Journal of Computational Physics 255 (2013) 521–539

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Improved statistical models for limited datasets in uncertainty quantification using stochastic collocation

Aravind Alwan, N.R. Aluru *

Department of Mechanical Science and Engineering, Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana–Champaign, 405 N. Mathews Avenue, Urbana, IL 61801, United States

ARTICLE INFO

Article history:
Received 18 March 2013
Received in revised form 11 July 2013
Accepted 14 August 2013
Available online 27 August 2013

Keywords:
Uncertainty quantification
Stochastic collocation
Density estimation
Moment matching
Reproducing kernel Hilbert space (RKHS)
Microelectromechanical systems (MEMS)

ABSTRACT

This paper presents a data-driven framework for performing uncertainty quantification (UQ) by choosing a stochastic model that accurately describes the sources of uncertainty in a system. This model is propagated through an appropriate response surface function that approximates the behavior of this system using stochastic collocation. Given a sample of data describing the uncertainty in the inputs, our goal is to estimate a probability density function (PDF) using the kernel moment matching (KMM) method so that this PDF can be used to accurately reproduce statistics like mean and variance of the response surface function. Instead of constraining the PDF to be optimal for a particular response function, we show that we can use the properties of stochastic collocation to make the estimated PDF optimal for a wide variety of response functions. We contrast this method with other traditional procedures that rely on the Maximum Likelihood approach, like kernel density estimation (KDE) and its adaptive modification (AKDE). We argue that this modified KMM method tries to preserve what is known from the given data and is the better approach when the available data is limited in quantity. We test the performance of these methods for both univariate and multivariate density estimation by sampling random datasets from known PDFs and then measuring the accuracy of the estimated PDFs, using the known PDF as a reference. Comparing the output mean and variance estimated with the empirical moments using the raw data sample as well as the actual moments using the known PDF, we show that the KMM method performs better than KDE and AKDE in predicting these moments with greater accuracy. This improvement in accuracy is also demonstrated for the case of UQ in electrostatic and electrothermomechanical microactuators. We show how our framework results in the accurate computation of statistics in micromechanical systems.

1. Introduction

Uncertainty quantification (UQ) has become a necessary step in the design of most modern engineering systems due to the need to create robust devices that can tolerate variations in the manufacturing process or in the operating environment. These variations or uncertainties can be represented by stochastic variables which perturb the deterministic behavior of the device about the nominal value for which it was designed. The UQ process consists of identifying the relevant uncertain parameters, assigning appropriate stochastic models to them and quantifying their effect on the final performance of the device. By leveraging fast and accurate deterministic solvers [1–5], it is possible to perform UQ on the computer so as to be able to predict the sensitivity of a given design to an applied set of uncertainties. In this paper, we restrict our focus

* Corresponding author.
E-mail address: aluru@illinois.edu (N.R. Aluru).
URL: http://www.illinois.edu/~aluru (N.R. Aluru).

0021-9991/$ – see front matter
http://dx.doi.org/10.1016/j.jcp.2013.08.024

to a broad category of devices that are collectively called microelectromechanical systems (MEMS). These devices have dimensions that are of the order of micrometers and are particularly sensitive to uncertainties that arise due to an inability to precisely control manufacturing tolerances. Performing UQ for these devices is further complicated by the fact that the characterization data, which describes the variation in the design parameters, is very often unavailable or sparse in quantity. Hence, our goal in this paper is to perform UQ in micromechanical devices by developing a framework that can produce reliable predictions given the limited amount of data describing the uncertainties.

To provide a context for the process of uncertainty quantification, we consider an example that illustrates the issues involved. Microelectromechanical devices are typically batch-fabricated on silicon wafers through a repeated process of etching or deposition after masking or exposing selective areas of the wafer. As a result of process variations during batch processing, there may be significant difference in material and geometric properties of devices fabricated on different parts of a wafer or even across different wafers. Uncertainty quantification is a valuable tool used by the designer to predict the effect of these variations on the performance of a nominal device by considering a numerical model of the device where some of the parameters are uncertain. UQ can be performed by either assuming stochastic models for random parameters or by fitting models to experimentally measured characterization data. For instance, a set of measurements of the elastic modulus of the material that forms the structural layer in a microactuator, can be used to estimate a stochastic model for elastic modulus, which can then be propagated through a numerical model of the device. The designer can then predict the effect of a particular stochastic parameter on some output quantity of interest, say the mean displacement of the actuator under an applied stimulus.

There are many different ways of propagating uncertainties through a numerical model of a device. Although traditionally, sampling-based methods were popular for design optimization in microsystems [6–8], they have been replaced by faster methods like generalized polynomial chaos [9–11] and more recently, by stochastic collocation [12,13] that do not rely on statistical sampling. Adaptive improvements to these methods [11,14–16] have further improved the efficiency by adaptively concentrating the computational effort in the areas where the error is large. With the help of these methods, we can carry out the step of propagating uncertainties to any arbitrary precision in an efficient manner.

With the advent of these efficient propagation methods, the error in predictions is no longer limited by the precision of the propagation methods, but rather, by the accuracy of the stochastic models that represent the input uncertainties. For many sources of randomness, the nature of uncertainty is not known in advance and in the absence of a physical justification, it is not possible to apply a standard distribution function to represent the uncertainty. Instead we resort to a data-driven approach, where we estimate a probabilistic model using experimentally measured characterization data. Although density estimation methods have been popular for a few decades with the advent of kernel density estimation (KDE) [17–20] and its adaptive modifications [21], these methods are mostly designed for asymptotic convergence when the size of the data set is large. On the other hand, for the purpose of UQ in MEMS, it is more appropriate to pick methods that try to preserve what is known from the limited data. We look at moment matching methods [22–24] that try to estimate PDFs that reproduce the statistics of either the raw data, or more interestingly, the statistics of the system response with respect to the data.

In this paper, we consider a method called kernel moment matching (KMM) [25], which can be used to estimate PDFs that accurately reproduce the moments of all functions that lie in a reproducing kernel Hilbert space (RKHS). Since we use stochastic collocation to construct approximations to the system response function, we leverage this property to identify a suitable RKHS for this process. We argue that the choice of this RKHS allows the resulting PDF to optimally reproduce the output statistics of any system modeled using stochastic collocation. This effectively decouples the density estimation process from the propagation step and allows for the estimation of PDFs without evaluating the actual system response function. Finally, we demonstrate through numerical examples that this process improves the accuracy of UQ in micromechanical systems like electrostatic and electrothermomechanical actuators. We propose that this method can be used in other engineering systems as well, where the amount of data that is available to characterize uncertainties is limited.

The rest of this paper is organized as follows: Section 2 gives a brief overview of uncertainty propagation, with a specific focus on stochastic collocation, in order to introduce some notation that will be used in the later section. The main contribution of this paper, which is the construction of stochastic models with limited data, is presented in Section 3, where we modify existing density estimation methods to suit this goal. Section 4 presents the numerical results, where we consider density estimation in univariate and multivariate cases in order to demonstrate the advantage of our framework over traditional approaches. This section also includes MEMS examples to which we applied synthetic uncertainties as well as models estimated from actual data. We finally present the conclusions in Section 5.

2. Propagation of uncertainties

Before we discuss density estimation, which is the primary focus of this paper, we introduce some concepts related to uncertainty propagation and the computation of statistics, since we will use these ideas later on to guide the density estimation process. The most important and computationally intensive step of uncertainty quantification is the process of propagating uncertainties through a numerical model of a device, which is nothing but a deterministic solver that can compute the output parameter of interest for a given set of device parameters. The UQ process starts with the identification of random parameters that act as sources of uncertainty in the model. Once we identify the uncertain parameters that are relevant, we can call the deterministic solver multiple times with different sets of input parameters in order to get an idea

of the relation between the input uncertainties and the output result. Since each deterministic run can take a significant amount of time, we try to develop a systematic way of doing the above procedure so that we can estimate the variation that would be expected in the actual device with the least possible computational effort.

Stochastic analysis of a system can be conceptually thought of as replacing a system of deterministic equations by an equivalent system of equations where the terms have stochastic variables. We apply a finite noise assumption that limits the number of sources of uncertainty to be a finite number n. Assuming that the range of variation in these n stochastic variables is continuous and bounded, we consider an n-dimensional random space, $\mathcal{D}^n$, which forms the stochastic space that describes the perturbations in the deterministic model. Each point, $\boldsymbol{\xi}$, in this stochastic space represents one random realization of the uncertainties. Since we assume the random variables to be continuous, there is a corresponding joint probability density function (PDF) that describes the probability density at any point in the random space. The output from the deterministic solver can be thought of as a function, $f(\boldsymbol{\xi})$ that maps points in $\mathcal{D}^n$ to $\mathbb{R}$. This function is called the system response function or response surface. Thus the goal of uncertainty quantification is to summarize the variation in $f(\boldsymbol{\xi})$, given the PDF that represents the uncertainty in $\boldsymbol{\xi}$.

Stochastic collocation (SC) [12,26] is a method of propagating uncertainties that tries to systematically choose a set of points in the random space so that the function evaluations at these points can be used to construct an interpolant which serves as a reasonably good proxy for the actual function itself. The interpolant is expressed as a weighted sum of basis functions at the nodes, where the weights are the values of the function at the corresponding node. The set of points is chosen according to a sparse-grid generation algorithm so that the interpolation error can be driven down by adding additional nodal points to refine the interpolant. For the chosen interpolation level, the response function is evaluated at the sparse gridpoints and these values are used to construct the interpolant. Since the form of the basis functions is known *a priori*, after the interpolant has been constructed one can easily compute the statistics of the interpolant for any given PDF without requiring additional function evaluations.

To construct an interpolant in the multi-dimensional random space, we pick gridpoints and basis functions by choosing combinations of univariate grids along each random dimension. We use the Smolyak algorithm [27] to generate a sparse grid by choosing only those combinations that satisfy a particular relationship. The interpolant is constructed in a hierarchical manner and can be expressed as,

$$A_{q,n}(f) = A_{q-1,n}(f) + \Delta A_{q,n}(f), \tag{1}$$

where $A_{q,n}(f)$ is the interpolant that approximates the system response function, q being an order parameter that specifies the level of the interpolant. We note that in the hierarchical form, the interpolant at a level q is constructed as the sum of the interpolant at the level $q-1$ and an incremental contribution, $\Delta A_{q,n}(f)$, at the level q. This recursive formula is continued all the way to $A_{-1,n}(f)$, which is defined to be zero. The incremental interpolant is defined as,

$$\Delta A_{q,n}(f) = \sum_{|\boldsymbol{k}|=n+q} \sum_{\boldsymbol{j}} l_{\boldsymbol{j}}^{\boldsymbol{k}} \cdot \underbrace{\left(f\left(\xi_{\boldsymbol{j}}^{\boldsymbol{k}}\right) - A_{q-1,n}(f)\left(\xi_{\boldsymbol{j}}^{\boldsymbol{k}}\right)\right)}_{\omega_{\boldsymbol{j}}^{\boldsymbol{k}}}, \tag{2}$$

where $l_{\boldsymbol{j}}^{\boldsymbol{k}} = l_{j_1}^{k_1} \otimes \cdots \otimes l_{j_n}^{k_n}$ denotes a multivariate basis function that is expressed as a product of univariate basis functions. Each basis function is weighted by the contribution to the interpolant at the corresponding gridpoint, $\xi_{\boldsymbol{j}}^{\boldsymbol{k}}$. In this paper, we use piecewise linear functions as the univariate basis functions used to construct the interpolant. Details concerning the form of these basis functions and the placement of gridpoints may be found in [15,16]. Fig. 1 shows the sparse grid generated by the Smolyak algorithm for a two-dimensional random domain for $q = 2$ along with the multivariate basis functions corresponding to three of the gridpoints.

