Electronic Journal of Statistics Vol. 12 (2018) 4002–4031

Bayes Linear Analysis of Sequential Optimal Design Problems

Matthew Jones[1], Michael Goldstein[1], Philip Jonathan[2], and David Randell[2]

[1]Department of Mathematical Sciences, Durham University, Lower Mountjoy, Stockton Road, Durham, UK, DH1 3LE

[2]Statistics & Data Science, Shell Projects & Technology, Concord Business Park, Manchester, UK, M22 0RR

September 5, 2017

Abstract

In a statistical or physical model, it is often the case that a set of design inputs must be selected in order to perform an experiment to collect data with which to update beliefs about a set of model parameters; frequently, the model also depends on a set of external variables which are unknown before the experiment is carried out, but which cannot be controlled. Sequential optimal design problems are concerned with selecting these design inputs in stages (at different points in time), such that the chosen design is optimal with respect to the set of possible outcomes of all future experiments which might be carried out. Such problems are computationally expensive.

We consider the calculations which must be performed in order to solve a sequential design problem, and we propose a framework using Bayes linear emulators to approximate all difficult calculations which arise; these emulators are designed so that we can easily approximate expectations of the risk by integrating the emulator directly, and so that we can efficiently search the design input space for settings which may be optimal. We also consider how the structure of the design calculation can be exploited to improve the quality of the fitted emulators. Our framework is demonstrated through application to a simple linear modelling problem, and to a more complex airbourne sensing problem, in which a sequence of aircraft flight paths must be designed so as to collect data which are informative for the locations of ground-based gas sources.

Keywords: Bayesian uncertainty analysis; Emulation; Sequential experimental design; Decision support; Remote sensing; Uncertainty quantification

1 INTRODUCTION

Scientific research increasingly relies on the specification and analysis of models which are intended to recreate the properties of a particular natural or man-made system. These models can vary greatly in complexity: in some instances, model predictions for a system are simple and easy to evaluate (for example, the atmospheric dispersion model discussed in Pasquill (1971)), whereas in others, simply evaluating the model to generate a single prediction may be a time-consuming, non-trivial task (for example, the climate model considered in Williamson et al. (2013)). Despite the diversity of fields in which modelling is undertaken and the range of different complexity levels, modellers often share a number of common goals.

Frequently, one of these goals is to infer certain parameters of a model using data collected from the system that the model is designed to represent (see, for example, Kennedy and O'Hagan (2001)); these parameters may be of direct interest themselves, or they may be of interest simply because we wish to calibrate the model so that it makes better predictions for unobserved states of the system. Generally, the modeller may also control some of the model inputs governing the observation process (for example, spatial locations at which observations of the real climate are made). Additionally, there may be inputs to the model which cannot be controlled when making observations on the real system (for example, the wind conditions at the point at which observations on the real climate are made), but which must be accounted for when making inferences from the data.

Once data has been collected from the system, decisions about the system must be made using the information provided by the model specification and the observed data; subsequently, under particular outcomes, these decisions will have known consequences. The general framework for a Bayesian decision analysis is presented in detail by, for example, Smith (2010) and Lindley (1972); Randell et al. (2010) perform such an analysis for a model describing a large offshore structure, where maintenance decisions must be made about individual components whose characteristics have a complex covariance structure. Sometimes, the experiments can be performed sequentially; that is, there is the opportunity to perform a sequence of experiments to learn about the system, and the benefit that could be obtained by continuing to experiment must be weighed against the cost of doing so. DeGroot (1970) provides an introduction to sequential decision-making, and Williamson and Goldstein (2012) provide an example in which a climate model must be used to choose a sensible CO_2 abatement policy at the present time and at fixed points in the future.

Combining model and decision problem specifications, the question of design arises naturally: given that some of the model inputs governing the measurement process may be controlled, how should these be selected so as to maximise the expected benefit of the observations? For non-sequential decision problems, optimal experimental design choices for common, simple scenarios are reviewed in Chaloner and Verdinelli (1995), and more complex, non-linear problems are considered in Ford et al. (1989). If a more complex model or loss function is specified, or for sequential problems, there is usually no analytic solution to the design calculations, and the resulting problem usually presents a computational challenge; Muller et al. (2007) provide a simulation procedure for

sequential design problems with simple forward models, which works by discretizing the design space and sampling the possible experimental outcomes from the model.

In this article, we develop an approximation framework which provides decision support for the sequential design problem; our procedure is designed to be able to cope with problems that have large numbers of stages, as well as problems in which the system model is an expensive function that may only be evaluated at a handful of parameter settings. Any approximation to the sequential design calculations will introduce numerical uncertainty; the procedure that we present is designed to track this uncertainty throughout the calculation, allowing the user to make an informed decision about whether to select an experiment subject to these uncertainties, or to carry out further analysis which may reveal more about the risks involved. Jones et al. (2016) proposed a similar framework to handle non-sequential design problems for expensive models.

The remainder of this article is laid out as follows: in Section 2, we introduce a notation for the sequential design problem, and present the standard backward induction algorithm for its solution. Then, in Section 3, we propose a framework which approximates the backward induction calculation using Bayes linear emulators. In Section 4, we consider an application to a simple linear model, and in Section 5, we consider a more complex application in atmospheric dispersion modelling. In Section 6, we discuss our results and propose avenues for future research. Additional details regarding the framework and the examples are provided in the supplementary material; sections in the supplement are labelled S1, S2 etc.

2 SEQUENTIAL OPTIMAL DESIGN

In this section, we introduce a notation for the general problem, and set out the Bayesian optimal experimental design framework in full.

2.1 Problem Definition and Notation

The general problem is this: we hold prior beliefs about a system we are studying in the form of a model which is a function of a number of input parameters. For each setting of its inputs (within some allowed range), the model can be run to generate a prediction for a set of system attributes (which we will refer to as the model outputs). We wish to use data from the system to update beliefs about some model parameters, before using these updated beliefs to make decisions; sets of observations may be taken sequentially, and after each has been observed, we have the option of either measuring the next set, or using current beliefs to make an immediate decision.

In what follows, 'stage j' refers to the point in time at which $(j-1)$ sets of observations have been made, and where we must consider whether to take the j^{th} set. We assume that a maximum of n experiments can be performed, and denote the j^{th} set of available observations by $z_j = \{z_{j1}, \ldots, z_{jn_{z_j}}\}$, $j = 1 \ldots, n$. We denote

the collection of observations collected up to and including stage j by $z_{[j]} = \{z_1, \ldots, z_j\}$. We assume that the model inputs can be divided into three classes:

- **Model parameters**: these are the parameters about which we wish to learn. They are denoted by $q = \{q_1, \ldots, q_{n_q}\}$.
- **Design inputs**: we may select these, and they control the behaviour of the experiment which is performed. We denote the set of design inputs affecting the j^{th} observation by $d_j = \{d_{j1}, \ldots, d_{jn_{d_j}}\}$ (for $d_j \in \mathcal{D}_j$), and we denote the collection up to and including stage j by $d_{[j]} = \{d_1, \ldots, d_j\}$.
- **External inputs**: we cannot control these, but they affect predictions for the system, though we are not interested in using the data z_j to learn about them. We denote the set of external inputs affecting the j^{th} observation by $w_j = \{w_{j1}, \ldots, w_{jn_{w_j}}\}$ (for $w_j \in \mathcal{W}_j$), and we denote the collection up to and including stage j by $w_{[j]} = \{w_1, \ldots, w_j\}$.

At each stage j, we proceed as follows: first, the design inputs d_j must be selected; then, the external inputs w_j become known; finally, the experimental data z_j are observed. After collecting the experimental data, we must either make an immediate decision, with no possibility of further sampling, or select a design d_{j+1} for the next experiment before collecting these observations using this configuration.

2.1.1 Decision Problem

To determine the value of any set of observations, and therefore to choose between making an immediate decision and paying for another set of observations, we must specify the decision problem that we will solve using our beliefs about model parameters q after j sets of observations have been made. Within a probabilistic Bayesian framework, our beliefs about q after the j^{th} experiment are summarised through the posterior distribution

$$\begin{aligned} p\left(q|z_{[j]}, w_{[j]}, d_{[j]}\right) &= \frac{p\left(z_{[j]}|q, w_{[j]}, d_{[j]}\right) p\left(q|w_{[j]}, d_{[j]}\right)}{p\left(z_{[j]}|w_{[j]}, d_{[j]}\right)} \\ &= \frac{p\left(z_{[j]}|q, w_{[j]}, d_{[j]}\right) p\left(q\right)}{p\left(z_{[j]}|w_{[j]}, d_{[j]}\right)} \end{aligned} \tag{1}$$

where we specify the conditional distribution $p\left(z_{[j]}|q, w_{[j]}, d_{[j]}\right)$ for the observations z_j given the model parameters, and $p\left(q\right)$ specifies our prior beliefs about q, which we assume do not depend on $\{w_{[j]}, d_{[j]}\}$.
For the decision problem, we specify the following components:

- a space $\mathcal{A}_j$ of possible actions $a_j = \{a_{j1}, \ldots, a_{jn_{a_j}}\}$ which might be taken at stage j.
- a loss function $L_j\left(a_j, q\right)$ which describes (in utility units) the cost of taking action a_j at stage j (having terminated sampling) and then realising model parameters q.

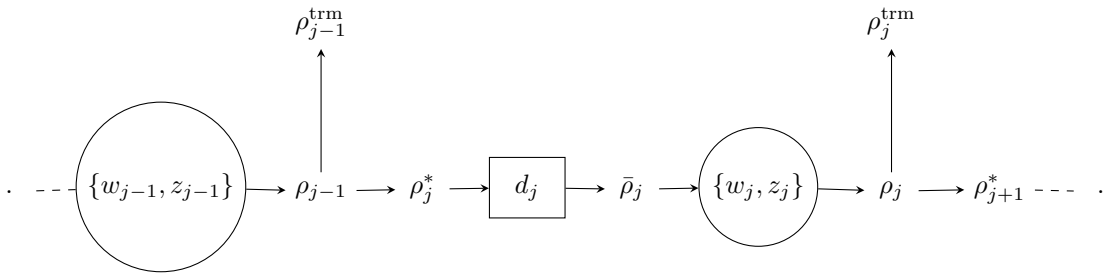


Figure 1: Graphical representation of the design procedure, showing quantities in the order that we choose, observe or compute them. Square nodes represent design parameters which we select, circular nodes represent random quantities which we observe when experimenting, and non-bordered nodes are risks which we compute.

- a function $c_j(d_j)$ which describes the cost (in utility units) of selecting design d_j for the experiment at stage j.

Based on this specification, we can evaluate our risk (expected loss) from making an immediate decision at stage j which is optimal against our current beliefs, as described by (1)

$$\rho_j^{\text{trm}}\left[z_{[j]}, w_{[j]}, d_{[j]}\right] = \min_{a_j \in \mathcal{A}_j} \int L_j(a_j, q)\, p\left(q | z_{[j]}, w_{[j]}, d_{[j]}\right) dq \tag{2}$$

where ρ_j^{trm} is referred to as the 'terminal risk' or the 'risk from an optimal terminal decision' at stage j. We denote the optimal decision at stage j, which minimises (2), by a_j^*.

We would now like to know: given our prior beliefs about the system, the costs of the observations which might be made, and the consequences of the decisions which might be taken, how many sets of observations should be collected, and at what settings of the design parameters? This is a problem in Bayesian sequential optimal experimental design, which can be solved using backward induction.

2.2 Extensive Form- Backward Induction

The backward induction algorithm is a well-known technique for solving decision and design problems; for an introduction to the algorithm, see, for example, DeGroot (1970). The algorithm begins at the final stage of the problem, where it is not possible to collect further observations, and then works back through the stages, deciding for each setting of the model inputs whether it is optimal to continue experimenting or to stop and make an immediate decision.

We iterate the following steps for $j = n, (n-1), \ldots 1$:

- We compute the overall risk, denoted by ρ_j, assuming that we act optimally at all future stages. At the final stage ($j = n$), no further observations are possible, and so the overall risk is equal to the terminal risk

$$\rho_n\left[z_{[n]}, w_{[n]}, d_{[n]}\right] = \rho_n^{\text{trm}}\left[z_{[n]}, w_{[n]}, d_{[n]}\right] \tag{3}$$

For all other stages ($1 \leq j < n$), we compare the risk from an immediate decision with that from future experimentation

$$\rho_j \left[z_{[j]}, w_{[j]}, d_{[j]}\right] = \min \left[\rho_j^{\text{trm}} \left[z_{[j]}, w_{[j]}, d_{[j]}\right], \rho_{j+1}^{*} \left[z_{[j]}, w_{[j]}, d_{[j]}\right]\right] \tag{4}$$

The risk ρ_{j+1}^{*} from optimal future experimentation is the output from the $(j+1)^{\text{th}}$ iteration of this procedure, defined in (6)

- At the point when we must choose between an immediate decision and further experimentation, the experimental outcomes $\{z_j, w_j\}$ are unknown, and so we compute the expectation $\bar{\rho}_j$ of the risk ρ_j with respect to our current beliefs

$$\begin{aligned} \bar{\rho}_j \left[z_{[j-1]}, w_{[j-1]}, d_{[j]}\right] &= \iint \rho_j \left[z_{[j]}, w_{[j]}, d_{[j]}\right] p\left(z_j, w_j | z_{[j-1]}, w_{[j-1]}, d_{[j]}\right) dz_j dw_j \\ &= \iint \rho_j \left[z_{[j]}, w_{[j]}, d_{[j]}\right] p\left(z_j | z_{[j-1]}, w_{[j]}, d_{[j]}\right) p\left(w_j | w_{[j-1]}, d_{[j]}\right) dz_j dw_j \end{aligned} \tag{5}$$

where in the second line, we have relied upon the assumption that we do not use the z_j to learn about the w_j.

- We find the optimal design d_j^{*} for the j^{th} experiment, as a function of the risk inputs $\{z_{[j-1]}, w_{[j-1]}, d_{[j-1]}\}$ at the previous stages, by minimising $\bar{\rho}_j$ over d_j, taking account of the cost $c_j\left(d_j\right)$ of experimentation

$$d_j^{*} = \underset{d_j \in \mathcal{D}_j}{\arg\min} \left[\bar{\rho}_j \left[z_{[j-1]}, w_{[j-1]}, d_{[j]}\right] + c_j\left(d_j\right)\right]$$

The minimum risk ρ_j^{*} is then

$$\rho_j^{*} \left[z_{[j-1]}, w_{[j-1]}, d_{[j-1]}\right] = \bar{\rho}_j \left[z_{[j-1]}, w_{[j-1]}, \{d_{[j-1]}, d_j^{*}\}\right] + c_j\left(d_j^{*}\right) \tag{6}$$

Note that when $j = 1$, we define $d_{[0]} = w_{[0]} = z_{[0]} = \emptyset$. A graphical representation of the relationship between the designs, observables and risks is provided in Figure 1; the backward induction procedure is written in pseudo-code in algorithm 1.

If we can perform these calculations, then the output from this algorithm is the optimal design d_1^{*} for the experiment at the first stage, and the corresponding optimal risk ρ_1^{*}. To decide how to proceed, we compare this risk (from an optimal future procedure) with the risk ρ_0^{trm} from making an optimal decision under our prior beliefs (before experimentation). If $\rho_1^{*} < \rho_0^{\text{trm}}$, then it is optimal to perform the first experiment (at d_1^{*}), and then to assess the benefit of the second experiment against an immediate decision under our beliefs after the first experiment; otherwise, it is optimal to make an immediate decision and to cease sampling. More generally, if we have collected data up to the k^{th} experiment, then we assess the optimal course of action by comparing

ρ^*_{k+1} with ρ^{trm}_k.

While the calculations (2), (4), (5) and (6) are simple to express, they generally represent a large computational challenge. It is not uncommon to find a problem in which the terminal risk (2) cannot be computed without recourse to numerical integration and optimisation methods; the intractability of this calculation in turn rules out a closed-form expression at any of the other steps of the procedure. Even in the situation where this calculation can be performed directly, numerical methods will generally be required by the time we must perform either (5) or (6). For previous discussions of the computational challenges involved in sequential decision and design problems, see, for example, Muller et al. (2007) or Williamson and Goldstein (2012); simple, discrete examples in which the backward induction can be performed exactly are discussed in Smith (2010) and Berger (1980).

