Simultaneous kriging-based sampling for optimization and uncertainty propagation

Janis Janusevskis, Rodolphe Le Riche

Simultaneous kriging-based sampling for optimization and uncertainty propagation

Deliverable no. 2.2.2-1 of the ANR / OMD2 project

Janis Janusevskis[1] and Rodolphe Le Riche[1,2]
[1] LSTI / CROCUS team
Ecole Nationale Supérieure des Mines de Saint-Etienne
Saint-Etienne, France
[2] CNRS UMR 5146
{janusevskis,leriche}@emse.fr

July 29, 2010

Abstract

Robust analysis and optimization is typically based on repeated calls to a deterministic simulator that aim at propagating uncertainties and finding optimal design variables. Without loss of generality a double set of simulation parameters can be assumed: x are deterministic optimization variables, u are random parameters of known probability density function and $f(x,u)$ is the objective function attached to the simulator. Most robust optimization methods involve two imbricated tasks, the u's uncertainty propagation (e.g., Monte Carlo simulations, reliability index calculation) which is recurcively performed inside optimization iterations on the x's. In practice, f is often calculated through a computationally expensive software. This makes the computational cost one of the principal obstacle to optimization in the presence of uncertainties.

This report proposes a new efficient method for minimizing the mean objective function, $\min_{x\in\mathbf{R}^n} \mathrm{E}_U f(x,U)$. The efficiency stems from the simultaneous sampling of f for uncertainty propagation and optimization, i.e., the hierarchical imbrication is avoided. $Y^t_{(x,u)}(\omega)$, a kriging (Gaussian process conditioned on t past calculations of f) model of $f(x,u)$ is built and the mean process, $Z^t_{(x)}(\omega) = \mathrm{E}_U[Y^t_{(x,u)}(\omega)]$, is analytically derived from it. The sampling criterion that yields both x and u is the one-step ahead minimum variance of the mean process Z at the maximizer of the expected improvement. The method is compared with Monte Carlo and kriging-based approaches on analytical test functions in two, four and six dimensions.

1 Introduction

Modern design analysis and optimization tasks are based on deterministic simulation models. In cases where these models involve time-consuming evaluations the methods based on Response Surface models (metamodels, surrogate models) are often used. A popular method for creating such metamodels is kriging or Gaussian Process (GP) regression [37], [35], which provides probabilistic estimates of the underlying model. These estimates can be used in optimization procedures to formulate the so called Infill Sampling Criteria (ISC) [40] that allow efficiently sample for points.

However, real life problems are almost always non-deterministic. The procedures of robust design attempt to incorporate and account for some of the uncertainties and find designs whose quality degrades in a controlled, gradual way due to perturbations in parameters.

Robust design optimization is of interest:

* to account for model errors. Conventional deterministic optimization depends on the objective function and the constraints. However, these functions are approximations of the real world. As long as one does not have a detailed knowledge of the error function of the model, one cannot be certain that the model optimum can be mapped to the true optimum [5].
* to account for manufacturing uncertainties or errors that are introduced during the manufacturing process.
* to account for uncertainties due to dynamic environments or noisy measurements.
* to account for uncertainties in the future and perform risk analysis [45].
* in the field of Multidisciplinary Design Optimization, where the disciplines are optimized without full information about the outputs of other disciplines (which are inputs of the subsystem to be optimized).

Extensive survey on robust optimization can be found in [32], [5]. Different formulations for robust design optimization may be found in [5], [22], [39], [9], [42], [38]. Often the robust design optimization is expressed as the simultaneous minimizer of the mean and variation of the objective function while satisfying probabilistic constraints. As stated in [5] "modeling robustness aspects is only one side of the coin, solving the related optimization problems is the other often computationally demanding side, and only a few computational tools have been developed so far" [2].
In the simplified strategies the optimization problem is reformulated so as to be solved using standard techniques of mathematical programming. These techniques of robust optimization are restricted to cases where the original problem can be well approximated by linear or quadratic functions [5].
Keeping aside special engineering problems where the robustness measures can be obtained explicitly and thus leading to a classical (deterministic) optimization problem, usually some sort of estimates of the robustness measures are used [9], [2]. The simple approach involves Monte Carlo (MC) simulations, thus leading to optimization of *noisy functions*. Given a fixed design point x, one estimates the statistics (mean value, variance, etc.) and use them as inputs for a (derivative free) numerical optimization algorithm. The main drawback of such approach is the number of points, k, necessary to obtain accurate MC estimates. Therefore a trade off must be made between the computational cost (in terms of number of runs k) and the precision (fidelity level) of the solution. To reduce the cost of repeated MC evaluations during the optimization, the number of runs k may be determined, using, for example, a statistical test [8], [39]. A detailed presentation of previous works where the noisy optimization is based on kriging metamodels is given in section 1.1.
Different approaches exist that try to break the nested relationship between MC sampling and optimization loops. When using MC simulations to obtain robustness measures, the metamodels are used in order to reduce the time associated with sampling and also to smooth and filter the noise of the obtained statistics. The surrogate models may be utilized at the level of the optimization, of the uncertainty quantification or at both levels. A review and different formulations of surrogate based approaches and case studies involving trust regions (in order to manage the accuracy of the approximations) can be found in [28].
Evolutionary Algorithms (EA) [10] are intrinsically robust to the noise in the objective function. A literature review addressing uncertainties in evolutionary optimization is given in [23]. The simplest approach is based on multiple evaluations of the objective function at every population member and the mean value is utilized as fitness (direct MC implementation). In other approaches the noisy value is used directly as an input for optimization algorithm, see for example [3], [4]. As stated in [5] there are reasons to believe that EA are especially good at direct robust optimization (i.e., without calculating robust measures explicitly). However both approaches increase the number of function evaluations per generation by a factor between three and 100 [17].
We consider problems where the true function is given by a complex deterministic simulator $f(x, u)$ such as a CFD solver, which gives a constant value when x and u are fixed. In the context of robust optimization we will look at u as an instance of random variables U, therefore function $f(x, U)$ becomes random itself. In such cases where the optimization is based on computation intensive simulations (even for deterministic optimization problems) stochastic search algorithms are not suitable. Assuming that gradient information and analytical expressions of the robustness measures are not available, we focus on the methods that are more suited for such problems and in particular, kriging based optimization algorithms. A new methods is proposed that simultaneously minimizes the expectation function and optimally samples the random variables.

1.1 A bibliography of kriging based optimization of noisy simulators

Kriging based optimization procedures have become popular for the optimization of time consuming simulators. The first algorithm was introduced by by Schonlau, Welch and Jones [25], [24] and named Efficient Global Optimization (EGO). EGO uses an ISC called Expected Improvement (EI) to provide the next point for evaluation by the deterministic simulator [40]. In recent years modifications to this approach have been proposed thus adapting it for the optimization of nosy functions and partially converged simulations.
Forrester and others [11] have used kriging optimization for the multi-fidelity models based on CFD simulations. The overall optimization is divided into two steps. In the first step the initial Design of Experiments (DOE) is run using low fidelity calculations, the metamodels are created and points are added until a criterion of approximation quality is satisfied. In the second step, additional high fidelity (fully converged) points are evaluated, using the error of approximation of the low and high fidelity solutions, the EI based optimization is performed.
In [20], Huang and others extend the EGO algorithm for stochastic black box optimization tasks (Sequential Kriging Optimization). The next point in the optimization is chosen by maximizing the Augmented Expected Improvement (AEI), where the kriging is used to filter the noise at the current best point. The EI is augmented by an utility function, which accounts for the uncertainty of the model, and multiplied by a factor, which allows to account for the diminishing improvement of additional replicates as the prediction becomes more accurate. The algorithm is compared with alternative methods (Revised Simplex Search –RSS–, Simultaneous Perturbation Stochastic Approximation –SPSA–, DIRECT).
In [19], the authors use multiple metamodels to describe the systemic error of the corresponding fidelity level (for modelling multiple fidelity models, see [26]). Authors extend the AEI function by connecting it to the evaluation

costs. The extended AEI is then employed to select both the location and the fidelity level for the next Finite Element (FE) simulation.
Vazquez et al. [43] empirically compared several EI criteria (using best observations, minimum of kriging mean, AEI and Informational Approach to Global Optimization [44]) for the optimization of noisy functions. The tests were generated by GP realizations. The kriging is used to filter white noise.
In [33], a kriging based method accounts for uncertainty (fidelity level) at every data point, filters the noise and selects next data point. The approach is based on modified EI that describes quantiles. The next point is chosen by maximizing the modified EI. The proposed improvement function is also used to estimate the needed fidelity level. Simulator time slots are added iteratively to increase fidelity level and decrease noise at the next point. The benefit of the new time slot is measured by the EI decrease. The algorithm takes into account limited resources.
In [27], a multi-step lookahead optimization procedure is proposed using Bayesian Monte Carlo integration to obtain marginalized hyper-parameters of the process model [30]. The Bayesian expected loss criterion is introduced which replaces EI. The next point minimizes the expected loss. The proposed method is extended to the noisy case where the objective is corrupted by Gaussian noise. The one-step and two-steps ahead techniques are compared for nosy and non noisy test beds with other methods, such as SNOBFIT [21] and Implicit Filtering (IF) for noisy functions. Conditioning problems are avoided by requiring that observations are sufficiently far from each other. When one observation is very close to an existing one, instead of directly adding the point, the authors add derivative information on the line that connects the two points.

Another approach has been introduced in the Machine Learning community, where the GP is constructed stimulating linear filters with white noise. Such procedure is equivalent to constructing GP using kernel convolution with white noise. It allows to obtain covariance kernels without directly specifying and parametrizing new covariance kernels. For example, it can model multiple outputs and their dependency, and be applied for models of different fidelity levels [7, 6].

Previously discussed methods deal with optimization problem of the form

$$\min_x f(x) + \epsilon$$

$$x \in R^m$$

where ϵ introduces some noise. More generally, robust optimization addresses the problem of

$$\min_x f(x, u),$$

where the source of the noise in the objective is due to uncertainties in some of the model variables and/or parameters u.
Note that only some of the most sophisticated approaches (e.g. [20] and [33]) provide means for choosing the precision (number of MC runs) of the simulations. And most of the methods dealing with noisy functions assume normally distributed and independent noise sources. In this report, a method is presented that solves the robust optimization problem $\min_x \mathbb{E}_U[f(x, U)]$. A kriging model of the $f(x, u)$ is created and analytically translated into a kriging model of $E_U[f(x, U)]$, called the projected model. An EI and a sampling criteria define next (x, u) points for simulations. An important feature of the method is therefore that the optimization and the sampling are handled simultaneously.

1.2 GP for noisy inputs and convolutions

There is a limited amount of papers that are devoted to building GP while accounting for uncertain inputs.
In the work of Girard [15] the uncertainty on the inputs is propagated to the predictive (conditional) mean and covariance from a Bayesian perspective. The predictive mean and variance at uncertain points and also the predictive mean and covariance of conditional Gaussian Process for uncertain inputs are obtained in the case of Gaussian kernels and normally and independently distributed noise on the input parameters. In the case of non Gaussian kernels the author suggest to use Taylor series approximations. The results are applied for multi step ahead time series analysis and control of non-linear dynamic systems [14], [34].
Some comments on the construction of covariance kernels in case of noisy input can be found in [41] (section 2.1.8.2). It is noted that when the inputs contain independent and normally distributed noise, it is possible to calculate the mean and variance of the resulting process. However in [15] and [41] authors wrongly state that the resulting process in not Gaussian.
In [18] a summary of possible applications of kernel convolutions is given. Some of those are non-parametric covariance modelling, dependent Gaussian processes, non-stationary covariance construction, reduction in dimension, etc.
In the work of Paciorek [31], a generalization of the kernel convolution covariance of Higdon et al. (1999) is used to construct new non-stationary kernels. The smoothing properties of the new non-stationary kernels depend on the smoothness of the underlying stationary kernels. A particular attention is given to the non-stationary version of

Matèrn correlation function and its smoothness properties.

2 Problem formulation

We minimize mean response of the simulator

$$\min_x \mathbb{E}_U[f(x,U)] \tag{1}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ is an instance of multivariate random variable U with joint probability distribution and $\mathbb{E}_U$ is the expectation measure. We assume that each evaluation of $f(x,u) \in \mathbb{R}$ involves time consuming call to a deterministic simulator. We say that $\mathbf{X} = \{x^1, ..., x^t\}$ and $\mathbf{U} = \{u^1, ..., u^t\}$, where $t \in \mathbb{N}$ is the number of initial design points (number of calls to the simulator $f(x,u)$) and $\mathbf{Y} = \{f(x^1,u^1), .., f(x^t,u^t)\}$ is the simulator responses at the design points.
By $Y_{(x,u)}(\omega)$ we denote Gaussian process with given mean $m_Y(x,u)$ and given covariance function $cov_Y(x,u;x',u')$

$$\begin{aligned} \mathbb{E}[Y_{(x,u)}(\omega)] &= m_Y(x,u), \\ \mathbb{COV}(Y_{(x,u)}(\omega), Y_{(x',u')}(\omega)) &= cov_Y(x,u;x',u'), \end{aligned} \tag{2}$$

$$Y_{(x,u)}(\omega) \sim \mathcal{GP}\left(m_Y(x,u), cov_Y(x,u;x',u')\right). \tag{3}$$

We assume that $f(x,u)$ can be approximated by $y(x,u)$ which is an instance of $Y_{(x,u)}(\omega)$ and $f(x,U)$ can be represented by $Y_{(x,U)}(\omega)$.
Let $Y^t_{(x,u)}(\omega)$ denote a GP conditioned on the t observations

$$Y^t_{(x,u)}(\omega) = \left[Y_{(x,u)}(\omega) | Y_{(\mathbf{X},\mathbf{U})}(\omega) = \mathbf{Y}\right]. \tag{4}$$

The conditional process $Y^t_{(x,u)}(\omega)$ utilizes data of the observations. The conditional expectation and the conditional covariance [37], [35] are

$$\mathbb{E}\left(Y^t_{(x,u)}(\omega)\right) = m_Y(x,u) + cov_Y(x,u;\mathbf{X},\mathbf{U}) cov_Y(\mathbf{X},\mathbf{U};\mathbf{X},\mathbf{U})^{-1}(Y - m_Y(\mathbf{X},\mathbf{U})), \tag{5}$$

$$\mathbb{COV}\left(Y^t_{(x,u)}(\omega)\right) = cov_Y(x,u;x,u) - cov_Y(x,u;\mathbf{X},\mathbf{U}) cov_Y(\mathbf{X},\mathbf{U};\mathbf{X},\mathbf{U})^{-1} cov_Y(\mathbf{X},\mathbf{U};x,u). \tag{6}$$

We can therefore estimate the mean (predicted kriging mean) and the variance (predicted kriging variance) of $Y^t_{(x,u)}(\omega)$ at any unobserved point (x,u).

