

## Durham Research Online

---

### Deposited in DRO:

16 December 2018

### Version of attached file:

Published Version

### Peer-review status of attached file:

Peer-reviewed

### Citation for published item:

Jones, Matthew and Goldstein, Michael and Jonathan, Philip and Randell, David (2018) 'Bayes linear analysis of risks in sequential optimal design problems.', *Electronic journal of statistics.*, 12 (2). pp. 4002-4031.

### Further information on publisher's website:

<https://doi.org/10.1214/18-EJS1496>

### Publisher's copyright statement:

This article is published under a Creative Commons Attribution 4.0 International License.

### Additional information:

## Use policy

---

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a [link](#) is made to the metadata record in DRO
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full DRO policy](#) for further details.

# Bayes linear analysis of risks in sequential optimal design problems

Matthew Jones and Michael Goldstein

*Department of Mathematical Sciences, Durham University  
Lower Mountjoy, Stockton Road, Durham, UK, DH1 3LE  
e-mail: [m.j.jones@durham.ac.uk](mailto:m.j.jones@durham.ac.uk); [michael.goldstein@durham.ac.uk](mailto:michael.goldstein@durham.ac.uk)*

Philip Jonathan

*Department of Mathematics and Statistics, Lancaster University,  
LA1 4YW, United Kingdom*

*Shell Research Limited, London SE1 7NA, UK  
e-mail: [philip.jonathan@shell.com](mailto:philip.jonathan@shell.com)*

David Randell

*Shell Global Solutions International BV, Grasweg 31,  
1031 HW Amsterdam, NL  
e-mail: [david.randell@shell.com](mailto:david.randell@shell.com)*

**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 airborne 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 and phrases:** Bayesian uncertainty analysis, emulation, sequential experimental design, decision support, remote sensing, uncertainty quantification.

Received January 2018.

## Contents

1	Introduction . . . . .	4003
2	Sequential optimal design . . . . .	4005
2.1	Problem definition and notation . . . . .	4005
2.1.1	Decision problem . . . . .	4006
2.2	Extensive form – backward induction . . . . .	4007
3	Approximation of the design calculation . . . . .	4010
3.1	Overview of the procedure . . . . .	4011
3.2	Modelling the risk . . . . .	4012
3.3	Approximating the expected risk . . . . .	4013
3.4	Characterising the candidate design space . . . . .	4014
3.5	Stopping . . . . .	4014
3.6	Input selection for the next wave . . . . .	4016
4	Example: Linear model . . . . .	4017
4.1	Model and decision problem . . . . .	4017
4.2	Running the algorithm . . . . .	4018
4.2.1	First wave . . . . .	4018
4.2.2	Second wave . . . . .	4020
5	Example: Airborne sensing problem . . . . .	4021
5.1	Model specification . . . . .	4021
5.2	Design problem . . . . .	4022
5.3	Running the algorithm . . . . .	4024
5.3.1	First wave . . . . .	4024
5.3.2	Second wave . . . . .	4026
6	Discussion . . . . .	4028
	References . . . . .	4029
	Supplementary Material . . . . .	4031

## 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<sub>2</sub> 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 Jones et al. (2018); 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  $j^{\text{th}}$  set. We assume that a maximum of  $n$  experiments can be performed, and denote the  $j^{\text{th}}$  set of available observations by  $z_j = \{z_{j1}, \dots, z_{jn_{z_j}}\}$ ,  $j = 1 \dots n$ . We denote the collection of observations collected up to and including stage  $j$  by  $z_{[j]} = \{z_1, \dots, z_j\}$ . 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  $q = \{q_1, \dots, q_{n_q}\}$ .
- **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  $j^{\text{th}}$  set of observations by  $d_j = \{d_{j1}, \dots, d_{jn_{d_j}}\}$  (for  $d_j \in \mathcal{D}_j$ ), and we denote the collection up to and including stage  $j$  by  $d_{[j]} = \{d_1, \dots, d_j\}$ .
- **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  $j^{\text{th}}$  observation by  $w_j = \{w_{j1}, \dots, w_{jn_{w_j}}\}$  (for  $w_j \in \mathcal{W}_j$ ), and we denote the collection up to and including stage  $j$  by  $w_{[j]} = \{w_1, \dots, w_j\}$ .

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. Note that the numbers of observations, design inputs and external inputs ( $n_{z_j}$ ,  $n_{d_j}$  and  $n_{w_j}$ ) need not be the same, either within a particular stage or between stages; for example, in the atmospheric dispersion example presented in Section 5, a set of 5 design inputs and 2 external inputs affects the characteristics of a set of 100 observations to be collected at each stage.

