in the next section using different types of approximations.

3. Stochastic dynamic optimization with chance constraints

In this section, we formulate the proposed approximated form of the Bayesian OED problem (8). Although many different approaches are available for approximating the integrals in (2) and (7), we focus on two particular choices that lead to a smooth optimization problem, which can be readily solved with state-of-the-art methods for dynamic optimization. The procedure for approximately solving (8) is then summarized at the end of this section.

3.1. Sample-based estimator for the expected utility

The expected utility in (7) does not have a closed-form solution, except when the forward model is a linear function of θ. Instead, this expression must be numerically approximated. By applying Bayes' rule to the quantities inside and outside of the logarithm, and then approximating the integrals using MC, we obtain the following nested MC estimator for the expected utility [20]

$$U(d)\approx \hat{U}_{N,M}(d,\theta_s,y_s)=\frac{1}{N}\sum_{i=1}^{N}\ln\left[\frac{p_{y|\theta,d}(y^{(i)}|\theta^{(i)},d)}{\frac{1}{M}\sum_{j=1}^{M}p_{y|\theta,d}(y^{(i)}|\tilde{\theta}^{(i,j)},d)}\right], \tag{9}$$

where $\theta_s=\{\theta^{(i)}\}_{i=1}^{N}\cup\{\tilde{\theta}^{(i,j)}\}_{i,j=1}^{N,M}$ are i.i.d. samples from the prior $p(\theta)$ and $y_s=\{y^{(i)}\}_{i=1}^{N}$ are independent samples from the likelihood $p(y|\theta^{(i)},d)$. The inner sum is needed to approximate the evidence evaluated at $y^{(i)}$, i.e., $p(y^{(i)}|d)$, which typically does not have an analytic form. The variance of this estimator is proportional to $A(d)/N+B(d)/NM$ and its bias (to leading order) is $C(d)/M$, where A, B, and C are constant terms that depend only on the distributions at hand [20]. Hence, the size of N controls variance while the size of M controls the bias. Note that the estimator $\hat{U}_{N,M}$ is asymptotically unbiased, but is biased for finite M. Although alternative numerical integration schemes can replace MC in (9), MC is likely the method of choice since its convergence properties are independent of dimension [28] and n_y+n_θ will often by large in practice. Additionally, MC can be directly applied to arbitrary priors and likelihood functions.

3.2. Moment-based approximation of chance constraints

The state constraints $x(t;d,\theta)\in\mathbb{X}$ in (2) can be generally described by a set of nonlinear inequality constraints of the form

$$\mathbb{X}=\{x\in\mathbb{R}^{n_x}: h(x)\leq 0\}, \tag{10}$$

where $h:\mathbb{R}^{n_x}\to\mathbb{R}^{n_c}$. Letting $c(x)=\max_{1\leq j\leq n_c}h_j(x)$ where h_j is the jth element of h, a MC estimator can also be developed for the state chance constraints [29]

$$\begin{aligned}P(h(x(t;d,\theta))\leq 0)&=\mathbb{E}\{\mathbf{1}_{[0,\infty)}(c(x(t;d,\theta)))\}\\ &\approx\frac{1}{N}\sum_{i=1}^{N}\mathbf{1}_{[0,\infty)}(c(x(t;d,\theta^{(i)}))),\end{aligned} \tag{11}$$

where $\mathbf{1}_{[0,\infty)}$ denotes the indicator function. It is important to note that since the indicator function and the max operator are non-smooth functions, they must be implemented with binary variables. Therefore, under this approximation, (8) must be recast as a mixed-integer nonlinear program (MINLP) with a potentially large number of binary variables, i.e., the optimization can be difficult and expensive to solve. In addition, the feasible region of the problem depends on the particular set of samples used in the approximation and can change in a non-smooth way whenever new samples are drawn.

A simpler alternative is to develop a *moment-based* approximation of the chance constraint. The most common example is the mean-variance representation [30]

$$\begin{aligned}&\mathbb{E}\{H_j(t,\theta,d)\}+r(t)\sqrt{\mathrm{Var}\{H_j(t,\theta,d)\}}\leq 0,\\ &j=1,\ldots,n_c,\quad \forall t\in[0,t_f],\end{aligned} \tag{12}$$

where $r(t)\in[0,\infty)$ is a backoff parameter that can vary over time and $H_j(t,\theta,d)=h_j(x(t;\theta,d))$ is the random variable associated with the jth constraint function. The parameterized constraints (12) will be smooth functions of time and the design variables whenever the original system equations are smooth in direct contrast to (11). Although these constraints can be straightforwardly derived when $H(\theta,d)$ is a Gaussian random vector [31], this does not suggest that (12) will guarantee satisfaction of the original chance constraints (2), as discussed next.

Whenever $H(t,\theta,d)$ is normally distributed, $r(t)=r$ is constant and its smallest possible value can be determined exactly from a quantile of the chi-squared distribution. Finding a suitable $r(t)$ becomes more challenging in the case of general distributions, and this is even further complicated by the fact that the shape of the distribution of $H(t,\theta,d)$ can change over time and with the design variable. One promising method for overcoming these challenges in the nonlinear setting is to use an iterative simulation-based procedure that requires $P(x(t;d^\star(r),\theta)\in\mathbb{X})$ to be estimated under the optimal design $d^\star(r)$ found by solving the Bayesian OED problem (8) subject to (12) in place of (2). If the probability of violation is greater (or less) than β, then r should be increased (or decreased). A lower bound of $r=0$ (corresponding to nominal constraints) can be used, while an upper bound for r can be determined from the "distributionally robust" Cantelli–Chebyshev inequality [32] that ensures the original chance constraints hold for every possible distribution that shares the same mean and covariance as $H(t,\theta,d)$. Note that the violation probability must still be estimated using, for example, MC, but this only needs to be done at the optimal design, as opposed to all of the designs visited during the optimization. Readers are referred to [33] for a similar methodology that has been applied in the context of nonlinear model predictive control. Note that alternative methods have been developed to approximate chance constraints using smoothing functions (e.g., [34]), which can be used in place of (12) in this work.

3.3. Sample average approximation for chance constrained Bayesian OED

The Bayesian OED problem with moment-based chance constraint approximation can be solved using either gradient-based or non-gradient-based methods. Although gradient-based methods require additional information, they are usually more efficient than their non-gradient-based counterparts. This is especially true in dynamic systems of the form (1) that are known to have certain structures that can be efficiently exploited. An important consideration, however, is that U can only be approximated by MC-based estimators such as $\hat{U}_{N,M}$, meaning that optimization methods for stochastic objective functions are needed. The Robbins–Monro (RM) stochastic approximation is one such gradient-based stochastic optimization method. RM is based on an iterative update that resembles steepest descent, except for the fact that it uses an unbiased estimator of the gradient, i.e., the samples used to estimate $U(d)$ may be different than those used to estimate $\nabla_d U(d)$. The choice of the step size sequence is often viewed as a key weakness of RM, as the performance of the algorithm can be very sensitive to the step size [23]. Sample average approximation (SAA), on the other hand, reduces the stochastic optimization problem to a deterministic one by fixing the noise throughout the entire optimization [24]. The main advantage of SAA over RM is that deterministic optimization methods can be directly applied, including state-of-the-art solvers that efficiently handle nonlinear objectives and constraints. For this reason, we focus on SAA exclusively in this work. It is worth noting, however, that the proposed approach is not restricted to SAA and could be solved using RM in a similar manner to that shown in [25], with the main difference being that the nonlinear constraints must be handled using either projection or barrier methods [35].

SAA requires all the “noise” samples θ_s and y_s to be fixed. However, the samples y_s are design-dependent, as they are distributed according to the likelihood function that in turn depends on the given design d. This issue can be addressed in practice by transforming y to be in terms of only design-independent random variables. One example of this transformation for a Gaussian likelihood function is as follows

$$y = G(\theta, d) + \epsilon = G(\theta, d) + C(\theta, d)z, \tag{13}$$

where C is a diagonal matrix with the non-zero entries representing the standard deviation of the noise that can generally depend on the parameters and design, and z is a vector of i.i.d. standard normal random variables. For example, the choice of elements $C_{i,j} = 0.1|G_i(\theta, d)|\delta_{ij}$ corresponds to “10% Gaussian noise on the ith component of the model” where δ_{ij} is the Kronecker delta. Other forms of the likelihood can be easily accommodated by replacing the right-hand size of (13) by a generic function of θ, d, and some design-independent random vector z.

Let $\mathcal{D}_r$ denotes the set of design variables that satisfy $d(t) \in \mathbb{D}$ and the mean-variance constraints (12) for a given backoff radius parameter r. We can then state the proposed SAA approximation to the Bayesian OED problem (8) as

$$\hat{d}_s = \text{argmax}_{d \in \mathcal{D}_r} \hat{U}_{N,M}(d, \theta_s, z_s), \tag{14}$$

where $\hat{d}_s$ and $\hat{U}_{N,M}(\hat{d}_s, \theta_s, z_s)$ are, respectively, the optimal design and objective values under a particular set of realizations of the random variables θ and z. A deterministic optimization algorithm can then be used to find $\hat{d}_s$ as an approximation to $d^\star$. Estimates of $U(\hat{d}_s)$ can be improved by applying the estimator $\hat{U}_{N',M'}(\hat{d}_s, \theta_{s'}, z_{s'})$ under a larger number of realizations $N' > N$ and $M' > M$ in order to attain a lower variance. Furthermore, multiple optimization runs $B > 0$ can be performed (often referred to as a *bootstrap*) to obtain a sampling distribution for the optimal design values and the optimal objective values, i.e., $\hat{d}_s^b$ and $\hat{U}_{N,M}(\hat{d}_s^b, \theta_s^b, z_s^b)$ for $b = 1, \ldots, B$. The sets θ_s^b and z_s^b are independently chosen for each optimization run, but remained fixed within each run. It has been shown that the optimal design and objective estimates converge in distribution to their respective true values under certain assumptions [24,36]. Lastly, stochastic bounds on the true optimal value can be constructed by estimating the optimality gap from the set of B replicate runs. Using the optimality gap estimator and/or its variance estimated from the MC standard error formula, one can decide whether more runs are required or which of the B optimal designs are most trustworthy.

The approximated Bayesian OED problem is still a challenging problem to solve due to the dynamic forward model. In fact, $NM + N$ separate ODEs of the form (1) must be integrated in order to evaluate (9) at a single design point. There are a number of ways to handle the infinite-dimensional nature of these ODE constraints (due to the continuous time variable t), including variational, sequential, and simultaneous approaches [27]. Simultaneous methods discretize both the state and design/control profiles in time using, for example, collocation of finite elements, and have the advantage of only solving the ODEs once at the optimal point, i.e., can avoid intermediate solutions that may not exist or require excessive computational effort. Even with this efficient implementation, the huge number of constraints needed to account for the $NM + N$ forward model evaluations can make (14) impractical to solve. In addition, we do not have a closed-form expression for the mean and covariance of $H(t, \theta, d)$, which is needed to evaluate the constraint function. Therefore, in the next section, we develop surrogate models for G and H that can greatly reduce the computational cost at each iteration of the optimization while still ensuring that the solutions found are accurate approximations to (14).

4. Arbitrary polynomial chaos expansions as surrogates

The main challenge in applying the aforementioned stochastic optimization algorithms to the constrained Bayesian OED problem is the complexity of the forward model and its gradients. In fact, only a single evaluation of $\hat{U}_{N,M}(d, \theta_s, z_s)$ requires $O(NM)$ separate solutions of the forward model while an even larger number of equations must be solved to calculate $\nabla_d \hat{U}_{N,M}(d, \theta_s, z_s)$ as the gradient is defined in terms of the sensitivities $\nabla_d G(\theta, d)$. Here, we address this challenge by replacing G with a simple surrogate model based on polynomial expansions (PCEs). This cheaper “surrogate” must be accurate over the entire support of the prior Θ and the entire design space $\mathcal{D}$. Not only does the surrogate model allow the nested MC estimator in (9) to be evaluated in a computationally tractable manner, but its polynomial form greatly simplifies the structure and complexity of the gradient of the expected utility.