3 Projected process

In order to minimize the expectation eq. (1) and take full advantage of Gaussian process metamodel eq. (4), it is possible to construct a new Gaussian process by taking the expectation of the conditioned process with respect to U

$$Z^t_{(x)}(\omega) = \mathbb{E}_U[Y^t_{(x,U)}(\omega)] = \int_{\mathbb{R}^m} Y^t_{(x,u)}(\omega) d_\mu(u), \tag{7}$$

where $d_\mu(u)$ is a probability measure on u.
From eq. (7) we can see that the $Z^t_{(x)}(\omega)$ is a linear combination of Gaussian processes and therefore also a Gaussian process and we call it "projected process". The link can be made with Bayesian Monte Carlo integration [36], where we have Gaussian prior on the function $y^t_{(x,u)}$.

3.1 Expectation

To obtain the mean of $Z_{(x)}(\omega)$ as a function of x we calculate

$$\begin{aligned} m_Z(x) =& \mathbb{E}_Y\left[Z_{(x)}(\omega)\right] = \mathbb{E}_Y\left[\mathbb{E}_U[Y_{(x,U)}(\omega)]\right] = \\ & \int_\Omega \int_{\mathbb{R}^m} y(x,u) d\mu(u) dP(\omega) \end{aligned} \tag{8}$$

If we assume that Fubinis condition is satisfied

$$\int_\Omega \int_{\mathbb{R}^m} |y(x,u)| d\mu(u) dP(\omega) < \infty, \tag{9}$$

we can rewrite eq. (8) as

$$m_Z(x) = \int_{\mathbb{R}^m} \int_{\Omega} y(x,u) dP(\omega) d\mu(u) = \int_{\mathbb{R}^m} \mathbb{E}_Y[Y_{(x,u)}(\omega)] d\mu(u) \tag{10}$$

in other terms, under Fubinis condition,

$$\mathbb{E}_Y\left[Z_{(x)}(\omega)\right] = \mathbb{E}_Y\left[\mathbb{E}_U[Y_{(x,U)}(\omega)]\right] = \mathbb{E}_U\left[\mathbb{E}_Y[Y_{(x,U)}(\omega)]\right]. \tag{11}$$

Note that, because Y^t is conditioned on points with finite values, Fubini's condition seems satisfied (we do not provide a rigorous proof). We see that the mean of $Z_{(x)}(\omega)$ is an integral over the u space of the mean of $Y_{(x,u)}(\omega)$

$$m_Z(x) = \int_{\mathbb{R}^m} m_Y(x,u) d\mu(u). \tag{12}$$

$Z^t_{(x)}(\omega)$ in eq. (7) is defined in terms of the conditional process $Y^t_{(x,u)}(\omega)$, therefore, the mean $m_Y(.)$ is the conditional mean in eq. (5). If $\mathbf{X}_* = [x^1_*, ...x^*_r]^\top$ is a $r \times n$ matrix of r test points we have

$$\begin{aligned} \mathbb{E}\left(Z^t_{(\mathbf{X}_*)}(\omega)\right) &= \int_{\mathbb{R}^m} m_Y(\mathbf{X}_*, u) d\mu(u) \\ &+ \int_{\mathbb{R}^m} cov_Y(\mathbf{X}^*, u; \mathbf{X}, \mathbf{U}) cov_Y(\mathbf{X}, \mathbf{U}; \mathbf{X}, \mathbf{U})^{-1} (Y - m_Y(\mathbf{X}, \mathbf{U})) d\mu(u) \end{aligned} \tag{13}$$

Here $cov_Y(\mathbf{X}, \mathbf{U}; \mathbf{X}, \mathbf{U})^{-1}$ is a $t \times t$ matrix and $(\mathbf{Y} - m_Y(\mathbf{X}, \mathbf{U}))$ is $t \times 1$ vector. If we denote by $\mathbf{c}$ the $t \times 1$ vector $cov_Y(\mathbf{X}, \mathbf{U}; \mathbf{X}, \mathbf{U})^{-1}(\mathbf{Y} - m_Y(\mathbf{X}, \mathbf{U}))$, then eq. (13) becomes

$$\begin{aligned} &\mathbb{E}\left(Z^t_{(\mathbf{X}_*)}(\omega)\right) = \int_{\mathbb{R}^m} m_Y(\mathbf{X}_*, u) d\mu(u) + \sum_{i=1}^{t} c_i \int_{\mathbb{R}^m} cov_Y(\mathbf{X}_*, u; x^i, u^i) d\mu(u) = \\ &\left(\int_{\mathbb{R}^m} m_Y(x^j_*, u) d\mu(u)\right)_{1 \le j \le r} + \left(\sum_{i=1}^{t} c_i \int_{\mathbb{R}^m} cov_Y(x^j_*, u; x^i, u^i) d\mu(u)\right)_{1 \le j \le r}. \end{aligned} \tag{14}$$

3.2 Covariance

To obtain the covariance of $Z_{(x)}(\omega)$ as a function of x and x' we use a similar reasoning

$$\begin{aligned} cov_Z(x, x') = \mathbb{COV}(Z_{(x)}(\omega), Z_{(x')}(\omega)) = \\ \mathbb{E}_Y\left[\left[Z_{(x)}(\omega) - \mathbb{E}_Y\left[Z_{(x)}(\omega')\right]\right]\left[Z_{(x')}(\omega) - \mathbb{E}_Y\left[Z_{(x')}(\omega'')\right]\right]\right] = \\ \mathbb{E}_Y\left[\left[\mathbb{E}_U[Y_{(x,U)}(\omega)] - \mathbb{E}_Y\left[\mathbb{E}_U[Y_{(x,U)}(\omega')]\right]\right]\right. \\ \left.\left[\mathbb{E}_U[Y_{(x',U)}(\omega)] - \mathbb{E}_Y\left[\mathbb{E}_U[Y_{(x',U)}(\omega'')]\right]\right]\right] \end{aligned} \tag{15}$$

Using the same transformation as in eq. (10) and assuming that eq. (9) is satisfied, we can rewrite eq. (15) as

$$\begin{aligned} cov_Z(x, x') = \mathbb{E}_Y\Big[&\left(\mathbb{E}_U[Y_{(x,U)}(\omega)] - \mathbb{E}_U\left[\mathbb{E}_Y[Y_{(x,U)}(\omega')]\right]\right) \\ &\left(\mathbb{E}_U[Y_{(x',U)}(\omega)] - \mathbb{E}_U\left[\mathbb{E}_Y[Y_{(x',U)}(\omega'')]\right]\right)\Big] \end{aligned} \tag{16}$$

Using the linearity of the expectation we write

$$\begin{aligned} cov_Z(x, x') = \mathbb{E}_Y\Big[&\mathbb{E}_U\Big(Y_{(x,U)}(\omega) - \mathbb{E}_Y[Y_{(x,U)}(\omega')]\Big) \\ &\mathbb{E}_U\Big(Y_{(x',U)}(\omega) - \mathbb{E}_Y[Y_{(x',U)}(\omega'')]\Big)\Big] \end{aligned} \tag{17}$$

By the definition of expectation

$$\begin{aligned} cov_Z(x, x') = \int_{\Omega}\Big[\int_{\mathbb{R}^m}&\left(Y_{(x,u)}(\omega) - \int_{\Omega} Y_{(x,u)}(\omega') dP(\omega')\right) d\mu(u) \\ &\int_{\mathbb{R}^m}\left(Y_{(x',u')}(\omega) - \int_{\Omega} Y_{(x',u')}(\omega'') dP(\omega'')\right) d\mu(u')\Big] dP(\omega) \end{aligned} \tag{18}$$

rewriting we obtain

$$cov_Z(x,x') = \int_\Omega \Big[\int_{\mathbb{R}^m} \int_{\mathbb{R}^m} \left(Y_{(x,u)}(\omega) - \int_\Omega Y_{(x,u)}(\omega') dP(\omega') \right) \left(Y_{(x',u')}(\omega) - \int_\Omega Y_{(x',u')}(\omega'') dP(\omega'') \right) d\mu(u) d\mu(u') \Big] dP(\omega) \tag{19}$$

If the Fubini's condition

$$\int_\Omega \Big[\int_{\mathbb{R}^m} \int_{\mathbb{R}^m} \Big| \left(Y_{(x,u)}(\omega) - \int_\Omega Y_{(x,u)}(\omega') dP(\omega') \right) \left(Y_{(x',u')}(\omega) - \int_\Omega Y_{(x',u')}(\omega'') dP(\omega'') \right) \Big| d\mu(u) d\mu(u') \Big] dP(\omega) < \infty \tag{20}$$

is satisfied (see related comment in section 3.1), it is possible to exchange the order of integration and we can rewrite eq. (19) as

$$cov_Z(x,x') = \int_{\mathbb{R}^m} \int_{\mathbb{R}^m} \int_\Omega \left(Y_{(x,u)}(\omega) - \int_\Omega Y_{(x,u)}(\omega') dP(\omega') \right) \left(Y_{(x',u')}(\omega) - \int_\Omega Y_{(x',u')}(\omega'') dP(\omega'') \right) dP(\omega) d\mu(u) d\mu(u') \tag{21}$$

We can see that the covariance of $Z_{(x)}(\omega)$ is a double integral over the $u \times u$ space of the covariance of $Y_{(x,u)}(\omega)$

$$cov_Z(x;x') = \int_{\mathbb{R}^m} \int_{\mathbb{R}^m} cov_Y(x,u;x'u') d\mu(u) d\mu(u') \tag{22}$$

Therefore if $\mathbf{X}_* = [x_*^1, ..., x_*^r]^\top$ is a $r \times n$ matrix of r test points, then

$$\mathbb{COV}\left(Z_{(\mathbf{X}_*)}(\omega)\right) = \left(cov_Z(x^i;x^j)\right)_{1\le i,j\le r} = \left(\int_{\mathbb{R}^m} \int_{\mathbb{R}^m} cov_Y(x^i,u;x^j,u') d\mu(u) d\mu(u') \right)_{1\le i,j\le r} \tag{23}$$

where $cov_Y(.)$ is given by conditional covariance of eq. (6). Let C denote $t \times t$ matrix $cov_Y(\mathbf{X},\mathbf{U};\mathbf{X},\mathbf{U})^{-1}$, then we can write

$$\begin{aligned} &\mathbb{COV}\left(Z^t_{(\mathbf{X}_*)}(\omega)\right) = \\ &\left(\int_{\mathbb{R}^m} \int_{\mathbb{R}^m} cov_Y(x_*^i,u;x_*^j,u') d\mu(u) d\mu(u') \right)_{1\le i,j\le r} \\ &- \left(\int_{\mathbb{R}^m} \int_{\mathbb{R}^m} \sum_k^t \sum_l^t c_{kl} cov_Y(x_*^i,u;x^l,u^l) cov_Y(x^k,u^k;x_*^j,u') d\mu(u) d\mu(u') \right)_{1\le i,j\le r} . \end{aligned} \tag{24}$$

By taking the integrals of each element in the sum and by the independence of the two integrals we rewrite eq. (24) as

$$\begin{aligned} &\mathbb{COV}\left(Z^t_{(\mathbf{X}_*)}(\omega)\right) = \\ &\left(\int_{\mathbb{R}^m} \int_{\mathbb{R}^m} cov_Y(x_*^i,u;x_*^j,u') d\mu(u) d\mu(u') \right)_{1\le i,j\le r} \\ &- \left(\sum_k^t \sum_l^t c_{kl} \int_{\mathbb{R}^m} cov_Y(x_*^i,u;x^l,u^l) d\mu(u) \int_{\mathbb{R}^m} cov_Y(x^k,u^k;x_*^j,u') d\mu(u') \right)_{1\le i,j\le r} \end{aligned} \tag{25}$$

4 Solutions of the integrals

To calculate the covariance and mean using formulas eq. (25) and eq. (14), we need to solve integrals in the form

$$I(x,u,x') = \int_{\mathbb{R}^m} cov_Y(x,u;x',u') d\mu(u'). \tag{26}$$

The covariance is typically built as a product of covariances for each dimension

$$cov_Y(x,u;x',u') = \prod_{i=1}^n cov_Y(x_i;x_i') \prod_{i=1}^m cov_Y(u_i;u_i'). \tag{27}$$

Since we consider stationary processes, we have

$$\prod_{i=1}^{m} cov_Y(u_i; u_i') = k(u' - u)$$

and for $d\mu(u')$ we have a probability measure with density function $g(u')$, then using the definition of convolutions

$$\begin{aligned} I(x, u, x') &= \prod_{i=1}^{n} cov_Y(x_i; x_i') \int_{\mathbb{R}^m} k(u' - u) g(u') d(u') \\ &= \prod_{i=1}^{n} cov_Y(x_i; x_i')\ g * k(u). \end{aligned} \tag{28}$$

We can view the new covariance as a weighted average of the old covariance at the point u, where the weighting function $g(-u')$ is the density function of U.
To approximately solve the integral for an arbitrary kernel and normally independently distributed U, it is possible to use Taylor approximation in a fashion similar as in [15].

4.1 Analytical solutions of the integrals for Gaussian kernel and independently normally distributed U

Here we show the analytical solution of the integrals for Gaussian kernels in the case of independently normally distributed U. For simplification we assume that the mean function $m_Y(x)$ is constant in u, therefore the solution of the first term in eq. (14) reduces to

$$\int_{\mathbb{R}^m} m_Y(\mathbf{X}_*, u) d\mu(u) = m_Y(\mathbf{X}_*)$$

4.1.1 Integral of the covariance

We further assume that $cov_Y(u; u')$ in eq. (26) is a Gaussian kernel of the form

$$cov_Y(u_i; u_i') = exp\left(-\frac{1}{2}\left(\frac{u_i - u_i'}{\theta_i}\right)^2\right). \tag{29}$$

U is a random variable that comes from a multivariate normal distribution with density function $f_{N(\mu,\Sigma)}$, where $\mu = \mu_i$ is the mean and $\Sigma = \text{diag}(\sigma_i^2)$ is the covariance matrix.

