



Data-driven stochastic inversion via functional quantization

Mohamed Reda El Amri, Celine Helbert, Olivier Lepreux, Miguel Munoz Zuniga, Clémentine Prieur, Delphine Sinoquet

► To cite this version:

Mohamed Reda El Amri, Celine Helbert, Olivier Lepreux, Miguel Munoz Zuniga, Clémentine Prieur, et al.. Data-driven stochastic inversion via functional quantization. *Statistics and Computing*, 2020, 30 (3), pp.525-541. 10.1007/s11222-019-09888-8 . hal-01704189v1

HAL Id: hal-01704189

<https://ifp.hal.science/hal-01704189v1>

Submitted on 19 Sep 2019 (v1), last revised 2 Feb 2021 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Data-driven stochastic inversion via functional quantization

Mohamed Reda El Amri · Céline Helbert · Olivier Lepreux · Miguel Munoz
Zuniga · Clémentine Prieur · Delphine Sinoquet

Received: date / Accepted: date

Abstract In this paper, we propose a new methodology for solving stochastic inversion problems through computer experiments, the stochasticity being driven by a functional random variables. This study is motivated by an automotive application. In this context, the simulator code takes a double set of simulation inputs: deterministic control variables and functional uncertain variables. This framework is characterized by two features. The first one is the high computational cost of simulations. The second is that the probability distribution of the functional input is only known through a finite set of realizations. In our context, the inversion problem is formulated by considering the expectation over the functional random variable. We aim at solving this problem by evaluating the model on a design, whose adaptive construction combines the so-called Stepwise Uncertainty Reduction (SUR) methodology with a strategy for an efficient expectation estimation. Two greedy strategies are introduced to sequentially estimate the expectation over the functional uncertain variable by adaptively selecting curves from the

initial set of realizations. Both of these strategies consider functional principal component analysis as a dimensionality reduction technique assuming that the realizations of the functional input are independent realizations of the same continuous stochastic process. The first strategy is based on a greedy approach for functional data-driven quantization, while the second one is linked to the notion of space filling design. Functional PCA is used as an intermediate step. For each point of the design built in the reduced space, we select the corresponding curve from the sample of available curves, thus guaranteeing the robustness of the procedure to dimension reduction. The whole methodology is illustrated and calibrated on an analytical example. It is then applied on the automotive industrial test case where we aim at identifying the set of control parameters leading to meet the pollutant emission standards of a vehicle.

Keywords functional random variable · Karhunen-Loève expansion · data reduction · functional quantization · set estimation · Gaussian process models.

Mohamed Reda El Amri
IFPEN, Rueil-Malmaison, France,
Université Grenoble Alpes, Laboratoire Jean Kuntzmann, Grenoble,
France
E-mail: mohamed-reda.el-amri@ifpen.fr

Céline Helbert
Université de Lyon, UMR 5208, Ecole Centrale de Lyon, Institut
Camille Jordan

Olivier Lepreux
IFPEN, Solaize, France

Miguel Munoz Zuniga
IFPEN, Rueil-Malmaison, France

Clémentine Prieur
Université Grenoble Alpes, Laboratoire Jean Kuntzmann, Grenoble,
France

Delphine Sinoquet
IFPEN, Rueil-Malmaison, France

1 Introduction

In recent years, computer models are omnipresent in engineering and sciences, because the corresponding physical experimentation is costly or even impossible to execute. Indeed, numerical simulations are often used to replace physical experiments as underlined in [3, 7]. Practitioners are not only interested in the response of their model for a given set of inputs (forward problem) but also in recovering the set of input values leading to a prescribed value or range of the output of interest. The problem of estimating such a set is called hereafter inversion problem.

We will consider a system that evolves in an uncertain environment, the uncertainties appear for example due to manufacturing tolerances or environmental conditions. The nu-

merical simulator modelling the system, denoted f , takes two types of input variables: a set of control variables $\mathbf{x} \in \mathbb{X}$, and a set of uncertain variables $\mathbf{v} \in \mathcal{V}$. Making any distributional assumptions for the uncertain variable \mathbf{v} , robust inversion consists in seeking the set of control variables $\mathbf{x} \in \mathbb{X}$ such that $\sup_{\mathbf{v} \in \mathcal{V}} f(\mathbf{x}, \mathbf{v})$ is smaller than a threshold c . Then, the difficulty of solving the robust inversion problem strongly depends on the uncertainty set \mathcal{V} . In our setting, \mathcal{V} is a functional space, and we consider instead the inversion problem under uncertainty as a stochastic inversion problem, assuming that the uncertainty has a probabilistic description. Let V denote the associated random variable, valued in \mathcal{V} , modelling the uncertainty. In our framework, we are interested in recovering the set $\Gamma^* := \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) = \mathbb{E}_V[f(\mathbf{x}, V)] \leq c\}$, with $c \in \mathbb{R}$. The functional random variable V is only known through a set of realizations and the expectation has to be estimated. Moreover, the simulations are time consuming and thus the usual Monte Carlo method to estimate the expectation ought to be avoided. Many reviews have been published to address this issue. Among the numerous techniques, the paper will focus on the ones based on the choice of a finite representative set of realizations of V , among the ones available. More precisely these approaches aim at minimizing the expected distance between a random draw from the probability distribution of V and this finite set. In the case of vector-valued random variables, this type of methods, introduced in [12] as principal points, was employed in various statistical applications, including quantizer design [20, 26] and stratified sampling. It is increasingly used for many engineering applications, where we are often faced with the challenge of working with *big data*. It is then necessary to reduce big data to manageable data. As for the case of functional random variables, various studies have been done in the Gaussian case (see [21, 27] and references therein). Here we work in the special case where the functional variable V could be non Gaussian and is only known through finite sample of realizations. This paper proposes two new methodologies to perform this data reduction or quantization for functional random variable and we investigate their performance in terms of integration errors. Inversion problems have already been carried out in many applications, notably reliability engineering (see, e.g., [3], [7]), climatology (see, e.g., [4], [13]) and many other fields. In the literature, one way to solve the problem is to adopt a sequential sampling strategy based on Gaussian process emulators. The underlying idea is that Gaussian process emulators, which capture prior knowledge about the regularity of the unknown function $g : \mathbf{x} \mapsto \mathbb{E}_V[f(\mathbf{x}, V)]$, make it possible to assess the uncertainty about Γ^* given a set of evaluations of g . More specifically, these sequential strategies for the estimation of an excursion set are closely related to the field of Bayesian global optimization (see, e.g., [9]). In the case of inversion problems, Stepwise Uncertainty Reduction (SUR)

strategies based on set measures were introduced in [33]. More recently, a parallel implementation of these strategies has been proposed in [7] and applied to the problem of recovery of an excursion set. Briefly, the strategy SUR gives sequentially the next location in the control space where to estimate the function g in order to minimize an uncertainty function. The key contribution of the present paper is to propose a data-driven adaptation of that procedure in the presence of functional uncertainties.

The paper is divided into five sections. Following this introduction, Section 2 is devoted to the introduction of two new adaptive methods to choose the finite representative set of the functional random variable for a reliable expectation estimation. In Section 3, we highlight the integration performance of our methods comparing to the standard Monte Carlo and to an existing method based on a probabilistic modelling with truncated principal component analysis (PCA). In Section 4, we introduce the Bayesian framework and fundamental notions for Stepwise Uncertainty Reduction (SUR) infill strategies in the context of computationally costly simulations. In Section 5, we introduce the new proposed data-driven methodology for stochastic inversion under functional uncertainties and describe our algorithm. Finally, in Section 6, we illustrate the overall procedure on an analytical example and then apply it to an industrial test case.

2 Functional data reduction

In this section, we introduce new data reduction strategies for functional data in a greedy fashion. The first one is based on notion coming from functional quantization. The second one is related to notion of space filling design. In this paper, data reduction aims at reducing the integration error when computing $\mathbb{E}[f(\mathbf{x}, V)]$. Therefore, we focus in Section 3 on the performance in terms of integration error of our strategies, comparing to standard procedures.

Context We consider the space $\mathcal{H} = \mathbb{L}^2(\Omega, \mathcal{F}, \mathbb{P}; \mathcal{V})$ of random processes V with realizations $V(\cdot, \omega) = \mathbf{v}$ in the space of deterministic square-integrable functions defined on $[0, T]$ denoted with $\mathcal{V} = \mathbb{L}^2([0, T])$ and equipped with norm $\|\mathbf{v}\| = (\int_0^T \mathbf{v}(t)^2 dt)^{1/2}$. The random variables $V(t, \cdot) = \eta$ lie in $\mathbb{L}^2(\Omega)$, the space of random variables with finite mean and variance, defined on $(\Omega, \mathcal{F}, \mathbb{P})$ and equipped with norm $\|\eta\|_{\mathbb{L}^2(\Omega)} = (\int_{\Omega} \eta^2 d\mathbb{P})^{1/2}$. All random processes discussed in this paper lie in \mathcal{H} which is equipped with norm

$$\|V\|_{\mathbb{L}^2} = (\mathbb{E}[\|V\|^2])^{1/2} = \left(\mathbb{E} \left[\int_0^T V(t)^2 dt \right] \right)^{1/2}, \quad (1)$$

for any $V \in \mathcal{H}$. The vast majority of realistic engineering problems can be addressed within this set of assumptions. Without loss of generality, we will consider a centred

stochastic process with finite variance. We aim at summarizing the distribution of V through a finite collection of deterministic functions $\{\mathbf{v}_j\}_{j=1}^l$ and corresponding weights $\{w_j\}_{j=1}^l$. Many reviews have been done on *functional quantization* [27, 21, 28]. For instance, Luschgy, Pagès and Wilbertz [21] propose different strategies for Gaussian processes. An optimal quantization of V consists in finding the subset $A \subset \mathcal{V}$ with $\text{card}(A) \leq l$ that minimizes

$$\left\| \min_{\mathbf{a} \in A} \|V - \mathbf{a}\| \right\|_{\mathbb{L}^2} = \left(\mathbb{E} \left[\min_{\mathbf{a} \in A} \|V - \mathbf{a}\|^2 \right] \right)^{1/2}. \quad (2)$$

Such a set is called an optimal l -quantizer. Let us denote as $A = \{\mathbf{a}_1, \dots, \mathbf{a}_l\}$. We define a neighbour projection associated to A as:

$$\pi_A := \sum_{i=1}^l \mathbf{a}_i \mathbf{1}_{C_{\mathbf{a}_i}(A)}, \quad (3)$$

where $\forall i \in \{1, \dots, l\}$ $C_{\mathbf{a}_i}(A)$ is the Voronoi partition induced by A and associated with \mathbf{a}_i :

$$C_{\mathbf{a}_i}(A) = \{\mathbf{v} \in \mathcal{V} \mid \forall h \in \{1, \dots, l\}, \|\mathbf{v} - \mathbf{a}_i\| \leq \|\mathbf{v} - \mathbf{a}_h\|\}. \quad (4)$$