These gains come at the cost of introducing a new source of error due to the polynomial approximation of the forward model; however, this error can often be kept low in practice. In fact, the error can always be decreased by increasing the order of the expansion for reasonably smooth functions, as discussed in this section. It is worth noting that we focus on PCE-based surrogates as they have been demonstrated to be effective in the context of Bayesian OED, yielding high accuracy and multiple order-of-magnitude speedups over direct evaluation of the forward model [15,25]. The proposed surrogate in this work has some important differences to that used in [15,25], including that our method readily applies to arbitrary prior distributions and the size of the surrogate does not scale with the number of design variables. We first present our proposed surrogate and then describe these differences in more detail at the end of this section.

4.1. PCE formulation

With slight abuse of notation, we describe the proposed PCE approximation in the context of a generic scalar function G. Whenever this function is multivariate, the procedure is simply applied to each component of G. As such, the developed procedure can be separately applied to each component of the forward model $G(\theta, d)$ and constraint function $H(\theta, d)$ in the Bayesian OED problem.

The truncated PCE approximation (in terms of the uncertain parameters only) is then defined in the following manner

$$G_L(\theta, d) = \sum_{i=1}^{L} a_i(d)\Psi_i(\theta), \tag{15}$$

where L is the total number of terms retained in the expansion; $a_i(d)$ are the expansion coefficients that depend on the design variables; and $\Psi_1, \ldots, \Psi_L$ are polynomial basis functions. We again highlight the fact that θ has an associated probability density $p(\theta)$ on which we have made no restrictions. We can define an inner product $\langle \cdot, \cdot \rangle_\theta$ operator with respect to $p(\theta)$ as

$$\langle f, g \rangle_\theta = \mathbb{E}\{f(\theta) g(\theta)\} = \int_\Theta f(\theta) g(\theta) p(\theta) d\theta, \tag{16}$$

for any functions f and g. We can also define a corresponding norm $\|f\|_\theta = \langle f, f \rangle_\theta^{1/2}$ using the definition of the inner product. Now, let $L_\theta^2 = \{f : \|f\|_\theta < \infty\}$ denote the Hilbert space of square integrable functions with respect to density $p(\theta)$. Thus, $G \in L_\theta^2$ is a necessary condition for (15) to converge as $L \to \infty$ and is equivalent to the random variable $G(\theta, d)$ having finite variance.

The basis functions can be any complete basis of L_θ^2; however, the computation of the expansion coefficients can be simplified by choosing the basis to be orthogonal with respect to $p(\theta)$. Thus, let $\Psi_1, \Psi_2, \dots$ be a polynomial orthonormal basis (ONB) of L_θ^2, i.e., each element Ψ_i is a polynomial and for all $i, j \geq 1$ we have

$$\langle \Psi_i, \Psi_j \rangle_\theta = \delta_{ij}, \tag{17}$$

where δ_{ij} is the Kronecker delta. In practice, the ONB is constructed to have the following properties: (i) the first polynomial is a constant $\Psi_1(\theta) = 1$ meaning $\mathbb{E}\{\Psi_i(\theta)\} = \delta_{1i}$ is a convenient expression for the expectation of polynomials and (ii) each polynomial Ψ_i contains exactly one monomial θ^α that is not contained in the previous set of polynomials $\Psi_1, \dots, \Psi_{i-1}$. Most often the polynomials are ordered by degree. Therefore, when approximating G as in (15), we first select the number of terms L and define the ansatz space $\mathcal{P}$ as the span of the first L polynomials

$$\mathcal{P} = \{\Psi_1, \dots, \Psi_L\}. \tag{18}$$

Note that the size of the expansion L can be chosen to include polynomials of any order, but L is most often chosen according to a "total order" truncation in which all polynomials with degree less than or equal to n_o are retained. This results in the total number of terms in (15) being equal to

$$L = \binom{n_\theta + n_o}{n_o} = \frac{(n_\theta + n_o)!}{n_\theta ! n_o !}, \tag{19}$$

which grows exponentially with respect to the number of uncertainties and the maximum order of polynomials in the expansion.

A variety of methods exist for numerically constructing the polynomial ONB. As mentioned earlier, the approach in [26] (that is applied in the context of Bayesian OED in [15,25]) assumes separable $p(\theta) = p(\theta_1) \cdots p(\theta_{n_\theta})$, i.e., statistically independent elements of $\theta = (\theta_1, \dots, \theta_{n_\theta})$, so that the construction of the ONB can be done for each dimension separately. These polynomials have been analytically derived for certain scalar probability densities coming from the Askey scheme, and can be derived numerically for generic distributions using algorithms based on three-term recurrence relations for univariate orthogonal polynomials [37]. Whenever θ is composed of statistically dependent elements, a more sophisticated numerical procedure is required to construct the ONB. One example is the Gram-Schmidt process, which is capable of orthonormalizing any starting basis of $\mathcal{P}$, such as the set of monic polynomials (see, e.g., [38] for details). An alternative method is based on a modified Cholesky decomposition of the Gram moment matrix [39], which has shown to be reasonably stable on a variety of examples in [40]. In any case, it is sufficient to know the statistical moments of θ up to a certain order to construct the polynomial ONB. This is an advantage of expanding in θ directly (as opposed to transforming θ into a set of independent random variables), as we are only required to know moments of θ as opposed to an exact expression for $p(\theta)$ [41].

Note that there do exist density functions for which L_θ^2 does not admit an ONB of polynomials, i.e., the space of polynomials is not dense in L_θ^2. Interested readers are referred to [42] for more details on this aspect and a list of sufficient conditions to verify the denseness of polynomials. However, since knowing θ is continuous with finite support is sufficient for the space of polynomials to be dense in L_θ^2 [42,Theorem 3.4], this will rarely be an issue in practice.

4.2. Convergence, optimality, and error analysis

Since $G \in L_\theta^2$ by assumption, we are able to expand it with respect to the ONB of polynomials $\{\Psi_1, \Psi_2, \dots\}$

$$G(\theta, d) = \sum_{i=1}^{\infty} a_i(d) \Psi_i(\theta), \tag{20}$$

where the equality sign in (20) should be interpreted in the mean-square sense [42], such that

$$\lim_{L \to \infty} \mathbb{E}\{(G(\theta, d) - G_L(\theta, d))^2\} = \lim_{L \to \infty} \|G - G_L\|_{L_\theta^2}^2 = 0. \tag{21}$$

In other words, the PCE (15) exhibits mean-square convergence. Standard probability theory states that mean-square convergence implies convergence in probability and also convergence in distribution, i.e., $F_{G(\theta,d)}(x) = \lim_{L \to \infty} F_{G_L(\theta,d)}(x)$ for all $x \in \mathbb{R}$ where F_X denotes the cumulative distribution function (CDF) of any random variable X. The rate of convergence depends on the regularity of G with respect to θ and, when G is a smooth function of θ, the convergence rate can be quite large. This means that high accuracy can be achieved in practice with a relatively low order expansion.

According to the Hilbert projection theorem, the best $\|\cdot\|_\theta$ approximation of G in the polynomial space $\mathcal{P}$ is the *orthogonal projection* of G onto $\mathcal{P}$ [43]. This statement can be given mathematically in terms of the optimality condition

$$G_L = \text{argmin}_{P \in \mathcal{P}} \quad \|G - P\|_\theta^2, \tag{22}$$

such that no other choice of coefficients $a_1, \dots, a_L$ will result in a smaller weighted L_θ^2 norm. Since the weight function in (22) is the density $p(\theta)$, the optimal expansion G_L must more closely match G in regions of Θ where the parameter has high probability in order to ensure this norm is small.

Whenever G_L is numerically calculated, we only find approximations to the expansion coefficients and thus obtain the following approximated polynomial

$$\tilde{G}_L(\theta, d) = \sum_{i=1}^{L} \tilde{a}_i(d) \Psi_i(\theta). \tag{23}$$

As such, the difference between G and $\tilde{G}_L$ can be split into two terms: a truncation error and an aliasing error [44]

$$G - \tilde{G}_L = \underbrace{G - G_L}_{\text{truncation error}} + \underbrace{G_L - \tilde{G}_L}_{\text{aliasing error}} = \sum_{i=L+1}^{\infty} a_i \Psi_i + \sum_{i=1}^{L} (a_i - \tilde{a}_i) \Psi_i. \tag{24}$$

According to the orthogonality property of the ONB (17), these two sources of error are orthogonal such that their squared L_θ^2 norm is additive

$$\mathbb{E}\{(G(\theta, d) - G_L(\theta, d))^2\} = \|G - G_L\|_{L_\theta^2}^2 = \sum_{i=L+1}^{\infty} a_i^2 + \sum_{i=1}^{L} (a_i - \tilde{a}_i)^2. \tag{25}$$

Since the ansatz space $\mathcal{P}$ is fixed, the truncation error is constant for a fixed forward model. This is directly controlled by the choice of L, i.e., larger L leads to lower truncation error and solely depends on the nonlinearity of G. Thus, different methods for approximating these expansion coefficients can easily be compared by the aliasing error that they introduce.

Another important property of PCE is that the moments of the random variable $G(\theta, d)$ can be easily computed from only the expansion coefficients

$$\mathbb{E}\{G(\theta, d)\} = a_1(d) \approx \tilde{a}_1(d), \tag{26}$$

$$\text{Var}\{G(\theta, d)\} = \sum_{i=2}^{\infty} a_i(d)^2 \approx \sum_{i=2}^{L} \tilde{a}_i(d)^2. \tag{27}$$

These equations can be straightforwardly substituted into the mean-variance chance constraint approximation (12) so that it can be expressed simply in terms of the PCE coefficients for $H(t, \theta, d)$.

4.3. Estimation of PCE coefficients

There are two main approaches for approximating the expansion coefficients: intrusive and non-intrusive [45]. The intrusive approach derives a new system of equations for the coefficients that is larger than the original deterministic system [46]. The difficulty of the intrusive approach strongly depends on the character of the original equations and is often prohibitive (or even impossible to derive) for nonlinear systems [47]. Non-intrusive methods, on the other hand, compute the expansion coefficients from only a finite number of parameter realizations [48]. The main advantage of these approaches is that a deterministic solver for $G(\theta, d)$ can be reused and treated as a black box. Non-intrusive methods also offer flexibility in choosing any function of the state trajectory as the model output, which may depend more smoothly on θ even when the state itself has less regular dependence. In other words, we can avoid representing $x(t;d, \theta)$ with a PCE and instead directly apply the method to $G(\theta, d)$.

Here, we use a non-intrusive approach for estimating the PCE coefficients. The derivation of the method starts from the fact that, by taking the inner product of the expansion (20) with one of the basis functions Ψ_i and applying the orthogonality property of the ONB, we obtain an analytic expression for the coefficients

$$a_i(d) = \langle G, \Psi_i \rangle_{L^2_\theta} = \int_\Theta G(\theta, d)\Psi_i(\theta)p(\theta)d\theta, \quad i = 1, \ldots, L. \tag{28}$$

Then, a set of n sample points $\theta^{(1)}, \ldots, \theta^{(n)} \in \Theta$ is chosen and the integrals in (28) are approximated using a finite number of forward model evaluations according to some chosen quadrature (or integration) rule [44]

$$\tilde{a}_i(d) = \sum_{j=1}^{n} w_j G(\theta^{(j)}, d)\Psi_i(\theta^{(j)}), \tag{29}$$

where $w_1, \ldots, w_n$ are corresponding weight values in the quadrature rule. The resulting approach is termed pseudo-spectral projection as it defines a mapping between the forward model G and polynomial $\tilde{G}_L$ that is a *discretized projection operator*.[1] If a convergent integration rule is employed such that $\lim_{Q\to\infty} \tilde{a}_i = a_i$, then $\lim_{Q\to\infty} \tilde{G}_L(\theta, d) = G_L(\theta, d)$ for all $\theta \in \Theta$ and convergence of $\tilde{G}_L$ to the true forward model G follows naturally.