We look at the product of Gaussian covariances as a density function

$$\prod_{i=1}^{m} cov_Y(u_i; u_i') = (2\pi)^{m/2} |D|^{1/2} f_{N(0,D)}(u' - u) \tag{30}$$

where $D = \text{diag}(\theta_i^2)$, then using eq. (28) the integral eq. (26) becomes

$$\begin{aligned} I(x, u, x') = &\prod_{i=1}^{n} cov_Y(x_i; x_i')\ (2\pi)^{m/2} |D|^{1/2} \int_{\mathbb{R}^m} f_{N(0,D)}(u - u') f_{N(\mu,\Sigma)}(u') du' = \\ &\prod_{i=1}^{n} cov_Y(x_i; x_i')\ (2\pi)^{m/2} |D|^{1/2} f_{N(\mu,\Sigma)} * f_{N(0,D)}(u). \end{aligned} \tag{31}$$

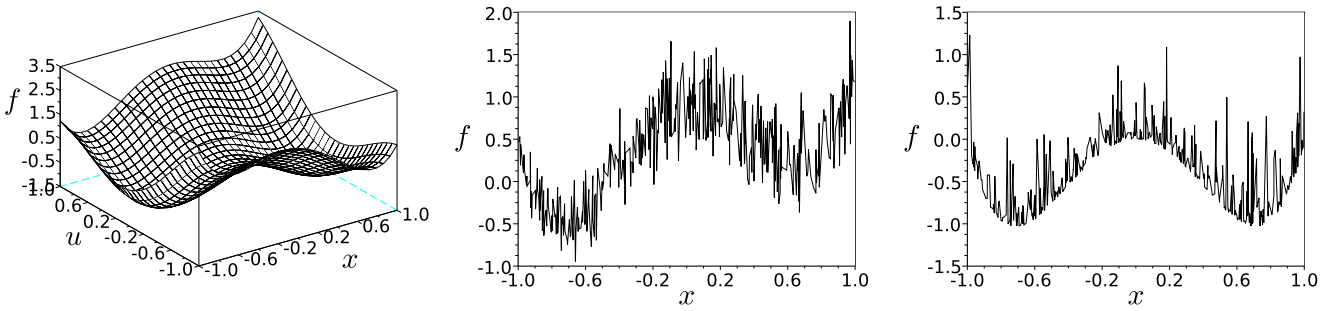
The convolution of two independent normal distributions is also normal,

$$f_{N(\mu,\Sigma)} * f_{N(0,D)}(u) = f_{N(\mu,\Sigma+D)}(u) \tag{32}$$

or

$$\begin{aligned} I(x, u, x') = &\prod_{i=1}^{n} cov_Y(x_i; x_i')\ (2\pi)^{m/2} |D|^{1/2} f_{N(\mu,\Sigma+D)}(u) = \\ &\prod_{i=1}^{n} cov_Y(x_i; x_i')\ \frac{|D|^{1/2}}{|\Sigma + D|^{1/2}} exp\left[-\frac{1}{2}(u - \mu)^T (\Sigma + D)^{-1} (u - \mu)\right]. \end{aligned} \tag{33}$$

A similar result is obtained under the name of Bayes-Hermite Quadrature [29].



(a) Camel-back test function

(b) Noisy instance for 1st case, $U \sim \mathcal{N}(0.5, 0.01)$

(c) Noisy instance for 2nd case, $U \sim \mathcal{N}(0.05, 0.04)$

Figure 1: Noisy Camel back test case. Plots (b) and (c) show $f(x, u)$, where the u's are i.i.d. for each new x.

4.1.2 Double integral of the covariance

To calculate the covariance of Z using eq. (25), we also need the analytical solution of the integral

$$I(x, x') = \int_{\mathbb{R}^m} \int_{\mathbb{R}^m} cov_Y(x, u; x', u') d\mu(u') d\mu(u). \tag{34}$$

Using the same reasoning as in the previous section, we write

$$\begin{aligned} I(x, x') = \prod_{i=1}^{n} cov_Y(x_i; x_i') \ \frac{|D|^{1/2}}{|\Sigma + D|^{1/2}} \int_{\mathbb{R}^m} exp\left[-\frac{1}{2}(u-\mu)^T(\Sigma + D)^{-1}(u-\mu)\right] d\mu(u) = \\ \prod_{i=1}^{n} cov_Y(x_i; x_i') \ (2\pi)^{m/2}|D|^{1/2} \int_{\mathbb{R}^m} f_{N(0,\Sigma+D)}(\mu - u) f_{N(\mu,\Sigma)}(u) du = \\ \prod_{i=1}^{n} cov_Y(x_i; x_i') \ (2\pi)^{m/2}|D|^{1/2} f_{N(0,\Sigma+D)} * f_{N(\mu,\Sigma)}(\mu) = \\ \prod_{i=1}^{n} cov_Y(x_i; x_i') \ (2\pi)^{m/2}|D|^{1/2} f_{N(\mu,2\Sigma+D)}(\mu) = \\ \prod_{i=1}^{n} cov_Y(x_i; x_i') \ \frac{|D|^{1/2}}{|2\Sigma + D|^{1/2}}. \end{aligned} \tag{35}$$

The results of eq. (33) and eq. (35) are identical to those presented in [15] (page 54). As mentioned in [15] the new kernel associated to Z^t is a function of $\Sigma + D = \text{diag}\left[\theta_1^2 + \sigma_1^2, ..., \theta_m^2 + \sigma_m^2\right]$, so that each parameter of the covariance function is weighted by the corresponding uncertainty of U in the given dimension.

4.2 2D example

Our intention here is to illustrate the approximation of $\mathbb{E}_U[f(x, U)]$ using $Z^t_{(x)}(\omega)$ and $Y_{(x,u)}(\omega)$ when $U \sim \mathcal{N}(\mu, \sigma^2)$. In this case we assume that $(x, u) \in \mathbb{R} \times \mathbb{R}$ and that the simulator $f(x, u)$ is the Camel back function

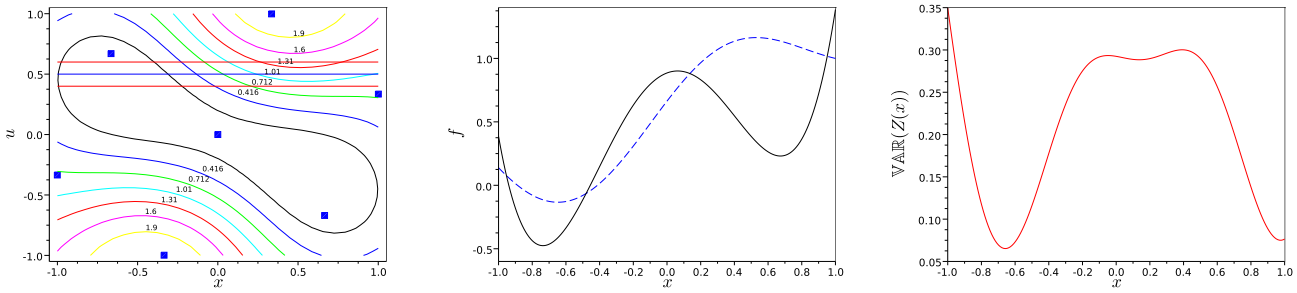
$$f(x, u) = (4 - 2.1u^2 + \frac{1}{3}u^4)u^2 + xu + (-4 + 4x^2)x^2 \tag{36}$$

in the interval $[-1, -1]^2$ (fig. 1).

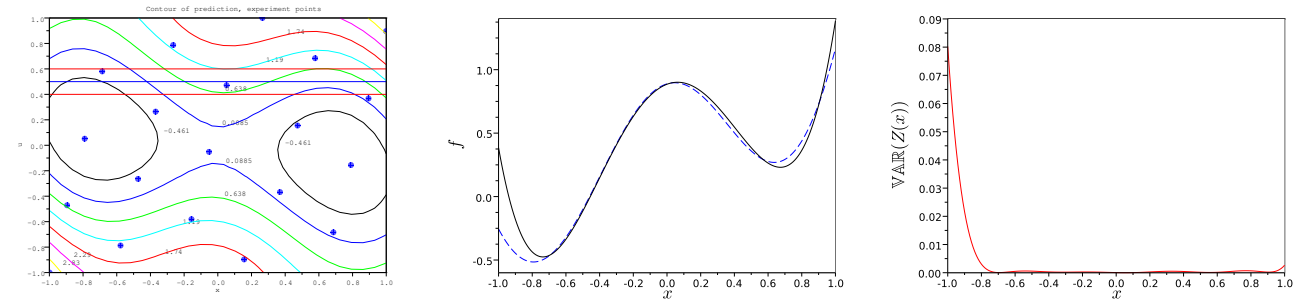
We use two different data sets: a 7 and a 20 points Latin Hypercube (LH) design optimized by MSE criterion [1]. The R^2 approximations measures of $f(x, u)$ by $\mathbb{E}[Y^t_{(x,u)}(\omega)]$ for both data sets are 0.6073 and 0.9842, respectively. We consider two uncertainty cases: in the first case, $\mu = 0.5$ and $\sigma = 0.1$, and in the second case, $\mu = 0.05$ and $\sigma = 0.2$. Instances of the function $f(x, U)$ are shown in fig. 1, by independently sampling u from each distribution.

The results for both cases are shown in fig. 2. The first and second rows corresponds to the first uncertainty case and the third and fourth - to the second. The targeted $\mathbb{E}_U[f(x, U)]$ is obtained by numerical integration.

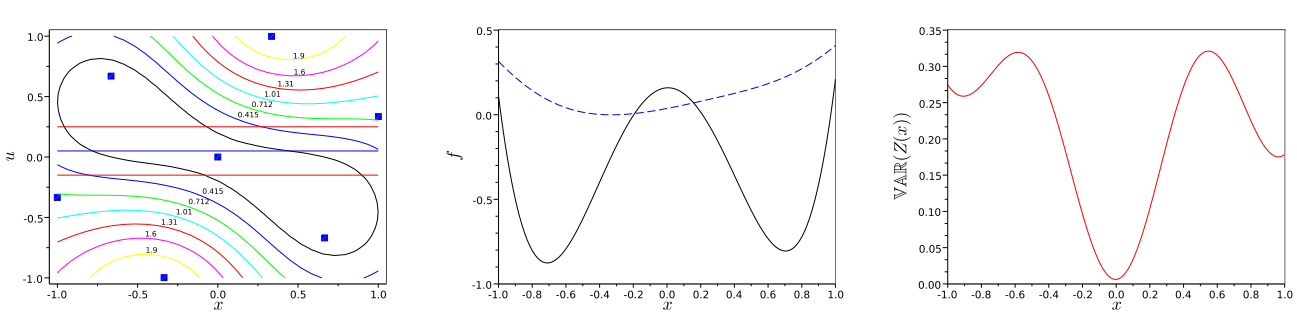
Figure 2 confirms, that the prediction accuracy of Z^t for $\mathbb{E}_U[f(x, U)]$ depends on our knowledge about the underlying function $f(x, u)$. Adding data points improves the approximation of $f(x, u)$, i.e., takes $\mathbb{E}(Z^t_{(x)}(\omega))$ closer to $\mathbb{E}_U[f(x, U)]$ and decreases $\mathbb{VAR}(Z^t_{(x)}(\omega))$.



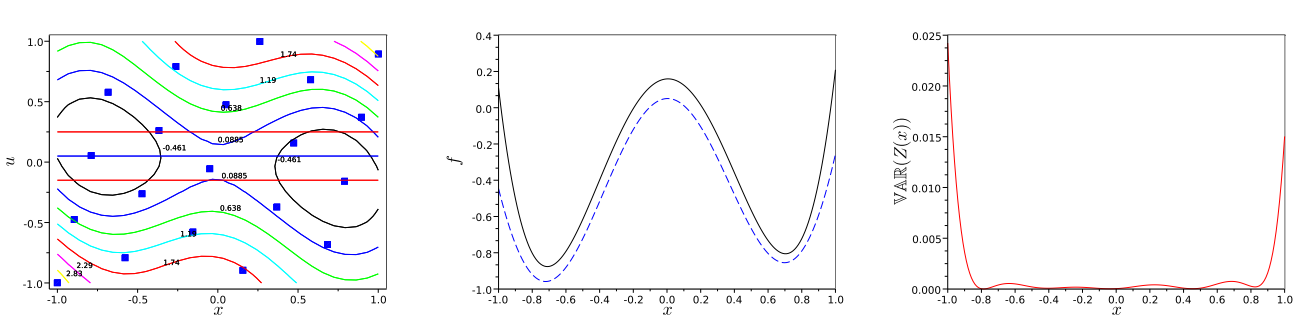
(a) DOE, $\mathbb{E}(Y^t_{(x,u)}(\omega))$, $1^{st}case$ (b) $\mathbb{E}(Z^t_{(x)}(\omega))$ and $\mathbb{E}(f(x,U))$ (c) $\mathbb{VAR}(Z^t_{(x)}(\omega))$



(d) DOE, $\mathbb{E}(Y^t_{(x,u)}(\omega))$, $1^{st}case$ (e) $\mathbb{E}(Z^t_{(x)}(\omega))$ and $\mathbb{E}(f(x,U))$ (f) $\mathbb{VAR}(Z^t_{(x)}(\omega))$



(g) DOE, $\mathbb{E}(Y^t_{(x,u)}(\omega))$, $2^{nd}case$ (h) $\mathbb{E}(Z^t_{(x)}(\omega))$ and $\mathbb{E}(f(x,U))$ (i) $\mathbb{VAR}(Z^t_{(x)}(\omega))$



(j) DOE, $\mathbb{E}(Y^t_{(x,u)}(\omega))$, $2^{nd}case$ (k) $\mathbb{E}(Z^t_{(x)}(\omega))$ and $\mathbb{E}(f(x,U))$ (l) $\mathbb{VAR}(Z^t_{(x)}(\omega))$

Figure 2: Kriging approximations of $f(x,u)$ and $\mathbb{E}_U(f(x,U))$ for the Camel back test function. Left column, DOE (squares) and contour lines of predicted mean $\mathbb{E}(Y^t_{(x,u)}(\omega))$. Middle column, predicted $\mathbb{E}(Z^t_{(x)}(\omega))$ (dashed) and $\mathbb{E}_U(f(x,U))$ (solid). Right column, predicted variance $\mathbb{VAR}(Z^t_{(x)}(\omega))$. First and second rows, $U \sim \mathcal{N}(0.5, 0.01)$, third and fourth rows, $U \sim \mathcal{N}(0.05, 0.04)$.

5 Infill strategy for robust global optimization

In order to solve the optimization problem of eq. (1), we utilize the statistical models of $f(x,u)$ and $\mathbb{E}_U[f(x,U)]$ (eq. (4) and eq. (7)) to define an infill sampling criterion [40]. We follow the EGO philosophy [25] that employs kriging models as approximations of a deterministic function and provides a measure of local uncertainty.
We propose an infill strategy that consists of two steps: at first we choose x^{next} by maximizing an Expected Improvement of the projected process $Z^t_{(x)}(\omega)$, eq. (7). x^{next} is an interesting point in the x (deterministic) space. Then we choose a new point for the actual simulation (x^{t+1}, u^{t+1}), in (x,u) (joint) space by minimizing the variance of the projected process $Z^{t+1}_{(x)}(\omega)$ at the interesting point x^{next}. Two goals are thus pursued simultaneously: a rapid minimization of the objective (mean response), and a spare use of the real (computationally costly) simulator f by a controlled sampling of the uncertain parameters.

5.1 Maximization of Expected Improvement

We define the improvement as

$$I^t_x(\omega) = \left[f^t_{min} - Z^t_{(x)}(\omega) \right]^+, \tag{37}$$

where $[.]^+$ denotes $\max(0,.)$.
As we do not have observations of $\mathbb{E}_U[f(x,U)]$, we say that f^t_{min} is a minimum of the predicted mean of the projected process $Z^t_{(x)}(\omega)$

$$f^t_{min} = \min_x \mathbb{E}\left[Z^t_{(x)}(\omega) \right]. \tag{38}$$

The Expected Improvement is simply

$$EI_Z(x) = \mathbb{E}\left[I^t_x(\omega) \right] = \mathbb{E}\left[f^t_{min} - Z^t_{(x)}(\omega) \right]^+ \tag{39}$$

for which we have an analytical expression in [25]

$$EI_Z(x) = \mathbb{VAR}(Z^t_{(x)}(\omega)) \left[u\Phi(u) + \phi(u) \right], \tag{40}$$

where

$$u = \frac{f^t_{min} - \mathbb{E}(Z^t_{(x)}(\omega))}{\mathbb{VAR}(Z^t_{(x)}(\omega))}$$

and where Φ and ϕ are the normal cumulative distribution function and density function, respectively.