If there is a computationally feasible way to approximate these calculations, however, the potential gains are large: designing observations that we make now to take into account their effect on data that we might collect in the future improves the overall quality of the information that we can collect, and the backward induction framework introduces the possibility of stopping once we have enough information to perform the task at hand, thus potentially saving the cost of unnecessary observations. A sequential procedure is guaranteed to have a risk which is no greater than that of the procedure in which we simply collect all of the observations before making a decision (DeGroot, 1970), and the potential benefits from such a procedure may be large. As discussed by Huan and Marzouk (2016), common strategies for approximating the full sequential design calculation where this is computationally infeasible include 'batch' design, in which designs for the experiments are all selected upfront, and 'greedy' design, in which the optimal design at each stage is selected without considering the possibility of further experiments. Both strategies may lead to the selection of sub-optimal designs, as they do not account for information which may be available from data collected in the future.

Previous work on approximating sequential design problems is presented by Huan and Marzouk (2016), who formulate the problem as a general dynamic programming procedure, and then use an iterative procedure based on linear regression models to approximate the design calculations. Their approximation uses a 'one-step lookahead' approximation to the full procedure, in which, at stage j, only the results of the j^{th} and $(j+1)^{\text{th}}$ experiments are considered. The procedure that we present in Section 3 is based on the backward induction algorithm 1, which provides the basis for a more natural approximation, since the information from all future experiments $n, (n-1), \ldots, (j+1)$ is accounted for in the risk function ρ_j at stage j. In addition, we use a more flexible class of models to approximate the risk functions at each stage, and outline a strategy for choosing basis and covariance functions which approximate the risks well, while retaining tractability.

3 APPROXIMATION OF THE DESIGN CALCULATION

In this section, we detail the procedure that we will use to approximate the general algorithm described in section 2.2. It is designed to follow the backward induction procedure as closely as possible, using Bayes

Algorithm 1 Backward induction algorithm for sequential design problems

for $k = 0, \ldots, (n-1)$ **do**
  for $j = n, (n-1), \ldots, (k+1)$ **do**
    Compute the risk
    if $j = n$ **then**

$$\rho_n\left[z_{[n]}, w_{[n]}, d_{[n]}\right] = \rho_n^{\text{trm}}\left[z_{[n]}, w_{[n]}, d_{[n]}\right]$$

    else

$$\rho_j\left[z_{[j]}, w_{[j]}, d_{[j]}\right] = \min\left[\rho_j^{\text{trm}}\left[z_{[j]}, w_{[j]}, d_{[j]}\right], \rho_{j+1}^{*}\left[z_{[j]}, w_{[j]}, d_{[j]}\right]\right]$$

    end if
    Compute the expected risk

$$\bar{\rho}_j\left[z_{[j-1]}, w_{[j-1]}, d_{[j]}\right] = \iint \rho_j p\left(z_j | z_{[j-1]}, w_{[j]}, d_{[j]}\right) p\left(w_j | w_{[j-1]}, d_{[j]}\right) dz_j dw_j$$

    Compute the optimal design, and corresponding risk

$$\rho_j^{*}\left[z_{[j-1]}, w_{[j-1]}, d_{[j-1]}\right] = \min_{d_j \in \mathcal{D}}\left[\bar{\rho}_j\left[z_{[j-1]}, w_{[j-1]}, d_{[j]}\right] + c_j\left(d_j\right)\right]$$

  end for
  if $\rho_k^{\text{trm}} \leq \rho_{k+1}^{*}$ **then**
    Cease sampling, and take decision a_k^{*}.
    Break
  else
    Observe $\{z_{k+1}, w_{k+1}\}$ at d_{k+1}^{*}.
  end if
end for

linear emulators to approximate calculations which cannot be performed analytically. The analysis proceeds in waves, in a similar way to the history matching procedure of Vernon et al. (2010) and the non-sequential design procedure of Jones et al. (2016): at the first wave, we model the backward induction calculations over the whole design space, and we search the model input space to rule out designs which are unlikely to be optimal; then, in subsequent waves, we re-fit our risk models in those parts of the design space which have not yet been ruled out, allowing us to build up a more accurate picture of the behaviour of the risk and the structure of the design space in these regions.

In Section 3.1, we provide an overview of our approximation procedure; each step is explained more fully in Sections 3.2 to 3.6. Further detail is provided in Section S2 of the supplementary material.

3.1 Overview of the Procedure

Throughout the remainder of the article, we use a superscript (i), $i = 1, 2, \ldots$ to index the current wave of the algorithm. The steps that we perform for each wave are the same, but the approximating emulators are re-focused on those parts of the design space which we believe may contain the optimal design; this point is discussed further in section S2.1. At each wave $i = 1, 2, \ldots$, we iterate the following steps for $j = n, (n-1), \ldots, 1$:

Emulate the risk We fit a model which approximates the risk surface at stage j. Our model for the risk ρ_j (defined in (4)) is denoted by $r_j^{(i)}$, and consists of a regression surface and a residual component

$$r_j^{(i)}\left[z_{[j]}, w_{[j]}, d_{[j]}\right] = \sum_p \beta_{jp}^{(i)} g_{jp}^{(i)}\left(z_{[j]}, w_{[j]}, d_{[j]}\right) + u_j^{(i)}\left(z_{[j]}, w_{[j]}, d_{[j]}\right) \tag{7}$$

where $g_j^{(i)} = \{g_{j1}^{(i)}, \ldots, g_{jn_{g^{(i)}}}^{(i)}\}$ is a known set of basis functions, the $\beta_{jp}^{(i)}$ are corresponding unknown weights, and $u_j^{(i)}$ is a zero-mean correlated residual process. We specify our prior beliefs about the uncertain components of the model, and combine these prior beliefs with a set of evaluations of the risk to create a second-order emulator. The details of this procedure are discussed further in Section 3.2; an introduction to Bayes linear methods and to second-order emulation is provided in Section S1.

Compute the expected risk We derive a model for the expected risk surface at stage j by integrating our model for the risk. Our model $\bar{r}_j^{(i)}$ for the expected risk $\bar{\rho}_j$ (defined in (5)) is

$$\bar{r}_j^{(i)}\left[z_{[j-1]}, w_{[j-1]}, d_{[j]}\right] = \iint r_j^{(i)} p\left(z_j | z_{[j-1]}, w_{[j]}, d_{[j]}\right) p\left(w_j | w_{[j-1]}, d_{[j]}\right) dz_j dw_j \tag{8}$$

The characterisation of beliefs about $\bar{r}_j^{(i)}$ is discussed in Section 3.3.

Characterise the candidate design space We eliminate parts of the design space which we deem unlikely to be optimal. Our approximation to the optimal risk ρ_j^* (defined in (6)) is denoted by $s_j^{(i)}$, with

$$s_j^{(i)}\left[z_{[j-1]}, w_{[j-1]}, d_{[j-1]}\right] = \bar{r}_j^{(i)}\left[z_{[j-1]}, w_{[j-1]}, \{d_{[j-1]}, d_j^*\}\right] + c_j\left(d_j^*\right) \tag{9}$$

The value of d_j^* is unknown; we represent our uncertainty about the optimal design setting by sampling candidate designs $\tilde{d}_j$ from within a candidate design space $\mathcal{D}_j^{(i)}$ which could plausibly contain the optimal design. Our strategy for characterising this space and for characterising our uncertainty about $s_j^{(i)}$ is discussed in Section 3.4.

3.2 Modelling the Risk

When stage j of the algorithm is reached, the risk ρ_j (equation (4)) is unknown, so our first task is to fit a model as an approximation. We choose to use a second-order emulator, as this is a flexible model which will simplify the calculations that we need to perform in sections 3.3 and 3.4. An introduction to second-order emulation is provided in Section S1.2.

The general form of the model that we use is given in equation (7). To fit the emulator, we begin by selecting the regression basis functions $g_{jp}^{(i)}$, and by making a prior specification for the $\beta_{jp}^{(i)}$ and $u_j^{(i)}$; we specify expectations $\mathrm{E}\left[\beta_{jp}^{(i)}\right]$ for each of the regression coefficients, covariances $\mathrm{Cov}\left[\beta_{jp}^{(i)}, \beta_{kq}^{(i)}\right]$ between pairs of coefficients, and

Algorithm 2 Approximation to the backward induction procedure.

1: **for** $i = 1, 2, \ldots$ **do**
2: **for** $j = n, (n-1), \ldots, 1$ **do**
3: Specify risk model

$$r_j^{(i)}\left[z_{[j]}, w_{[j]}, d_{[j]}\right] = \sum_p \beta_{jp}^{(i)} g_{jp}^{(i)}\left(z_{[j]}, w_{[j]}, d_{[j]}\right) + u_j^{(i)}\left(z_{[j]}, w_{[j]}, d_{[j]}\right)$$

4: Approximate the expected risk

$$\bar{r}_j^{(i)}\left[z_{[j-1]}, w_{[j-1]}, d_{[j]}\right] = \iint r_j^{(i)} p\left(z_j | z_{[j-1]}, w_{[j]}, d_{[j]}\right) p\left(w_j | w_{[j-1]}, d_{[j]}\right) dz_j dw_j$$

5: Characterise the minimum risk

$$s_j^{(i)}\left[z_{[j-1]}, w_{[j-1]}, d_{[j-1]}\right] = \bar{r}_j^{(i)}\left[z_{[j-1]}, w_{[j-1]}, \{d_{[j-1]}, d_j^*\}\right] + c_j\left(d_j^*\right)$$

6: **end for**
7: **end for**

covariances $\mathrm{Cov}\left[u_j^{(i)}(.), u_j^{(i)}(.')\right]$ between pairs of residual function evaluations. Strategies for selecting the basis functions and making an appropriate prior specification are discussed in section S2.1.

In order to fit the emulator, we generate evaluations of the risk function. At wave i, we denote the set of $N_j^{(i)}$ risk values that we use to fit the model by $R_j^{(i)} = \{R_{j1}^{(i)}, R_{j2}^{(i)}, \ldots, R_{jN_j^{(i)}}^{(i)}\}$; $R_{jk}^{(i)}$ is the k^{th} evaluation of the risk obtained at input setting $\{z_{[j]k}, w_{[j]k}, d_{[j]k}\}$. At the final stage ($j = n$), $R_{jk}^{(i)}$ is an evaluation of the terminal risk ρ_n^{trm} (corresponding to the definition (3))

$$R_{nk}^{(i)} = \rho_n^{\text{trm}}\left[z_{[n]k}, w_{[n]k}, d_{[n]k}\right] \tag{10}$$

For all other stages ($j < n$), $R_{jk}^{(i)}$ is generated by comparing the terminal risk ρ_j^{trm} with $s_{j+1}^{(i)}$, our approximation to the risk from an optimal decision at stage $(j+1)$ (corresponding to the definition (4))

$$R_{jk}^{(i)} = \min\left[\rho_j^{\text{trm}}\left[z_{[j]k}, w_{[j]k}, d_{[j]k}\right], s_{j+1}^{(i)}\left[z_{[j]k}, w_{[j]k}, d_{[j]k}\right]\right] \tag{11}$$

where the characterisation of $s_{j+1}^{(i)}$ is discussed in section 3.4.

Due to uncertainty introduced through approximations to the risk, we are generally not able to evaluate the $R_{jk}^{(i)}$ exactly; instead, we assess $\mathrm{E}\left[R_{jk}^{(i)}\right]$ and $\mathrm{Cov}\left[R_{jk}^{(i)}, R_{jl}^{(i)}\right]$ by sampling, and we fit the emulator to the mean values, using the covariances to characterise the measurement error structure. This issue is discussed further in Section S2.1. Once the characteristics of the risk evaluations have been assessed, we can compute adjusted expectations $\mathrm{E}_{R_j^{(i)}}\left[r_j^{(i)}\left[z_{[j]}, w_{[j]}, d_{[j]}\right]\right]$ and covariances $\mathrm{Cov}_{R_j^{(i)}}\left[r_j^{(i)}\left[z_{[j]}, w_{[j]}, d_{[j]}\right], r_j^{(i)}\left[z'_{[j]}, w'_{[j]}, d'_{[j]}\right]\right]$ for any new input settings, as detailed in Section S1.2.1.

3.3 Approximating the Expected Risk

We use our model $r_j^{(i)}$ to compute an approximation $\bar{r}_j^{(i)}$ to the expected risk $\bar{\rho}_j$ (defined in equation (8)). As outlined in, for example, O'Hagan (1991) and Rasmussen and Ghahramani (2002), the characteristics of the expectation of a stochastic process can be derived by integrating the characteristics of the process directly; in this instance, the expectation of $\bar{r}_j^{(i)}$ is

$$\mathrm{E}_{R_j^{(i)}}\left[\bar{r}_j^{(i)}\left[z_{[j-1]}, w_{[j-1]}, d_{[j]}\right]\right] = \int \mathrm{E}_{R_j^{(i)}}\left[r_j^{(i)}\left[.\right]\right] p\left(z_j | z_{[j-1]}, w_{[j]}, d_{[j]}\right) p\left(w_j | w_{[j-1]}, d_{[j]}\right) dz_j dw_j$$

where $\mathrm{E}_{R_j^{(i)}}\left[r_j^{(i)}\right]$ is our adjusted expectation for $r_j^{(i)}$ (computed in Section 3.2), and the covariance between $\bar{r}_j^{(i)}$ values at any new pair of input settings is

$$\mathrm{Cov}_{R_j^{(i)}}\left[\bar{r}_j^{(i)}\left[z_{[j-1]}, w_{[j-1]}, d_{[j]}\right], \bar{r}_j^{(i)}\left[z'_{[j-1]}, w'_{[j-1]}, d'_{[j]}\right]\right] = \int \mathrm{Cov}_{R_j^{(i)}}\left[r_j^{(i)}\left[.\right], r_j^{(i)}\left[.'\right]\right] p\left(z_j, w_j | z_{[j-1]}, w_{[j-1]}, d_{[j]}\right) p\left(z'_j, w'_j | z'_{[j-1]}, w'_{[j-1]}, d'_{[j]}\right) dz_j dw_j dz'_j dw'_j$$

where $\mathrm{Cov}_{R_j^{(i)}}\left[r_j^{(i)}\left[.\right], r_j^{(i)}\left[.'\right]\right]$ is our adjusted covariance for the risk $r_j^{(i)}$. These calculations are performed for a general emulator in Section S1.2. In order to evaluate the above expressions, we must be able to compute expectations of both the basis and covariance functions with respect to the distributions $p\left(z_j | \dots\right)$ and $p\left(w_j | \dots\right)$; in practice, this either means that we must choose particular types of covariance functions and probability distributions in order to ensure integrability of the product, or that we must numerically compute the required integrals. This point is discussed further in section S2.2.

3.4 Characterising the Candidate Design Space

We now characterise our approximation $s_j^{(i)}$ (equation (9)) to the risk ρ_j^* from an optimal design at stage j, which will then be used as an input to the $(j-1)^{\text{th}}$ stage of the algorithm (equation (7), Section 3.2), or to select a design for the j^{th} experiment (Section 3.5). We do this using a sampling procedure, which interrogates our fitted emulator at a space-filling set of trial design inputs, and then selects the design which minimises the risk over this trial design set. Designs which are selected in this manner are referred to as 'candidate designs' and denoted by $\tilde{d}_j$, and the subset of the full design space identified through this procedure is referred to as the 'candidate design space' and is denoted by $\mathcal{D}_j^{(i)}$ (where $\mathcal{D}_j^{(0)} = \mathcal{D}_j$, the full design space). The sampling procedure used to identify candidate designs is outlined in algorithm 3.

Using this sampling procedure, our uncertainty about $s_j^{(i)}$ is characterised by evaluating the moments of $\bar{r}_j^{(i)}$ at sampled values of $\tilde{d}_j$; the computation of $\mathrm{E}\left[s_j^{(i)}\left[.\right]\right]$ and $\mathrm{Cov}\left[s_j^{(i)}\left[.\right], s_j^{(i)}\left[.'\right]\right]$ is discussed in Section S2.3. As discussed in e.g. Hennig and Schuler (2012), Adler (1981) and Jones et al. (1998), exactly characterising the extrema of stochastic processes is a challenging and open problem. The approach adopted here efficiently

generates a conservative estimate of our uncertainty about the minimum.

Algorithm 3 Sample a candidate design $\tilde{d}_j$ at stage j, wave i.