The key step in any non-intrusive PCE method is the selection of integration points and weights to be used to approximate the coefficients. The number of points n should be as small as possible to achieve a desired level of accuracy in the PCE approximation (23). A wide variety of integration (or sampling) rules for multidimensional spaces have been proposed and applied in the context of PCE. Broadly speaking, these methods can be categorized as follows: (i) grid-based, (ii) randomized, (iii) monomial cubature rules, or (iv) optimization-based.

Grid-based methods such as tensor and sparse grids [50] are the most commonly used integration rules since they can be easily derived from univariate Gaussian quadrature rules, which are optimal in one dimension [51]. However, tensor-grid quadrature suffers from the *curse of dimensionality* due to the exponential growth of the number of points with dimension of the parameter space. Sparse grids are directly constructed from tensor grids and are built to accurately capture functional features in each separate parameter dimension while investing fewer points in the cross terms between parameters. Although sparse grids have fewer points than the full tensor grid, they have increasingly large error with increasing dimension and are known to produce negative weight values. Another key limitation of tensor and sparse grids is that they require the uncertain parameters to be statistically independent. If the parameters are dependent, then a transformation must be applied, which may place integration points in low probability regions of Θ that contribute only a very small amount to the PCE projection. Randomized integration rules, on the other hand, select points by randomly sampling from the parameter distribution $p(\theta)$ via MC methods [52]. MC is often the method of choice for approximation of high-dimensional integrals, but are known to require a large number of points to achieve low error due to their relatively slow rates of convergence.

Monomial cubature rules are nongrid-based methods that can be more effective than sparse grids when integrating functions that are well represented by total-degree polynomials [53]. These can be thought of as efficient multivariate extensions of Gaussian quadrature. Their main downside, however, is that effective cubature rules have only been constructed for a very specific set of probability distributions, integration domains, and polynomial degree of exactness. Optimization-based integration rules are based on the same idea as monomial cubature rules, with the main difference being that the quadrature rule is not selected manually. Instead, the integration points and weights are determined numerically through the use of some optimization procedure. In this way, efficient quadrature rules can be constructed for any distribution $p(\theta)$ and any desired polynomial degree. Also, constraints on the position of the points and value of the weights can readily be incorporated.

The main cost of non-intrusive PCE arises from the forward model simulations at fixed nodes n, and these simulations must be repeated for every d visited when numerically solving the Bayesian OED problem (8). Thus, we adopt the optimization-based methodology here so that n can be minimized without compromising accuracy of the integration rule. There are two main types of optimization-based methods available: moment matching and optimized stochastic collocation (OSC). The moment-matching rule corresponds to a non-negative measure on Θ that minimizes a sensitivity function subject to the constraints that the measure matches moments of $p(\theta)$ up to a certain finite order [54]. This corresponds to an infinite-dimensional linear program (LP) that must be heuristically solved in practice. One approach, presented in [39], is based on three steps: (i) solve a finite-dimensional LP wherein moments are matched based on a fine grid of Θ, (ii) locate the “clusters” of sample points obtained from the LP solution, and (iii) refine this solution by locally solving a nonlinear least-squares problem with initial guess corresponding to the clustered integration rule. However, the derived moment matching rule can be sensitive to the choice of the initial grid and the clustering step. The OSC method, on the other hand, derives the optimal points and weights through

[1] Regression methods are an alternative class of non-intrusive PCE in which the discrete quadrature rule is directly applied to the optimality condition in (22), and these can straightforwardly be used in place of pseudo-spectral methods in this work [49].

the formal minimization of an integration operator error norm [49]. Here, we adopt the OSC method since it limits the number of heuristic choices by the user and has been shown to effectively handle uncertainty dimensions up to around ten.

4.4. The optimized stochastic collocation method

The OSC method is summarized in this subsection. OSC is formulated as a polynomial optimization problem with an objective function that is adapted to be able to efficiently and accurately approximate the PCE coefficients. First, define the exact integral operator for a generic function $f \in L^2_\theta$ as

$$I : L^2_\theta \to \mathbb{R} : f \mapsto \int_\Theta f(\theta)p(\theta)d\theta, \tag{30}$$

while, for a given list of points $\boldsymbol{\theta} = (\theta^{(1)}, \ldots, \theta^{(n)})$ and weights $w = (w_1, \ldots, w_n)$, the discrete quadrature operator is defined as

$$\mathrm{Q}_{(\boldsymbol{\theta},w)} : L^2_\theta \to \mathbb{R} : f \mapsto \sum_{j=1}^{n} w_j f(\theta^{(j)}). \tag{31}$$

The operators I and $\mathrm{Q}_{(\boldsymbol{\theta},w)}$ must be bounded in order to define an operator norm to measure the distance between them. Thus, we restrict I and $\mathrm{Q}_{(\boldsymbol{\theta},w)}$ to a finite-dimensional test space $\mathcal{T} \subseteq L^2_\theta$. Let $\mathfrak{L}(\mathcal{T}, \mathbb{R})$ denote the space of all bounded linear operators from $\mathcal{T}$ to $\mathbb{R}$. For any operator $A \in \mathfrak{L}(\mathcal{T}, \mathbb{R})$, the induced operator norm on the space $\mathfrak{L}(\mathcal{T}, \mathbb{R})$ is defined as:

$$\|A\|_{\mathfrak{L}(\mathcal{T},\mathbb{R})} = \sup_{f \in \mathcal{T}} \frac{\|Af\|_{\mathbb{R}}}{\|f\|_\theta}. \tag{32}$$

The OSC method can then be summarized using this induced norm as follows:

1. Choose a finite-dimensional test space $\mathcal{T}$ and number of integration points n.
2. Find the optimal integration points and weights by solving

$$(\boldsymbol{\theta}_{\text{osc}}, w_{\text{osc}}) = \underset{\substack{\boldsymbol{\theta} \in \Theta^n \\ w \in [0,\infty)^n}}{\operatorname{argmin}} \quad \|I - \mathrm{Q}_{(\boldsymbol{\theta},w)}\|^2_{\mathfrak{L}(\mathcal{T},\mathbb{R})}. \tag{33}$$

Note that there are some basic relationships between $\mathcal{T}$ and n. Mainly, the number of integration points is bounded by $dim(\mathcal{P}) = L \le n \le t = dim(\mathcal{T})$ since $n < L$ points cannot even distinguish the L different ansatz functions and t points are always capable of reducing the operator error norm to zero [49].

A good choice for the test space $\mathcal{T}$ can be derived from the integrals that we want to approximate in (29). Whenever $G \in \mathcal{P}$, then we would like $\tilde{G}_L = G$, i.e., there is no truncation or aliasing error for polynomial models within the ansatz space $\mathcal{P}$ (18). This means that the integral of all products of two elements in $\mathcal{P}$ have to be exact, which corresponds to the test space

$$\mathcal{T} = \text{span}\{\Psi_i\Psi_j,\ 1 \le i, j \le L\}. \tag{34}$$

Based on this choice of $\mathcal{T}$, it is then desired to choose n large enough so that the operator norm is reduced to zero. A simple procedure can be derived from the degrees of freedom (DOF) in the optimization problem (33). The number of DOF in the optimization is $n(n_\theta + 1)$. In order to reduce the objective function in (33) to zero, t equations must be satisfied. Therefore, if we choose $\mathcal{T}$ and n such that

$$t = n(n_\theta + 1), \tag{35}$$

then we may have enough integration points to be able to satisfy all t conditions. Since t is fixed according to (34), we should initially select $n = t/(n_\theta + 1)$, which is much lower than the upper bound of t. However, it is important to note that this is a heuristic choice and cases exist that n has to be larger or can be chosen smaller. Thus, a practical approach is to first choose n according to the DOF condition (35) and then numerically perform the optimization (33). If the minimum objective value is not small enough, then n can be increased by one and the optimization repeated until the objective has been reduced to a sufficiently low value. Note that alternative choices of $\mathcal{T}$ and n are discussed in [49].

Remark 1. Whenever L is chosen using the "total order" truncation method with maximum order n_o, then the choice of test space in (34) effectively doubles the PCE order such that the dimensionality of the test space is $t = \frac{(n_\theta + 2n_o)!}{n_\theta! 2n_o!}$.

We can now derive an expression for the operator norm in (33) explicitly in terms of the integration points $\boldsymbol{\theta}$ and weights w. Since the elements of $\mathcal{T}$ in (34) are polynomials, they can be represented as coordinate vectors with respect to the ONB, i.e., any function $f \in \mathcal{T}$ can be written as $f = \sum_{i=1}^{t} c_i\Psi_i$. Based on this representation, the numerator of the induced norm can be written as

$$\|Af\|_{\mathbb{R}} = \|A(\sum_{i=1}^{t} c_i\Psi_i)\|_2 = \|\sum_{i=1}^{t} c_i A\Psi_i\|_2 \le \|c\|_2\|A\Psi\|_2, \tag{36}$$

where $c = (c_1, \ldots, c_t)$ and $A\Psi = (A\Psi_1, \ldots, A\Psi_t)$ denotes the vector representation of any operator A with respect to the ONB Ψ. The inequality above directly follows from the well-known Cauchy–Schwartz inequality. Similarly, we can apply this representation to the squared denominator of the induced norm to derive

$$\begin{aligned} \|f\|^2_\theta &= \int_\Theta \left(\sum_{i=1}^{t} c_i\Psi_i(\theta) \right)^2 p(\theta)d\theta \\ &= \sum_{i=1}^{t}\sum_{j=1}^{t} c_i c_j \langle \Psi_i, \Psi_j \rangle_\theta = \sum_{i=1}^{t} c_i^2 = \|c\|^2_2. \end{aligned} \tag{37}$$

Since the supremum is achieved when the inequality exactly holds, we can combine these two expressions to derive a finite-dimensional representation of the operator norm as the 2-norm of its vector representation, i.e., $\|A\|_{\mathfrak{L}(\mathcal{T},\mathbb{R})} = \|A\Psi\|_2$.

We recall (17) to find that $I\Psi_i = \delta_{1i}$, meaning that the vector representation of I can be written as

$$I\Psi = \mathrm{e}_1 = (1, 0, \ldots, 0). \tag{38}$$

For $\mathrm{Q}_{(\boldsymbol{\theta},w)}$, from (31), we find that

$$\mathrm{Q}_{(\boldsymbol{\theta},w)}\Psi_i = \sum_{j=1}^{n} w_j \Psi_i(\theta^{(j)}), \tag{39}$$

which can be easily converted into its vector representation that is an explicit function of the integration points and weights:

$$\mathrm{Q}_{(\boldsymbol{\theta},w)}\Psi = \boldsymbol{\Psi}(\boldsymbol{\theta})w, \tag{40}$$

where $\boldsymbol{\Psi}(\boldsymbol{\theta})$ is a $t \times n$ matrix:

$$\boldsymbol{\Psi}(\boldsymbol{\theta}) = \begin{bmatrix} \Psi_1(\theta^{(1)}) & \cdots & \Psi_1(\theta^{(n)}) \\ \vdots & \ddots & \vdots \\ \Psi_t(\theta^{(1)}) & \cdots & \Psi_t(\theta^{(n)}) \end{bmatrix}. \tag{41}$$

We can then calculate the squared operator norm of $I - \mathrm{Q}_{(\boldsymbol{\theta},w)}$ as

$$\|I - \mathrm{Q}_{(\boldsymbol{\theta},w)}\|^2_{\mathfrak{L}(\mathcal{T},\mathbb{R})} = \|\mathrm{e}_1 - \boldsymbol{\Psi}(\boldsymbol{\theta})w\|^2_2, \tag{42}$$

which is a sum of squares of polynomial functions. This is a smooth function with structure that can be easily exploited by gradient-based optimization algorithms. An important practical issue in the OSC method (33) is finding the global minimum. Since the lowest attainable value of the objective is known to be zero, we are guaranteed to have found a global optimum as long as this bound is reached.