By maximizing EI eq. (40) we obtain an interesting point in the x space,

$$x^{next} = \arg\max_x EI_Z(x). \tag{41}$$

5.2 Minimization of Variance

Maximizing the EI gives some insight (x^{next}) on where to sample in the x space in order to optimize ($\min_x \mathbb{E}_U[f(x,U)]$). However, before the simulator f can be called, u must be chosen. In order to sample points in the (x,u) space, we propose to use the point that reduces the most the uncertainty in the estimated $\mathbb{E}_U[f(x,U)]$ at the point of interest x^{next}. The process Z^t approximates $\mathbb{E}_U[f(x,U)]$. It is a projection of Y^t from the (x,u) to the x space and the variance of Z^t at a given x depends on our knowledge about the simulator around our point of interest in the (x,u) space. We choose the next point for simulation (x^{t+1}, u^{t+1}) such that

$$(x^{t+1}, u^{t+1}) = \arg\min_{x,u} \mathbb{VAR}\ Z^{t+1}_{(x^{next})}(\omega). \tag{42}$$

Note that, such sampling criterion follows the principle of deterministic EGO, where sampling at a new point reduces the uncertainty at this point to zero, thus eventually allowing sample points outside of current minimum in unexplored areas. The sampling criterion of eq. (42) involves a non linear optimization. With the estimation of kriging hyper parameters, the determination of f^t_{min} eq. (38), the maximization of EI eq. (41), this is the fourth non linear optimization of the overall procedure, which therefore ends up being somewhat calculation intensive. Note however, that none of these optimization tasks involve calls to the simulator f. They only work with the underlying regression models Y and Z. This is the reason why the method described in this paper, like all kriging based methods, is suitable

for computationally expesive simulators. From eq. (25) we obtain

$$\begin{aligned}\mathbb{VAR}\ Z^{t+1}_{(x^{next})}(\omega) = &\left(\int_{\mathbb{R}^m}\int_{\mathbb{R}^m} cov_Y(x^{next},u;x^{next},u')d\mu(u)d\mu(u')\right)\\ -&\Big(\int_{\mathbb{R}^m} cov_Y(x_{next},u;\mathbf{X}^{t+1},\mathbf{U}^{t+1})d\mu(u)\\ &cov_Y(\mathbf{X}^{t+1},\mathbf{U}^{t+1};\mathbf{X}^{t+1},\mathbf{U}^{t+1})^{-1}\\ &\int_{\mathbb{R}^m} cov_Y(\mathbf{X}^{t+1},\mathbf{U}^{t+1};x^{next},u')d\mu(u'),\Big)\end{aligned} \tag{43}$$

where $\mathbf{X}^{t+1}=(\mathbf{X}^t,\mathbf{x}^{t+1})$ and $\mathbf{U}^{t+1}=(\mathbf{U}^t,\mathbf{u}^{t+1})$ and $\mathbf{X}^t,\ \mathbf{U}^t$ are observed data points. In $\mathbb{VAR}\ Z^{t+1}$ (eq. (42), eq. (43)) the hyper parameters of the underlying Gaussian process Y are estimated using only the t previously observed data points $(\mathbf{X}^t,\mathbf{U}^t,\mathbf{Y}^t)$.
In order to solve the minimization task of eq. (42) we need to calculate eq. (43) for different $(\mathbf{x}^{t+1},\mathbf{u}^{t+1})$. The first integral in eq. (43) is constant with respect to (x^{t+1},u^{t+1}) and we denote it as σ^2_Z

$$\sigma^2_Z=\left(\int_{\mathbb{R}^m}\int_{\mathbb{R}^m} cov_Y(x^{next},u;x^{next},u')d\mu(u)d\mu(u')\right).$$

We also define the vector

$$k(x^{t+1},u^{t+1})=\int_{\mathbb{R}^m} cov_Y(\mathbf{X}^{t+1},\mathbf{U}^{t+1};x_{next},u)d\mu(u),$$

the matrix

$$\mathbf{C}^{t+1}=cov_Y(\mathbf{X}^{t+1},\mathbf{U}^{t+1};\mathbf{X}^{t+1},\mathbf{U}^{t+1})$$

and the upper triangular matrix $\mathbf{T}^{t+1}$ obtained by Cholesky decomposition such that $\mathbf{T}^{t+1\prime}\mathbf{T}^{t+1}=\mathbf{C}^{t+1}$ (where $'$ denotes transposition). Now we can express the variance in eq. (43) as

$$\begin{aligned}\mathbb{VAR}Z^{t+1}_{(x^{next})}(\omega) =&\sigma^2_Z - k(x^{t+1},u^{t+1})'(C^{t+1})^{-1}k(x^{t+1},u^{t+1})\\ =&\sigma^2_Z - k(x^{t+1},u^{t+1})'(\mathbf{T}^{t+1})^{-1}(\mathbf{T}^{t+1\prime})^{-1}k(x^{t+1},u^{t+1})\\ =&\sigma^2_Z - ((\mathbf{T}^{t+1\prime})^{-1}k(x^{t+1},u^{t+1}))'(\mathbf{T}^{t+1\prime})^{-1}k(x^{t+1},u^{t+1}).\end{aligned} \tag{44}$$

To calculate the above variance we only need to solve the linear system

$$(\mathbf{T}^{t+1\prime})\mathbf{b}=\mathbf{k}'.$$

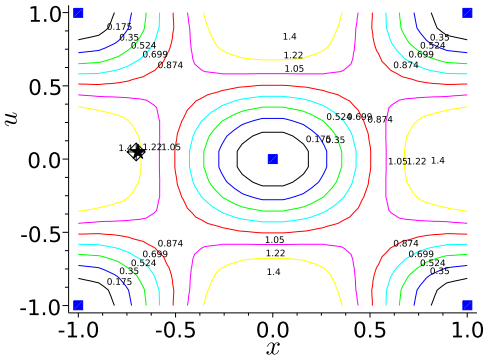
Cholesky decomposition has complexity $O((t+1)^3)$, and it becomes time consuming as t increases. To further decrease the computation time, we use Cholesky decomposition of block matrices of $\mathbf{C}^{t+1}$,

$$\mathbf{C}^{t+1}=\begin{bmatrix}\mathbf{C}^t & \mathbf{c}^{t+1}\\ \mathbf{c}^{t+1\prime} & s\end{bmatrix}$$

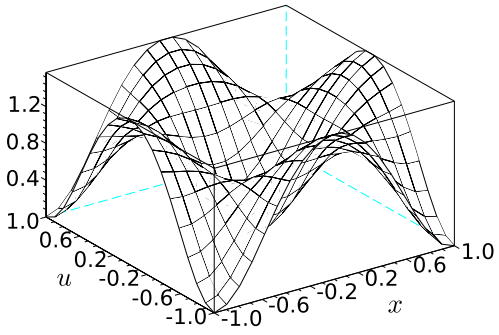
where $\mathbf{c}^{t+1}=cov_Y(\mathbf{X}^t,\mathbf{U}^t;\mathbf{x}^{t+1},\mathbf{u}^{t+1})$ and $s=cov_Y(\mathbf{x}^{t+1},\mathbf{u}^{t+1};\mathbf{x}^{t+1},\mathbf{u}^{t+1})$. Let $\mathbf{T}^t$ be the upper triangular matrix obtained by Cholesky decomposition $\mathbf{T}^{t\prime}\mathbf{T}^t=\mathbf{C}^t$. $\mathbf{T}^t$ is independent of $(\mathbf{x}^{t+1},\mathbf{u}^{t+1})$. $\mathbf{T}^{t+1}$ can be expressed in terms of $\mathbf{T}^t$ for any $(\mathbf{x}^{t+1},\mathbf{u}^{t+1})$ by using $\mathbf{T}^t$, $(\mathbf{T}^t)^{-1}$ and matrix multiplication as

$$\mathbf{T}^{t+1}=\begin{bmatrix}\mathbf{T}^t & (\mathbf{T}^{t\prime})^{-1}\mathbf{c}^{t+1}\\ \mathbf{O} & \left(s-\mathbf{c}^{t+1\prime}(\mathbf{T}^t)^{-1}(\mathbf{T}^{t\prime})^{-1}\mathbf{c}^{t+1}\right)^{1/2}\end{bmatrix}.$$

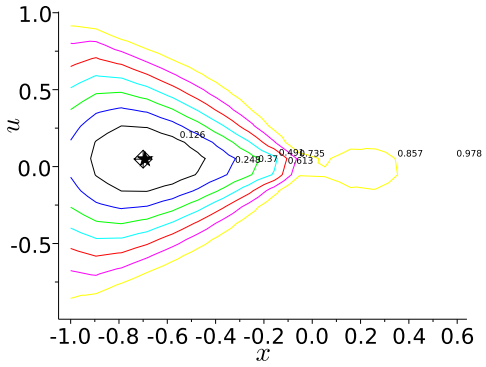
With the block matrix decomposition, the costly $O(t^3)$ Cholesky decomposition needs to be performed only once per optimization (eq. (42)).
The criterion eq. (42) will typically choose a point (x^{t+1},u^{t+1}) that is close to (x^{next},μ). However, once sampling near (x^{next},μ) has been done, it may also choose points that are away from (x^{next},μ). We illustrate such situation in fig. 3 and fig. 4, where we show two cases with different initial DOEs and x^{next}. We show contour lines and surfaces plots of $\mathbb{VAR}\ (Z^{t+1}_{(x^{next})}(\omega))$ and the underlying $\mathbb{VAR}\ (Y^t_{(x,u)}(\omega))$. In the first case the $x^{next}=-0.7$. The corresponding surface of the variance to minimize is shown in the panels (c) and (d) of fig. 3 . In this case the minimum (x^{t+1},u^{t+1}) (star) coincide with the (x^{next},μ) (dimond). In the second case the new point is $x^{next}=-0.5$. The corresponding surface of the variance to minimize is shown in the panels (c) and (d) of fig. 4. As we already have some points in the (x,u) space around (x^{next},μ) (diamond), the minimum point (star) of $\mathbb{VAR}\ (Z^{t+1}_{(x^{next})}(\omega))$ is away from the (x^{next},μ) moving towards the direction of higher uncertainty in $Y^t_{(x,u)}(\omega)$.



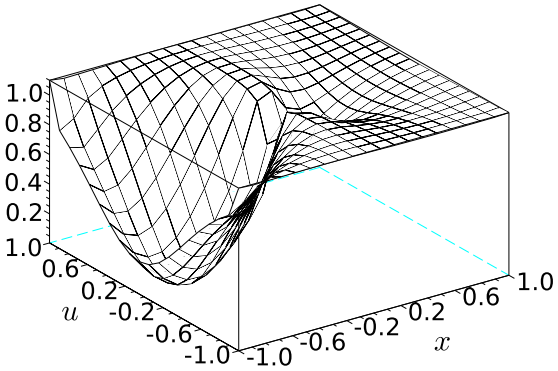
(a) DOE, (x^{next}, μ), (u^{t+1}, x^{t+1}), $\mathbb{VAR}(Y^t_{(x,u)}(\omega))$



(b) $\mathbb{VAR}(Y^t_{(x,u)}(\omega))$



(c) (x^{next}, μ), (u^{t+1}, x^{t+1}), $\mathbb{VAR}(Z^{t+1}_{(x^{next})}(x,u))$



(d) $\mathbb{VAR}(Z^{t+1}_{(x^{next})}(x,u))$

Figure 3: Illustration of the sampling criterion on the Camel back function. Panel (a): contour lines of predicted variance $\mathbb{VAR}(Y^t_{(x,u)}(\omega))$, initial DOE (squares), point (x^{next}, μ) (diamond) and point $(x^{t+1}, u^{t+1}) = \arg\min \mathbb{VAR}\ (Z_{(x^{next})}(\omega))$ (star). (b) : 3D surface of the predicted variance $\mathbb{VAR}(Y^t_{(x,u)}(\omega))$. (c): contour lines of the predicted variance $\mathbb{VAR}(Z^{t+1}_{(x^{next})}(\omega))$ as function of (x^{t+1}, u^{t+1}), the optimum (x^{t+1}, u^{t+1}) (star) and point (x^{next}, μ) (diamond). (d): 3D surface of predicted variance $\mathbb{VAR}(Z^{t+1}_{(x^{next})}(\omega))$. Notice how (x^{next}, μ) and (x^{t+1}, u^{t+1}) coincide.

6 Implementation and Results

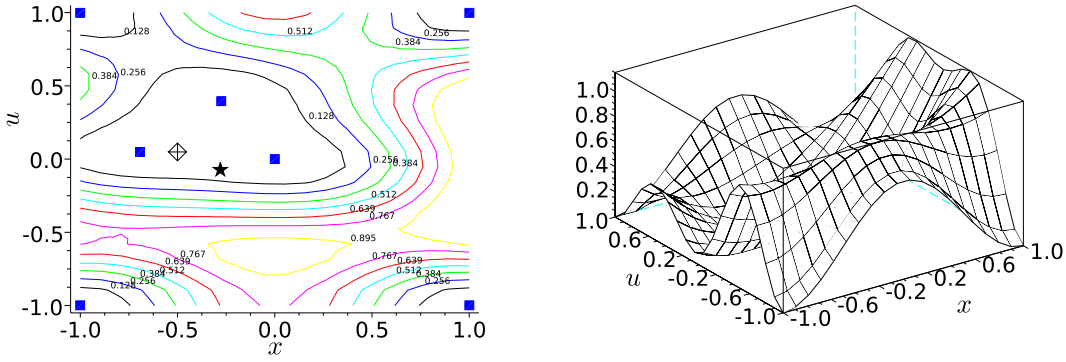
The proposed approach to solve eq. (1) may be summarized in an algorithm 1.