According to Eq. (2), the incremental interpolant is written as a weighted sum of these basis functions, where the weights, $\omega_{\boldsymbol{j}}^{\boldsymbol{k}}$, are referred to as hierarchical surpluses. The hierarchical surplus at a gridpoint is the difference between the values of the function and the interpolant at the previous level, at that gridpoint. Thus, the basis functions corresponding to the newly added gridpoints are weighted by the incremental contribution to the interpolant at those points. Thus we see that the resulting interpolant is piecewise linear and can be constructed in a straight-forward way once the range of each random dimension is established. By increasing the order parameter, q, it is possible to refine the interpolant to achieve the desired level of precision.

Having constructed the interpolant using stochastic collocation, one can easily estimate statistical indicators like the mean and variance of the system response by computing the moments of $f(\boldsymbol{\xi})$ with respect to the joint PDF of the stochastic input parameters. Denoting this joint PDF by $\rho(\boldsymbol{\xi})$, we can express the mean and variance of the output as,

$$\text{Mean} = \mathbb{E}_{\boldsymbol{\xi}\sim\rho}\left[f(\boldsymbol{\xi})\right] = \int_{\mathcal{D}^n} f(\boldsymbol{\xi})\rho(\boldsymbol{\xi})\,d\boldsymbol{\xi}, \tag{3}$$

$$\text{Variance} = \mathbb{E}_{\boldsymbol{\xi}\sim\rho}\left[f^2(\boldsymbol{\xi})\right] - \left(\mathbb{E}_{\boldsymbol{\xi}\sim\rho}\left[f(\boldsymbol{\xi})\right]\right)^2 = \int_{\mathcal{D}^n} f^2(\boldsymbol{\xi})\rho(\boldsymbol{\xi})\,d\boldsymbol{\xi} - \left(\int_{\mathcal{D}^n} f(\boldsymbol{\xi})\rho(\boldsymbol{\xi})\,d\boldsymbol{\xi}\right)^2, \tag{4}$$

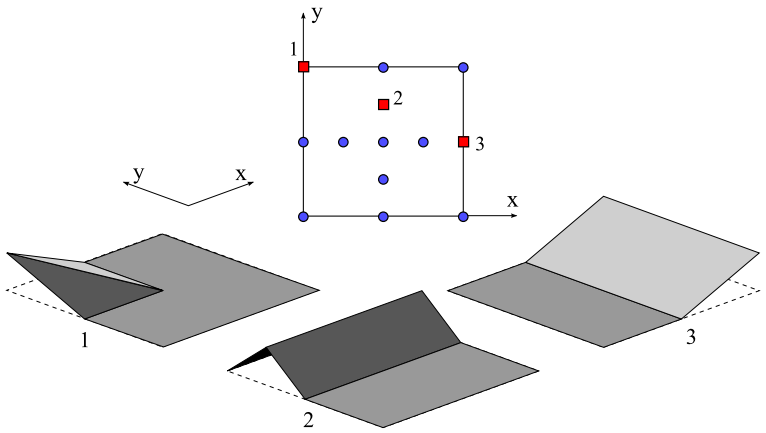


Fig. 1. Multivariate grid of order $q = 2$ for 2 random dimensions ($n = 2$). The basis functions corresponding to three different gridpoints (denoted by red squares) are also shown. Basis function 1 is a product of l_1^2 along x and l_2^2 along y. Similarly, case 2 is a product of l_1^1 and l_2^3, while case 3 is a product of l_2^2 and l_1^1.

where $\mathbb{E}_{\boldsymbol{\xi}\sim\rho}[g(\boldsymbol{\xi})]$ is the expectation of a function $g(\boldsymbol{\xi})$ with respect to the $\rho(\boldsymbol{\xi})$, which is nothing but the integral of the product of the function and the PDF. To evaluate these integrals, we use a multivariate numerical quadrature scheme over the entire random space. Normally, this process would call for evaluating the function at the quadrature points. However, we can approximate the value of these integrals by replacing the actual response function in Eq. (3) and Eq. (4), with its sparse-grid interpolant, $A_{q,n}(f)$. The advantage of using the interpolant is that having constructed it as a weighted sum of known basis functions, we can evaluate its value at the quadrature points with much lower computational effort. Moreover, it has been shown that the error in the statistical moments that are computed using the sparse interpolant, is bounded and monotonically decreases as we increase the refinement level of the sparse grid [28]. An important point to note in the computation of variance is that the first term involves the expectation of f^2. This is done by constructing an interpolant for f^2 using the same multivariate grid and basis functions as for f. Since we already have the values of f at the gridpoints, the construction of this additional interpolant for f^2 is a fairly trivial task.

3. Generation of stochastic models

In the previous section, we showed how stochastic collocation can be used to generate an interpolant that approximates the system response, which can be used in conjunction with the joint PDF of the inputs to estimate the statistics of the output parameter of interest. Since the error in the interpolant can be decreased to an arbitrarily low value by sufficiently refining the interpolant [29], the accuracy of the output statistics is limited only by the accuracy of the joint PDF that is used to model the input uncertainty. The output statistics are important because they are the only quantities that can be directly validated against experimental measurements. In this section, we discuss various density estimation methods and try to compare the efficacy of these methods in generating the PDF that best replicates the output statistics.

For most micromechanical devices, it is a relatively simple task to identify input parameters that are uncertain and hence, are associated with appropriate stochastic variables. However, we may not always know the nature of the uncertainty that is associated with these inputs. For any parameter, we may only have a set of measurements that are obtained through the course of several characterization experiments, which are performed as a part of model calibration. This set of measurements can be thought of as a single random set of values that are sampled from the actual, though unknown, distribution that describes the uncertainty in that parameter. Our goal is to use this set of measurements to estimate a data-driven PDF for the stochastic inputs so that our stochastic model is based on the actual uncertainty in the data.

Since characterization experiments are difficult and time-consuming, the given set of measurements is usually quite small, hardly containing about 50 samples. This poses a challenge during estimation, since we have to contend with the statistical noise in the samples due to insufficient size of the data set. An additional wrinkle to deal with is the fact that in some cases, we may not have correlated sets of measurements for different random parameters, that are obtained from the same set of devices. In these cases, we shall assume that the samples measured across different sets of devices are independent. We also impose an additional constraint that the estimated density should have a non-parametric form. Non-parametric estimators try not to make any assumptions about the nature of the underlying distribution, which is in contrast with parametric estimation, where we assume that the uncertainty is described by some standard distribution function. This is important in the case of MEMS, since we do not have a sizable sample or a physical justification to draw inferences about the actual distribution.

3.1. Kernel density estimation

One of the most popular methods for non-parametric density estimation is kernel density estimation (KDE) [20]. Consider a data set, $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}$, where each point is randomly chosen from $\mathcal{D}^n$ and is one specific instance of the n random input parameters. We use KDE to construct an estimate $\hat{\rho}(\boldsymbol{x})$ to the unknown PDF $\rho_0(\boldsymbol{x})$. KDE applies the concept of kernel smoothing, where symmetric functions called kernels are centered at the data points and the PDF at any point is expressed as the sum of contributions from these kernels. The bandwidth of these kernels controls the amount of smoothing that is applied to the data and this parameter is tuned so that the resulting PDF is optimal with respect to some risk function. This optimization procedure is done in such a way that the estimated density function converges to the actual PDF in an asymptotic sense as the size of data set increases. The estimated PDF is expressed in terms of the data set as,

$$\hat{\rho}(\boldsymbol{x}) = \frac{1}{M}\sum_{i=1}^{M} K_{\boldsymbol{\Sigma}}(\boldsymbol{x} - \boldsymbol{x}_i), \tag{5}$$

where $K_{\boldsymbol{\Sigma}}(\boldsymbol{x})$ is the kernel function. The kernel function can be of many different forms, but one that is popularly used is the Gaussian kernel. The Gaussian kernel is given by,

$$K_{\boldsymbol{\Sigma}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x}\right), \tag{6}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix associated with the kernel bandwidths and $|\boldsymbol{\Sigma}|$ is the determinant of this matrix. We assume that the covariance matrix is diagonal and that it is identical for all the kernels. Expressing $\boldsymbol{\Sigma}$ as $\mathrm{diag}([h_1, h_2, \ldots, h_n])$, we see that $\hat{\rho}(\boldsymbol{x})$ depends on n unknown parameters. We use an optimization procedure to determine the value of $\boldsymbol{\Sigma}$ that optimizes some risk function. A common choice for the objective function is the maximum likelihood cross-validation (MLCV) score [30,31]. This uses likelihood as the objective function and uses leave-one-out cross-validation to tweak the objective function in order to avoid the trivial solution where the bandwidths are all zero. The MLCV score can be written as,

$$MLCV(\boldsymbol{\Sigma}) = \frac{1}{M}\sum_{i=1}^{M} \log \hat{\rho}_{-i}(\boldsymbol{x}_i), \tag{7}$$

where $\hat{\rho}_{-i}(x)$ is the leave-one density that uses the same value of $\boldsymbol{\Sigma}$ as used in $\hat{\rho}(x)$, but omits the i-th data point from the estimate. The value of $\boldsymbol{\Sigma}$ that maximizes this score is chosen as the optimal bandwidth, $\boldsymbol{\Sigma}^*$, which is used in Eq. (5) to get the final estimated density.

3.2. Adaptive kernel density estimation

Although KDE generates a data-driven non-parametric estimate of the PDF, it is limited by the fact that the same bandwidth is used for all the kernels. For sparse data, especially where there are a few random outliers, KDE tends to over-smooth the density by choosing a large bandwidth for the kernels [32]. This over-smoothing is mainly caused by the fact that KDE does not have the flexibility to spatially adapt the PDF depending on the sparsity in the data. When estimating PDFs for sparse data sets, it is often desirable to concentrate the mass of the PDF in regions where there is a greater concentration of data points and to reduce the mass in regions where the data points are sparse. This motivates an adaptive modification to the KDE procedure mentioned above. The adaptive kernel density estimation (AKDE) method proposed by Abramson [21], expresses the estimated density as,

$$\hat{\rho}(\boldsymbol{x}) = \frac{1}{M}\sum_{i=1}^{M} \frac{1}{\lambda_i} K_{\boldsymbol{\Sigma}'}(\boldsymbol{x} - \boldsymbol{x}_i), \tag{8}$$

where $\boldsymbol{\Sigma}' = \lambda_i \boldsymbol{\Sigma}$ is the modified bandwidth covariance matrix and λ_i is a scaling factor that spatially modifies the effective bandwidth from one data point to the next. λ_i is chosen to be equal to $\sqrt{G/\hat{\rho}_{KDE}(\boldsymbol{x}_i)}$, where G is the geometric mean of $\hat{\rho}_{KDE}(\boldsymbol{x}_i)$ for $i = 1, 2, \ldots, M$. By modifying the effective bandwidth in this way, we see that the bandwidth is increased for data points for which $\hat{\rho}_{KDE}(\boldsymbol{x}_i)$ is small. This effectively reduces the contribution from those data points, by making the spread diffuse. On the other hand, the bandwidth is decreased at points where the KDE estimate is high, causing the final PDF to become sharper at those points. This process of adaptively modifying the bandwidth results in a PDF estimate that is higher in regions of higher concentration of data points and lower elsewhere. This reduces the excessive smoothing caused by KDE to a certain extent. The density estimation is performed in a two-step procedure, where we first estimate the PDF using KDE and use the values of this estimate at the data points to evaluate the local bandwidth factors. These factors are then used to determine the PDF using Eq. (8).

3.3. Matching moments of the PDF

So far we have looked at density estimation methods based on maximizing the likelihood of the estimated density. The idea of maximizing the likelihood is motivated by the fact that the density estimate will converge to actual PDF asymptotically as M increases. However, when the data set is small there is significant noise in the data due to insufficient sampling. In this case, maximizing the likelihood may not be the obvious choice for the objective function [25]. In the UQ framework, the estimated PDFs are propagated through the system response function and ultimately used to compute the statistics of the output response. For this task, it may be better to estimate the PDFs so that they reproduce the output statistics with greater accuracy.