### 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  $j^{\text{th}}$  experiment are summarised through the posterior distribution

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

where we specify the conditional distribution  $p(z_{[j]}|q, w_{[j]}, d_{[j]})$  for the observations  $z_{[j]}$  given the model parameters, and  $p(q)$  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  $a_j = \{a_{j1}, \dots, a_{jn_{a_j}}\}$  which might be taken at stage  $j$ .
- a loss function  $L_j(a_j, q)$  which describes (in utility units) the cost of taking action  $a_j$  at stage  $j$  (having terminated sampling) and then realising model parameters  $q$ .
- 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,

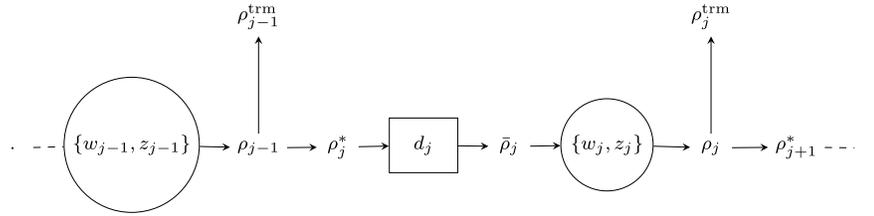


Fig 1: Graphical representation of the design procedure, showing quantities in the order that we choose, observe or compute them. Square nodes represent design parameters which we select, circular nodes represent random quantities which we observe when experimenting, and non-bordered nodes are risks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

as described by (1)

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

where  $\rho_j^{\text{trm}}$  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  $j = n, (n - 1), \dots, 1$ :

- We compute the overall risk, denoted by  $\rho_j$ , 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

$$\rho_n [z_{[n]}, w_{[n]}, d_{[n]}] = \rho_n^{\text{trm}} [z_{[n]}, w_{[n]}, d_{[n]}] \quad (3)$$

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

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

The risk  $\rho_{j+1}^*$  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  $\rho_j$  with respect to our current beliefs

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

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  $j^{\text{th}}$  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  $c_j(d_j)$  of experimentation

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

The minimum risk  $\rho_j^*$  is then

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

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  $\rho_1^*$ . To decide how to proceed, we compare this risk (from an optimal future procedure) with the risk  $\rho_0^{\text{trm}}$  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  $k^{\text{th}}$  experiment, then we assess the optimal course of action by comparing  $\rho_{k+1}^*$  with  $\rho_k^{\text{trm}}$ .

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’ or ‘myopic’ 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  $j^{\text{th}}$  and  $(j + 1)^{\text{th}}$  experiments are considered. Drovandi et al. (2013) use the ‘myopic’ approximation in conjunction with a sequential Monte-Carlo algorithm to find approximate solutions to the sequential design problem. Drovandi et al. (2014) adopt a similar approach which additionally incorporates model uncertainty.

Section 4.2 of the article by Huan and Marzouk considers an example which illustrates the drawbacks of using a myopic approximation in a sequential design problem. A vehicle is to be used to measure concentrations of a contaminant at a sequence of locations; the observation times are fixed, and so the vehicle is constrained in how far it can travel before it needs to make the next observation. A myopic design strategy chases after the next available measurement at the cost of potentially reducing the quality of the information available from future experiments. A fully sequential strategy, however, balances the expected quality of the immediately available measurement with the expected quality of future measurements, and so prevents the vehicle from moving to regions too far away from those where good-quality measurements might be made in forthcoming experiments.

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  $n, (n - 1), \dots, (j + 1)$  is accounted for in the risk function  $\rho_j$  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.

---

**Algorithm 1** Backward induction algorithm for sequential design problems
 

---

1: **for**  $k = 0, \dots, (n - 1)$  **do**  
 2:   **for**  $j = n, (n - 1), \dots, (k + 1)$  **do**  
 3:     Compute the risk  
 4:     **if**  $j = n$  **then**

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

5:     **else**

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

6:     **end if**

7:     Compute the expected risk

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

8:     Compute the optimal design, and corresponding risk

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

9:     **end for**

10:   **if**  $\rho_k^{\text{trm}} \leq \rho_{k+1}^*$  **then**  
 11:     Cease sampling, and take decision  $a_k^*$ .

12:     **Break**

13:   **else**

14:     Observe  $\{z_{k+1}, w_{k+1}\}$  at  $d_{k+1}^*$ .

15:   **end if**

16: **end for**

---

### 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 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  $(i)$ ,  $i = 1, 2, \dots$  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  $i = 1, 2, \dots$ , we iterate the following steps for  $j = n, (n - 1), \dots, 1$ :

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

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

where  $g_j^{(i)} = \{g_{j1}^{(i)}, \dots, g_{jn_{g^{(i)}}}^{(i)}\}$  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. For an illustration of the risk emulation procedure, see Section 4.2 and Section S3.1.1 of the supplementary material.

**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

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

The characterisation of beliefs about  $\bar{r}_j^{(i)}$  is discussed in Section 3.3. The computation of the expected risk for a simple example is considered in Section 4.2 and in Section S3.1.2 of the supplementary material.

**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  $\rho_j^*$  (defined in (6)) is denoted by  $s_j^{(i)}$ , with

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

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. This procedure is applied to a simple example in Section 4.2, with further details provided in Section S3.1.3 of the supplementary material.