1: Generate $M_j^{(i)}$ space-filling trial designs $\{d_{j1}, d_{j2}, \ldots, d_{jM_j^{(i)}}\}$ within the candidate space $\mathcal{D}_j^{(i-1)}$

2: Jointly sample $\bar{r}_j^{(i)}\left[z_{[j-1]}, w_{[j-1]}, \{d_{[j-1]}, d_{jk}\}\right]$ over the set of all trial designs d_{jk} from a Gaussian distribution

3: Set

$$\tilde{d}_j = \arg\min_{d_{jk}} \left[\bar{r}_j^{(i)}\left[z_{[j-1]}, w_{[j-1]}, \{d_{[j-1]}, d_{jk}\}\right] + c_j\left(d_{jk}\right)\right]$$

3.5 Stopping

Assume that, at wave i, the procedure has been run back to stage $j = k$, and that we have already selected $d_{[k-1]}$ and observed $\{z_{[k-1]}, w_{[k-1]}\}$; based on our beliefs $\bar{r}_k^{(i)}$ about the risk at this point, we must choose between an immediate decision based on our current beliefs $p\left(q|z_{[k-1]}, w_{[k-1]}, d_{[k-1]}\right)$ about the model parameters, and an experiment at stage k using a design which we believe may be optimal. If we knew the risk function exactly, this choice would be simple: we would choose to experiment if $\rho_k^* < \rho_{k-1}^{\text{trm}}$, and otherwise make an immediate decision a_{k-1}^* (see Section 2.2 and algorithm 1). However, since the true risks are unknown, we must instead make a choice which takes account of our uncertainties about them. First, we discuss the selection of a design at which we would perform the experiment; then we consider what we should do given this choice of design; lastly, we consider the benefit that we may obtain from running further waves of the procedure.

Choosing a design First, we must choose a design for the k^{th} experiment. We denote the chosen design by $\hat{d}_k$, and select $\hat{d}_k$ to minimise our expected risk

$$\hat{d}_k = \arg\min_{d_k} \left[\mathrm{E}_{R_k^{(i)}}\left[\bar{r}_k^{(i)}\left[d_k\right]\right] + c_k\left(d_k\right)\right]$$

This minimum is identified either through use of a suitable numerical optimisation procedure, or through interrogation of the mean surface at a large, space-filling set of design inputs.

Choosing a course of action Having fixed $\hat{d}_k$, we must now determine whether we believe that this experiment should be carried out; we do this based on a comparison of our expectation for the risk from this experiment with the risk from an immediate decision. If

$$\mathrm{E}_{R_k^{(i)}}\left[\bar{r}_k^{(i)}\left[\hat{d}_k\right]\right] + c_k\left(\hat{d}_k\right) < \rho_{k-1}^{\text{trm}}$$

then we choose to carry out the k^{th} experiment at this design setting; otherwise, we opt for an immediate decision based on our beliefs $p\left(q|z_{[k-1]}, w_{[k-1]}, d_{[k-1]}\right)$.

Assessing the value of further waves Having fully determined our course of action based upon our current beliefs about the risk function, we also wish to make a judgement about the value of running further waves of the approximation procedure (algorithm 2) to learn more about the risk. To help us make this judgement, we can compute the expected value of perfect information (EVPI) about the risk. If we knew the risk function $\bar{r}_k^{(i)}$ exactly, and we also knew the optimal design setting d_k^*, then the risk from an optimal course of action would be

$$\min \left[\rho_{k-1}^{\text{trm}},\ \bar{r}_k^{(i)} \left[d_k^* \right] + c_k \left(d_k^* \right) \right]$$

Suppose that, after having chosen some $\hat{d}_k$ as the design setting for the next experiment, we discover that d_k^* is in fact the true optimal design for the k^{th} experiment; in this situation, our expectation for the loss incurred by choosing to experiment at $\hat{d}_k$ rather than at d_k^* is

$$\begin{aligned} v_{k-1}^{(i)} = \min & \left[\rho_{k-1}^{\text{trm}}, \mathrm{E} \left[\bar{r}_k^{(i)} \left[\hat{d}_k \right] \right] + c_k \left(\hat{d}_k \right) \right] \\ & - \mathrm{E} \left[\min \left[\rho_{k-1}^{\text{trm}},\ \bar{r}_k^{(i)} \left[d_k^* \right] + c_k \left(d_k^* \right) \right] \right] \end{aligned}$$

The expectation of the second term is approximated by sampling candidate designs $\tilde{d}_k$ as outlined in Section 3.4 and algorithm 3.

The EVPI $v_{k-1}^{(i)}$ constitutes an upper bound on the amount that we should be willing to pay to learn about the risk from the k^{th} experiment; we therefore decide whether to carry out a further wave of analysis by comparing $v_{k-1}^{(i)}$ with the resource cost (set-up, computer time etc.) that would be incurred by performing another wave of the procedure 2. This comparison requires a judgement on the part of the user. We should certainly not pay more than $v_{k-1}^{(i)}$ for further analysis, since we are sure that we will not gain this much; however, paying $c_{\text{wv}_{i+1}}$ for wave $(i+1)$ will not necessarily result in a correspondingly valuable reduction in our uncertainty about the risk. The actual reduction in the risk will depend on the characteristics of the problem under study, and the quality of the emulators that we can fit. It may be possible, for some problems, to make simple judgements about the reduction in uncertainty that we might achieve (in the style of Goldstein and Seheult (2008)) and then compare the resulting EVPI with the current value; this is beyond the scope of the current work.

3.6 Input Selection for the Next Wave

If we decide using EVPI (Section 3.5) that we expect to gain from further analysis, then we may choose to run another wave of the algorithm (indexed by $(i+1)$). At this wave, we re-emulate the risk functions inside the candidate design spaces identified by the emulators for the previous wave. At the first wave, we fit our emulators over the whole of the design space, allowing us to build up a picture of risk behaviour over the whole of the design space, and to make an initial identification of designs which can be ruled out as unlikely to minimise the risk. At later waves, we can then focus our modelling efforts on those regions of the space which we have not

yet ruled out, building up a more accurate picture of risk behaviour in these regions and potentially allowing us to rule out more parts of the design space. This approach is similar to history matching: see, for example, Vernon et al. (2010).

When generating the risk data for these new emulators, we should be careful to focus only on those parts of the design space which are still interesting. This issue is discussed further in Section S2.4.

4 EXAMPLE: LINEAR MODEL

We first illustrate the method described in section 3 through application to a simple Bayesian linear model. In Section 4.1, we specify our model linking the model parameters and design inputs to the data, and specify the decision problem which we will solve; then, in Section 4.2, we run the approximate backward induction algorithm, and interpret the results.

4.1 Model and decision problem

We assume that a scalar observation z_j is available at each stage j, and that these observations are related to the model parameters and design inputs as

$$z_j = \sum_{k=1}^{n_{h_j}} h_{jk}\left(d_j\right) q_k + \epsilon_j$$

where d_j is a scalar design input, $h_j\left(d_j\right) = \left(h_{j1}\left(d_j\right), \ldots, h_{jn_{h_j}}\left(d_j\right)\right)^{\mathrm{T}}$ is a vector of n_{h_j} basis functions, and we assume that there are no external parameters w_j affecting the outcome at any stage. We assume that $q = \left(q_1, \ldots, q_{n_{h_j}}\right)^{\mathrm{T}}$ has a multivariate Gaussian prior distribution ($q \sim \mathcal{N}\left(\mu_q, V_q\right)$), and that the errors ϵ_j are independent, with zero-mean Gaussian distributions ($\epsilon_j \sim \mathcal{N}\left(0, v_{\epsilon_j}\right)$). We specify that our losses at all stages depend on the value of a new observation $\hat{z}\left(q\right) = h\left(\hat{d}\right)^{\mathrm{T}} q + \hat{\epsilon}$ at a known location $\hat{d}$ as

$$L_j\left(q, a_j\right) = l\left(\hat{z}\right)\left(\hat{z} - a_j\right)^2 \tag{12}$$

where $l\left(\hat{z}\right)$ is a known weight function. Using this loss function, the risk from an optimal terminal decision at stage j is

$$\rho_j^{\mathrm{trm}}\left[z_{[j]}, d_{[j]}\right] = \min_{a_j \in \mathcal{A}_j} \int l\left(\hat{z}\right)\left(\hat{z} - a_j\right)^2 p\left(\hat{z} | z_{[j]}, d_{[j]}\right) d\hat{z} \tag{13}$$

Due to the Gaussian specifications for the prior and the error structure, the distribution $p\left(\hat{z}|z_{[j]}, d_{[j]}\right)$ is also a Gaussian distribution; we find that $\hat{z}|z_{[j]}, d_{[j]} \sim \mathcal{N}\left(\mu_{\hat{z}}\left(z_{[j]}, d_{[j]}\right), V_{\hat{z}}\left(d_{[j]}\right)\right)$ where

$$
\begin{aligned}
V_{\hat{z}}\left(d_{[j]}\right) &= h\left(\hat{d}\right)^{\mathrm{T}} \hat{V}_q\left(d_{[j]}\right) h\left(\hat{d}\right) + v_{\hat{\epsilon}} & \hat{V}_q\left(d_{[j]}\right) &= \left[V_q^{-1} + \frac{1}{v_\epsilon} H\left(d_{[j]}\right)^{\mathrm{T}} H\left(d_{[j]}\right)\right] \\
\mu_{\hat{z}}\left(z_{[j]}, d_{[j]}\right) &= h\left(\hat{d}\right)^{\mathrm{T}} \hat{\mu}_q\left(z_{[j]}, d_{[j]}\right) & \hat{\mu}_q\left(z_{[j]}, d_{[j]}\right) &= \hat{V}_q\left(d_{[j]}\right)\left[V_q^{-1}\mu_q + \frac{1}{v_\epsilon} H\left(d_{[j]}\right)^{\mathrm{T}} z_{[j]}\right]
\end{aligned}
$$

where $H\left(d_{[j]}\right)$ is the design matrix created by stacking the vectors $h_j\left(d_j\right)$ as rows, and $\hat{\mu}_q\left(z_{[j]}, d_{[j]}\right)$ and $\hat{V}_q\left(d_{[j]}\right)$ are the parameters of the posterior Gaussian distribution $p\left(q|z_{[j]}, d_{[j]}\right)$.
If we differentiate the integral from (13) with respect to a_j and set to zero, we find that the optimal decision is

$$
a_j^* = \frac{1}{\mathrm{E}\left[l\left(\hat{z}\right)\right]} \mathrm{E}\left[l\left(\hat{z}\right)\hat{z}\right]
$$

for $\mathrm{E}\left[l\left(\hat{z}\right)\right] > 0$, where expectations are taken with respect to $p\left(\hat{z}|z_{[j]}, d_{[j]}\right)$, and that

$$
\rho_j^{\mathrm{trm}}\left[z_{[j]}, d_{[j]}\right] = \mathrm{E}\left[l\left(\hat{z}\right)\hat{z}^2\right] - \frac{1}{\mathrm{E}\left[l\left(\hat{z}\right)\right]} \mathrm{E}\left[l\left(\hat{z}\right)\hat{z}\right]^2
$$

If we choose a polynomial expression as the weighting function, the Gaussian form of the predictive distribution means that the risk can be computed in closed form using expressions for the non-central moments of a univariate Gaussian; due to the simplicity of this specification, then, we can compute the terminal risk at any stage in closed-form, without having to resort to numerical integration or sampling schemes.

4.2 Running the algorithm

We now run algorithm 2 for a two-stage version of the problem outlined in Section 4.1. For this example, we specify that $d_j \in [-1, 1]$ for both stages, and we fix the basis function vector for both stages to be

$$
h_j\left(d_j\right) = \left(1, (1 + d_j)(1 - d_j)\right)^{\mathrm{T}}
$$

The prior parameters of $p\left(q\right)$ are fixed to $\mu_q = (0, 0)^{\mathrm{T}}$ and $V_q = \mathrm{diag}\left(0.5^2, 0.5^2\right)$, and the measurement error variance is chosen to be different at each stage, with $v_{\epsilon_1} = (0.5)^2$ and $v_{\epsilon_2} = (0.1)^2$, so that we have the option of a more accurate measurement at the second stage. The weighting function for the loss (12) is chosen to be $l\left(\hat{z}\right) = 1 + \hat{z}^2$, so that $l\left(\hat{z}\right) > 0$ everywhere. The observation costs are set to be constant, with $c_1\left(d_1\right) = 0.05$ and $c_2\left(d_2\right) = 0.2$, so that the second measurement is also more expensive.

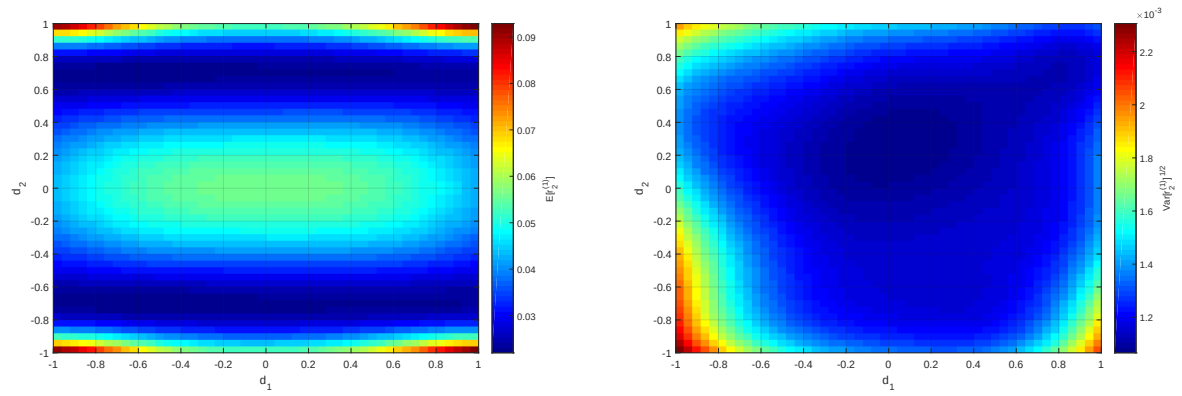


Figure 2: Emulator $r_2^{(1)}$ fitted to the risk at stage 2. Figure 2(a) shows the adjusted expectation $\mathrm{E}_{R_2^{(1)}}\left[r_2^{(1)}\right]$ for a grid of d_1 and d_2 values, and Figure 2(b) shows the corresponding standard deviation values $\mathrm{Var}_{R_2^{(1)}}\left[r_2^{(1)}\right]^{1/2}$. Figure 2(c) shows the true value of the risk ρ_2 for the same grid of design inputs, and 2(d) shows the absolute standardised distance between the true risk and the mean surface of the emulator. For all plots, z_1 and z_2 are fixed to the same quantiles of their respective conditional distributions.

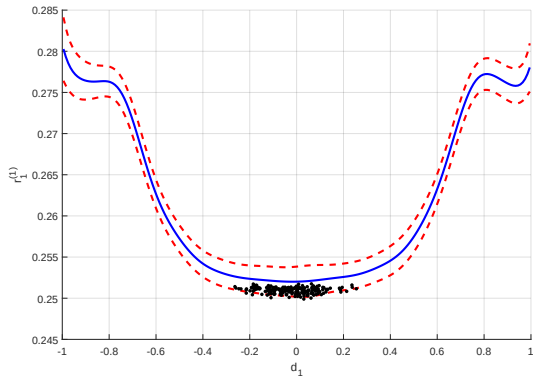


Figure 3: Emulator $\bar{r}_1^{(1)}$, with $\mathrm{E}_{R_1^{(1)}}\left[\bar{r}_1^{(1)}\right]$ shown in solid blue and $\mathrm{E}_{R_1^{(1)}}\left[\bar{r}_1^{(1)}\right] \pm 3\mathrm{Var}_{R_1^{(1)}}\left[\bar{r}_1^{(1)}\right]^{1/2}$ shown in dashed red. Also shown (as black markers) are 200 candidate design-risk sample pairs $(\tilde{d}_1, \bar{r}_1^{(1)}\left[\tilde{d}_1\right])$ from the space $\mathcal{D}_1^{(1)}$ (generated using the procedure 3).

4.2.1 First Wave

At the first wave of the algorithm ($i = 1$), we fit emulators to the risk over the whole of the design space. Because of the simplicity of this problem, the terminal risks ρ_j^{trm} are simple to evaluate at both stages; we therefore use these as the basis functions for our emulators $r_j^{(1)}$, with the data z_j substituted for its conditional expectation. We use separable squared exponential covariance functions, which ensure that it is easy for us to compute moments of $\bar{r}_j^{(1)}$. Further details of the model, and the fitting procedure, are provided in Section S3.1 of the supplementary material.

Figure 2 displays aspects of the model $r_2^{(1)}$; Figures 2(a) and 2(b) show the mean $\mathrm{E}_{R_2^{(1)}}\left[r_2^{(1)}\right]$ and standard deviation $\mathrm{Var}_{R_2^{(1)}}\left[r_2^{(1)}\right]^{1/2}$ surfaces of the emulator respectively, Figure 2(c) shows the true risk ρ_2 which the model is designed to represent, and Figure 2(d) shows the standardised discrepancy $(\rho_2 - \mathrm{E}_{R_2^{(1)}}\left[r_2^{(1)}\right])/\mathrm{Var}_{R_2^{(1)}}\left[r_2^{(1)}\right]^{1/2}$ between the true risk and the mean prediction under the model. These plots show that the true risk lies within three standard deviation error bars of the mean prediction at all points, and that the model captures important aspects of the variation in the risk across the design space.