Before proceeding to the actual estimation procedure, we introduce some terminology to better define the goal of reproducing output statistics. Given the actual PDF describing the input uncertainty, we can compute the moments of the system response function to get the actual mean and variance of the output, as described in Eqs. (3) and (4). However, in practice, we do not know the actual PDF and we only have a sample derived from this PDF. This sample can be used to compute the empirical or sample mean, which is the average system response evaluated at each of the points in the data set. For large data sets, the value of empirical mean will be very close to the actual one if the data set represents a reasonably good sample of the actual uncertainty. However, that is not the case for small data sets where there can be a significant difference between the two. In addition to these two quantities, we can also estimate a PDF from the given data and compute the mean of the system response with respect to this PDF. We refer to this as the estimated mean. Depending on the estimation method used, the value of the estimated mean may be very different from the actual and empirical mean.

Our primary goal for PDF estimation is to be able to represent the uncertainty in the system response as accurately as possible. We argue that given the limited size of the data, a PDF that reproduces the statistics of the system response with greater accuracy, is more useful when performing UQ since it can be easily validated against experimental measurements. Ideally we would prefer an estimation method that optimizes the PDF so that the estimated mean is known to be close to the actual mean of the response function. However it is not possible to test this with real data, since we do not know the actual PDF from which the data has been sampled. Instead we can compare different estimation procedures by providing them with artificial data samples derived from known synthetic PDFs. If the match between the estimated and actual mean can be shown for these known PDFs, one might hope that the procedure would work well for the case where the actual PDF is unknown. We can also test the accuracy of the PDF by comparing the estimated mean with the empirical mean. Since the empirical mean can be readily computed from just the data sample, we can perform this comparison even for real data sets. Finally, in addition to the mean, similar arguments can be applied to higher order moments of the PDF.

An important point to note in this discussion about matching the moments is that we are referring to the moments of a specific system response function. However, we would ideally like to estimate PDFs that are not specific to one particular function, but rather, work well for all or at least a large class of functions. This idea of generating a transferable stochastic model is important to ensure that the density estimation step need not be repeated for every new response function. Instead of optimizing the PDF for a specific function, if we can optimize it for some common property that can be identified for a large class of functions, then the resulting PDF will be optimal for all functions in that family. We shall use this idea to guide the process of generating a transferable model.

3.3.1. Kernel moment matching

There has been a lot of research work in generating PDF estimates based on this concept of moment matching [22–24]. A recent paper by Song et al. [25] generalizes this idea to non-parametric density estimators. Their method, called kernel moment matching (KMM), tries to optimize the PDF such that it reproduces the moments of all functions which have the common property that they belong to some reproducing kernel Hilbert space (RKHS). They argue that if we have some knowledge of the function class for which we will be computing moments, then this can be used in the optimization procedure to ensure that the resulting PDF will be optimal for all functions in that space.

The method discussed in [25] is briefly presented here; additional details may be found in the original paper. We first assume that our system response function belongs to a known reproducing kernel Hilbert space, $\mathcal{H}$. We will examine the significance of this assumption in greater detail at the end of this section, but for now we will take it for granted. This space has a corresponding reproducing kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ such that for any function $g \in \mathcal{H}$, we have $g(\boldsymbol{x}) = \langle g, k(\boldsymbol{x}, \cdot)\rangle_{\mathcal{H}}$, where $k(\boldsymbol{x}, \cdot)$ is called the feature map of the RKHS. The feature map associates a point, $\boldsymbol{x}$ in the domain of the RKHS, with a function in the RKHS. Varying $\boldsymbol{x}$ is equivalent to horizontally translating the kernel function to different points in the domain.

For any PDF, $\rho(\boldsymbol{x})$, that is derived from the data set, X, we can define two functions: the kernel mean map of the PDF, $\mu[\rho] = \mathbb{E}_{\boldsymbol{x}\sim\rho}[k(\boldsymbol{x}, \cdot)]$ and the empirical mean map, $\mu[X] = \frac{1}{M}\sum_{i=1}^{M} k(\boldsymbol{x_i}, \cdot)$. These quantities are linear transformations of the kernel map and hence they map the PDF and the given data set respectively, to functions that are elements of the RKHS. Similarly we can map the actual PDF, $\rho_0(\boldsymbol{x})$, to the RKHS by defining $\mu[\rho_0] = \mathbb{E}_{\boldsymbol{x}\sim\rho_0}[k(\boldsymbol{x}, \cdot)]$. These quantities can be viewed as the expected shape of the function that is obtained by translating the feature map according to a given PDF. In other words, the kernel mean map associates an element of the RKHS, i.e. a function in the Hilbert space, with a given probability density function (PDF). This provides a way of embedding PDFs in a Hilbert space, so that different PDFs can be compared with respect to that space.

Song et al. [25] use a result proved in [33] to argue that for a given RKHS, $\mu[\rho_0]$ can be approximated well by $\mu[X]$ and so, the estimated density, $\hat{\rho}$, should be chosen such that $\mu[\hat{\rho}]$ matches $\mu[X]$, since this automatically ensures that $\mu[\hat{\rho}]$ will

be close to $\mu[\rho_0]$. They show that choosing $\hat{\rho}$ in this manner, puts an upper bound on the difference between $\mathbb{E}_{\boldsymbol{x}\sim\hat{\rho}}[f(\boldsymbol{x})]$ and $\mathbb{E}_{\boldsymbol{x}\sim\rho_0}[f(\boldsymbol{x})]$ for functions that belong to the RKHS. In other words, they show that if the goal of PDF estimation is to match the estimated mean with the actual mean for functions that belong to the RKHS, then it is sufficient to minimize the distance between the kernel mean map of estimated PDF and the empirical mean map. This can be posed as an optimization problem, where we choose $\hat{\rho}$ that optimizes the following objective function:

$$\min_{\hat{\rho}} \left\| \mu[X] - \mu[\hat{\rho}] \right\|_{\mathcal{H}}^2, \tag{9}$$

where $\|\cdot\|_{\mathcal{H}}$ is the norm operator that is defined in terms of the inner product as, $\|g(\boldsymbol{x})\|_{\mathcal{H}}^2 = \langle g(\boldsymbol{x}), g(\boldsymbol{x})\rangle_{\mathcal{H}}$, for any function, $g(\boldsymbol{x})$ in the RKHS.

To solve this optimization problem, we choose $\hat{\rho}$ to have a general form similar to that used in KDE. We assume that $\hat{\rho}$ is the weighted sum of contributions from kernels centered at each of the data points and can be expressed as,

$$\hat{\rho}(\boldsymbol{x}) = \sum_{i=1}^{M} \alpha_i K_{\boldsymbol{\Sigma}}(\boldsymbol{x} - \boldsymbol{x}_i); \qquad \sum_{i=1}^{M} \alpha_i = 1,\ \alpha_i > 0, \tag{10}$$

where the weights, α_i, are tunable parameters that can be varied to increase or decrease the dependence of the estimated density on each data point. Unlike KDE, this weighted form of the estimator is not affected by the issues relating to over-smoothing, since it has the flexibility to decrease the contribution from outliers by reducing the corresponding weight. By plugging Eq. (10) into Eq. (9), we get,

$$\min_{\boldsymbol{\alpha}} \left\| \mu[X] - \mu[\hat{\rho}] \right\|_{\mathcal{H}}^2; \qquad \sum_{i=1}^{M} \alpha_i = 1,\ \alpha_i > 0. \tag{11}$$

By substituting the expressions for $\mu[X]$ and $\mu[\hat{\rho}]$ into Eq. (11) and using the reproducing property of $k(\boldsymbol{x}, \boldsymbol{x}')$, we get a form for the objective function that can be solved using convex optimization methods. Appendix A details the steps involved in casting the above optimization problem in terms of a quadratic program. The final expression [25] that we get is given by,

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T(\boldsymbol{Q} + \gamma \boldsymbol{I})\boldsymbol{\alpha} - \boldsymbol{l}^T\boldsymbol{\alpha}; \qquad \sum_{i=1}^{M} \alpha_i = 1,\ \alpha_i > 0, \tag{12}$$

where $\boldsymbol{Q}$ is an $M \times M$ matrix whose elements are given by,

$$Q_{ij} = \langle \mu[\rho_i], \mu[\rho_j] \rangle_{\mathcal{H}} = \mathbb{E}_{\boldsymbol{x}\sim\rho_i, \boldsymbol{x}'\sim\rho_j}\left[k(\boldsymbol{x}, \boldsymbol{x}')\right], \tag{13}$$

and $\boldsymbol{l}$ is an $M \times 1$ vector whose elements are given by,

$$l_j = \langle \mu[X], \mu[\rho_j] \rangle_{\mathcal{H}} = \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_{\boldsymbol{x}'\sim\rho_j}\left[k(\boldsymbol{x}_i, \boldsymbol{x}')\right]. \tag{14}$$

In the above equations, ρ_i is defined as $K_{\boldsymbol{\Sigma}}(\boldsymbol{x} - \boldsymbol{x}_i)$, $\boldsymbol{I}$ is the identity matrix and γ is a constant used to introduce regularization in the quadratic program. In this paper, we use a value of 10^{-10} for γ. By solving Eq. (12), we can obtain the values of the weights, α_i, that can be used in Eq. (10) to get the estimated PDF. We have adapted this procedure to be used in conjunction with KDE in a nested fashion, wherein we choose the covariance matrix, $\boldsymbol{\Sigma}$, that maximizes the MLCV score, while ensuring that the weights are optimized using the KMM procedure for each iterative guess of $\boldsymbol{\Sigma}$. The optimization routine is written in the Python programming language, where we use the CVXOPT package [34] for solving the quadratic program, while the maximization of MLCV score is done using the BFGS algorithm [35] that is implemented in the optimization package included in Python's scientific computing library, SciPy [36]. The whole process is summarized in Algorithm 1.

The advantage of choosing the weights according to Eq. (12) is that it directly ensures that the estimated mean computed using the resulting PDF is close to empirical mean for all functions in the RKHS. An additional advantage of this procedure, as shown in [25], is that choosing the density in this manner, puts an upper bound on the difference between the estimated mean and the actual mean which is calculated with respect to the original PDF, ρ_0, from which the data is sampled. Similar to the previous result, this result holds for functions that lie in the RKHS. This gives us confidence that the estimated density, $\hat{\rho}$ is optimal not only for the given data set but is also the best estimate of the actual PDF as far as computing the mean of functions in the RKHS is concerned. In other words, the KMM method tries to reproduce features of the actual PDF that are most responsible for estimating the statistics of the system response function, assuming that we know that this function lies in a known RKHS. This result is achieved without requiring the actual system response function but instead, using the property that the function, for which we are computing the expectation, lies in a specific RKHS. This can be readily seen from the fact that the objective function mentioned in Eq. (12) is defined in terms of quantities that only depend on

Algorithm 1: Algorithm for estimating the PDF using the KMM method.