---

**Algorithm 2** Approximation to the backward induction procedure.

---

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

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

- 4:     Approximate the expected risk

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

- 5:     Characterise the minimum risk

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

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

### 3.2. Modelling the risk

When stage  $j$  of the algorithm is reached, the risk  $\rho_j$  (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  $E[\beta_{jp}^{(i)}]$  for each of the regression coefficients, covariances  $\text{Cov}[\beta_{jp}^{(i)}, \beta_{kq}^{(i)}]$  between pairs of coefficients, and covariances  $\text{Cov}[u_j^{(i)}(\cdot), u_j^{(i)}(\cdot)]$  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  $R_j^{(i)} = \{R_{j1}^{(i)}, R_{j2}^{(i)}, \dots, R_{jN_j^{(i)}}^{(i)}\}$ ;  $R_{jk}^{(i)}$  is the  $k^{\text{th}}$  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  $\rho_n^{\text{trm}}$  (corresponding to the definition (3))

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

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

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

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  $E [R_{jk}^{(i)}]$  and  $\text{Cov} [R_{jk}^{(i)}, R_{jl}^{(i)}]$  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  $E_{R_j^{(i)}} [r_j^{(i)} [z_{[j]}, w_{[j]}, d_{[j]}]]$  and covariances  $\text{Cov}_{R_j^{(i)}} [r_j^{(i)} [z_{[j]}, w_{[j]}, d_{[j]}], r_j^{(i)} [z'_{[j]}, w'_{[j]}, d'_{[j]}]]$  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

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

where  $E_{R_j^{(i)}} [r_j^{(i)}]$  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

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

where  $\text{Cov}_{R_j^{(i)}} \left[ r_j^{(i)} [ \cdot ], r_j^{(i)} [ \cdot' ] \right]$  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  $p(z_j | \dots)$  and  $p(w_j | \dots)$ ; 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  $\rho_j^*$  from an optimal design at stage  $j$ , which will then be used as an input to the  $(j - 1)$ <sup>th</sup> stage of the algorithm (equation (7), Section 3.2), or to select a design for the  $j$ <sup>th</sup> 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  $E \left[ s_j^{(i)} [ \cdot ] \right]$  and  $\text{Cov} \left[ s_j^{(i)} [ \cdot ], s_j^{(i)} [ \cdot' ] \right]$  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.

### 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  $p(q | z_{[k-1]}, w_{[k-1]}, d_{[k-1]})$  about the model parameters, and an experiment at stage  $k$  using a design which we

---

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

---

- 1: Generate  $M_j^{(i)}$  space-filling trial designs  $\{d_{j1}, d_{j2}, \dots, d_{jM_j^{(i)}}\}$  within the candidate space  $\mathcal{D}_j^{(i-1)}$
- 2: Jointly sample  $\tilde{r}_j^{(i)} [z_{[j-1]}, w_{[j-1]}, \{d_{[j-1]}, d_{jk}\}]$  over the set of all trial designs  $d_{jk}$  from a Gaussian distribution
- 3: Set

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


---

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  $d_{k-1}^*$  (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  $k^{\text{th}}$  experiment. We denote the chosen design by  $\hat{d}_k$ , and select  $\hat{d}_k$  to minimise our expected risk

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

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

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

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

**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  $\tilde{r}_k^{(i)}$  exactly, and we also knew the

optimal design setting  $d_k^*$ , then the risk from an optimal course of action would be

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

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  $k^{\text{th}}$  experiment; in this situation, our expectation for the loss incurred by choosing to experiment at  $\hat{d}_k$  rather than at  $d_k^*$  is

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

The expectation of the second term is approximated by sampling candidate designs  $\hat{d}_k$  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  $k^{\text{th}}$  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

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

where  $d_j$  is a scalar design input,  $h_j(d_j) = (h_{j1}(d_j), \dots, h_{jn_{h_j}}(d_j))^T$  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  $q = (q_1, \dots, q_{n_{h_j}})^T$  has a multivariate Gaussian prior distribution ( $q \sim \mathcal{N}(\mu_q, V_q)$ ), and that the errors  $\epsilon_j$  are independent, with zero-mean Gaussian distributions ( $\epsilon_j \sim \mathcal{N}(0, v_{\epsilon_j})$ ). We specify that our losses at all stages depend on the value of a new observation  $\hat{z}(q) = h(\hat{d})^T q + \hat{\epsilon}$  at a known location  $\hat{d}$  as

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

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

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

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

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

$$\hat{\mu}_q(z_{[j]}, d_{[j]}) = \hat{V}_q(d_{[j]}) \left[ V_q^{-1} \mu_q + \frac{1}{v_\epsilon} H(d_{[j]})^\top z_{[j]} \right]$$

where  $H(d_{[j]})$  is the design matrix created by stacking the vectors  $h_j(d_j)$  as rows, and  $\hat{\mu}_q(z_{[j]}, d_{[j]})$  and  $\hat{V}_q(d_{[j]})$  are the parameters of the posterior Gaussian distribution  $p(q|z_{[j]}, d_{[j]})$ .