Stopping Based on this first wave of analysis, we assess the optimal course of action under our current beliefs (Section 3.5). First, we interrogate the expected risk at a Latin hypercube of 2000 points, and find that $\hat{d}_1 = -0.018$, with $\mathrm{E}_{R_1^{(1)}}\left[\bar{r}_1^{(1)}\right] + c_1\left(\hat{d}_1\right) = 0.2520$ and $\mathrm{Var}_{R_1^{(1)}}\left[\bar{r}_1^{(1)}\right] = (0.0006)^2$ at this point. The risk from an immediate prior decision is $\rho_0^{\text{trm}} = 0.6345$, and so it is clear that it is optimal under current beliefs about the risk to carry out at least the first experiment. The EVPI for the risk is 0.0011; this should be compared to the cost of another wave in order to determine whether further analysis should be performed. In any case, we perform another wave to further illustrate the procedure from Section 3.

4.2.2 Second Wave

At the second wave, we re-fit the emulators within the candidate design space from the first wave. In order to cut down on computation time, we characterise the candidate design space at both stages using simple limits (see Section S2.3). The candidate design space $\mathcal{D}_1^{(1)}$ is illustrated in Figure 3; a set of 200 candidate design samples are shown alongside the emulator $\bar{r}_1^{(1)}$ that they are drawn from. For our emulators, we use the same basis and covariance functions as at the first wave. Further details of the fit at this wave are provided in Section S3.2.

Stopping We repeat the assessment from Section 3.5 using the emulators from the second wave. First, we interrogate the mean surface at a Latin hypercube of 2000 points, fixing $\hat{d}_1 = 0.103$, where $\mathrm{E}\left[\bar{r}_1^{(2)}\right] + c_1\left(\hat{d}_1\right) = 0.2524$ and $\mathrm{Var}\left[\bar{r}_1^{(2)}\right] = (0.0002)^2$. After this wave, the EVPI for the risk is 0.0001; a set of 200 samples from the candidate design space are shown in Figure 4, alongside the emulators $\bar{r}_1^{(i)}$ for waves $i = 1, 2$. Using this plot and the reduced EVPI value, we see that we have reduced our uncertainty about the risk at the second

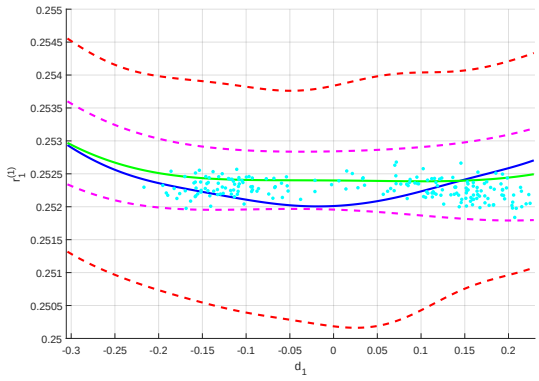


Figure 4: Emulators $\bar{r}_1^{(i)}$ for both waves $i = 1, 2$ within the candidate design space $\mathcal{D}_1^{(1)}$; $\mathrm{E}_{R_1^{(i)}}\left[\bar{r}_1^{(i)}\right]$ is shown as a solid line, and $\mathrm{E}_{R_1^{(i)}}\left[\bar{r}_1^{(i)}\right] \pm 3\mathrm{Var}_{R_1^{(i)}}\left[\bar{r}_1^{(i)}\right]^{1/2}$ is dashed (blue/red for wave 1 and green/magenta for wave 2). Also shown (as cyan markers) are are 200 candidate design-risk sample pairs $(\tilde{d}_1, \bar{r}_1^{(1)}\left[\tilde{d}_1\right])$ from the space $\mathcal{D}_1^{(2)}$ (generated using the procedure 3).

wave; it is becoming clear that the risk is rather flat in this region, so the candidate design space $\mathcal{D}_1^{(2)}$ is not much smaller than the space $\mathcal{D}_1^{(1)}$ from the first wave.

5 EXAMPLE: AIRBOURNE SENSING PROBLEM

We now apply the procedure outlined in Section 3 to a more complex problem. We consider an atmospheric dispersion problem, in which our goal is to infer the emission rates of a set of ground-based gas sources using concentration measurements collected along a sequence of flight paths. We would like to plan the sequence of flights in such a way that we obtain the 'best information' about possible sources of gas (according to some loss function).

In Section 5.1, we outline the model that we use for this problem, and in Section 5.2, we specify the components of the design problem that we will solve. Then, in Section 5.3, we outline the application of the procedure 2 to this problem, and discuss the results produced.

5.1 Model Specification

A commonly-used model for an atmospheric dispersion problem is the stationary Gaussian plume (see, for example, Hirst et al. (2012), Stockie (2011) or Jones et al. (2016)); this is a simple solution to the advection-diffusion equation (which gives a more general description of atmospheric dispersion), obtained under a number of simplifying assumptions, which describes the steady-state concentration downwind of a source under a wind direction which is constant over a suitable time-scale.

We denote the location of an individual measurement by $x = (x_x, x_y, x_h)^{\mathrm{T}}$ and the location of a single source by $c = (c_x, c_y, c_h)^{\mathrm{T}}$; we project the source-observation vector onto the wind direction vector $w = (w_x, w_y)$ as

follows

$$\omega\left(x,c,w\right) = \begin{pmatrix}\omega_x\\ \omega_y\\ \omega_z\end{pmatrix} = \frac{1}{\|w\|}\begin{pmatrix} w_x & w_y & 0\\ -w_y & w_x & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix} x_x - c_x\\ x_y - c_y\\ x_h\end{pmatrix}$$

In terms of this wind-projected distance, the contribution made by a source located at c with emission rate ψ to the measurement made at x is given by $a\left(\omega,\sigma\right)\psi$, where a is the Gaussian plume coupling coefficient computed as

$$a\left(\omega,\gamma\right) = \frac{1}{2\pi\|w\|\sigma_y\sigma_h}\exp\left[-\frac{\omega_y^2}{2\sigma_y^2}\right]\left[\exp\left[-\frac{(\omega_h - c_h)^2}{2\sigma_h^2}\right] + \exp\left[-\frac{(\omega_h + c_h)^2}{2\sigma_h^2}\right]\right]$$

$\sigma = (\sigma_y, \sigma_h)^{\mathrm{T}}$ are horizontal and vertical plume standard deviations, which depend on the downwind distance from source to measurement as $\sigma_y = \omega_y \tan\left(\gamma_y\right)$ and $\sigma_h = \omega_h \tan\left(\gamma_h\right)$, where $\gamma = (\gamma_y, \gamma_h)^{\mathrm{T}}$ are horizontal and vertical plume opening angles (measured in degrees), which can be estimated from atmospheric data, and are treated as known for the purposes of this analysis.

In this example, we design for a multi-source problem; the concentration contribution from a set of n_s sources located at $\{c_k\}_{k=1}^{n_s}$ with emission rates $\{\psi_k\}_{k=1}^{n_s}$ to a measurement at x under wind field w is simply the sum of the individual source contributions

$$y\left(x,w\right) = \sum_{k=1}^{n_s} a\left(c_k\right)\psi_k \tag{14}$$

where y is measured in parts per volume (ppv). In Section 5.2, we combine this model with a function describing the flight path to obtain the full model specification.

5.2 Design Problem

Flight parametrisation We are interested in inferring the emission rates of a grid of ground-based sources within a rectangular survey area. The sensor which we will use to collect concentration measurements is to be mounted on an aircraft, which will be flown at some altitude over the survey area. We assume that the sensor will make observations at a fixed rate during the course of a flight, and that flight paths will be pre-planned according to some low-dimensional parametrisation. We sepcify that each flight will consist of n_{fl} regularly-spaced, parallel transects of a given length. Each flight path is completely determined by five parameters:

- a central point $(d_x, d_y) \in [0, 5000] \times [0, 5000]$ (metres);
- an altitude $d_h \in [100, 300]$ (metres, assumed constant over the whole flight);
- a transect length $d_w \in [500, 1000]$ (metres);
- a transect spacing $d_d \in [100, 500]$ (metres, determining the perpendicular distance between each of the transects).

We specify that n_{ob} measurements will be made on each transect, and that all transects are perpendicular to the wind direction. We specify a deterministic map between the five design parameters and the vector of locations of the individual concentration measurements, which we denote by $x_p = t_p(d)$. Combining this mapping with the concentration model (14), our model for the data z_j observed at stage j of the problem is

$$z_{jp} = y\left(t_p\left(d_j\right), w_j\right) + \epsilon_{jp} \tag{15}$$

where $d_j = \{d_{jx}, d_{jy}, d_{jh}, d_{jw}, d_{jd}\}$ is the five-dimensional design input setting at the j^{th} stage, $w_j = \{w_{jx}, w_{jy}\}$ is the two dimensional wind field input (the external parameter set) at stage j, and ϵ_j is a $(n_{\text{fl}} \times n_{\text{ob}})$ dimensional vector of uncorrelated measurement errors. We assume that the wind parameters are independently uniformly distributed at each stage, over the following ranges (all units are metres/second):

- $w_{1x} \sim \mathcal{U}\left([-3,-2]\right)$, $w_{1y} \sim \mathcal{U}\left([2,3]\right)$
- $w_{2x} \sim \mathcal{U}\left([2,3]\right)$, $w_{2y} \sim \mathcal{U}\left([-3,-2]\right)$
- $w_{3x} \sim \mathcal{U}\left([-3,-2]\right)$, $w_{3y} \sim \mathcal{U}\left([-3,-2]\right)$

Model and decision problem We specify the following loss function for all stages of the problem

$$L_j\left(q, a_j\right) = C + l \sum_k \left(q_k - a_{jk}\right)^2$$

where $q = \{\psi_1, \ldots, \psi_{n_s}\}$ is the set of emission rates for the grid of sources. C is a baseline cost, and l is a scalar multiplier for the quadratic component. Under this loss function specification, the risk from an optimal terminal decision is

$$\rho_j^{\text{trm}}\left[z_{[j]}, w_{[j]}, d_{[j]}\right] = C + l \sum_k^{n_s} \text{Var}\left[q_k | z_{[j]}, w_{[j]}, d_{[j]}\right] \tag{16}$$

The cost of the flight path is also assumed to be constant across stages, and consists of a constant setup cost for each flight, and a cost per unit distance flown

$$c_j\left(d_j\right) = c_{\text{set}} + c_{\text{dst}}\Big[2\big((d_{jx} - x_{0x})^2 + (d_{jy} - x_{0y})^2\big)^{1/2} + (n_{\text{fl}} - 1)d_{jd} + n_{\text{fl}} d_{jw}\Big]$$

where c_{set} is the constant setup cost, c_{dst} is the cost per unit distance, and $x_0 = (x_{0x}, x_{0y}, 0)^{\text{T}}$ is the location of the airport from which the aircraft takes off. We choose to make a prior second-order specification for all of the components of the model (15), and we characterise the posterior distribution (1) in terms of our adjusted moments at each stage; for further discussion of this, see Section S4.1 of the supplementary material. Figure 5 illustrates an inference carried out using flights parametrised in this way; Figures 5(a) to 5(c) show the expected concentration fields for three different sets of wind conditions, and Figures 5(d) and 5(e) show the adjusted

moments $\{\mathrm{E}_{z_{[3]}}[q_k]\}$ and $\{\mathrm{Var}_{z_{[3]}}[q_k]^{1/2}\}$ computed using a set of observations collected on the flight paths shown.

5.3 Running the Algorithm

In this section, we outline the application of the approximate sequential design algorithm 2 to this problem, and discuss its results. Further details related to this section are provided in Section S4 of the supplementary material.

5.3.1 First Wave

At the first wave of the algorithm, we fit our emulators over the whole of the design space. At each stage, the emulator is fitted to the risk using the procedure outlined in Section 3.2 and Section S2.1. In all models, the regression basis consists of an intercept term and an approximation to the risk based on a comparison between the current terminal risk and the risk from a good design at the next stage, and the correlation function is chosen to have a squared exponential form. Further details of the modelling choices made in this example are given in Section S4.2.

Figure 6 shows a set of 100 samples from the candidate design space at stage $j = 1$ after wave $i = 1$ of the algorithm. We see that our modelling of the risk function has restricted the ranges of settings of all components of d_1 which appear in our candidate design space; this restriction is perhaps greatest in the (d_{1x}, d_{1y}) plane.

Stopping After the first wave, we assess the optimal course of action under our current beliefs (Section 3.5). First, we interrogate the expected risk (including the design cost) at a Latin hypercube of 2000 points and fix $\hat{d}_1$ as the design which minimises the risk over these points. Our expected risk at this point is $\mathrm{E}_{R_1^{(1)}}\left[\bar{r}_1^{(1)}\left[\hat{d}_1\right]\right] + c_1\left(\hat{d}_1\right) = 32.36$, with variance $\mathrm{Var}_{R_1^{(1)}}\left[\bar{r}_1^{(1)}\left[\hat{d}_1\right]\right] = (1.19)^2$; this is compared to the risk from an optimal decision under the prior, which is $\rho_0^{\mathrm{trm}} = 196.34$. From this, it is clear that, based on this analysis, we should conduct at least the first experiment.

Based on a set of 100 samples from the candidate design space, we asses that the EVPI for the risk is 0.62; in practice, whether we choose to perform further analysis on this basis will depend on the cost of our computational resources relative to the risk from the experiment. In any case, we perform another wave to further illustrate aspects of the procedure from Section 3.

5.3.2 Second Wave

At the second wave of the algorithm, we re-fit another sequence of emulators in the candidate design spaces at each stage (see Section 3.4). Figure 7 shows a set of candidate designs for each of the 3 stages of the problem sampled according to the procedure in algorithm 3. From this, we see that the candidate design spaces are strongly restricted in (d_x, d_y) space at all stages; on this basis, we decide to generate designs for the second

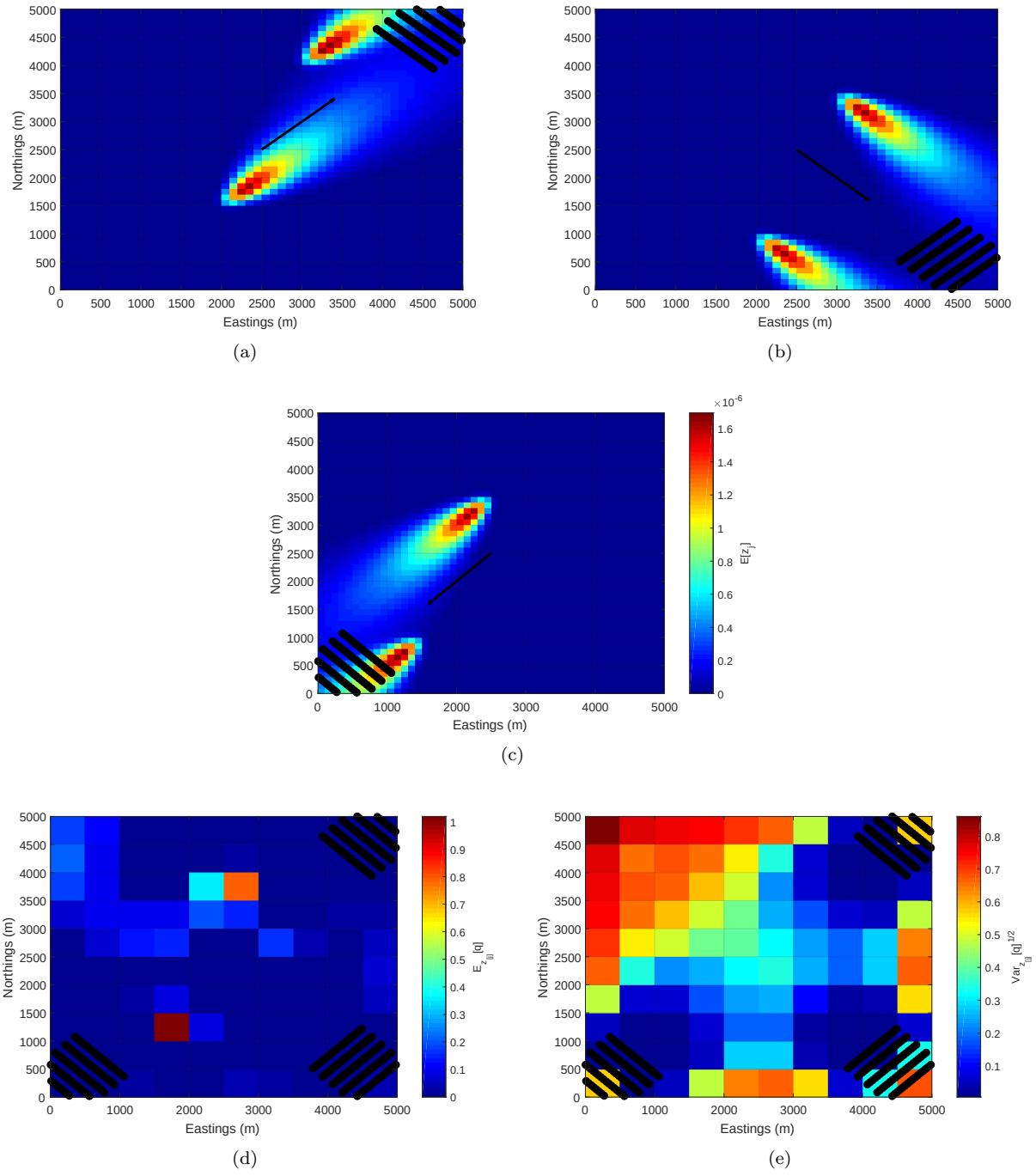


Figure 5: An example inference using the model from Sections 5.1 and 5.2. Figures 5(a) to 5(c) show the expected concentration measurements for a grid of points in (x, y) space, for an emission rate of $\psi_k = 1$ at both sources (locations indicated by magenta markers), under the wind conditions indicated by the black arrows. Observations of this concentration field are made at the black markers. Figures 5(d) and 5(e) show the adjusted beliefs $\mathrm{E}_{z_{[3]}}[q]$ and $\mathrm{Var}_{z_{[3]}}[q]$ computed by updating the prior specification from Section 5.1 using this observed data.