The A -quantization of V is defined by:

$$\hat{V}_l := \pi_A(V). \quad (5)$$

The projection π_A transforms V into its nearest neighbor in the set A . Voronoi partition is optimal in the sense that, for any random variable $V'_l : \Omega \rightarrow A$, one has $\mathbb{E}\|V - V'_l\|^2 \geq \mathbb{E}\|V - \hat{V}_l\|^2$ (see, [28]). Finally the l -quantization error of V is defined by

$$e_l(V) = \inf \left\{ \left(\mathbb{E}\|V - \hat{V}_l\|^2 \right)^{1/2}, \right. \\ \left. \hat{V}_l : \Omega \rightarrow \mathcal{V}, \text{card}(\hat{V}_l(\Omega)) \leq l \right\}. \quad (6)$$

From a computational point of view, the cost of minimizing the error defined in Eq. (6) is not negligible. Even in the finite-dimensional space \mathbb{R}^m , the numerical search of an optimal solution leads to an increasing computational cost when l or m grows (see [19]). Luschgy, Pagès propose in [20] a greedy version of the \mathbb{L}^2 -quantization problem for U an \mathbb{R}^m valued random vector. The greedy-quantization is easier to compute in terms of complexity but provides a possible sub-optimal quantizer $\{\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_l\}$. The authors in [20] prove that the \mathbb{L}^2 -quantization error at level l induced by $\{\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_l\}$ goes to 0 at rate $l^{-1/m}$. The idea of such a procedure is to determine sequentially the sequence $(\hat{\mathbf{u}}_l)_{l \geq 1}$. The first vector $\hat{\mathbf{u}}_1$ achieves the error $e_1(U)$. Then, for $l \geq 2$,

$$\forall l \geq 2, \hat{\mathbf{u}}_l \in \arg\min_{\mathbf{u} \in \mathbb{R}^m} (\mathbb{E}\|U - \hat{U}_l\|^2)^{1/2}, \quad (7)$$

where \hat{U}_l is the l -quantization induced by $\{\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_{l-1}\} \cup \{\mathbf{u}\}$.

In the present work, we propose a sequential strategy in an infinite-dimensional setting under the assumption that the random process V may not be Gaussian. In this framework, Miranda and Bocchini [22, 23] propose a one-shot algorithm that produces an optimal functional quantizer but which depends on a simulation procedure for V . In the following, we propose a greedy algorithm to compute a l -quantization of V . In our framework, the functional random variable is only known through a finite set of realizations. The specificity of our procedure is first that it does not require a simulation algorithm of the unknown process V (which is known only from a finite set of realizations), and secondly our quantizer can be sequentially increased in a greedy fashion. One ingredient in our methodology is the PCA decomposition of V (also known as Karhunen-Loève expansion).

Let us briefly recall the Karhunen-Loève expansion which is the most commonly employed method to reduce the statistical complexity of random fields indexed over bounded intervals, with continuous covariance function.

The Karhunen-Loève expansion Let $V \in \mathcal{H}$ be a random process with zero mean and continuous covariance function $C(t, s)$. Then

$$V(t) = \sum_{i=1}^{\infty} u_i \psi_i(t), \quad t \in [0, T], \quad (8)$$

where $\{\psi_i\}_{i=1}^{\infty}$ are orthogonal and normalized eigenfunctions of the integral operator corresponding to C :

$$\lambda_i \psi_i(t) = \int_0^T C(t, s) \psi_i(s) ds. \quad (9)$$

The $\{u_i\}_{i=1}^{\infty}$ denotes a set of orthogonal random variables with zero mean and variance λ_i , where λ_i is the eigenvalue corresponding to the eigenfunction ψ_i . Equation (8) can be truncated after m -terms:

$$V(t) \simeq \sum_{i=1}^{m_{KL}} u_i \psi_i(t). \quad (10)$$

Computational details for functional PCA The covariance structure of the process V is unknown and has to be estimated from the data. More precisely, $C(s, t)$ is estimated from the centered sample $\Xi = \{\mathbf{v}_i\}_{i=1}^N$ by:

$$C^N(t, s) = \frac{1}{N} \sum_{i=1}^N \mathbf{v}_i(s) \mathbf{v}_i(t). \quad (11)$$

The eigenvalue problem defined by Eq. (9) is then solved by replacing C by C^N (see, e.g., [6] for convergence results). That approximated eigenvalue problem is solved, e.g., by discretizing the trajectories $\{\mathbf{v}_i\}_{i=1, \dots, N}$ on $[0, T] : \kappa =$

$\{v_i(t_j)\}_{i=1,\dots,N}^{j=1,\dots,N_T}$. It leads to the empirical covariance matrix defined as $\tilde{C}^N = \frac{1}{N} \mathbf{K}^\top \mathbf{K}$. We then have to solve a classical multivariate PCA with N_T variables given by a sample of size N . Standard PCA involves an $O(\min(N_T^3, N^3))$ search for directions of maximum variance. In the case of discretized curves $N_T \gg N$, thus the complexity of PCA is $O(N^3)$.

Other approaches to implement functional PCA can be found in the literature. In [31], e.g., the authors propose to expand the curves as linear combinations of spline basis functions, and to apply PCA to the coefficients of the curves on the spline basis. There are several criteria for the choice of the truncation argument m_{KL} [14]. One can cite the Kaiser-Guttman criterion which consists of choosing the first components with eigenvalues higher than 1. Instead of using the absolute value of the explained variance, as indicated by the eigenvalue, the choice of m_{KL} could be based on the "percentage of variance" given by the proportion $\frac{\lambda_i}{\sum_{i=1}^N \lambda_i}$. By this way we choose m_{KL} so that the percentage of variance explained by the first components exceeds a certain threshold. Often to avoid the arbitrary choice of the threshold, we display the eigenvalues in a downward curve and extract the components on the steep slope.

What is important to note is that functional PCA is done once for all, as an offline pre-processing step in our global inversion procedure.

\mathbb{L}^2 -Greedy Functional Quantization (\mathbb{L}^2 -GFQ) Now we aim at optimally exploring the range of variations of V with a few elements in Ξ . Let

$\mathcal{G} = \{(\langle \mathbf{v}, \psi_1 \rangle, \dots, \langle \mathbf{v}, \psi_{m_{KL}} \rangle)^\top, \mathbf{v} \in \Xi\} = \{\mathbf{u}_i\}_{i=1}^N$, with $\mathbf{u}_i = (\langle \mathbf{v}_i, \psi_1 \rangle, \dots, \langle \mathbf{v}_i, \psi_{m_{KL}} \rangle)^\top$, be the set of the first m coefficients in KL expansion (see Fig. 1). Since we place

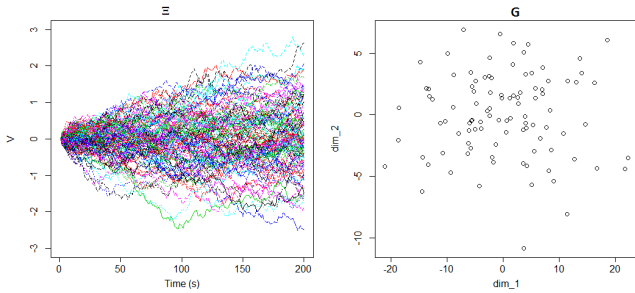


Fig. 1 Ξ is a sample of 100 realizations of V (left) and \mathcal{G} the corresponding representation in the truncated space of coefficients with $m_{KL} = 2$ (right).

ourselves in a finite space $\mathcal{G} \subset \mathbb{R}^{m_{KL}}$, a first step consists in an efficient and sequential strategy for the selection of a vectorial l -quantizer. A first solution is to apply the greedy vectorial quantization procedure described by Equation (7).

The sequential construction is detailed below

$$\begin{aligned} \hat{D}_1 &= \{\hat{\mathbf{u}}_1\} \text{ where } \hat{\mathbf{u}}_1 \text{ is a solution of } e_1(\mathbf{U}) \text{ from Eq. (6)} \\ \forall l \geq 2, \hat{D}_l &= \hat{D}_{l-1} \cup \{\hat{\mathbf{u}}_l\} \\ \text{where } \hat{\mathbf{u}}_l &\in \arg \min_{\mathbf{u} \in \mathcal{G}} (\mathbb{E} \|\mathbf{U} - \mathbf{U}_l\|^2)^{1/2}, \\ l &= l + 1, \end{aligned} \quad (12)$$

\mathbf{U} is a random vector with discrete uniform distribution on \mathcal{G} and \mathbf{U}_l is the l -quantization induced by $\{\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_{l-1}\} \cup \{\mathbf{u}\}$. A collection of representative curves associated to our functional random variable V is obtained by recovering the curves in the initial sample Ξ that correspond to the selected points \hat{D}_l :

$$\hat{D}_l = \{\hat{\mathbf{u}}_i\}_{i=1}^l \subset \mathcal{G} \longrightarrow \Theta'_l = \{\theta'_i\}_{i=1}^l \subset \Xi \quad (13)$$

This step is important as it allows to recover the functional variability of V and not only the variability of its first m_{KL} coefficients. Figure 2 shows the algorithm up to step $l = 10$ on the example of Figure 1. Note that, from this figure, we can see that \mathbb{L}^2 -quantization leads to selecting central points and so the sample $\Theta'_{l=10}$ is mainly representative of the mean behaviour of V . We also note that the extreme points are not chosen in the beginning of the construction. To address this issue, we propose a method that aims at exploring the range of variations of V by selecting central points and also extreme points. One way to do so consists in the construction of a space filling design in $\mathbb{R}^{m_{KL}}$. Let \tilde{D} be this design. As the distribution of V is only known through Ξ , it is natural to search \tilde{D} included in \mathcal{G} such that the chosen points are well spread all over the space. Let us recall the notion of space-filling design, from a purely model-free stance.

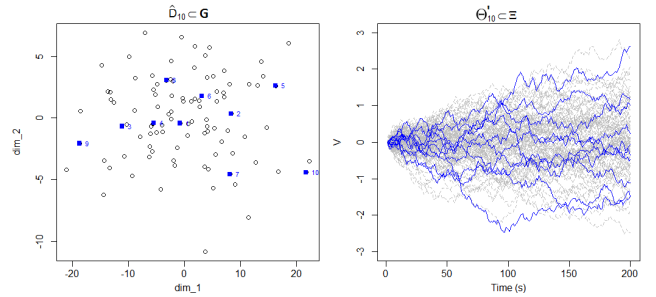


Fig. 2 Sequential design of 10 points (bold points) in the set of the coefficients \mathcal{G} (left) and their corresponding bold curves in Ξ (right).

Space-Filling Design and criterion Let us define

$D = \{\mathbf{u}_1, \dots, \mathbf{u}_l\}$ a collection of l points. We denote by $\text{dist}_{ij} = \|\mathbf{u}_i - \mathbf{u}_j\|$ the euclidean distance between two design points \mathbf{u}_i and \mathbf{u}_j of D . One must then attempt to make the smallest distance between neighboring points in D as

large as possible. We call a design that maximizes $\phi_{Mm}(D) = \min_{i \neq j} \text{dist}_{ij}$, a **maximin-distance design** (see Johnson et al [17]). There are several other intrinsic criteria in literature such as discrepancy that measures whether the distribution of the points of D is close to a uniform distribution. See Pronzato et al. [30] for a detailed overview on the subject. In the following, we consider the maximin-distance criterion to construct our design, and since we want to select points from the set of coefficients \mathcal{G} , the design \tilde{D} can be obtained by finding the design of l points among N , that maximizes the criterion ϕ_{Mm} .