If we differentiate the integral from (13) with respect to  $a_j$  and set to zero, we find that the optimal decision is

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

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

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

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

$$h_j(d_j) = (1, (1 + d_j)(1 - d_j))^\top$$

The prior parameters of  $p(q)$  are fixed to  $\mu_q = (0, 0)^\top$  and  $V_q = \text{diag}(0.5^2, 0.5^2)$ , 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  $l(\hat{z}) = 1 + \hat{z}^2$ , so that  $l(\hat{z}) > 0$  everywhere. The observation costs are set to be constant, with  $c_1(d_1) = 0.05$  and  $c_2(d_2) = 0.2$ , so that the second measurement is also more expensive.

##### 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  $\rho_j^{\text{trm}}$  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

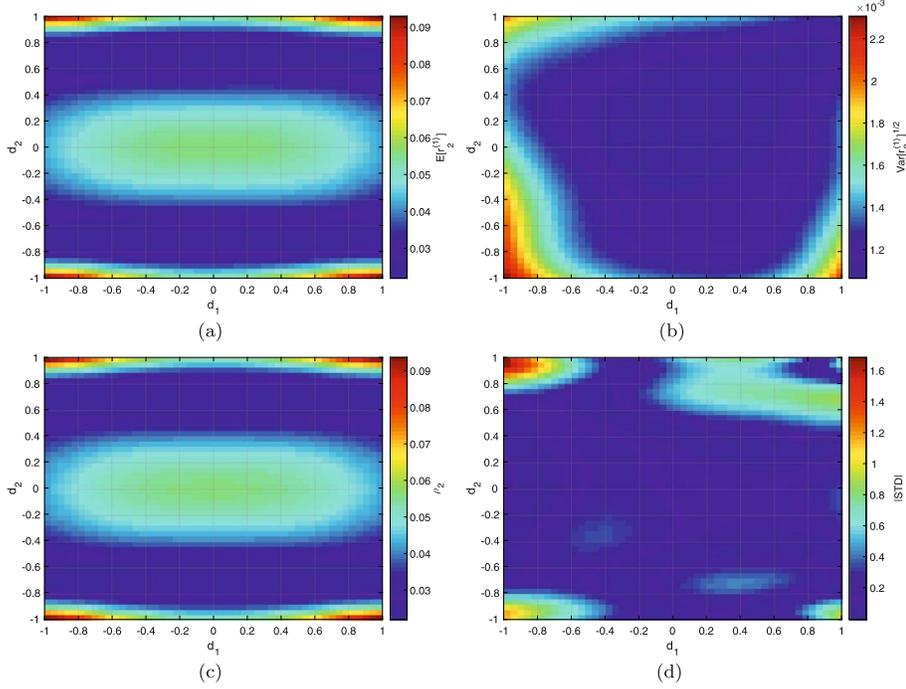


Fig 2: Emulator  $r_2^{(1)}$  fitted to the risk at stage 2. Figure 2(a) shows the adjusted expectation  $E_{R_2^{(1)}}[r_2^{(1)}]$  for a grid of  $d_1$  and  $d_2$  values, and Figure 2(b) shows the corresponding standard deviation values  $\text{Var}_{R_2^{(1)}}[r_2^{(1)}]^{1/2}$ . Figure 2(c) shows the true value of the risk  $\rho_2$  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.

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  $E_{R_2^{(1)}}[r_2^{(1)}]$  and standard deviation  $\text{Var}_{R_2^{(1)}}[r_2^{(1)}]^{1/2}$  surfaces of the emulator respectively, Figure 2(c) shows the true risk  $\rho_2$  which the model is designed to represent, and Figure 2(d) shows the standardised discrepancy  $(\rho_2 - E_{R_2^{(1)}}[r_2^{(1)}]) / \text{Var}_{R_2^{(1)}}[r_2^{(1)}]^{1/2}$  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.

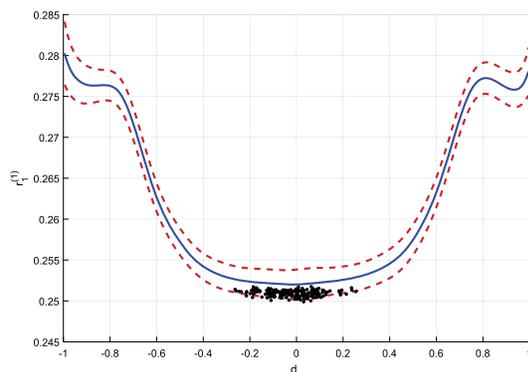


Fig 3: Emulator  $\tilde{r}_1^{(1)}$ , with  $E_{R_1^{(1)}}[\tilde{r}_1^{(1)}]$  shown in solid blue and  $E_{R_1^{(1)}}[\tilde{r}_1^{(1)}] \pm 3\text{Var}_{R_1^{(1)}}[\tilde{r}_1^{(1)}]^{1/2}$  shown in dashed red. Also shown (as black markers) are 200 candidate design-risk sample pairs  $(\tilde{d}_1,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

**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  $E_{R_1^{(1)}}[\tilde{r}_1^{(1)}] + c_1(\hat{d}_1) = 0.2520$  and  $\text{Var}_{R_1^{(1)}}[\tilde{r}_1^{(1)}] = (0.0006)^2$  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  $\tilde{r}_1^{(1)}$  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  $E[\tilde{r}_1^{(2)}] + c_1(\hat{d}_1) = 0.2524$