4.5. Global versus local PCE with respect to the design space

PCE is simply an orthogonal polynomial approximation to random functions and thus can be applied to G in various ways. For example, in [15] a *single* (or global) PCE is constructed for $G(\theta, d)$ over the entire product of the parameter and design spaces. In this way, a random vector $\xi \in \mathbb{R}^{n_s}$, $n_s = n_\theta + n_d$ is defined to have one dimension associated to each component of θ and one to each component of d. The density $p(\xi)$ is required to be separable and is assumed to map to the joint space $(\theta, d) = T(\xi)$ based on some (possibly) nonlinear diffeomorphism $T : \mathbb{R}^{n_s} \rightarrow \Theta \times \mathcal{D}$ that preserves the probability density functions of ξ and (θ, d). The global PCE can then be defined similarly to (15), except now in terms of this new random vector ξ

$$G(\theta(\xi), d(\xi)) \approx \sum_{i=1}^{L} b_i \Phi_i(\xi), \tag{43}$$

where $b_1, \ldots, b_L$ are the global expansion coefficients and $\Phi_1, \ldots, \Phi_L$ are polynomials that are orthogonal with respect to $p(\xi)$.

The main advantage of this approach is that the coefficients are constant and therefore only need to be computed once before solving the Bayesian OED problem; however, there are two important limitations. First, the number of terms in (43) increases exponentially with $n_s = n_\theta + n_d$ and the truncated order n_o. The effect of this growth is twofold: time-varying trajectories $d(t)$ must be heavily discretized in order to keep n_d small, and accuracy must be sacrificed when G is highly nonlinear in d to keep n_o small. Second, we must select some probability distribution for the design variables. This distribution represents the weight function that governs what regions of $\mathcal{D}$ that the PCE should be most accurate. Therefore, the probability distribution should be proportional to how often values of d are visited during the optimization algorithm. Since this quantity is too complex to extract in practice, a heuristic strategy must be applied instead. For example, in [15], a uniform weight function over the bounded design space is chosen. As a result, the surrogate might be inaccurate near the unknown optimal design.

The proposed PCE-based surrogate (15) avoids both of these issues by developing a *local* surrogate around each design encountered during the optimization. Therefore, the size of the surrogate is completely independent of n_d, and we do not need to artificially define a distribution over the design space. Although the coefficients must be updated at every iteration in the proposed approach, the OSC rule used to define the quadrature operator in (29) ensures that this process only requires a minimal number of forward model simulations. This means that we can significantly reduce the number of full model evaluations in (14) from $O(NM)$ to merely n. These features suggest that the proposed approach is especially advantageous in dynamic systems, which can very easily result in OED problems with n_d on the order of tens to hundreds of independent variables.

The final important difference between (15) and (43) is related to the essence of the so-called germ ξ. The *generalized polynomial chaos* (gPC) method requires the stochastic parameters to be statistically independent in order to simplify the basis construction. As long as T is a density-preserving transformation, then ξ can be chosen as any set of independent random variables. The Rosenblatt transformation is the most common example as it applies to any collection of continuous random variables [55]. However, a known problem with the Rosenblatt transformation is that, even for simple problems, T can quickly become discontinuous and highly nonlinear. In fact, it has been shown that transformations between some standard scalar random variables exhibit Gibbs phenomena and thus deteriorate the convergence rate of the expansion [44]. Even when this transformation is reasonably well-behaved, it can be complicated to determine and expensive to evaluate. Therefore, it is preferred to expand in terms of θ when possible. This implementation of PCE has been referred to as *arbitrary polynomial chaos* (aPC) since there are no restrictions on $p(\theta)$, and can be interpreted as a generalization of gPC. We explicitly represent (15) using aPC because this helps keep n_o small, which directly results in lower values for n due to smaller-sized test spaces (34).

5. Numerical results

5.1. The dynamic forward model

The well-known Lotka–Volterra (LV) system has been used to model the nonlinear and oscillatory dynamics of interacting predator and prey populations, and is a commonly used benchmark problem in the dynamic OED literature, e.g., [56]. The time-evolution of LV system is governed by the ODEs

$$\dot{x}_1(t) = x_1(t) - (1 + 0.25\theta_1)x_1(t)x_2(t) - 0.4x_1(t)d(t) \tag{44a}$$

$$\dot{x}_2(t) = -x_1(t) + (1 + 0.25\theta_2)x_1(t)x_2(t) - 0.2x_2(t)d(t), \tag{44b}$$

where $t \in [0, t_f]$ is the time variable with $t_f = 12$, $x_1(t)$ is the normalized prey population, $x_2(t)$ is the normalized predator population, and θ_1 and θ_2 are the unknown parameters for which we have limited information. The design profile $d(t)$ can be manipulated throughout the experiment, and is constrained to the domain $d(t) \in [0, 1]$ for all $t \in [0, t_f]$. We also assume a noisy measurement of the predator population can be made at the final time, i.e., $y = x_2(t_f) + \epsilon$. The noise is modeled as a zero-mean Gaussian random variable with standard deviation $\sigma = 0.1|x_2(t_f)|$, i.e., the noise variance is *state-dependent* and equals 10% of the signal. For this study, we select a statistically-dependent prior in terms of two coupled beta distributions

$$\theta_1 \sim \mathcal{B}(2, 2), \qquad \theta_2 | \theta_1 \sim \mathcal{B}(\theta_1 + 3, -\theta_1 + 2). \tag{45}$$

A contour plot of this joint distribution on the support $(\theta_1, \theta_2) \in [-1, 1]^2$ is shown in Fig. 1. This prior was chosen as an example of one that is able to capture potential relationships between parameters.

The OED formulation (8) seeks the design $d^\star(t)$ such that, when the experiment is performed, on average the predator signal yields the greatest information gain from prior to posterior, i.e., the information gain is averaged over all possible prior parameters and over all possible resulting measured predator populations. State chance constraints are added to the problem after the initial comparisons.

Note that all NLP optimization problems discussed in this case study are numerically solved using CasADi [57] that automatically passes the required derivatives (based on a symbolic implementation of the equations) to the interior point solver IPOPT [58]. In addition, all computations were performed on a MacBook Pro with 8 GB of RAM and a 2.6 GHz Intel i5 processor.

5.2. Local and global PCE implementations

Evaluating the forward model requires solving the ODE in (44) at fixed realizations of θ and extracting the predator population at the final time. These equations are integrated with CVODE [59] set to a tolerance of 10^{-8}, which is an error-controlled solver for stiff and non-stiff initial value problems. As discussed in Section 4,

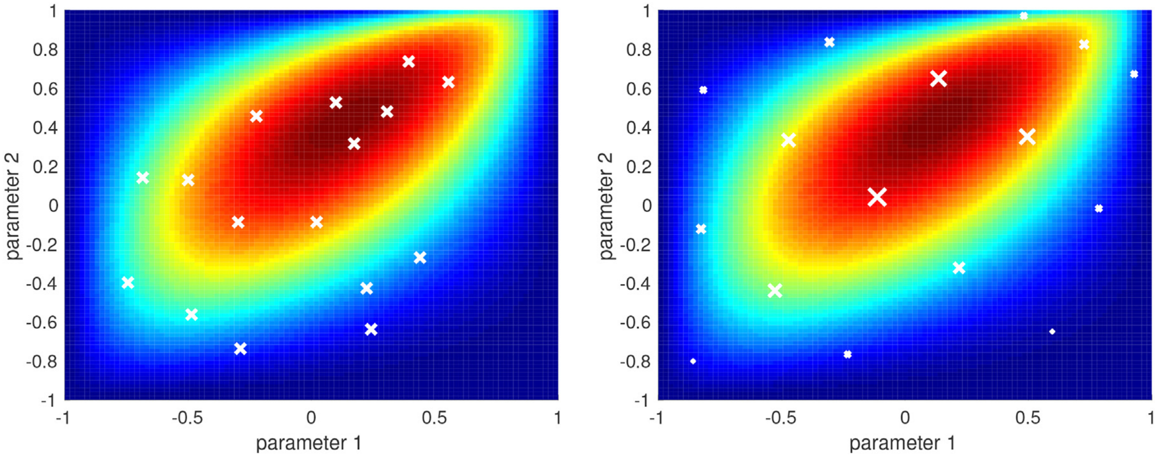


Fig. 1. Randomly sampled initial condition provided to OSC optimization (33) (left) and derived optimal OSC quadrature rule with $n = 16$ nodes that can exactly integrate $t = 45$ polynomials (right). The nodes are shown with a white 'x', and the size of each node is proportional to its weight. The colors represent contours of the joint parameter distribution in (45). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the full forward model can be replaced with a PCE-based surrogate to improve computational efficiency of the MC estimator for the expected utility. In this example, we compare two surrogate modeling approaches: (i) the proposed method that we will refer to as "local" or "design-dependent" PCE for short and (ii) the "global" PCE method proposed in [15,25].

The local PCE method expands in terms of θ directly, meaning the polynomials must be constructed to be orthogonal to (45). This was done by applying the modified Cholesky decomposition to the Gram moment matrix, composed of moments of a finite number of moments of θ. The coefficients of the expansion are estimated using a quadrature rule chosen as the solution to the OSC problem (33) with a test space $\mathcal{T}$ composed of all polynomials up to degree 8, which corresponds to $t = dim(\mathcal{T}) = \frac{(8+2)!}{8!2!} = 45$. Based on the DOF condition in (35), n was initially set equal to 15; however, this did not produce satisfactory objective values near zero and so was increased by one. For $n = 16$ points, we were able to consistently find solutions to (33) that result in a norm of zero (global optimum) starting from an initial condition with equal weights and nodes sampled randomly from $p(\theta)$. This indicates the OSC rule can integrate 45 polynomials exactly using only 16 points. One such example of a converged OSC rule is shown in Fig. 1. This rule matches intuition as points with larger weights are concentrated in regions of the density function with higher values. Note that the OSC problem took about 2 seconds on average to solve and more than 90% of the runs converged to the global optimum.

Global PCE, on the other hand, expands with respect to the joint parameter and design space, meaning the design space must be discretized before it can be applied. To this end, the design profile is discretized into N_T piecewise constant intervals $[0, t_f] = [t_0, t_1) \cup \cdots \cup [t_{N_T-1}, t_{N_T})$ such that $d(t) = d_i$ for all $t \in [t_i, t_{i+1})$ and $i = 1, \ldots, N_T$. For comparison purposes, we fix the number of intervals at $N_T = 2$. We thus have two additional variables (d_1, d_2) to include in the PCE, which we assume are independent and uniformly distributed in order to build the surrogate (43). We again note that this is heuristic choice, as we do not know the distribution of designs that will be visited during the optimization procedure. The parameter and design variables must then be mapped to a 4-dimensional germ $\xi = T(\theta_1, \theta_2, d_1, d_2)$ that has statistically independent elements. In this case, we choose $\xi_1 \sim \beta(2, 2)$, $\xi_2 \sim \beta(2, 2)$, $\xi_3 \sim \mathcal{U}(-1, 1)$, $\xi_4 \sim \mathcal{U}(-1, 1)$, and T according to the Rosenblatt transformation. The coefficients of the global PCE were determined with a tensor product of Gaussian quadrature rules of order 10 that resulted in a total of 10^4 forward model evaluations.