Maximin-Greedy Functional Quantization (Maximin-GFQ)

Finding the design \tilde{D} is a computationally difficult task. We could adapt one of the optimal design algorithms used in the literature such as simulated annealing (see Morris et al. [24]) and stochastic evolutionary algorithm (see Jin et al. [16]) for our purpose. Here we propose a one-point-at-time greedy algorithm for the generation of our design. The sequential construction is described below

Initialization: $\tilde{D}_1 = \{\tilde{\mathbf{u}}_1\}$ where $\tilde{\mathbf{u}}_1$ is randomly chosen

$$\forall l \geq 2, \tilde{D}_l = \tilde{D}_{l-1} \cup \{\tilde{\mathbf{u}}_l\}$$

$$\text{where } \tilde{\mathbf{u}}_l \in \arg \max_{\mathbf{u} \in \mathcal{G}} \phi_{Mm}(\tilde{D}_{l-1} \cup \{\mathbf{u}\}),$$

$$l = l + 1.$$

(14)

The algorithm starts with a random point $\tilde{\mathbf{u}}_1$, the next point is chosen among the points in \mathcal{G} in order to maximize the maximin-distance criterion. Besides the sequentiality as for the \mathbb{L}^2 -GFQ method, the points are now chosen in order to explore the range of variations of V at each step using a distance criterion. Meaning that at each step, the exploration of the domain is reasonable. The technique to recover the curves remains the same:

$$\tilde{D}_l = \{\tilde{\mathbf{u}}_i\}_{i=1}^l \subset \mathcal{G} \longrightarrow \Theta_l'' = \{\theta_i''\}_{i=1}^l \subset \Xi \quad (15)$$

In order to make a comparison, we start the Maximin-GFQ method Eq. (15) with the same point as the \mathbb{L}^2 -GFQ method Eq. (13), i.e., $\tilde{\mathbf{u}}_1 = \hat{\mathbf{u}}_1$. Figure 3 shows the results up to step $l = 10$ of both procedures. One can observe that the greedy maximin method covers well the range of variation of V contrary to the \mathbb{L}^2 -GFQ method, which provides a well distributed points only on the first component. The \mathbb{L}^2 -GFQ seems to be more influenced by the higher order KL expansion. In the following, in order to improve the readability, we adopt the simplified notation Θ_l that refers to one of the two constructions Θ_l'' and Θ_l' . In this way and in the same spirit as before (see equations (3),(5)), we define Θ_l -quantization of the stochastic process V as

$$\hat{V}_l = \sum_{i=1}^l \theta_i \mathbf{1}_{C_{\theta_i}}(V), \quad (16)$$

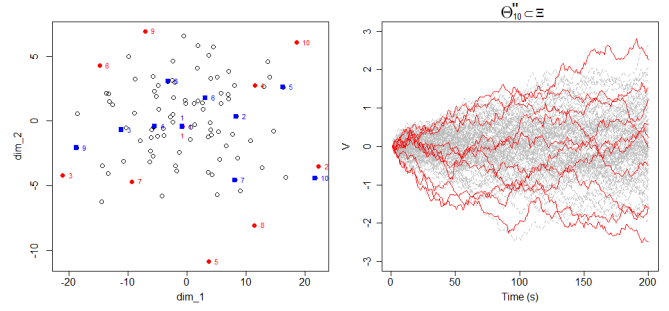


Fig. 3 Left: Two designs of $l = 10$ points in the 2D-coefficients set \mathcal{G} . Maximin-GFQ (red circle points) and greedy \mathbb{L}^2 -GFQ (blue square points). Right: The corresponding red curves for the Maximin-GFQ procedure (right).

where $\{C_{\theta_i}(\Theta_l) : \theta_i \in \Theta_l\}$ is a Voronoi partition of Ξ induced by Θ_l as defined in Equation (4).

Regarding the computational time devoted to perform the functional quantization, Figure 4 shows the time needed to compute the whole Greedy Functional Quantization as a function of point set size l . The KL expansion is done in a prior unaccounted step that takes 0.11 seconds. These algorithms are implemented in the software R and all computations are performed on a 8-core Intel 2.80 Ghz processor. From this figure, the running time of the maximin-GFQ method grows much more slowly than the \mathbb{L}^2 -GFQ one. Indeed, the discrete optimization in the \mathbb{L}^2 -GFQ method (Eq.(12)) involves an empirical estimation of the expectation at each step, thus the algorithm has complexity $O(N^2 \times l \times m_{KL})$. The latter becomes time consuming as l increases comparing to the maximin-GFQ algorithm which has complexity $O(N \times l \times m_{KL})$.

The reasons for performing a dimension reduction are twofold. The first one is computational as illustrated in Figure 4, where both methods are compared to the \mathbb{L}^2 -GFQ without KL decomposition with complexity $O(N^2 \times l \times N_T)$. The second one is related to the properties of the maximin criterion. Indeed, the space filling quality is higher in a small dimensional space since in a high one too many points are chosen at the boundary (see e.g. [1,30]).

We recall that this functional PCA step is done once for all in a pre-processing step.

In this section, we have introduced two data-driven greedy original procedures for functional quantization, quantization being an alternative to Monte Carlo methods for numerical integration. In the next section we highlight the performance of these procedures through two analytical examples.

3 Numerical integration

Let $h : \mathcal{V} = \mathbb{L}^2([0, T]) \longrightarrow \mathbb{R}$ be a continuous function, and let \hat{V}_l be a l -quantization. It is natural to approximate $\mathbb{E}[h(V)]$

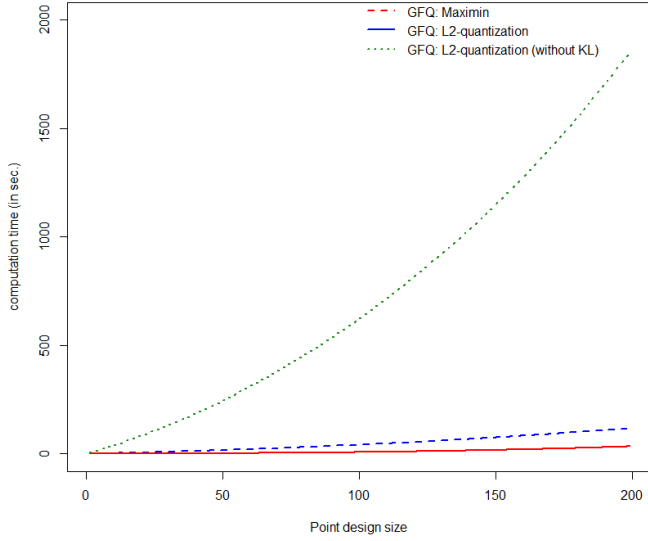


Fig. 4 Computation time (in seconds) of the GFQ methods as a function of point set size ($N = 200, N_T = 200$).

by $\mathbb{E}[h(\hat{V}_l)]$. This quantity $\mathbb{E}[h(\hat{V}_l)]$ is simply the finite weighted sum:

$$\mathbb{E}[h(\hat{V}_l)] = \sum_{i=1}^l h(\theta_i) \mathbb{P}(\hat{V}_l = \theta_i), \quad (17)$$

where the distribution $(\mathbb{P}(\hat{V}_l = \theta_i))_{i=1:l}$ of \hat{V}_l can be approximated empirically by $(\text{card}(C_{\theta_i}(\Theta_l) \cap \Xi) / \text{card}(\Xi))_{i=1:l}$. For a given i , this is the proportion of curves among N which are closer to θ_i than to any other θ_j , $j \neq i$. This proportion acts as a weight in the computation of the expectation. Assigning weights can bring a significant improvement (see L'Ecuyer et al. [18]).

Remark Under regularity assumptions, the integration error can be bounded by the quantization error. E.g., if h is Lipschitz in the sense that $\forall \mathbf{v}, \mathbf{v}' \in \mathcal{V}$, $|h(\mathbf{v}) - h(\mathbf{v}')| \leq c \|\mathbf{v} - \mathbf{v}'\|$, then

$$\begin{aligned} |\mathbb{E}[h(\mathbf{V})] - \mathbb{E}[h(\hat{V}_l)]| &\leq \mathbb{E}|h(\mathbf{V}) - h(\hat{V}_l)| \\ &\leq c \mathbb{E}\|\mathbf{V} - \hat{V}_l\| \\ &\leq c(\mathbb{E}\|\mathbf{V} - \hat{V}_l\|^2)^{1/2}. \end{aligned} \quad (18)$$

Returning to our original notation $\mathbb{E}[f(\mathbf{x}, \mathbf{V})]$, the proposed methodologies for an efficient estimation of the expectation over a functional random variable are summarized in Algorithms 1 and 2.

Algorithm 1 maximin-GFQ: Numerical integration

- 1: **Inputs:** initial sample (Ξ), truncation argument (m_{KL}), \mathbf{x} value where the expectation will be evaluated and set size of the quantization (l).
 - 2: $\mathcal{G} \leftarrow$ truncation of the KL expansion after m_{KL} -terms.
 - 3: Construct the set of representative points \hat{D}_l defined by Eq. (14).
 - 4: Induce from \hat{D}_l the corresponding set of representative curves Θ_l'' Eq. (15).
 - 5: Define $\hat{V}_l \leftarrow \Theta_l''$ -quantization of \mathbf{V} Eq. (16).
 - 6: Perform the computation of the expectation $\mathbb{E}[f(\mathbf{x}, \hat{V}_l)]$ Eq. (17):
 - 7: $\mathbb{E}[f(\mathbf{x}, \mathbf{V})] \approx \mathbb{E}[f(\mathbf{x}, \hat{V}_l)] = \sum_{i=1}^l f(\mathbf{x}, \theta_i'') \mathbb{P}(\hat{V}_l = \theta_i'')$.
-

Algorithm 2 \mathbb{L}^2 -GFQ: Numerical integration

- 1: **Inputs:** initial sample (Ξ), truncation argument (m_{KL}), \mathbf{x} value where the expectation will be evaluated and set size of the quantization (l).
 - 2: $\mathcal{G} \leftarrow$ truncation of the KL expansion after m_{KL} -terms.
 - 3: Construct the set of representative points \hat{D}_l defined by Eq. (12).
 - 4: Induce from \hat{D}_l the corresponding set of representative curves Θ_l' Eq. (13).
 - 5: Define $\hat{V}_l \leftarrow \Theta_l'$ -quantization of \mathbf{V} Eq. (16).
 - 6: Perform the computation of the expectation $\mathbb{E}[f(\mathbf{x}, \hat{V}_l)]$ Eq. (17):
 - 7: $\mathbb{E}[f(\mathbf{x}, \mathbf{V})] \approx \mathbb{E}[f(\mathbf{x}, \hat{V}_l)] = \sum_{i=1}^l f(\mathbf{x}, \theta_i') \mathbb{P}(\hat{V}_l = \theta_i')$.
-

In the sequel, we compare these two algorithms to the standard Monte Carlo, whose steps are outlined in Algorithm 3. The estimation of the expectation is sequentially calculated in the same vein as the GFQ procedures.