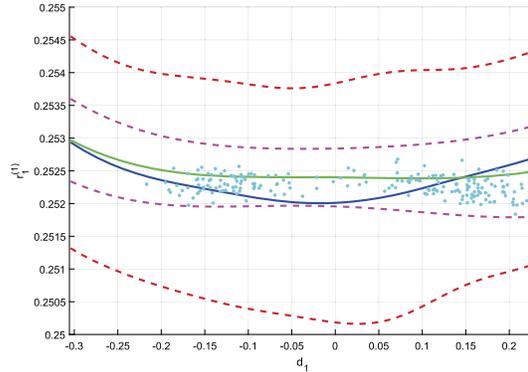


Fig 4: Emulators  $\bar{r}_1^{(i)}$  for both waves  $i = 1, 2$  within the candidate design space  $\mathcal{D}_1^{(1)}$ ;  $E_{R_1^{(i)}}[\bar{r}_1^{(i)}]$  is shown as a solid line, and  $E_{R_1^{(i)}}[\bar{r}_1^{(i)}] \pm 3\text{Var}_{R_1^{(i)}}[\bar{r}_1^{(i)}]^{1/2}$  is dashed (blue/red for wave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

and  $\text{Var}[\bar{r}_1^{(2)}] = (0.0002)^2$ . 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 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: Airborne 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  $x = (x_x, x_y, x_h)^T$  and the location of a single source by  $c = (c_x, c_y, c_h)^T$ ; we project the source-observation vector onto the wind direction vector  $w = (w_x, w_y)$  as follows

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

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

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

$\sigma = (\sigma_y, \sigma_h)^T$  are horizontal and vertical plume standard deviations, which depend on the downwind distance from source to measurement as  $\sigma_y = \omega_y \tan(\gamma_y)$  and  $\sigma_h = \omega_h \tan(\gamma_h)$ , where  $\gamma = (\gamma_y, \gamma_h)^T$  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

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

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 specify that each flight will consist of  $n_f$  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  $n_{\text{ob}}$  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

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

where  $d_j = \{d_{jx}, d_{jy}, d_{jh}, d_{jw}, d_{jd}\}$  is the five-dimensional design input setting at the  $j^{\text{th}}$  stage,  $w_j = \{w_{jx}, w_{jy}\}$  is the two dimensional wind field input (the external parameter set) at stage  $j$ , and  $\epsilon_j$  is a  $(n_{\text{fl}} \times n_{\text{ob}})$  dimensional vector of uncorrelated measurement errors. In general, if there were further systematic effects in the concentration profile not related to the source contributions (e.g. a smoothly-varying background concentration), we would include additional terms in equation (15). We assume that the wind parameters are independently uniformly distributed at each stage, over the following ranges (all units are metres/second):

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

These prior distributions are chosen to give an example in which the prevailing wind direction is different at all stages. When designing for data collection over a real region, suitable prior distributions could be constructed from historic measurements of the wind field, or from wind data collected immediately prior to the measurement campaign.

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

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

where  $q = \{\psi_1, \dots, \psi_{n_s}\}$  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

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

In this example, 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; under this assumption, the conditional variances in equation (16) are simply equivalent to the adjusted variances  $\text{Var}_{z_{[j]}} [q_k | w_{[j]}, d_{[j]}]$ . For further discussion of this point, see Section S4.1 of the supplementary material.

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

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

where  $c_{\text{set}}$  is the constant setup cost,  $c_{\text{dst}}$  is the cost per unit distance, and  $x_0 = (x_{0x}, x_{0y}, 0)^T$  is the location of the airport from which the aircraft takes off. 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  $\{E_{z_{[3]}} [q_k]\}$  and  $\{\text{Var}_{z_{[3]}} [q_k]^{1/2}\}$  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