Data: Set of points, X, that represent data sampled from the unknown PDF, ρ_0.
Result: Optimal $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$ that can be used in Eq. (10) to get the estimated PDF, $\hat{\rho}$.

begin
  Initialize $\alpha_j = 1/M, \ \forall j = 1, 2, \ldots, M$.
  Initialize $\boldsymbol{h} \longleftarrow (h_1, h_2, \ldots, h_n)$, where $h_i = 1$.
  Define $\boldsymbol{\Sigma}(\boldsymbol{h}) \longleftarrow diag(\boldsymbol{h})$ as the current value of $\boldsymbol{\Sigma}$.
  Define $F \longleftarrow -MLCV(\boldsymbol{\Sigma}(\boldsymbol{h}))$ as the current value of the objective function.
  Minimize F using the BFGS algorithm as follows:
  repeat
    Obtain a new guess, $\boldsymbol{h}'$, by performing one step of the BFGS method.
    Compute $\boldsymbol{Q}$ and $\boldsymbol{l}$ using Eq. (13) and Eq. (14) respectively.
    Solve the quadratic program given by Eq. (12) to get the new value of $\boldsymbol{\alpha}$.
    Compute $F' \longleftarrow -MLCV(\boldsymbol{\Sigma}(\boldsymbol{h}'))$.
    Update $\boldsymbol{h} \longleftarrow \boldsymbol{h}'$.
  until $|(F - F')/max(F, F')| < tolerance$.
  Return $\boldsymbol{\Sigma}(\boldsymbol{h})$ and $\boldsymbol{\alpha}$ as the optimal parameters for the estimated PDF.
end

$k(\boldsymbol{x}, \boldsymbol{x}')$ and not on the actual system response function. Hence the entire estimation procedure can be performed in an offline manner without requiring computationally expensive evaluations of the system response function.

The original KMM method proposed in [25] allows the choice of the RKHS to be dependent on the properties of the system response function. However, in the context of MEMS, the response surfaces may vary widely from one situation to the next, depending on the physical model used to simulate the device as well as on the particular output quantity that we are interested in. In general, the typical response surfaces that we encounter may not even have any common property, much less lie in a known RKHS. To counter this problem, we review our UQ procedure and note that irrespective of the nature of the system response function, we always use stochastic collocation to construct an interpolant that approximates the actual function. It is this interpolant that is used when estimating output statistics. Therefore, even though the function may have any general form, stochastic collocation projects it into a subspace that is spanned by piecewise linear basis functions. We readily see that for the univariate case these piecewise linear basis functions belong to the Sobolev space $W_2^1(\mathbb{R})$, which consists of all functions whose derivatives lie in $L_2(\mathbb{R})$. This space is known to be an RKHS and its kernel is the Ornstein–Uhlenbeck kernel [37], given by,

$$k(x, y) = \frac{1}{2} e^{-|x-y|}, \tag{15}$$

which can be extended to the multivariate case as follows,

$$k(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2^n} \prod_{i=1}^{n} e^{-|x^i - y^i|}, \tag{16}$$

where x^i and y^i are the i-th components of $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively.

Hence, we propose a modification to the original KMM method, where we use the above kernel in Eq. (12) to estimate the density that is optimal for all piecewise linear functions that belong to the Sobolev RKHS. This means that this PDF is optimal for any function that is approximated by stochastic collocation using piecewise linear basis functions. This greatly extends the scope of the KMM method by incorporating it into a general framework for UQ, since we do not need to know the exact function class as we only deal with the approximate interpolant generated by stochastic collocation. It is important to note here that the PDF is not being calibrated for a specific response function, but rather, it is optimized for any response function that can be approximated by the stochastic collocation procedure mentioned above.

The PDF that is estimated using this method can also be used to compute the variance according to Eq. (4). We readily see that in addition to evaluating $\mathbb{E}_{\boldsymbol{\xi} \sim \rho}[f(\boldsymbol{\xi})]$, we also need to compute the expectation of f^2, which involves constructing a stochastic collocation interpolant for f^2 as well. Since this interpolant is also built using the same piecewise linear basis, it lies within the RKHS that we considered for f. This means that the PDF that is optimized to reproduce the mean of the system response will also be optimal for the variance, since it is optimal for both the terms in Eq. (4). The same argument can be extended to any higher order moment, since the expectations of higher powers of f can be computed using the same procedure. This is an additional advantage over the regular KMM method since the PDFs that are optimized for the stochastic collocation interpolant, can be used to accurately compute not only the mean but also higher order moments.

3.3.2. Test procedure

We have stated that the goal of density estimation in this paper is to reduce the prediction error measured with respect to empirical mean and variance, since the empirical quantities are good proxies for the actual ones, when the latter is not known. In order to compare the performance of the three density estimation methods discussed above, we employ a test procedure where we sample a known PDF to obtain data, which is then used to estimate the PDF using each of the three

methods. The PDF may be a univariate PDF that describes the variation in a single random variable or may be the joint density for multiple random variables. The generated PDFs can then be used to compute the estimated moments for specific system response functions. This allows us to compare the estimated moments with respect to the empirical value that is obtained using the data, as well as the actual moment that can be calculated using the known parent PDF.

The first step of this procedure is data sampling. We consider a known PDF and generate multiple replicates of data, where each replicate is a data set that comprises samples that are drawn independently from the same PDF. Each sample is a point in the bounded random space in which we perform UQ. The samples are drawn using the Metropolis–Hastings algorithm, where we generate a Markov chain of samples drawn from the PDF such that the distribution of the samples matches that of the parent PDF. The replicates are all of a known size, M, ranging from 5 to 160. The number of replicates is chosen to be large enough to allow for sufficient averaging when computing the statistics. For each replicate, we estimate the PDF using KDE, AKDE and KMM. The density estimation procedure has already been described in the previous sections. Thus, we obtain one PDF corresponding to each of the three methods, for each replicate.

We also choose functions for which the moments will be calculated. These functions may be randomly generated from a known RKHS or may be specifically chosen. For the univariate case, we specifically choose 100 random functions that belong to the RKHS described in Section 3.3.1, with the reproducing kernel function given by Eq. (15). Given that each replicate contains M samples denoted by $\{x_1, x_2, \ldots, x_M\}$, we generate the functions as $f = \sum_{i=1}^{M_0} \omega_i k(x_i, \cdot)$, where M_0 is a random number between 1 and M, $k(x, y)$ is the reproducing kernel and ω_i is a random value between -1 and 1. This method of generating random functions ensures that the functions will lie in the required RKHS. Consequently, we expect that the KMM procedure will be able to estimate PDFs that will reproduce the mean with greater accuracy.

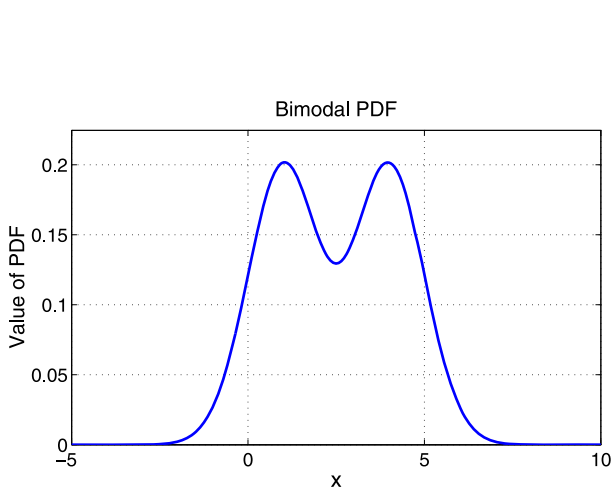
For each function, we compute the estimated values of the mean with respect to each of the estimated PDFs. We also compute the empirical means for each random function with respect to all the replicates. Finally, we compute the actual mean of each of the functions with respect to the known PDF. We use these values to compute the normalized error in the estimated expectations, $\epsilon = \|(\mathbb{E}_{x\sim\hat{\rho}}[f(x)] - \mu_0)/\mu_0\|$, where $\mathbb{E}_{x\sim\hat{\rho}}[f(x)]$ is the estimated mean, while μ_0 could be the empirical mean or the actual mean. Since we are generating the functions randomly, the error in the expectations is normalized so that the values are readily comparable. We can then calculate the median normalized error in the expectation (MNEE) for each of the three estimation methods. The MNEE can be calculated with respect to the empirical mean as well as the actual mean, by using the corresponding value for μ_0 in the expression mentioned above. In general, a lower value of the error should indicate that the method is able to generate PDFs that reproduce the statistics more accurately for the given functions.

The MNEE value that is computed in this manner, is an average quantity and only reflects the general trend in the error values. A more rigorous comparison of the performance can be obtained by performing a paired sign test [38]. The paired sign test is meant to test the hypothesis that the difference between the medians is significant. To do this, we first compile the list of normalized error values for the PDFs that are obtained using each of the three estimation methods. Each list is ordered by the replicate index as well as the random function for which the expectation error is computed. We count the number of cases where error values obtained using KDE are lower than KMM. Assuming that this number should follow a binomial distribution with a probability of 0.5, we can compute the p-value, which is the probability that the error obtained using KDE is lower than KMM. If this value is 0.5, then there is no significant difference between the performance of the two methods. However, if the p-value is lower than some small threshold, then we can conclude that KMM performs better than KDE in a majority of the cases. In this paper, we use a significance level of 0.01 to perform the sign test. This testing procedure is used to compare KDE, AKDE and KMM with each other and we conclude that a particular method performs the best only if the p-value with respect to both the other two methods is lower than the significance level. When there is no one method that shows a significant benefit, we conclude that all three methods perform equally well.

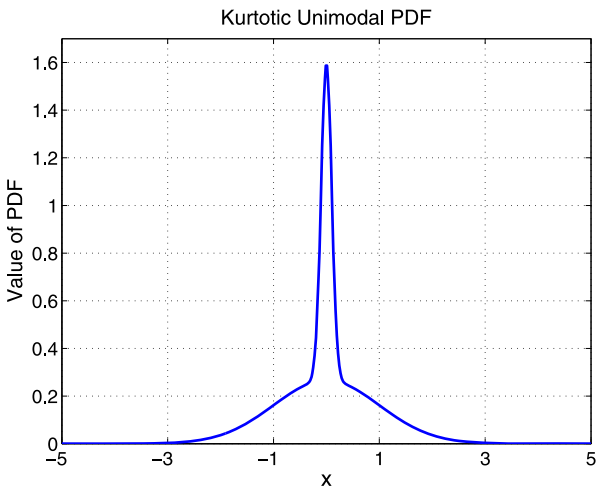
The above testing procedure can be extended to the case of multivariate PDFs as well. The only difference in this case is that we use stochastic collocation to compute the expectations. Hence, we are free to choose any system response function that can be approximated well by stochastic collocation. This removes the restriction that the function must lie in the RKHS, since stochastic collocation automatically ensures that the interpolant is constructed by projecting the function into this space. We choose arbitrary test functions to test the performance and then demonstrate the framework for models of micromechanical devices. As mentioned in Section 3.3.1, the additional benefit of choosing the RKHS that corresponds to the stochastic collocation interpolant is that the estimated PDFs reproduce the variance in addition to the mean. This is attributed to the fact that the interpolant for f^2 lies in the same space as that for f. Hence, for the multivariate case, we can also compare the median normalized error in variance (MNEV) between the three methods. The MNEV value is obtained using the same procedure used for MNEE described above, except that we do it for the variance instead of the mean. We show that KMM works better than KDE and AKDE in estimating the output statistics with greater accuracy, even when the size of the data set is limited. This makes the method ideal for use in the UQ of micromechanical devices.