Algorithm 3 Crude MC: Numerical integration

- 1: **Inputs:** initial sample (Ξ), \mathbf{x} value where the expectation will be evaluated and set size of the quantization (l).
 - 2: Sample $\tilde{\mathbf{V}}_l = \{\tilde{\mathbf{v}}_i\}_{i=1}^l$ where $\tilde{\mathbf{v}}_1, \dots, \tilde{\mathbf{v}}_l \stackrel{i.i.d.}{\sim} \mathbb{U}_\Xi$, where \mathbb{U}_Ξ a discrete uniform distribution on Ξ .
 - 3: Estimate the expectation $\mathbb{E}[f(\mathbf{x}, \mathbf{V})]$ by l MC runs to $f(\mathbf{x}, \tilde{\mathbf{v}})$:
 - 4: $\mathbb{E}[f(\mathbf{x}, \mathbf{V})] \approx \frac{1}{l} \sum_{i=1}^l f(\mathbf{x}, \tilde{\mathbf{v}}_i)$.
-

We consider two analytical examples to highlight the integration performances of the two Greedy Functional Quantization methods (GFQ) in comparison with crude Monte Carlo. The first example is defined as an additive Lipschitz function, i.e., sum of the 2D Bohachevsky function and uncertainties. The second example does not verify the Lipschitz assumptions to mimic real applications.

Application 1 We consider a functional f defined as

$$f : (\mathbf{x}, \mathbf{v}) \mapsto (x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7) + \int_T e^{\mathbf{v}_t} dt,$$

where $\mathbf{x} = (x_1, x_2) = (50, -80)$ and \mathbf{V} is a standard Brownian motion on \mathbb{R} and $[0, T] = [0, 1]$. We suppose that a sample Ξ of $N = 200$ realizations of \mathbf{V} is available and that the

probability distribution of V is unknown. In this example, we fix the truncation argument at 2 to explain 90% of the variance. Because of the random choice of the starting curve, the maximin-GFQ methods have a stochastic behaviour like the Monte Carlo method. To account for these variabilities in the test, the performance is averaged over 200 independent runs for the MC method and 200 runs related to all the possibilities of the starting curve for the maximin-GFQ method.

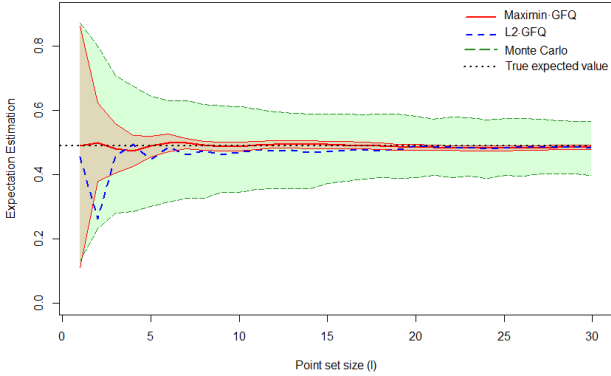


Fig. 5 Application 1. Sequential expectation estimation. Lines denote the average estimates, and colored bands mark the 25-th and 75-th quantiles (Monte Carlo (in green) and maximin-GFQ (in red)). Horizontal axis denotes the number of curves l used for the expectation estimation.

The results are shown in Figure 5. We observe that for any choice of l , the integration error induced by any of both GFQ methods is significantly smaller than the standard Monte Carlo. One can note that the maximin-GFQ method is less sensitive to the starting point from set size $l \geq 25$. We also remark that for a small size $l \leq 5$, the maximin-GFQ method is not yet stabilized implying more uncertainties in the estimation. From $l \geq 10$ (see Fig. 3), stability is reached thanks to the procedure of selection of different kind of curves (centred and extreme).

Application 2 We define a function f by mixing control variables and uncertainties. This function involves $\max(\mathbf{v})$ and $\min(\mathbf{v})$, so catching the variability of V becomes important. The function f is given by

$$f : (\mathbf{x}, \mathbf{v}) \mapsto \max_t \mathbf{v}_t \cdot |0.1 \cos(x_1 \max_t \mathbf{v}_t) \sin(x_2) \cdot (x_1 + x_2 \min_t \mathbf{v}_t)^2| \cdot \int_0^T (30 + \mathbf{v}_t)^{\frac{x_1 - x_2}{20}} dt,$$

where $\mathbf{x} = (x_1, x_2) = (2.95, 3.97)$ and V is a standard Brownian motion on \mathbb{R} and $[0, T] = [0, 1]$. To mimic real applications, we assume in the procedure that the probability distribution of V is unknown. We suppose that a sample Ξ of $N = 200$ realizations of V is available. We note that the two GFQ methods depend on the truncation argument m_{KL} . In

this example, we fix it at 2 to explain 90% of the variance (results are similar for other truncation argument and \mathbf{x} values, and are omitted for brevity).

Due to the stochastic nature of the Monte Carlo and the maximin-GFQ methods, the performance of the method is averaged over 200 independent runs for the MC method and 200 runs related to all the possibilities of the starting curves for the maximin-GFQ method. The results of the integration

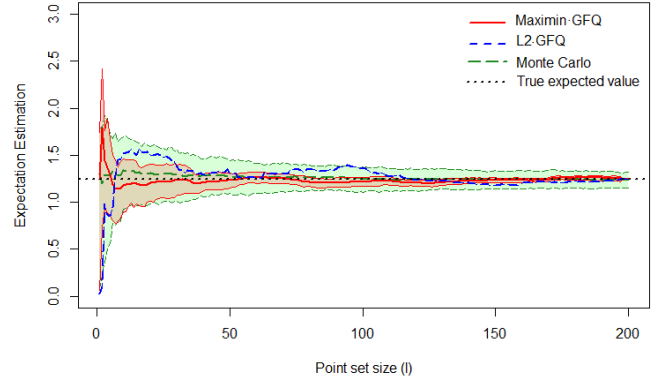


Fig. 6 Application 2. Sequential expectation estimation. Lines denote the average estimates, and colored bands mark the 25-th and 75-th quantiles (Monte Carlo (in green) and maximin-GFQ (in red)). Horizontal axis denotes the number of curves l used for the expectation estimation.

algorithms are shown in Figure 6. The lines indicate average estimate and the colored bands mark the area between the 25-th and 75-th quantiles. Here two observations can be made. First, for any choice of l , the integration error induced by both GFQ methods is significantly smaller than the standard Monte Carlo. Secondly, for maximin-GFQ method, the variability induced by the choice of the starting point is weak from set size $l \geq 20$.

We recall that our procedure is based on a dimension reduction. However, once the space filling design has been built in $\mathbb{R}^{m_{KL}}$, we go back to the infinite-dimensional space \mathcal{V} by selecting the corresponding curves. We expect from such a procedure to be robust to the dimension reduction. To illustrate this intuition, we compare the maximin-GFQ algorithm to an existing method, called hereafter Fpca method. This latter consists in sampling independently the KL random variables $U \in \mathbb{R}^{m_{KL}}$ whose probability distribution is estimated beforehand and denoted \mathbb{P}_U (see, e.g., [25] for a detailed overview on the subject). Then we obtain the desired curves using the linear combination of Equation (10). The results of the comparison are shown in Figure 7. We note that the Fpca method leads to a biased estimation due to the regularization induced by reducing the dimension. In summary, these simulations show that the two GFQ methods benefit improved performances over Monte Carlo in-

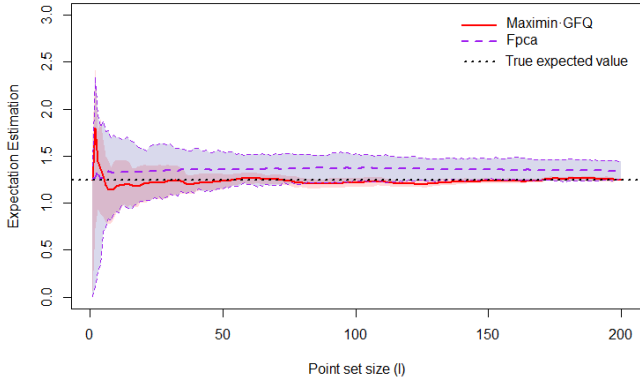


Fig. 7 Application 2. Sequential expectation estimation. Lines denote the average estimates, and coloured bands mark the 25-th and 75-th quantiles (Fpca (in blue) and maximin-GFQ (in red)). Horizontal axis denotes the number of curves l used for the expectation estimation.

merical accuracy.

In this section, we presented a methodology to efficiently estimate the expectation over V at a point \mathbf{x} in the control space \mathbb{X} . In the next section, we recall an existing method to address the inversion problem in the control variable space in the context of computationally costly simulations. This strategy is defined on a Bayesian framework and on the so-called Stepwise Uncertainty Reduction strategy (SUR). Let us start with some presentation of SUR paradigm.

4 Background on SUR strategies

Let $f : \mathbb{X} \times \mathcal{V} \rightarrow \mathbb{R}$ denote a real-valued continuous function, where \mathbb{X} is a bounded subset of \mathbb{R}^p , $p \geq 1$, and \mathcal{V} a functional space on which a random variable V is defined. Moreover, we suppose that a finite set of N independent and identically distributed realizations of the functional random variable V is available. In the following, we consider the expectation over the distribution of the functional random variable and we are interested in characterizing the set of control variables which leads to safe behaviour of a system:

$$\begin{aligned} \Gamma^* &:= \{\mathbf{x} \in \mathbb{X}, \mathbb{E}_V[f(\mathbf{x}, V)] \in C\} \\ &:= \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) \in C\} \text{ with } C = (\infty, c], c \in \mathbb{R}. \end{aligned} \quad (19)$$

While the function f depends on two separate types of inputs (control and uncertain variables), our objective function g depends only on the control variables, i.e., for each setting of control variables, the objective function is the mean of f over the unknown distribution of the uncertain variable.

The estimation of Γ^* by evaluating the function g at each grid point of the discretized version of \mathbb{X} requires far too many evaluations of g . Therefore, statistical methods based on a reduced number of evaluation points are widely used to overcome this latter difficulty by focusing the evaluations on the 'promising' subregion of the control space.

These methods usually begin by an exploration phase, during which the output of the code is computed on an experimental design of size n . This initial design is then sequentially expanded by adding new goal oriented points. These sequential strategies have been used in recent years for many purposes, such as the failure probability estimation ([3]) and target regions ([29]) whose main idea is to decrease the kriging variance at the points where the kriging mean is close to the threshold c . Unlike the two aforementioned methods, we are interested in the whole excursion set. In ([10],[11]) the sampling criterion is based on concept of random closed sets and applied to identify the set Γ^* . In this work we adopt this strategy and the procedure is introduced below.

4.1 Random closed set and bayesian framework

In a Bayesian framework, we assume that g is a realization of an almost surely continuous Gaussian process $Y \sim GP(m, k)$ with a mean structure m , defined as, $m(\mathbf{x}) = \mathbb{E}[Y_{\mathbf{x}}]$, $\mathbf{x} \in \mathbb{X}$, and a covariance kernel k , defined as, $k(\mathbf{x}, \mathbf{x}') := \text{Cov}(Y_{\mathbf{x}}, Y_{\mathbf{x}'}), \mathbf{x}, \mathbf{x}' \in \mathbb{X}$. Due to the stochastic nature of $(Y_{\mathbf{x}})_{\mathbf{x} \in \mathbb{X}}$, the associated excursion set,

$$\Gamma := \{\mathbf{x} \in \mathbb{X}, Y_{\mathbf{x}} \in C\} \quad (20)$$

is a random closed set. From the assumption that g is a realization of Y , the true unknown set Γ^* can be seen as a realization of the random closed set Γ . It is possible to compute a mean and deviation for this random set Γ by the Vorob'ev approach (see [10]). We use the Vorob'ev expectation Q_{α^*} as an estimator of the true excursion set and Vorob'ev deviation $\mathbb{E}[\mu(\Gamma \triangle Q_{\alpha^*})]$ to quantify the uncertainty of the random closed set Γ . Therefore, we implement a Stepwise Uncertainty Reduction strategy (SUR) that aims at reducing uncertainty on Γ by adding new evaluation points step by step as proposed by [10]. The principle of SUR strategies are also recalled in Subsection 4.2.