Algorithm 1 Compleate method (EI-VAR)

Create an initial DOE in the (x, u) space and obtain simulator responses.
while stopping criterion not met **do**
  Create regression model $Y^t_{(x,u)}(\omega)$ of the deterministic simulator in the joint (x, u) space;
  Obtain regression model $Z^t_{(x)}(\omega)$ in the deterministic space (x), using covariance information of $Y^t_{(x,u)}(\omega)$;
  Maximize $EI_Z(x)$ to choose x^{next};
  Minimize $\mathbb{VAR}\ Z^{t+1}_{(x^{next})}(\omega)$ to obtain the next point (x^{t+1}, u^{t+1});
  Calculate simulator response at the next point $f(x^{t+1}, u^{t+1})$ and update DOE.
end while

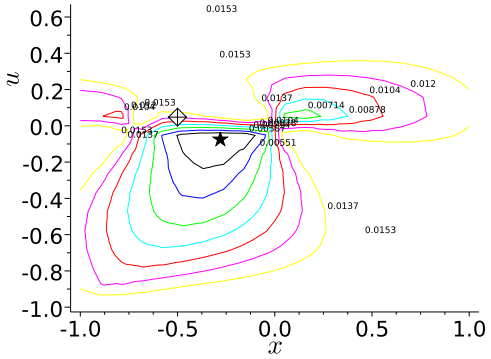
The proposed algorithm involves 4 optimization subtasks at every iteration:

1. Maximizing the Likelihood to obtain covariance parameters and create $Y_{(x,u)}(\omega)$.

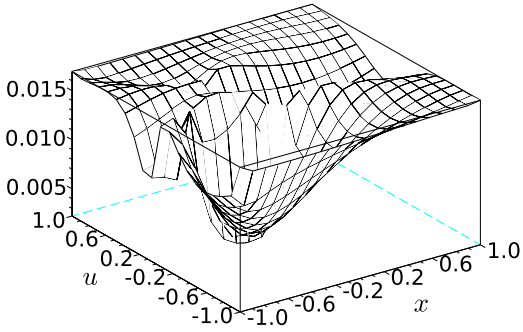


(a) DOE, (x^{next}, μ), (u^{t+1}, x^{t+1}), $\mathbb{VAR}(Y^t_{(x,u)}(\omega))$

(b) $\mathbb{VAR}(Y^t_{(x,u)}(\omega))$



(c) (x^{next}, μ), (u^{t+1}, x^{t+1}), $\mathbb{VAR}(Z^{t+1}_{(x^{next})}(x,u))$



(d) $\mathbb{VAR}(Z^{t+1}_{(x^{next})}(x,u))$

Figure 4: Illustration of the sampling criterion on the Camel back function. Panel (a): contour lines of predicted variance $\mathbb{VAR}(Y^t_{(x,u)}(\omega))$, initial DOE (squares), point (x^{next}, μ) (diamond) and point $(x^{t+1}, u^{t+1}) = \arg\min \mathbb{VAR}\ (Z_{(x^{next})}(\omega))$ (star). (b): 3D surface of the predicted variance $\mathbb{VAR}(Y^t_{(x,u)}(\omega))$. (c): contour lines of the predicted variance $\mathbb{VAR}(Z^{t+1}_{(x^{next})}(\omega))$ as function of (x^{t+1}, u^{t+1}), the optimum (x^{t+1}, u^{t+1}) (star) and point (x^{next}, μ) (diamond). (d): 3D surface of predicted variance $\mathbb{VAR}(Z^{t+1}_{(x^{next})}(\omega))$. Notice how (x^{next}, μ) and (x^{t+1}, u^{t+1}) differ.

2. Finding the minimum
$$f^t_{min} = \min_x \left[\mathbb{E}_Y\ Z^t_{(x)}(\omega)\right].$$

3. Maximizing the expected improvement
$$x^{next} = \arg\max_x EI_Z(x).$$

4. Obtaining the next point in the joined space
$$(x^{t+1}, u^{t+1}) = \arg\min_{x,u} \mathbb{VAR}\ Z^{t+1}_{(x^{next})}(\omega).$$

To solve these optimization problems, which are typically multi-modal we use the CMA-ES algorithm [16], a state-of-the-art global optimizer.
Numerical problems may be encountered during the minimization of the variance in the step 4: when the new point (x, u) is added close to existing points, the covariance matrix becomes badly conditioned. To overcome this problem, we increase the diagonal of the covariance matrix by adding a "noise" term for the calculation of the $\mathbb{VAR}\ [Z^{t+1}_{(x^{next})}(\omega)]$.

In order to reduce the computational burden and eliminate numerical problems encountered solving the 4th sub-optimization, we investigate also simplified approach (algorithm 2) where $x^{t+1} = x^{next}$ and u^{t+1} is sampled from the corresponding distribution (thus eliminating the 4th sub-optimization problem).

In the following, test cases will be shown on analytical "simulators" f. It should be kept kept in mind that these test

Algorithm 2 Simplified approach, with sampling u (EI-Sample)

Create an initial DOE in the (x, u) space and obtain simulator responses.
while stopping criterion not met **do**
  Create regression model $Y^t_{(x,u)}(\omega)$ of the deterministic simulator in the joint (x, u) space;
  Obtain regression model $Z^t_{(x)}(\omega)$ in the deterministic space (x), using covariance information of $Y^t_{(x,u)}(\omega)$;
  Maximize $EI_Z(x)(\omega))$ to choose x^{next};
  Sample u^{t+1} from the corresponding distribution;
  Calculate simulator response at the next point $f(x^{next}, u^{t+1})$ and update DOE.
end while

cases are meant to illustrate the method but that the real scope of application is expensive simulators.

6.1 Graphical illustration on the 2D Camel back test function

To illustrate behaviour of the proposed algorithm we use the same test case as in section 4.2, where $x, u \in \mathbb{R} \times \mathbb{R}$, the simulator $f(x, u)$ is the Camel back function eq. (36) in the interval $[-1, -1]^2$. The initial DOE consists of 5 points (one point in each corner and one point in the center). We consider two uncertainty cases: $\mu = 0.5$ and $\sigma = 0.1$ in the first case and $\mu = 0.05$ and $\sigma = 0.2$, in the second case.
Figure 5 and fig. 6 show some steps of the optimization. It can be seen that already after the 5th iteration the predicted mean $\mathbb{E}(Z_{(x)}(\omega))$ captures quite well the shape of the objective $\mathbb{E}(f(x, U))$ and the predicted variance $\mathbb{VAR}(Z_{(x)}(\omega))$ is quite small around the optimum $x^* = \arg\min_x \mathbb{E}_U(f(x, U))$. During the next iterations points are added in the optimal region and the approximating surface becomes there even more precise. In order to reduce the model uncertainty, at the iterations 7, 13 and 16 the EI chooses points that are away from the current optimum. After theses steps the uncertainty about the model is very small and for the subsequent iterations EI chooses points that are in the vicinity of local minimum.
It is interesting to notice that during the initial iterations the points are spread in the x space but for the u's are close to the mean μ of U. After some iterations when the variance at the point of maximum EI has been reduced, the new points may be away from the (x^{next}, μ) and tend to spread more in the (x, u) space.
Figure 7 provides a comparison between the complete method EI-VAR (algorithm 1) and simplified approach EI-Sample (algorithm 2). The first row of fig. 7 corresponds to the DOE of the EI-Sample in the first case ($U \sim \mathcal{N}(0.5, 0.01)$). The second and the third rows of fig. 7 show the DOE in the second case ($U \sim \mathcal{N}(0.05, 0.04)$) of the complete (EI-VAR) and simplified (EI-Sample) approach, respectively.
As can be seein in fig. 6 and fig. 7, even though the proposed method is global, the DOE produced are focused on the region of the optimum (x^*). Although, in the current case, such early focus is a sign of efficiency, it may become an issue of premature convergence to a non global optimum on more misleading problems.
The early convergence is a typical feature of the one step lookahead EI based optimizers [24], [12], [13]: if the variance far from the current optimum is small, the EI tends to zero in these regions. A good practice when using EI based method is to use an initial DOE which is large enough to capture the overall trend (complexity) of the underlying simulator.
Note that the complete method (EI-VAR) spreads points more away from the (x^{next}, μ). Such a better space filling property explains why, in fig. 8, $\mathbb{E}(Y^t_{(x,u)}(\omega))$ converges faster, in terms of square error, to $f(x, u)$ with the complete approach (EI-VAR) then with the simplified one (EI-Sample).

6.2 Comparisons

In the rest of this section we illustrate performance of the proposed method on analytical test cases and compare it to the MC based optimization.

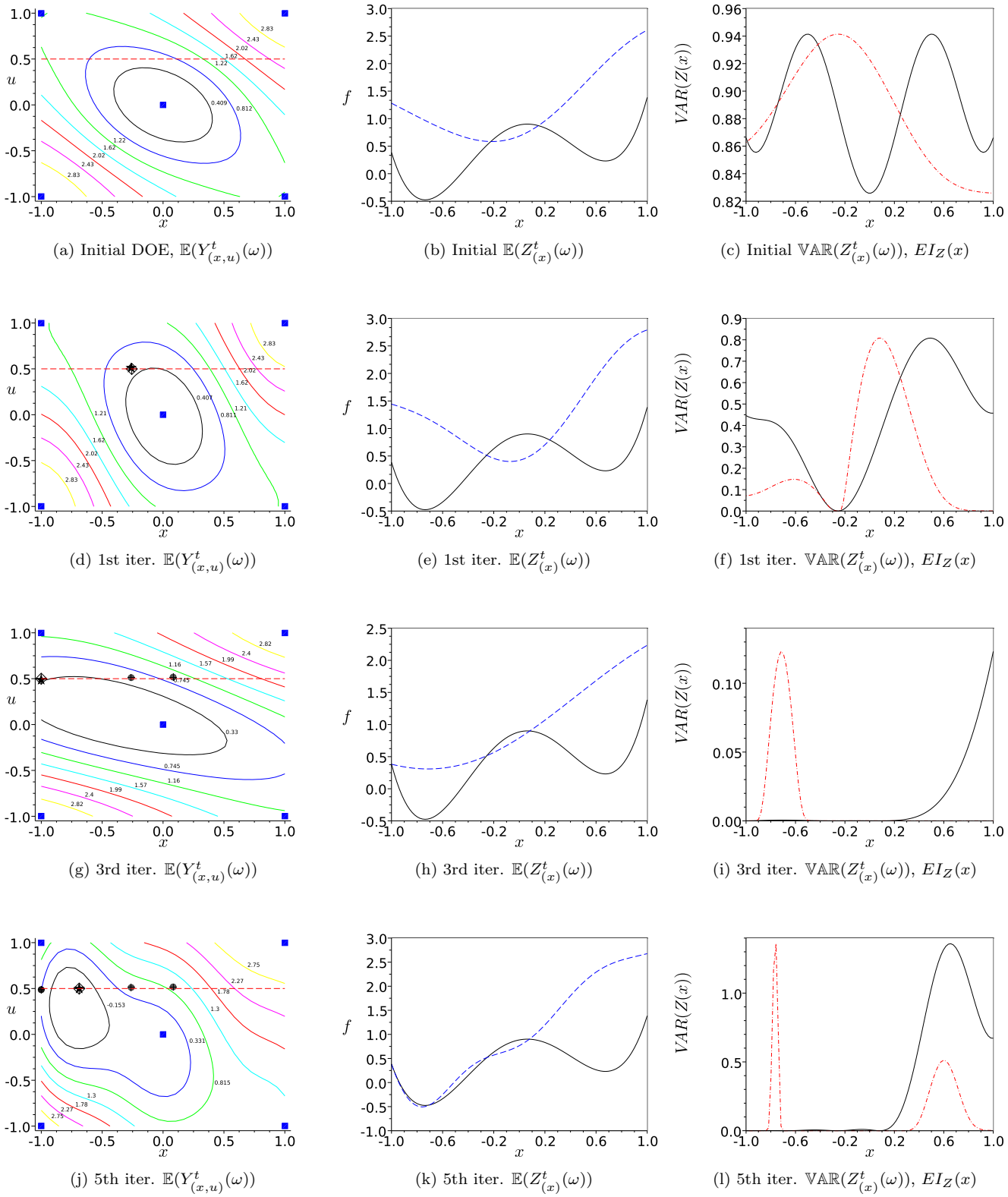
6.2.1 Comparison test cases

The test cases are based on the Michalewitz's function

$$f(x) = -\sum_{i=1}^{d} \sin(x_i) \left[\sin(\frac{ix_i^2}{\pi})\right]^2,$$

where x is in the interval $[0, \pi]^d$. The simulator is composed as an additive function

$$f(x, u) = f(x) + f(u).$$



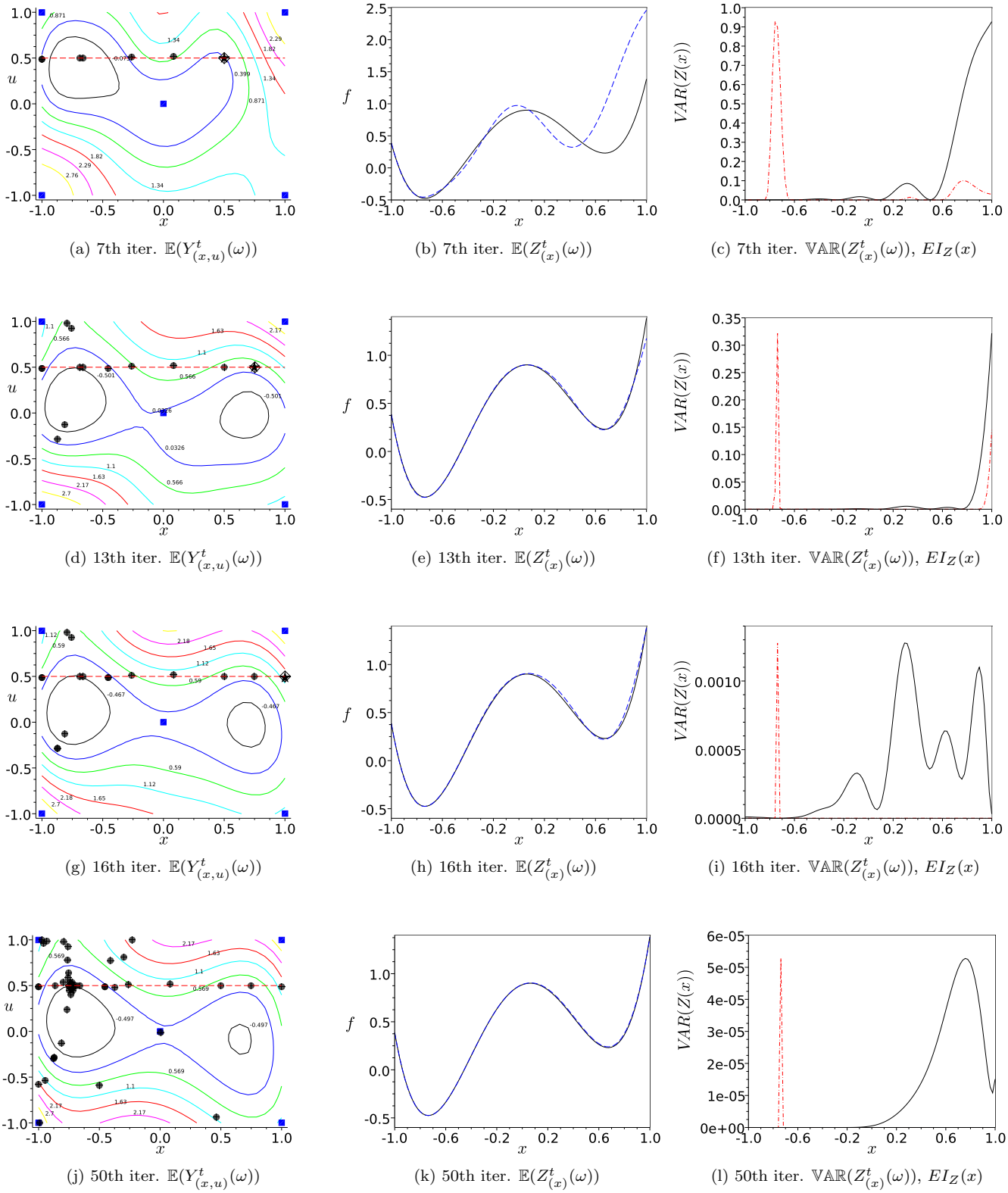
(a) Initial DOE, $\mathbb{E}(Y^t_{(x,u)}(\omega))$ (b) Initial $\mathbb{E}(Z^t_{(x)}(\omega))$ (c) Initial $\mathbb{VAR}(Z^t_{(x)}(\omega))$, $EI_Z(x)$

(d) 1st iter. $\mathbb{E}(Y^t_{(x,u)}(\omega))$ (e) 1st iter. $\mathbb{E}(Z^t_{(x)}(\omega))$ (f) 1st iter. $\mathbb{VAR}(Z^t_{(x)}(\omega))$, $EI_Z(x)$

(g) 3rd iter. $\mathbb{E}(Y^t_{(x,u)}(\omega))$ (h) 3rd iter. $\mathbb{E}(Z^t_{(x)}(\omega))$ (i) 3rd iter. $\mathbb{VAR}(Z^t_{(x)}(\omega))$, $EI_Z(x)$

(j) 5th iter. $\mathbb{E}(Y^t_{(x,u)}(\omega))$ (k) 5th iter. $\mathbb{E}(Z^t_{(x)}(\omega))$ (l) 5th iter. $\mathbb{VAR}(Z^t_{(x)}(\omega))$, $EI_Z(x)$

Figure 5: Illustration of the simultaneous kriging based algorithm on the 2D Camel back function ($U \sim \mathcal{N}(0.5, 0.01)$). Each row represents an iteration (between the 1st and the 5th). Left column: contour lines of predicted mean $\mathbb{E}(Y^t_{(x,u)}(\omega))$, initial DOE (squares), point (x^{next}, μ) where $x^{next} = \arg\max_x EI_Z(x)$ (diamond), new point (x^{t+1}, u^{t+1}) where $(x^{t+1}, u^{t+1}) = \arg\min_{x,u} \mathbb{VAR}\ Z^{t+1}_{(x^{next})}(\omega)$ (star), points from previous iterations (circles). Middle column: objective $\mathbb{E}(f(x, U))$ (solid) and predicted mean $\mathbb{E}(Z^t_{(x)}(\omega))$ (dashed). Right column: predicted variance $\mathbb{VAR}(Z^t_{(x)}(\omega))$ (solid) and rescaled $EI_Z(x)$ (dashed).