4. Numerical results

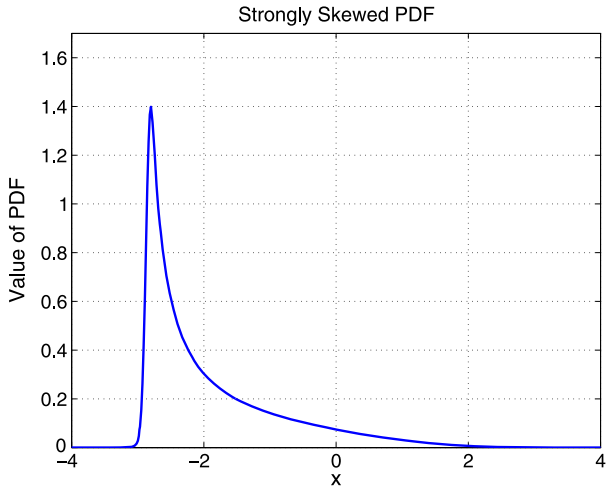
In this section, we look at some numerical examples where we compare the performance of the three density estimation methods discussed above. We start with density estimation for univariate PDFs, followed by bivariate PDFs. This will be followed by PDF estimation for a micromechanical device, where we generate samples from a known synthetic distribution and then perform density estimation on the samples to compare the results with the actual PDF. Finally, we look at a real



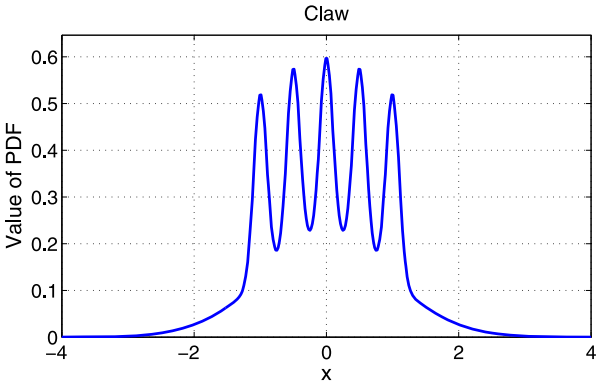
(a) Bi-modal PDF.



(b) Kurtotic uni-modal PDF.



(c) Strongly skewed PDF.



(d) Claw PDF.

Fig. 2. Standard PDFs used to test the performance of the estimation methods.

world example, where we look at actual data samples for the parameters of a micromechanical device and try to estimate the unknown PDF from which they were derived.

4.1. Univariate PDFs

In order to compare the performance of the three density estimation methods, we first choose the four known distributions shown in Fig. 2. We sample each of the distributions and estimate the density for each sample using each of the three density estimation methods. We then generate random functions in the RKHS and compute the expectations of these functions with respect to the three estimated PDFs. These values are compared with the empirical means of the functions to obtain the median normalized error in expectation. This value is compared across the three density estimation procedures for sample sizes varying from 5 to 160. For each case, the results are aggregated over 50 sample sets and 100 random functions. Thus we compute the statistics over 5000 values corresponding to each value of the sample size. A similar procedure is also done by comparing the estimated means with the actual means, which is the expectation of the random functions with respect to the known distribution. These results have been summarized in Table 1.

Table 1 compares the median errors in the estimated means to the empirical and actual means respectively. Typically, a lower value indicates that a particular method is performing better than the others. For each row, we also perform a paired sign test between the values to determine if the difference in performance between the three methods is significant. If one of the methods shows a significantly better performance, then this is indicated by representing its value in a bold font. When none of the values in a row are in a bold font, it means that there is no significant difference in the performance of the three methods for that particular case.

Table 1
MNEE computed with respect to empirical and actual mean for univariate PDFs.

PDF type	M	Empirical mean			Actual mean		
		KDE	AKDE	KMM	KDE	AKDE	KMM
Bimodal	5	0.42107657	0.41878547	**0.39873682**	**0.27104233**	0.27201588	0.27141127
	10	0.30087889	0.29989356	**0.29103855**	**0.18789780**	0.19434668	0.19557148
	20	0.23977959	0.24294416	**0.23622297**	0.16793487	0.16953210	**0.16172740**
	40	0.15827776	0.15802949	**0.15313689**	0.10418371	0.10619682	0.10886268
	80	0.12890512	0.12898494	**0.12984552**	0.09035235	0.09274833	0.09312720
	160	0.09088651	0.08993969	0.09044580	0.06289690	0.06020423	**0.05891577**
KUD	5	0.30586604	0.3000465	**0.25910434**	0.26412656	0.24823495	**0.21500147**
	10	0.21020434	0.19781770	**0.16034669**	0.17004458	0.15079007	**0.12368095**
	20	0.14914699	0.13577501	**0.11962624**	0.11990512	0.09767222	**0.08177646**
	40	0.09994164	0.09009459	**0.08217391**	0.08266654	0.06808542	**0.06146460**
	80	0.07111639	0.06518895	**0.06451986**	0.05039992	0.04383713	**0.04385121**
	160	0.04470673	0.03884934	**0.03949587**	0.03680616	0.02968527	**0.02973749**
SSD	5	0.36246449	0.35585839	**0.33250831**	0.25515661	0.24066083	**0.22182885**
	10	0.26578278	0.24623124	**0.22420096**	0.17798267	0.15410840	**0.13739876**
	20	0.17202963	0.16548228	**0.15918609**	0.12810928	0.10738178	**0.09736551**
	40	0.10677000	0.09479283	**0.08985452**	0.07125182	0.06284033	**0.06148162**
	80	0.07209760	0.06834409	**0.06633426**	0.04957669	**0.04562024**	0.04468647
	160	0.04795668	0.04638099	**0.04666429**	0.03377096	0.03075097	0.03143608
Claw	5	0.29165746	0.28868708	**0.25614023**	0.17523004	0.17702440	**0.15922803**
	10	0.19860527	0.19642110	**0.18293233**	**0.12192943**	0.12526554	0.12746289
	20	0.14351700	0.14095954	**0.13103473**	0.10381259	0.09901543	0.09077674
	40	0.10761135	0.09934702	**0.08575641**	0.07354243	0.06612561	**0.05930838**
	80	0.06461594	0.06164044	**0.05813168**	0.04604840	0.04466935	**0.03998803**
	160	0.05027149	0.04891416	**0.04536690**	0.03319306	0.03168688	**0.02962931**

We see from Table 1, that when comparing with the empirical means, the KMM method has the lowest values of MNEE and performs better than both KDE and AKDE in all the cases. Similarly, we conclude that it also works quite well when the comparison metric is with respect to the actual mean. There are only 4 cases out of 20, where either KDE or AKDE is doing better and most of them belong to the cases where the sample size is either 5 or 10. This indicates that the benefit obtained by KMM for these cases is counterbalanced by the inherent noise in the data due to the small number of samples. This shows that we need at least a reasonable number of samples in order to obtain consistent results with KMM.

4.2. Bivariate PDFs

We now consider the case of bivariate PDFs, where there are 2 random dimensions instead of one. We choose the PDF shown in Fig. 3, which is a product of two bimodal univariate PDFs. In this section and the following ones, we assume that when there are multiple random dimensions, the corresponding random variables are independent and hence the joint multivariate PDF is a product of individual univariate PDFs.

To test the performance of the estimation methods, we need to compute the expectations of functions in the random domain. Here, instead of choosing random functions that lie within the RKHS, we choose four standard test functions called the Genz test functions. The functions are shown in Fig. 4. The parameters of these functions are randomly generated and they do not lie within any standard RKHS. However, if we use stochastic collocation to approximate these functions, we project them into a subspace that is an RKHS associated with the Ornstein–Uhlenbeck kernel discussed earlier. Thus we extend the application of the method to any function, which can be approximated reasonably well using stochastic collocation. For the case of the four functions shown in Fig. 4, we construct interpolants using stochastic collocation such that the maximum interpolation error is lower than 10^{-7} in each case. This is to ensure that the numerical errors due to stochastic collocation are well below the errors in the estimated PDFs and hence, will not affect the statistics.

We generate the data for density estimation by drawing 200 sets of samples from the bivariate PDF, with the size of each set varying from 5 to 160. For each sample set, we estimate the PDFs using KDE, AKDE and KMM, and compute the expectations of each of the four test functions. These values are compared with respect to the empirical and actual means to obtain the error in expectations and the results are presented in Table 2. We see that the median error in the estimates generated using KMM are significantly lower than the corresponding values for KDE and AKDE. This trend is maintained for all the four test functions. The only cases where no method is significant are when the number of samples is very low.

One of the additional benefits of using KMM optimized for stochastic collocation is that not only does it reproduce the mean, but it is also optimal for higher moments, since they involve terms that are merely expectations of powers of the original function. Hence we compare the errors in the variances (MNEV) computed using the three estimation methods with

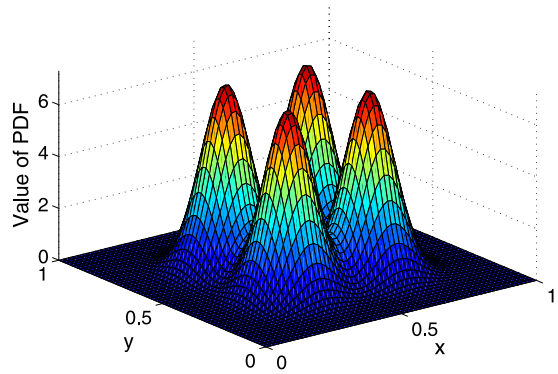
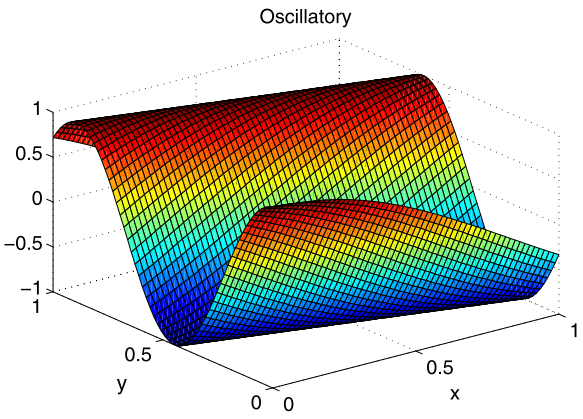
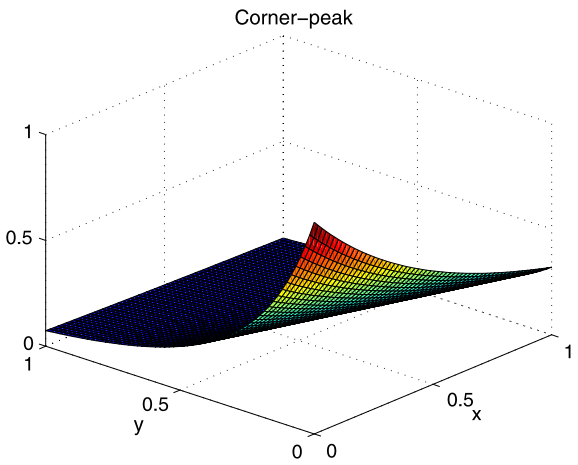


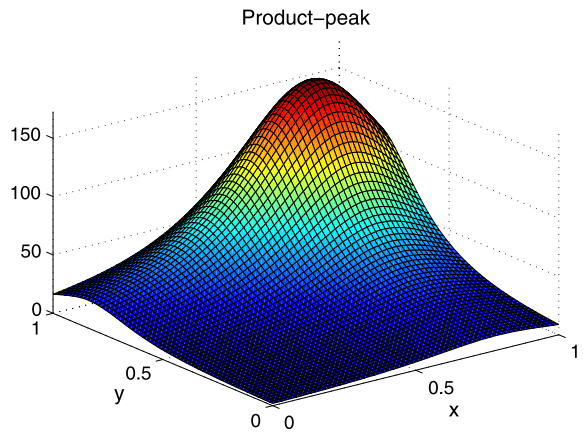
Fig. 3. 2D bimodal PDF used to test the performance of the estimation methods.



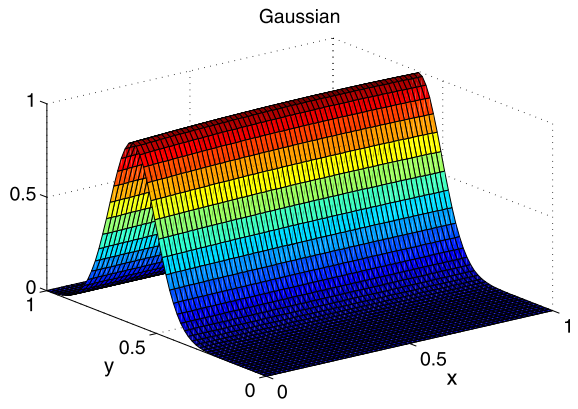
(a) Oscillatory.