4.2 SUR strategies

The principle of stepwise uncertainty reduction (SUR) (see, e.g., [3]; [7]) is to define an uncertainty measure, depending on the objective to be fulfilled, and to sequentially choose the points that decrease most this uncertainty. In other words, the aim of the SUR strategy is to construct a sequence of evaluation locations in order to reduce the *expected* uncertainty on a quantity of interest.

Here, we work in the setting where g is a sample path of a random process Y . The uncertainty function for an estimate of Γ is defined as a function $\mathcal{H}^{\text{uncert}}$ that associates to any finite sequence of observations $(\mathcal{X}_n, \mathbf{g}_{\mathcal{X}_n})$ a real value representing the uncertainty on the estimation of Γ . When

n observations are available, we denote by $\mathcal{H}_n^{\text{uncert}}$ the uncertainty at step n . We assume that we have r evaluations left. The objective of the SUR strategy is to find r optimal locations $\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+r}$ such that the uncertainty $\mathcal{H}_{n+r}^{\text{uncert}}$ is as small as possible.

In what follows, we consider the Vorob'ev deviation as the uncertainty function, at step n ,

$$\mathcal{H}_n^{\text{uncert}} = \mathbb{E}[\mu(\Gamma \triangle Q_{n, \alpha_n^*}) \mid Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}], \quad (21)$$

where Q_{n, α_n^*} is the Vorob'ev expectation conditionally to the n available observations. One way of constructing the sequence $\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+r}$ is to choose at each step the point that gives the smallest uncertainty $\mathcal{H}_{n+1}^{\text{uncert}}$,

$$\mathcal{H}_{n+1}^{\text{uncert}}(\mathbf{x}) = \mathbb{E}[\mu(\Gamma \triangle Q_{n+1, \alpha_{n+1}^*}) \mid Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}, Y_{\mathbf{x}}] \quad (22)$$

We note that the future uncertainty $\mathcal{H}_{n+1}^{\text{uncert}}$ is function of $Y_{\mathbf{x}}$ given $Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}$. Therefore, at each step we choose the point that gives the smallest uncertainty in expectation, that is :

$$\begin{aligned} \mathbf{x}_{n+1} &\in \operatorname{argmin}_{\mathbf{x} \in \mathbb{X}} \mathbb{E}_{n, \mathbf{x}}[\mathcal{H}_{n+1}^{\text{uncert}}(\mathbf{x})] \\ &:= \operatorname{argmin}_{\mathbf{x} \in \mathbb{X}} \mathcal{J}_n(\mathbf{x}), \end{aligned} \quad (23)$$

where $\mathbb{E}_{n, \mathbf{x}}$ denotes the expectation with respect to $Y_{\mathbf{x}} \mid Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}$ (for detailed formula of $\mathcal{J}_n(\cdot)$ see [10]).

After having evaluated the function g at the optimal location \mathbf{x}_{n+1} , we update the parameters of the posterior mean and covariance, and we restart until the evaluation budget r is spent. Such strategy is called *one-step lookahead*, which means that we select the next evaluation point as if it were the last one. A comparison of such a strategy to the space filling strategy based on Sobol' sequences is given in [10]. The authors highlight the effectiveness of the SUR strategy through an analytical example. For more theoretical perspectives on the SUR strategies, see [2] and references therein.

4.3 SUR strategy adapted to noisy observations

In our context, we can not compute exact evaluations of the expectation over the probability distribution of \mathbf{V} . We propose in Section 2 sequential algorithms that efficiently approximate the value of $g(\mathbf{x})$ by $\mathbb{E}[f(\mathbf{x}, \hat{\mathbf{V}}_l)]$. On the design points, the n evaluations $\mathbf{g}(\mathcal{X}_n)$ are replaced by their approximation $\tilde{\mathbf{g}}(\mathcal{X}_n) = (\mathbb{E}[f(\mathbf{x}_1, \hat{\mathbf{V}}_l)], \dots, \mathbb{E}[f(\mathbf{x}_n, \hat{\mathbf{V}}_l)])$. For that reason, we do not want to build an exact interpolant at points $\mathbf{x}_1, \dots, \mathbf{x}_n$. We rather consider that $\tilde{\mathbf{g}}(\mathcal{X}_n)$ are realizations of a Gaussian vector $(\tilde{Y}_{\mathbf{x}_1}, \dots, \tilde{Y}_{\mathbf{x}_n})^T$ defined by $\tilde{Y}_{\mathbf{x}_i} := Y_{\mathbf{x}_i} + \varepsilon_i$ where $\varepsilon_1, \dots, \varepsilon_n$ are independent centred Gaussian variables of variance $\tau_1^2, \dots, \tau_n^2$. Conditionally to $\tilde{Y}_{\mathbf{x}_1}, \dots, \tilde{Y}_{\mathbf{x}_n}$ the process Y is still Gaussian except that we add the variances $\{\tau_i^2\}_{i=1}^n$ to the diagonal elements of the covariance matrix.

Remark In our context, we estimate the expectation empirically by l calls to the function f . The well-known Monte Carlo methods (Crude MC, FPCA) allow us to quantify the noise of estimation and to integrate it into GP modelling (kriging with noisy observations). Consequently, we define the variance components $\{\tau_i^2\}_{i=1}^n$ as $\tau_i^2 = \operatorname{var}(f(\mathbf{x}_i, \hat{\mathbf{V}}_l))$ for the crude MC method and $\operatorname{var}(f(\mathbf{x}_i, \mathbf{V}_l^{\text{pca}}))$ for the FPCA method. We also note that the two deterministic GFQ methods are in the same spirit than the Quasi-MC methods. The error of estimation depends on the variations of f that is most often not tractable. Therefore we assume a negligible integration error and thus the observations are assumed noise-free.

5 Algorithm coupling SUR and functional quantization

The whole computational aspect is carried out in the **R** environment : we use DiceKriging package [32] for Gaussian modelling and the sampling criterion \mathcal{J}_n Eq. (23), used in order to select the next evaluation \mathbf{x}_{n+1} of the function g , is already implemented in the KrigInv package [11]. We exploit the kriging update formulas [8] for faster updates of posterior mean and covariance. When \mathbf{x}_{n+1} is identified, l calls to the simulator have to be performed to approximate the expectation on that point. The sequentiality of our estimation method of the expectation on \mathbf{x}_{n+1} leads us to define a *stopping criterion* on the expectation estimation \hat{m} . Thus l is chosen sufficiently large so that l_0 consecutive 'expectation variations' are smaller than a threshold ε . Besides, the number of calls l will naturally depend on \mathbf{x}_{n+1} . In practice, at each step of the estimation we evaluate the absolute difference between two consecutive estimations of the expectation,

$$e_l(\mathbf{x}_{n+1}) = |\hat{m}_{l-1}(\mathbf{x}_{n+1}) - \hat{m}_l(\mathbf{x}_{n+1})|, \quad (24)$$

where $\hat{m}_l(\mathbf{x}_{n+1}) = \mathbb{E}[f(\mathbf{x}_{n+1}, \hat{\mathbf{V}}_l)]$, we denote by $|\cdot|$ the absolute value function. In the following, the *stopping criterion* is defined by the following relation,

$$\forall 0 \leq j \leq l_0, e_{l-j}(\mathbf{x}_{n+1}) \leq \varepsilon \quad (25)$$

It ensures that the quantities e_l are smaller than a prescribed tolerance ε on the l_0 previous steps in the estimation. These two parameters are set by practitioners. It allows to use fewer curves without losing estimation accuracy.

Remark The parameters (l_0, ε) are closely linked to the allocated budget. Moreover the parameter l_0 can be set in practice between [2, 5] regarding the stability we want to achieve with the method. About the parameter ε , it will be intuitively calibrated depending both on the precision and on the scale of the outputs.

The strategy SUR could be stopped if the allocated number

of simulations is reached. However, we define in this work an additional stopping criterion based on the Vorob'ev deviation and close to the one defined for the expectation estimate. Thus, the strategy is carried out until the following stopping criterion is verified

$$\forall 0 \leq j \leq l_0^{\text{SUR}}, e_{l-j}^{\text{SUR}}(\mathbf{x}_{n+1}) \leq \varepsilon^{\text{SUR}}, \quad (26)$$

where $e_i^{\text{SUR}} = |\mathbb{E}_{i-1}[\mu(\Gamma \triangle Q_{i-1, \alpha_{i-1}^*})] - \mathbb{E}_i[\mu(\Gamma \triangle Q_{i, \alpha_i^*})]|$ is the absolute error between two successive Vorob'ev deviations. The condition Eq. (26) tests if all the quantities are smaller than a tolerance ε^{SUR} on l_0^{SUR} consecutive steps. The global methodology to perform inversion in presence of functional uncertainty proposed in this paper is summarized in Algorithm 4.

Algorithm 4 Data-driven stochastic inversion under functional uncertainties

```

1: Create an initial design of experiments (DoE) of  $n$  points in the control space  $\mathbb{X}$ .
2: Alg  $\mathcal{A} \leftarrow$  choose one of the Algorithms 1,2 and 3.
3: Estimate the expectation (Alg.  $\mathcal{A}$ ).
4: Deduce  $\{\tau_i^2\}_{i=1}^n$  (if Crude MC or Fpca methods).
5: while Stopping criterion Eq. (26) not met (SUR) do
6:    $\mathbf{x}_{n+1} \leftarrow$  Sampling criterion  $\mathcal{J}_n$ .
7:   Set  $l = 1$ .
8:   while Stopping criterion Eq. (25) not met (Expectation Estimation) do
9:     Approximate the expectation by  $\mathbb{E}[f(\mathbf{x}_{n+1}, \hat{V}_l)]$  (Alg.  $\mathcal{A}$ ).
10:    Set  $l = l + 1$ .
11:   end while
12:    $\tau_{n+1}^2 \leftarrow \text{var}(f(\mathbf{x}_{n+1}, \hat{V}_l))$  (if Crude MC or Fpca methods).
13:   Update DoE.
14:   Set  $n = n + 1$ .
15: end while
16: end

```

Remark on stage 2 of Algorithm 4 Due to their sampling based nature, the Crude MC and Fpca methods are sensitive to the resulting estimation errors. Consequently, we consider the adaptation of the SUR strategy for noisy observations (see, Subsection 4.3).

6 Numerical tests

In this section we apply the proposed data-driven methodology for stochastic inversion under functional uncertainties to two test cases. On an analytical test case, we compare the methods combining the SUR strategy to the four approach to estimate the expectation. We then present in Subsection 6.2 an application to the industrial automotive test case which motivate our study.