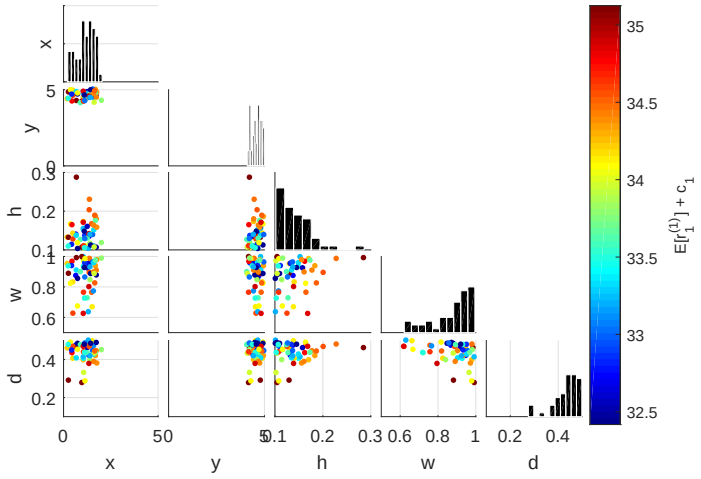


Figure 6: Scatter plot of candidate design points $\tilde{d}_1$ sampled from emulator $\bar{r}_1^{(1)}$ using the procedure 3. The colour scale indicates the expected risk $\mathrm{E}_{R_1^{(1)}}\left[\bar{r}_1^{(1)}\right] + c_1\left(\tilde{d}_1\right)$ at each point.

wave of the procedure by approximating the candidate design space in (d_x, d_y) using simple limits. For further discussion of this point, see Section S2.4. Our emulator fitting procedure at this wave is the same as that at the first wave. Further information specific to the procedure at this wave is provided in Section S4.2.

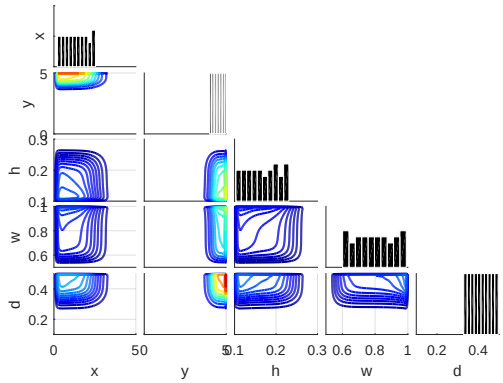
Figure 8 shows the expectation of the risk $\mathrm{E}_{R_1^{(2)}}\left[\bar{r}_1^{(2)}\right] + c_1\left(\tilde{d}_1\right)$ at a set of 100 candidate designs $\tilde{d}_1$, sampled as in algorithm 3; this should be compared with the equivalent set of samples from the candidate design space at the first wave shown in Figure 6. We see that after this wave, the candidate design space has roughly the same shape in the (d_{1x}, d_{1y}) plane, and that we have started to see greater restrictions placed on the settings of d_{1h}, d_{1w} and d_{1d} that appear in our candidate design space.

Stopping At the end of this wave, we again search the mean risk using a Latin hypercube of 2000 points, and find that $\mathrm{E}_{R_1^{(2)}}\left[\bar{r}_1^{(2)}\left[\hat{d}_1\right]\right] + c_1\left(\hat{d}_1\right) = 29.21$, with $\mathrm{Var}_{R_1^{(2)}}\left[\bar{r}_1^{(2)}\left[\hat{d}_1\right]\right]^{1/2} = (0.35)^2$. The EVPI after this wave is 0.04, so the second wave of analysis has resulted in a reduction of our uncertainty about the risk.

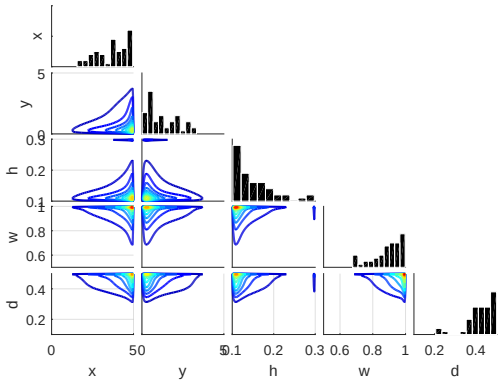
6 DISCUSSION

In this study, we considered the Bayesian optimal design problem in the situation where the data is available from a series of experiments (with associated costs), and where there is the option after each to evaluate the expected benefit from the remaining series of experiments and to either stop and use the data to make decisions, or to collect the next set of observations. We outlined the backward induction procedure that is used to solve such problems, and explained the computational issues that this algorithm presents in the general case.

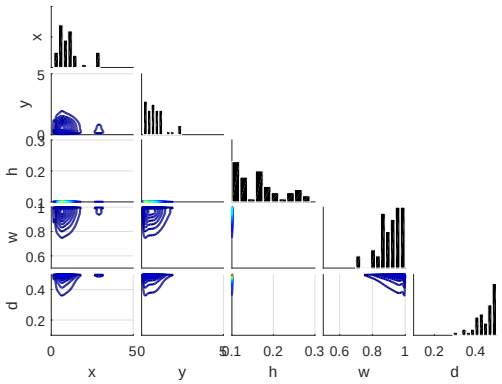
An approximating framework was proposed which uses Bayes linear emulators to perform some of the difficult calculations; these emulators capture a large amount of the structure of the problem, and allow us to use various existing tools to track the uncertainty in the calculation through the stages of the numerical procedure. This



(a) $j = 1$



(b) $j = 2$



(c) $j = 3$

Figure 7: Density contour plots of the candidate design spaces $\mathcal{D}_j^{(1)}$ at each stage $j = 1, 2, 3$; designing within the candidate design space at each stage restricts the candidate design space at each subsequent wave. Red contours enclose regions with higher densities of sampled points, and blue contours enclose regions with lower densities.

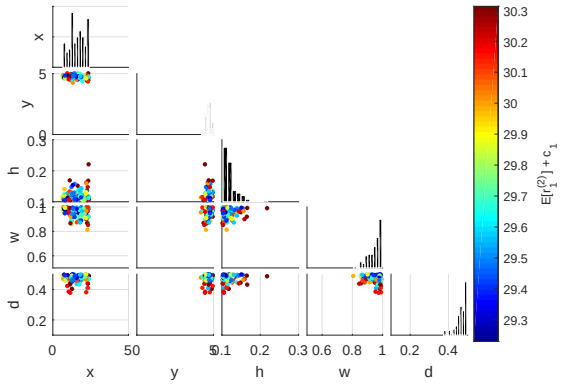


Figure 8: Scatter plot of $\tilde{d}_1$ sampled from $\bar{r}_1^{(2)}$ using the procedure 3. Colours indicate $\mathrm{E}_{R_1^{(2)}}\left[\bar{r}_1^{(2)}\right] + c_1\left(\tilde{d}_1\right)$ at each point.

approximation proves beneficial in application to both a simple linear model example, and to a more complex atmospheric dispersion modelling problem.

This research suggests several interesting areas for possible future research: first, it is often the case that the data collected on the system during the course of a sequential sampling scheme is used not only to make inferences about the parameters of the model, but also to motivate improvements to the model. The experimentation plan may specify that model development is to take place after the completion of all stages, or improvements may be planned between experimental stages. Goldstein and Rougier (2009) introduced a framework which links improved versions of a model to the current implementation and to the system under study; if we were to replace the model specification in section 3 with this framework, then the approximate backward induction procedure could be modified in order to generate designs which would take into account both the availability of future observations, and the possibility of future model development.

Assessing our uncertainty about the risk at its minimum is the most difficult task which we must perform as part of our backward induction approximation. The procedure presented in section 3.4 works well for low-dimensional problems where the risk function is relatively smooth; however, in higher-dimensional problems, or where there are multiple, disconnected regions of the design space which could minimise the risk, it becomes more difficult to use. Additionally, the procedure is sensitive to the size of the trial design used, and it is computationally difficult to draw enough samples to assess the variability in the minimum risk for each input setting. Characterising the distribution of the minimum of a stochastic process is an active research area (see, for example, Hennig and Schuler (2012) or Adler (1981)); theoretical guarantees for the behaviour of the sampling scheme from section 3.4, or the development of an alternative technique which does have such guarantees would increase confidence in our ability to properly track uncertainties across the stages for a greater range of sequential design problems.

References

R.J. Adler [1981]. The Geometry of Random Fields. Wiley.

J.O. Berger [1980]. Statistical Decision Theory. Springer.

K. Chaloner and I. Verdinelli [1995]. "Bayesian Experimental Design: A Review." *Statistical Science*, 10:273–304.

M.H. DeGroot [1970]. Optimal Statistical Decisions. Wiley.

I. Ford, D.M. Titterington, and C.P. Kitsos [1989]. "Recent Advances in Nonlinear Experimental Design." *Technometrics*, 31(1):49–60. ISSN 00401706. doi:10.2307/1270364.

M. Goldstein and J. Rougier [2009]. "Reified Bayesian Modelling and Inference for Physical Systems." *Journal of Statistical Planning and Inference*, 139:1221–1239. doi:10.1016/j.jspi.2008.07.019.

M. Goldstein and A. Seheult [2008]. "Prior Viability Assessment for Bayesian Analysis." *Journal of Statistical Planning and Inference*, 138(5):1271–1286. ISSN 03783758. doi:10.1016/j.jspi.2007.04.023.

P. Hennig and C.J. Schuler [2012]. "Entropy Search for Information-efficient Global Optimization." *Journal of Machine Learning Research*, 13:1809–1837.

B. Hirst, P. Jonathan, F. Gonzalez del Cueto, D. Randell, and O. Kosut [2012]. "Locating and Quantifying Gas Emission Sources Using Remotely Obtained Concentration Data." *Atmospheric Environment*, 45:141–158.

X. Huan and Y. Marzouk [2016]. "Sequential Bayesian Optimal Experimental Design via approximate dynamic programming." *Forthcoming*.

D.R. Jones, M. Schonlau, and W.J. Welch [1998]. "Efficient Global Optimization of Expensive Black-Box Functions." *Journal of Global Optimization*, 13:455–492.

M. Jones, M. Goldstein, D. Randell, and P. Jonathan [2016]. "Bayes Linear Analysis for Bayesian Optimal Design." *Journal of Statistical Planning and Inference*, 171:115–129. doi:10.1016/j.jspi.2015.10.011.

M.C. Kennedy and A. O'Hagan [2001]. "Bayesian Calibration of Computer Models." *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 63:425–464.

D.V. Lindley [1972]. Bayesian Statistics, A Review. Society for Industrial and Applied Mathematics.

P. Muller, D.A. Berry, A.P. Grieve, M. Smith, and M. Krams [2007]. "Simulation Based Sequential Bayesian Design." *Journal of Statistical Planning and Inference*, 137:3140–3150. doi:http://dx.doi.org/10.1016/j.jspi.2006.05.021.

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

F. Pasquill [1971]. "Atmospheric Dispersion of Pollution." *Quarterly Journal of the Royal Meteorological Society*, 97:369–395.

D. Randell, M. Goldstein, G. Hardman, and P. Jonathan [2010]. "Bayesian Linear Inspection Planning for Large-scale Physical Systems." *Proceedings of the Institution of Mechanical Engineers part O: Journal of Risk and Reliability*, 224:333–345. doi:10.1243/1748006XJRR322.

C.E. Rasmussen and Z. Ghahramani [2002]. "Bayesian Monte Carlo." In "Advances in Neural Information Processing Systems," MIT Press.

J.Q. Smith [2010]. Bayesian Decision Analysis: Principles and Practice. Cambridge University Press.

J.M. Stockie [2011]. "The Mathematics of Atmospheric Dispersion Modelling." *SIAM Review*, 53:349–372.

I. Vernon, M. Goldstein, and R.G. Bower [2010]. "Galaxy Formation: a Bayesian Uncertainty Analysis." *Bayesian Analysis*, 5:619–669. doi:10.1214/10-BA524.

D. Williamson and M. Goldstein [2012]. "Bayesian Policy Support for Adaptive Strategies Using Computer Models for complex physical systems." *Journal of the Operational Research Society*, 63:1021–1033. doi: 10.1057/jors.2011.110.

D. Williamson, M. Goldstein, L. Allison, A. Blaker, P. Challenor, L. Jackson, and M. Yamazaki [2013]. "History Matching for Exploring and Reducing Climate Model Parameter Space Using Observations and a Large Perturbed Physics Ensemble." *Climate Dynamics*, 41(7-8):1703–1729. ISSN 09307575. doi: 10.1007/s00382-013-1896-4.

Supplementary Material: Bayes Linear Analysis of Sequential Optimal Design Problems

Matthew Jones[1], Michael Goldstein[1], Philip Jonathan[2], and David Randell[2]

[1]Department of Mathematical Sciences, Durham University, Lower Mountjoy, Stockton Road, Durham, UK, DH1 3LE

[2]Statistics & Data Science, Shell Projects & Technology, Concord Business Park, Manchester, UK, M22 0RR

S1 BAYES LINEAR ANALYSIS

A Bayes linear analysis is a framework within which we update our prior beliefs about a collection of quantities in the light of observations; it differs from a probabilistic Bayesian analysis in that it treats expectation rather than probability as a primitive. Prior expectations, variances and covariances are specified for all components of the problem, and then upon learning the values of certain quantities, beliefs about the remainder are updated using linear fitting. A comprehensive introduction to the Bayes linear framework is provided in Goldstein and Wooff (2007).

The Bayes linear approach can be viewed either as a generalization of the fully Bayesian approach in which we remove the requirement that we must make a full probability specification for all quantities; alternatively, it can be viewed as a simple approximation to the fully Bayesian treatment, for situations where we are unwilling to make a full probability specification for all quantities, or where the full Bayesian analysis would be too difficult or time-consuming to be practical. Regardless of our views on the general appropriateness of a Bayes linear treatment, the simplicity of the Bayes linear framework allows us to make progress in problems where this would not be possible using a fully probabilistic specification; specifically, the use of Bayes linear methods can introduce tractability into design problems which would otherwise be too computationally difficult to solve.

In this section, we outline the Bayes linear methods which are used for analysis in the main article. In Section S1.1, we introduce the elements of a general Bayes linear analysis; then, in Section S1.2, we use these elements to develop a common model for an uncertain function. In Section S1.3, we demonstrate how this type of model can be used to compute our beliefs about the expectation of a function, and in Section S1.4, we outline how

it can be used to approximate a general, complex model in a way that makes the approximate procedure from Section 3 of the main article tractable.

S1.1 General Bayes Linear Analysis

Suppose that we are interested in a collection of quantities $C = \{A, B, D\}$, where $A = \{A_1, \ldots, A_{n_A}\}$, $B = \{B_1, \ldots, B_{n_B}\}$ and $D = \{D_1, \ldots, D_{n_D}\}$, and that we can make a second-order prior specification for the collection, consisting of expectations and variances for all components and covariances between all pairs of components. If we then observe the values of the collection D, we want to know how we should alter our beliefs about the other quantities in the light of this information. Under the Bayes linear framework, we use the new information to compute adjusted beliefs (Goldstein and Wooff, 2007).