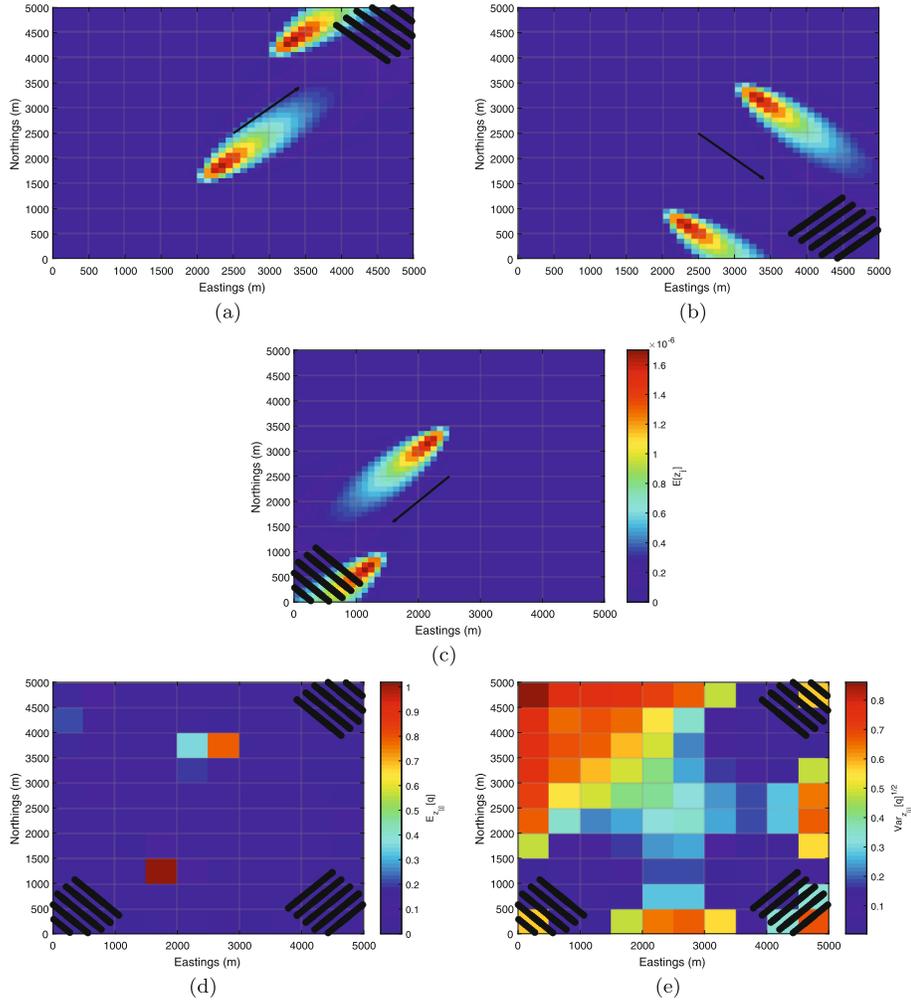


Fig 5: An example inference using the model from Sections 5.1 and 5.2. Figures 5(a) to 5(c) show the expected concentration measurements for a grid of points in  $(x, y)$  space, for an emission rate of  $\psi_k = 1$  at both sources (locations indicated by magenta markers), under the wind conditions 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  $E_{z[3]}[q]$  and  $\text{Var}_{z[3]}[q]$  computed by updating the prior specification from Section 5.1 using this observed data.

which minimises the risk over these points. Our expected risk at this point is  $E_{R_1^{(1)}} \left[ \bar{r}_1^{(1)} \left[ \hat{d}_1 \right] \right] + c_1 \left( \hat{d}_1 \right) = 32.36$ , with variance  $\text{Var}_{R_1^{(1)}} \left[ \bar{r}_1^{(1)} \left[ \hat{d}_1 \right] \right] = (1.19)^2$ ; this is compared to the risk from an optimal decision under the prior, which is  $\rho_0^{\text{trm}} = 196.34$ . From this, it is clear that, based on this analysis, we should

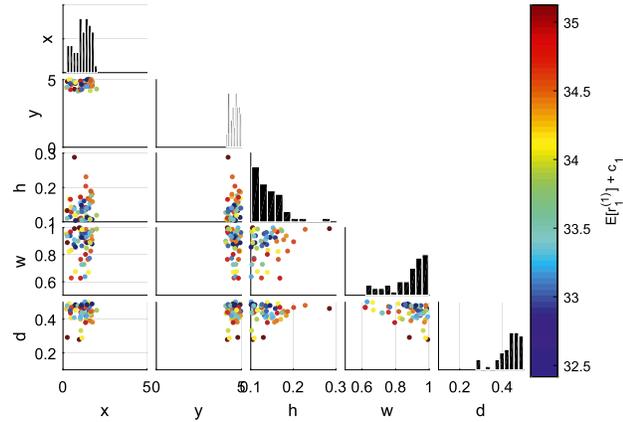


Fig 6: Scatter plot of candidate design points  $\tilde{d}_1$  sampled from emulator  $\tilde{r}_1^{(1)}$  using the procedure 3. The colour scale indicates the expected risk  $E_{R_1^{(1)}} \left[ \tilde{r}_1^{(1)} \right] + c_1(\tilde{d}_1)$  at each point.

conduct at least the first experiment.

Based on a set of 100 samples from the candidate design space, we assess 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 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  $E_{R_1^{(2)}} \left[ \tilde{r}_1^{(2)} \right] + c_1(\tilde{d}_1)$  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.