6.1 Analytical example

In this example, we define the function f as follows:

$$f : (\mathbf{x}, \mathbf{v}) \mapsto \max_t \mathbf{v}_t \cdot |0.1 \cos(x_1 \max_t \mathbf{v}_t) \sin(x_2) \cdot (x_1 + x_2 \min_t \mathbf{v}_t)^2| \cdot \int_0^T (30 + \mathbf{v}_t)^{\frac{x_1 - x_2}{20}} dt, \quad (27)$$

where the control variable \mathbf{x} lies in $\mathbb{X} = [1.5, 5] \times [3.5, 5]$, and \mathbf{V} is a standard Brownian motion. We suppose that a sample of N realizations of \mathbf{V} is available, denoted by \mathcal{E} , and these realizations are discretized uniformly on 100 points. The objective is to construct the set $\Gamma^* := \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})] \leq c\}$, where $c = 1.2$.

Here we consider a Gaussian process prior $(Y_{\mathbf{x}})_{\mathbf{x} \in \mathbb{X}} \sim GP(m, k)$, with constant mean function and Matérn covariance kernel with $\nu = 5/2$. The initial DoE consists of a 9 points LHS design optimized by maximin criterion. The hyper-parameters of the Gaussian process Y are estimated by Maximum Likelihood Estimation (MLE). Figure 8 shows the initial design of experiments and the target set Γ^* obtained from a 30×30 grid experiment, where at each grid point the expectation is approximated by a Monte Carlo Method over 5000 realizations of \mathbf{V} . We aim at estimating the set Γ^* using the SUR strategy to choose the next evaluation point as defined in Section 4, and the methods presented in Section 2 to provide an efficient estimation of the expectation. We proceed

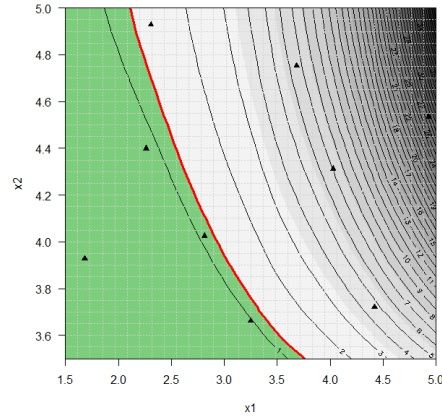


Fig. 8 Analytical example. Contour plot of the function g , the set of interest (green) with boundary (red line), the initial design of experiments (black triangles).

to add one point at each iteration of the SUR strategy until the condition Eq. (26) for $(l_0^{\text{SUR}}, \varepsilon^{\text{SUR}}) = (4, 5 \cdot 10^{-3})$ is reached. The covariance parameters are re-estimated at each step by MLE. Since this criterion is based on the Vorob'ev deviation, the objective is to reduce the uncertainty on the set estimate until stability. For the sequential estimation of the expectation, we test the sensitivity to the parameters (l_0, ε)

of criterion Eq. (25) (see Table 1).

The estimation of the expectation at the proposed point by SUR is carried out with one of the methods detailed in Section 2 (Fpca, crude MC, maximin-GFQ, \mathbb{L}^2 -GFQ). As presented in Section 5, the estimation is done sequentially and it depends on the *stopping criteria* parameters l_0 , ε and on the truncation argument m_{KL} . This latter is set at $m_{KL} = 7$ in order to explain 97% of the variance. The four expectation estimation methods are sequential as detailed in Section 2. Indeed, the two GFQ methods are sequential by definition. The crude MC method is sequential because at each step a curve is drawn with replacement from the available sample Ξ (see Algorithm 3). The same goes for the probabilistic approach (Fpca), at each step we add a new curve built as already explained.

The first test consists in fixing the available sample of realizations of V ($N=200$). For this fixed sample, we compare the obtained results for different l_0 and ε . Table 1 lists the parameters tested in this section.

l_0	4	2	3	4
ε	10^{-2}	5.10^{-3}	5.10^{-3}	5.10^{-3}

Table 1 Analytical example. Estimation of expectation *stopping criteria* parameters

To compare the performance of the various methods we use the ratio between the volume of the symmetric difference between the true set Γ^* and the estimated set at last iteration, $\mu(\Gamma^* \triangle Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}^*})$ and the volume of the true set, $\mu(\Gamma^*)$. As shown earlier in Figure 6, the maximin-GFQ method is not very sensitive to the starting point. Thus, in the following test, we consider the deterministic version of the maximin-GFQ method by fixing the starting point to the one of \mathbb{L}^2 -GFQ method.

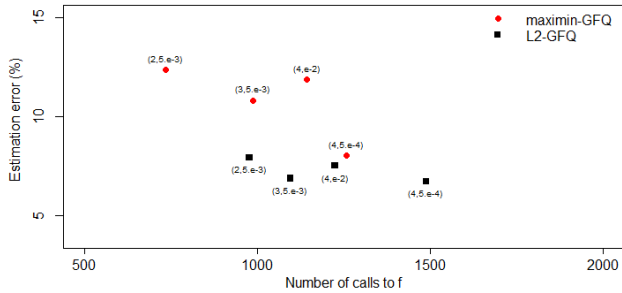


Fig. 9 Analytical example. The relative error obtained by the two GFQ methods for different values of l_0 and ε as a function of the number of calls to the function f .

	$\mu(\Gamma^* \triangle Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}^*}) / \mu(\Gamma^*)$	
(l_0, ε)	maximin-GFQ	\mathbb{L}^2 -GFQ
(4,1.e-2)	11.86 %	7.50 %
(2,5.e-3)	12.34 %	7.93 %
(3,5.e-3)	10.80 %	6.87 %
(4,5.e-3)	8.02 %	6.79 %

	Cumulative number of calls to f	
(l_0, ε)	maximin-GFQ	\mathbb{L}^2 -GFQ
(4,1.e-2)	1144 (21)	1225 (42)
(2,5.e-3)	735 (21)	978 (22)
(3,5.e-3)	989 (18)	1096 (21)
(4,5.e-3)	1259 (19)	1489 (26)

Table 2 Analytical example. (Top) The relative error obtained by the two GFQ methods for different values of l_0 and ε . (Bottom) The cumulative number of calls to the function f (in brackets are the number of iterations required to reach the stopping criterion in the SUR strategy).

From the comparison results displayed in Table 2 and plotted on Figure 9, we note that the two GFQ methods are sensitive to the parameters l_0 and ε . The \mathbb{L}^2 -GFQ method performs well in set estimation error terms, the maximin-GFQ provides better results in terms of cost. In the following comparison tests, we consider only the \mathbb{L}^2 -GFQ method as it gives much better set estimation error for a reasonable number of calls to the function f .

Regarding the second test, the two expectation estimation methods (Crude MC and Fpca) have a stochastic behaviour because of the sampling steps. To account for these variabilities, the performance of each method is averaged over 30 independent runs. The results are summarized in Tables 4 and 3. The results indicate that the three methods are sensitive to the parameters l_0 and ε : larger is the parameter l_0 , i.e., when seeking to a stability of the estimation, smaller is the error but higher is the number of calls to the function f . The \mathbb{L}^2 -GFQ method performs well in terms of error and cost. The cumulative number of calls to f has been decreased by a factor greater than 3 in comparison with the two other methods.

	$\mu(\Gamma^* \triangle Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}^*}) / \mu(\Gamma^*)$		
(l_0, ε)	Crude MC	Fpca	\mathbb{L}^2 -GFQ
(4,1.e-2)	9.53 % (4.12)	9.89 % (4.14)	7.50 %
(2,5.e-3)	9.84 % (3.24)	10.86 % (2.82)	7.93 %
(3,5.e-3)	9.54 % (3.81)	7.29 % (1.07)	6.87 %
(4,5.e-3)	8.98 % (2.62)	7.01 % (1.21)	6.79 %

Table 3 Analytical example. The average relative error obtained by the crude MC, Fpca for different values of l_0 and ε (in brackets are the standard deviation for the crude MC and Fpca methods).

	Cumulative number of calls to the function f		
(l_0, ε)	Crude MC	Fpca	\mathbb{L}^2 -GFQ
$(4, 1.e-2)$	2849 (27)	2805 (24)	1225 (42)
$(2, 5.e-3)$	2393 (26)	2670 (23)	978 (22)
$(3, 5.e-3)$	3537 (23)	3661 (20)	1096 (21)
$(4, 5.e-3)$	4400 (23)	4278 (20)	1489 (26)

Table 4 Analytical example. The average cumulative number of calls to the function f (written in brackets are the number of iterations required to reach the stopping criterion in the SUR strategy).

Figure 10 shows the set estimation error and the Vorob'ev deviation as a function of the iteration number for the three methods and $(l_0, \varepsilon) = (4, 5.e-3)$. For the crude MC and Fpca methods, the dotted lines indicate average error decay, and the coloured bands mark the area between the 25-th and 75-th error quantiles. Note that the three methods show a strong decrease in the set estimation error. The main observation that can be made is that, for a small total number of calls to f (see, Table 4), the convergence rate for the proposed approach (\mathbb{L}^2 -GFQ) is better in comparison to the Crude MC and Fpca methods.

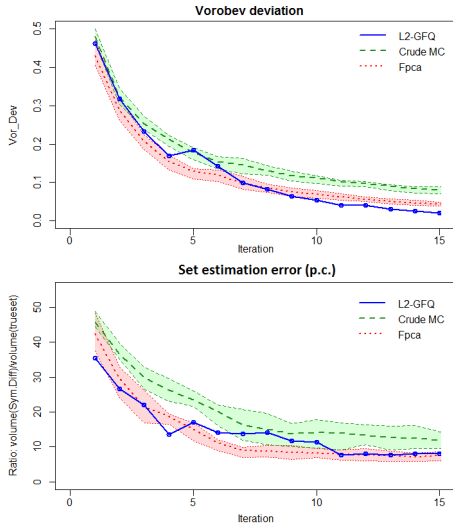


Fig. 10 Analytical example. Results for $(l_0, \varepsilon) = (4, 5.e-3)$. Lines denote the average, and coloured bands mark the 25-th and 75-th quantiles (Fpca (in red) and Crude MC (in green)). Top: The Vorob'ev deviation. Bottom: The set estimation error $\mu(\Gamma^* \triangle Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}^*}) / \mu(\Gamma^*)$.

In the following, the stopping criteria for SUR ($l_0^{\text{SUR}} = 4, \varepsilon^{\text{SUR}} = 5.10^{-3}$) and for the expectation estimation ($l_0 = 4, \varepsilon = 5.10^{-3}$) are chosen in order to offer a good compromise between the accuracy and the number of model evaluations.

Table 5 compares the sensitivity of the methods to the size of the available sample \mathcal{E} , denoted by N . In each cell of the table, we perform 20×20 independent runs. Indeed, for each value of N , we generate 20 training samples \mathcal{E} of size N and

for each sample we perform 20 runs for each method. The table summarizes the results averaged over the 400 runs.