Definition S1.1. *Our **adjusted expectation** for the collection B given observation of the collection D is*

$$E_D[B] = E[B] + Cov[B, D]\, Var[D]^{-1}\big[D - E[D]\big] \tag{S1}$$

Definition S1.2. *Our **adjusted variance** for the collection B given observation of the collection D is*

$$Var_D[B] = Var[B] - Cov[B, D]\, Var[D]^{-1} Cov[D, B] \tag{S2}$$

Definition S1.3. *Our **adjusted covariance** between the collections B and A given observation of the collection D is*

$$Cov_D[A, B] = Cov[A, B] - Cov[A, D]\, Var[D]^{-1} Cov[D, B] \tag{S3}$$

S1.2 Second-order Emulation

A second-order emulator is a particular Bayes linear model which we use to predict the behaviour of a function over its whole input domain based on a set of assumptions about its structure and a number of observations made (possibly with error) at known input locations. Emulation of complex functions can also be carried out using either a fully probabilistic specification (see, for example, Kennedy and O'Hagan (2001) or Goldstein and Rougier (2004)). We adopt the Bayes linear approach because it eliminates the need for us to make high-order moment specifications that we do not necessarily believe (as discussed in Goldstein and Wooff (2007)), and because it simplifies some of the analyses that we may wish to perform using our fitted emulator (e.g. learning about uncertain inputs; see Goldstein and Rougier (2006)).

Using $f(\theta) = \{f_1(\theta), \ldots, f_{n_f}(\theta)\}$ to denote a general multi-output function of inputs $\theta = \{\theta_1, \ldots, \theta_{n_\theta}\}$, our

model consists of a regression surface and a residual component

$$f_i(\theta) = \sum_p \beta_{ip} g_p(\theta) + u_i(\theta) \tag{S4}$$

where $g = \{g_1, \ldots, g_{n_g}\}$ is a set of known basis functions, the β_{ip} are corresponding uncertain weights, and u_i is a correlated residual process. We make a second-order prior specification for each of the uncertain components of the model, which consists of expectations $\mathrm{E}[\beta_{ip}]$ for all weights, covariances $\mathrm{Cov}[\beta_{ip}, \beta_{jq}]$ between pairs of weights and a covariance function $\mathrm{Cov}[u_i(\theta), u_j(\theta')]$ for the residual process. This prior specification is then combined with a set of noisy observations of the function to produce adjusted expectations $\mathrm{E}_F[f_i(\theta)]$ for all new inputs θ and adjusted covariances $\mathrm{Cov}_F[f_i(\theta), f_j(\theta)]$ for all new pairs of inputs. This adjustment procedure is detailed in Section S1.2.1, and the determination of the correlation parameters for the covariance function is considered in Section S1.2.2.

This type of model is useful in a number of different settings. If f is a complex function with high-dimensional input and output spaces that takes a long time to evaluate on a computer, then a model of the form (S4) can be fitted as a surrogate representation. A carefully specified model can be used to capture the important features of the function's behaviour, and can be run in a fraction of the time that it would take to run the function itself; the surrogate model can therefore be used to carry out more detailed analyses which would not originally have been possible.

Throughout the remainder of this section, we assume the Einstein summation convention, under which repeated indices are summed over; for example $\sum_p \beta_{ip} g_p(\theta) = \beta_{ip} g_p(\theta)$.

S1.2.1 Adjustment

Suppose that we observe the noise-corrupted function output at n_F known input locations $\{\theta_1, \ldots, \theta_{n_F}\}$. We use F_{ik} to denote the i^{th} component of the k^{th} function output, and we assume that

$$F_{ik} = f_i(\theta_k) + \eta_{ik}$$

If we assume that the noise components have mean zero, and we specify $\mathrm{Cov}[\eta_{ik}, \eta_{jl}]$ for all pairs of noise terms, then combining this with our prior specification from earlier, we have that

$$\begin{aligned} \mathrm{E}[F_{ik}] &= \mathrm{E}[\beta_{ip}] G_{pk} \\ \mathrm{Cov}[F_{ik}, F_{jl}] &= G_{pk} \mathrm{Cov}[\beta_{ip}, \beta_{jq}] G_{ql} + \mathrm{Cov}[u_i(\theta_k), u_j(\theta_l)] + \mathrm{Cov}[\eta_{ik}, \eta_{jl}] \end{aligned}$$

where the $G_{pk} = g_p(\theta_k)$ are the elements of the usual basis matrix. Our adjusted beliefs about the function outputs at new input settings are now

$$\mathrm{E}_F[f_i(\theta)] = \mathrm{E}_F[\beta_{ip}]\, g_p(\theta) + \mathrm{E}_F[u_i(\theta)] \tag{S5}$$

$$\begin{aligned}\mathrm{Cov}_F[f_i(\theta), f_j(\theta')] &= g_p(\theta)\,\mathrm{Cov}_F[\beta_{ip}, \beta_{jq}]\, g_q(\theta') + g_p(\theta)\,\mathrm{Cov}_F[\beta_{ip}, u_j(\theta')] \\ &\quad + \mathrm{Cov}_F[u_i(\theta), \beta_{jq}]\, g_q(\theta) + \mathrm{Cov}[u_i(\theta), u_j(\theta')]\end{aligned} \tag{S6}$$

where the individual adjusted components are detailed below.

Adjusted expectation The components of the adjusted expectation are computed through simple application of the Bayes linear adjustment rule (S1). For the regression coefficients, we have

$$\mathrm{E}_F[\beta_{ip}] = \mathrm{E}[\beta_{ip}] + \mathrm{Cov}[\beta_{ip}, F_{rs}]\,\mathrm{Var}[F]^{-1}_{rstu}\big[F_{tu} - \mathrm{E}[F_{tu}]\big]$$

where

$$\mathrm{Cov}[\beta_{ip}, F_{rs}] = \mathrm{Cov}[\beta_{ip}, \beta_{rt}]\, G_{ts}$$

and $\mathrm{Var}[F]^{-1}$ is computed as follows: we define $\tilde{F}$ such that $\tilde{F}_{(i+(k-1)n_f)} = F_{ik}$, and set

$$\mathrm{Var}[F]^{-1}_{ikjl} = \mathrm{Var}\left[\tilde{F}\right]^{-1}_{(i+(k-1)n_f),(j+(l-1)n_f)}$$

For the residual process, the adjusted expectation is

$$\mathrm{E}_F[u_i(\theta)] = \mathrm{Cov}[u_i(\theta), F_{rs}]\,\mathrm{Var}[F]^{-1}_{rstu}\big[F_{tu} - \mathrm{E}[F_{tu}]\big]$$

where

$$\mathrm{Cov}[u_i(\theta), F_{rs}] = \mathrm{Cov}[u_i(\theta), u_r(\theta_s)]$$

Adjusted covariance Likewise, the components of the adjusted covariance are computed through repeated application of S3. For the regression coefficients

$$\mathrm{Cov}_F[\beta_{ip}, \beta_{jq}] = \mathrm{Cov}[\beta_{ip}, \beta_{jq}] - \mathrm{Cov}[\beta_{ip}, F_{rs}]\,\mathrm{Var}[F]^{-1}_{rstu}\mathrm{Cov}[F_{tu}, \beta_{jq}]$$

and for the residual process

$$\mathrm{Cov}_F[u_i(\theta), u_j(\theta')] = \mathrm{Cov}[u_i(\theta), u_j(\theta')] - \mathrm{Cov}[u_i(\theta), F_{rs}]\,\mathrm{Var}[F]^{-1}_{rstu}\mathrm{Cov}[F_{tu}, u_j(\theta')]$$

Covariances between the components are

$$\mathrm{Cov}_F\left[\beta_{ip}, u_j\left(\theta'\right)\right] = -\mathrm{Cov}\left[\beta_{ip}, F_{rs}\right]\mathrm{Var}\left[F\right]^{-1}_{rstu}\mathrm{Cov}\left[F_{tu}, u_j\left(\theta'\right)\right]$$

where we have used the fact that the β_{ip} and the u_j are a priori uncorrelated.

S1.2.2 Fixing the Correlation Parameters

Generally, the covariance function which we choose will have the following form

$$\mathrm{Cov}\left[u_i\left(\theta\right), u_j\left(\theta'\right)|\lambda\right] = V_{ij}k\left(\theta, \theta'|\lambda\right)$$

where V is a matrix of marginal covariances, and k is a correlation function which depends on a set of parameters λ. In a fully Bayesian analysis, we would specify prior beliefs about $\{V, \lambda\}$ and then update these beliefs using the data F; generally, however, this procedure is too computationally intensive to be practical here. Instead, then, we adopt a common approximate treatment (see, for example, Kennedy and O'Hagan (2001), Craig et al. (2001)); we empirically fix V to a setting $\hat{V}$ which is empirically determined from the residuals of the initial mean regression surface $\mathrm{E}\left[\beta_{ip}\right] g_p\left(\theta\right)$, and we fix the correlation parameters λ to a setting $\hat{\lambda}$ determined through leave-one-out cross validation. For an outline of the leave-one-out cross validation procedure, see e.g. Rasmussen and Williams (2006).

S1.3 Bayes Linear Quadrature

Suppose that we are interested in the value of the expectation of the function f with respect to a certain subset of its inputs. We split the input set $\theta = \{a, b\}$, where the values of the $b = \{b_1, \ldots, b_{n_b}\}$ are known, and the $a = \{a_1, \ldots, a_{n_a}\}$ are unknown, with our beliefs about these components specified through a distribution $p\left(a\right)$. We want to compute

$$\bar{f}_i\left(b\right) = \int f_i\left(a, b\right) p\left(a\right) da$$

but we cannot do so directly; this could be because we do not have access to an algebraic expression for f, or it could be because we do not have access to an expression for the definite or indefinite integral for this combination of f and p. In either case, a simple approximating strategy is to fit a model of the kind outlined in section S1.2 and then to integrate.

This strategy was proposed for Gaussian process models by O'Hagan (1991), who referred to it as 'Bayesian quadrature'; it was also considered by Rasmussen and Ghahramani (2002), who called it 'Bayesian Monte Carlo'. In both cases, the same procedure is followed: a Gaussian process model is fitted to f, and then the mean and covariance functions for the expectation $\bar{f}$ of f are computed as the expectations of the mean and covariance

functions. We perform the same calculations as these authors, but in the Bayes linear setting.

The (adjusted) expectation of $\bar{f}$ is simply the integral of the adjusted expectation of f

$$\mathrm{E}_F\left[\bar{f}_i\left(b\right)\right] = \int \mathrm{E}_F\left[f_i\left(a,b\right)\right] p\left(a\right) da \tag{S7}$$

The covariance between $\bar{f}$ values at any pair of input points is also just the integral of the covariance between f values

$$\mathrm{Cov}_F\left[\bar{f}_i\left(b\right), \bar{f}_j\left(b'\right)\right] = \iint \mathrm{Cov}_F\left[f_i\left(a,b\right), f_j\left(a',b'\right)\right] p\left(a\right) p\left(a'\right) da da' \tag{S8}$$

Substituting the expressions (S5) and (S6) into the expressions (S7) and (S8), we obtain

$$\begin{aligned}
\mathrm{E}_F\left[\bar{f}_i\left(b\right)\right] &= \mathrm{E}_F\left[\beta_{ip}\right] \bar{g}_p\left(b\right) + \mathrm{E}_F\left[\bar{u}_i\left(b\right)\right] \\
\mathrm{Cov}_F\left[\bar{f}_i\left(b\right), \bar{f}_j\left(b'\right)\right] &= \bar{g}_p\left(b\right) \mathrm{Cov}_F\left[\beta_{ip}, \beta_{jq}\right] \bar{g}_q\left(b'\right) + \bar{g}_p\left(b\right) \mathrm{Cov}_F\left[\beta_{ip}, \bar{u}_j\left(b'\right)\right] \\
&\quad + \mathrm{Cov}_F\left[\bar{u}_i\left(b\right), \beta_{jq}\right] \bar{g}_q\left(b\right) + \mathrm{Cov}\left[\bar{u}_i\left(b\right), \bar{u}_j\left(b'\right)\right]
\end{aligned}$$

where

$$\bar{g}_p\left(b\right) = \int g_p\left(a,b\right) p\left(a\right) da$$

and

$$\begin{aligned}
\mathrm{Cov}_F\left[\bar{u}_i\left(b\right), \bar{u}_j\left(b'\right)\right] &= \mathrm{Cov}\left[\bar{u}_i\left(b\right), \bar{u}_j\left(b'\right)\right] - \mathrm{Cov}\left[\bar{u}_i\left(b\right), F_{rs}\right] \mathrm{Var}\left[F\right]^{-1}_{rstu} \mathrm{Cov}\left[F_{tu}, \bar{u}_j\left(b'\right)\right] \\
\mathrm{Cov}_F\left[\beta_{ip}, \bar{u}_j\left(b'\right)\right] &= -\mathrm{Cov}\left[\beta_{ip}, F_{rs}\right] \mathrm{Var}\left[F\right]^{-1}_{rstu} \mathrm{Cov}\left[F_{tu}, \bar{u}_j\left(b'\right)\right]
\end{aligned}$$

with

$$\begin{aligned}
\mathrm{Cov}\left[\bar{u}_i\left(b\right), u_j\left(\theta'\right)\right] &= \int \mathrm{Cov}\left[u_i\left(a,b\right), u_j\left(\theta'\right)\right] p\left(a\right) da \\
\mathrm{Cov}\left[\bar{u}_i\left(b\right), \bar{u}_j\left(b'\right)\right] &= \iint \mathrm{Cov}\left[u_i\left(a,b\right), u_j\left(a',b'\right)\right] p\left(a\right) p\left(a'\right) da da'
\end{aligned}$$

In Section 3.3 of the main article, this procedure is used to compute beliefs about the expected risk directly from our emulator for the risk.

S1.4 Bayes Linear Uncertainty Analysis

Frequently, when studying the behaviour of a set of properties of a real system $y\left(b\right) = \{y_1\left(b\right), \ldots, y_{n_f}\left(b\right)\}$ as a function of inputs b, we construct a function $f\left(a,b\right)$ which is designed to mimic its behaviour; in this context, we refer to f as a simulator. The simulator f is a function of the inputs b at which we can measure system behaviour, but also of a set of simulator-specific inputs a which do not affect system behaviour, but which must

be specified in order to generate a prediction for y. A common approach (e.g. Goldstein and Rougier (2006), Craig et al. (1997)) to relating the simulator to the system is to assume the existence of an unknown 'best' input setting a^* such that

$$y_i(b) = f_i(a^*, b) + \delta_i(b)$$

where $\delta_i(b) = \{\delta_1(b), \ldots, \delta_{n_f}(b)\}$ is the discrepancy between the simulator evaluated at a^* and the system, assumed to be independent of $\{f, a^*\}$.

For computationally expensive simulators f, a common approach is to construct a surrogate emulator; we generate a set of runs F on the simulator at known input settings, and then compute the adjusted moments $\mathrm{E}_F[f_i(a,b)]$ and $\mathrm{Cov}_F[f_i(a,b), f_k(a',b')]$ (as in Section S1.2), which we use to approximate the simulator at input settings where it has not been run. When coupled with a second-order uncertainty specification $\mathrm{E}[\delta_i(b)]$ and $\mathrm{Cov}[\delta_i(b), \delta_k(b')]$ for the discrepancy, this gives us a corresponding uncertainty specification for the system y itself.

We can use a model constructed in this way to approximate the design calculation for the system. Using the notation of Section 2.1 from the main article, if we assume that our data z_j are noise-corrupted measurements of the system, and we can divide the inputs b into external and design inputs $\{w_j, d_j\}$ for each stage j, then

$$z_{jp} = y_p(w_j, d_j) + \epsilon_{jp}$$

where ϵ is assumed to be independent of y. Our beliefs about z_j are then

$$\begin{aligned}
\mathrm{E}[z_{jp}] &= \mathrm{E}\left[\hat{f}_p(w_j, d_j)\right] + \mathrm{E}[\delta_p(w_j, d_j)] \\
\mathrm{Cov}[z_{jp}, z_{kq}] &= \mathrm{Cov}\left[\hat{f}_p(w_j, d_j), \hat{f}_q(w_k, d_k)\right] + \mathrm{Cov}[\delta_p(w_j, d_j), \delta_q(w_k, d_k)] + \mathrm{Cov}[\epsilon_{jp}, \epsilon_{kq}]
\end{aligned}$$

where $\hat{f}_p(w_j, d_j) = f_p\left(\{a_1^*, \ldots, a_{n_a}^*\}, \{w_j, d_j\}\right)$, and the covariance of the data with the best input setting is

$$\mathrm{Cov}\left[z_{jp}, a_q^*\right] = \mathrm{Cov}\left[\hat{f}_p(w_j, d_j), a_q^*\right]$$

For carefully-selected emulator basis and covariance functions, we can compute the above moments algebraically from the expressions (S7) and (S8) (see, e.g. Craig et al. (2001), Goldstein and Rougier (2009)). If we are collecting the data z_j with the intention of learning about the best input parameters a^*, then we can identify the model parameters with the best inputs ($q = a^*$) and perform the approximate design procedure from Section 3 of the main article using this model specification. Wherever we must characterise the posterior distribution (1) (section 2.1.1 of the main article), we do so using our adjusted beliefs computed using the above second-order specification.