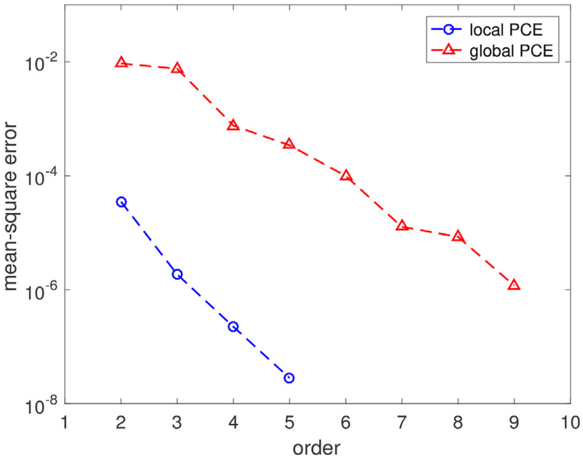


Fig. 2. The mean-square error (MSE), i.e., L^2_θ norm versus truncation order of the local and global PCE surrogates for the forward model in the LV system.

Both the local and global PCE method are implemented using total-order polynomial truncation. In order to select this truncation order, we calculated the L^2_θ mean-squared error (MSE) for various truncation orders, which is plotted in Fig. 2. We clearly observe that the error decreases as order increases for both methods; however, the local expansion exhibits a faster rate of convergence and has errors more than an order-of-magnitude lower than the global approach. This is not surprising as local PCE does not expand in the design space so that it can directly capture nonlinear effects with respect to d. For global PCE, we selected order 9 as this resulted in reasonably small MSE while retaining less than one thousand terms in the expansion, i.e., $L = 715$. Based on this choice, we selected order 4 for local PCE, which corresponds to $L = 15$, as this is the smallest order with lower MSE than global PCE of order 9.

5.3. Stochastic dynamic optimization implementation and results

We now discuss the implementation of the proposed SAA optimization (14) and analyze the results for varying number of samples. We first focus on the local PCE method and then compare performance to the global PCE method.

Regardless of the choice of sample sizes N and M, the local PCE method must impose n separate ODE constraints corresponding to the nodes of the OSC rule. A direct transcription approach was used to discretize the state profile in time using collocation of finite elements. We chose 20 elements and used a third-order collocation scheme within each element. Note that the design profile is only discretized into 2 elements in order for the local method to be more fairly compared with the global approach. As such, the dynamic optimization problem has been converted to a large-scale NLP that again can readily be solved using CasADi and IPOPT. We chose to use the limited-memory BFGS estimate of the Hessian, as opposed to exact evaluation of the Hessian, since this provided computational savings in this problem (i.e., cost per iteration decreased more than number of iterations increased).

Under SAA, each choice of sample sets θ_s and z_s yields a different deterministic objective. Example realizations of this objective surface are shown in Figs. 3–5. For each realization, a local optimum is found efficiently in only a relatively few (usually less than 15) iterations. Note that for low N, the objective realizations can be extremely different including the location of the optimum points as well as the estimated maximum value of the expected utility. In general, however, the objectives have less variability as N is increased. Looking at $N = 101$ in Fig. 5, we consistently see two designs that (locally) maximize the objective and one design that minimizes the objective. To better understand the performance of the proposed method, we conducted 1000 independent bootstrap runs, over a matrix of N and M values. Each optimization is initialized with a uniformly distributed random design to assess the performance of the method on average. Histograms of the optimal design variables resulting from each set of 1000 optimization runs are shown in Table 1. We can immediately recognize that more designs cluster around the three local optima as N and M are increased. The distribution of final designs is not enough to understand the robustness of the optimization results. For example, if U is flat near the optimum, then the suboptimal designs need not be close to the true optimal design to be considered good designs in practice. A "high-quality" estimate of the objective $\hat{U}_{1001,1001}$ is computed for each of the 1000 designs in Table 1 to evaluate robustness, and the resulting histograms are shown in Table 2. We can again see that performance improves as N and M increase. It is interesting to note that all histograms in Table 2 are bimodal. The higher mode reflects a mixture of the two maxima while the lower mode corresponds to the minimum design. Although the variance in these modes decreases with increasing N and M, both modes are always present. Around 70% of runs converge to the high expected utility mode while 30% of the runs converge to the low mode. Note that similar features are observed when using global PCE as the surrogate model.

5.4. Comparison between local and global PCE surrogates

To compare the local and global PCE surrogates, we develop a single integrated measure of the quality of the solutions from the SAA optimization. As suggested in [25], we use the following MSE expression as this metric

$$\text{MSE} = \frac{1}{B}\sum_{b=1}^{B}\left(\hat{U}_{1001,1001}(\hat{d}^b, \theta_{s'}^b, z_{s'}^b) - U^{\text{ref}}\right)^2 \tag{46}$$

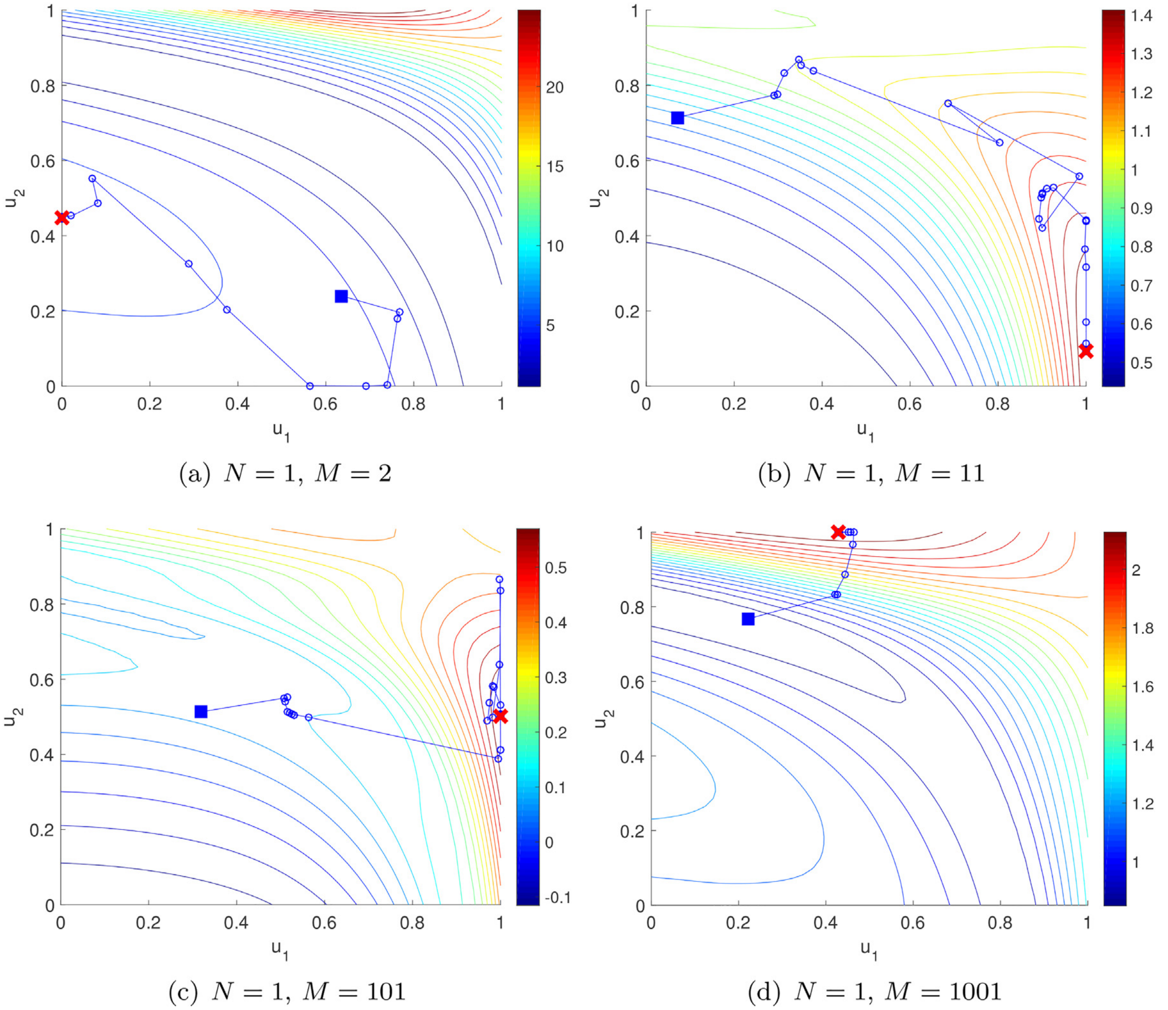
where $\hat{d}^b$, $b = 1, \ldots, B$ are the final designs from the optimization algorithm for $B = 1000$ bootstraps (using either local or global PCE) and U^{ref} is the true maximum expected information gain. Since the true maximum is not available in this case study, we take U^{ref} to be the maximum value of the objective over all runs. An important issue in the evaluation of the MSE is that it will be significantly biased by the designs that lead to the local minimum. To avoid this bias, we initialized all optimizations at [1, 1] in the design space that consistently produced designs that globally maximize the objective. Figure 6 describes the solution quality in relation to computational effort by plotting the MSE against average computational time per run for both the local and global PCE methods. Each symbol represents a particular value of N, i.e., ×, ○ and □ represent $N = 1$, $N = 11$, and $N = 101$, respectively, and the four different M values are shown through average run times.

There are a number of interesting observations within Fig. 6. Both methods generally result in lower MSE as N and M increase. We also observe that N has a much larger effect than M for this particular example. However, in the $N = 101$ case for the global method, we see that MSE is nearly constant as M increases. This is likely due to the fact that the global PCE has a larger L_θ^2 error than the local method and thus hits its lowest achievable error around this value for N. As expected, the global PCE method has solution times that are directly proportional to the total number of samples used to evaluate the expected utility since the cost of the model (43) is the same per sample. The local PCE method, on the other hand, requires a minimum of approximately 1 second to find a solution, regardless of N and M. This is due to the fact that the local method has a fixed cost corresponding to the $n = 16$ discretized ODE constraints. These nonlinear constraints are the dominant cost in the optimization when $N = 1$, $M = 2$ all the way to $N = 101$, $M = 101$. In other words, we see no increase in the approximately 1 second solution time when there are $NM + N = 3$ versus $NM + N = 10,302$ total evaluations of the polynomial (15) needed at each iteration. This highlights the importance of the minimal OSC rule as 16 forward model evaluations are more expensive than over 10,000 surrogate evaluations, meaning we can expect the solution time to massively increase if the full model is evaluated at each sample instead of the surrogate (if even possible to store all of the constraints in memory). We do see that the cost of the polynomial evaluations overtake the ODE cost for the largest considered case of $N = 101$, $M = 1001$. The local PCE method, however, is increasingly cheaper to evaluate than the global method as the number of samples increases. This is a direct consequence of the local expansion having a factor of 50 less terms in the expansion than the global method. In fact, the local method is 7.5 times cheaper than the global method in the largest case considered, while also producing solutions with two order-of-magnitudes lower MSE.

5.5. Scaling with respect to number of design variables

The previous analysis focused on the case of $N_T = 2$ discretized design variables. To understand the effect that N_T has on the optimization when using the proposed local PCE method, the average computational time to solve (14) (over ten independent runs with $N = 101$ and $M = 101$) is plotted against the number of design variables N_T in Fig. 7. We can clearly see that the cost scales sublinearly with respect N_T, which is mainly due to the fact that the state discretization level is fixed at 20 elements in all cases as well as sparsity being exploited in the gradient computation. It is important to note that, as N_T increases, the size of the surrogate model remains fixed and cost only increases due to the larger number of decision variables. This is in sharp contrast to the global PCE model, which grows exponentially in size as N_T increases. For example, the global expansion has more than two million terms for 20 design variables, 2 parameters, and a truncation order of 9. As such, we were unable to apply the global method for $N_T = 20$, while the local method only took approximately 65 seconds to find a solution.