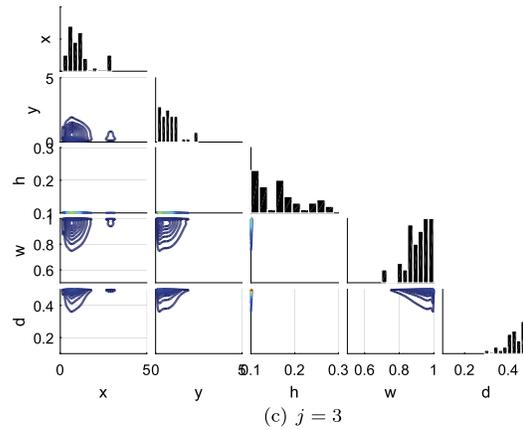
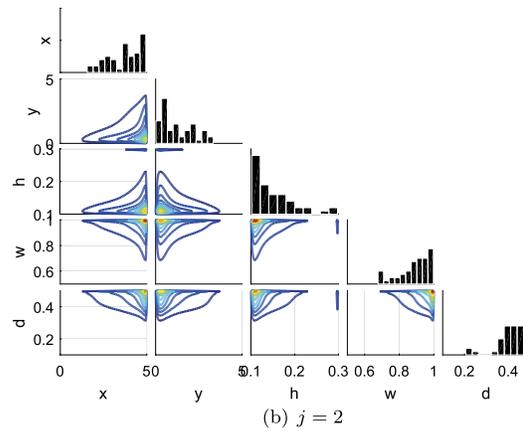
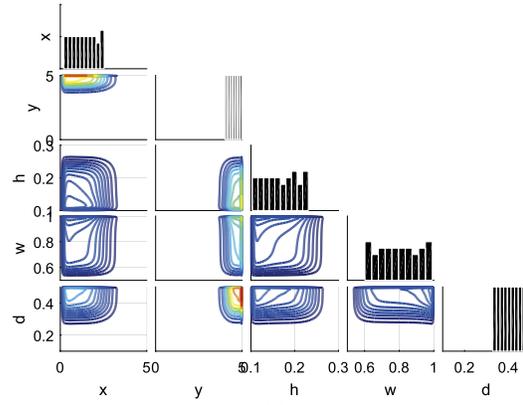


Fig 7: Density contour plots of the candidate design spaces  $\mathcal{D}_j^{(1)}$  at each stage  $j = 1, 2, 3$ ; designing within the candidate design space at each stage restricts the candidate design space at each subsequent wave. Red contours enclose regions with higher densities of sampled points,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

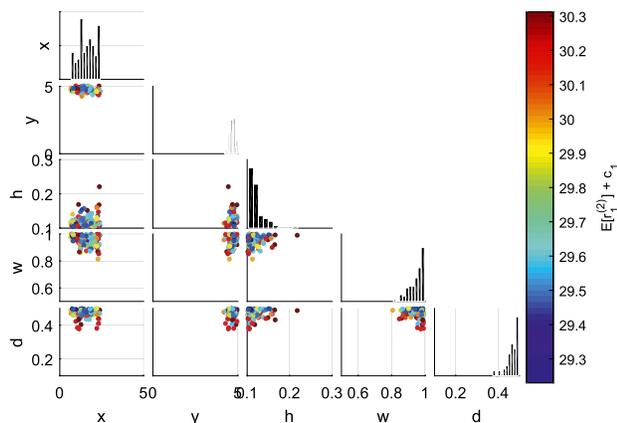


Fig 8: Scatter plot of  $\tilde{d}_1$  sampled from  $\tilde{r}_1^{(2)}$  using the procedure 3. Colours indicate  $E_{R_1^{(2)}} \left[ \tilde{r}_1^{(2)} \right] + c_1(\tilde{d}_1)$  at each point.

**Stopping** At the end of this wave, we again search the mean risk using a Latin hypercube of 2000 points, and find that  $E_{R_1^{(2)}} \left[ \tilde{r}_1^{(2)} \left[ \hat{d}_1 \right] \right] + c_1(\hat{d}_1) = 29.21$ , with  $\text{Var}_{R_1^{(2)}} \left[ \tilde{r}_1^{(2)} \left[ \hat{d}_1 \right] \right]^{1/2} = (0.35)^2$ . 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 approximation proves beneficial in application to both a simple linear model example, and to a more complex atmospheric dispersion modelling problem.

In the work reported here, we have considered problems with up to 3 potential future experiments. This involved hours of computation on a reasonably powerful laptop. In general, we might want to consider problems with many more stages in the backward induction calculation. For problems with reason-

ably high-dimensional collections of data, design inputs and external inputs at each stage, computational complexity may become prohibitive. There is a trade-off between the per-stage dimensionality of the design problem to be solved and the number of stages that can be handled using our approximation: for problems with lower-dimensional input spaces, we can potentially consider a larger number of future experiments; whereas for problems with higher-dimensional input spaces, we may be restricted to a smaller number of stages.