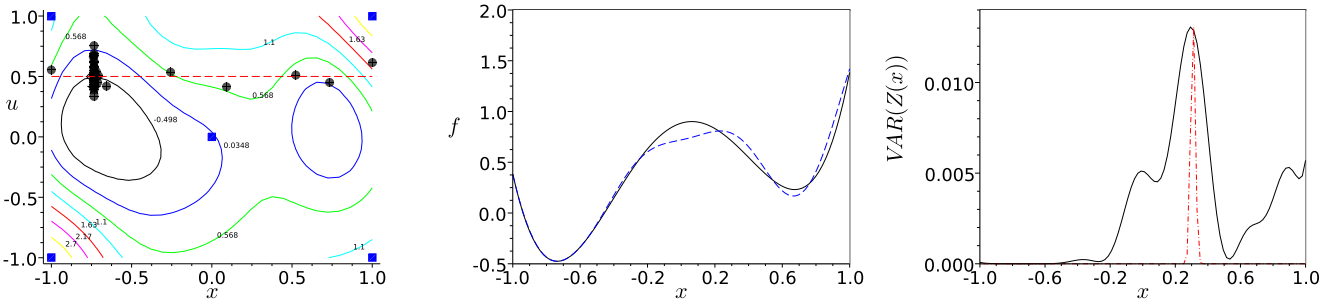
(a) 7th iter. $\mathbb{E}(Y^t_{(x,u)}(\omega))$ (b) 7th iter. $\mathbb{E}(Z^t_{(x)}(\omega))$ (c) 7th iter. $\mathbb{VAR}(Z^t_{(x)}(\omega))$, $EI_Z(x)$

(d) 13th iter. $\mathbb{E}(Y^t_{(x,u)}(\omega))$ (e) 13th iter. $\mathbb{E}(Z^t_{(x)}(\omega))$ (f) 13th iter. $\mathbb{VAR}(Z^t_{(x)}(\omega))$, $EI_Z(x)$

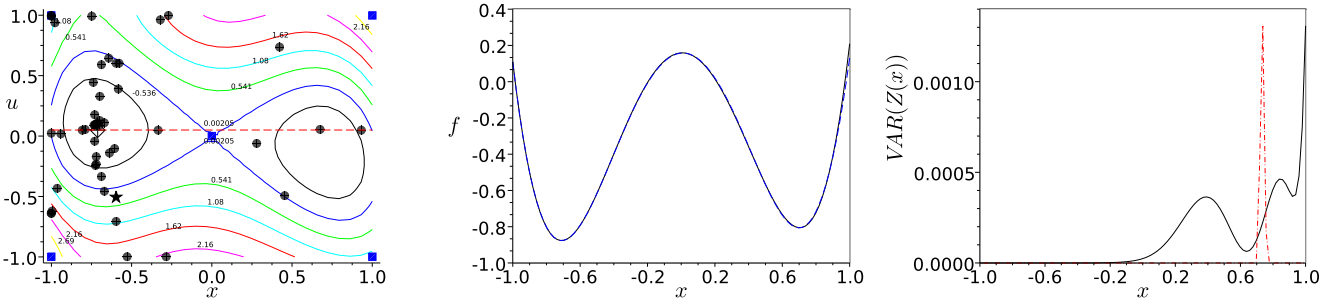
(g) 16th iter. $\mathbb{E}(Y^t_{(x,u)}(\omega))$ (h) 16th iter. $\mathbb{E}(Z^t_{(x)}(\omega))$ (i) 16th iter. $\mathbb{VAR}(Z^t_{(x)}(\omega))$, $EI_Z(x)$

(j) 50th iter. $\mathbb{E}(Y^t_{(x,u)}(\omega))$ (k) 50th iter. $\mathbb{E}(Z^t_{(x)}(\omega))$ (l) 50th iter. $\mathbb{VAR}(Z^t_{(x)}(\omega))$, $EI_Z(x)$

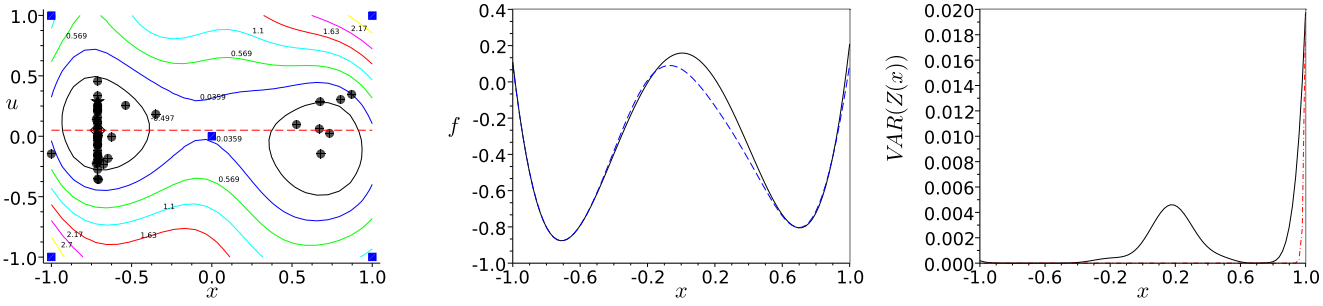
Figure 6: Illustration of the simultaneous kriging based algorithm on the 2D Camel back function ($U \sim \mathcal{N}(0.5, 0.01)$). Continuation of fig. 5. Left column: contour lines of predicted mean $\mathbb{E}(Y^t_{(x,u)}(\omega))$, initial DOE (squares), point (x^{next}, μ) where $x^{next} = \arg\max_x EI_Z(x)$ (diamond), new point (x^{t+1}, u^{t+1}) where $(x^{t+1}, u^{t+1}) = \arg\min_{x,u} \mathbb{VAR}\ Z^{t+1}_{(x^{next})}(\omega)$ (star), points from previous iterations (circles). Middle column: objective $\mathbb{E}(f(x,U))$ (solid) and predicted mean $\mathbb{E}(Z^t_{(x)}(\omega))$ (dashed). Right column: predicted variance $\mathbb{VAR}(Z^t_{(x)}(\omega))$ (solid) and rescaled $EI_Z(x)$ (dashed).



(a) 5Oth iter. $\mathbb{E}(Y^t_{(x,u)}(\omega))$ (b) 5Oth iter. $\mathbb{E}(Z^t_{(x)}(\omega))$ (c) 5Oth iter. $\mathbb{VAR}(Z^t_{(x)}(\omega))$, $EI_Z(x)$

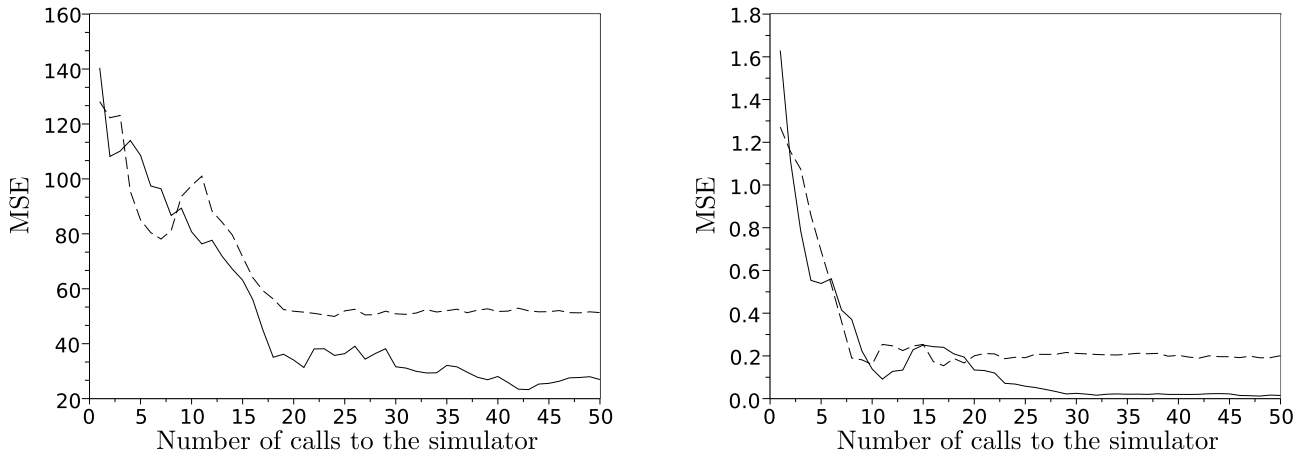


(d) 5Oth iter. $\mathbb{E}(Y^t_{(x,u)}(\omega))$ (e) 5Oth iter. $\mathbb{E}(Z^t_{(x)}(\omega))$ (f) 5Oth iter. $\mathbb{VAR}(Z^t_{(x)}(\omega))$, $EI_Z(x)$



(g) 50th iter. $\mathbb{E}(Y^t_{(x,u)}(\omega))$ (h) 50th iter. $\mathbb{E}(Z^t_{(x)}(\omega))$ (i) 50th iter. $\mathbb{VAR}(Z^t_{(x)}(\omega))$, $EI_Z(x)$

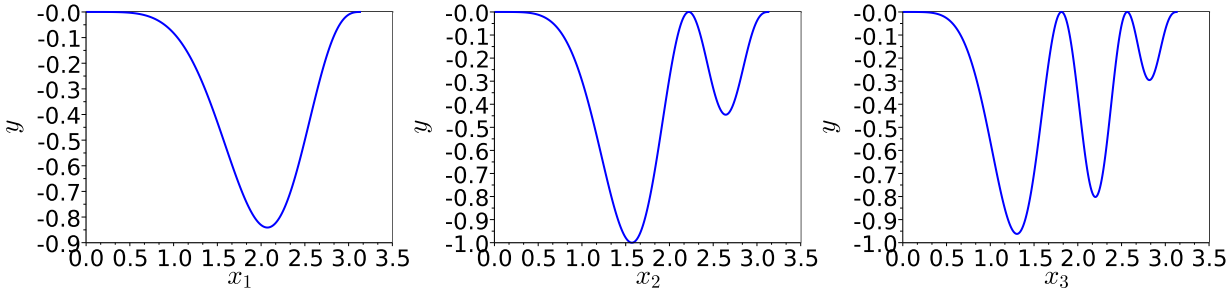
Figure 7: 50th iterations on the Camel back test case. 1st row: $U \sim \mathcal{N}(0.5, 0.01)$, simplified approach (EI-Sample). 2nd and 3rd rows: $U \sim \mathcal{N}(0.05, 0.04)$, complete (EI-VAR) and simplified (EI-Sample) approach, respectively. Left column: contour lines of predicted mean $\mathbb{E}(Y^t_{(x,u)}(\omega))$, initial DOE (squares), point (x^{next}, μ) where $x^{next} = \arg\max_x EI_Z(x)$ (diamond), new point (x^{t+1}, u^{t+1}) where $(x^{t+1}, u^{t+1}) = \arg\min_{x,u} \mathbb{VAR}\ Z^{t+1}_{(x^{next})}(\omega)$ (star), points from previous iterations (circles). Middle column: objective $\mathbb{E}(f(x, U))$ (solid) and predicted mean $\mathbb{E}(Z^t_{(x)}(\omega))$ (dashed). Right column: predicted variance $\mathbb{VAR}(Z^t_{(x)}(\omega))$ (solid) and rescaled $EI_Z(x)$ (dashed).



(a) First case: $U \sim \mathcal{N}(0.5, 0.01)$

(b) Second case: $U \sim \mathcal{N}(0.05, 0.04)$

Figure 8: MSE error of $\mathbb{E}(Y^t_{(x,u)}(\omega))$, compared to $f(x,u)$, where f is the Camel back function. Solid line: complete method (EI-VAR), dashed line : simplified method (EI-Sample).



(a) 1st dimension

(b) 2nd dimension

(c) 3rd dimension

Figure 9: Components of Michalewitz test function for the first, second and third dimension

The complexity (multi-modality) of the test case is dependent on the number of dimensions n and m of x and u, respectively. The components for 3 dimensions are shown in fig. 9.

For comparison we use three test cases with equal number of x and u dimensions:

- 2 dimensional test case ($n = 1$ and $u = 1$), where $\mu = 1.5$ and $\sigma = 0.2$;
- 4 dimensional test case ($n = 2$ and $u = 2$), where $\mu = (1.5, 2.1)'$; and $\Sigma = \text{diag}(0.2^2, 0.2^2)$;
- 6 dimensional test case ($n = 3$ and $u = 3$), where $\mu = (1.5, 2.1, 2)'$; and $\Sigma = \text{diag}(0.2^2, 0.2^2, 0.3^2)$.

6.2.2 Settings of internal optimization

The proposed method is a deterministic algorithm which involves several optimization subtasks. To solve these tasks we use a stochastic optimization method CMA-ES [16], which introduces some randomness in the proposed method. The CMA-ES optimization algorithm has three parameters: a population size λ, an initial standard deviation σ_0 and a maximum number of calls to the objective t^{max}. For each of the four optimization subtasks, the setting are the following:

- for finding kriging hyper parameters (maximizing likelihood) $t^{max} = 200 + 100(n + m)$, $\lambda = 4 + \lfloor 3 \log(n + m) \rfloor$ and σ_0 is one third of the domain.
- for finding the minimum of $\mathbb{E}_Y\ Z^t_{(x)}(\omega)$ and maximizing the expected improvement $EI_Z(x)$, $t^{max} = 300n\lambda$, $\lambda = 4 + \lfloor 3 \log(n) \rfloor n$ and initial standard deviation σ_0 is one half of the domain.