(b) Corner peak.



(c) Product peak.



(d) Gaussian.

Fig. 4. Genz test functions used for computation of expectations in the 2D case.

the empirical and actual variances. These results are presented in Table 3. Similar to the results for the mean, the error in variance is lowest for PDFs estimated using KMM. This confirms that the KMM method optimized for stochastic collocation can accurately reproduce higher moments as well.

Table 2
MNEE computed with respect to empirical and actual mean for bivariate PDF.

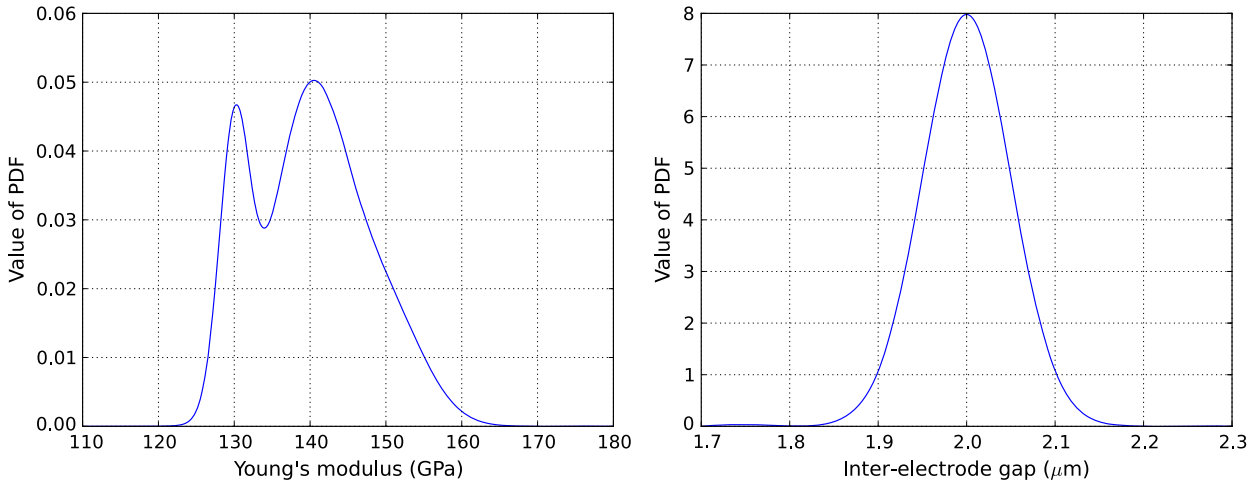
Test function	M	Empirical mean			Actual mean		
		KDE	AKDE	KMM	KDE	AKDE	KMM
Oscillatory	5	0.02428374	0.02512921	0.02773819	0.18071239	0.18160128	0.18171281
	10	0.01238129	0.01223823	0.01207922	0.12121725	0.12128283	0.11839372
	20	0.00697282	0.00704720	**0.00607294**	0.08598392	0.08628193	**0.07829371**
	40	0.00591729	0.00609203	**0.00497229**	0.05828129	0.05821348	**0.05164829**
	80	0.00513934	0.00502384	**0.00428399**	0.04153739	0.04148132	**0.03602382**
	160	0.00208292	0.00214859	**0.00124935**	0.02901790	0.02890129	**0.01927238**
Corner peak	5	0.00782382	0.00827239	0.00759234	0.04741893	0.04748797	0.04123923
	10	0.00369813	0.00392838	**0.00303949**	0.03098234	0.03097213	0.02946292
	20	0.00189723	0.00207547	**0.00168292**	0.02193620	0.02180193	**0.017033384**
	40	0.00143241	0.00118144	**0.00096284**	0.01619182	0.01611230	**0.013188302**
	80	0.00099372	0.00107382	**0.00082183**	0.01028230	0.01028690	**0.008026482**
	160	0.00061438	0.00073045	**0.00047592**	0.00808104	0.00817402	**0.004171918**
Product peak	5	0.13364384	0.14422999	**0.1214626**	0.30712382	0.30811239	**0.28182949**
	10	0.12002374	0.12147343	0.12039185	0.21617499	0.21952678	0.21127392
	20	0.06881239	0.06821287	**0.05729208**	0.17782888	0.17938304	**0.16971229**
	40	0.05318738	0.05589823	**0.04412920**	0.11459335	0.11648290	**0.09293920**
	80	0.03873276	0.03913728	**0.03271956**	0.07401409	0.07517265	**0.06618104**
	160	0.02612432	0.02758283	**0.01981827**	0.05889294	0.06117128	**0.05482910**
Gaussian	5	0.04301287	0.04239128	**0.02343748**	0.05372779	0.05229224	**0.04291920**
	10	0.01651240	0.01669023	**0.00913457**	0.02901341	0.02988123	**0.02681953**
	20	0.01033618	0.01061838	**0.00438674**	0.02278129	0.02294729	**0.01912281**
	40	0.00701023	0.00727549	**0.00220183**	0.01451828	0.01451038	**0.01318439**
	80	0.00558432	0.00540192	**0.00310137**	0.01023749	0.01032044	**0.00849227**
	160	0.00298101	0.00323740	**0.00082301**	0.00691139	0.00701282	**0.00528492**

Table 3
MNEV computed with respect to empirical and actual variance for bivariate PDF.

Test function	M	Empirical variance			Actual variance		
		KDE	AKDE	KMM	KDE	AKDE	KMM
Oscillatory	5	0.35787834	0.37738444	**0.25311598**	0.44697377	0.45763045	**0.27156778**
	10	0.2390894	0.24186769	**0.18601935**	0.18229442	0.18986481	**0.15079715**
	20	0.13009142	0.13347679	0.11763531	0.13663031	0.13735511	**0.10370658**
	40	0.09279845	0.09380968	**0.08034692**	0.08412601	0.08478872	**0.07768013**
	80	0.07267532	0.07037185	**0.05305658**	0.06326487	0.06568962	**0.05138308**
	160	0.04789011	0.04747552	**0.04474528**	0.04542822	0.04876326	**0.03762029**
Corner peak	5	0.71563634	0.78264406	**0.25643194**	0.68905756	0.69277839	**0.35558353**
	10	0.2768589	0.30134636	**0.14074007**	0.26361951	0.25187095	**0.18955332**
	20	0.15210102	0.1556804	**0.09197129**	0.20353475	0.20632385	**0.13943959**
	40	0.11389174	0.125408	**0.0594214**	0.09784787	0.10358764	**0.07856193**
	80	0.07052442	0.07927639	**0.04488021**	0.07678436	0.07735829	**0.06704293**
	160	0.04842303	0.05810011	**0.03229334**	0.05417571	0.05984178	**0.04159789**
Product peak	5	0.48681322	0.4935726	**0.39300303**	0.57614556	0.56880473	**0.4229983**
	10	0.29413249	0.28787331	**0.28198732**	0.27845821	0.27642848	0.28784277
	20	0.16584175	0.16423995	0.17140714	0.22156759	0.2159918	**0.19448553**
	40	0.14198772	0.1402631	**0.13564335**	0.14284206	0.13930311	**0.1184679**
	80	0.10237679	0.10271796	0.10052229	0.09141467	0.08919647	0.09585292
	160	0.06953516	0.06907068	**0.06436754**	0.06045815	0.06051215	**0.05523534**
Gaussian	5	0.87080819	0.92791772	**0.41638964**	0.61725005	0.6396841	**0.39423494**
	10	0.43581791	0.4368056	**0.2500894**	0.30756345	0.3067043	**0.23661505**
	20	0.2448594	0.260313	**0.13231394**	0.29101989	0.29054637	**0.192427**
	40	0.24475749	0.25543318	**0.17057539**	0.21531359	0.21278763	**0.14865019**
	80	0.21480841	0.21974602	**0.15871722**	0.21184851	0.21445405	**0.1529298**
	160	0.04903665	0.06578042	**0.02581183**	0.076862	0.0798696	**0.05734016**

4.3. Micromechanical switch with synthetic uncertainties

In this section, we look at UQ of micromechanical devices. We use the numerical model for an electrostatic microactuator, that was developed in [5]. The electrostatic microactuator consists of a deformable cantilever beam that is pulled towards a fixed ground plate due to electrostatic traction that is generated by the application of an external potential difference between the electrodes. We consider the Young's modulus of the deformable electrode and the dimension of the air gap



(a) Young's modulus. (b) Inter-electrode gap.

Fig. 5. Synthetic PDFs used to generate samples for performing UQ in an electrostatic microswitch.

between the electrodes as two independent sources of uncertainty. In a real device, we have no idea about the nature of the uncertainty. Consequently, we do not have a known PDF from which we can draw samples. Since our goal is to compare the performance of the estimation methods for micromechanical devices, we assign synthetic PDFs to each of the uncertain parameters and draw samples from the joint density. The PDFs that were chosen for the Young's modulus and the air gap are shown in Fig. 5.

We first draw 200 sets of samples of sizes ranging from 5 to 160 using the same procedure mentioned in the previous sections. We then perform density estimation using KDE, AKDE and KMM. The nominal dimensions of the microactuator are chosen to be such that the movable electrode is 100 μm long and 1 μm thick. It is separated from the ground plate by a mean distance of 2 μm. We consider two cases, in which the externally applied potential difference between the two electrodes is 6 V and 10 V respectively, and we look at the uncertainty in the displacement of the tip of the movable electrode. We then propagate the estimated PDFs for the Young's modulus and the inter-electrode gap through the numerical model of the device using stochastic collocation. We compute the MNEE of the output displacement with respect to the empirical mean as well as the actual mean, which corresponds to the known forms of the PDFs shown in Fig. 5. The results are shown in Table 4.

We see that for most values of M, the KMM method estimates the mean displacement as well as the variance with lowest error. Using the paired sign test, we also determine that the difference in performance is significant for most of the cases, except for very small sample sets with 5 and 10 samples. This is consistent with results observed in the previous sections, where it was seen that small sample sets have a large amount of sampling variation, which affects the final result. Aside from these extreme cases, we see a reasonable reduction in the error estimates for the PDFs generated using KMM. This benefit holds true when computing both the empirical moments as well as the actual ones. This gives us confidence that even when we do not know the actual PDF that models the uncertainty in a parameter, the data-driven PDF estimated using KMM comes closest to replicating the statistics in the micromechanical device.

4.4. Micromechanical switch with data-driven uncertainties

In this final section, we demonstrate the UQ process for micromechanical devices where the actual nature of uncertainty is not known. In addition to the electrostatic microactuator considered in Section 4.3, we also consider the hybrid electrothermomechanical (ETM) actuator [5], which uses a combination of electrostatic traction and electrothermal expansion for actuation. This actuator also comprises two electrodes, where one of them is a fixed ground plate, while the other is movable and shaped similar to the Guckel electrothermal actuator [39]. This electrode has two terminals and by applying a potential difference between them, the electrode undergoes electrothermal expansion that causes it to move towards the ground plate. The presence of the grounded plate next to the movable electrode also causes electrostatic traction, which leads to additional displacement. Thus, the actuator moves due to the combination of electrostatic and electrothermal actuation.

The device that we consider in this paper is identical to the one in [5], where a complete discussion of numerical model of the device may be found. The only difference here is that although there are temperature gradients in the device, we do not consider the variation in material properties with temperature for the sake of simplicity. We assume that the Young's modulus and inter-electrode gap are variable in both the electrostatic microactuator and the hybrid ETM device. We also consider the thermal conductivity in the hybrid ETM actuator to be a source of uncertainty. We use the characterization data given in [16] to generate data-driven PDFs for each of the material parameters. The PDF for Young's modulus is generated

Table 4
MNEE/MNEV with respect to empirical and actual moments for electrostatic microswitch with data from synthetic PDFs.