This work 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. *The Geometry of Random Fields*. Wiley, 1981. [MR0611857](#)
- J.O. Berger. *Statistical Decision Theory*. Springer, 1980. [MR0580664](#)
- K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10: 273–304, 1995. [MR1390519](#)
- M.H. DeGroot. *Optimal Statistical Decisions*. Wiley, 1970. [MR2288194](#)
- C.C. Drovandi, J.M. McGree, and A.N. Pettitt. Sequential Monte Carlo for Bayesian sequentially designed experiments for discrete data. *Comput. Stat. Data Anal.*, 57: 320–335, 2013. [MR2981091](#)

- C.C. Drovandi, J.M. McGree, and A.N. Pettitt. A Sequential Monte Carlo Algorithm to Incorporate Model Uncertainty in Bayesian Sequential Design. *Journal of Computational and Graphical Statistics*, 23: 3–24, 2014. doi: [10.1080/10618600.2012.730083](https://doi.org/10.1080/10618600.2012.730083). MR3173758
- I. Ford, D.M. Titterton, and C.P. Kitsos. Recent advances in nonlinear experimental design. *Technometrics*, 31 (1): 49–60, 1989. ISSN 00401706. URL <http://www.jstor.org/stable/1270364>.
- M. Goldstein and J. Rougier. Reified bayesian modelling and inference for physical systems. *Journal of Statistical Planning and Inference*, 139: 1221–1239, 2009. doi: [10.1016/j.jspi.2008.07.019](https://doi.org/10.1016/j.jspi.2008.07.019). MR2479863
- M. Goldstein and A. Seheult. Prior Viability Assessment for Bayesian Analysis. *Journal of Statistical Planning and Inference*, 138 (5): 1271–1286, 2008. ISSN 03783758. doi: [10.1016/j.jspi.2007.04.023](https://doi.org/10.1016/j.jspi.2007.04.023).
- P. Hennig and C.J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13: 1809–1837, 2012. MR2956343
- B. Hirst, P. Jonathan, F. Gonzalez del Cueto, D. Randell, and O. Kosut. Locating and quantifying gas emission sources using remotely obtained concentration data. *Atmospheric Environment*, 45: 141–158, 2012.
- X. Huan and Y. Marzouk. Sequential bayesian optimal experimental design via approximate dynamic programming. *Forthcoming*, 2016. URL <https://arxiv.org/abs/1604.08320>.
- D.R. Jones, M. Schonlau, and W.J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13: 455–492, 1998. MR1673460
- M. Jones, M. Goldstein, P. Jonathan, and D. Randell. Supplementary Material: Bayes Linear Analysis of Risks in Sequential Optimal Design Problems. doi: [10.1214/18-EJS1496SUPP](https://doi.org/10.1214/18-EJS1496SUPP).
- M. Jones, M. Goldstein, D. Randell, and P. Jonathan. Bayes linear analysis for bayesian optimal design. *Journal of Statistical Planning and Inference*, 171: 115–129, 2016. doi: [10.1016/j.jspi.2015.10.011](https://doi.org/10.1016/j.jspi.2015.10.011). MR3458072
- M.C. Kennedy and A. O’Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 63: 425–464, 2001. MR1858398
- D.V. Lindley. *Bayesian Statistics, A Review*. Society for Industrial and Applied Mathematics, 1972. MR0329081
- P. Muller, D.A. Berry, A.P. Grieve, M. Smith, and M. Krams. Simulation based sequential bayesian design. *Journal of Statistical Planning and Inference*, 137: 3140–3150, 2007. doi: [10.1016/j.jspi.2006.05.021](https://doi.org/10.1016/j.jspi.2006.05.021).
- A. O’Hagan. Bayes-hermite quadrature. *Journal of Statistical Planning and Inference*, 29: 245–260, 1991. MR1144171
- F. Pasquill. Atmospheric dispersion of pollution. *Quarterly Journal of the Royal Meteorological Society*, 97: 369–395, 1971.
- D. Randell, M. Goldstein, G. Hardman, and P. Jonathan. 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, 2010. doi: [10.1243/1748006XJRR322](https://doi.org/10.1243/1748006XJRR322).
- C.E. Rasmussen and Z. Ghahramani. Bayesian monte carlo. In *Advances in Neural Information Processing Systems*. MIT Press, 2002.
- J.Q. Smith. *Bayesian Decision Analysis: Principles and Practice*. Cambridge University Press, 2010. [MR2828346](https://doi.org/10.1017/CBO9780511527018)
- J.M. Stockie. The mathematics of atmospheric dispersion modelling. *SIAM Review*, 53: 349–372, 2011. [MR2806641](https://doi.org/10.1137/09M1086641)
- I. Vernon, M. Goldstein, and R.G. Bower. Galaxy formation: a bayesian uncertainty analysis. *Bayesian Analysis*, 5: 619–669, 2010. doi: [10.1214/10-BA524](https://doi.org/10.1214/10-BA524). [MR2740148](https://doi.org/10.1214/10-BA524)
- D. Williamson and M. Goldstein. Bayesian policy support for adaptive strategies using computer models for complex physical systems. *Journal of the Operational Research Society*, 63: 1021–1033, 2012. doi: [10.1057/jors.2011.110](https://doi.org/10.1057/jors.2011.110).
- D. Williamson, M. Goldstein, L. Allison, A. Blaker, P. Challenor, L. Jackson, and M. Yamazaki. 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, 2013. ISSN 09307575. doi: [10.1007/s00382-013-1896-4](https://doi.org/10.1007/s00382-013-1896-4).

## Supplementary Material

### Supplementary Material: Bayes Linear Analysis of Risks in Sequential Optimal Design Problems

(doi: [10.1214/18-EJS1496SUPP](https://doi.org/10.1214/18-EJS1496SUPP); .pdf).