S2 DESIGN ALGORITHM

In this section, we provide additional detail relating to the approximate sequential design algorithm presented in Section 3. In Section S2.1, we discuss the practical challenges relating to the fitting of an emulator as an approximation to the risk. Then, in Section S2.2, we consider the integration of the components of this model to obtain an approximation to the expected risk. In Section S2.3, we consider the implementation of the candidate design sampler from Section 3.4 of the main article. Finally, in Section S2.4, we consider how we might select an informative design for additional waves of the algorithm.

S2.1 Fitting the Emulator

In this Section, we consider practical aspects of fitting the model $r_j^{(i)}$ as an approximation to the risk ρ_j at the i^{th} wave.

S2.1.1 Evaluating the Risk

The risk evaluations that we use to fit the model $r_j^{(i)}$ are generated as in (10) (at the final stage, $j = n$) or (11) (all other stages, $j < n$); because of our uncertainty about $s_{j+1}^{(i)}$ (our approximation to the risk from an optimal design at stage $(j+1)$, defined in (9)), we cannot evaluate (11) for $(j < n)$ directly. To fit our emulator then, we compute $\mathrm{E}\left[R_{jk}^{(i)}\right]$ and $\mathrm{Cov}\left[R_{jk}^{(i)}, R_{jl}^{(i)}\right]$, and we fit the emulator to the expected values, treating the covariance specification as the covariance of the measurement error.

When generating risk evaluations for stages $j < n$, we must compute moments of the expression (11); using only our second-order beliefs, this is not possible, and so we make a distributional specification for $s_{j+1}^{(i)}$ characterised by $\mathrm{E}\left[s_{(j+1)k}^{(i)}\right]$ and $\mathrm{Cov}\left[s_{(j+1)k}^{(i)}, s_{(j+1)l}^{(i)}\right]$ (computed as in Section S2.3) and characterise the first- and second-order moments of the $\{R_{jk}^{(i)}\}$ by sampling. Generally, we choose a multivariate Gaussian, as this is simple to characterise using a second-order specification; if we have specific beliefs about the higher-order moments of the risk, then we may choose a different distributional specification to reflect these.

S2.1.2 Specifying the Prior

Our first task when fitting an emulator of the form (7) is to choose a set of basis functions, and make a second-order prior specification for the regression coefficients and the residual process. Generally, a more accurate emulator can be fitted if we choose basis functions $g_{jp}^{(i)}$ that can describe as much of the global structure of the risk as possible. If the terminal risk ρ_j^{trm} is simple to evaluate, then using this (or a related approximation to (3)) as a basis function may be an effective choice.

Another important factor to consider when selecting the basis functions is our ability to compute their expectations with respect to $p\left(z_j | z_{[j-1]}, w_{[j]}, d_{[j]}\right)$ and $p\left(w_j | w_{[j-1]}, d_{[j]}\right)$ (as in equation (8), Section S2.2); the ability to integrate these functions algebraically will reduce the computational complexity of computing (8) later. If

choosing to base the $g_{jp}^{(i)}$ on the terminal risk, then this can be achieved by replacing the inputs z_j and w_j with their conditional expectations $\mathrm{E}\left[z_j|z_{[j-1]}, w_{[j]}, d_{[j]}\right]$ and $\mathrm{E}\left[w_j|w_{[j-1]}, d_{[j]}\right]$, and then absorbing the effect on the output of variation around this mean level using the residual or nugget components. This strategy is adopted in both of the examples; see Sections S3 and S4.

Once the basis functions have been selected, we fix the prior moments of the $\beta_{jp}^{(i)}$. We do this by generating a sample of risks as outlined in Section S2.1.1, and using this to carry out an initial linear regression. The parameter estimates from this regression fit are used to fix $\mathrm{E}\left[\beta_{jp}^{(i)}\right]$ and $\mathrm{Cov}\left[\beta_{jp}^{(i)}, \beta_{jq}^{(i)}\right]$, and the residuals from the fitted surface are used to empirically fix the marginal variance $\mathrm{Var}\left[u_j^{(i)}\right]$ of the residual process.

S2.1.3 Jointly Updating

Using the prior specification made in Section S2.1.2, we jointly update the regression and residual components, as detailed in Section S1.2. As discussed in Section S1.2.2, the covariance function that we choose will generally depend on a number of correlation parameters; since a fully Bayesian treatment of these parameters would be too time-consuming, we instead fix these parameters using a leave-one-out cross validation procedure, and condition all subsequent calculations on this setting.

S2.1.4 Checking Inter-Wave Correspondence

Using the procedure from Section 3, we have the option to perform multiple waves of analysis. At wave $i = 1$, we emulate each of the risk functions for the first time; at subsequent waves $i = 2, 3, \ldots$ of the algorithm, however, we re-emulate the same risk functions over sub-regions of the design input space. We hope that, as we focus on smaller portions of the design space, we will be able to more accurately capture risk behaviour using our models; at a minimum, however, we expect that the predictive error bars of the emulators fitted at wave i should overlap with the predictive error bars of the emulators at waves $1, \ldots, (i-1)$. For each emulator fitted at later waves, this property can be checked to ensure that the behaviour of the procedure is as expected; failure to satisfy this criterion indicates that the fit of one or more of the emulators is a poor approximation to the risk.

S2.2 Computed the Expected Risk

Once the risk emulator $r_j^{(i)}$ has been fitted (Section S2.1), we integrate our model (7) to find our approximation $\bar{r}_j^{(i)}$ to the expected risk. At this step, we must be able to characterise the distributions $p\left(z_j|z_{[j-1]}, w_{[j]}, d_{[j]}\right)$ and $p\left(w_j|w_{[j-1]}, d_{[j]}\right)$, and compute the integrals of the basis and correlation functions with respect to these distributions. For general models $p\left(z_j|q, w_j, d_j\right)$ and prior specifications $p\left(q\right)$, these conditional distributions are complex objects; we may not be able to characterise them exactly, requiring us to rely on sampling algorithms or other numerical methods to investigate their characteristics.

For such complex distributions, we must numerically evaluate the integrals of basis or covariance functions, or adopt a strategy which approximately characterises the distributions in a way which allows us to perform the integrals directly. One possible strategy is to approximate using distributions characterised by the Bayes linear adjusted moments $\{\mathrm{E}_{z_{[j-1]}}\left[z_j|w_{[j]},d_{[j]}\right],\mathrm{Var}_{z_{[j-1]}}\left[z_j|w_{[j]},d_{[j]}\right]\}$ and $\{\mathrm{E}_{w_{[j-1]}}\left[w_j|d_{[j]}\right],\mathrm{Var}_{w_{[j-1]}}\left[w_j|d_{[j]}\right]\}$. If this, or any other, approximation strategy is adopted, its effect on our beliefs must be considered carefully.

S2.3 Characterising the Optimal Design Space

Trial design size The algorithm 3 is our general procedure for characterising the candidate design space. In order to use this procedure, we must choose a size $M_j^{(i)}$ for the trial design. Ideally, we would like to be able to determine the minimum sample size beyond which the characteristics of the sampled candidate designs do not change; however, there is no way to do this analytically, and it would be too computationally expensive to compare a large number of sizes $M_j^{(i)}$ empirically. Instead, then, we adopt a simple approach, where we generate samples of candidate designs $\tilde{d}_j$ and corresponding risks $\bar{r}_j^{(i)}+c_j\left(\tilde{d}_j\right)$ for a range of different values for $M_j^{(i)}$, and choose the design size which we judge to represent an appropriate trade-off between time to generate a sample, and stabilisation of sample characteristics.

Characterising the risk We cannot evaluate $s_j^{(i)}$ (equation (9)) exactly, since we do not know the optimal design setting d_j^*. Replacing the true optimal design with a candidate design $\tilde{d}_j$ sampled according to algorithm 3, our beliefs about the risk within the candidate design space are

$$\begin{aligned}
\mathrm{E}\left[s_j^{(i)}\right] &= \mathrm{E}\left[\mathrm{E}_{R_j^{(i)}}\left[\bar{r}_j^{(i)}\left[\tilde{d}_j\right]\right]+c_j\left(\tilde{d}_j\right)\right] \\
\mathrm{Cov}\left[s_j^{(i)},(s_j^{(i)})'\right] &= \mathrm{E}\left[\mathrm{Cov}_{R_j^{(i)}}\left[\bar{r}_j^{(i)}\left[\tilde{d}_j\right],\bar{r}_j^{(i)}\left[\tilde{d}_j\right]\right]\right] \\
&\quad + \mathrm{Cov}\left[\mathrm{E}_{R_j^{(i)}}\left[\bar{r}_j^{(i)}\left[\tilde{d}_j\right]\right]+c_j\left(\tilde{d}_j\right),\mathrm{E}_{R_j^{(i)}}\left[\bar{r}_j^{(i)}\left[\tilde{d}_j\right]\right]+c_j\left(\tilde{d}_j\right)\right]
\end{aligned} \tag{S9}$$

where we have used the compact notations $s_j^{(i)}=s_j^{(i)}\left[z_{[j-1]},w_{[j-1]},d_{[j-1]}\right]$ and $(s_j^{(i)})'=s_j^{(i)}\left[z'_{[j-1]},w'_{[j-1]},d'_{[j-1]}\right]$, we have suppressed the dependence on the inputs up to stage $(j-1)$ of all quantities on the right-hand side, and the outer expectations are taken with respect to $\tilde{d}_j$ by sampling design settings according to algorithm 3. Sampling candidate designs according to algorithm 3 can be computationally demanding, and in large problems, repeatedly generating candidate designs $\tilde{d}_j$ for each input setting may be computationally infeasible. In such problems, we simplify the approach further by characterising using a single candidate design; we simply set $\mathrm{E}\left[s_j^{(i)}\right]$ to be the expectation of $\bar{r}_j^{(i)}$ at the sampled point, and the first term of (S9) is set to be the covariance between $\bar{r}_j^{(i)}$ values at the two different input settings. For the second term in equation (S9), we approximate by sampling the variance of the expectation within the candidate design space at an handful of different input settings, and then fixing the variance of the expectation for any input setting to the average of these values; the

covariances between pairs of expectations at different input settings are fixed to zero.

Simple characterisation of design spaces As written, the procedure 3 for sampling candidate minima is recursive; we must be able to generate a set of trial designs inside the space $\mathcal{D}_j^{(i-1)}$ in order to sample from $\mathcal{D}_j^{(i)}$. For problems where we carry out multiple waves, it is likely that a recursive implementation will be infeasible. Instead, we may choose to approximate the candidate design space by characterising each space $\mathcal{D}_j^{(i)}$ in terms of simple limits. Approximating the spaces in this way will capture the ranges of design values which we have not yet ruled out, but will ignore any dependencies between the components of d_j induced by the shape of the risk emulator.

S2.4 Designing for Future Waves

At the first wave of the algorithm, we select the design inputs used to generate the data $R_j^{(1)}$ in a space-filling way (for example, a Latin hypercube); this is a common approach to achieving good coverage of the input space when fitting emulators (see e.g. Vernon et al. (2010)). At later waves $i = 2, 3, \ldots$, we re-fit emulators in sub-regions of the design space in order to build up a more accurate picture of risk behaviour; we select the design inputs used for the emulator fits at these waves so that they focus only on design input regions which our emulators at wave i indicate may be interesting.

At wave $i > 1$, we choose design inputs d_j for the emulator fit which:

- lie in the candidate design space $\mathcal{D}_j^{(i-1)}$ (characterised as in algorithm 3), and;
- satisfy the following criterion for $j > 1$, $k = 1, \ldots, (i-1)$

$$\frac{\mathrm{E}_{R_j^{(i)}}\left[r_j^{(i)}\left[z_{[j-1]}, w_{[j-1]}, d_{[j]}\right]\right] + c_j\left(d_j\right) - \rho_{j-1}^{\mathrm{trm}}\left[z_{[j-1]}, w_{[j-1]}, d_{[j-1]}\right]}{\mathrm{Var}_{R_j^{(i)}}\left[r_j^{(i)}\left[z_{[j-1]}, w_{[j-1]}, d_{[j]}\right]\right]^{1/2}} \leq 3 \tag{S10}$$

By the three-sigma rule of Pukelsheim (1994) (for continuous, unimodal distributions), the criterion (S10) corresponds to a specification that we only select design inputs for which there is a greater than 5% chance under our current beliefs that we will continue sampling at least as far as stage j.

S3 EXAMPLE: LINEAR MODEL

In this section, we provide additional detail relating to the linear model example presented in Section 4 of the main article.

S3.1 First Wave: Details

Stage 2 We begin the first wave by fitting $r_2^{(1)}$ as an approximation to ρ_2. Following the procedure outlined in Section S2.1, we begin by selecting basis functions for this emulator; at this stage, the risk $\rho_2 = \rho_2^{\text{trm}}$, and so we select the terminal risk as the basis of our approximation. We fix $g_{21}^{(1)} = 1$ and $g_{22}^{(1)} = \rho_2^{\text{trm}}\left[\tilde{z}_{[2]}, d_{[2]}\right]$, where $\tilde{z}_{[2]} = \{z_{[1]}, \mathrm{E}\left[z_2|z_{[1]}, d_{[2]}\right]\}$ is the set of data inputs with z_2 substituted for its conditional expectation. For the covariance function, we choose a separable form, as follows

$$\mathrm{Cov}\left[u_2^{(1)}\left(z_{[2]}, d_{[2]}\right), u_2^{(1)}\left(z'_{[2]}, d'_{[2]}\right)\right] = V_2^{(1)} \prod_{j=1}^{2}\left[k_{\text{se}}\left(z_j, z'_j|\lambda_z\right) k_{\text{se}}\left(d_j, d'_j|\lambda_d\right)\right]$$

where k_{se} is a squared exponential (SE) correlation function (see Appendix A). Note that we specify the same correlation lengths for each input at both stages. These choices of basis and covariance function ensure that we will be able to compute the moments of (8) algebraically. We carry out out initial regression using 200 risk evaluations, fixing the prior specification for the regression components and the marginal variance $V_2^{(1)}$ of the residual process. We then jointly update the regression and residual components using 300 risk evaluations, fixing the correlation lengths $\{\lambda_z, \lambda_d\}$ using leave-one-out cross validation (where we test 2000 candidate settings, chosen according to a Latin hypercube, and choose the best of these). Finally, we check our fitted emulator against a new sample of 100 risks; none of these lie outside predictive three-standard deviation error bars.

Once the emulator has been fitted, we characterise our approximation $\bar{r}_2^{(1)}$ to the expected risk $\bar{\rho}_2$ at this stage (equation (8), Section S1.3). Because of the simplicity of the problem, the distribution $p\left(z_2|z_{[1]}, d_{[2]}\right)$ is a Gaussian. The basis function was designed not to depend on z_2 so we have that $\bar{g}_{22}^{(1)} = g_{22}^{(1)}$. For the residual process, to compute the moments from Section S1.3, we compute integrals of the covariance function. Integrating in the first argument, we have

$$\begin{aligned}\mathrm{Cov}\left[\bar{u}_2^{(1)}\left(z_{[1]}, d_{[2]}\right), u_2^{(1)}\left(z'_{[2]}, d'_{[2]}\right)\right] &= V_2^{(1)} k_{\text{se}}\left(z_1, z'_1|\lambda_z\right) k_{\text{se}}\left(d_1, d'_1|\lambda_d\right) \\ &\quad \times \bar{k}_{\text{se}}\left(z'_1|\lambda_z\right) k_{\text{se}}\left(d_2, d'_2|\lambda_d\right)\end{aligned}$$

where $\bar{k}_{\text{se}}$ is the SE correlation function integrated once with respect to $p\left(z_2|z_{[1]}, d_{[2]}\right)$ (see Appendix A.1). Integrating again, we have

$$\begin{aligned}\mathrm{Cov}\left[\bar{u}_2^{(1)}\left(z_{[1]}, d_{[2]}\right), \bar{u}_2^{(1)}\left(z'_{[1]}, d'_{[2]}\right)\right] &= V_2^{(1)} k_{\text{se}}\left(z_1, z'_1|\lambda_z\right) k_{\text{se}}\left(d_1, d'_1|\lambda_d\right) \\ &\quad \times \bar{\bar{k}}_{\text{se}}\left(\lambda_z\right) k_{\text{se}}\left(d_2, d'_2|\lambda_d\right)\end{aligned}$$

where $\bar{\bar{k}}_{\text{se}}(.)$ is the SE function integrated in both arguments (see Appendix A.1).