	$\mu(\Gamma^* \triangle Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}^*}) / \mu(\Gamma^*)$		
	Crude MC	Fpca	\mathbb{L}^2 -GFQ
$N=50$	15.38 % (8.76)	13.25 % (5.09)	11.13 % (6.48)
$N=100$	9.60 % (4.65)	8.80 % (3.60)	9.08 % (5.41)
$N=200$	8.22 % (2.18)	7.71 % (2.35)	7.02 % (2.31)

	Cumulative number of calls to f		
	Crude MC	Fpca	\mathbb{L}^2 -GFQ
$N=50$	4281 (22)	4343 (21)	1044 (27)
$N=100$	4262 (22)	4313 (21)	1236 (25)
$N=200$	4152 (22)	4552 (21)	1262 (24)

Table 5 Analytical example. (Top) The average set estimation error obtained for different sample size and methods and $m_{KL} = 7$ (in brackets are the standard deviation). (Bottom) The average cumulative number of calls to the function f (in brackets are the number of iterations required to reach the stopping criterion in the SUR strategy).

We note that for a larger sample size, the recovering error is smaller. This can be explained by the fact that with a large sample size, the available information on variable V enables an effective estimation of the expectation. We know that the \mathbb{L}^2 -GFQ and the probabilistic modelling (Fpca) depend on the truncation argument. To better understand the effect of the number of dimensions m , we fix the stopping criteria for the SUR strategy and expectation estimation, and we consider different values of $m_{KL} = \{2, 3, 4, 5, 6\}$. Each cell of Table 6 represents the result averaged over 14×20 independent runs. For each m , we generate 14 samples \mathcal{E} of size $N=200$, and for each of them we perform 20 runs of each method.

	$\mu(\Gamma^* \triangle Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}^*}) / \mu(\Gamma^*)$	
	Fpca	\mathbb{L}^2 -GFQ
$m_{KL} = 2$	11.43 % (3.70)	8.90 % (3.71)
$m_{KL} = 3$	10.70 % (3.38)	7.72 % (3.38)
$m_{KL} = 4$	9.24 % (3.18)	7.40 % (3.13)
$m_{KL} = 5$	8.94 % (2.66)	7.05 % (5.09)
$m_{KL} = 6$	8.27 % (1.67)	6.96 % (3.32)

	Cumulative number of calls to f	
	Fpca	\mathbb{L}^2 -GFQ
$m_{KL} = 2$	3855 (18)	1286 (26)
$m_{KL} = 3$	4418 (24)	1139 (21)
$m_{KL} = 4$	4438 (21)	1236 (20)
$m_{KL} = 5$	4542 (21)	1214 (25)
$m_{KL} = 6$	4955 (19)	1142 (21)

Table 6 Analytical example. (left) The average set estimation error obtained by the Fpca and the \mathbb{L}^2 -GFQ methods for different values of m (in brackets are the standard deviation). (Right) The average cumulative number of calls to the function f (in brackets are the number of iterations required to reach the stopping criterion in the SUR strategy).

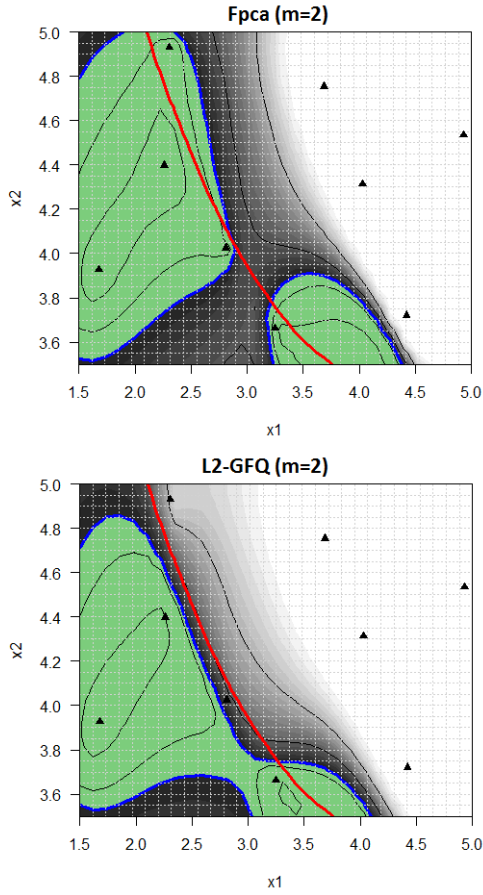


Fig. 11 Analytical example. Results based on the initial DoE in the case $m_{KL} = 2$. The coverage function, the boundary of the true set (red), the estimated sets (green).

Table 6 shows that for all values of m_{KL} , the \mathbb{L}^2 -GFQ method outperforms the probabilistic Fpca modelling. As shown in Table 7, for high truncation argument, the explained variance increases, that explains the decrease of the estimation error for the probabilistic modelling (Fpca). On the other hand, the \mathbb{L}^2 -GFQ accuracy seems to be almost constant for $m_{KL} \geq 3$. This can be explained by the fact that the KL expansion is only used to define a space filling design, and the information lost by the truncation is recovered by taking the corresponding curve in the set \mathcal{E} . On the contrary, the probabilistic modelling which is based on Fpca gives better results for higher m_{KL} . However the errors in Table 6 seem to be bounded below. To go below that bound, we probably need to increase the size of \mathcal{E} .

m_{KL}	2	3	4	5	6
Explained variance	90.2 %	93.4 %	95.1 %	96 %	96.7%

Table 7 Analytical example. The explained variance in function of m_{KL}

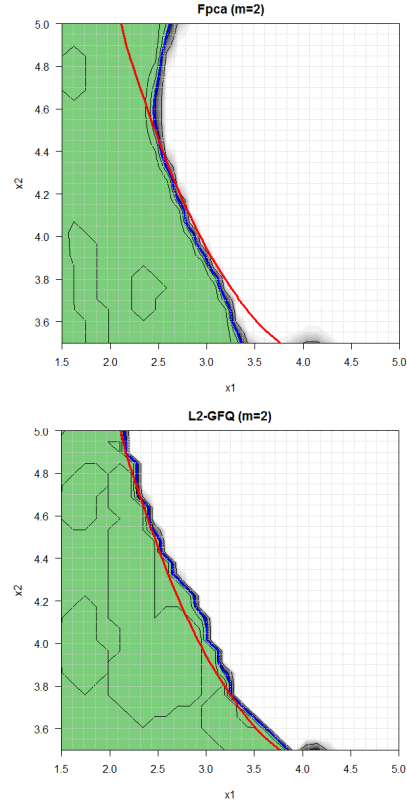


Fig. 12 Analytical example. Results at the last iteration in the case $m_{KL} = 2$. The coverage function, the boundary of the true set (red), the estimated sets (green).

6.2 IFPEN test case

In this section we test the proposed method on an automotive test case from IFPEN. The problem concerns an after-treatment device of diesel vehicles, called Selective Catalytic Reduction (SCR). This latter consists on a basic process of chemical reduction of nitrogen oxides (NO_x) to diatomic nitrogen (N_2) and water (H_2O) by the reaction of NO_x and ammonia (NH_3). The reaction itself occurs in the SCR catalyst. Ammonia is provided by a liquid-reductant agent injected upstream of the SCR catalyst. The amount of ammonia introduced into the reactor is a critical quantity: overdosing causes undesirable ammonia slip downstream of the catalyst, whereas under-dosing causes insufficient NO_x reduction. In practice, ammonia slip is restricted to a prescribed threshold.

We use an emission-oriented simulator developed by IFPEN, which models the vehicle, its engine and the exhaust after-treatment system. It takes the vehicle driving cycle profile as input and provides the time-series of corresponding exhaust emissions as output. A realistic SCR control law is used in this simulator. See [5] for an example of such a control law. In this study, we choose two control variables as input and a functional one considered as random. The con-

trol variables are parameters of the SCR control law. They set the targeted level of NH_3 storage in the catalyst and then are indirectly related to the NH_3 injected. They lie in $\mathbb{X} = [0, 0.6]^2$. The functional random variable describes the evolution of vehicle speed on I , with $I = [0, 5400\text{s}]$. These functional uncertainties come from an available sample of 100 real driving cycles. Regarding the discretization of the real-driving cycle, we have one observation per second, so $\forall i \in \{1, \dots, 100\}$, $\mathbf{v}_i \in \mathbb{R}^{5400}$. A subset of that sample is represented in Figure 13.

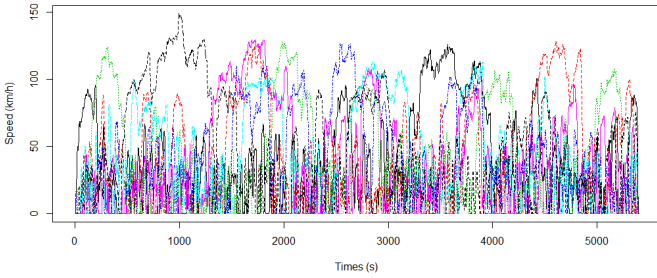


Fig. 13 Automotive test case. Sample of 7 real driving cycles.

In short, the ammonia emissions peak during a driving cycle is modelled as a function,

$$f : \mathbb{X} \times \mathcal{V} \rightarrow \mathbb{R} \\ (x, V) \mapsto f(x, V) = \max_{t \in I} \text{NH}_3^{\text{slip}}(t)$$

We are interested in recovering the set $\Gamma^* = \{x \in \mathbb{X}, g(x) = \mathbb{E}_V[f(x, V)] \leq c\}$, with $c = 30\text{ppm}$. Conducting this study on a full grid would consist in covering the space $[0, 0.6]^2$ with a fine mesh and evaluating the code 100 times at each point. Knowing that each simulation takes about two minutes, such study would require many computational hours, and thus the use of meta-models allows to tackle this computational issue.

Here we consider a Gaussian process prior $(Y_x)_{x \in \mathbb{X}} \sim GP(m, k)$, with constant mean function and Matérn covariance kernel with $\nu = 5/2$. The initial DoE consists of a 8 points LHS design optimized with respect to the maximin criterion. The covariance kernel hyper-parameters are estimated by maximizing the likelihood.