Voltage	M	KDE	AKDE	KMM	KDE	AKDE	KMM
		Empirical mean			Actual mean		
$V = 6$	5	0.00368273	0.00386583	0.00421364	0.02152349	0.02151828	0.02038219
	10	0.00261298	0.00282389	**0.00182721**	0.01369292	0.01391938	0.01316382
	20	0.00125245	0.00167819	**0.00119387**	0.00798191	0.00797238	**0.00718388**
	40	0.00083204	0.00104109	**0.00072481**	0.00673478	0.00672992	**0.00562819**
	80	0.00069103	0.00072348	**0.00047153**	0.00511929	0.00528458	**0.00417824**
	160	0.00053248	0.00077291	**0.00032819**	0.00316583	0.00327211	**0.00227184**
$V = 10$	5	0.00359202	0.00391729	0.00361839	0.02192920	0.02119192	0.02082717
	10	0.00269191	0.00280239	**0.00172819**	0.01302738	0.01346382	0.01290183
	20	0.00121728	0.00168459	**0.00110337**	0.00797384	0.00792398	**0.00701368**
	40	0.00081230	0.00107268	**0.00069828**	0.00671291	0.00675749	**0.00589210**
	80	0.00069504	0.00072839	**0.00038718**	0.00511923	0.00521018	**0.00421859**
	160	0.00054375	0.00078193	**0.00028928**	0.00313739	0.00322399	**0.00201838**
		Empirical variance			Actual variance		
$V = 6$	5	1.29679953	1.31916525	**0.38607979**	1.07307297	1.04691617	**0.29729643**
	10	0.69574413	0.68707174	**0.25481894**	0.68980756	0.68504184	**0.14530951**
	20	0.39337067	0.46185852	**0.14247495**	0.27541241	0.32637438	**0.16933753**
	40	0.26190165	0.31704523	**0.07883383**	0.19217039	0.23392704	**0.09312244**
	80	0.13077288	0.17885524	**0.06401951**	0.09312679	0.13080898	**0.07061041**
	160	0.1143963	0.16754132	**0.05207364**	0.05895633	0.08286171	0.05137748
$V = 10$	5	1.27890204	1.33025914	**0.39406603**	1.09324049	1.11874026	**0.25787827**
	10	0.71691072	0.71414789	**0.24349418**	0.6916739	0.68568407	**0.10534344**
	20	0.40290203	0.4647116	**0.12395943**	0.28781397	0.33653517	**0.18085136**
	40	0.2529714	0.31219072	**0.08282513**	0.18453605	0.22629601	**0.10200548**
	80	0.13250213	0.19267291	**0.06490482**	0.09365743	0.11802317	**0.06799152**
	160	0.10825347	0.15913895	**0.04615518**	0.05819478	0.0716369	0.0486542

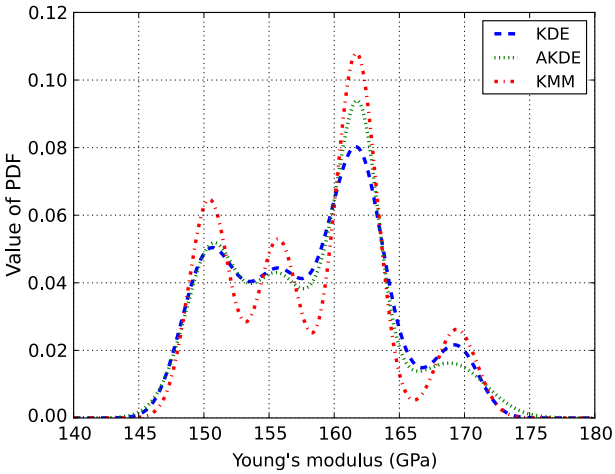
Table 5
Empirical MNEE for electrostatic and hybrid ETM microswitches, with respect to the data-driven PDFs.

Device	Voltage (V)	KDE	AKDE	KMM
Electrostatic	6	1.48120112e−03	1.44347645e−03	**4.25203117e−05**
	7	1.38503853e−03	1.43483850e−03	**1.07210444e−04**
	8	1.35268691e−03	1.46323658e−03	**1.50383339e−04**
	9	1.45205334e−03	1.51699939e−03	**1.86809383e−04**
	10	1.52665983e−03	1.60255455e−03	**2.10348398e−04**
Hybrid ETM	6	1.158555e−02	1.232756e−02	**1.113008e−02**
	7	1.162554e−02	1.238204e−02	**1.118794e−02**
	8	1.166544e−02	1.240749e−02	**1.119719e−02**
	9	1.166441e−02	1.240397e−02	**1.119480e−02**
	10	1.140687e−02	1.250298e−02	**1.101946e−02**

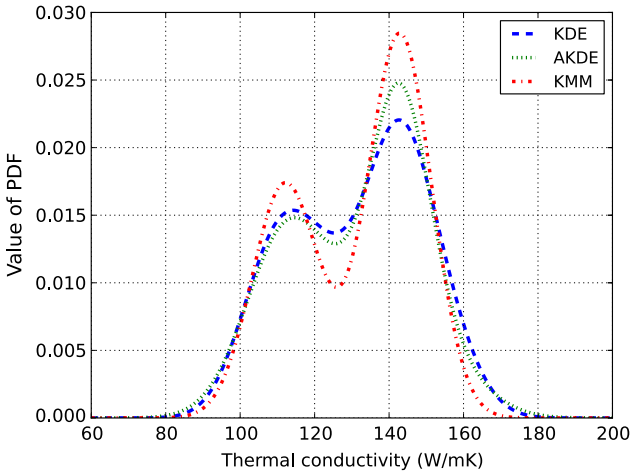
using a dataset with 31 measurement samples, while the thermal conductivity dataset contains 23 samples. Since there is no readily available data for the variation in the inter-electrode gap, we simply use a set of 50 values that are uniformly sampled in the range [1.8, 2.2] μm. The resulting PDFs are shown in Fig. 6.

Since the data-driven PDFs are generated from a single sample of data, in this section we do not vary the size of the data sample. Instead, for both actuators, we vary the potential difference from 6 to 10 V and compute the expected mean and variance of the tip displacement in each case. Except for the inter-electrode gap, we do not know the actual PDF from which the samples were generated. Hence, we only compare the MNEE and MNEV values with respect to the empirical moments. The results are presented in Table 5 and Table 6, respectively.

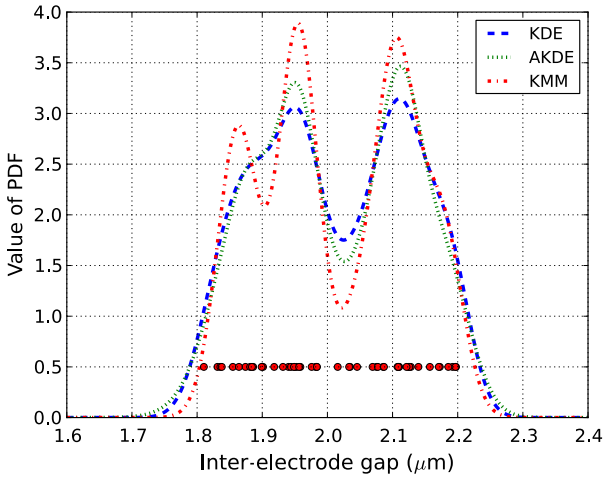
We see that the error in the estimated moments is lowest for the KMM method and the difference is significant for all values of the potential difference. The latter observation is expected because as long as the estimated PDF is optimal, it should work well for all values of the potential difference. In the hybrid ETM actuator case, compared to KDE and AKDE, we see that there is not much reduction in error due to KMM, although the reduction is still significant according to the paired sign test. We also note that the overall magnitudes of the normalized errors are much larger in the hybrid ETM actuator. This difference could be caused by inaccuracies in the PDF estimated for thermal conductivity, which may in turn arise from an insufficient number of samples. As demonstrated in Section 4.3, these inaccuracies readily decrease as the size of the sample set used for PDF estimation increases. Although we cannot compare the above estimates with respect to the actual PDF, we see that the KMM method does a good job of producing data-driven PDFs that reproduce the sample means.



(a) Young's modulus.



(b) Thermal conductivity.



(c) Inter-electrode gap.

Fig. 6. Data-driven PDFs used to generate samples for performing UQ in an electrostatic microswitch.

Table 6
Empirical MNEV for electrostatic and hybrid ETM microswitches, with respect to the data-driven PDFs.

Device	Voltage (V)	KDE	AKDE	KMM
Electrostatic	6	0.00992424	0.01039842	**0.00031457**
	7	0.00897439	0.00989154	**0.00081739**
	8	0.00958137	0.01102392	**0.00113849**
	9	0.0109684	0.01079084	**0.00128029**
	10	0.01058963	0.01237423	**0.00152929**
Hybrid ETM	6	0.12437816	0.14104549	**0.10758425**
	7	0.12120495	0.12604988	**0.11727252**
	8	0.12717478	0.12808531	**0.11537662**
	9	0.12317171	0.13541386	**0.1186502**
	10	0.1214178	0.12729905	**0.11652573**

5. Conclusions

In this paper, we develop a complete data-driven approach for the estimation and propagation of uncertainties using stochastic collocation. Performing UQ in micromechanical devices is hindered by the fact that the amount of data that is available to characterize the uncertainties is limited. Given this limitation, we restrict the goal of UQ to accurately estimating the statistics of a parameter of interest. We choose the kernel moment matching method to ensure that the resulting PDFs will be optimal in terms of reproducing the statistics for all functions that belong to a known RKHS. One of the novel

contributions in this paper, is to identify an RKHS that is specific to stochastic collocation. This ensures that irrespective of the properties of the actual system response function, the PDFs may be tuned to be optimal for the piecewise linear form of the interpolant constructed using stochastic collocation. This greatly extends the scope of the KMM method, because we do not need to know the specific function class of the actual system response, as long as it is possible to approximate it well using collocation. An added benefit is that the PDFs are also optimal for calculating the variance and higher order moments, since powers of the response function can also be approximated by the same interpolant.

We present results that demonstrate the reduction in the median error in the estimated mean and variance, when compared to the empirical and actual values. We see that KMM performs better than other estimation methods like KDE and AKDE for almost all the cases. When the number of samples is very low, we see that no method is significantly better. This is likely due to the inherent noise in the sample sets since they are so small. For larger sample sizes, we readily see an improvement in the case of KMM. This has been shown for both univariate PDFs as well as multivariate ones. Finally, we demonstrate how this method can be used to improve the accuracy of UQ in micromechanical devices. Since UQ in this field is plagued by problems relating to insufficient data, it is expected that this method will help improve the accuracy of estimated statistics.

Acknowledgements

This work is supported by the National Science Foundation under grant numbers 0810294 and 0941497, and by the Department of Energy. The authors gratefully acknowledge the use of the parallel computing resource provided by the Computational Science and Engineering Program at the University of Illinois. The CSE computing resource, provided as part of the Taub cluster, is devoted to high performance computing in engineering and science.