High-quality expected utility estimates for each discretization level are also shown in Fig. 7. As expected, the optimal objective value increases as the number of design variables increases due to the fact that the design profile has more freedom. We see a large increase (more than 50%) in the optimal objective for $N_T = 5$ whereas fairly minor increases for larger N_T. This suggests that five intervals provide enough freedom in this problem to find a solution



(a) $N = 1,\ M = 2$

(b) $N = 1,\ M = 11$

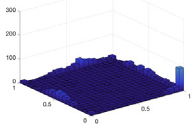
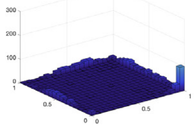
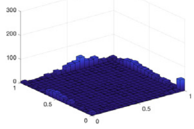
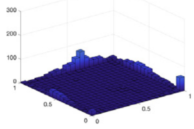
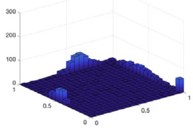
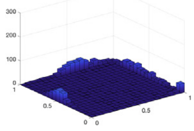
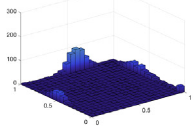
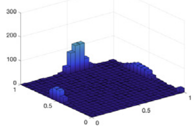
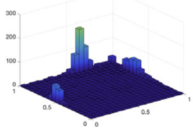
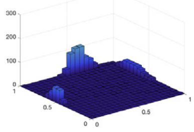
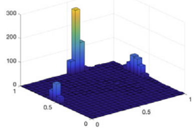
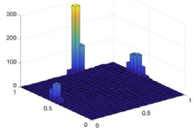
(c) $N = 1,\ M = 101$

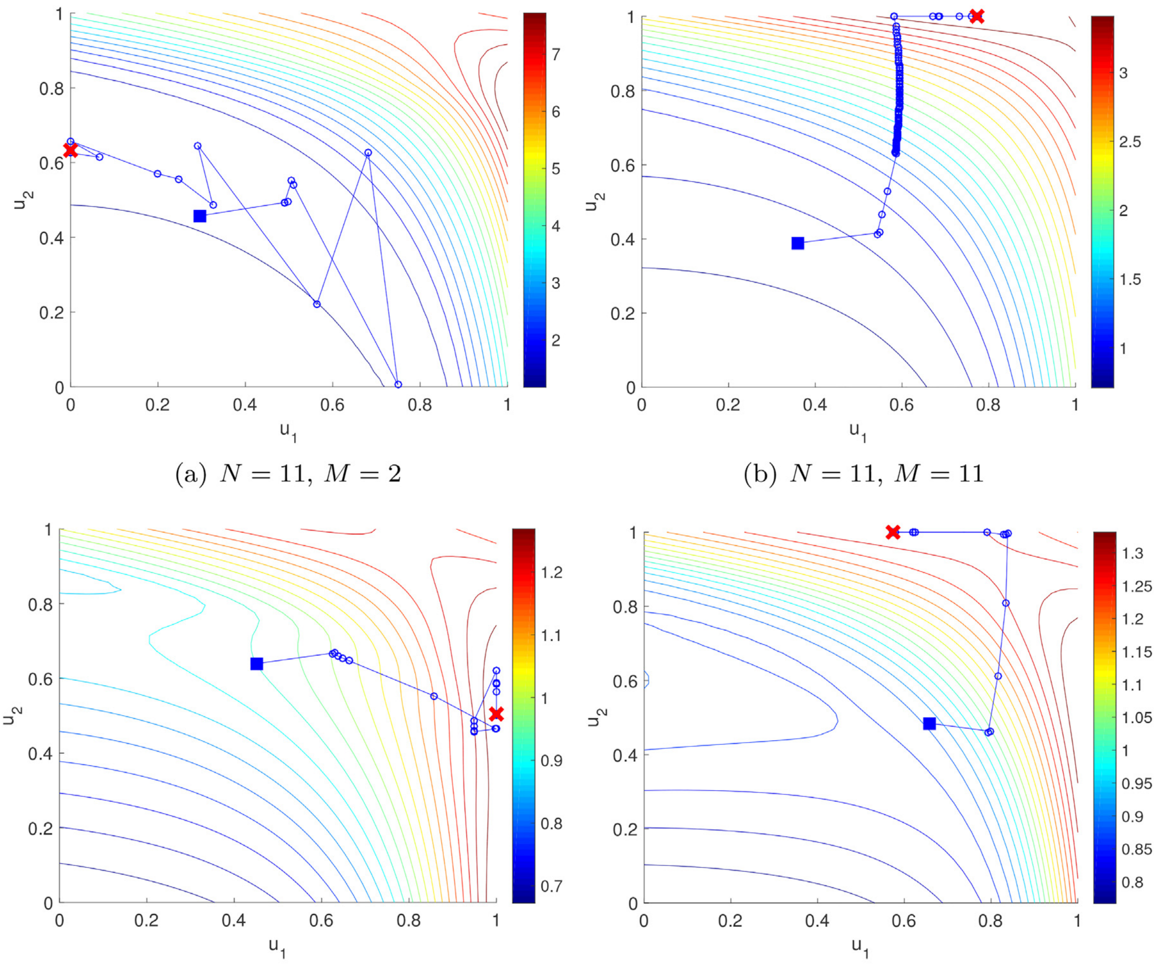
(d) $N = 1,\ M = 1001$

Fig. 3. Realizations of the objective surface using SAA and the corresponding iterations of IPOPT, with $N = 1$ and four separate M values. The blue □ is the starting point and the red × is the final converged point. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1

Histograms of the optimal design variables $\hat{d}_s$ from 1000 independent bootstrap runs of SAA over a matrix of N and M sample sizes. For each histogram, the bottom-right axis represents $d(t) = d_1$ for $t \in [0, t_f/2)$, the bottom-left axis represents $d(t) = d_2$ for $t \in [t_f/2, t_f)$, and the vertical axis represents frequency.

N \ M	2	11	101	1001
1				
11				
101				



(a) $N = 11,\ M = 2$

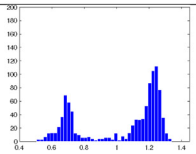
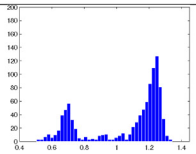
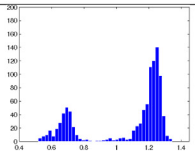
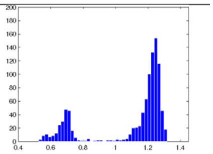
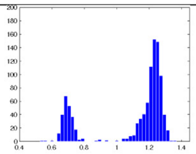
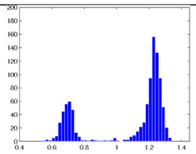
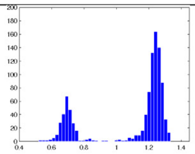
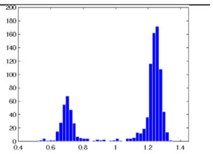
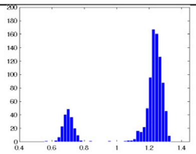
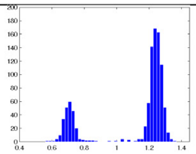
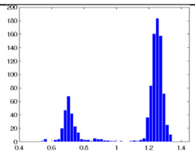
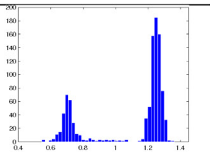
(b) $N = 11,\ M = 11$

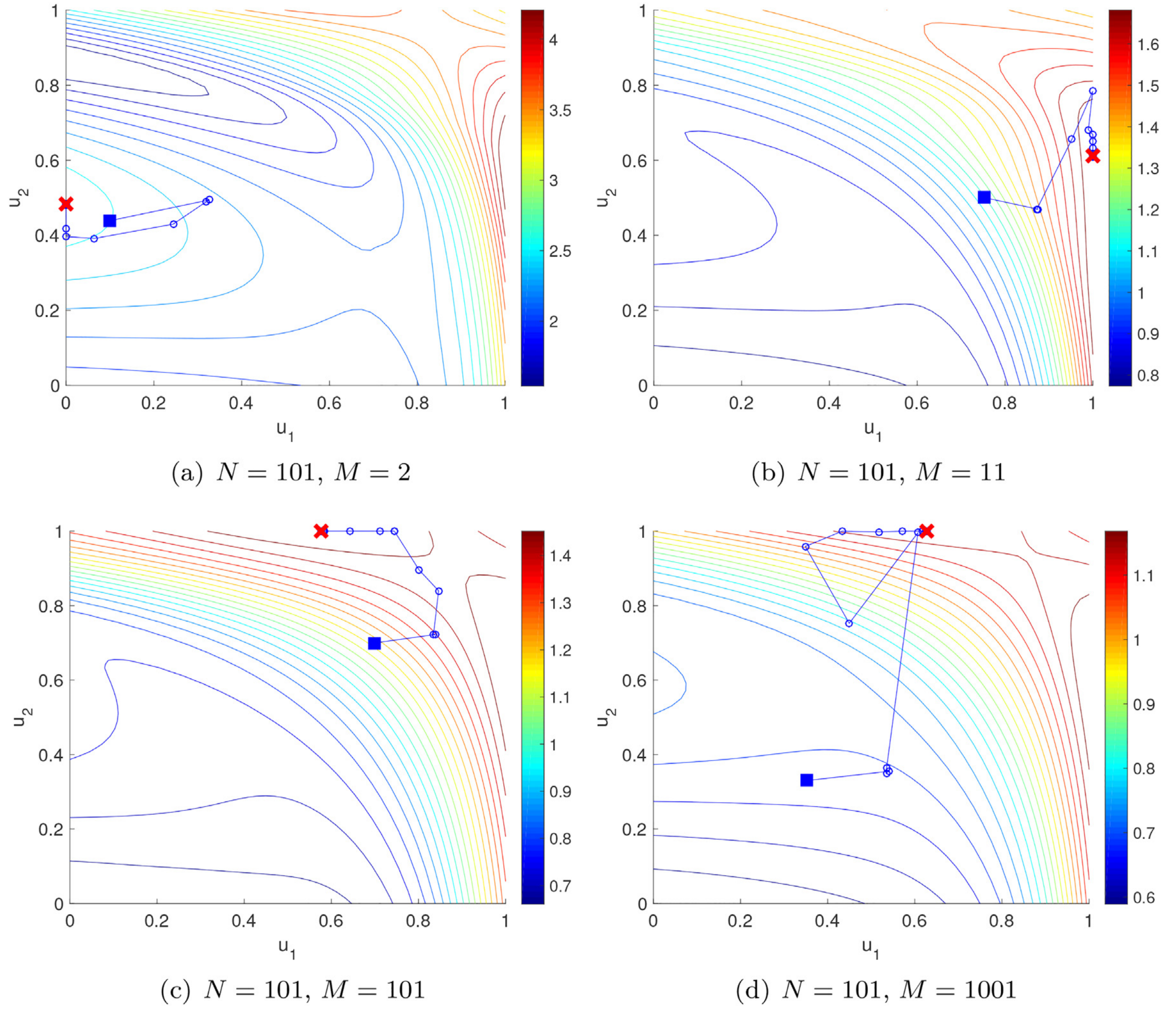
(c) $N = 11,\ M = 101$

(d) $N = 11,\ M = 1001$

Fig. 4. Realizations of the objective surface using SAA and the corresponding iterations of IPOPT, with $N = 11$ and four separate M values. The blue □ is the starting point and the red × is the final converged point. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
High-quality estimates of the expected utility (or information gain in this case) at the optimal designs resulting from 1000 independent runs of SAA. For each histogram, the horizontal axis represents values of $\hat{U}_{1001,1001}$ and the vertical axis represents frequency.