Finally at this stage, we assess the additional variance contribution to $s_2^{(1)}$ that we include to represent our

uncertainty about the location of the optimal design (Section S2.3). We set the design size for the candidate design sampling algorithm to $M_2^{(1)} = 200$, and we generate 20 candidate designs and corresponding risks at each of 10 input settings $\{z_{[1]}, d_{[1]}\}$ using algorithm 3. As outlined in section S2.3, we fix the second component of (S9) to $(7.8 \times 10^{-4})^2$ and assume that this is constant across the input domain.

Stage 1 We now fit the emulator $r_1^{(1)}$ which approximates the risk at the first stage. We follow the procedure in Section S2.1; our basis functions are $g_{11}^{(1)} = 1$ and $g_{12}^{(1)} = \rho_1^{\text{trm}}\left[\tilde{z}_{[1]}, d_{[1]}\right]$, and our covariance function again has a separable squared exponential form

$$\text{Cov}\left[u_1^{(1)}\left(z_{[1]}, d_{[1]}\right), u_1^{(1)}\left(z'_{[1]}, d'_{[1]}\right)\right] = V_1^{(1)} k_{\text{se}}\left(z_1, z'_1|\lambda_z\right) k_{\text{se}}\left(d_1, d'_1|\lambda_d\right)$$

For this emulator, we carry out our initial regression using 200 risk evaluations, and we perform our joint update and cross-validation using 500 risk points (with 2000 candidate correlation parameter settings for the cross-validation). We check the fitted emulator by predicting an additional set of 100 risk points, and find that 1% lie outside predictive three-standard deviation error bars. The integrals of the covariance function with respect to $p\left(z_1|d_{[1]}\right)$ correspond to those at the second wave.

S3.2 Second Wave: Details

Stage 2 We judge that the basis and covariance functions that were used at the first wave performed well, and so we choose to make the same specifications for all stages at this wave. To fit the emulator $r_2^{(2)}$, we use 200 risk points for the initial regression, 300 for the joint update and 100 for model checking. Again, 2000 candidate correlation settings are tested in the cross-validation. Again, none of the validation data lie outside of the three-standard deviation error bars.

After fitting, we compute the additional variance contribution to $s_2^{(2)}$ (Section S2.3). Again, we set the design size to $M_2^{(2)} = 200$, and we generate 20 candidate designs and corresponding risks at each of 10 input locations using algorithm 3. We fix the second component of equation (S9) to $(6.7 \times 10^{-4})^2$, the average of these values, and assume that this is constant across the input domain.

Stage 1 To fit this emulator ($r_1^{(2)}$), we use 200 risk evaluations for the initial regression, 200 for the joint update (with 2000 correlation settings tested in the cross-validation) and 100 for model checking. All of the validation points are found to lie within three-standard deviation error bars.

S4 EXAMPLE: AIRBOURNE SENSING PROBLEM

In this section, we provide additional detail relating to the atmospheric dispersion example presented in Section 5 of the main article.

S4.1 Prior Specification and Posterior Risk

We make a prior second-order specification for all of the components of the model (15) introduced in Section 5.2. Previous treatments of this problem have opted for more complex prior specifications, e.g. Hirst et al. (2012) allowed for an unknown number of sources at unknown locations, and carried out their inference using a reversible jump MCMC scheme; in our treatment, we opt instead for this simple, second-order specification, as it allows us to characterise the reduction in uncertainty achieved by observing particular data sets, while ensuring the tractability of the design calculations. The prior specification consists of the components $\mathrm{E}[q_k]$ and $\mathrm{Cov}[q_k, q_l]$ for each of the model parameters $q_k = \psi_k$, and $\mathrm{Var}[\epsilon_{jp}]$ for the noise components ϵ_{jp} (we assume that $\mathrm{Cov}[\epsilon_{jp}, \epsilon_{kq}] = 0$ for $jp \neq kq$).

For the chosen loss function, the terminal risk in this problem is given in equation (16) (Section 5.2); in general, this depends on the posterior covariances between pairs of model parameters. Under our second-order specification, then, we simply characterise this risk using the adjusted covariance $\mathrm{Cov}_{z_{[j]}}\left[q_k, q_l | w_{[j]}, d_{[j]}\right]$.

S4.2 Emulator Fit: Details

Basis and covariance functions We make the same choices for the basis and covariance functions at all three stages of the problem, at both waves. We specify that $g^{(i)}_{j1} = 1$ for $j = 1, 2, 3$, that at stage n

$$g^{(i)}_{n2}\left(z_{[n]}, w_{[n]}, d_{[n]}\right) = \rho^{\mathrm{trm}}_n\left[\tilde{z}_{[n]}, \tilde{w}_{[n]}, d_{[n]}\right]$$

and that at stages $j < n$

$$g^{(i)}_{j2}\left(z_{[j]}, w_{[j]}, d_{[j]}\right) = \min\left[\rho^{\mathrm{trm}}_j\left[\tilde{z}_{[j]}, \tilde{w}_{[j]}, d_{[j]}\right], \mathrm{E}_{R^{(1)}_{j+1}}\left[\bar{r}^{(i)}_{j+1}\left[\tilde{z}_{[j]}, \tilde{w}_{[j]}, \tilde{d}_{[j+1]}\right]\right] + c_{j+1}\left(\xi_{j+1}\right)\right]$$

where $\tilde{z}_{[j]} = \{z_{[j-1]}, \mathrm{E}_{z_{[j-1]}}\left[z_j | \tilde{w}_{[j]}, d_{[j]}\right]\}$, $\tilde{w}_{[j]} = \{w_{[j-1]}, \mathrm{E}\left[w_j\right]\}$, and $\tilde{d}_{[j+1]} = \{d_{[j]}, \xi_{j+1}\}$, where ξ_{j+1} is a pre-specified design setting, selected in a region of the design space which, based on initial data exploration, we believe may be near-optimal for a wide range of risk input settings at the previous stages.

For the covariance function, we again choose a separable squared exponential form at all stages

$$\mathrm{Cov}\left[u^{(i)}_j\left(z_{[j]}, w_{[j]}, d_{[j]}\right), u^{(i)}_j\left(z'_{[j]}, w'_{[j]}, d'_{[j]}\right)\right] = V^{(i)}_j \prod_{p=1}^{j}\left[\left[\prod_{q=1}^{2} k_{\mathrm{se}}\left(w_{pq}, w_{pq} | \lambda_{w_q}\right)\right]\left[\prod_{q=1}^{5} k_{\mathrm{se}}\left(d_{pq}, d_{pq} | \lambda_{d_q}\right)\right]\right]$$

where $k_{\text{se}}(.,.|\lambda)$ is defined in Appendix A, and we specify that the correlation parameters λ_{w_q} and λ_{d_q} corresponding to particular external and design inputs are the same across all stages j. When fitting an emulator, the parameters $\{\lambda_{w_1}, \lambda_{w_2}, \lambda_{d_1}, \ldots, \lambda_{d_5}\}$ are fixed using cross-validation (see Section S1.2.2).

In order to characterise $\bar{r}_j^{(i)}$, we must compute integrals of the basis and covariance functions (Section 3.3 of the main article). Because of the way the basis functions have been selected, we simply have that $\bar{g}_j^{(i)} = g_j^{(i)}$ for all i, j. For the covariance function, integrating in the first argument, we have that

$$\begin{aligned}\text{Cov}\left[\bar{u}_j^{(i)}\left(z_{[j-1]}, w_{[j-1]}, d_{[j]}\right), u_j^{(i)}\left(z'_{[j]}, w'_{[j]}, d'_{[j]}\right)\right] &= V_j^{(i)}\Big[\prod_{q=1}^{2}\bar{k}_{\text{se}}\left(w'_{jq}|\lambda_{w_q}\right)\Big[\prod_{p=1}^{j-1}k_{\text{se}}\left(w_{pq}, w_{pq}|\lambda_{w_q}\right)\Big]\Big] \\ &\quad \times\Big[\prod_{p=1}^{j}\prod_{q=1}^{5}k_{\text{se}}\left(d_{pq}, d_{pq}|\lambda_{d_q}\right)\Big]\end{aligned}$$

and integrating again in the second argument, we have that

$$\begin{aligned}\text{Cov}\left[\bar{u}_j^{(i)}\left(z_{[j-1]}, w_{[j-1]}, d_{[j]}\right), \bar{u}_j^{(i)}\left(z'_{[j-1]}, w'_{[j-1]}, d'_{[j]}\right)\right] &= V_j^{(i)}\Big[\prod_{q=1}^{2}\bar{\bar{k}}_{\text{se}}\left(\lambda_{w_q}\right)\Big[\prod_{p=1}^{j-1}k_{\text{se}}\left(w_{pq}, w_{pq}|\lambda_{w_q}\right)\Big]\Big] \\ &\quad \times\Big[\prod_{p=1}^{j}\prod_{q=1}^{5}k_{\text{se}}\left(d_{pq}, d_{pq}|\lambda_{d_q}\right)\Big]\end{aligned}$$

where $\bar{k}_{\text{se}}(.|\lambda)$ and $\bar{\bar{k}}_{\text{se}}(\lambda)$ are computed as in Appendix A.2

A SQUARED EXPONENTIAL CORRELATION FUNCTION

In both of the examples (Sections 4 and 5 of the main article), we use squared exponential correlation functions for our risk emulators. This has the following form for general scalar inputs $\{x, x'\}$

$$k_{\text{se}}(x, x'|\lambda) = \exp\big[-\frac{\lambda}{2}(x - x')^2\big]$$

where λ is a correlation parameter which governs the rate at which the correlation between function outputs at x and x' decays to zero as the squared distance between the inputs increases. Correlation functions for multi-input functions are constructed as products of correlation functions for the individual inputs; see Sections S3.1 and S4.2. When emulating, the parameters λ are fixed through cross validation; see Section S1.2.2.

The SE correlation function is chosen for its simplicity, and because it is simple to integrate with respect to both Gaussian and uniform distributions. In Section A.1, we compute integrals of a single function with respect to a Gaussian distribution (the approach taken in Section S3.1) and in Section A.2, we compute integrals with respect to a uniform distribution (the approach taken in Section S4.2).

A.1 Integrals: Gaussian Case

Suppose that $x \sim \mathcal{N}(\mu, v)$; then, the integral of k_{se} in its first argument is

$$\begin{aligned}\bar{k}_{\text{se}}(x'|\lambda) &= \int_{-\infty}^{\infty} k_{\text{se}}(x, x'|\lambda)\, p(x)\, dx \\ &= \sqrt{\frac{2\pi}{\lambda}} \int_{-\infty}^{\infty} \Big[\sqrt{\frac{\lambda}{2\pi}} \exp\big[-\frac{\lambda}{2}(x'-x)^2\big] \frac{1}{\sqrt{2\pi v}} \exp\big[-\frac{1}{2v}(x-\mu)^2\big]\Big] dx \\ &= \sqrt{\frac{2\pi}{\lambda}} \Big[\frac{1}{\sqrt{2\pi\hat{v}}} \exp\big[-\frac{1}{2\hat{v}}(x'-\mu)^2\big]\Big]\end{aligned}$$

using standard results for the Gaussian distribution, where $\hat{v} = v + \frac{1}{\lambda}$. If $x' \sim \mathcal{N}(\mu', v')$, then the integral of the SE correlation function in both arguments is

$$\begin{aligned}\bar{\bar{k}}_{\text{se}}(\lambda) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} k_{\text{se}}(x, x'|\lambda)\, p(x)\, p(x')\, dx\, dx' \\ &= \int_{-\infty}^{\infty} \bar{k}_{\text{se}}(x'|\lambda)\, p(x')\, dx' \\ &= \sqrt{\frac{2\pi}{\lambda}} \Big[\frac{1}{\sqrt{2\pi\hat{\hat{v}}}} \exp\big[-\frac{1}{2\hat{\hat{v}}}(\mu-\mu')^2\big]\Big]\end{aligned}$$

where $\hat{\hat{v}} = v + v' + \frac{1}{\lambda}$.

A.2 Integrals: Uniform Case

In the case where $x \sim \mathcal{U}(l, u)$ has a uniform distribution on $[l, u]$, then the integral of k_{se} in its first argument is

$$\begin{aligned}\bar{k}_{\text{se}}(x'|\lambda) &= \int_{-\infty}^{\infty} k_{\text{se}}(x, x'|\lambda)\, p(x)\, dx \\ &= \int_{l}^{u} \exp\big[-\frac{\lambda}{2}(x'-x)^2\big] \frac{1}{(u-l)} dx \\ &= \frac{1}{(u-l)} \sqrt{\frac{2\pi}{\lambda}} \Big[\Phi\left(\sqrt{\lambda}(u-x')\right) - \Phi\left(\sqrt{\lambda}(l-x')\right)\Big]\end{aligned}$$

where Φ is the CDF for a standard Gaussian distribution $\mathcal{N}(0, 1)$. Assuming that $x' \sim \mathcal{U}(l', u')$, the integral in both arguments is

$$\begin{aligned}\bar{\bar{k}}_{\text{se}}(\lambda) &= \int_{-\infty}^{\infty} \bar{k}_{\text{se}}(x'|\lambda)\, p(x')\, dx' \\ &= \frac{1}{(u-l)(u'-l')} \sqrt{\frac{2\pi}{\lambda}} \Big[\Big(\bar{\Phi}\left(\sqrt{\lambda}(u-u')\right) - \bar{\Phi}\left(\sqrt{\lambda}(u-l')\right)\Big) \\ &\qquad - \Big(\bar{\Phi}\left(\sqrt{\lambda}(l-u')\right) - \bar{\Phi}\left(\sqrt{\lambda}(l-l')\right)\Big)\Big]\end{aligned}$$

where

$$\bar{\Phi}(\eta) = \frac{-1}{\sqrt{\lambda}}\Big[\eta\Phi(\eta) + \phi(\eta)\Big]$$

and ϕ is the PDF of a standard Gaussian distribution $\mathcal{N}(0,1)$.

References

P.S. Craig, M. Goldstein, J.C. Rougier, and A.H. Seheult [2001]. "Bayesian Forecasting for Complex Systems Using Computer Simulators." *Journal of the American Statistical Association*, 96:717–729.

P.S. Craig, M. Goldstein, A.H. Seheult, and J.A. Smith [1997]. "Pressure Matching for Hydrocarbon Reservoirs: A Case Study in the Use of Bayes Linear Strategies for Large Computer Experiments." In C. Gatsonis, J.S. Hodges, R.E. Kass, R. McCulloch, P. Rossi, and N.D. Singpurwalla, editors, "Case Studies in Bayesian Statistics," volume 3, pages 36–93. New York: Springer-Verlag.

M. Goldstein and J. Rougier [2004]. "Probabilistic Formulations for Transferring Inferences from Mathematical Models to Physical Systems." *SIAM Journal on Scientific Computing*, 26:467–487. doi:10.1137/S106482750342670X.

M. Goldstein and J. Rougier [2006]. "Bayes Linear Calibrated Prediction for Complex Systems." *Journal of the American Statistical Association*, 475:1132–1143. doi:10.1198/016214506000000203.

M. Goldstein and J. Rougier [2009]. "Reified Bayesian Modelling and Inference for Physical Systems." *Journal of Statistical Planning and Inference*, 139:1221–1239. doi:10.1016/j.jspi.2008.07.019.

M. Goldstein and D. Wooff [2007]. Bayes Linear Statistics: Theory and Methods. Wiley.

B. Hirst, P. Jonathan, F. Gonzalez del Cueto, D. Randell, and O. Kosut [2012]. "Locating and Quantifying Gas Emission Sources Using Remotely Obtained Concentration Data." *Atmospheric Environment*, 45:141–158.

M.C. Kennedy and A. O'Hagan [2001]. "Bayesian Calibration of Computer Models." *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 63:425–464.

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

F. Pukelsheim [1994]. "The Three Sigma Rule." *The American Statistician*, 48:88–91.

C.E. Rasmussen and Z. Ghahramani [2002]. "Bayesian Monte Carlo." In "Advances in Neural Information Processing Systems," MIT Press.

C.E. Rasmussen and C.K.I. Williams [2006]. Gaussian Processes for Machine Learning. The MIT Press.

I. Vernon, M. Goldstein, and R.G. Bower [2010]. "Galaxy Formation: a Bayesian Uncertainty Analysis." *Bayesian Analysis*, 5:619–669. doi:10.1214/10-BA524.