Appendix A. Derivation of the KMM objective function

We start with the KMM objective function given in Eq. (11). Rewriting the norm in terms of the inner product operator, we get,

$$\min_{\boldsymbol{\alpha}} \langle \mu[X] - \mu[\hat{\rho}], \mu[X] - \mu[\hat{\rho}] \rangle_{\mathcal{H}}; \qquad \sum_{i=1}^{M} \alpha_i = 1,\ \alpha_i > 0. \tag{17}$$

Since the inner product is a linear operator, we can expand the above expression as,

$$\min_{\boldsymbol{\alpha}} \langle \mu[X], \mu[X] \rangle_{\mathcal{H}} - 2\langle \mu[X], \mu[\hat{\rho}] \rangle_{\mathcal{H}} + \langle \mu[\hat{\rho}], \mu[\hat{\rho}] \rangle_{\mathcal{H}}; \qquad \sum_{i=1}^{M} \alpha_i = 1,\ \alpha_i > 0. \tag{18}$$

The first term is a constant that depends only on X and is independent of $\boldsymbol{\alpha}$. Hence, it can be dropped from the objective function. We now consider Eq. (10), in which $\hat{\rho}$ is expressed as $\sum_{i=1}^{M} \alpha_i \rho_i$, where ρ_i is defined as $K_{\boldsymbol{\Sigma}}(\boldsymbol{x} - \boldsymbol{x}_i)$. Plugging this expression into $\mu[\hat{\rho}]$ and noting that this is a linear transformation, we can write,

$$\mu[\hat{\rho}] = \sum_{i=1}^{M} \alpha_i \mu[\rho_i] = \sum_{i=1}^{M} \alpha_i \mathbb{E}_{\boldsymbol{x} \sim \rho_i}\big[k(\boldsymbol{x}, \cdot)\big]. \tag{19}$$

We can now re-write Eq. (18) as,

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{M} \sum_{j=1}^{M} \alpha_i \alpha_j \langle \mu[\rho_i], \mu[\rho_j] \rangle_{\mathcal{H}} - 2 \sum_{j=1}^{M} \alpha_j \langle \mu[X], \mu[\rho_j] \rangle_{\mathcal{H}}; \qquad \sum_{i=1}^{M} \alpha_i = 1,\ \alpha_i > 0, \tag{20}$$

which can also be written in matrix form as,

$$\min_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T \boldsymbol{Q}\, \boldsymbol{\alpha} - 2\boldsymbol{l}^T \boldsymbol{\alpha}; \qquad \sum_{i=1}^{M} \alpha_i = 1,\ \alpha_i > 0, \tag{21}$$

where $Q_{ij} = \langle \mu[\rho_i], \mu[\rho_j] \rangle_{\mathcal{H}}$ and $l_j = \langle \mu[X], \mu[\rho_j] \rangle_{\mathcal{H}}$ are the elements of the $M \times M$ matrix, $\boldsymbol{Q}$, and the $M \times 1$ vector, $\boldsymbol{l}$, respectively. This can be cast into the usual form of a quadratic programming problem as,

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \boldsymbol{\alpha}^T (\boldsymbol{Q} + \gamma \boldsymbol{I}) \boldsymbol{\alpha} - \boldsymbol{l}^T \boldsymbol{\alpha}; \qquad \sum_{i=1}^{M} \alpha_i = 1,\ \alpha_i > 0, \tag{22}$$

where $\gamma > 0$ is the regularization constant and $\boldsymbol{I}$ is the identity matrix of size $M \times M$. To simplify the expressions for Q_{ij} and l_j, we use the fact that $\mu[X] = \frac{1}{M} \sum_{i=1}^{M} k(\boldsymbol{x_i}, \cdot)$ and $\mu[\rho_i] = \mathbb{E}_{\boldsymbol{x} \sim \rho_i}[k(\boldsymbol{x}, \cdot)]$. We can then express Q_{ij} as,

$$\begin{aligned} Q_{ij} &= \left\langle \mathbb{E}_{\boldsymbol{x}\sim\rho_i}\left[k(\boldsymbol{x},\cdot)\right], \mathbb{E}_{\boldsymbol{x}'\sim\rho_j}\left[k(\boldsymbol{x}',\cdot)\right]\right\rangle_{\mathcal{H}} \\ &= \mathbb{E}_{\boldsymbol{x}\sim\rho_i,\boldsymbol{x}'\sim\rho_j}\left[\left\langle k(\boldsymbol{x},\cdot), k(\boldsymbol{x}',\cdot)\right\rangle_{\mathcal{H}}\right] \\ &= \mathbb{E}_{\boldsymbol{x}\sim\rho_i,\boldsymbol{x}'\sim\rho_j}\left[k(\boldsymbol{x},\boldsymbol{x}')\right], \end{aligned} \tag{23}$$

where we have used the reproducing property of the RKHS to reduce the expression in the last step. Similarly we can simplify the expression for l_j as,

$$l_j = \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_{\boldsymbol{x}'\sim\rho_j}\left[k(\boldsymbol{x}_i,\boldsymbol{x}')\right]. \tag{24}$$

The expectations can be evaluated by computing the corresponding multivariate integrals, given by,

$$Q_{ij} = \int_{\mathbb{R}^M}\int_{\mathbb{R}^M} k(\boldsymbol{x},\boldsymbol{x}') K_{\boldsymbol{\Sigma}}(\boldsymbol{x}-\boldsymbol{x}_i) K_{\boldsymbol{\Sigma}}(\boldsymbol{x}'-\boldsymbol{x}_j)\, d\boldsymbol{x}\, d\boldsymbol{x}', \tag{25}$$

$$l_j = \frac{1}{M}\sum_{i=1}^{M}\int_{\mathbb{R}^M} k(\boldsymbol{x}_i,\boldsymbol{x}') K_{\boldsymbol{\Sigma}}(\boldsymbol{x}'-\boldsymbol{x}_j)\, d\boldsymbol{x}'. \tag{26}$$

For a given covariance matrix, $\boldsymbol{\Sigma}$, all the quantities in the above expressions are known and the integrals can easily be evaluated using numerical quadrature. We can thus solve the quadratic program given in Eq. (22) to get the weights and thereafter, compute the estimated density.

References

[1] S.D. Senturia, N. Aluru, J. White, Simulating the behavior of MEMS devices: Computational methods and needs, IEEE Comput. Sci. Eng. 4 (January 1997) 30–43.
[2] G. Li, N. Aluru, Efficient mixed-domain analysis of electrostatic MEMS, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 22 (November 2003) 1228–1242.
[3] S. De, N.R. Aluru, Full-Lagrangian schemes for dynamic analysis of electrostatic MEMS, J. Microelectromech. Syst. 13 (October 2004) 737–758.
[4] S. Telukunta, S. Mukherjee, Fully Lagrangian modeling of MEMS with thin plates, J. Microelectromech. Syst. 15 (4) (2006) 795–810.
[5] A. Alwan, N. Aluru, Analysis of hybrid electrothermomechanical microactuators with integrated electrothermal and electrostatic actuation, J. Microelectromech. Syst. 18 (5) (2009) 1126–1136.
[6] M. Liu, K. Maute, D.M. Frangopol, Multi-objective design optimization of electrostatically actuated microbeam resonators with and without parameter uncertainty, Reliab. Eng. Syst. Saf. 92 (10) (2007) 1333–1343.
[7] J. Han, B. Kwak, Robust optimal design of a vibratory microgyroscope considering fabrication errors, J. Micromech. Microeng. 11 (6) (2001) 662.
[8] J. Wittwer, M. Baker, L. Howell, Robust design and model validation of nonlinear compliant micromechanisms, J. Microelectromech. Syst. 15 (1) (2006) 33–41.
[9] D. Xiu, G. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2) (2002) 619–644.
[10] D. Xiu, G. Karniadakis, Modeling uncertainty in flow simulations via generalized polynomial chaos, J. Comput. Phys. 187 (May 2003) 137–167.
[11] X. Wan, G. Karniadakis, An adaptive multi-element generalized polynomial chaos method for stochastic differential equations, J. Comput. Phys. 209 (2) (2005) 617–642.
[12] D. Xiu, J. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM J. Sci. Comput. 27 (2005) 1118–1139.
[13] B. Ganapathysubramanian, N. Zabaras, Sparse grid collocation schemes for stochastic natural convection problems, J. Comput. Phys. 225 (1) (2007) 652–685.
[14] X. Ma, N. Zabaras, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, J. Comput. Phys. 228 (8) (2009) 3084–3113.
[15] N. Agarwal, N. Aluru, A domain adaptive stochastic collocation approach for analysis of MEMS under uncertainties, J. Comput. Phys. 228 (20) (2009) 7662–7688.
[16] A. Alwan, N. Aluru, Uncertainty quantification of MEMS using a data-dependent adaptive stochastic collocation method, Comput. Methods Appl. Mech. Eng. 200 (45) (2011) 3169–3182.
[17] E. Parzen, On estimation of a probability density function and mode, Ann. Math. Stat. 33 (3) (1962) 1065–1076.
[18] D. Scott, G. Terrell, Biased and unbiased cross-validation in density estimation, J. Am. Stat. Assoc. 82 (400) (1987) 1131–1146.
[19] S. Sheather, M. Jones, A reliable data-based bandwidth selection method for kernel density estimation, J. R. Stat. Soc. B 53 (3) (1991) 683–690.
[20] M.P. Wand, M.C. Jones, Kernel Smoothing, Chapman & Hall, London, 1995.
[21] I.S. Abramson, On bandwidth variation in kernel estimates – A square root law, Ann. Stat. 10 (4) (1982) 1217–1223.
[22] O. Barndorff-Nielsen, Information and Exponential Families in Statistical Theory, Wiley, Chichester, New York, 1978.
[23] M. Dudik, S. Phillips, R. Schapire, Performance guarantees for regularized maximum entropy density estimation, in: Learning Theory, 2004, pp. 472–486.
[24] K. Kokkinakis, A. Nandi, Exponent parameter estimation for generalized Gaussian probability density functions with application to speech modeling, Signal Process. 85 (9) (2005) 1852–1858.
[25] L. Song, X. Zhang, A. Smola, A. Gretton, B. Schölkopf, Tailoring density estimation via reproducing kernel moment matching, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 992–999.
[26] D. Xiu, Efficient collocational approach for parametric uncertainty analysis, Commun. Comput. Phys. 2 (2) (2007) 293–309.
[27] S. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, Sov. Math. Dokl. 4 (1963) 240–243.
[28] P. Gajda, Smolyak's algorithm for weighted l_1-approximation of multivariate functions with bounded rth mixed derivatives over $\mathbb{R}^d$, Numer. Algorithms 40 (2005) 401–414.
[29] V. Barthelmann, E. Novak, K. Ritter, High dimensional polynomial interpolation on sparse grids, Adv. Comput. Math. 12 (4) (2000) 273–288.
[30] R. Duin, On the choice of smoothing parameters for parzen estimators of probability density functions, IEEE Trans. Comput. 100 (11) (1976) 1175–1179.

[31] N. Agarwal, N. Aluru, A data-driven stochastic collocation approach for uncertainty quantification in MEMS, Int. J. Numer. Methods Eng. 83 (5) (2010) 575–597.
[32] D. Comaniciu, V. Ramesh, P. Meer, The variable bandwidth mean shift and data-driven scale selection, in: Proceedings of the Eighth IEEE International Conference on Computer Vision, vol. 1, IEEE, 2001, pp. 438–445.
[33] Y. Altun, A. Smola, Unifying divergence minimization and statistical inference via convex duality, in: Learning Theory, 2006, pp. 139–153.
[34] M. Anderson, J. Dahl, L. Vandenberghe, CVXOPT: Python software for convex optimization, 2009.
[35] R. Byrd, P. Lu, J. Nocedal, C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM J. Sci. Comput. 16 (5) (1995) 1190–1208.
[36] E. Jones, T. Oliphant, P. Peterson, et al., SciPy: Open source scientific tools for Python, 2001.
[37] G.E. Fasshauer, Green's functions: Taking another look at kernel approximation, radial basis functions, and splines, in: M. Neamtu, L. Schumaker (Eds.), Approximation Theory XIII: San Antonio 2010, in: Springer Proceedings in Mathematics, vol. 13, Springer, New York, 2012, pp. 37–63.
[38] J. Gibbons, S. Chakraborti, Nonparametric Statistical Inference, Marcel Dekker Inc., New York, 1992 (Ch. 5).
[39] H. Guckel, J. Klein, T. Christenson, K. Skrobis, M. Laudon, E. Lovell, Thermo-magnetic metal flexure actuators, in: 5th Technical Digest, Solid-State Sensor and Actuator Workshop, June 1992, pp. 73–75.