M N	2	11	101	1001
1				
11				
101				



(a) $N = 101,\ M = 2$

(b) $N = 101,\ M = 11$

(c) $N = 101,\ M = 101$

(d) $N = 101,\ M = 1001$

Fig. 5. Realizations of the objective surface using SAA and the corresponding iterations of IPOPT, with $N = 101$ and four separate M values. The blue □ is the starting point and the red × is the final converged point. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

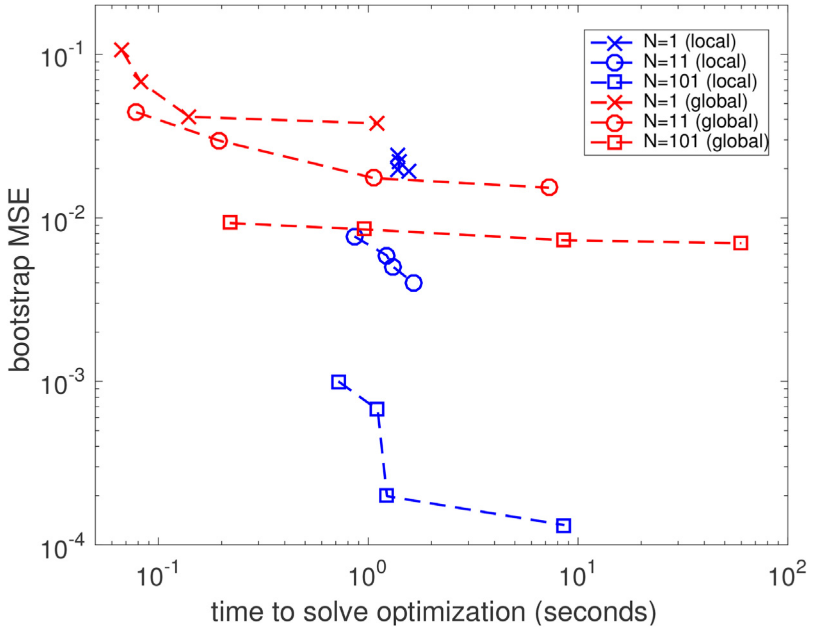


Fig. 6. Mean-square error (MSE) for the set of "bootstrap" optimization runs, defined by (46), versus the average run time for SAA under both the local and global PCE surrogate models and various choices of inner and outer sample sizes N and M.

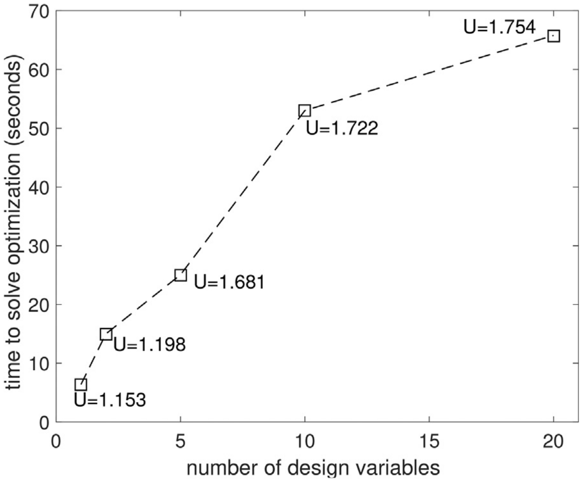


Fig. 7. Average run time of SAA with the local PCE surrogate versus the number of discretized design variables. The maximum expected utility achieved for each discretization level is also shown.

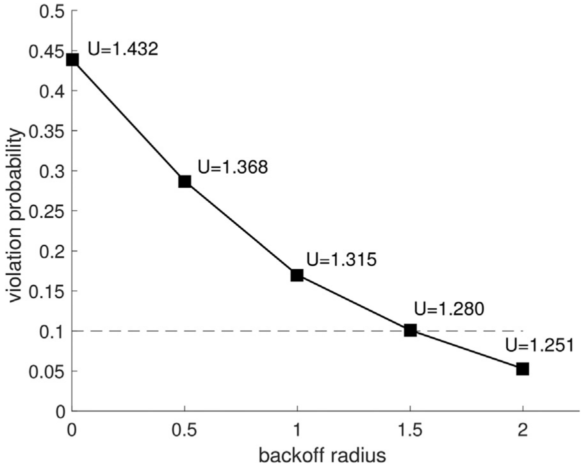


Fig. 8. The estimated state constraint violation probability under the optimal design profile versus the backoff radius. The maximum expected utility for each backoff radius is also shown.

that nearly maximizes the original dynamic problem (8), and these solutions can be found in around 25 seconds whereas the original problem is unsolvable.

5.6. Chance constraint approximation and tuning

An important consideration in this work is chance constraints of the form (2). Here, we consider box state constraints $(x_1(t), x_2(t)) \in [0, 3] \times [0, 1.5]$ and $\beta = 0.1$, meaning 10% of the state profiles are allowed to violate the box constraints. However, the only active state constraint for the Bayesian OED problem at hand is $x_2(t) \leq 1.5$, such that $P(x(t; d, \theta) \in \mathbb{X}) = P(x_2(t; d, \theta) \leq 1.5)$. For tractability purposes, we enforce the smooth moment-based approximation (12) instead of the original chance constraint. The required mean and variance terms can be expressed as a function of the design variables using the coefficients of the local PCE surrogate as shown in (26) and (27).

As discussed in Section 3, we cannot directly select a good value for the backoff radius parameter r as the distribution of x_2 is unknown at the optimal design. Therefore, we explore a simple simulation-based procedure to determine what value of r results in the largest possible objective. All that is required is to solve the OED problem with $\mathbb{E}\{x_2(t; d, \theta)\} + r\sqrt{\text{Var}\{x_2(t; d, \theta)\}} \leq 1.5$ added for a given r. Once the optimal design for this problem has been found, denoted by $d^\star(r)$, MC simulations are performed on the forward model to empirically estimate $P(x_2(t;d^\star(r), \theta))$. Note that the surrogate model can be used to further speed up the estimation of this violation probability, as established in [33]. If the violation is greater than β, then r should be increased and, if on the other hand, the observed violation is less than β, then r should be decreased. This procedure is repeated until satisfactory convergence is achieved. The basic methodology is presented in Fig. 8, which plots the estimated violation probability under 1000 MC runs versus the backoff parameter r, with other parameters set to $N = M = 101$ and $N_T = 5$. We see that r has a nonlinear effect on $P(x_2(t;d^\star(r), \theta) \leq 1.5)$. We also see that the constraint violation equals the desired level of 10% when the radius is $r = 1.5$. Due to the simple structure of the constraints combined with the cheap surrogate model, these average solution times are virtually identical with or without (12) included.

We emphasize the fact that the violation probability only had to be estimated at the optimal design, as opposed to being calculated at every iteration of the optimization and expressed with binary variables. High-quality estimates of the expected utility are also shown for each backoff radius in Figure 8 and, as expected, it decreases with increasing radius due to the reduced feasible region. The expected utility shows almost a linear decrease with increasing β for this problem, but in general this effect can be highly nonlinear. This type of Pareto analysis is useful to perform to probe the tradeoff between performance and "robustness" to parameter uncertainty. It is worth noting that this type of Pareto curve can be straightforwardly traced over $\beta \in [0, 1]$ by finding the r that yields $P(x_2(t;d^\star(r), \theta) \leq 1.5) = 1 - \beta$.

6. Conclusions and future work

This paper studies the stochastic optimization problem arising from a general nonlinear formulation for Bayesian optimal experiment design (OED), with a particular focus on dynamic systems subject to state chance constraints. The main objective function of interest is the expected information gain in the model parameters due to an experiment, which can be written in terms of the KL divergence from the posterior to the distribution. Since this expected information gain cannot be evaluated exactly, we must resort to a finite-sample Monte Carlo (MC) approximation to the objective function using the well-known sample average approximation (SAA) method. Although a similar SAA approximation can be applied to the chance constraints, these constraints require integer variables (non-smooth) and produce a feasible region that changes for every set of realizations used. Therefore, we propose a smooth moment-based approximation to the chance constraints that has a tunable backoff parameter, which can be determined through a limited number of simulations to ensure the original constraints are satisfied. Methods for computing the objective must also deal with the fact that the estimator of the expected information gain is not a simple MC sum, but involves a nested sum of MC estimates. It is therefore expensive to evaluate the objective and/or its gradients, as each sample in the estimator requires the dynamic forward model to be integrated over time. As suggested in previous work [15,25], we look to circumvent these challenges by approximating the forward model with polynomial chaos expansions (PCEs) and subsequently computing the expected information gain with PCEs instead.

The main contribution of this paper is to develop a new PCE-based surrogate model that is design-dependent, i.e., coefficients are updated locally at each design. In this way, the exponential growth with respect to the number of design variables that occurs in the global PCE method of [15,25] can be avoided. Another key feature of the proposed local PCE method is that the expansions are defined in terms of polynomials that are orthogonal with respect to arbitrary priors, meaning they most accurately approximate the forward model in high probability regions of the parameter space. Since the expansion coefficients are defined as weighted multi-dimensional integrals, the cost of estimating these coefficients is directly proportional to the number of nodes in a chosen discrete quadrature rule. Therefore, we apply an optimization-based procedure to numerically derive a rule that ensures high accuracy integration with a minimal number of nodes. We compare the performance of the proposed local PCE method to the global method of [15,25] on the problem of estimating parameters from noisy data in a dynamically evolving predator-prey system. Numerical experiments are performed over a matrix of inner- and outer-loop sample sizes to examine their impact on bias and variance of the objective function. Unsurprisingly, we see that the solution quality improves as the sample sizes increase, but observe that the outer sample size has a larger effect than inner sample size. We also note that multiple local solutions exist, though the global solution is found approximately 70% of the time. When comparing the local and global PCE methods for only two design variables, we observe that the proposed local method can provide a significant speedup and lower error, especially as the sample size increases. Another important observation is that the average solution time with the local PCE surrogate scales sublinearly with respect to the design profile discretization level. This is a result of the coefficients being updated at each design, which ensures the size of the expansion is independent of the design profile. This is in sharp contrast to the global PCE model, which grows exponentially in size as the discretization level increases. We also show that the moment-based constraints can be tuned to guarantee satisfaction of the original chance constraints, and how the surrogate model ensures these added computations are negligible compared to the main cost of estimating the expected information gain.

The developed approach is based on a nested MC estimator for the expected information gain. Some known issues with this estimator are that it requires a large number of samples due to the double-loop sample-average structure and the inner loop can suffer from arithmetic underflow for small sample sizes, diffuse priors, or concentrated posteriors. Future work should focus on developing methods that can avoid these issues. One such example is to replace samples from the prior with those drawn from an importance sampling distribution. A good candidate importance sampler can be derived from the Laplace approximation (LA) [60], which expands the posterior distribution in terms of a second-order Taylor series around the maximum a posteriori (MAP) estimate. Although this reduces the inner-loop sample size, it comes at the cost of constructing the LA, so it is not obvious this will reduce the overall computational burden. Note that LA has also been used to directly approximate certain expected utilities in large-scale OED problems [61]. It would also be interesting to explore methods, such as parallelized solvers [62] or distributed optimization algorithms [63], that are capable of exploiting the structure of the NLPs derived from stochastic dynamic optimization problems in order to achieve further reductions in the solution time.