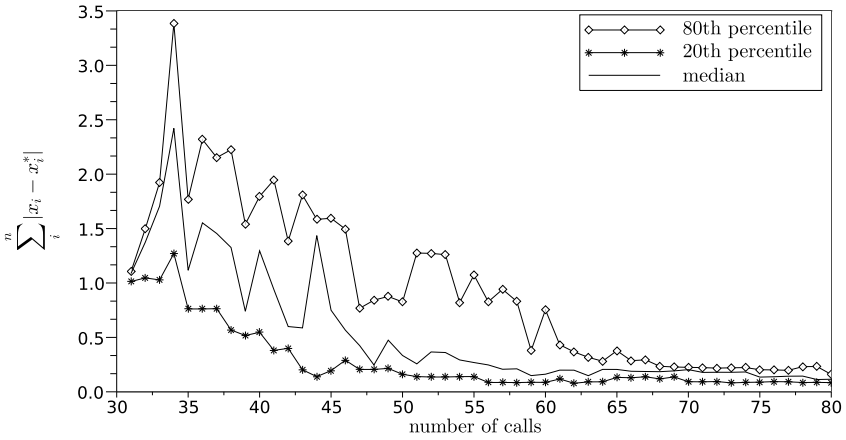


Figure 10: Median (solid), 80th (diamond) and 20th (star) percentiles for 6D test case using 10 independent runs with fixed initial DOE for all runs.

- for minimizing the variance $\mathbb{VAR}Z^{t+1}_{(x^{next})}(\omega)$,$t^{max} = 300(n+m)\lambda$ where population size $\lambda = 4 + \lfloor 3\log(n+m) \rfloor$ and σ_0 is one half of the domain.

Using such optimization strategy for 6 dimensional test case on 3.6 GHz CPU required about 30 to 200 seconds for hyperparameter estimation, 8 to 180 seconds for minimization of the mean of projected process, 8 to 30 seconds for the maximization of the EI and 17 to 50 seconds for the minimization of the variance at the next point of interest for every iteration of the proposed method. This strategy is not optimal and alternative study to explore the best optimization procedure for every sub-optimization would be needed.
In fig. 10 median, 80th and 20th percentiles are shown for 6D test case using 10 independent runs of the proposed method (EI-VAR) with fixed initial design (MSE optimized LH [1]) for all runs. It reflects the variation in the performance of the proposed method due to random sub-optimization procedures.

6.2.3 Comparison methods

To illustrate performance of the proposed method we compare it to the MC based optimization. We use EI and kriging model with noisy MC estimations of the objective values $\mathbb{E}_U(f(x,U))$. The precision (fidelity level) of the MC estimation of the objective is limited by the inverse of the number of calls k to the underlying simulator $f(x,u)$. For illustration and in order to keep the procedure as simple as possible, the k values are kept constant. In order to approximate the $\mathbb{E}_U(f(x,U))$ and filter the error of MC estimate, we use kriging model with Gaussian covariance kernel and homogeneous noise which is added on the diagonal of the covariance matrix. We assume that this noise is not known and together with the kernel hyper parameters is estimated by maximizing the Likelihood. To choose the next point for MC estimation we use maximum EI, where an improvement is defined as

$$I^t_x(\omega) = \left[f^t_{min} - Y^t_{(x)}(\omega) \right]^+,$$

and $Y^t_{(x)}(\omega)$ is a kriging model of the noisy MC estimate conditioned on the nosy observations $\mathbf{Y}$. The current minimum is the best noisy observation $f^t_{min} = min(\mathbf{Y})$.
For the maximization of the Likelihood and EI the CMA-ES is used.
The method is summarized in algorithm 3. Note that the kriging model is built in n dimensions as the objective $\mathbb{E}_U(f(x,U))$ is function only of x. The point which is chosen by EI leads to k MC calls to the simulator $f(x,u)$ at every iteration.

6.2.4 Experimental procedure

To compare the performance of the various methods we use the absolute distance between the optimum given by the method and the true minimum $|x - x^*|$ where $x^* = \arg\min_x \mathbb{E}_U[f(x,U)]$ versus number of calls to the simulator $f(x,u)$.
As we optimize stochastic functions we perform 10 runs of each method and use the average. The Random Latin Hypercube design (RLHD) was used as an initial DOE for all methods creating new initial design for every run. The

Algorithm 3 EI based on MC estimations of $\mathbb{E}_U(f(x,U))$ (EI-MC)

Create an initial DOE in the x space.
Estimate $\mathbb{E}_U(f(x,U))$ by k MC runs to the $f(x,u)$.
while stopping criterion not met **do**
  Create kriging model $Y^t_{(x)}(\omega)$ of the noisy simulator in the x space, by maximizing likelihood;
  Maximize $EI_Y(x))$ to choose x^{next};
  Estimate objective $\mathbb{E}_U(f(x^{next},U))$ by k MC runs to the simulator $f(x,u)$;
  Update DOE.
end while

number of points for initial DOE is $t = 5d$, where d is a number of dimensions ($d = n + m$ for the EI-VAR and EI-Sample and $d = n$ for MC based approaches EI-MC). When $d = 1$ the number of points for initial DOE $t = 3$ (2D test case for EI-MC).
A 50 iterations budget is used as a stopping criterion for EI-VAR and EI-Sample and 30 iterations budget for EI-MC.

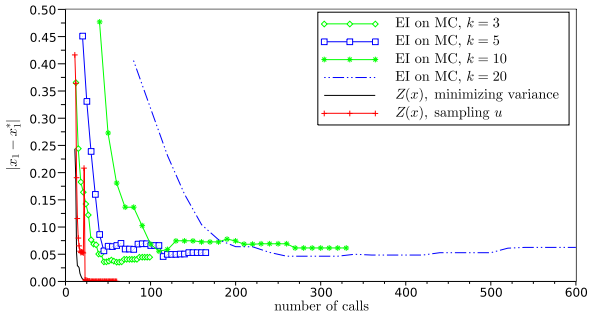
6.2.5 Results

The results are summarized in fig. 11, fig. 12 and fig. 13. The mean of the convergence for every dimension and total for all dimensions of x are given. The confidence intervals are illustrated using 20th and 80th percentile.
The main conclusion from the results of the test cases is that the convergence rates for the proposed approach (EI-VAR) are much better in comparison to the MC based optimization (EI-MC). The precision of the solutions of the MC based approaches is limited by the number of MC runs and in some cases (low number of MC runs k) did not converge to the true optimum (see for example fig. 13). For MC based approaches (EI-MC), where higher number of MC samples k where used (5,10,20) (thus providing more precise estimation of the objective), the proposed method (EI-VAR) has converged even before the initial DOE for EI-MC has been obtained. By definition EI-VAR is $O(k)$ times less expensive for iteration and it indicates the potential benefit of the proposed approach as usually MC methods require at least hundreds of runs per estimation. Notice also that the confidence intervals for the (EI-VAR and EI-Sample) are smaller compared to the MC based approach and tend to decrease with the number of sampled points.
The convergence of the simplified approach (EI-Sample) seems to be similar to the complete method (EI-VAR) for 2D and 4D test cases. However with more complex model as in 6D case, the results of simplified approach are much worse than the complete approach and may even be out performed by MC based optimization (this can be observed for the 3rd component of x fig. 13), therefore the complete approach (EI-VAR) is safer and preferable to the simplified one (EI-Sample).

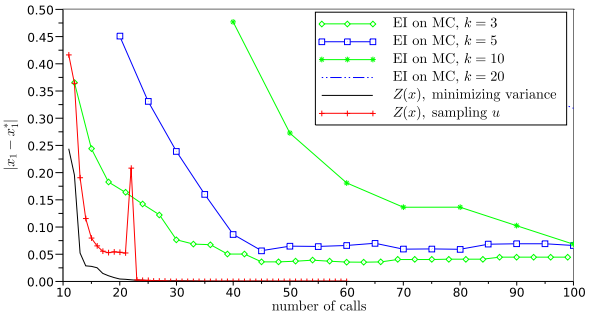
Finally in fig. 14 we show 20th and 80th percentile of convergence rates for 10 runs of the complete approach (EI-VAR) on the 6D test function for two cases: initial DOE is fixed MSE optimizd LH and initial DOE is RLH independently generated for every run. The fixed design leads to slightly better convergence and slightly less variance in the results. The results indicate that for the given test function the method is somehow robust to initial DOE and most of the variance in the results are due to the stochastic nature of the sub-optimization procedures.

7 Conclusion and Remarks

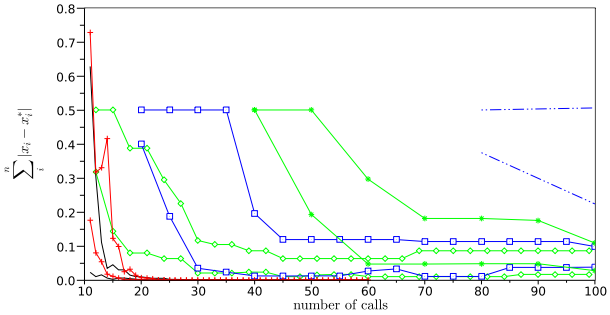
In this paper, a new robust optimization method was proposed for problems where some of the input parameters are random. The method minimizes the expected output of the simulator by, first, creating a kriging model in the joined space of deterministic and probabilistic input parameters. Then another "projected" Gaussian process is built to represent the mean output of the simulator. The maximization of the EI together with the minimization of the variance are used to define a sampling criterion for the optimization and simultaneous sampling in the uncertain parameter space.
The method is compared with simplified MC based optimization on several test cases and the results indicate significant enhancement in terms of precision and number of calls to the simulator. Another advantage of the proposed method is that one does not have to choose the number of the MC simulations. Furthermore the kriging model of the simulator and expectation measure are obtained, which may be used for other analyses. For example one can use these models to numerically obtain other statistical measures such as variance.
The potential limitations are associated with the fixed covariance kernel and noise models. The performance of the proposed method probably is dependant on the ability of the regression model Y to model the true simulator, therefore it may encounter problems when the Gaussian kernel is not good choice for the covariance function. The Gaussian covariance kernel yields very smooth (infinitely differentiable) trajectories. Such strong smoothness assumptions are



(a) Mean convergence results

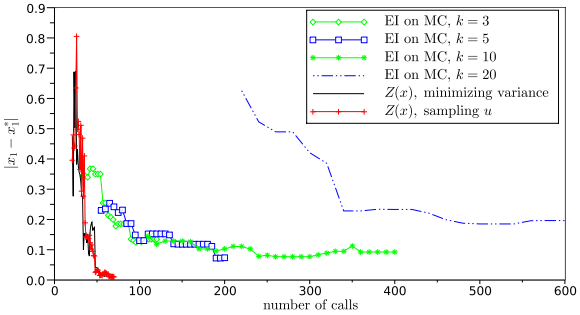


(b) Mean convergence results (zoomded in)

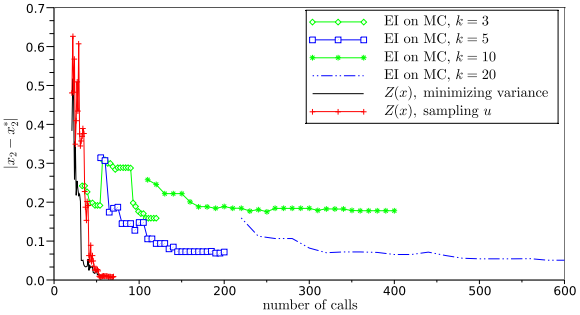


(c) 20th and 80th percentile

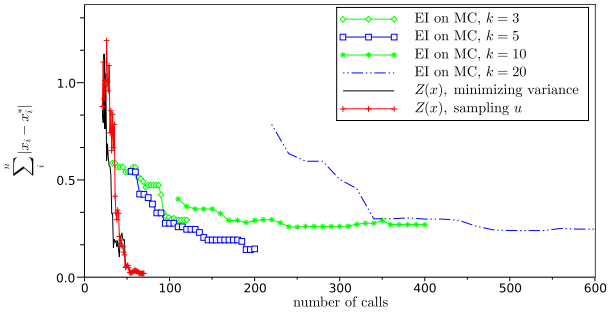
Figure 11: Convergence rates of different methods for 2D Michalewitz test function ($n = 1$, $m = 1$). The EI-MC with fixed nummber of runs $k = 3$ (diamond), $k = 5$ (square), $k = 10$ (star), $k = 20$ (dotted), complete method by minimizing variance (EI-VAR) (solid) and simplified approach by sampling u (EI-Sample) (cross).



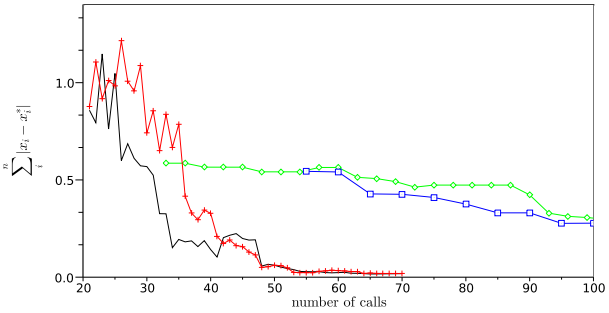
(a) Mean convergence for x_1



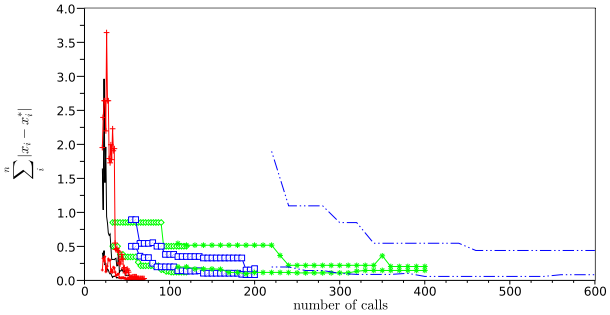
(b) Mean convergence for x_2



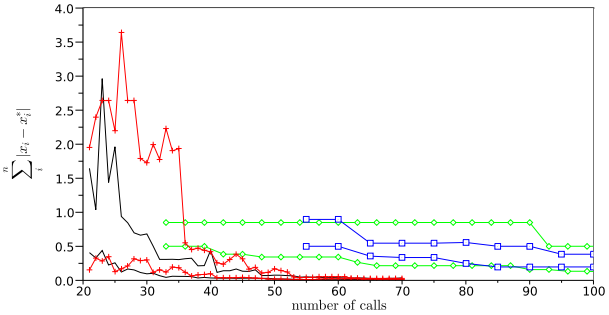
(c) Mean convergence for all dimensions



(d) Mean convergence for all dimensions (zoomed in)

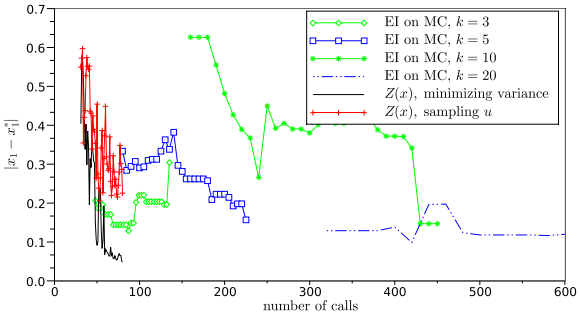


(e) 20th and 80th percentile of convergence

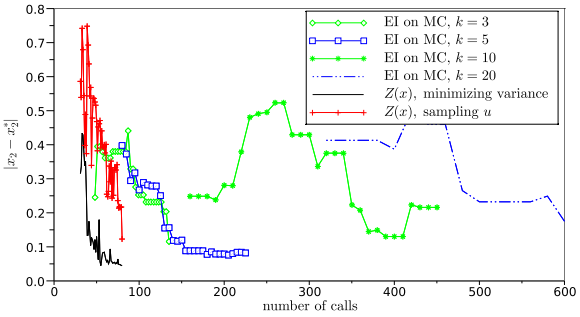


(f) 20th and 80th percentile of convergence (zoomed in)

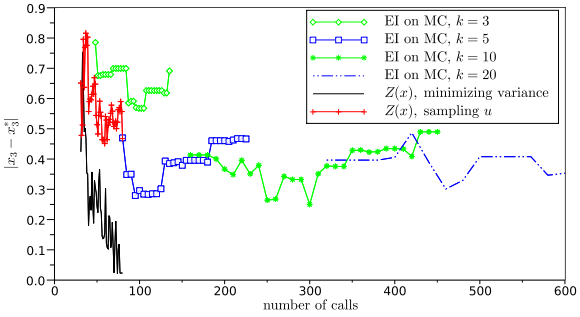
Figure 12: Convergence rates of different methods for 4D Michalewitz test function ($n = 2$, $m = 2$). The EI-MC with fixed nummber of runs $k = 3$ (diamond), $k = 5$ (square), $k = 10$ (star), $k = 20$ (dotted), complete method by minimizing variance (EI-VAR) (solid) and simplified approach by sampling u (EI-Sample) (cross).