As for the analytical example, we proceed to add one point at each iteration for the SUR strategy until the stopping criterion with $(l_0^{\text{SUR}}, \epsilon^{\text{SUR}}) = (4, 5 \cdot 10^{-3})$ is verified. Concerning the expectation estimation, we set the stopping criterion parameters at $(l_0, \epsilon) = (4, 10^{-2})$ and the truncation argument is set at $m_{KL} = 20$ in order to explain 80% of the variance. The algorithm was stopped at the 62-point design because the Vorob'ev deviation appears as stabilized, in other words, the absolute error between the Vorob'ev deviations of the points 58-62 are smaller than 0.005, as shown in Figure 14. We note that for each additional point, the new observed response affects the estimation of the excursion set and its un-

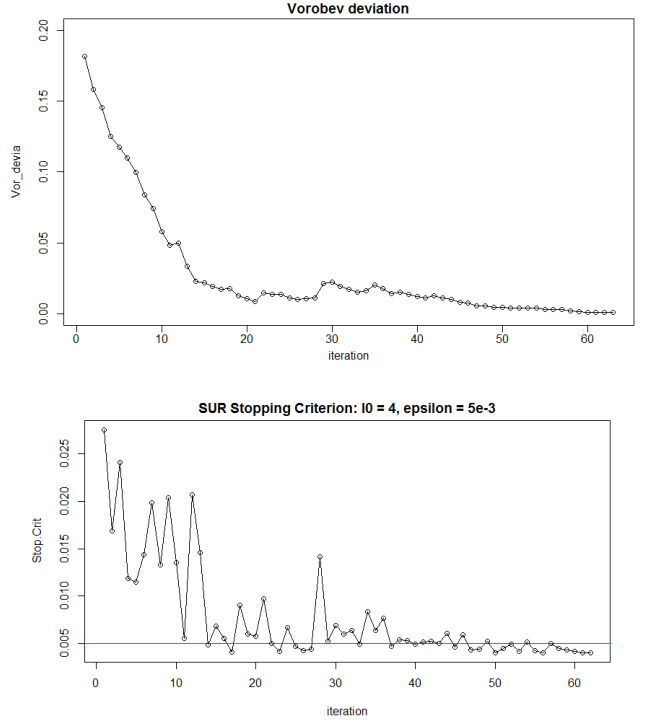


Fig. 14 Automotive test case. Top : Decrease of the Vorob'ev deviation at each iteration when new points are added. Bottom : Evolution of the absolute error Eq. (26) and the red line represents the stopping criterion.

certainty. Thus, although the Vorob'ev deviation generally decreases, it is not a monotonic decreasing. The stopping criterion is constructed to check the stability of convergence by taking into account the last four iterations. In searching for the true set, the SUR algorithm heavily visits the boundary region of Γ^* and explore also potentially interesting regions (cf. Fig. 15). In each added point, Figure 15 shows the number of necessary driving cycles to estimate the expectation. We remark that instead of taking the whole sample (100 driving cycles), it was sufficient to sequentially and wisely choose a reduced and representative number of driving cycles below 35. In the present case, the excursion domain Γ^* is well recovered by the algorithm. Actually, after 62 iterations (1575 evaluations) the whole domain \mathbb{X} has an excursion probability close to either 0 or 1.

7 Conclusions

In this paper, a new method of inversion under uncertainty was proposed for problems where some of the input parameters are functional random variables with unknown probability distribution (only a sample is available). The objective is to recover the set of control variables leading to robustly ensure some constraints by taking into account the uncertainties. The method is composed of two steps : a sequential strategy to estimate the excursion set, and the modelling

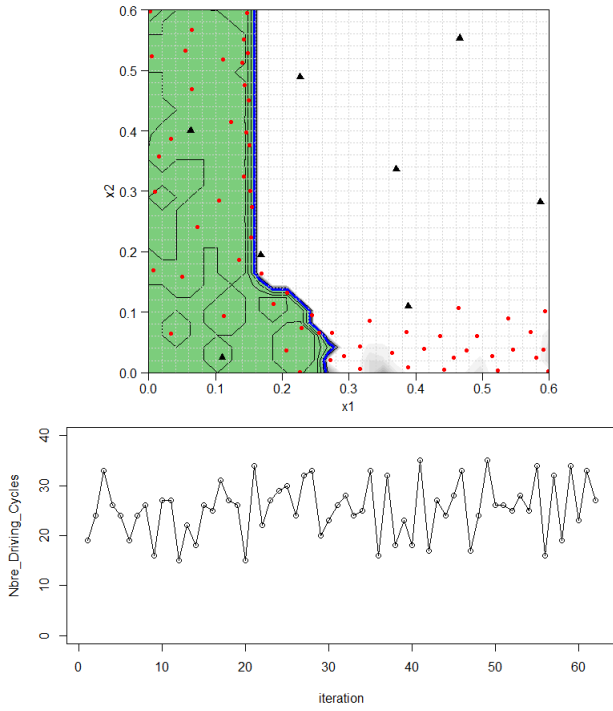


Fig. 15 Automotive test case. Left : Coverage probability function (grey scale), estimate set (green) after 62 added points and 1575 function evaluations, initial DoE (black triangles), the sequentially added points (red circles). Right : number of driving cycles used to estimate expectation at each added point.

of functional uncertainties. To solve the first issue a kriging model in the control input space is built. It makes possible to assess the uncertainty on the set of interest given a sample of evaluations. Then a sequential strategy (SUR) proposed by [10] and based on the kriging model is used to sequentially and efficiently choose new evaluation points to improve the excursion set estimation. For the second issue, we consider the expectation to model uncertainties and we propose two sequential approaches to estimate the expectation at each point proposed by SUR. Each curve is represented by its coefficients in a truncated KL decomposition. The chosen points in the KL coefficients finite set, each one corresponding to a curve, are sequentially added and chosen either to approximate a maximin space filling design or to reduce the quantization error. This methodology leads to an efficient estimation of the expectation, as illustrated on the application on an analytical test case with two control inputs and a functional random one. The results illustrate significant enhancement in term of precision and number of calls to the simulator. We also applied this method to the automotive test case which motivated this research work. The obtained result agrees with the intuitions made from physics behind the simulator. The paper focuses on the mean of $f(x, V)$ and here we choose to construct a GP model for the unobservable integrated response g . In the optimization context and for discrete and continuous random variables,

existing works deal with the case of unobservable response (see [34],[15]). The authors propose to build a GP model for the simulator f and induce a new GP model by integrating the previous one over the distribution of the uncertain variables. The adaptation of these works in the context of inversion and functional random variables is an on-going work. Other functionals of the output distribution may also be of great importance. For example, practitioners may be interested in ensuring a certain level of reliability, leading to consider a probabilistic constraint. The proposed method could be adapted to that case by seeing the probability as an expectation, at least for moderate risk levels.

Acknowledgments

The authors would like to thank the anonymous reviewers and the associate editor for their helpful comments which substantially improve this paper. We also thank the Inria Associate Team UNcertainty QUantification is ESential for Oceanic & Atmospheric flows PROBLEms. This work was supported by IFPEN and the OQUAIDO chair.

References

1. Abtini, M.: Plans prédictifs à taille fixe et séquentiels pour le krigage. Ph.D. thesis, Ecole Centrale Lyon (2018)
2. Bect, J., Bachoc, F., Ginsbourger, D.: A supermartingale approach to gaussian process based sequential design of experiments. arXiv preprint arXiv:1608.01118 (2016)
3. Bect, J., Ginsbourger, D., Li, L., Picheny, V., Vazquez, E.: Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing* **22**(3), 773–793 (2012)
4. Bolin, D., Lindgren, F.: Excursion and contour uncertainty regions for latent gaussian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **77**(1), 85–106 (2015)
5. Bonfils, A., Creff, Y., Lepreux, O., Petit, N.: Closed-loop control of a scr system using a nox sensor cross-sensitive to nh3. *IFAC Proceedings Volumes* **45**(15), 738–743 (2012)
6. Cardot, H., Ferraty, F., Sarda, P.: Functional linear model. *Statistics & Probability Letters* **45**(1), 11–22 (1999)
7. Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., Richet, Y.: Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics* **56**(4), 455–465 (2014)
8. Chevalier, C., Emery, X., Ginsbourger, D.: Fast update of conditional simulation ensembles. *Mathematical Geosciences* **47**(7), 771–789 (2015)
9. Chevalier, C., Ginsbourger, D.: Fast computation of the multi-points expected improvement with applications in batch selection. In: *International Conference on Learning and Intelligent Optimization*, pp. 59–69. Springer (2013)
10. Chevalier, C., Ginsbourger, D., Bect, J., Molchanov, I.: Estimating and quantifying uncertainties on level sets using the vorobev expectation and deviation with gaussian process models. In: *mODa 10—Advances in Model-Oriented Design and Analysis*, pp. 35–43. Springer (2013)
11. Chevalier, C., Picheny, V., Ginsbourger, D.: Kriginv: An efficient and user-friendly implementation of batch-sequential inversion strategies based on kriging. *Computational statistics & data analysis* **71**, 1021–1034 (2014)

12. Flury, B.A.: Principal points. *Biometrika* **77**(1), 33–41 (1990)
13. French, J.P., Sain, S.R., et al.: Spatio-temporal exceedance locations and confidence regions. *The Annals of Applied Statistics* **7**(3), 1421–1449 (2013)
14. Jackson, D.A.: Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches. *Ecology* **74**(8), 2204–2214 (1993)
15. Janusevskis, J., Le Riche, R.: Simultaneous kriging-based estimation and optimization of mean response. *Journal of Global Optimization* **55**(2), 313–336 (2013)
16. Jin, R., Chen, W., Sudjianto, A.: An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference* **134**(1), 268–287 (2005)
17. Johnson, M.E., Moore, L.M., Ylvisaker, D.: Minimax and maximin distance designs. *Journal of statistical planning and inference* **26**(2), 131–148 (1990)
18. LEcuyer, P., Lemieux, C.: Recent advances in randomized quasi-monte carlo methods. In: *Modeling uncertainty*, pp. 419–474. Springer (2005)
19. Levvard, C.: High-dimensional vector quantization: convergence rates and variable selection. Ph.D. thesis, Universite de Paris 11 (2014)
20. Luschgy, H., Pagès, G.: Greedy vector quantization. *Journal of Approximation Theory* **198**, 111–131 (2015)
21. Luschgy, H., Pagès, G., Wilbertz, B.: Asymptotically optimal quantization schemes for gaussian processes on hilbert spaces. *ESAIM: Probability and Statistics* **14**, 93–116 (2010)
22. Miranda, M., Bocchini, P.: Functional quantization of stationary gaussian and non-gaussian random processes. *Safety, Reliability, Risk and Life-Cycle Performance of Structures and Infrastructures* pp. 2785–2792 (2013)
23. Miranda, M.J., Bocchini, P.: A versatile technique for the optimal approximation of random processes by functional quantization. *Applied Mathematics and Computation* **271**, 935–958 (2015)
24. Morris, M.D., Mitchell, T.J.: Exploratory designs for computational experiments. *Journal of statistical planning and inference* **43**(3), 381–402 (1995)
25. Nanty, S., Helbert, C., Marrel, A., Pérot, N., Prieur, C.: Sampling, metamodeling, and sensitivity analysis of numerical simulators with functional stochastic inputs. *SIAM/ASA Journal on Uncertainty Quantification* **4**(1), 636–659 (2016)
26. Pagès, G.: Introduction to optimal vector quantization and its applications for numerics (2014)
27. Pagès, G., Printems, J.: Functional quantization for numerics with an application to option pricing. *Monte Carlo Methods and Applications mcma* **11**(4), 407–446 (2005)
28. Pages, G., Printems, J.: Optimal quantization for finance: from random vectors to stochastic processes. In: *Handbook of Numerical Analysis*, vol. 15, pp. 595–648. Elsevier (2009)
29. Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R.T., Kim, N.H.: Adaptive designs of experiments for accurate approximation of a target region. *Journal of Mechanical Design* **132**(7), 071008 (2010)
30. Pronzato, L., Müller, W.G.: Design of computer experiments: space filling and beyond. *Statistics and Computing* **22**(3), 681–701 (2012)
31. Ramsay, J.O.: *Functional data analysis*. Wiley Online Library (2006)
32. Roustant, O., Ginsbourger, D., Deville, Y.: Dicekriging, diceoptim: Two r packages for the analysis of computer experiments by kriging-based metamodeling and optimization (2012)
33. Vazquez, E., Bect, J.: A sequential bayesian algorithm to estimate a probability of failure. *IFAC Proceedings Volumes* **42**(10), 546–550 (2009)
34. Williams, B.J., Santner, T.J., Notz, W.I.: Sequential design of computer experiments to minimize integrated response functions. *Statistica Sinica* pp. 1133–1152 (2000)