Lastly, this paper focuses on batch (or open-loop) OED, where the experiment is fully designed before any data are actually collected. An important area of future research is sequential (or closed-loop) OED, where data from previous experiments can be used to guide the design of future experiments. The closed-loop OED problem can be rigorously formulated using dynamic programming [64,Chapter 3], but significant computational challenges must be overcome for this to be practically solvable, especially since the state must be represented in terms of the posterior distribution of the parameters. The proposed PCE-based surrogate model could potentially be used to address some of these challenges and, thus, help pave the way for OED to be solved in real-time in a fully Bayesian setting.

References

[1] E. Walter, L. Pronzato, Qualitative and quantitative experiment design for phenomenological models – a survey, Automatica 26 (2) (1990) 195–213.
[2] A.C. Atkinson, A.N. Donev, Optimum Experimental Designs, Oxford University Press, New York, 2007.
[3] G. Franceschini, S. Macchietto, Model-based design of experiments for parameter precision: state of the art, Chem. Eng. Sci. 63 (19) (2008) 4846–4872.
[4] S. Streif, F. Petzke, A. Mesbah, R. Findeisen, R.D. Braatz, Optimal experimental design for probabilistic model discrimination using polynomial chaos, in: Proceedings of the IFAC World Congress, Cape Town, 2014, pp. 4103–4109.
[5] M. Martin-Casas, A. Mesbah, Discrimination between competing model structures of biological systems in the presence of population heterogeneity, IEEE Life Sci. Lett. 2 (3) (2016) 23–26.
[6] K. Chaloner, I. Verdinelli, Bayesian experimental design: a review, Stat. Sci. (1995) 273–304.
[7] J.O. Berger, Statistical Decision Theory and Bayesian Analysis, Springer New York, New York, NY, 1985.
[8] G.E.P. Box, H.L. Lucas, Design of experiments in non-linear situations, Biometrika 46 (1959) 77–90.
[9] S. Körkel, E. Kostina, H.G. Bock, J.P. Schlöder, Numerical methods for optimal control problems in design of robust optimal experiments for nonlinear dynamic processes, Optim. Methods Softw. 19 (2004) 327–338.
[10] L. Pronzato, E. Walter, Robust experiment design via stochastic approximation, Math. Biosci. 75 (1985) 103–120.
[11] Y. Chu, J. Hahn, Integrating parameter selection with experimental design under uncertainty for nonlinear dynamic systems, AIChE J. 54 (9) (2008) 2310–2320.
[12] I. Bauer, H.G. Bock, S. Körkel, J.P. Schlöder, Numerical methods for optimum experimental design in dae systems, J. Comput. Appl. Math. 120 (2000) 1–25.
[13] A. Mesbah, S. Streif, A probabilistic approach to robust optimal experiment design with chance constraints, IFAC-PapersOnLine 48 (8) (2015) 100–105.
[14] M.C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, J. R. Stat. Soc. B: Stat. Methodol. 63 (2001) 425–464.
[15] X. Huan, Y. Marzouk, Simulation-based optimal Bayesian experimental design for nonlinear systems, J. Comput. Phys. 232 (2013) 288–317.
[16] D.V. Lindley, On a measure of the information provided by an experiment, Ann. Math. Stat. (1956) 986–1005.
[17] E.G. Ryan, C.C. Drovandi, J.M. McGree, A.N. Pettitt, A review of modern computational algorithms for Bayesian optimal design, Int. Stat. Rev. 84 (2016) 128–154.
[18] M.A. Clyde, P. Müller, G. Parmigiani, Exploring expected Utility Surfaces by Markov Chains, Technical Report, Duke University, 1996.
[19] P. Müller, B. Sansó, M. De Iorio, Optimal Bayesian design by inhomogeneous Markov chain simulation, J. Am. Stat. Assoc. 99 (2004) 788–798.
[20] K.J. Ryan, Estimating expected information gains for experimental designs with application to the random fatigue-limit model, J. Comput. Graph. Stat. 12 (2003) 585–603.
[21] J.A. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (1965) 308–313.
[22] J.C. Spall, An overview of the simultaneous perturbation method for efficient optimization, Johns Hopkins APL Technical Digest 19 (4) (1998) 482–492.
[23] H. Kushner, G.G. Yin, Stochastic Approximation and Recursive Algorithms and Applications, Vol. 35: Applications of Mathematics, Springer, 2003.
[24] A. Shapiro, Asymptotic analysis of stochastic programs, Ann. Oper. Res. 30 (1991) 169–186.
[25] X. Huan, Y. Marzouk, Gradient-based stochastic optimization methods in Bayesian experimental design, Int. J. Uncertain. Quantif. 4 (2014).
[26] D. Xiu, G.E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2002) 619–644.
[27] L.T. Biegler, An overview of simultaneous strategies for dynamic optimization, Chem. Eng. Process. Process Intensif. 46 (2007) 1043–1053.
[28] R.E. Caflisch, Monte Carlo andquasi-Monte Carlo methods, Acta Numer. 7 (1998) 1–49.
[29] K. Pagnoncelli, B.S. Ahmed, A. Shapiro, Sample average approximation method for chance constrained programming:theory and applications, J. Optim. Theory Appl. 142 (2009) 399–416.
[30] A. Nemirovski, A. Shapiro, Convex approximations of chance constrained programs, SIAM J. Optim. 17 (2006) 969–996.
[31] J.A. Paulson, A. Mesbah, An efficient method for stochastic optimal control with joint chance constraints for nonlinear systems, Int. J. Robust Nonlinear Control (2017) 1–21.
[32] A. Ben-Tal, L.E. Ghaoui, A. Nemirovski, Robust Optimization, Princeton University Press, 2009.
[33] J.A. Paulson, A. Mesbah, Nonlinear model predictive control with explicit backoffs for stochastic systems under arbitrary uncertainty, in: Proceedings of the 6th IFAC Conference on Nonlinear Model Predictive Control, Madison, WI, 2018, pp. 622–633.
[34] A. Geletu, A. Hoffmann, M. Kloppel, P. Li, An inner-outer approximation approach to chance constrained optimization, SIAM J. Optim. 27 (2017) 1834–1857.

[35] P. Carbonetto, M. Schmidt, N.D. Freitas, An interior-point stochastic approximation method and an L1-regularized delta rule, Advances in Neural Information Processing Systems (2009) 233–240.

[36] A.J. Kleywegt, A. Shapiro, T. Homem-de Mello, The sample average approximation method for stochastic discrete optimization, SIAM J. Optim. 12 (2002) 479–502.

[37] W. Gautschi, On generating orthogonal polynomials, SIAM J. Sci. Stat. Comput. 3 (3) (1982) 289–317.

[38] J.A. Paulson, E.A. Buehler, A. Mesbah, Arbitrary polynomial chaos for uncertainty propagation of correlated random variables in dynamic systems, IFAC-PapersOnLine 50 (2017) 3548–3553.

[39] J.A. Paulson, A. Mesbah, Arbitrary polynomial chaos for quantification of general probabilistic uncertainties: shaping closed-loop behavior of nonlinear systems, in: Proceedings of the 57th IEEE Conference on Decision and Control, Miami, 2018, in press.

[40] J. Feinberg, V.G. Eck, H.P. Langtangen, Multivariate polynomial chaos expansions with dependent variables, SIAM J. Sci. Comput. 40 (1) (2018) A199–A223.

[41] S. Oladyshkin, W. Nowak, Data-driver uncertainty quantification using the arbitrary polynomial chaos expansion, Reliab. Eng. Syst. Saf. 106 (2012) 179–190.

[42] O. Ernst, A. Mugler, H. Starkloff, E. Ullmann, On the convergence of generalized polynomial chaos expansions, ESAIM: Math. Model. Numer. Anal. 46 (2012) 317–339.

[43] C. Soize, R. Ghanem, Physical systems with random uncertainties: chaos representations with arbitrary probability measure, SIAM J. Sci. Comput. 26 (2004) 395–410.

[44] D. Xiu, Efficient collocational approach for parametric uncertainty analysis, Commun. Comput. Phys. 2 (2007) 293–309.

[45] K.-K.K. Kim, D.E. Shen, Z.K. Nagy, R.D. Braatz, Wiener's polynomial chaos for the analysis and control of nonlinear dynamical systems with probabilistic uncertainties [historical perspectives], IEEE Control Syst. 33 (2013) 58–67.

[46] R.G. Ghanem, P.D. Spanos, Stochastic finite element method: response statistics, in: Stochastic Finite Elements: A Spectral Approach, Springer, 1991, pp. 101–119.

[47] B.J. Debusschere, H.N. Najm, P.P. Pébay, M. Knio, O.R.G. Ghanem, O.P. Le Maitre, Numerical challenges in the use of polynomial chaos representations for stochastic processes, SIAM J. Sci. Comput. 26 (2004) 698–719.

[48] D. Xiu, Fast numerical methods for stochastic computations: a review, Commun. Comput. Phys. 5 (2009) 242–272.

[49] M. Sinsbeck, W. Nowak, An optimal sampling rule for nonintrusive polynomial chaos expansions of expensive models, Int. J. Uncertain. Quantif. 5 (2015) 275–295.

[50] T. Gerstner, M. Griebel, Dimension-adaptive tensor-product quadrature, Computing 71 (2003) 65–87.

[51] B. Sudret, Global sensitivity analysis using polynomial chaos expansions, Reliab. Eng. Syst. Saf. 93 (2008) 964–979.

[52] S. Hosder, R. Walters, M. Balch, Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables, 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (2007) 1939.

[53] D.L. Wei, Z.S. Cui, J. Chen, Uncertainty quantification using polynomial chaos expansion with points of monomial cubature rules, Comput. Struct. 86 (2008) 2102–2108.

[54] E.K. Ryu, S.P. Boyd, Extensions of Gauss quadrature via linear programming, Foundat. Comput. Math. 15 (2015) 953–971.

[55] M. Rosenblatt, Remarks on a multivariate transformation, Ann. Math. Stat. 23 (1952) 470–472.

[56] D. Telen, F. Logist, E. Van Derlinden, I. Tack, J. Van Impe, Optimal experiment design for dynamic bioprocesses: a multi-objective approach, Chem. Eng. Sci. 78 (2012) 82–97.

[57] J.A.E. Andersson, J. Gillis, G. Horn, J.B. Rawlings, M. Diehl, CasADi – a software framework for nonlinear optimization and optimal control, math. Program. Comput. (2018) 1–36.

[58] A. Wächter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Math. Program. Ser. A 18 (2005) 25–57.

[59] S.D. Cohen, A.C. Hindmarsh, P.F. Dubois, Cvode, a stiff/nonstiff ODE solver in C, Comput. Phys. 10 (1996) 138–143.

[60] Q. Long, M. Scavino, R. Tempone, S. Wang, Fast estimation of expected information gains for Bayesian experimental designs based on Laplace approximations, Comput. Methods Appl. Mech. Eng. 259 (2013) 24–39.

[61] A. Alexanderian, N. Petra, G. Stadler, O. Ghattas, A fast and scalable method for A-optimal design of experiments for infinite-dimensional Bayesian nonlinear inverse problems, SIAM J. Sci. Comput. 38 (2016) A243–A272.

[62] J. Kang, N. Chiang, C.D. Laird, V.M. Zavala, Nonlinear programming strategies on high-performance computers, in: Proceedings of the 54th IEEE Conference on Decision and Control, IEEE, Osaka, 2015, pp. 4612–4620.

[63] Y. Jiang, P. Nimmegeers, D. Telen, J. Van Impe, B. Houska, A distributed optimization algorithm for stochastic optimal control, IFAC-PapersOnLine 50 (1) (2017) 11263–11268.

[64] X. Huan, Numerical Approaches for Sequential Bayesian Optimal Experimental Design, Ph.D. Thesis, Massachusetts Institute of Technology, 2015.