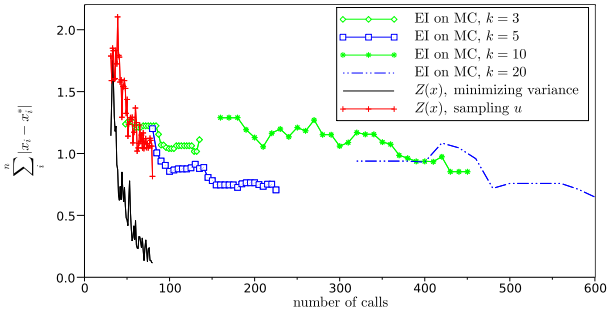
(a) Mean convergence for x_1



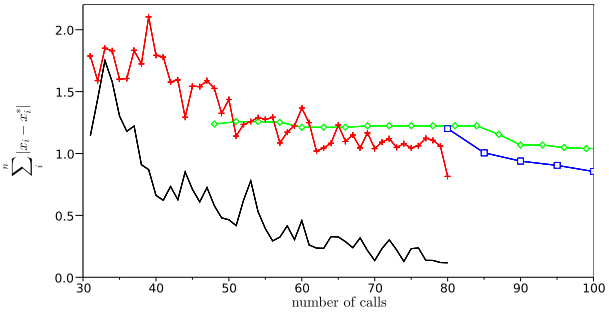
(b) Mean convergence for x_2



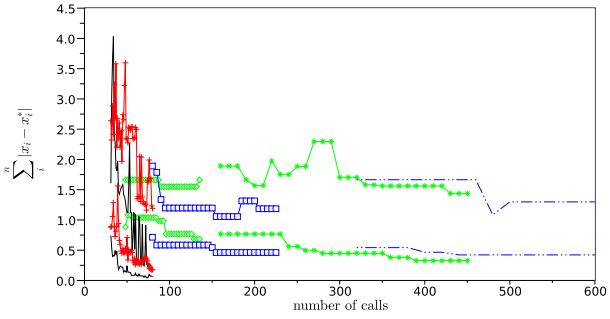
(c) Mean convergence for x_3



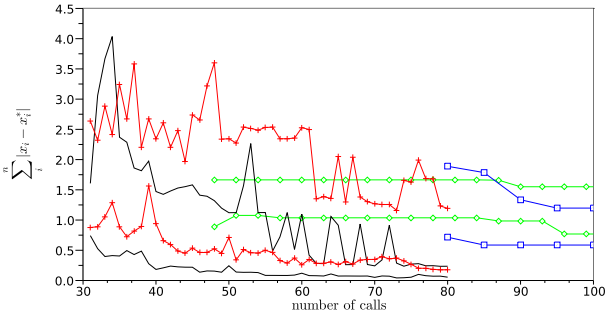
(d) Mean convergence for all dimensions



(e) Mean convergence for all dimensions (zoomed in)

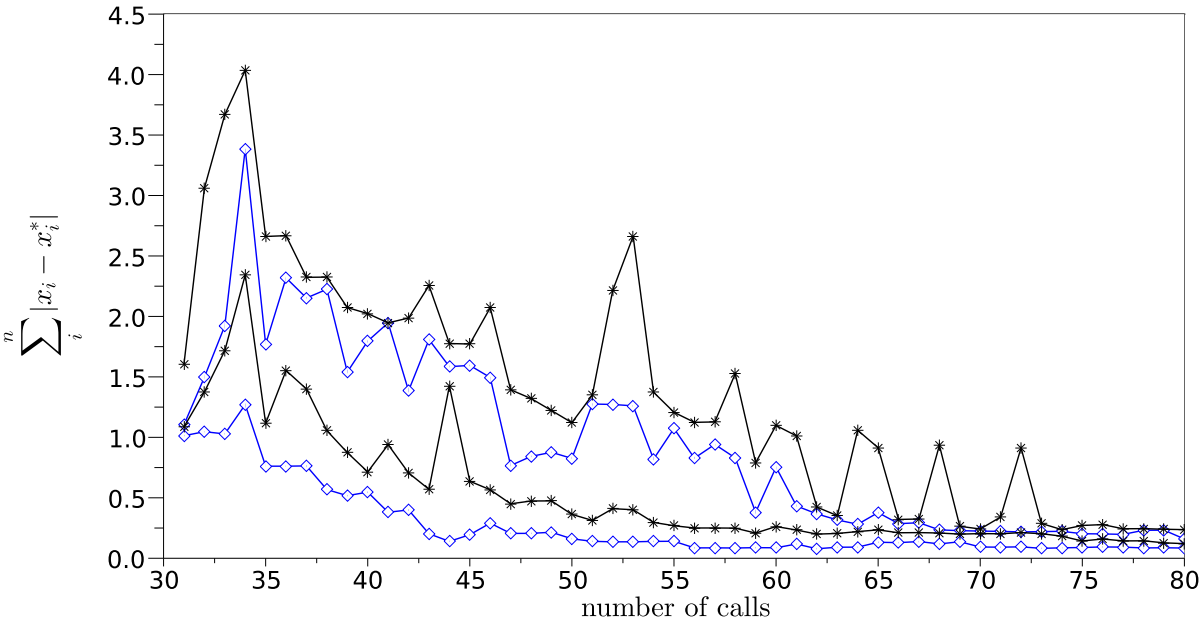


(f) 20th and 80th percentile of convergence



(g) 20th and 80th percentile of convergence (zoomed in)

Figure 13: Convergence rates of different methods for 6D Michalewitz test function ($n = 3$, $m = 3$). The EI-MC with fixed nummber of runs $k = 3$ (diamond), $k = 5$ (square), $k = 10$ (star), $k = 20$ (dotted), complete method by minimizing variance (EI-VAR) (solid) and simplified approach by sampling u (EI-Sample) (cross).



(a)

Figure 14: The convergence of the complete approach (EI-VAR). 20th and 80th percentile of 6D test case using 10 runs with fixed initial design (dimond) and generating new random LH for every run (star).

unrealistic for modelling many physical processes [35]. The application to other analytical and practical test problems as well as possibility of using other kernel functions and other noise models is a subject for further research.
From practical point of view the method may be applicable only to costly simulators, as it involves four global optimization subtasks. For effective implementation the study of efficient strategies to solve these subtasks would be needed.

References

[1] Janis Auzins. Direct optimization of experimental designs. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number AIAA Paper: 2004-4578, 2004.

[2] Huang Beiqing and Du Xiaoping. A robust design method using variable transformation and gauss-hermite integration. *International Journal for Numerical Methods in Engineering*, 66(12):1841–1858, 2006.

[3] Hans-Georg Beyer. Towards a theory of 'evolution strategies'. Some asymptotical results from the $(1, +\lambda)$-theory. *Evolutionary Computation*, 1:165–188, 1993.

[4] Hans-Georg Beyer and Bernhard Sendhoff. Functions with noise-induced multimodality: a test for evolutionary robust optimization-properties and performance analysis. *IEEE Trans. Evolutionary Computation*, 10(5):507–526, 2006.

[5] Hans-Georg Beyer and Bernhard Sendhoff. Robust optimization - a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, July 2007.

[6] Phillip Boyle. *Gaussian processes for regression and optimisation.* PhD thesis, 2007.

[7] Phillip Boyle and Marcus Frean. Dependent gaussian processes. In *Advances in Neural Information Processing Systems 17*, pages 217–224. MIT Press, 2005.

[8] Erick Cantú-Paz. Adaptive sampling for noisy problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004), volume 3103 of Lecture Notes in Computer Science*, pages 947–958. Springer, 2004.

[9] Xiaoping Du and Wei Chen. Towards a better understanding of modeling feasibility robustness in engineering design. *ASME Journal of Mechanical Design*, 122:385–394, 1999.

[10] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer, October 2008.

[11] Alexander I. Forrester, Neil W. Bressloff, and Andy J. Keane. Optimization using surrogate models and partially converged computational fluid dynamics simulations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 462(2071):2177–2204, July 2006.

[12] Alexander I. Forrester and Donald R. Jones. Global optimization of deceptive functions with sparse sampling. 2008.

[13] David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. In Yoel Tenne and Chi-Keong Goh, editors, *Computational Intelligence in Expensive Optimization Problems*, Springer series in Evolutionary Learning and Optimization, pages 131–162. springer, 08 2009.

[14] A. Girard, C. E. Rasmussen, J. Quiñonero-Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs - application to multiple-step ahead time series forecasting. In *Advances in Neural Information Processing Systems*, pages 529–536. MIT Press, 2003.

[15] Agathe Girard. *Approximate methods for propagation of uncertainty with gaussian process models.* PhD thesis, 2004.

[16] N. Hansen. The CMA evolution strategy: a comparing review. In J.A. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, editors, *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pages 75–102. Springer, 2006.

[17] Nikolaus Hansen, André S. P. Niederberger, Lino Guzzella, and Petros Koumoutsakos. A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. *Trans. Evol. Comp*, 13(1):180–197, 2009.

[18] D. Higdon. Space and space-time modeling using process convolutions. *Quantitative methods for current environmental issues*, page 37–56, 2002.

[19] D. Huang, T. Allen, W. Notz, and R. Miller. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32(5):369–382, November 2006.

[20] D. Huang, T. T. Allen, W. I. Notz, and N. Zeng. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization*, 34(3):441–466, 2006.

[21] Waltraud Huyer and Arnold Neumaier. SNOBFIT – stable noisy optimization by branch and fit. *ACM Trans. Math. Softw.*, 35(2):1–25, 2008.

[22] Ruichen Jin, Xiaoping Du, and Wei Chen. The use of metamodeling techniques for optimization under uncertainty. In *2001 ASME Design Automation Conference, Paper No. DAC21039*, pages 99–116, 2001.

[23] Yaochu Jin and J. Branke. Evolutionary optimization in uncertain environments-a survey. *Evolutionary Computation, IEEE Transactions on*, 9(3):303–317, 2005.

[24] Donald R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

[25] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[26] M. C. Kennedy and A. O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

[27] R. Garnett M. A. Osborne and S. J. Roberts. Gaussian processes for global optimization. In *3rd International Conference on Learning and Intelligent Optimization (LION3)*, 2009.

[28] S. F. Wojkiewicz T. Trucano M. S. Eldred, A. A. Giunta. Formulations for surrogate-based optimization under uncertainty. *9 th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (AIAA Paper 2002-5585).

[29] A. O'Hagan. Bayes-hermite quadrature. *Journal of Statistical Planning and Inference*, 29:245–260, 1991.

[30] Michael A Osborne, Alex Rogers, Sarvapali Ramchurn, Stephen J Roberts, and N. R. Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes. In *International Conference on Information Processing in Sensor Networks (IPSN 2008)*, pages 109–120, April 2008.

[31] C.J. Paciorek. *Nonstationary Gaussian Processes for Regression and Spatial Modelling*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2003.

[32] Gyung-Jin Park, Tae-Hee Lee, Hee Lee Kwon, and Kwang-Hyeon Hwang. Robust design : An overview. *AIAA journal*, 44(1):181–191, 2006.

[33] V. Picheny, D. Ginsbourger, and Y. Richet. Noisy expected improvement and on-line computation time allocation for the optimization of simulators with tunable fidelity. In *EngOpt 2010 - 2nd International Conference on Engineering Optimization*.

[34] J. Quiñonero-Candela, A. Girard, J. Larsen, and C. E. Rasmussen. Propagation of uncertainty in bayesian kernel models - application to multiple-step ahead forecasting. In *International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 701–704, Hong-Kong, apr 2003. IEEE.

[35] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, December 2005.

[36] Carl Edward Rasmussen and Zoubin Ghahramani. Bayesian monte carlo.

[37] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, November 1989.

[38] Nikolaos V. Sahinidis. Optimization under uncertainty: State-of-the-art and opportunities. *Computers and Chemical Engineering*, 28:971–983, 2004.

[39] Daniel Salazar, Rodolphe Le Riche, and Xavier Bay. An empirical study of the use of confidence levels in RBDO with Monte Carlo simulations. In *Multidisciplinary Design Optimization in Computational Mechanics*. Wiley, 11 2009.

[40] Michael James Sasena. *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD thesis, 2002.

[41] M. Seeger. *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*. PhD thesis, University of Edinburgh, July 2003.

[42] Alexandros Taflanidis. *Stochastic system design and applications to stochastically robust structural control*. PhD thesis, 2007.

[43] Julien Villemonteix, Emmanuel Vazquez, Maryan Sidorkiewicz, and Eric Walter. Global optimization of expensive-to-evaluate functions: an empirical comparison of two sampling criteria. *Journal of Global Optimization*, 43(2-3):373–389, 2009.

[44] Julien Villemonteix, Emmanuel Vázquez, and Eric Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *J. Global Optimization*, 44(4):509–534, 2009.

[45] Pingtao Yan and MengChu Zhou. A life cycle engineering approach to development of flexible manufacturing systems. *IEEE Transactions on Robotics*, 19(3):465–473, 2003.